Harrison Williams – Research Statement

Web: https://harriswms.github.io/ **Email:** hrwill@vt.edu

With a market size of $600 billion in 2023—projected to grow to over $4 trillion by 2032 [1]—the Internet of Things (IoT) has been called a pillar of the fourth industrial revolution. Deeply-embedded intelligent edge systems are rapidly becoming pervasive in every field where technology interacts with the physical world, spanning massive-scale industrial and agricultural sensing deployments, deep-space satellites running sophisticated machine learning workloads, and tiny implantable devices for medical monitoring or drug delivery. Despite their potential to revolutionize a wide variety of disciplines, the most ambitious use cases—truly ubiquitous computing in the form of smart dust, decade-long autonomous sensing deployments, and effective yet non-intrusive healthcare devices—are still out of reach owing to a combination of size, performance, and efficiency limitations. Broadly, my research develops **cross-stack hardware and software systems to improve the efficiency, lifespan, and performance of these resource-constrained edge computing systems.**

Vision Today's embedded and mobile systems maximize performance under stringent constraints by closely designing around the capabilities and limitations of the technologies traditionally at their foundation (e.g., CMOS logic, SRAM and flash memory, Li-ion batteries). While this design approach—combined with decades of "free" downscaling of digital hardware as a result of Moore's law and Dennard scaling—has enabled a new class of high-performance yet low-power systems, it also means that *system design is tightly coupled to the constraints of these mature technologies*. The recent breakdown of these trends poses a crucial question for computer architects and system designers: how can we continue building ever-smaller and smarter edge systems to meet the demands of emerging applications as the scaling of our proven technologies slows? Innovations at individual layers of the system stack—such as new techniques for energy delivery and new memory technologies for data storage—offer a path forward, but require a fundamental shift in the way we design systems to better leverage these nascent technologies. I believe realizing the long-term goal of truly ubiquitous computing culminating in the form of "smart dust" requires **re-imagining system design around these emerging technologies at the hardware, software, and platform levels**.

Key Contributions I am principally a **computer architecture** and **systems** researcher; my doctoral work has led to five publications in top-tier venues for those communities [ASPLOS'20, ISCA'21, ASPLOS'24, ASPLOS'25, ATC'24] and one best paper award [ENSsys'24]. The cross-domain nature of my research has led to collaborations across the ECE, CS, and Aerospace departments at Virginia Tech and beyond to find new application areas and perspectives for my work and solve mutually interesting problems beyond my "home" communities [ISCAS'23]. Underlying my approach is a dedication to open sourcing research artifacts and active engagement with external groups to maximize the impact of my work. Beyond academia, I have supported engineers at funding organizations (DARPA, ONR) to integrate my research into their own deployed systems—bridging the gap between research and practice.

Doctoral Research

My doctoral research focuses on system support for energy harvesting devices, computational edge systems which forego batteries in favor of harvesting circuits and a short-term energy store (e.g., a capacitor). Batteryless systems bring intelligence and connectivity to applications where batteries are a non-starter for size or safety reasons, but also represent an extreme case of energy-constrained computing: an energy harvester may only be able to buffer enough energy to sustain several seconds of operation, forcing designers to closely monitor energy levels throughout execution, extend computation across frequent power failures, and meticulously schedule atomic tasks to avoid interruption. These challenges require a fundamentally new approach based on cross-layer co-design from the circuit and architectural foundations up to the software layer.

[1] Internet of Things Market Analysis: https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307

Hardware/Software Co-design for Batteryless Computing

Intermittent Computing Background Many batteryless systems operate in a perpetual energy deficit; even under ideal input power conditions, harvested energy is too weak to sustain continuous operation. Software on these devices executes *intermittently*: in short bursts of computation punctuated by frequent recharge cycles. Intermittent systems extend long-running computation across these power cycles by checkpointing volatile intermediate state to persistent Non-Volatile Memory (NVM) before a power loss and restoring it at the start of the subsequent on-period, allowing arbitrarily long programs to complete on intermittent power. Choosing when, how, and what to checkpoint is nontrivial and depends on software behavior, energy dynamics, and the limitations of the underlying hardware.

NVM-Invariant Intermittent Computation [ASPLOS'20] A major challenge for intermittent systems is that NVM write performance is on the critical path of program execution, but writing to the most common type of embedded NVM (flash memory) is prohibitively slow, energy-intensive, and endurance-limited—limiting intermittent execution to systems integrating emerging high-performance Non-Volatile RAMs (NVRAMs) based on ferroelectric or magnetic technologies. To enable intermittent operation on systems without high-performance NVM, my research in this area explores persisting program state directly in Static RAM (SRAM), the volatile memory of choice for low-power embedded systems. I developed the concept of *time-dependent non-volatility*, an emergent property of energy harvesting systems where remnant energy on the supply rail holds volatile SRAM above its data retention voltage for minutes to hours depending on capacitor size and temperature—rendering SRAM functionally non-volatile for far longer than the often seconds-long off times experienced by many intermittent systems. To take advantage of this effect, I built a software checkpointing system called TotalRecall [ASPLOS'20] which records program checkpoints "in-place" in SRAM, completely eschewing NVM writes and eliminating data movement overhead. TotalRecall treats SRAM as a "best-effort" NVM using integrity checks verifying that SRAM fully retained state during an off period, enabling for the first time low-overhead and long-lived intermittent execution on flash- and ROM-based devices.

Efficient State Rollback for Atomic Operations [ATC'24] TotalRecall demonstrated the potential for SRAM-based checkpointing, but required that user code be able to arbitrarily stop and restart at any point within the program—breaking applications containing uninterruptible atomic operations such as radio transmissions, which must be rolled back and re-executed if they are interrupted. To bring rollback to SRAM checkpointing systems, I developed a new programming and execution model called Camel [ATC'24] using the idea of volatile and non-volatile *worlds*—copies of global state with varying consistency requirements across power cycles. Programmers using the Camel model decompose their code into tasks, which operate directly on the volatile world and roll back to the task beginning using the non-volatile world when interrupted. Between tasks, code inserted by a custom LLVM compiler pass 1) atomically swaps the worlds to lock in progress from the preceding task and 2) copies the minimal set of updated state to the newly-volatile world required for the subsequent task to correctly execute. Camel's techniques to minimize memory consumption and data movement both enable efficient rollback on SRAM-based checkpointing systems and halve runtime overhead compared to prior work on NVRAM platforms, improving on the state of the art for systems integrating *any* kind of memory technology.

Lightweight, Integrated Energy Monitoring [ISCA'21] Checkpointing techniques like TotalRecall and other energy-aware intermittent systems depend on measurement circuits to track energy levels and make decisions about checkpointing and execution; unfortunately, the integrated components best suited for energy monitoring (analog-to-digital converters) are a poor fit for energy-constrained operation as they sacrifice power efficiency for signal processing fidelity not required for energy monitoring. As a result, many batteryless systems dissipate more power monitoring energy than they do executing useful work. To overcome this, I built a low-power integrated energy monitoring system called Failure Sentinels [ISCA'21]. The key insight behind Failure Sentinels is the high sensitivity of digital circuits to supply voltage: the propagation delay of logic gates varies with supply voltage, which correlates to remaining energy on a capacitor-powered system. Failure Sentinels measures the propagation delay of digital gates using a self-oscillating circuit composed entirely of standard logic cells

and forwards that information to software to convert propagation delay to stored energy. The digital-only design sacrifices unnecessary resolution and speed compared to a standard analog approach in exchange for reducing energy consumption to below just 1% of the digital core itself; my evaluation found that Failure Sentinels increased energy available for useful work by up to 75% by minimizing power consumption and providing ”just enough” functionality for intelligent batteryless operation.

Energy-Adaptive Buffering Architecture [ASPLOS'24] Batteryless devices buffer harvested energy in supercapacitors to power bursts of work isolated from the volatility of incoming power. The size of the capacitor is an important design factor: smaller capacitors charge to an operational voltage faster, but can only buffer a limited amount of energy; larger capacitors require more energy to enable the system but capture more energy during periods of high power input. Poorly fit capacitors cause energy loss in the form of waste heat and leakage, but current systems use a single capacitor chosen at design time—forcing a tradeoff between charge time and buffer capacity, and wasting energy when short-term power dynamics do not fit capacitor choice. My insight was that unpredictable power dynamics mean that buffer size must *vary* along with energy input; I then demonstrated how to close this trade space and eliminate these sources of waste using an adaptive, variable-size capacitor array called REACT [ASPLOS'24]. REACT consists of a set of capacitors and a switching fabric connecting them to a common rail in different configurations, presenting a low equivalent capacitance when stored energy is low (enabling rapid operation) and gradually increasing capacitance as buffered energy increases (avoiding waste heat while providing arbitrarily high capacity). A key challenge developing REACT was efficiently varying capacitance without wasting energy to leakage or charge movement between capacitors; to do so, REACT's design takes inspiration from a charge pump, a power electronics circuit used for voltage conversion that varies voltage and equivalent capacitance with near-zero loss. As a result, REACT both efficiently increases capacitance to capture incoming energy and *reclaims* energy for use from disconnected capacitors as it reduces capacity, eliminating both the capacity-charge time trade space and a major source of energy waste associated with single-capacitor systems.

Software Support for Emerging Technologies

My current research beyond batteryless devices focuses on software techniques to improve the performance of novel embedded memory technologies; these technologies have important qualities with the potential to transform embedded system design, but suffer important drawbacks that keep them from widespread adoption. I believe mitigating these disadvantages at the *system* level is a promising approach to render them truly practical and enable deployments which leverage their unique capabilities.

Mitigating Memory Underperformance in Software [ASPLOS'25] Non-Volatile RAM (NVRAM) technologies are foundational to many batteryless systems for checkpointing, but also show promise as a general-purpose memory owing to their density advantage over SRAM and lower write energy compared to flash memory. Unfortunately, NVRAMs are held back by low common-case software execution performance: with read performance inferior to flash and lower performance for any access compared to SRAM, existing NVRAMs are a poor replacement for *either* memory outside specific applications. Commercial NVRAM systems consume over twice as much power during active-mode execution compared to an otherwise identical flash-based device—discouraging developers from building applications using these memories. To eliminate this performance-based barrier to adoption, I introduced a new software-only execution model for NVRAM-based systems called SwapRAM [ASPLOS'25]. SwapRAM is based on the observation that NVRAM can be used to store both constant data and code and as main program memory; combined with the insight that *the majority of memory accesses in a register-based architecture are to instruction memory rather than main memory*, SwapRAM inverts the traditional embedded memory arrangement: code is cached in and executed primarily out of small, high-performance SRAM, while data resides in lower-performance NVRAM. The key challenge in building SwapRAM was implementing a performant instruction cache entirely in software. I developed a set of static code transformations and runtime systems based on observations about embedded software and the architecture of typical NVRAM-based systems. The end result is a programmer-transparent software system that brings the common-case performance of NVRAM-based systems up to those using more mature technologies.

Future Work

The overarching goal of my research is to maintain the pace of innovation and improvement by rethinking systems and architectures in light of the fundamentally new technologies and application areas they face today. Below, I describe several research arcs I plan to explore as I establish and grow my lab.

Short-Term Vision: Cross-Layer Techniques for Energy Efficiency

Energy is one of the largest limiting factors to the long-term goal of sophisticated smart-dust scale systems. The demise of traditional scaling laws means we can no longer expect substantial improvements simply from advances in CMOS technology; we must look elsewhere on the system stack for ways to improve efficiency. In the short term, I will expand on the practical software and hardware foundations developed in my doctoral work to overcome the energy bottleneck limiting today's edge systems.

Intelligent Energy Management on Battery-Powered Systems While batteryless energy harvesters are one solution to the large, and growing, performance gap between mobile systems and the batteries powering them, there is a large class of applications for which batteryless operation is a poor fit (e.g., safety-critical systems). My work on energy storage in batteryless systems [ASPLOS'24] shows how integrating software intelligence with power system design improves energy harvesters, but what about battery-powered systems? Several important factors for battery performance—including efficiency, lifespan, and effective capacity—depend highly on load behavior, but existing systems have few ways to modulate power draw without dramatically reducing performance or eliminating functionality. I plan to explore how combining intermittent computing techniques with hardware systems to manipulate power draw enables mobile systems to optimize battery behavior and enable smaller, cheaper, and longer-lived devices without sacrificing performance. In tandem with novel hardware systems, machine learning models have demonstrated promise in predicting system energy dynamics and battery degradation mechanics—I am exploring *online* applications of these models to similarly improve energy-constrained systems. This research arc is the basis of my most recently submitted NSF proposal.

Optimizing Memory Interfaces for Embedded Software Advances in low-power, high-speed microarchitectures mean that the memory wall plaguing general-purpose processors is now a problem for even the smallest, most resource-constrained systems. Fortunately, the constrained and predictable nature of embedded firmware (relative to software on a commodity system) means static analysis and online code profiling are promising techniques for predicting future memory access patterns—enabling sophisticated caching and instruction/data prefetching to overcome access penalties. I am extending my work in this area [ASPLOS'25] by 1) collaborating with language and compiler experts to improve performance by integrating control/data-flow analysis and transformations, and 2) adapting the techniques I have developed to different memory technologies (e.g., flash-based systems) and application areas (e.g., intermittent systems). In the longer term, I intend to explore how architectural-level modifications allow us to more efficiently integrate rich software-level information into cache systems.

Long-Term Vision: Extending the Reach of Next-Generation Technologies

Novel logic and memory technologies have demonstrated value in overcoming specific constraints of embedded IoT systems, but have promise in fields beyond this space. Realizing the potential of these technologies in new application areas will require device designers and system architects to consider an entirely different set of constraints; in the long term, I plan to explore how next-generation technologies and design approaches can transform computing *beyond* traditional IoT deployments.

Architecture and System Design for Extreme Environments A large body of work has explored how traditional CMOS logic and memory technologies can be adapted to harsh operating conditions such as high-radiation space deployments or low-temperature systems deep underwater. As these extreme-edge deployments mature, they will face increasing demand for next-generation technologies to improve performance in much the same way as "typical" systems. New technologies see different

tradeoffs and opportunities in extreme environments (e.g., several NVRAM technologies are naturally rad-hard but face increased error rate and power draw when cold, while subthreshold logic is extremely energy-efficient but particularly susceptible to radiation-induced errors), but there is little work on addressing these challenges at the architectural level. I plan to expand on my existing work mitigating the drawbacks of these technologies [ASPLOS'25] to determine how we can best take advantage of their unique capabilities and performance in this growing application area.

Rethinking Device and Architecture Design in Commodity-Class Systems While novel logic and memory technologies have demonstrated their value in the context of highly constrained embedded systems, they have several qualities that make them attractive for desktop-, server-, and datacenter-scale computing. New power management and low-power logic designs offer a path for high-performance processor designers to overcome the power wall, while novel memory technologies have significant density advantages over SRAM and power advantages over DRAM. However, system development at commodity computing scales requires a different set of priorities from embedded computing: extreme cost sensitivity, reliability concerns, and fundamentally different workload profiles all mean today's architectures and interfaces developed for edge computing systems are a poor fit for general-purpose computing. How can we adapt existing designs for next-generation logic and memory technologies to realize the advantages of the underlying hardware in commodity-class systems? I plan to collaborate with developers from the HPC communities to build systems incorporating a broad understanding of the needs of HPC systems with the device- and interface-level insights developed in my prior research arcs.

Funding

I plan to primarily seek support from the NSF's Software and Hardware Foundations (SHF) program, where I have experience soliciting funding (one funded proposal drafted as a student and one submitted proposal as a co-PI). A core tenet of my approach to research is developing *physical artifacts* demonstrating the concepts developed in lab: as such, I believe my work will also be of interest to industry- and product-oriented funding organizations (e.g., DARPA, ONR, NASA), giving me the resources and connections to both push the intellectual state of the art and build systems with real-world impact.

References

[ATC'24] Harrison Williams, Saim Ahmad, and Matthew Hicks. "A Difference World: High performance, NVM-invariant, Software-only Intermittent Computation". In: *USENIX Annual Technical Conference*. **ATC**. 2024.

[ASPLOS'25] Harrison Williams and Matthew Hicks. "A Software Caching Runtime for Embedded NVRAM Systems". In: *International Conference on Architectural Support for Programming Languages and Operating Systems*. **ASPLOS**. 2025.

[ENSsys'24] Harrison Williams and Matthew Hicks. "A Survey of Prototyping Platforms for Intermittent Computing Research". In: *International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*. **ENSsys**. 2024. **Best Paper Award**.

[ASPLOS'24] Harrison Williams and Matthew Hicks. "Energy-Adaptive Buffering for Efficient, Responsive, and Persistent Batteryless Systems". In: *International Conference on Architectural Support for Programming Languages and Operating Systems*. **ASPLOS**. 2024.

[ASPLOS'20] Harrison Williams, Xun Jian, and Matthew Hicks. "Forget Failure: Exploiting SRAM Data Remanence for Low-Overhead Intermittent Computation". In: *International Conference on Architectural Support for Programming Languages and Operating Systems*. **ASPLOS**. 2020.

[ISCA'21] Harrison Williams, Michael Moukarzel, and Matthew Hicks. "Failure Sentinels: Ubiquitous Just-in-Time Intermittent Computation via Low-Cost Hardware Support for Voltage Monitoring". In: *International Symposium on Computer Architecture*. **ISCA**. 2021.

[ISCAS'23] Fariborz Lohrabi Pour, Harrison Williams, Matthew Hicks, and Dong Sam Ha. "RF Energy Harvesting in Minimization of Age of Information with Updating Erasures". In: *IEEE International Symposium on Circuits and Systems*. **ISCAS**. 2023.