IAA: Incidental Approximate Architectures for Extremely Energy-Constrained Energy Harvesting Scenarios using IoT Nonvolatile Processors

Kaisheng Ma, Jinyang Li, Xueqing Li, and Yongpan Liu
Tsinghua University

Yuan Xie
University of California, Santa Barbara

Mahmut Kandemir, Jack Sampson, and Vijaykrishnan Narayanan
Pennsylvania State University

Battery-less IoT devices powered through energy harvesting face a fundamental imbalance between the potential volume of collected data and the amount of energy available for processing that data locally. We explore a combination of approximate computing and intermittent computing-incidental approximate architecture to suit nonvolatile processors (NVPs).

The shift from battery-powered systems to self-powered systems promises to fuel the next revolution in IoT. The ability to power IoT devices using ambient, scavenged energy liberates them from the lifetime, deployment, and servicing limitations of a fixed battery. While ambient energy sources are notoriously fickle, concurrent advances in energy harvesting, ultra-low-power computation, and nonvolatile memory have enabled a new generation of processors, known as nonvolatile processors (NVPs), which tightly integrate nonvolatile memory elements into the logic fabric of the processor, thereby enabling almost instantaneous stopping and starting of execution through parallel distributed backup and restore functionality for processor state. For NVPs with microarchitectural hardware-managed backup,[1] systems can make persistent progress on a compute task even if only one instruction successfully completes between power interruptions.

But energy is expensive and unstable. Simply resuming tasks might not be the best option, for such a phenomenon is observed: If an NVP has been without power for some substantial time, resuming work on the input it was processing when power failed might have lower utility, from an application perspective, than moving on to processing the newest input.

Published by the IEEE Computer Society
0272-1732/18/$33.00 ©2018 IEEE

In this article, we introduce an incidental approximate architecture (IAA) to address opportunistic responsiveness versus quality tradeoffs for older inputs under unstable power income. The article makes the following contributions:

- We introduce incidental computing, wherein older computation is carried out in a best-effort fashion during the execution of newer computations.
- In this extended article, the evaluation of a chain of approximate and non-approximatable kernels are re-evaluated as a holistic IoT workflow.
- In this extended article, we show that the energy-harvesting systems using NVPs with IAA policies can perform very comparably, almost the same as battery-powered solutions with volatile processors when the average power is the same, although with some acceptable quality loss.

INCIDENTAL COMPUTING

A Key Observation for IoT Applications

In many deployment scenarios, catching up quickly after a power failure might take priority over the quality of response. Furthermore, such applications often contain kernels with independent loop iterations that could conceivably be skipped over in their entirety. However, skipping represents, in a sense, a maximum quality reduction, especially if each iteration performs the same essential computation on different data, which is the common case in image/signal processing kernels. Finally, while average power—even during periods of sufficient power to allow for uninterrupted execution—is low in harvested systems, peak power can be substantially higher than average.

Incidental Computing: Roll-Forward Instead of Roll-Back

To take advantage of these observations, we propose incidental approximate computing for NVPs, as shown in Figure 1. Instead of rolling back after power failure, incidental computing rolls forward to process the most recent and (presumably most important) new data. If there is additional power available beyond that needed to process the new data, then older data will be processed at reduced quality; incomplete executions from before a power failure are regarded as incidental and their importance drops over time, but even a low-effort/quality completion will often be preferable to skipping them entirely.

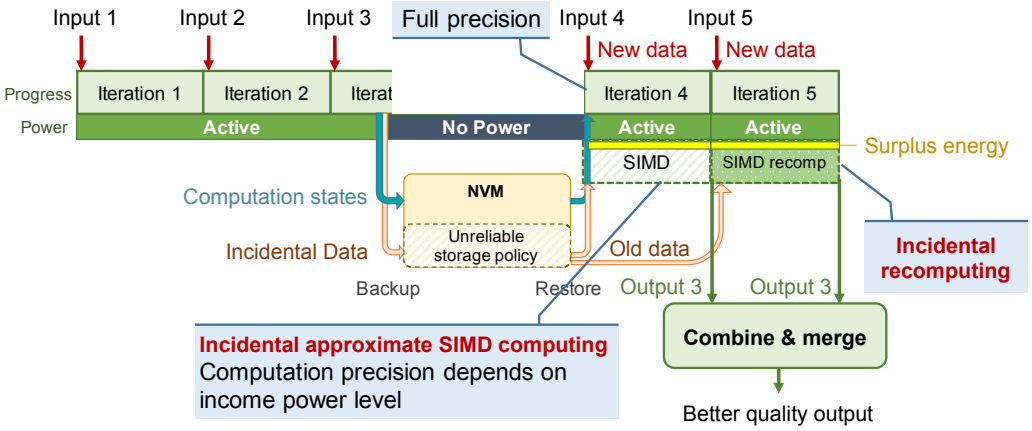

Figure 1. Incidental approximation concept.

Below, we discuss the details of incidental approximate computing as shown in Figure 1. When a power failure happens, the computation states are backed up with the stored energy, with some

data marked as incidental by the programmer in the compiling step. When the power recovers, instead of recovering from the backed-up PC, the PC is set to the beginning place for processing a new input. From the application's perspective, the program rolls forward to process newer data from the buffer. As a result, the newest captured data are always processed as the first priority. During processing of the new data, the microarchitecture controller compares the current computation state to the state backed up using the old data. If there is a match, an SIMD strategy is applied to the old state and data. Note that the computation precision of the newly added SIMD for old data depends on the income power level under the control of the programmer during the compiling step. In this way, a minimum quality can be guaranteed for the program controlled by the programmer, and the energy beyond the amount necessary for full precision processing of new data is instead applied to the old data for incidental SIMD computing. If the computation is interrupted again, both the new data and SIMD-ed old data become incidental, and a newest data computation begins.

Recompute and Combine (RAC)

We assume that, in general, the importance of data drops over time. If some old data are later found to be "interesting" and demand high-precision output to validate "uncommon results," an incidental recomputing can be performed. Instead of inserting an interrupt into the current program, incidental recomputing employs incidental SIMD to recompute the old data and tracks the precision of subcomponent outputs, as shown in Figure 1. These two versions of the outputs can then be merged by combining the best precision subcomponents from each run. After multiple recomputations and merges, we expect much-better-quality outputs. It is important to emphasize that, in this incidental recompute method, a better-quality result can be achieved without affecting the current data processing loop.

Incidental Backup

For the three power profiles used in this work (see Figure 2 in Ma *et al.*[3]), power supply unreliability would cause an NVP to perform as many as 1,400 to 1,700 backups per minute, costing 20.1 percent to 33 percent of the total income energy (simulated and measured with running MiBench[4]). Approximate computing provides an opportunity to substantially mitigate these overheads by:

- relaxing the reliability (write energy reduction brings the probability of flipped data storage beyond expected retention time) of the "lower-order" nonvolatile memory bits used to back up data during power emergencies, and
- using commensurately less energy for backup and recovery operations.

Moreover, if the energy reserves needed for backup are reduced, fewer power emergencies may occur.

Current NVPs[1,2] utilize nonvolatile technologies with maximum retention times on the order of a decade or more, as well as parameters tuned to maximize both retention and reliability. However, most power emergencies in wearable harvesting devices last just a few milliseconds and are rarely more than a fraction of a second. Figure 2 plots the duration (a) and frequency of power emergencies (b) in the examined traces. By matching the retention time to the power interval profile, the write energy can be significantly reduced. From the perspective of write energy for the backup operation, Figure 2 shows the relation between spin-transfer torque RAM (STT-RAM) write current and write pulse width for different retention times. We note that 77 percent of write energy can be saved, for instance, by reducing the retention time from one day to 10 ms. However, applying a retention time reduction uniformly is very difficult to implement profitably for two main reasons: (1) future power income is, in general, very difficult to predict and (2) the cost of prediction failures can be very high.

Approximate computing eases the practical adoption of such an approach. Higher-order bits are retained with longer duration, preventing catastrophic quality loss, while lower-order bits can be unreliably persisted, saving energy. We consider three retention time reduction functions to

shape the retention time in a way that reduces from the most-significant bit to the least-significant bit, as shown in Figure 2. We design retention policies based on observations of the relationship between bitwidth precision and final result quality, considering both program features and power source profiles.[5] We provide three retention time shaping policies to trade off between energy and qualities with parameters tuned through regression analysis: Linear, Log, and Parabola.

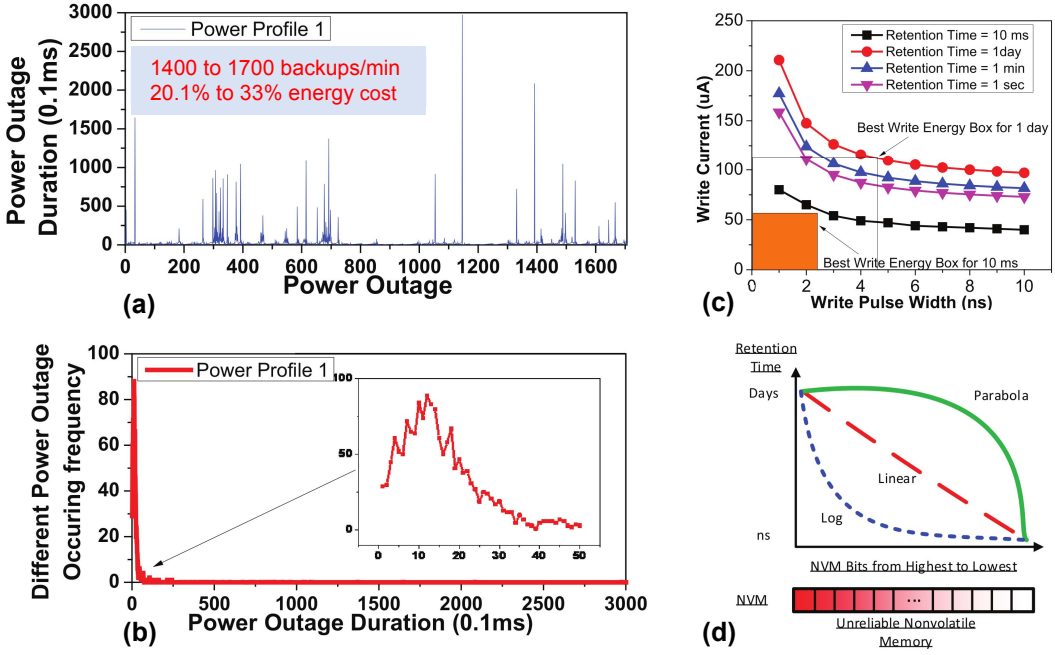

Figure 2. Incidental backup. (A) power outage duration, (b) statistics, (c) STT-RAM write energy/retention time,[6] and (d) proposed retention time shaping methods.

EVALUATION

In this section, we first evaluate the best incidental and low-precision parallel and assembling (LPA) approximation configurations for each kernel independently, and then we provide results from the holistic evaluation of the set of kernels as a data processing pipeline that both produces and consumes approximate data.

Our simulation framework consists of two parts. The first part is a functional simulator, the core of which is a modified 8051 RTL, which we further modified with support for incidental computing logic and approximate memory. For framework compatibility, the inputs are generated as ROM arrays, and the outputs are generated through general-purpose input/output (GPIO) P2 and P3. The RTL running in Modelsim initializes the ROM, RAM, and so on. The quality analysis for image outputs is performed by computing peak signal-to-noise ratio (PSNR) and mean squared error (MSE) in Matlab. The second part of our framework is a system-level simulator derived from the work by Ma *et al*.[1] This system-level simulation is implemented in Matlab, and Python handles the system-level components (including parameters and features of analog front-end circuits, capacitor, and so on), which cannot be implemented in RTL.

The inputs to this simulator are the power profiles sampled every 0.1 ms and the system configuration parameters such as the system capacitor size, capacitor leakage, chip leakage, front-end circuit efficiency, system start threshold, backup energy threshold, and recovery threshold. This system-level simulator controls the RTL simulator steps and gets the decoded instructions to decide various polices that dictate energy consumption. The system-level simulator, together with the functional simulator, generate important output metrics such as the amount of forward progress and the number of backups.

Real tasks for IoT deployments will consist of a mix of both precise and approximate kernels. A set of kernels are used in this article to represent a prototypical IoT workload and highlight the variation in potentials for approximability. Consider a typical system with sensors, CPU, storage, and wireless transmitter. The assumed functions are signal processing from sensors with FFT1024. The data are then run through the basic image processing kernels integral, sobel, and median and then analyzed by small-scale neural networks of 300 inferences of counter propagation network (CPN)-based angle detection and 300 inferences on digits recognition based on ADALINE. For pictures with important information, they are compressed by JPEG to reduce image size, followed by an encryption/signing through advanced encryption standard (AES) and ecure hash algorithm (SHA), after which the data are sent out by transmitter or stored in durable nonvolatile memory.

Figure 3 shows the forward progress gain of the approximate NVPs over the precise NVPs. The gain varies from 1.4X to 21.2X, depending on the kernel, approximation methodology, and power profiles. An average of 5.78X more forward progress can be brought by approximate computing. The largest gains were seen in inference kernels. A breakdown of gains in the ADALINE kernel shows 7.1X gains from LPA, 1.4X gains from incidental backup with retention time shaping (RTA), and 1.6-2.0X gains from incidental computing with MBL (dynamic bitwidth with power profiles with minimal bits limits configuration).

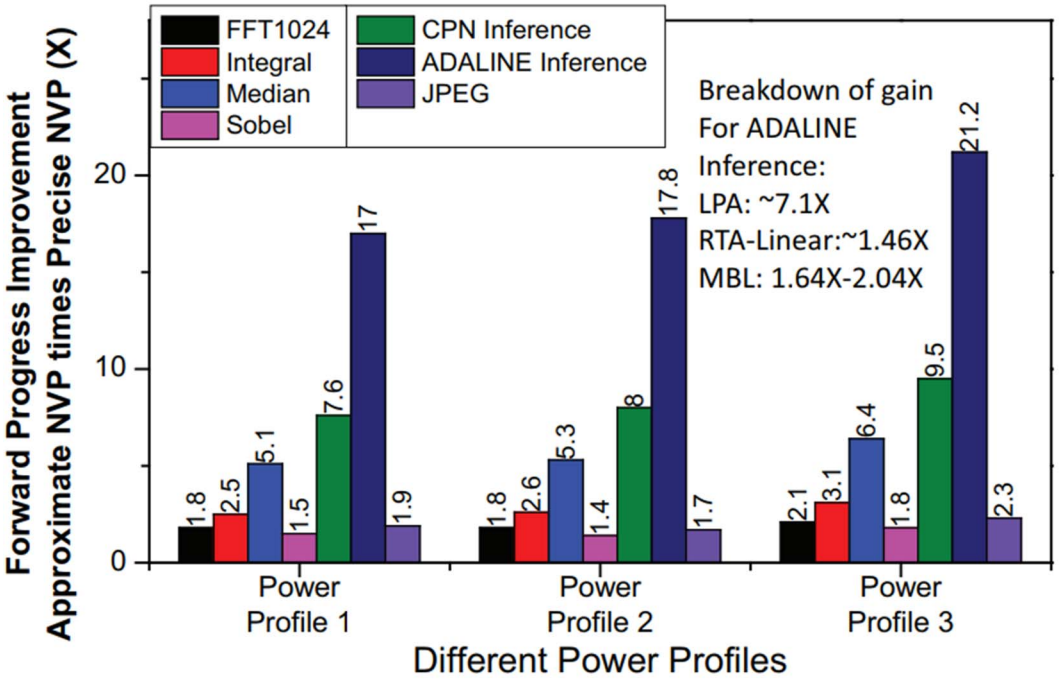

Figure 3. Forward progress gain brought by the approximate NVPs over the traditional precise NVPs.

Figure 4 illustrates the results of a system-level simulation running all of the kernels in sequence. Even with some precise computation in the mix, the NVP approximate computing system can reduce running time by an average of 83 percent. The running time of the approximate NVP is actually very comparable to a volatile processor powered with 33-µW stable power, achieving 98-percent, 129-percent, and 99-percent running time for power traces with average power of 32.87 µW, 21.63 µW, and 24.82 µW, respectively. This indicates that the approximate NVPs are able to buy back much of what was lost due to power instability with approximation efficiency.

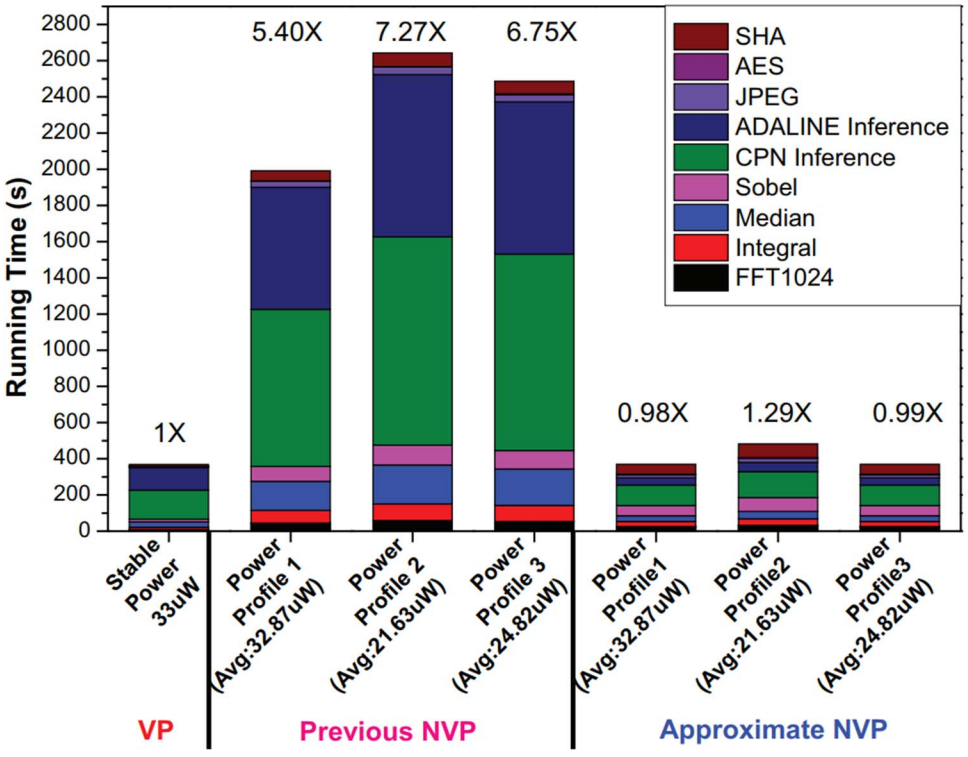

Figure 4. Execution time savings brought by the approximate NVPs over the traditional precise NVPs.

RELATED WORK

Nonvolatility in Processors

By designing distributed nonvolatile logic or elements at a microarchitectural level, computation state can be checkpointed before power outages occur with only on-chip energy storage and without programmer intervention.[7,8] Various materials can implement the nonvolatile features, such as ferroelectric RAM (FeRAM)-based NVPs,[9] resistive RAM (ReRAM)-based NVPs,[10] and magnetoresistive RAM (MRAM)-based NVPs.[11] Many cross-layer works leveraging integrated nonvolatility to address intermittency have also been explored, including OS and high-level synthesis approaches,[12,13] programming-language and compiler approaches,[14,15] hardware/software approaches,[16] and software-based approaches.[17] The different approaches for achieving continuous computation under unstable power supply have differing tradeoffs. In this work, rather than achieving continuous computation, we target incidental computing. By combining incidental computing and energy-harvesting NVPs, we find optimization opportunities in both incidental computation and backup.

Approximation

There is a substantial body of work focusing on approximate computing in the general-purpose computing domain. A statistical guarantee method in controlling quality has recently been proposed for an approximate accelerator.[18] Quality detection and error correction by exact recomputing on host processor is proposed by Khudia *et al*.[19] A pipeline-parallel approach for producing progressively higher-quality output across multi-kernel execution chains through iterative recomputation is described by San Miguel *et al*.[20] A self-tuning approximation with quality feedback control for graphics engines is proposed by Samadi *et al*.[21]

Another form of approximation is approximate storage.[22] Approximation is often a system-level approach, requiring support at multiple layers, cross-layer optimization, and co-design. Configurable tradeoffs between precision and energy are explored.[23] A "Rely" programming model for verifying unreliable hardware is developed,[24] but random power failures are not modeled. Approximation in energy harvesting is explored in software[25] but not targeted on NVPs.

Our approach's key point of divergence is optimizing approximate computing to a specific application scenario—energy harvesting—with the help of traditional NVPs to handle the unstable power supply. The application requirements of post-processing sensed data in real time and locally, with limited harvested energy, challenges traditional NVPs. As a result, approximate computing alone cannot solve the problem because the newly sensed data are still urgent to process, while historical buffered data's value drops over time. Observing this, our approach focuses on the incidental computing of historical buffered data and proposes incidental recomputing to enhance the quality without affecting processing the newest data. Incidental computing offers appealing opportunities in the notion of gradient approximate backup and recovery, which tries to match the data importance and retention time to power outages. In combination, NVPs, approximation, and incidental computing open new areas for optimizing energy-harvesting IoT systems.

CONCLUSION

Technology trends leading to the proliferation of IoT devices operating on harvested energy demand a corresponding revolution of the abilities of processors to adapt to unstable power supplies. Adopting approximate computing approaches in NVPs not only improves their forward progress, but also provides a means to optimize for responsiveness and efficiency, utilizes unique features of NVPs (namely, frequent backup and recovery operations), and matches these optimizations to the fundamental patterns present in IoT workloads. We explore the concept of IAA to address opportunistic responsiveness versus quality tradeoffs under unstable power income. Through experimentation with a workload consisting of an ensemble of applications that tolerate varying degrees of approximation and using various power profiles from energy-scavenged sources, our results reveal that an NVP can achieve similar computational progress with an unreliable power supply, by using approximations, to a volatile processor with a reliable power supply, under certain assumptions regarding acceptable output quality degradation. This article motivates the need for further work that more deeply explores approximate computation techniques in NVPs, especially efforts on understanding how quality-feedback loops can be provided both internally to NVP kernel processing pipelines and by the more robustly powered devices that ultimately consume the work performed on the NVPs.

ACKNOWLEDGMENTS

This work was supported in part by NSF ASSIST, in part by NSF Expeditions in Computing Award-1317560, and in part by the Center for Low Energy Systems Technology (one of the six Semiconductor Research Corporation (SRC) STARnet Centers sponsored by Microelectronics Advanced Research Corporation (MARCO) and DARPA). This work was also supported in part by National Natural Science Foundation of China (NSFC) Grant 61674094 and the Beijing Innovation Center for Future Chip.

REFERENCES

1. K. Ma et al., "Architecture exploration for ambient energy harvesting nonvolatile processors," *IEEE 21st International Symposium on High Performance Computer Architecture* (HPCA), 2015, pp. 526–537.
2. S. Khanna et al., "An FRAM-Based Nonvolatile Logic MCU SoC Exhibiting 100% Digital State Retention at VDD=0V Achieving Zero Leakage With < 400ns Wakeup

Time for ULP Applications," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 1, January 2014, pp. 95–106.
3. K. Ma et al., "Incidental computing on IoT nonvolatile processors," *50th Annual IEEE/ACM International Symposium on Microarchitecture* (MICRO), 2017, pp. 204–218.
4. M.R. Guthaus et al., "MiBench: A free, commercially representative embedded benchmark suite," *IEEE International Workshop on Workload Characterization*, 2001, pp. 3–14.
5. K. Ma et al., "Evaluating tradeoffs in granularity and overheads in supporting nonvolatile execution semantics," *18th International Symposium on Quality Electronic Design* (ISQED), 2017, pp. 39–44.
6. A. Jog et al., "Cache revive: architecting volatile STT-RAM caches for enhanced performance in CMPs," *Proceedings of the 49th Annual Design Automation Conference*, 2012, pp. 243–252.
7. G. Merrett et al., "Energy-driven computing: Rethinking the design of energy harvesting systems," *Design, Automation & Test in Europe Conference & Exhibition* (DATE), 2017, pp. 960–965.
8. A. Rodriguez et al., "Approaches to transient computing for energy harvesting systems: A quantitative evaluation," *Proceedings of the 3rd International Workshop on Energy Harvesting & Energy Neutral Sensing Systems*, 2015, pp. 3–8.
9. Y. Wang et al., "A 3us wake-up time nonvolatile processor based on ferroelectric flip-flops," *Proceedings of the ESSCIRC* (ESSCIRC), 2012, pp. 149–152.
10. Y. Liu et al., "A 65nm ReRAM-enabled nonvolatile processor with 6× reduction in restore time and 4× higher clock frequency using adaptive data retention and self-write-termination nonvolatile logic," *IEEE International Solid-State Circuits Conference* (ISSCC), 2016, pp. 84–86.
11. S. Senni et al., "Non-volatile processor based on MRAM for ultra-low-power IoT devices," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 13, no. 2, 2017.
12. J. Van Der Woude and H. Matthew, "Intermittent Computation without Hardware Support or Programmer Intervention," *12th USENIX Symposium on Operating Systems Design and Implementation* (OSDI), 2016, pp. 17–32.
13. A. Mirhoseini, M.S. Ebrahim, and K. Farinaz, "Idetic: A high-level synthesis approach for enabling long computations on transiently-powered ASICs," *IEEE International Conference on Pervasive Computing and Communications* (PerCom), 2013, pp. 216–224.
14. A. Colin and L. Brandon, "Chain: tasks and channels for reliable intermittent programs," *ACM SIGPLAN Notices - OOPSLA*, vol. 51, no. 10, 2016, pp. 514–530.
15. B. Ransford, J. Sorber, and K. Fu, "Mementos: System support for long-running computation on RFID-scale devices," *ACM SIGPLAN Notices*, vol. 47, no. 4, 2012, pp. 159–170.
16. H. Jayakumar, R. Arnab, and R. Vijay, "QuickRecall: A low overhead HW/SW approach for enabling computations across power cycles in transiently powered computers," *27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*, 2014, pp. 330–335.
17. D. Balsamo et al., "Hibernus++: a self-calibrating and adaptive system for transiently-powered embedded devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 12, 2016, pp. 1968–1980.
18. D. Mahajan et al., "Towards statistical guarantees in controlling quality tradeoffs for approximate acceleration," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, 2016, pp. 66–77.
19. D. Khudia et al., "Rumba: An online quality management system for approximate computing," *ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA)*, 2015, pp. 554–566.
20. J. San Miguel and N. Enright Jerger, "The anytime automaton," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, 2016, pp. 545–557.
21. M. Samadi et al., "Sage: Self-tuning approximation for graphics engines," *46th Annual IEEE/ACM International Symposium on Microarchitecture* (MICRO), 2013, pp. 13–24.
22. K. Ma et al., "Nonvolatile processor optimization for ambient energy harvesting scenarios," *Non-volatile Memory Technology Symposium* (NVMTS), 2015.

23. N. Mishra et al., "A probabilistic graphical model-based approach for minimizing energy under performance constraints," *ACM SIGARCH Computer Architecture News*, vol. 43, no. 1, 2015, pp. 267–281.
24. M. Carbin, M. Sasa, and C.R. Martin, "Verifying quantitative reliability for programs that execute on unreliable hardware," *ACM SIGPLAN Notices*, vol. 48, no. 10, 2013, pp. 33–52.
25. A. Sampson et al., *Accept: A programmer-guided compiler framework for practical approximate computing*, technical report UW-CSE-15-01, University of Washington, 2015.

ABOUT THE AUTHORS

Kaisheng Ma is an assistant professor at the Institute for Interdisciplinary Information Sciences (IIIS) at Tsinghua University. His research interests include nonvolatile processor architecture, machine learning, big data, neural networks, and neuromorphic computing. Ma has a PhD in computer science and engineering from Pennsylvania State University. Contact him at kaishengthu@163.com.

Jinyang Li is a master's degree candidate in the Department of Electronic Engineering at Tsinghua University. His research interests include nonvolatile processor-based applications and wearable devices in the wireless healthcare field. Li has a bachelor's degree in electronic engineering from Tsinghua University. Contact him at lijy15@mails.tsinghua.edu.cn.

Xueqing Li is an assistant professor in the Department of Electronic Engineering at Tsinghua University. His research interests include high-performance data converters, wireless transceivers, self-powered nonvolatile systems, and circuits and systems using emerging devices. Li has a PhD in electronics engineering from Tsinghua University. He is a member of the IEEE. Contact him at xueqingli@tsinghua.edu.cn.

Yongpan Liu is an associate professor in the Department of Electronic Engineering at Tsinghua University. His research interests include nonvolatile computation, low-power VLSI design, emerging circuits and systems, and design automation. Liu has a PhD in electronics engineering from Tsinghua University. He is a member of the IEEE; ACM; and Institute of Electronics, Information, and Communication Engineers. Contact him at ypliu@tsinghua.edu.cn.

Yuan Xie is a professor in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. His research interests include computer architecture, electronic design automation, and VLSI design. Xie has a PhD in electrical engineering from Princeton University. He is an IEEE Fellow. Contact him at yuanxie@ece.ucsb.edu.

Mahmut Kandemir is a professor in the Department of Computer Science and Engineering at Pennsylvania State University. His research interests include optimizing compilers, runtime systems, embedded systems, I/O and high-performance storage, and power-aware computing. Kandemir has a PhD in electrical engineering and computer science from Syracuse University. He is an IEEE Fellow. Contact him at kandemir@cse.psu.edu.

Jack Sampson is an assistant professor in the Department of Computer Science and Engineering at Pennsylvania State University. His research interests include low-power and energy-efficient computing platforms, microarchitectural specialization, and the application of emerging logic and memory devices to novel computing platforms. Sampson has a PhD in computer science from the University of California at San Diego. He is a member of the IEEE. Contact him at sampson@cse.psu.edu.

Vijaykrishnan Narayanan is a Distinguished Professor of Computer Science and Engineering and Electrical Engineering at Pennsylvania State University. His research focuses on energy-efficient computing. Narayanan has a PhD in computer science from the University of South Florida. He is a Fellow of the IEEE and ACM. Contact him at vijay@cse.psu.edu.