

Hybrid Drowsy SRAM and STT-RAM Buffer Designs for Dark-Silicon-Aware NoC

Jia Zhan, *Student Member, IEEE*, Jin Ouyang, *Member, IEEE*, Fen Ge, *Member, IEEE*,
Jishen Zhao, *Member, IEEE*, and Yuan Xie, *Fellow, IEEE*

Abstract—The breakdown of Dennard scaling prevents us from powering all transistors simultaneously, leaving a large fraction of dark silicon. This crisis has led to innovative work on power-efficient core and memory architecture designs. However, the research for addressing dark silicon challenges with network-on-chip (NoC), which is a major contributor to the total chip power consumption, is largely unexplored. In this paper, we comprehensively examine the network power consumers and the drawbacks of the conventional power-gating techniques. To overcome the dark silicon issue from the NoC's perspective, we propose DimNoC, a dim silicon scheme, which leverages recent drowsy SRAM design and spin-transfer torque RAM (STT-RAM) technology to replace pure SRAM-based NoC buffers. In particular, we propose two novel hybrid buffer architectures: 1) a hierarchical buffer architecture, which divides the input buffers into a set of levels with different power states and 2) a banked buffer architecture, which organizes the drowsy SRAM and the STT-RAM in different banks, and accesses them in an interleaved fashion to hide the long write latency of STT-RAM. In addition, our hybrid buffer design enables NoC data retention mechanism by storing packets in drowsy SRAM and nonvolatile STT-RAM in a lossless manner. Combined with flow control schemes, the NoC data retention mechanism can improve network performance and power simultaneously. Our experiments over real workloads show that DimNoC can achieve 30.9% network energy saving, 20.3% energy-delay product reduction, and 7.6% router area reduction compared with pure SRAM-based NoC design.

Index Terms—Dark silicon, drowsy cache, network-on-chip (NoC), power-gating, spin-transfer torque RAM (STT-RAM).

I. INTRODUCTION

DENNARD scaling (scaling down of the supply voltage [16]), which has offered us near-constant chip power with doubling number of transistors, has come to an end [53]. Computer designers are seeking ways to stay on the performance curve using emerging many-core processors without exceeding the thermal design power (TDP). Dynamic voltage/frequency scaling of cores can mitigate

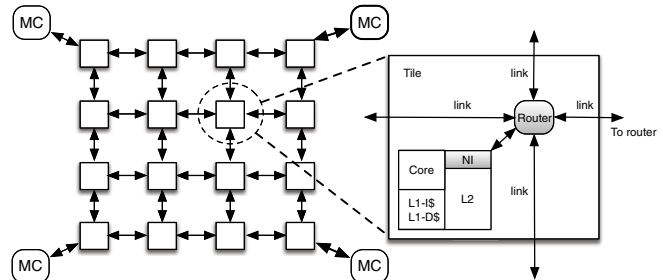


Fig. 1. 4×4 NoC-based multicore architecture. In each node, the local processing elements (core, L1, L2 caches, and so on) are attached to a router through an NI. Routers are interconnected through links to form an NoC. Four memory controllers are attached at the four corners, which will be used for off-chip memory access.

the issue. However, many-core processors are becoming to incorporate more transistors than those can remain powered up simultaneously. Recent studies refer to the fraction of chip area which is entirely powered-OFF as dark silicon [11], [15], [19], [45], [53].

To tackle the challenges of many-core scaling in the dark silicon era, researchers have proposed various approaches [42], [49], [58] to improve the power efficiency of the whole chip. However, most of the previous research efforts focus on core or memory subsystem optimizations—dark-silicon-aware on-chip interconnects design has not received sufficient attention. Fig. 1 shows a 4×4 network-on-chip (NoC)-based multicore architecture. NoC critically impacts the overall performance of many-core processors and consumes a significant portion of total power budget. Prior research and industrial prototypes have validated that about 10%–36% [5], [22], [50] of total chip power is consumed by NoC. Moreover, as the majority of on-chip core/cache components may be put in the dormant mode by aggressive power-gating/clock-gating, NoC will become a potential major power contributor at runtime due to its significant leakage power [57].

Driven by these observations, there are increasing research efforts to aggressively shut down parts of the NoC, leveraging the conventional power-gating [41] techniques to minimize idle power. NoC power-gating is heavily dependent on the traffic. In order to benefit from power-gating, an adequate idle period (namely, break-even time) of routers should be guaranteed to ensure that they are not frequently woken up and gated-off. Various schemes [8], [13], [31], [44], [57] have been proposed to maximize the benefits of power-gating.

An efficient NoC power-management technique should embrace power-gating for maximum power saving, but the

Manuscript received August 4, 2015; revised December 5, 2015; accepted January 12, 2016.

J. Zhan and Y. Xie are with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93111 USA (e-mail: jzhan@ece.ucsb.edu; yuanxie@ece.ucsb.edu).

J. Ouyang is with NVIDIA Corporation, Santa Clara, CA 95050 USA (e-mail: jouyang@nvidia.com).

F. Ge is with the Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: gefen@nuaa.edu.cn).

J. Zhao is with the Department of Electrical and Computer Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064 USA (e-mail: jishen.zhao@ucsc.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVLSI.2016.2536747

major penalties (generally takes 10–20 cycles [8], [13], [44] depending on the frequency) for waking up the gated blocks are undesirable. To achieve a better tradeoff between leakage saving and wake-up penalty, we explore the replacement of the conventional NoC buffers by drowsy SRAM, which applies similar circuitry in drowsy cache [17] to introduce an intermediate sleep mode between power-ON state and power-gating state.

In addition to power-management techniques, emerging nonvolatile memory (NVM) technologies entails new opportunities to enhance NoC power efficiency. Since buffer is the dominant leakage consumer in NoC, it may be constituted by NVM, whose leakage power is considerably lower than the conventional SRAM. Among alternative NVM technologies, such as phase-change memory (PCM), spin-transfer torque RAM (STT-RAM), and resistive RAM, STT-RAM is particularly promising to replace NoC buffers, because STT-RAM combines the speed of SRAM, the density of DRAM, and the nonvolatility of flash memory, with low-leakage power consumption. Moreover, STT-RAM shows higher endurance [7], [23] compared with other NVM, which make it more attractive to design on-chip buffers that must endure frequent packet accesses.

Therefore, by replacing the conventional SRAM buffers with drowsy SRAM, we can significantly reduce the wake-up penalty of sleeping routers; by replacing the conventional SRAM buffers with STT-RAM, we can benefit from the low-leakage nature of STT-RAM even under normal operations and thus avoid unnecessary power-gating of routers. However, the long write latency of STT-RAM is still undesirable. In order to design a general power-management scheme for NoC that can robustly adapt to sporadic traffic patterns, we take advantage of both drowsy SRAM and STT-RAM techniques and propose a novel architecture design called DimNoC. It explores two candidates to replace the conventional NoC buffers: drowsy SRAM and nonvolatile STT-RAM, and designs two novel hybrid buffer architectures for the best NoC power efficiency. The rational behind the hybrid organization is that, we can use drowsy SRAM to reduce the wake-up penalty of every sleeping block, and leverage STT-RAM to further reduce the number of unnecessary power-gating operations, and, thus, reduce the number of high-cost wake-up operations. Moreover, the dense STT-RAM cells compensate for the extra area overhead of drowsy circuit and control logic.

In particular, the first hybrid buffer design is called hierarchical buffer (HB), in which drowsy SRAM and STT-RAM comprise separate virtual channels (VCs) and form multiple levels. Depending on the traffic load, different levels are activated successively. STT-RAM is designed to be lastly activated to avoid frequent long-latency write accesses. The second design is called banked buffer (BB), in which drowsy SRAM buffers and STT-RAM buffers are logically interleaved within each VC, but physically separated into multiple banks to hide the long write latency of STT-RAM.

Another important reason of integrating drowsy SRAM and STT-RAM is that both of them can be put into retention state because of their drowsy property and nonvolatility. The prior hybrid designs only put these RAMs in sleeping state at

low traffic load. However, we discover that by temporarily stalling some router input ports and put their buffers in the retention state in the case of traffic hot spots, we can not only reduce the network congestion but also significantly conserve network power. In the meantime, those stalled packets can be losslessly preserved in low power or even zero leakage state, and resume transmission when the congestion is relieved.

In particular, this paper makes the following contributions.

- 1) Comprehensively analyzes the implications of dark silicon crisis on NoC architecture design and reveals inefficiency of the conventional power-gating techniques.
- 2) Proposes DimNoC, which uses a dim silicon approach to tackle the dark silicon crisis from the NoC's perspective. In particular, it investigates drowsy SRAM and non-volatile STT-RAM as replacements for the conventional NoC buffers.
- 3) Proposes two novel hybrid drowsy SRAM/STT-RAM buffer designs: HB and BB. The BB design can seamlessly hide the multicycle write delay.
- 4) The first work to propose the concept and application of data retention in NoC, enabled by both drowsy SRAM and nonvolatile STT-RAM.

II. CHALLENGES AND OPPORTUNITIES

A. Router Power Analysis

To better understand the power distribution of a router, we simulate a classic wormhole router with Design Space Exploration for Network Tool (DSENT) [47] and plot the power breakdown of both dynamic power and leakage power under process technologies of 45, 32, and 22 nm. We use a frequency of 1 GHz, and the corresponding supply voltages for different technologies are, by default, 1, 0.9, and 0.8 V, respectively. For sensitivity studies, we vary the supply voltage around the default values. The flit (the basic flow control unit. A packet may contain multiple flits) width is set to be 128 bits. Each input port of a router comprises two VCs, and each VC is 4 flit in depth. As a motivational example, we conservatively set the traffic load to be 0.3 flits/cycle on average in order to calculate dynamic power. Note that real workloads usually demonstrate lower traffic load to the on-chip network [35].

As shown in Fig. 2(b), with each process technology, the percentage of leakage power increases as the supply voltage reduces. In addition, as technology scales from 45 to 22 nm, the ratio of leakage power increases and substantially outweighs that of dynamic power. For example, leakage power is around 63.4% of total power with a supply voltage of 0.9 V at 32-nm technology. These results also reveal the power crisis in the dark silicon era due to the increasing leakage power. The breakdown of dynamic and leakage power in different router components is shown in Fig. 2(c) and (d) (in 32-nm technology with a supply voltage of 0.9 V), respectively. An important observation is that buffer power contributes a dominant portion to the router power, especially in leakage power. In summary, Fig. 2 reveals that leakage power dominates the power budget in the dark silicon era and buffer becomes the primary power consumer for NoC.

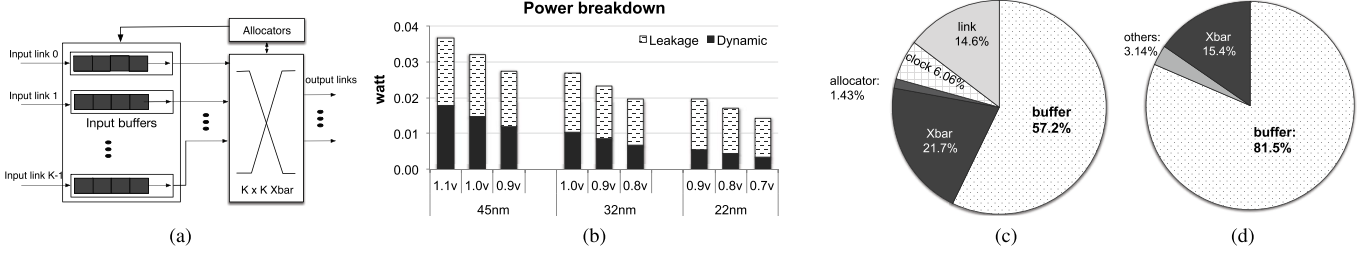


Fig. 2. Dynamic and leakage power breakdown of a VC-based router. The router is operated at 1 GHz. Each input port consists of two VCs with each VC capable of storing four 128-bit flits. (a) Router microarchitecture. (b) Router power breakdown when varying the operating voltage and frequency. (c) Dynamic power breakdown. (d) Leakage power breakdown.

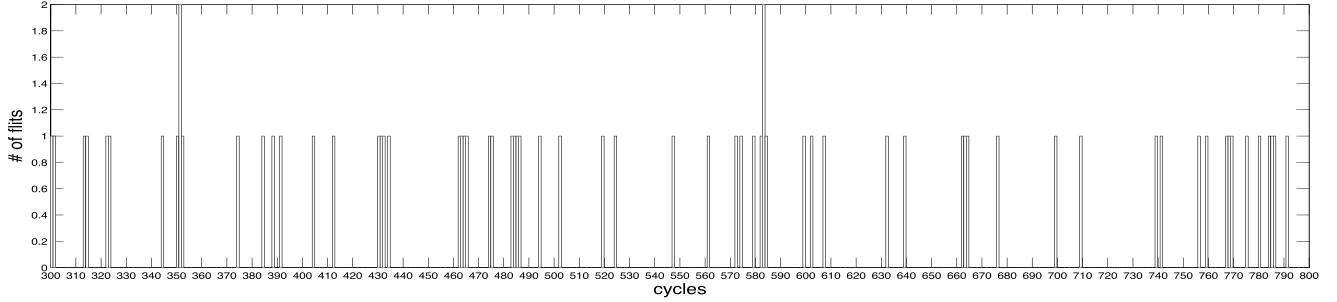


Fig. 3. Snapshot of the arrival patterns of network packets for a router, running application disparity on a 16-core system.

B. Limitation of Conventional Power-Gating

In the dark silicon background, as a significant percent of transistors cannot be switched ON due to TDP limit, researchers start to explore power-gating on NoC. In particular, they propose to completely shut down idle routers or links and then wake them up when new packets arrive. Power-gating is implemented by inserting appropriately sized transistor(s) with high threshold voltage (nonleaky sleep switch) between V_{dd} and the block. Therefore, the entire router block can be switched between ON-state and OFF-state by asserting and deasserting the sleep signal.

However, applying power-gating to on-chip routers has been elusive due to several fundamental concerns: from the energy's perspective, switching router OFF and ON would incur energy overhead. Thus, packet interarrival time has to be long enough in order to compensate for the corresponding overhead. From the performance's perspective, the wake-up delay of a gated router is significant, and may incur considerable latency overhead if routers are frequently switched OFF and ON.

Therefore, the benefit of applying power-gating on NoC is largely traffic dependent. As a motivational example, we run a real workload (disparity) on an NoC-based 16-core system.¹ A snapshot of the flit arrival timestamps for a specific router is shown in Fig. 3. Here, we conservatively consider the break-even time of routers as 10 cycles [8], [13], [44]. As we can see, there are a lot of short intervals between consecutive flit arrivals which are less than 10 cycles. There are also cases when multiple flits arrive simultaneously, as the long beam between 580 and 590 cycles indicates.

¹The detailed experimental setup is described in Section V.

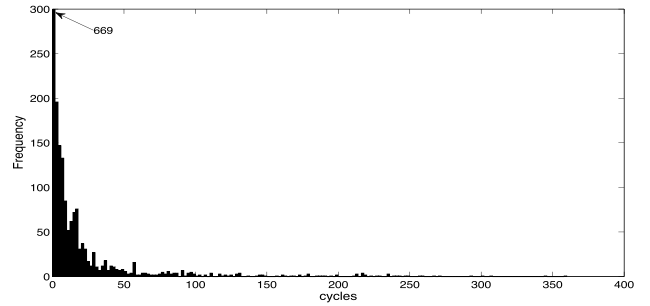


Fig. 4. Histogram of flit interarrival time for a router on a 16-core system with a real workload (disparity) running.

Furthermore, we conduct statistic analysis on the intervals of consecutive flit arrivals. The histogram in Fig. 4 shows two important observations. First, some long interarrival periods exist in the network which imply the opportunity of power-gating. Second, most intervals are less than 20 cycles, and among them, a significant portion is <2 cycles.

Consequently, the conventional power-gating techniques will be detrimental to the network performance because of frequent ON–OFF-state transitions of network routers, and even offset the power saving gained from long idle intervals.

C. Fine-Grained Power Management

The conventional power-gating techniques shut down a router only when it is completely idle, i.e., all the packets in buffers and other pipelines are depleted, which limits its application in skewed or heavy-loaded network traffic. Correspondingly, the whole router needs to be woken up for a new access, which incurs substantial wake-up overhead. However, depending on the traffic, buffer utilization may vary

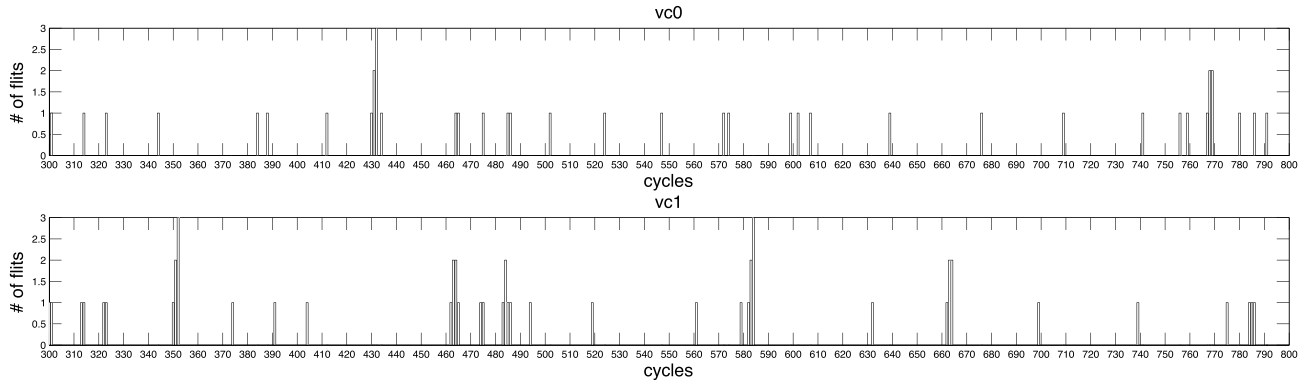


Fig. 5. Buffer utilization for different VCs (assuming two VCs per input port) of a router's input port on a 16-core system with a real workload (disparity) running.

over time and different VCs may have different occupancies. To test this hypothesis, using the same disparity example, we print out the buffer occupancies for two VCs in a physical port, as shown in Fig. 5. With each VC, its buffer occupancy varies over time. Furthermore, when comparing VC0 with VC1, their buffer utilizations are also different at most time. There are periods when VC0 is busy while VC1 is idle and vice versa.

The variation of buffer utilization indicates that only a subset of buffers is required at most time. A finer-granularity power management of buffers can potentially save power.

III. OUR METHOD: DimNoC

We propose DimNoC, a dim silicon scheme to tackle the aforementioned challenges in designing NoC in the dark silicon era. Because buffer is the primary NoC power consumer, especially in terms of leakage power, we focus on the buffer architecture optimization. We leverage both drowsy SRAM and STT-RAM techniques in NoC buffer design. The drowsy SRAM buffers introduce an intermediate power state between power ON and power-gating which can achieve a better performance power tradeoff. The STT-RAM buffers dissipate low-leakage power, endure frequent packet accesses, and own high-density cells. These advantages make them as promising candidates to replace the traditional SRAM-based buffer designs.

By integrating drowsy SRAM with STT-RAM, we can take advantage of both RAMs to significantly reduce wake-up penalty of gated buffers using drowsy circuit, to avoid unnecessary power-gating and wake-up utilizing low-leakage STT-RAM, to cut the area budget with the dense STT-RAM cells, and to losslessly preserve packets in the retention state to benefit both performance and power.

In the rest of this section, we first introduce the detailed design of drowsy SRAM-based buffers and STT-RAM-based buffers, then we describe two novel hybrid buffer architectures to accommodate these two technologies. At last, we further investigate how the data retention property of our DimNoC design can benefit network performance and power.

A. Drowsy SRAM Buffers

Instead of completely shutting down the on-chip components in response to power shortage, another option is to

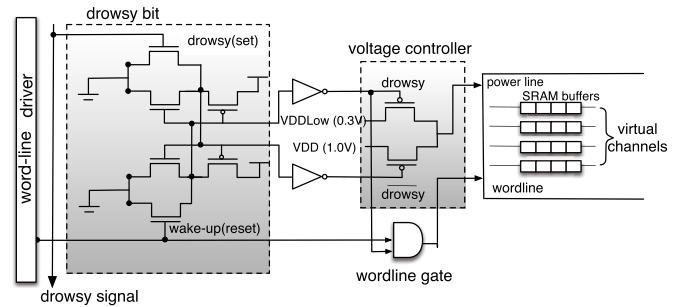


Fig. 6. Circuit design for drowsy buffer. The SRAM buffer is connected to the drowsy circuit which mainly contains a drowsy bit and a voltage controller. Depending on the state of the drowsy bit, the voltage controller switches the operating voltage between high and low states. In the drowsy bit, the word line, bit lines, and two pass transistors are not shown for simplicity.

operate them under-clocked, namely dim silicon. Regarding NoC design, this means that various nodes will operate at heterogeneous frequencies. A couple of prior work [34], [56] employs voltage and frequency scaling on individual network routers/links to implement heterogeneous frequencies. However, this technique requires per-node on-chip voltage regulators that may generate large hardware overhead. Moreover, the asynchronous communication between different frequency domains incurs extra latency overhead that may severely degrade network performance.

In order to achieve a practical DimNoC design, we do not adopt the conventional frequency tuning on individual routers, but rather explore the tradeoff between leakage saving and wake-up penalty. Power-gating aggressively cuts off the idle power but unavoidably generates high wake-up overhead. Therefore, it would be useful to introduce an intermediate sleep mode, in which a router operates at low-power state but does not need a significant effort to power it fully ON. Motivated by drowsy caches [17], we design similar low-power circuit for network buffers, since they dominate the total router power.

Fig. 6 shows our drowsy buffer design. The drowsy circuit includes a drowsy bit, a voltage controller, and a word-line gating circuit. Depending on the state of the drowsy bit, the voltage controller switches the operating voltage between high (active) or low (drowsy) states. In addition, the word-line gating circuit prevents access to the drowsy cells;

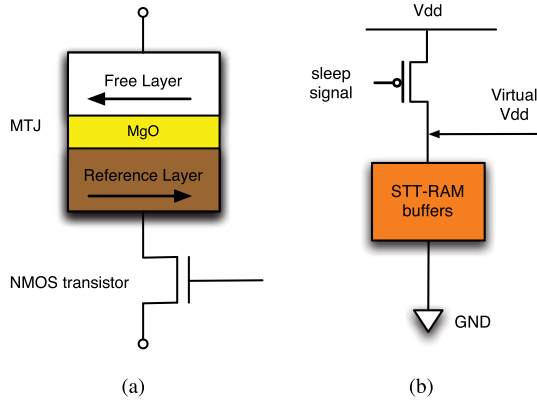


Fig. 7. (a) 1T1J STT-RAM cell structure, where the resistance of MTJ indicates the binary (1 or 0) information. (b) Sleep signal transistor is used to power-gate STT-RAM buffers to cut peripheral leakage.

otherwise, unexpected access to the drowsy cells will corrupt data. Upon buffer read, the drowsy bit is checked to determine the condition of the buffers. If the target buffers are in the normal mode, we can read the contents of the buffers without any performance loss. However, if the buffers are in the drowsy state, the buffers will be woken up automatically during the next cycle, and the data can be accessed during consecutive cycles.

Drowsy SRAM has faster transition delay than power-gated SRAM, though leakage power is not completely removed. As shown in Fig. 4, there are a lot of short intervals that are less than the ~ 10 cycle break-even time of power-gating. In contrast, the wake-up latency for drowsy SRAM is ~ 1 – 2 cycles [9], [17]. Unlike power-gating, as illustrated later in Section IV, input buffers do not need to wait until all flits are removed before entering the low-power drowsy mode.

B. STT-RAM Buffers

Apart from seeking new SRAM buffer architectures, we may embrace emerging NVM technologies to replace SRAM buffers. STT-RAM has the following advantages than SRAM or other NVM, which make it a good candidate to dim NoC buffers in the dark silicon era.

- 1) Like other resistive memories, STT-RAM relies on nonvolatile, resistive information storage in a cell, and thus exhibit near-zero leakage in the data array. Fig. 7(a) shows the structure of an STT-RAM cell. It uses a 1T1J structure which comprises of an access transistor and a magnetic tunnel junction (MTJ) for binary storage. An MTJ contains two ferromagnetic layers (reference layer and free layer) and one tunnel barrier layer (MgO). The direction of the reference layer is fixed, whereas the direction of the free layer can be changed by forcing the driving current. If the directions of these two layers are the same, the resistance of the MTJ is low, indicating a 0 state and vice versa for a 1 state.
- 2) STT-RAM uses smaller 1T1J cells as opposed to the typical six-transistor SRAM cells. An SRAM cell size is ~ 120 – $200 F^2$, whereas an STT-RAM cell size

is ~ 6 – $50 F^2$ [7]. As a result, for the same capacity as SRAM, the dense STT-RAM cells can cut the buffer area budget.

- 3) The endurance of STT-RAM (10^{15} writes [7], [23]) is significant higher than other NVM (e.g., 10^9 writes for PCM). This makes STT-RAM superior than other NVM, since there will be frequent packet accesses in the network.
- 4) Compared with volatile memories, such as SRAM, the nonvolatile STT-RAM can preserve data under power-gating state without losing information. Furthermore, because the data array of STT-RAM has negligible leakage, the peripheral circuit becomes the dominant leakage consumer in STT-RAM buffers. Simulation results from NVsim [14] show that approximate half of STT-RAM die area is occupied by peripheral circuit, i.e., half of the chip is leaky. Therefore, we can further cutoff the peripheral leakage power through power-gating. Fig. 7(b) shows the gating circuit, which uses a sleep transistor inserted between V_{dd} and the STT-RAM buffers to control the ON/OFF states.

STT-RAM read operations achieve comparable read latency and energy as SRAM. However, the write latency and energy of STT-RAM are relatively higher than an SRAM access. With the advance of technology, recent designs have demonstrated shorter write latency of 2–4 ns [12], [28], [29], [38], [59], which corresponds to 2–4 cycles in 1-GHz clock frequency.

Even with more conservative STT-RAM design, the write overhead can be mitigated by relaxing the nonvolatility of STT-RAM [7], [18], [26], [43], [46], [48]. In particular, STT-RAM generally has 10+ years of retention time, which is unnecessary for on-chip buffers. This is because NoC is mainly used for delivering packets rather than storing them. Even under high network congestion, the waiting time of packets is at most in the magnitude of microseconds. Therefore, by sacrificing the nonvolatility of STT-RAM from 10+ years to a few milliseconds or even microseconds, faster write speed and smaller write energy can be obtained. For example, Jog *et al.* [26] operate 10-ms retention time MTJ at a switching time of 2 ns with $61\text{-}\mu\text{A}$ write current.

The significant reduction of STT-RAM write latency provides enough opportunities to integrate STT-RAM as on-chip buffers. However, we do not aggressively assume the write latency gap between STT-RAM and SRAM can be completely eliminated. Moreover, in the hybrid buffer designs described in Section III-C, we propose novel techniques to further reduce STT-RAM write overhead and even seamlessly hide the long write latency.

C. Hybrid Buffer Architectures

As illustrated in the beginning of this section, hybrid buffers can potentially accommodate the advantages of both drowsy SRAM and nonvolatile STT-RAM for power-efficient NoC design in the dark silicon era. In this section, we propose two novel hybrid buffer architectures.

- 1) *Hierarchical Buffer*: To allow fine-grained activation/power-gating of different VCs, we divide the VCs into multiple levels and activate different levels of VCs successively

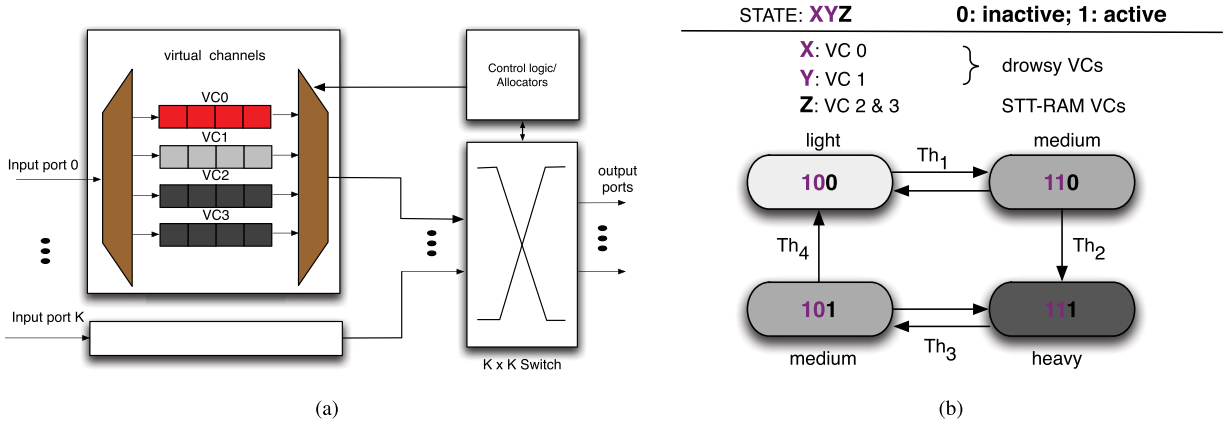


Fig. 8. HB architecture with four VCs as an example. VC0 and VC1 are drowsy VCs, and the rest are STT-RAM VCs. (a) VC0 is active and VC1 is in the drowsy state, whereas VC2 and VC3 are power-gated. (b) State transitions for different VCs in response to the traffic load. The thresholds are based on the buffer occupancy.

according to the network load. Naturally, we employ drowsy SRAM to design lower level VCs, which can be switched between the power-ON state and the drowsy state. Being aware of the short interarrival periods of consecutive packets, as shown in Fig. 4, we do not opt to power gate the drowsy buffers in order to reserve resources for instant wake-up. Correspondingly, higher-level VCs are made of STT-RAM and will be activated lastly to accommodate heavy traffic. In this way, the number of write accesses to STT-RAM will be significantly reduced. Moreover, power-gating techniques will be applied on STT-RAM VCs to take advantage of the long idle periods under very light traffic.

Fig. 8(a) shows an HB design, where VC0 and VC1 are drowsy VCs, while VC2 and VC3 are STT-RAM VCs. Note that we use four VCs here for illustration purposes, less or more numbers of VCs are also applicable, which subject to the cost constraint of employing more buffers. Therefore, we separate drowsy VCs and STT-RAM VCs into multiple levels, and allow fine-grained activation of different levels in a hierarchical manner. For example, in Fig. 8(a), VC0 (level 1) is active and VC1 (level 2) is in the drowsy state, whereas the rest STT-RAM VCs (level 3) are power-gated.

Fig. 8(b) shows the state-transition diagram of different VCs in response to variation of network traffic. X, Y, and Z are used to represent the status of VC0, VC1, and {2, 3}, respectively. 1 means active, whereas 0 stands for inactive. Note that inactive means drowsy for VC0 and VC1, and power-gated for VC2 and VC3. Initially, assuming the traffic load is light, only VC0 is activated and the rest VCs are inactive (100). When the incoming traffic exceeds a certain threshold (Th_1) of buffer occupancy, the remaining drowsy VC1 will be promptly activated (110). In the case of further increase of network load or abrupt arrival of burst packets that exceeds another threshold (Th_2), the STT-RAM VCs will be switched ON (111). Reversely, when the network traffic decreases (Th_3), we turn OFF part of drowsy VCs instead of the STT-RAM VCs (101), because the wake-up latency of STT-RAM VCs is much higher than that of drowsy SRAM VCs, and thus we do not power-gate STT-RAM VCs

immediately to avoid unnecessary wake-up in the case of network fluctuation. Finally, if the network traffic further decreases (Th_4), the STT-RAM VCs will be shut down (100). In addition, there will also be $110 \Rightarrow 100$ and $101 \Rightarrow 111$ transitions due to network fluctuation.

2) *Banked Buffer*: Apart from the VC-based partitioning which separates the drowsy SRAM VCs from STT-RAM VCs, we propose a more fine-grained hybrid buffer architecture, in which drowsy SRAM and STT-RAM buffers are logically interleaved within each VC, but physically organized as separate banks. This design allows simultaneous access to multiple banks so as to hide the longer write latency of STT-RAM.

Fig. 9(a) shows the logical view of the BB, where STT-RAM buffers and drowsy SRAM buffers are organized in an interleaved fashion within each VC. Note that we use four VCs to be consistent with Fig. 8(a) and the BB design is indeed applicable to any number of VCs or even without VC partitioning. Fig. 9(b) shows the physical architecture inside a single VC. In particular, each VC is separated into two banks. One is the STT-RAM bank, and the other is the drowsy SRAM bank. Then, the incoming flits will go through a bank-selection multiplexer and write into the appropriate bank. For example, assuming a two-cycle write latency for STT-RAM, during cycle 0, flit 1 will be written into the STT-RAM bank. Subsequently, at cycle 1, flit 2 will be directed into the drowsy SRAM bank and complete the write operation. Meanwhile, flit 1 will also complete its two-cycle write operation in the STT-RAM bank at this cycle, and thus both flits are readable at the next clock cycle. Similarly, the following flit 3 and flit 4 will be transferred into the STT-RAM bank and drowsy SRAM bank, respectively. In this way, the long write latency of STT-RAM can be hidden. Fig. 10 shows the timing diagram corresponding to Fig. 9(b). As we can see, all four flits can be read in sequential cycles.

Since the input VC is divided into two banks, the proposed BB design requires two write pointers and a read pointer, as shown in Fig. 9(b). The control logic will generate these pointers to guarantee proper write and read access. Note that, in this example, we assume a two-cycle STT-RAM write

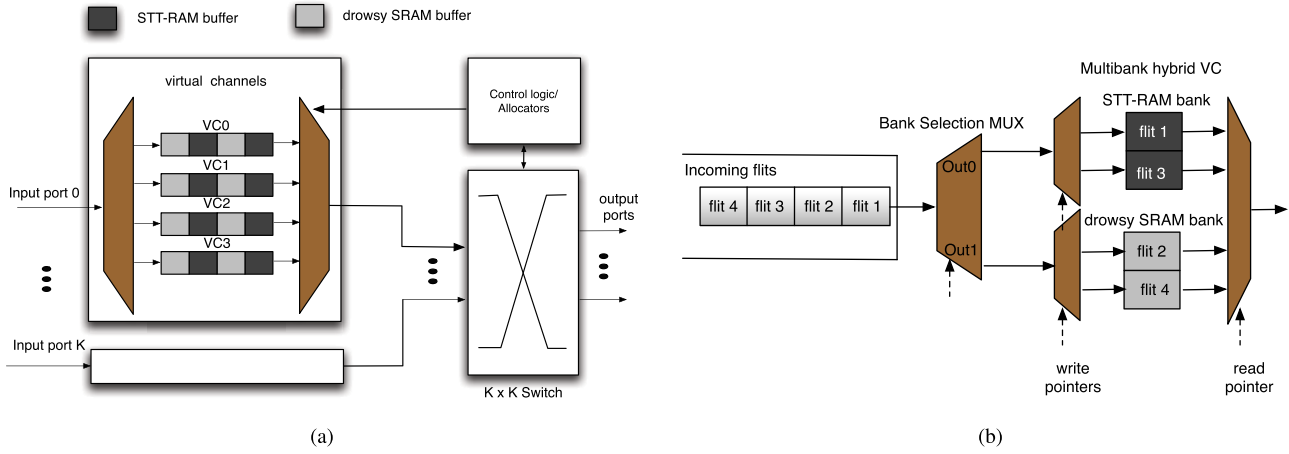


Fig. 9. BB architecture where each VC consists of both STT-RAMs and drowsy SRAMs in an interleaved fashion, so that consecutive flits will write into different RAMs. However, within each VC, the STT-RAM buffers and drowsy SRAMs are separated into two banks, so that these sequential buffer writes can be parallelized to hide the long write latency of STT-RAM. (a) Logical view of the BB architecture. (b) Physical design of a banked VC.

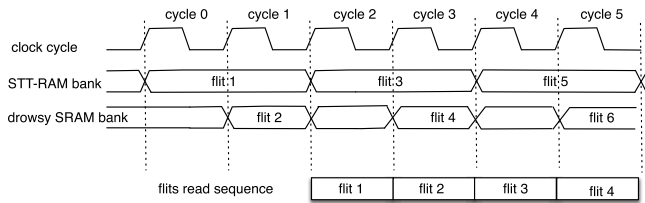


Fig. 10. Timing diagram for a dual-bank BB access. A write operation takes two cycles for STT-RAM bank and one cycle for drowsy SRAM bank. Four flits can be read in sequential cycles.

latency for clarification. In general, for a n -cycle STT-RAM write latency, we can divide the STT-RAM buffers into $n - 1$ banks. Together with the SRAM bank, the long write latency of STT-RAM can be hidden successfully. As validated in Section III-B, the value of n is between 2 and 4.

D. Data Retention in Network-on-Chip

1) *Feasibility of Data Retention:* Apart from the aforementioned advantages of adopting drowsy SRAM and STT-RAM in designing NoC buffers, another important property of these two technologies is data retention [39]. In particular, in the drowsy mode, the content in drowsy SRAM can be preserved in low-power mode; in power-gating mode, with the nonvolatility of STT-RAM, the data in STT-RAM buffers are also preserved with near-zero leakage. Therefore, when activated, packets in both drowsy SRAM and STT-RAM buffers can resume transmission immediately with intact data. To leverage this property, packets need to be stalled in the buffers for a long period, and no write/read access is allowed to the buffers. This is counter intuitive, because NoC buffers are used as temporary storage and should process packets as soon as possible to avoid router pipeline stalls. Nevertheless, in this paper, we explore the opportunities of preserving data in NoC without degrading network performance, which fully utilizes the property of data retention in drowsy SRAM and STT-RAM. In such a case, the input buffers do not need to be depleted before entering drowsy state or power-gating state. In our design,

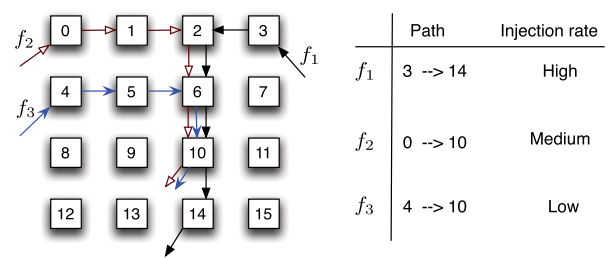


Fig. 11. Case for network congestion: three network flows are mapped to a 4×4 mesh network with different injection rates (indicated by high, medium, and low). Some of the paths are overlapped and cause network congestion due to limited network resources (e.g., buffers).

though STT-RAM relaxes retention time from 10+ years to a few milliseconds for faster access speed, this retention time is still more than sufficient for storing packets in the network.

2) *Implications on Flow Control:* Depending on the workload characteristics and mapping strategies, network traffic is usually nonuniformly distributed and even causes traffic hot spots. Source throttling [37], [51] mitigates the congestion, but its long feedback delay is inefficient for runtime traffic fluctuation. Alternatively, we can throttle the flit transmission from neighboring routers. For example, as shown in Fig. 11, assume there are three network flows: f_1 , f_2 , and f_3 . Among them, f_1 and f_2 have very high packet arrival rates, whereas f_3 has the lowest rate. Router 6 is highly possible to be heavily congested. Therefore, sending flits from a lower loaded router 5 to router 6 will worsen the congestion and involve in unpredictable delay. Instead, router 5 can stall the few flits in its own input buffers. In such a case, both these input buffers as well as the destined input port of router 6 could be put in low-power drowsy mode (SRAM) or even power-gating mode (STT-RAM) to save power. Finally, once router 6 becomes less congested, a 1-bit wake-up signal is sent to router 5. Then, the stalled flits can be retrieved quickly and resume transmission. In this way, while the flow control mechanism improves network performance, the data retention

TABLE I
OVERALL BUFFER MANAGEMENT POLICY WHICH LEVERAGES
THE PROPERTY OF DATA RETENTION

Buffer occupancy	Xbar requests to the same output port		
	light	medium	high
light	100	110	111
medium	110	110	100
high	111	111	000

TABLE II
SYSTEM AND INTERCONNECT CONFIGURATION

core count	16	topology	4×4 2D Mesh
L1 I & D cache	private, 32KB	router pipeline	four-stage
L2 cache	shared, 512KB/bank	VC count	4 VCs per port
cache line size	64B	buffer depth	4 buffers per VC
frequency	1GHz	packet length	5 flits, 16B/flit

property of drowsy SRAM and STT-RAM can losslessly preserve packet content and save static power significantly.

3) *Overall Buffer Management Policy*: The HB and BB hybrid buffer designs employ a hierarchical power-management scheme which activates VCs successively based on the overall buffer occupancy. Considering the data retention property, we introduce another metric, the number of crossbar requests to the same output port [20], to assist in VC activation/deactivation decisions. As a result, this adds another dimension to the state transitions in Fig. 8(b), as shown in Table I. For example, assuming input port k of a router has the lowest arrival rate among all the ports. When its buffer occupancy is low and the number of crossbar requests to its output port rises from low to high, all the VCs will be activated in serial ($100 \rightarrow 110 \rightarrow 111$). However, as its buffer occupancy turns high, port k will put VCs in the retention state ($111 \rightarrow 100 \rightarrow 000$) to avoid congestion and reduce power.

IV. EXPERIMENTS

We build a Pin-based functional simulator [40] to collect instruction traces from applications. The traces are then fed into a cycle-level trace-driven multicore simulator integrated with Garnet [2] and DSENT [47] to evaluate the network performance and power under 32-nm technology. Note that, here, we evaluate an 16-core system with 4×4 mesh network, but our mechanism can be applied to a variety of network sizes and topologies as well. The detailed system configurations are listed in Table II.

We use CACTI [1] and NVsim [14], combined statistics collected from recent STT-RAM prototypes [27], [30] to estimate the latency and energy consumption of each SRAM and STT-RAM access. Furthermore, we obtain the scaled latency and energy of accesses to drowsy SRAM [17] and STT-RAM with relaxed nonvolatility [48]. Table III shows the parameters of various designs.

We evaluate our hybrid buffer designs with the San Diego Vision Benchmark Suite [52]. We also construct synthetic traffic to study the benefits of leveraging data retention in network. Table IV summarizes and compares various buffer design techniques which are evaluated in our experiments. We compare our design with a prior work that employs

look-ahead routing-based scheme [31], and the latest NoC power-gating design called node-router decoupling (NoRD) [8].

A. Energy and Performance Evaluation

1) *Energy Savings*: Because our primary goal is to conserve network energy, we run different benchmarks with all buffer design techniques listed in Table IV. Fig. 12 shows the total network energy results. Note that we assume the wake-up latency of the conventional power-gating, look-ahead routing-based power-gating, and drowsy buffers are 10 cycles [8], 5 cycles [31], and 2 cycles [9], [17], respectively. In addition, the write latency of STT-RAM is two cycles. Possible longer write latencies will be evaluated in sensitivity studies later. There are four important observations to be highlighted.

- 1) Compared with the baseline All_SRAM design, All_SRAM_PG, which employs the conventional power-gating techniques to turn OFF an entire router whenever it is idle, indeed incurs 5.82% energy overhead due to frequent router wake-up, averaged over all the benchmarks.
- 2) All_SRAM_PG_LA [31] employs look-ahead routing to hide some wake-up latency and achieves 11.9% energy saving over the baseline. NoRD [8] provides bypass to transmit packets through gated routers and thus increases the power-gating duration. It reduces the network energy by 16.4%.
- 3) All_DrowsySRAM put routers into low-power drowsy state instead of completely shutting them down to further accelerate the wake-up process. On average, it cuts down the energy by 18.2%.
- 4) When replacing SRAM by STT-RAM, as shown in All_STTRAM, there are still energy savings compared with the baseline, indicating the benefit of low-leakage outweighs the overhead of high write energy of STT-RAM. However, the amount of energy saving (17.1%) is not very significant.
- 5) To avoid unnecessary write operations to STT-RAM, HB uses hybrid buffers and only activates the STT-RAM VCs at high network load. As a result, it achieves 30.9% energy savings on average, which is mainly due to the reduction of leakage energy (we achieved 43.9% leakage energy reduction on average). Alternatively, BB accesses drowsy SRAM bank and STT-RAM bank in an interleaved fashion within each VC, which successfully hides the long write latency of STT-RAM and achieves 26.3% energy saving on average.

2) *Performance Overhead*: The proposed buffer designs achieve energy savings by deactivating some of the network resources, which would unavoidably sacrifice the network performance to some degree. To evaluate the performance overhead incurred by these buffer designs, Fig. 13 shows the average packet latencies when running different benchmarks, including queuing latency in the network interface (NI) and network latency from source nodes to destination nodes.

Unsurprisingly, All_SRAM_PG leads to significant network delay, adding 44.9% on average across all the benchmarks.

TABLE III
1-KB SRAM AND STT-RAM BUFFER CONFIGURATIONS

	cell size (F^2)	write latency (cycle)	read latency (cycle)	write energy/bit (pJ)	read energy/bit (pJ)	leakage power (mW)
SRAM	146	1	1	0.049	0.063	1.797
DrowsySRAM	146	1	1	0.049	0.063	0.165
STT-RAM (10yr)	50.67	10	1	0.534	0.153	0.042
STT-RAM (10ms)	45.60	2	1	0.286	0.082	0.044

TABLE IV
COMPARISONS OF DIFFERENT BUFFER DESIGN TECHNIQUES

All_SRAM	Baseline, which employs pure conventional SRAM based buffers
All_SRAM_PG	Buffers are designed with pure SRAM, and conventional power-gating technique is applied
All_SRAM_PG_LA	Buffers are designed with pure SRAM, and look-ahead routing based power-gating technique is applied [31]
NoRD	Buffers are designed with pure SRAM, and bypass paths are introduced to deliver packets without waking up gated routers [8]
All_DrowsySRAM	Buffers are designed with pure <i>drowsy</i> SRAM
All_STTRAM	Buffers are designed with pure STT-RAM
HB	Hierarchical Buffer: Lower-level VCs are pure SRAM based, and higher-level VCs are pure STT-RAM based
BB	Banked Buffer: Drowsy SRAM and STT-RAMs are constructed into separate banks, and are accessed in an interleaved fashion in each VC

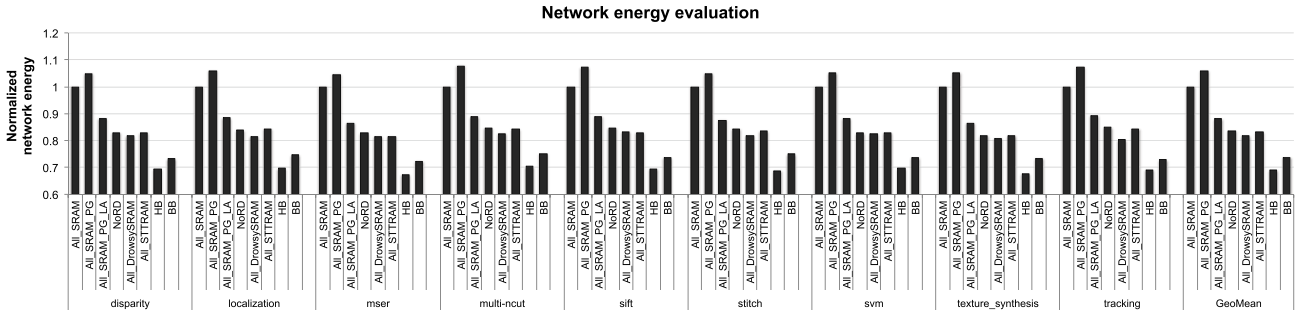


Fig. 12. Network energy comparisons for different buffer designs and power-management schemes.

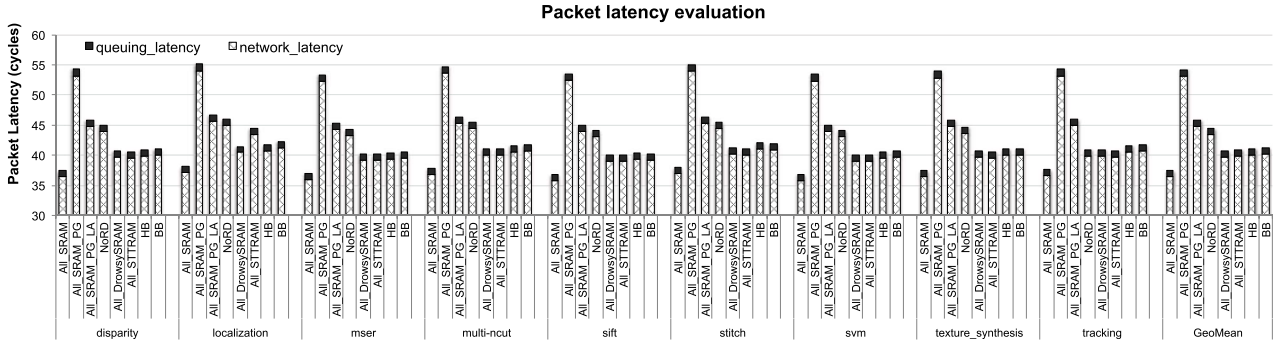


Fig. 13. Packet latency comparisons for different buffer designs and power-management schemes.

By mitigating the wake-up delay, All_SRAM_PG_LA and All_DrowsySRAM amortize the latency overhead to 22.3% and 8.8% on average, respectively. NoRD reduces the number of wake-up operations through bypass but still suffers from 18.6% performance overhead due to packet detours.

For All_STTRAM, the average network latency overhead is only 9.4%. HB and BB slightly increase the overhead to 10.4% and 10.8%, respectively. This is because they further apply fine-grained power-gating on individual VCs.

To better understand the previous latency results, Fig. 14 shows the corresponding link utilizations. For all

the benchmarks, All_SRAM_PG has the least link utilization. This is because the long wake-up delay on the sleep router will block packets and thus result in under-utilized links. In contrast, All_DrowsySRAM and other STT-RAM-based designs have the closest utilization to the baseline. A general trend we can observe from Fig. 14 is that, the less link utilization, the higher performance overhead.

B. Sensitivity Study on STT-RAM Writes

The aforementioned buffer designs which contain STT-RAM assumes a two-cycle write latency. The reduction

The bar chart displays link utilization values for ten different tasks: disparity, localization, msr, multi-ncut, sift, stitch, svm, texture_synthesis, tracking, and GeoMean. Each task has eight bars representing different methods: All SRAM, All SRAM PG, All SRAM PG LA, All SRAM PG LA NoD, All DrowsSRAM, All DrowsSRAM NoD, All STTRAM, and BB. The y-axis ranges from 0.026 to 0.038.

Task	All SRAM	All SRAM PG	All SRAM PG LA	All SRAM PG LA NoD	All DrowsSRAM	All DrowsSRAM NoD	All STTRAM	BB
disparity	0.0345	0.0300	0.0325	0.0330	0.0340	0.0345	0.0335	0.0330
localization	0.0355	0.0305	0.0330	0.0335	0.0345	0.0335	0.0340	0.0335
msr	0.0335	0.0290	0.0315	0.0325	0.0330	0.0325	0.0330	0.0325
multi-ncut	0.0355	0.0300	0.0330	0.0335	0.0345	0.0335	0.0340	0.0335
sift	0.0340	0.0290	0.0315	0.0325	0.0330	0.0335	0.0330	0.0325
stitch	0.0350	0.0300	0.0325	0.0330	0.0340	0.0345	0.0335	0.0335
svm	0.0345	0.0300	0.0320	0.0325	0.0335	0.0340	0.0335	0.0335
texture_synthesis	0.0340	0.0290	0.0315	0.0325	0.0330	0.0335	0.0330	0.0325
tracking	0.0355	0.0310	0.0330	0.0335	0.0345	0.0340	0.0345	0.0340
GeoMean	0.0345	0.0300	0.0325	0.0330	0.0340	0.0345	0.0335	0.0335

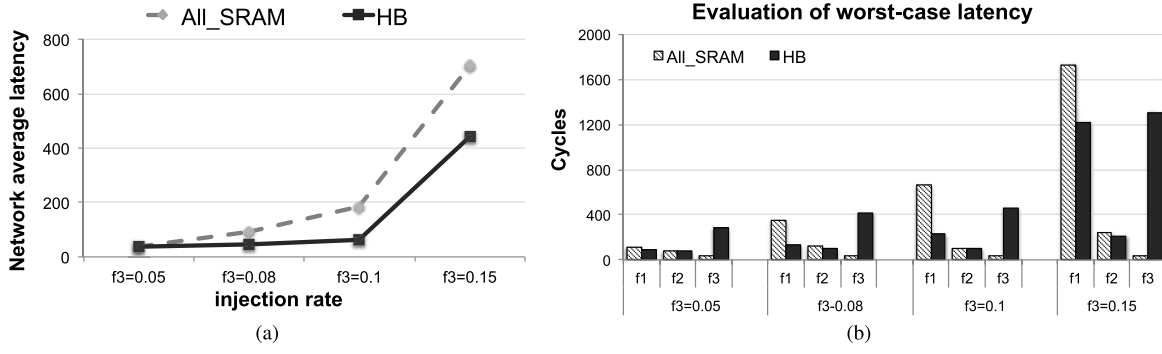


Fig. 16. Latency comparisons when the injection rate of f_3 varies. (a) Latency comparisons. (b) Worse-case latencies.

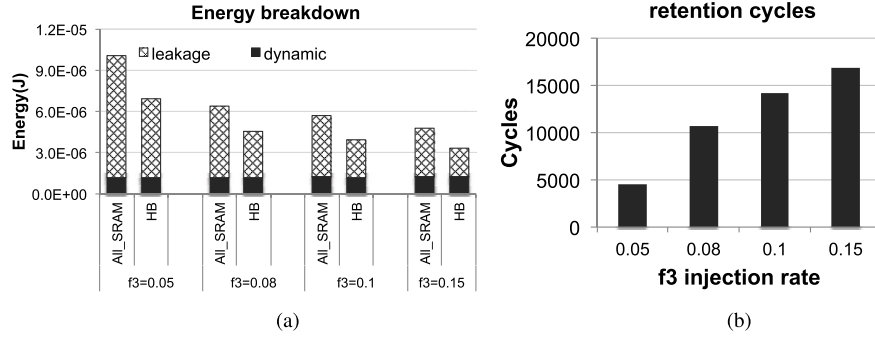


Fig. 17. Energy comparisons when the injection rate of f_3 varies. (a) Energy comparisons. (b) Number of cycles in the retention state.

average network latency boosts sharply, indicating the network is close to being saturated.

Although the retention-based flow control is helpful to reduce the average network latency, it may degrade the worst case latency due to starvation. For example, f_3 packets from the west input port of router 6 may not be granted for crossbar traversal for a very long period. To overcome this problem, we use a simple scheme which triggers the inactive buffers after a certain time threshold. As shown in Fig. 16(b), at lower injection rates, the slowest packet latency is larger than that of the All_SRAM baseline. However, as congestion becomes more intense at high injection rates (0.1 and 0.15), our mechanism reduces the worst case packet latency considerably. On the other hand, there are other studies to recover STT-RAM from the retention state to avoid data corruption, although out of the scope of this paper. Nevertheless, the retention time of STT-RAM in our design is 10 ms, or 10 000 000 cycles, which is sufficient to sustain packets in the network.

The earlier flow control mechanism is mainly driven by the data retention property of drowsy SRAM and nonvolatile STT-RAM, which can significantly reduce network power while ensuring data integrity. Fig. 17(a) shows the overall network energy consumption with different f_3 injection rates. For fair comparison, we assume the unused routers and ports in Fig. 11 can be completely shut down in both All_SRAM and our HB design. We can see leakage power is significantly reduced in all the cases, leading to an average network energy saving of 30.3%. Fig. 17(b) further plots the total number of cycles when the network has buffers in the retention state. Its trend can be explained in a similar mechanism, as shown

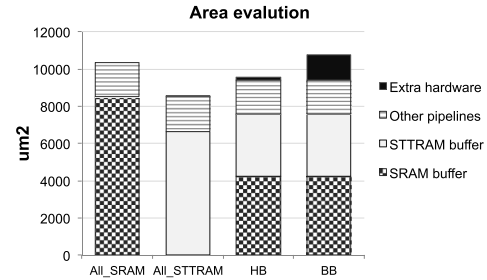


Fig. 18. Router area breakdown for different buffer designs.

in Fig. 16(a). Before the network saturates, buffers can be more frequently put in the retention state when the injection rate increases.

D. Hardware Implementation

The HB and BB designs require modifications to the conventional pure SRAM-based router architecture. In particular, for HB, as shown in Fig. 8(a), drowsy circuit is introduced to low-level VCs, and the higher-level VCs are replaced by STT-RAM. Sleep transistors are also needed for power-gating purposes. Moreover, for BB, as shown in Fig. 9, additional multiplexers are required to enable access to different buffer banks. To evaluate the area and power overhead of our proposed hybrid buffer designs, we synthesize a parametrized RTL router implementation [4] using Synopsis Design Compiler, and substitute the power/area numbers of SRAM and STT-RAM buffers through CACTI [1] and NVsim [14] simulation. For fair comparison, we keep the buffer capacity consistent throughout all buffer designs.

For SRAM, a typical cell size is $\sim 146 F^2$ [1] in the 32-nm technology. When relaxing the nonvolatility of STT-RAM for faster write speed, the cell size of STT-RAM is also affected. Fig. 18 shows the area breakdown for different router architectures. The All_STTRAM router design significantly reduces the area because of the high cell density of STT-RAM. It achieves 17.8% area saving compared with the baseline All_SRAM. The HB design also achieves 7.6% area saving. More complex logic units are required for the BB, but the overall 4.1% overhead is negligible.

From an energy' perspective, HB and BB will increase the dynamic buffer access energy due to the integration of STT-RAM. Correspondingly, leakage power will be significantly reduced compared with pure SRAM routers. We have incorporated the hardware overhead in the energy evaluation (Fig. 12).

E. Discussion

1) *VC-Based Buffer Partitioning*: VCs are typically used in NoC to both avoid deadlock and improve performance. Many recent research assume 4+ VCs [8], [13], [44] for the same size NoC as ours. For example, Balfour and Dally [3] reveal that the most efficient NoC configurations require 4–8 VCs for a single traffic class. Nevertheless, our HB design is applicable to one drowsy SRAM VC and one STT-RAM VC, and the BB design works for any number of VCs or even without VC partitioning.

Given multiple VCs already exist in most NoC designs for performance purposes, this paper strives to manage the power states of those VCs to achieve optimal power efficiency. In addition, it is interesting to explore the optimal number of buffers to be allocated to drowsy SRAM and STT-RAM, and incorporate a unified buffer architecture [36] to dynamically change the number of VCs based on traffic. This is outside the scope of this paper and we leave it as future work.

2) *NoC Power Translated Into Chip Power Saving*: It has been widely reported by various studies [21], [35] that the power consumption of NoC is dominated by buffers, and also verified by our experimental results, as shown in Fig. 2. In turn, power consumed by NoC is projected to constitute a significant portion (10%–36% as in some research and industrial many-core prototypes [5], [22], [50]) of total chip power.

In this paper, we evaluate a simple 16-node mesh network for illustration purposes. In real many-core designs, more complex and power-consuming network will be employed to support different traffic classes. For example, in Tiler's recent TILE64 or TILE-Gx72 processor chips, five-independent networks are integrated. Our method can be applied in such scenarios which will generate significant chip power savings.

Moreover, in the dark silicon era, the majority of on-chip transistors may be put in the dormant state due to TDP limit. However, without further optimization, the NoC components must remain active; otherwise, a gated-off router will block packet-forwarding and the access to the local but shared caches/directories. As a result, the ratio of NoC power over chip power rises substantially and may even lead to higher NoC power than that of the remaining active resources [57].

V. RELATED WORK

A. NoC Power-Gating

Look-ahead routing [31] hides some wake-up delay but still encounters frequent router wake-up. Router parking [44] leverages adaptive routing to avoid unnecessary router wake-up, but it does not work for a shared last-level cache. NoRD [8] introduces bypass paths to forward packets or access the local shared caches even when the attached router is power-gated, but still suffers from packet detour and scalability issues. Catnap [13] employs course-granularity power-gating techniques that turn ON and OFF the entire subnetwork, which is inflexible for skewed or highly asymmetric traffic patterns. NoC sprinting [57] only activates a subset of routers based on workload characteristics, but it relies on the ON/OFF status of the underlying cores and requires special phase-change materials to support the computational sprinting [42] process. Recently, Mirhosseini *et al.* [33] proposed VC power-gating techniques with traditional buffers. In contrast, our approach not only performs flexible fine-granularity power management at buffer level but also incorporates drowsy SRAM and STT-RAM to achieve a better performance power tradeoff.

B. NoC Buffer Power Optimization

Because buffer is the dominate power consumer of NoC, recent research efforts have been devoted to reduce buffer power. Among them, bufferless NoC [35] and elastic-buffered NoC [32] physically carve away nearly all buffers from NoC to save leakage power (and area), but suffer from performance penalties under high load. In addition, intrinsically, they are not able to support VCs, and therefore, their feasibility is unclear in real systems, where multiple traffic classes are often needed to avoid protocol-level deadlock. In contrast, our design still retains buffers and VCs but adaptively change their power states based on activities, so as to reduce leakage power when idling but delivers high performance when sprinting [42].

C. STT-RAM as NoC Buffer

STT-RAM-based caches or hybrid caches [10], [24], [54] have been well explored in the on-chip cache domain. Recently, Jang *et al.* [25] first propose to leverage STT-RAM in NoC buffers, but mainly for performance purposes by designing larger buffers with dense STT-RAM cells. Their design suffers from power overhead in moderate or high network load. In contrast, we propose two novel hybrid buffer designs based on drowsy SRAM and STT-RAM, which successfully cut down the network power. Among them, the BB design seamlessly hides the long write delay of STT-RAM. Moreover, DimNoC is the first work to leverage nonvolatility of STT-RAM in NoC for saving power.

D. Dark-Silicon-Aware NoC Design

Most of the previous dark silicon research focuses on core or memory subsystem optimizations—dark-silicon-aware on-chip interconnects design has not received sufficient attention. Recently, NoC-sprinting [57] provides an adaptive NoC design to support fine-grained computational sprinting for different workloads; DarkNoC [6] proposes a multinet

design with each network optimized for different voltage–frequency ranges for the optimal energy efficiency. This paper is an extension of our prior work [55].

VI. CONCLUSION

In this paper, we propose DimNoC to tackle the dark silicon crisis from the NoC's perspective, which integrates drowsy SRAM and STT-RAM to design buffers in a router. Moreover, two hybrid buffer architectures, HB and BB, are designed with efficient power-management strategies. Experimental results on the San Diego Vision Benchmark Suite show that DimNoC can achieve 30.9% energy savings on average, 20.3% network EDP reduction, and 7.6% router area reduction because of the high-density of STT-RAM cells.

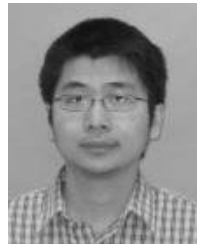
REFERENCES

- [1] S. J. E. Wilton and N. P. Jouppi, "CACTI: An enhanced cache access and cycle time model," *IEEE J. Solid-State Circuits*, vol. 31, no. 5, pp. 677–688, 1996.
- [2] N. Agarwal, T. Krishna, L.-S. Peh, and N. K. Jha, "GARNET: A detailed on-chip network model inside a full-system simulator," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw. (ISPASS)*, Apr. 2009, pp. 33–42.
- [3] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP on-chip networks," in *Proc. 20th Annu. Int. Conf. Supercomput.*, 2006, pp. 187–198.
- [4] D. U. Becker, "Efficient microarchitecture for network-on-chip routers," Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, 2012.
- [5] S. Bell *et al.*, "TILE64 processor: A 64-core SoC with mesh interconnect," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2008, pp. 88–598.
- [6] H. Bokhari, H. Javaid, M. Shafique, J. Henkel, and S. Parameswaran, "darkNoC: Designing energy-efficient network-on-chip with multi-Vt cells for dark silicon," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.
- [7] M.-T. Chang, P. Rosenfeld, S.-L. Lu, and B. Jacob, "Technology comparison for large last-level caches (L³CS): Low-leakage SRAM, low write-energy STT-RAM, and refresh-optimized eDRAM," in *Proc. IEEE 19th Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2013, pp. 143–154.
- [8] L. Chen and T. M. Pinkston, "NoRD: Node-router decoupling for effective power-gating of on-chip routers," in *Proc. 45th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2012, pp. 270–281.
- [9] X. Chen and L.-S. Peh, "Leakage power modeling and optimization in interconnection networks," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, 2003, pp. 90–95.
- [10] Y.-T. Chen *et al.*, "Dynamically reconfigurable hybrid cache: An energy-efficient last-level cache design," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Mar. 2012, pp. 45–50.
- [11] H.-Y. Cheng, J. Zhan, J. Zhao, Y. Xie, J. Sampson, and M. J. Irwin, "Core vs. uncore: The heart of darkness," in *Proc. 52nd Annu. Design Autom. Conf. (DAC)*, 2015, Art. no. 121.
- [12] K. C. Chun, H. Zhao, J. D. Harms, T.-H. Kim, J.-P. Wang, and C. H. Kim, "A scaling roadmap and performance evaluation of in-plane and perpendicular MTJ based STT-MRAMs for high-density cache memory," *IEEE J. Solid-State Circuits*, vol. 48, no. 2, pp. 598–610, Feb. 2013.
- [13] R. Das, S. Narayanasamy, S. K. Satpathy, and R. G. Dreslinski, "Catnap: Energy proportional multiple network-on-chip," in *Proc. 40th Annu. Int. Symp. Comput. Archit. (ISCA)*, 2013, pp. 320–331.
- [14] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 7, pp. 994–1007, Jul. 2012.
- [15] H. Esmailzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proc. 38th Annu. Int. Symp. Comput. Archit. (ISCA)*, 2011, pp. 365–376.
- [16] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki, "Toward dark silicon in servers," *IEEE Micro*, vol. 31, no. 4, pp. 6–15, Jul./Aug. 2011.
- [17] K. Flautner, N. S. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: Simple techniques for reducing leakage power," in *Proc. 29th Annu. Int. Symp. Comput. Archit. (ISCA)*, 2002, pp. 148–157.
- [18] N. Goswami, B. Cao, and T. Li, "Power-performance co-optimization of throughput core architecture using resistive memory," in *Proc. IEEE 19th Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2013, pp. 342–353.
- [19] N. Goulding *et al.*, "GreenDroid: A mobile application processor for a future of dark silicon," in *Proc. Hot Chips*, 2010, pp. 1–39.
- [20] P. Gratz, B. Grot, and S. W. Keckler, "Regional congestion awareness for load balance in networks-on-chip," in *Proc. IEEE 14th Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2008, pp. 203–214.
- [21] P. Gratz, C. Kim, R. McDonald, S. W. Keckler, and D. Burger, "Implementation and evaluation of on-chip network architectures," in *Proc. IEEE Int. Conf. Comput. Design (ICCD)*, Oct. 2006, pp. 477–484.
- [22] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-GHz mesh interconnect for a Teraflops processor," *IEEE Micro*, vol. 27, no. 5, pp. 51–61, Sep./Oct. 2007.
- [23] ITRS. (2013). *Process Integration, Devices, and Structures (PIDS)*. [Online]. Available: http://www.itrs.net/Links/2013ITRS/2013Tables/PIDS_2013Tables.xlsx.
- [24] A. Jadidi, M. Arjomand, and H. Sarbazi-Azad, "High-endurance and performance-efficient design of hybrid cache architectures through adaptive line replacement," in *Proc. 17th IEEE/ACM Int. Symp. Low-Power Electron. Design (ISLPED)*, Aug. 2011, pp. 79–84.
- [25] H. Jang, B. S. An, N. Kulkarni, K. H. Yum, and E. J. Kim, "A hybrid buffer design with STT-MRAM for on-chip interconnects," in *Proc. 6th IEEE/ACM Int. Symp. Netw. Chip (NoCS)*, May 2012, pp. 193–200.
- [26] A. Jog *et al.*, "Cache revive: Architecting volatile STT-RAM caches for enhanced performance in CMPs," in *Proc. 49th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2012, pp. 243–252.
- [27] J. P. Kim *et al.*, "A 45 nm 1 Mb embedded STT-MRAM with design techniques to minimize read-disturbance," in *Proc. Symp. VLSI Circuits (VLSIC)*, Jun. 2011, pp. 296–297.
- [28] T. Kishi *et al.*, "Lower-current and fast switching of a perpendicular TMR for high speed and high density spin-transfer-torque MRAM," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, Dec. 2008, pp. 1–4.
- [29] E. Kitagawa *et al.*, "Impact of ultra low power and fast write operation of advanced perpendicular MTJ on power reduction for high-performance mobile CPU," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, Dec. 2012, pp. 29.4.1–29.4.4.
- [30] C. J. Lin *et al.*, "45 nm low power CMOS logic compatible embedded STT MRAM utilizing a reverse-connection 1T/1MTJ cell," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, Dec. 2009, pp. 1–4.
- [31] H. Matsutani, M. Koibuchi, D. Wang, and H. Amano, "Run-time power gating of on-chip routers using look-ahead routing," in *Proc. Asia South Pacific Design Autom. Conf. (ASPDAC)*, 2008, pp. 55–60.
- [32] G. Micheliogiannakis and W. J. Dally, "Elastic buffer flow control for on-chip networks," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 295–309, Feb. 2013.
- [33] A. Mirhosseini, M. Sadrosadati, A. Fakhrazadehgan, M. Modarressi, and H. Sarbazi-Azad, "An energy-efficient virtual channel power-gating mechanism for on-chip networks," in *Proc. Design, Autom. Test Eur. Conf. Exhibit.*, 2015, pp. 1527–1532.
- [34] A. K. Mishra, R. Das, S. Eachempati, R. Iyer, N. Vijaykrishnan, and C. R. Das, "A case for dynamic frequency tuning in on-chip networks," in *Proc. 42nd Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2009, pp. 292–303.
- [35] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," in *Proc. 36th Annu. Int. Symp. Comput. Archit.*, 2009, pp. 196–207.
- [36] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, "ViChar: A dynamic virtual channel regulator for network-on-chip routers," in *Proc. 39th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2006, pp. 333–346.
- [37] G. P. Nychis, C. Fallin, T. Moscibroda, O. Mutlu, and S. Seshan, "On-chip networks from a networking perspective: Congestion and scalability in many-core interconnects," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 407–418, 2012.
- [38] T. Ohsawa *et al.*, "A 1.5 nsec/2.1 nsec random read/write cycle 1 Mb STT-RAM using 6T2MTJ cell with background write for non-volatile e-memories," in *Proc. Symp. VLSI Technol. (VLSIT)*, 2013, pp. C110–C111.

- [39] X. Pan and R. Teodorescu, "NVSleep: Using non-volatile memory to enable fast sleep/wakeup of idle cores," in *Proc. 32nd IEEE Int. Conf. Comput. Design (ICCD)*, Oct. 2014, pp. 400–407.
- [40] H. Patil, R. Cohn, M. Charney, R. Kapoor, A. Sun, and A. Karunanidhi, "Pinpointing representative portions of large Intel Itanium programs with dynamic instrumentation," in *Proc. 37th Int. Symp. Microarchitecture (MICRO)*, 2004, pp. 81–92.
- [41] M. Powell, S.-H. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar, "Gated-V_{dd}: A circuit technique to reduce leakage in deep-submicron cache memories," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, 2000, pp. 90–95.
- [42] A. Raghavan *et al.*, "Computational sprinting," in *Proc. IEEE 18th Int. Symp. High-Perform. Comput. Archit. (HPCA)*, Feb. 2012, pp. 1–12.
- [43] M. H. Samavatian, H. Abbasitabar, M. Arjomand, and H. Sarbazi-Azad, "An efficient STT-RAM last level cache architecture for GPUs," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.
- [44] A. Samih, R. Wang, A. Krishna, C. Maciocco, C. Tai, and Y. Solihin, "Energy-efficient interconnect via router parking," in *Proc. IEEE 19th Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2013, pp. 508–519.
- [45] M. Shafique, S. Garg, J. Henkel, and D. Marculescu, "The EDA challenges in the dark silicon era," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.
- [46] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan, "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," in *Proc. IEEE 17th Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2011, pp. 50–61.
- [47] C. Sun *et al.*, "DSENT—A tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling," in *Proc. 6th IEEE/ACM Int. Symp. Netw. Chip (NoCS)*, May 2012, pp. 201–210.
- [48] Z. Sun *et al.*, "Multi retention level STT-RAM cache designs with a dynamic refresh scheme," in *Proc. 44th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2011, pp. 329–338.
- [49] K. Swaminathan, E. Kultursay, V. Saripalli, V. Narayanan, M. T. Kandemir, and S. Datta, "Steep-slope devices: From dark to dim silicon," *IEEE Micro*, vol. 33, no. 5, pp. 50–59, Sep/Oct. 2013.
- [50] M. B. Taylor *et al.*, "The raw microprocessor: A computational fabric for software circuits and general-purpose programs," *IEEE Micro*, vol. 22, no. 2, pp. 25–35, Mar./Apr. 2002.
- [51] M. Thottethodi, A. R. Lebeck, and S. S. Mukherjee, "Self-tuned congestion control for multiprocessor networks," in *Proc. 7th Int. Symp. High-Perform. Comput. Archit. (HPCA)*, 2001, pp. 107–118.
- [52] S. K. Venkata *et al.*, "SD-VBS: The San Diego vision benchmark suite," in *Proc. IEEE Int. Symp. Workload Characterization (IISWC)*, Oct. 2009, pp. 55–64.
- [53] G. Venkatesh *et al.*, "Conservation cores: Reducing the energy of mature computations," in *Proc. 15th Ed. Archit. Support Program. Lang. Oper. Syst.*, 2010, pp. 205–218.
- [54] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Hybrid cache architecture with disparate memory technologies," *ACM SIGARCH Comput. Archit. News*, vol. 37, no. 3, pp. 34–45, 2009.
- [55] J. Zhan, J. Ouyang, F. Ge, J. Zhao, and Y. Xie, "DimNoC: A dim silicon approach towards power-efficient on-chip network," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–6.
- [56] J. Zhan, N. Stoimenov, J. Ouyang, L. Thiele, V. Narayanan, and Y. Xie, "Designing energy-efficient NoC for real-time embedded systems through slack optimization," in *Proc. 50th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, May/Jun. 2013, pp. 1–6.
- [57] J. Zhan, Y. Xie, and G. Sun, "NoC-sprinting: Interconnect for fine-grained sprinting in the dark silicon era," in *Proc. 51st ACM/EDAC/IEEE Annu. Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.
- [58] Y. Zhang, L. Peng, X. Fu, and Y. Hu, "Lighting the dark silicon by exploiting heterogeneity on future processors," in *Proc. 50th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, May/Jun. 2013, pp. 1–7.
- [59] H. Zhao *et al.*, "Low writing energy and sub nanosecond spin torque transfer switching of in-plane magnetic tunnel junction for spin torque transfer random access memory," *J. Appl. Phys.*, vol. 109, no. 7, p. 07C720, 2011.



computer architecture, with an emphasis on network-on-chip for many-core processors.



Jia Zhan (S'12) received the B.S. degree from the Harbin Institute of Technology, Harbin, China, in 2011. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA, USA.

He was with the Department of Computer Science and Engineering, Pennsylvania State University, State College, PA, USA. His current research interests include a broad range of

Jin Ouyang (M'09) received the B.S. degree in microelectronics from Peking University, Beijing, China, in 2007, and the Ph.D. degree in computer engineering from Pennsylvania State University, State College, PA, USA, in 2012.

He is currently a Senior Engineer with NVIDIA Corporation, Santa Clara, CA, USA. His current research interests include computer architectures and VLSI systems and circuits.



Fen Ge (M'14) was a Visiting Scholar with the Department of Computer Science and Engineering with Pennsylvania State University, State College, PA, USA, from 2013 to 2014. She is currently an Associate Professor with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China. Her current research interests include network-on-chip and embedded systems design.



Jishen Zhao (M'10) received the Ph.D. degree from Pennsylvania State University, State College, PA, USA.

She is currently an Assistant Professor with the University of California at Santa Cruz, Santa Cruz, CA, USA. Her current research interests include computer architecture and electronic design automation, with an emphasis on emerging technologies and high-performance computing.



Yuan Xie (F'15) was with IBM, Armonk, NY, USA, from 2002 to 2003, and AMD Research China Lab, Beijing, China, from 2012 to 2013. He has been a Professor with Pennsylvania State University, State College, PA, USA, since 2003. He is currently a Professor with the Electrical and Computer Engineering Department, University of California at Santa Barbara, Santa Barbara, CA, USA. His current research interests include computer architecture, Electronic Design Automation, and VLSI design.