Approximate Hardware Techniques for Energy-Quality Scaling Across the System

Younghyun Kim, Joshua San Miguel, Setareh Behroozi,
Tianen Chen, Kyuin Lee, Yongwoo Lee, Jingjie Li, and Di Wu
Department of Electrical and Computer Engineering
University of Wisconsin–Madison
Madison, Wisconsin
{younghyun.kim, jsanmiguel, sbehroozi, tianen.chen, kyuin.lee, yongwoo.lee, jingjie.li}@wisc.edu, di.wu@ece.wisc.edu

***Abstract*—For error-resilient applications, such as machine learning and signal processing, a significant improvement in energy efficiency can be achieved by relaxing exactness constraint on output quality. This paper presents a taxonomy of hardware techniques to exploit the trade-off between energy efficiency and quality in various computer subsystems. We classify approximate hardware techniques according to target subsystem and support for dynamic energy-quality scaling.**

***Index Terms*—energy-quality scaling, approximate computing, survey**

I. Introduction

Energy efficiency is a major concern in computer system design, which dictates the performance, operating cost, lifetime, and physical dimensions. A common approach to save energy is to exploit timing slack, such as in dynamic power management (DPM) and dynamic voltage and frequency scaling (DVFS) that scale down computing performance by lowering supply voltage and operating frequency as long as the correctness of results is not compromised. *Approximate computing* is a new computing paradigm to exploit *quality slack* between the minimum quality required by the application and the maximum quality producible by the system [1]–[3]. By relaxing correctness constraint, more aggressive power-saving techniques can be applied to certain error-resilient applications, such as machine learning, signal processing, and multimedia applications, to explore *energy-quality trade-offs* without resulting in significant quality degradation.

While *compute* subsystems for *data processing* is a key element in computer systems, they are not the only subsystems that account for a major share of energy consumption, particularly in low-power systems. *Non-compute* subsystems for *data acquisition, transfer, and storage* as well as for *user interaction*, such as sensors, actuators, user interfaces, and network interfaces, also consume a significant portion of energy. Therefore, it is important to identify energy-quality trade-offs in these subsystems and exploit them to enable *full-system energy-quality scaling* to achieve high energy efficiency that approximate computing alone cannot deliver.

In this paper, we present a taxonomy of approximate hardware techniques encompassing both compute and non-compute subsystems. We classify techniques according to their *target subsystem* and *dynamic scalability*, as shown in Table I. We categorize target subsystems into i) processors and logic, ii) memory and storage, iii) input and output, and iv) interconnects and communication. An approximate technique is deemed dynamically scalable if its energy-quality trade-off can be adjusted at runtime within a certain range. Otherwise, i.e., if the energy-quality trade-off is set at design time, it is categorized as a static technique. Software-level approximate computing techniques, such as loop perforation [4] and lossy compression [5], as well as application-specific techniques, such as those for neural networks [6], [7], are not in the scope of this paper.

This work was supported in part by the Wisconsin Alumni Research Foundation and NSF under award CNS-1845469.

II. Taxonomy

We categorize approximate hardware techniques into four based on their target subsystem. Note that some techniques appear multiple times if they are applicable to different subsystems.

A. Processors and Logic

Processors and logic circuits are the main target subsystems of various *approximate computing* techniques, which include both general-purpose processors and special-purpose custom logic circuits. Various approximate computing techniques have been proposed at different levels of computing abstraction—at the device level, logic level, and architecture level.

Approximate arithmetic units perform near-accurate arithmetic calculations at a much lower energy cost using smaller core arithmetic units, with a superior energy-quality trade-off than that of simple truncation. The energy-accuracy trade-offs of static approximate arithmetic units are mainly determined by the design parameters of the core arithmetic units, such as bit width (BW) [8]–[10], [13]–[21], [24], [26], [27]. On the other hand, some approximate arithmetic units support dynamic scaling by selectively applying error correction or performing progressive Taylor approximation [11], [12], [22], [23], [25], [28].

Approximate load value prediction is a processor-level approximation technique to mitigate the memory wall problem. Observing the error resilience of applications, when program data yields a load miss in the processor cache, its

TABLE I
TAXONOMY OF APPROXIMATE HARDWARE TECHNIQUES.

Technique	Subsystem	References	Quality control knobs	Trade-off decision	
				Static	Dynamic
Processors and logic					
Approx. arithmetic units	Adder	[8]–[10]	Core arithmetic unit BW	✓	
		[11], [12]	Error correction		✓
	Multiplier	[13]–[17]	Core arithmetic unit BW	✓	
	Divider	[18]–[21]	Core arithmetic unit BW	✓	
		[22], [23]	Taylor approximation order		✓
	Exponentiation	[24]	Taylor approximation order	✓	
		[25]	Taylor approximation order		✓
	Logarithm	[26], [27]	Core arithmetic unit BW	✓	
		[28]	Taylor approximation order		✓
Approx. load value prediction	Microprocessor	[29]–[33]	Cache miss/fetch ratio		✓
Voltage overscaling (VOS)	Logic circuit	[34]–[41]	Supply voltage		✓
Approx. logic synthesis (ALS)		[42]–[47]	Boolean function exactness	✓	
Clock overgating		[48]	Clock gating schedule	✓	
Memory and storage					
Approx. memory	SRAM	[49], [50]	Cell structure and size	✓	
		[49], [51]–[53]	Supply voltage		✓
	DRAM	[54]–[56]	Refresh rate		✓
		[57]	Supply voltage		✓
		[58]	Restore time		✓
Approx. storage	Non-volatile memory (NVM)	[59], [60]	Guard band width		✓
		[61]–[63]	Error correction		✓
		[64]	Reusing faulty blocks		✓
Input and output					
Approx. sensing	Off-chip sensors	[65], [66]	Supply voltage		✓
Approx. displays	LCD	[67], [68]	Backlight brightness		✓
	OLED	[69]–[71]	Supply voltage		✓
Interconnects and communication					
Approx. interconnects	On-chip interconnect	[72], [73]	Error threshold		✓
		[74]	Lossy injection rate		✓
	Off-chip interconnect	[73], [75]–[77]	Error threshold		✓
		[78]	Error threshold	✓	
		[79]	Memory access BW		✓
Approx. wireless networks	WiFi	[80]	Error correction and retransmission		✓
	On-chip wireless interconnect	[81]	Latency threshold		✓

value can be predicted approximately with a learning-based predictor. This approximate prediction is not speculative (thus no rollback is required) and eliminates some data fetches, reducing the energy and latency of accessing off-chip memory. This technology is available for CPUs with minimal hardware overhead [29], [30], [32], [33] as well as GPUs with no extra logic [31]. The energy-quality trade-off is indicated by the approximation degree, i.e., the ratio of actual memory fetches and cache misses [29].

Voltage overscaling (VOS) is a technique that lowers supply voltage below the minimum level required to meet the timing constraint of the circuit to reduce dynamic and static power consumption at the cost of timing errors at critical path [34]–[41]. VOS typically supports dynamic scaling through runtime supply voltage adaptation. This technique is widely applicable to a broad range of subsystems, including logic, memory, and even sensors, if operation reliability is gracefully degraded when the supply voltage is lowered.

Another technique that is generally applicable to logic circuits is **approximate logic synthesis (ALS)** that synthesizes a given Boolean function into a logic circuit that generates partially correct output. This design-time technique is performed by identifying and removing highly-correlated signals, leaving only one of them, or by introducing don't-care terms in the logic function to allow aggressive logic simplification [42]–[47]. Similarly, **clock overgating** reduces dynamic power consumption by gating the clock signal to flip-flops more aggressively, with some loss of exactness in logic output [48].

B. Memory and Storage

A variety of *approximate memory* and *approximate storage* techniques have been proposed at different levels of the memory hierarchy. A common technique in this approach is to divide memory space into error-prone regions and error-free regions for approximable data and non-approximable data, respectively. Error-prone regions are often further divided into multiple sub-regions with different error rates.

Design-time approximate SRAM techniques include using **heterogeneous cell structures** (both 6-T and 8-T cells) and **heterogeneous cell sizes** [49], [50]. More significant bits are stored in stronger cells since errors in these locations are more pronounced than errors in less significant bits. Accessing DRAM is performed using sequences of commands with strict timing constraints, and violating the constraints results in unstable reads and writes. For example, each DRAM cell must be refreshed every 64 ms to prevent retention errors due to leakage, and this operation is responsible for 10–50% of total power consumption and 10–45% of throughput loss in DRAM [82]. **Refresh rate reduction** is a runtime technique that relaxes the minimum refresh interval requirement and uses much longer intervals of up to tens of seconds for the regions designated for approximate data [54]–[56]. Similarly, using a shorter write recovery time than the minimum constraint will cause bit errors but improve power efficiency and throughput [58]. The above-mentioned VOS technique can also be applied to both SRAM [49], [51]–[53] and DRAM [57].

Approximate storage techniques on non-volatile memory (NVM), such as NAND Flash and phase-change memory (PCM), typically aim at improving throughput and lifetime, rather than energy efficiency. A common technique for approximate NVM is to **reduce the width of guard bands**, which is the gap between the cell threshold voltage or cell resistance distributions of adjacent symbols. Wide guard bands prevent confusion between adjacent symbols but require an iterative, fine-grained program-and-verify write procedure to precisely set the threshold voltage or resistance. Therefore, allowing narrow guard bands, either by using coarse-grained program-and-verify writes [59] or by decreasing the overall range [60], the number of iterative writes can be reduced.

NVMs typically use error correction codes (ECC) to fix bit errors, which incur both storage area overhead and latency overhead. Therefore, **selectively applying ECC** only to non-approximable data reduces read and write latencies for performing error correction and free the space reserved for ECC for other purposes [61]–[63]. Unlike these active approaches that intentionally incur errors in normal cells, [64] is a passive approach that tolerates unavoidable errors due to cell wear-out. This approach aims to prolong the lifetime of NVM by reusing blocks with too many dead cells as approximate data storage.

C. Input and Output

Computer systems interact with the physical world, including human users and other systems, through input and output devices. It is common that embedded systems are equipped with various sensors, such as in wearable devices, which often consume a significant amount of energy for continuous sensing [66]. Reducing the sampling rate or precision of sensing is a widely used technique for saving energy at the expense of low-resolution measurement. The application of VOS has been studied for sensors, such as accelerometer, magnetometer, barometer, thermometer, etc [65], [66].

Display is one of the most power-hungry components in mobile devices [83]. Blacklight dimming is an effective power-saving technique in LCDs at the cost of reduced screen brightness and contrast that results in user experience degradation. To compensate for the loss of brightness, pixel values can be adjusted to be perceived as the original bright pixel with brighter backlight [67], [68], but some distortion of bright images is unavoidable. OLED displays that have no backlight can also take advantage of a similar energy-quality trade-off by lowering the supply voltage level of the pixel array to save power [69]–[71]. They can be considered as an **approximate display** technique that trade user-perceived quality for power.

D. Interconnects and communication

On-chip interconnects consume a significant amount of energy in modern system-on-chip (SoC) devices, up to 50% [84]. The energy consumption of off-chip interconnects also is not negligible when the system requires heavy off-chip memory access or continuous reading of high-bandwidth sensors [76]. **Approximate on-chip and off-chip interconnect** techniques aim to reduce power consumption of the transfer of approximable data within the SoC [72]–[74] or between the SoC and off-chip components [73], [75]–[79]. These techniques largely rely on the fact that the data values have a non-uniform distribution or a high spatiotemporal locality.

Unlike on-chip or off-chip interconnects that are relatively free from transmission errors, wireless interfaces consume significant energy for preventing or correcting transmission errors. **Approximate wireless interface** techniques reduce the energy overhead of forward error correction and retransmission by just accepting erroneous packets while minimizing the impact of errors by assigning similar bits to adjacent symbols [80]. In [81], approximate wireless communication is used for on-chip communication, where the energy-quality trade-off is controlled by dropping packets.

III. Conclusion

We presented a taxonomy that categorizes approximate hardware techniques for various subsystems across the system, based on their target components and dynamic scalability. These techniques offer a great potential of energy savings for various emerging error-resilient applications, such as machine learning, yet their adoption is largely limited to isolated application, rather than the integrated application of multiple techniques to achieve the system-level optimality. To take full advantage of energy-quality scalability across the system, a full-system design framework is required to efficiently handle the increased design complexity due to the new added dimension—“quality” [85].

References

[1] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” in *IEEE European Test Symposium (ETS)*, 2013, pp. 1–6.

[2] S. Venkataramani, S. T. Chakradhar, K. Roy, and A. Raghunathan, “Approximate computing and the quest for computing efficiency,” in *Design Automation Conference (DAC)*, 2015, pp. 120:1–120:6.

[3] M. Alioto, “Energy-quality scalable adaptive VLSI circuits and systems beyond approximate computing,” in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2017, pp. 127–132.

[4] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. Rinard, "Managing performance vs. accuracy trade-offs with loop perforation," in *ACM European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 2011, pp. 124–134.

[5] M. Samadi, J. Lee, D. A. Jamshidi, A. Hormati, and S. Mahlke, "SAGE: Self-tuning approximation for graphics engines," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2013, pp. 13–24.

[6] Q. Zhang, T. Wang, Y. Tian, F. Yuan, and Q. Xu, "ApproxANN: An approximate computing framework for artificial neural network," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp. 701–706.

[7] E. H. Lee, D. Miyashita, E. Chai, B. Murmann, and S. S. Wong, "LogNet: Energy-efficient neural networks using logarithmic computation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5900–5904.

[8] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: Imprecise adders for low-power approximate computing," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2011, pp. 409–414.

[9] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, "Approximate XOR/XNOR-based adders for inexact computing," in *IEEE International Conference on Nanotechnology (NANO)*, 2013, pp. 690–693.

[10] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, "EvoApprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, pp. 258–261.

[11] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2013, pp. 48–54.

[12] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *Design Automation Conference (DAC)*, 2012, pp. 820–825.

[13] C. Lin and I. Lin, "High accuracy approximate multiplier with error correction," in *IEEE International Conference on Computer Design (ICCD)*, 2013, pp. 33–38.

[14] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2014, pp. 1–4.

[15] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power- and area-efficient approximate Wallace tree multiplier for error-resilient systems," in *International Symposium on Quality Electronic Design (ISQED)*, 2014, pp. 263–269.

[16] S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: a dynamic range unbiased multiplier for approximate applications," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2015, pp. 418–425.

[17] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, 2015.

[18] R. Zendegani, M. Kamal, A. Fayyazi, A. Afzali-Kusha, S. Safari, and M. Pedram, "SEERAD: A high speed yet energy-efficient rounding-based approximate divider," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016, pp. 1481–1484.

[19] S. Vahdat, M. Kamal, A. Afzali-Kusha, M. Pedram, and Z. Navabi, "TruncApp: A truncation-based approximate divider for energy efficient DSP applications," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, pp. 1635–1638.

[20] H. Jiang, L. Liu, F. Lombardi, and J. Han, "Adaptive approximation in arithmetic circuits: A low-power unsigned divider design," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 1411–1416.

[21] H. Saadat, H. Javaid, and S. Parameswaran, "Approximate integer and floating-point dividers with near-zero error bias," in *Design Automation Conference (DAC)*, 2019, pp. 161:1–161:6.

[22] S. Behroozi, J. Li, J. Melchert, and Y. Kim, "SAADI: A scalable accuracy approximate divider for dynamic energy-quality scaling," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2019, pp. 481–486.

[23] J. Melchert, S. Behroozi, J. Li, and Y. Kim, "SAADI-EC: A quality-configurable approximate divider for energy efficiency," *IEEE Transactions on VLSI Systems*, vol. 27, no. 11, pp. 2680–2692, 2019.

[24] P. Nilsson, A. U. R. Shaik, R. Gangarajaiah, and E. Hertz, "Hardware implementation of the exponential function using Taylor series," in *IEEE Nordic Circuits and Systems Conference (NORCHIP)*, 2014, pp. 1–4.

[25] D. Wu, T. Chen, C. Chen, O. Ahia, J. San Miguel, M. Lipasti, and Y. Kim, "SECO: A scalable accuracy approximate exponential function via cross-layer optimization," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2019, pp. 1–6.

[26] J. N. Mitchell, "Computer multiplication and division using binary logarithms," *IRE Transactions on Electronic Computers*, no. 4, pp. 512–517, 1962.

[27] M. B. Sullivan and E. E. Swartzlander, "Truncated logarithmic approximation," in *IEEE Symposium on Computer Arithmetic (ARITH)*, 2013, pp. 191–198.

[28] M. R. Weirich, G. Paim, E. A. C. da Costa, and S. Bampi, "A fixed-point natural logarithm approximation hardware design using Taylor series," in *New Generation of Circuits & Systems Conference (NGCAS)*, 2018, pp. 53–56.

[29] J. San Miguel, M. Badr, and N. E. Jerger, "Load value approximation," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2014, pp. 127–139.

[30] B. Thwaites, G. Pekhimenko, H. Esmaeilzadeh, A. Yazdanbakhsh, J. Park, G. Mururu, O. Mutlu, and T. Mowry, "Rollback-free value prediction with approximate loads," in *International Conference on Parallel Architecture and Compilation Techniques (PACT)*, 2014, pp. 493–494.

[31] M. Sutherland, J. San Miguel, and N. E. Jerger, "Texture cache approximation on gpus," in *Workshop on Approximate Computing Across the Stack*, 2015.

[32] A. Yazdanbakhsh, B. Thwaites, H. Esmaeilzadeh, G. Pekhimenko, O. Mutlu, and T. C. Mowry, "Mitigating the memory bottleneck with approximate load value prediction," *IEEE Design and Test*, vol. 33, no. 1, pp. 32–42, 2016.

[33] A. Yazdanbakhsh, G. Pekhimenko, B. Thwaites, H. Esmaeilzadeh, O. Mutlu, and T. C. Mowry, "RFVP: Rollback-free value prediction with safe-to-approximate loads," *ACM Transactions on Architecture and Code Optimization*, vol. 12, no. 4, pp. 62:1–62:26, 2016.

[34] P. K. Krause and I. Polian, "Adaptive voltage over-scaling for resilient applications," in *Design, Automation & Test in EuropeConference & Exhibition (DATE)*, 2011, pp. 1–6.

[35] S. Lee, L. K. John, and A. Gerstlauer, "High-level synthesis of approximate hardware under joint precision and voltage scaling," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, pp. 187–192.

[36] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *Transactions on VLSI Systems*, vol. 9, no. 6, pp. 813–823, 2001.

[37] A. B. Kahng, S. Kang, R. Kumar, and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2010, pp. 825–831.

[38] G. Zervakis, K. Koliogeorgi, D. Anagnostos, N. Zompakis, and K. Siozios, "VADER: Voltage-driven netlist pruning for cross-layer approximate arithmetic circuits," *IEEE Transactions on VLSI Systems*, vol. 27, no. 6, pp. 1460–1464, 2019.

[39] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, "Design of voltage-scalable meta-functions for approximate computing," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2011, pp. 1–6.

[40] H. Amrouch, S. B. Ehsani, A. Gerstlauer, and J. Henkel, "On the efficiency of voltage overscaling under temperature and aging effects," *IEEE Transactions on Computers*, vol. 68, no. 11, pp. 1647–1662, 2019.

[41] V. K. Chippa, D. Mohapatra, K. Roy, S. T. Chakradhar, and A. Raghunathan, "Scalable effort hardware design," *IEEE Transactions on VLSI Systems*, vol. 22, no. 9, pp. 2004–2016, 2014.

[42] S. Venkataramani, A. Sabne, V. Kozhikkottu, K. Roy, and A. Raghunathan, "SALSA: systematic logic synthesis of approximate circuits," in *Design Automation Conference (DAC)*. IEEE, 2012, pp. 796–801.

[43] C. Li, W. Luo, S. S. Sapatnekar, and J. Hu, "Joint precision optimization and high level synthesis for approximate computing," in *Design Automation Conference (DAC)*, 2015, pp. 1–6.

[44] Y. Wu and W. Qian, "An efficient method for multi-level approximate logic synthesis under error rate constraint," in *Design Automation Conferece (DAC)*, 2016, pp. 128:1–128:6.

[45] G. Pasandi, S. Nazarian, and M. Pedram, "Approximate logic synthesis: A reinforcement learning-based technology mapping approach," in *International Symposium on Quality Electronic Design (ISQED)*, 2019, pp. 26–32.

[46] J. Miao, A. Gerstlauer, and M. Orshansky, "Multi-level approximate logic synthesis under general error constraints," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 504–510.

[47] S. Venkataramani, K. Roy, and A. Raghunathan, "Substitute-and-simplify: A unified design paradigm for approximate and quality configurable circuits," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2013, pp. 1367–1372.

[48] Y. Kim, S. Venkataramani, K. Roy, and A. Raghunathan, "Designing approximate circuits using clock overgating," in *Design Automation Conference (DAC)*, 2016, pp. 1–6.

[49] I. J. Chang, D. Mohapatra, and K. Roy, "A priority-based 6T/8T hybrid SRAM architecture for aggressive voltage scaling in video applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 2, pp. 101–112, 2011.

[50] J. Kwon, I. J. Chang, I. Lee, H. Park, and J. Park, "Heterogeneous SRAM cell sizing for low-power H.264 applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 10, pp. 2275–2284, 2012.

[51] L. Yang and B. Murmann, "SRAM voltage scaling for energy-efficient convolutional neural networks," in *International Symposium on Quality Electronic Design (ISQED)*, 2017, pp. 7–12.

[52] F. Frustaci, M. Khayatzadeh, D. Blaauw, D. Sylvester, and M. Alioto, "SRAM for error-tolerant applications with dynamic energy-quality management in 28 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 5, pp. 1310–1323, 2015.

[53] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger, "Architecture support for disciplined approximate programming," in *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2012, pp. 301–312.

[54] S. Liu, K. Pattabiraman, T. Moscibroda, and B. G. Zorn, "Flikker: Saving DRAM refresh-power through critical data partitioning," in *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2011, pp. 213–224.

[55] A. Raha, H. Jayakumar, S. Sutar, and V. Raghunathan, "Quality-aware data allocation in approximate DRAM," in *International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, 2015, pp. 89–98.

[56] M. Jung, D. M. Mathew, C. Weis, and N. Wehn, "Approximate computing with partially unreliable dynamic random access memory—Approximate DRAM," in *Design Automation Conference (DAC)*, 2016, pp. 1–4.

[57] S. Koppula, L. Orosa, A. G. Yağlıkçı, R. Azizi, T. Shahroodi, K. Kanellopoulos, and O. Mutlu, "EDEN: Enabling energy-efficient, high-performance deep neural network inference using approximate DRAM," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2019, pp. 166–181.

[58] X. Zhang, Y. Zhang, B. R. Childers, and J. Yang, "DrMP: Mixed precision-aware DRAM for high performance approximate and precise computing," in *International Conference on Parallel Architectures and Compilation Techniques (PACT)*, 2017, pp. 53–63.

[59] A. Sampson, J. Nelson, K. Strauss, and L. Ceze, "Approximate storage in solid-state memories," in *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2013, pp. 25–36.

[60] J. Cui, Y. Zhang, L. Shi, C. J. Xue, W. Wu, and J. Yang, "ApproxFTL: On the performance and lifetime improvement of 3-D NAND Flash-based SSDs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 10, pp. 1957–1970.

[61] Q. Guo, K. Strauss, L. Ceze, and H. S. Malvar, "High-density image storage using approximate memory cells," in *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2016, pp. 413–426.

[62] F. Li, Y. Lu, Z. Wu, and J. Shu, "ASCache: An approximate SSD cache for error-tolerant applications," in *Design Automation Conference (DAC)*, 2019, pp. 214:1–214:6.

[63] Q. Li, L. Shi, J. Yang, Y. Zhang, and C. J. Xue, "Leveraging approximate data for robust Flash storage," in *Design Automation Conference (DAC)*, 2019, pp. 215:1–215:6.

[64] L. Han, H. Amrouch, Z. Shao, and J. Henkel, "Rebirth-FTL: Lifetime optimization via approximate storage for NAND Flash," in *IEEE Non-Volatile Memory Systems and Applications Symposium (NVMSA)*, 2019, pp. 1–6.

[65] P. Stanley-Marbell and M. Rinard, "A hardware platform for efficient multi-modal sensing with adaptive approximation," *arXiv:1804.09241*, 2018.

[66] ——, "Lax: Driver interfaces for approximate sensor device access," in *ACM Hot Topics in Operating Systems (HoTOS)*, 2015, pp. 27–27.

[67] B. Anand, K. Thirugnanam, J. Sebastian, P. G. Kannan, A. L. Ananda, M. C. Chan, and R. K. Balan, "Adaptive display power management for mobile games," in *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2011, pp. 57–70.

[68] N. Chang, I. Choi, and H. Shim, "Dls: dynamic backlight luminance scaling of liquid crystal display," *Transactions on VLSI Systems*, vol. 12, no. 8, pp. 837–846, 2004.

[69] D. Shin, Y. Kim, N. Chang, and M. Pedram, "Dynamic voltage scaling of OLED displays," in *Design Automation Conference (DAC)*, 2011, pp. 53–58.

[70] X. Chen, J. Zheng, Y. Chen, M. Zhao, and C. J. Xue, "Quality-retaining OLED dynamic voltage scaling for video streaming applications on mobile devices," in *Design Automation Conference (DAC)*, 2012, pp. 1000–1005.

[71] D. Shin, Y. Kim, N. Chang, and M. Pedram, "Dynamic driver supply voltage scaling for organic light emitting diode displays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 7, pp. 1017–1030, 2013.

[72] R. Boyapati, J. Huang, P. Majumder, K. H. Yum, and E. J. Kim, "APPROX-NoC: A data approximation framework for network-on-chip architectures," in *International Symposium on Computer Architecture (ISCA)*, 2017, pp. 666–677.

[73] J. R. Stevens, A. Ranjan, and A. Raghunathan, "AxBA: An approximate bus architecture framework," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018, pp. 1–8.

[74] Z. Li, J. San Miguel, and N. E. Jerger, "The runahead network-on-chip," in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2016, pp. 333–344.

[75] Y. Kim, S. Behroozi, V. Raghunathan, and A. Raghunathan, "AxSerBus: A quality-configurable approximate serial bus for energy-efficient sensing," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2017, pp. 1–6.

[76] S. Behroozi, V. Raghunathan, A. Raghunathan, and Y. Kim, "A quality-configurable approximate serial bus for energy-efficient sensory data transfer," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 3, pp. 379–390, 2018.

[77] D. J. Pagliari, E. Macii, and M. Poncino, "Serial T0: Approximate bus encoding for energy-efficient transmission of sensor signals," in *Design Automation Conference (DAC)*, 2016, pp. 1–6.

[78] P. Stanley-Marbell and M. Rinard, "Reducing serial I/O power in error-tolerant applications by efficient lossy encoding," in *Design Automation Conference (DAC)*, 2016, pp. 62:1–62:6.

[79] Y. Tian, Q. Zhang, T. Wang, F. Yuan, and Q. Xu, "ApproxMA: Approximate memory access for dynamic precision scaling," in *Great Lakes Symposium on VLSI (GLSVLSI)*, 2015, pp. 337–342.

[80] S. Sen, S. Gilani, S. Srinath, S. Schmitt, and S. Banerjee, "Design and implementation of an "approximate" communication system for wireless media applications," in *ACM SIGCOMM Conference (SIGCOMM)*, 2010, pp. 15–26.

[81] V. Fernando, A. Franques, S. Abadal, S. Misailovic, and J. Torrellas, "Replica: A wireless manycore for communication-intensive and approximate data," in *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2019, pp. 849–863.

[82] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu, "RAIDR: Retention-aware intelligent DRAM refresh," in *International Symposium on Computer Architecture (ISCA)*, 2012, pp. 1–12.

[83] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone." in *USENIX Annual Technical Conference (ATC)*, 2010.

[84] V. Raghunathan, M. B. Srivastava, and R. K. Gupta, "A survey of techniques for energy efficient on-chip communication," in *Design Automation Conference (DAC)*, 2003, pp. 900–905.

[85] A. Raha and V. Raghunathan, "Approximating beyond the processor: Exploring full-system energy-accuracy tradeoffs in a smart camera system," *IEEE Transactions on VLSI Systems*, vol. 26, no. 12, pp. 2884–2897, Dec 2018.