Failure Sentinels: Ubiquitous Just-in-time Intermittent Computation via Low-cost Hardware Support for Voltage Monitoring

Harrison Williams
Department of Computer Science
Virginia Tech
Blacksburg, USA
hrwill@vt.edu

Michael Moukarzel
Department of Computer Science
Virginia Tech
Blacksburg, USA
mamoukar@gmail.com

Matthew Hicks
Department of Computer Science
Virginia Tech
Blacksburg, USA
mdhicks2@vt.edu

***Abstract*—Energy harvesting systems support the deployment of low-power microcontrollers untethered by constant power sources or batteries, enabling long-lived deployments in a variety of applications previously limited by power or size constraints. However, the limitations of harvested energy mean that even the lowest-power microcontrollers operate intermittently—waiting for the harvester to slowly charge a buffer capacitor and rapidly discharging the capacitor to support a brief burst of computation. The challenges of the intermittent operation brought on by harvested energy drive a variety of hardware and software techniques that first enabled long-running computation, then focused on improving performance. Many of the most promising systems demand dynamic updates of available energy to inform checkpointing and mode decisions.**

Unfortunately, existing energy monitoring solutions based on analog circuits (e.g., analog-to-digital converters) are ill-matched for the task because their signal processing focus sacrifices power efficiency for increased performance—performance not required by current or future intermittent computation systems. This results in existing solutions consuming as much energy as the microcontroller, stealing energy from useful computation. To create a low-power energy monitoring solution that provides just enough performance for intermittent computation use cases, we design and implement *Failure Sentinels*, an on-chip, fully-digital energy monitor. *Failure Sentinels* leverages the predictable propagation delay response of digital logic gates to supply voltage fluctuations to measure available energy. Our design space exploration shows that *Failure Sentinels* provides 30–50mV of resolution at sample rates up to 10kHz, while consuming less than 2μA of current. Experiments show that *Failure Sentinels* increases the energy available for software computation by up to 77%, compared to current solutions. We also implement a RISC-V-based FPGA prototype that validates our design space exploration and shows the overheads of incorporating *Failure Sentinels* into a system-on-chip.

I. Introduction

Continuous advances in the design and manufacture of tiny, low-power computing devices have opened the door for microcontrollers in applications previously limited by size, power, or cost constraints. Today's microcontroller-based sensor motes are small enough to monitor cellular temperature [1] and cheap enough to be deployed in high volumes inside of groceries [2] or with consumer goods to secure supply chains [3]. These advances have also enabled the use of sensor motes in more extreme and inaccessible environments such as space [4], deep underwater [5], or even embedded in concrete [6]. The challenge for today's designers is to build systems that best leverage this rapid down-scaling of computing hardware.

One major hurdle for the widespread deployment of tiny computing platforms is batteries, which have not experienced the same level of continual scaling as transistors. A typical lithium battery measuring 1 cm^3 can supply a low-power microcontroller drawing 300 μW for less than 4 months [7], after which the device is useless without a battery replacement—which is at best costly and at worst infeasible. Batteries also carry a risk of fire or explosion, limiting their use in sensitive applications such as medical implants, space deployments, or aircraft. The limitations of batteries are driving work in a new direction: energy harvesting platforms, which replace the battery with a transducer to capture energy from the environment and a buffer capacitor to store the gathered energy until it is sufficient to power the on-board devices. The source and amount of available energy depends on the operating environment: many deployed systems are powered by RFID readers [2], [8], while other promising energy harvesters leverage thermal [9], photovoltaic [10] or piezoelectric [11] effects.

While energy harvesting opens up new opportunities for self-sufficient devices, it also poses challenges for the system designer. The unpredictable nature of harvested energy, low power output of transducer circuits, and energy buffer size limitations mean that a microcontroller running on harvested energy may maintain operation for a few hundred milliseconds—too short for many useful software applications to complete before power fails and program state is lost. Past work proposes a variety of techniques to support long-running program execution across numerous and frequent power failures, the most promising approach being *just-in-time checkpointing*: saving a snapshot of the program state to non-volatile memory when power failure is imminent [12]–[16].

Unfortunately, the requirement of a voltage monitor limits just-in-time checkpointing approaches. To know when power failure is imminent, just-in-time approaches track the voltage across the buffer capacitor (voltage is a surrogate for energy) using an Analog-to-Digital Converter (ADC) [12] to measured the voltage then comparing the measurement to a user-defined voltage threshold. However, ADCs are among the most power-intensive peripherals available on modern low-power microcontrollers: integrated ADCs typically consume *as much or more power than the processor core itself* (see Table I), reducing useful computation time by over 50% even before considering the software overhead introduced by checkpoints. Recent just-in-time approaches replace ADCs with their lighter weight cousins, the analog voltage comparator [13]–[15], [17]. This decision trades lower voltage resolution for marginally decreased current consumption. No matter the mechanism to monitor voltage, recent advancements in just-in-time checkpointing have exposed the voltage monitor as the primary source of run-time overhead—over an order of magnitude more than checkpointing [16].

The key issue is that ADCs and analog comparators are not optimized to support intermittent computation. For intermittent computation use cases, energy efficiency is paramount as long as resolution and sample rate are sufficient. Existing voltage monitors have been optimized in the opposite direction: performance first, then energy. The key to unlocking the promise of just-in-time approaches is a low-power, all-digital, on-chip supply-voltage monitor with just enough resolution and sample rate to meet the demands of current and future intermittent computation use cases. With such a voltage monitor, it is practical to make energy availability a first-class abstraction provided by the hardware, improving existing intermittent computation systems (Section II-C) and enabling future ones.

In order to enable efficient voltage monitoring on energy harvesting systems, we develop *Failure Sentinels*—a low power, all digital, reference-free, on-chip voltage monitor designed to scale with the rest of the system. *Failure Sentinels* leverages the predictable gate-delay response of digital circuits to a changing supply voltage to inform software decisions about available energy; *Failure Sentinels* works by counting the number of times a signal traverses a self-oscillating feedback loop during a fixed time period as a reference-free indicator of buffer capacitor voltage, which itself indicates available energy. We design *Failure Sentinels* using only CMOS (Complementary Metal-Oxide Semiconductor, the technology of choice for digital integrated circuits) logic, ensuring that it scales along with the rest of the system's digital logic and take advantage of the corresponding price, power, and size benefits. *Failure Sentinels* exposes a broad design space to system designers to allow them to tune a variety of performance parameters such as resolution and sample rate to find the balance of performance and energy consumption that is just-right for their use case.

We implement and evaluate *Failure Sentinels* in SPICE and on a RISC-V-based system-on-chip running on an Artix-7 Field-Programmable Gate Array (FPGA) [18]. We use the SPICE implementation to explore *Failure Sentinels*'s trade space across process nodes and voltages. We use the FPGA implementation to validate the SPICE results and to provide a real-world demonstration of *Failure Sentinels*'s performance. Finally, we evaluate *Failure Sentinels* against analog alternatives on energy harvesting power traces to explore the system-level impact. Our evaluation indicates that *Failure Sentinels* reduces runtime overhead by 24%–70% compared to existing solutions, provides a flexible and scalable design space, and enables a variety of system designs previously limited by voltage monitor options.

This paper makes the following three contributions:

- We evaluate existing voltage monitors and identify the power and scalability as the primary hurdles for their use in current and future intermittent computation systems (Section II-B).
- We design *Failure Sentinels*, an on-chip voltage monitor (Section III). *Failure Sentinels* leverages the power and space scaling of wholly-digital circuits and enables designers to build in *just enough* resolution and sample rate to meet at near-zero additional power and area (Section IV-B).
- We build *Failure Sentinels* in simulation and on FPGA hardware to explore the power/accuracy trade space that different *Failure Sentinels* implementations expose to designers (Section IV). Our results show that *Failure Sentinels* improves intermittent system performance by up to 77% by eliminating a major source of power consumption, freeing up energy for useful computation (Section V).

II. Background and Related Work

Although the size and power consumption of modern devices continue to decrease, harvested energy is typically too weak and unreliable to guarantee enough power to continuously support current microcontrollers [10], [12], [15], [16]. Instead, energy harvesting circuits slowly feed power into a buffer capacitor until enough energy is available to support a short burst of computation. Once computation starts, the microcontroller and peripherals rapidly drain the capacitor until the system reaches the minimum operating voltage, and the charge-discharge cycle repeats. The limitations of the harvesting circuit mean that devices running on harvested energy can restart dozens of times per second [12], [19]. Given that programs and programmers alike are not prepared for such operating conditions, previous work proposes a variety of strategies to stretch long-running computation across frequent power cycles, referred to as intermittent computation.

A. Supporting Intermittent Computation

Most current systems to support software on intermittently-powered platforms fall broadly into one of two categories; while each commits some portion of volatile memory (typically architectural registers, main memory, and any peripheral registers) to non-volatile memory, they can do so *just-in-time* before power failure [12]–[17] by measuring available energy or *continuously* [20]–[24] throughout execution. The choice of checkpointing strategy is the primary determinant of system performance. Just-in-time systems theoretically maximize

Platform	MSP430FR5969 [25]	PIC16LF15386 [26]
Core I_{in} (μA/MHz)	110	90
ADC I_{in} (μA)	265	295
Comp. I_{in} (μA)	35	75
Core V_{min} (V)	1.8	1.8
Ref. V_{min} (V)	1.8	2.5

TABLE I: Core versus ADC/comparator power requirements of sensor-mote-class microcontrollers, including voltage reference draw.

performance by only recording one checkpoint per power cycle and simplify software's interface as existing software is supported by linking against a library-level interrupt handler, but they depend on a voltage monitor attached to the buffer capacitor that interrupts computation to store a checkpoint when voltage falls below a threshold value (indicating imminent power loss). Unfortunately, practical considerations limit the applicability and performance of just-in-time approaches as existing voltage monitoring solutions are ill-suited for the voltage monitoring use case. The key to unlocking the promise of just-in-time approaches is a low power, scalable, on-chip supply-voltage monitor with just enough resolution and sample rate.

B. Monitoring Supply Voltage

Modern low-power microcontrollers include two components suitable for supply voltage monitoring: an Analog-to-Digital Converter (ADC) and an analog comparator. Unfortunately, the signal-processing focus on resolution and sample rates driving ADC design makes them unsuitable for supporting intermittent computation because of their relatively high power consumption: Table I shows that each component (and supporting circuitry)[1] requires current on-par with the processor itself.[2] This means that over half of the energy harvested is wasted on checking for imminent power failure—as opposed to computation. The wasted energy will only increase for future systems due to the discrepancy in scaling between digital logic and ADCs: performance/Watt for processors tends to scale at 2x every 1.57 years [30], while performance/Watt scales at 2x approximately every 2.6 years [31].

To avoid the power-hungry nature of full-fledged ADCs, recent intermittent computation systems employ single-bit analog comparators [13], [14], [17]. While single-bit analog comparators improve on ADCs, they still waste 21%–45% of harvested energy on reference voltage generation. Single-bit solutions also limit utility as many current and emerging intermittent computation systems demand dynamic, fine-grain, and poll-able voltage monitoring; essentially, the ideal solution is making available energy a first-class abstraction provided by the hardware to software at near-zero energy cost.

[1]Both components require a reference voltage to compare against the measured signal, typically provided by a diode [27] or internal bandgap reference generator [25], [28].

[2]While discrete low-power ADCs exist [29], their cost is on par with the microcontroller itself and their standalone nature adds size and complexity to the system.

C. Enabling Future Intermittent Systems with Practical Voltage Monitoring

The first systems to address intermittent computing on small batteryless devices focus on *enabling* long-running intermittent programs [12], [16], [20], [21], [23]; more recent ones focus on *optimizing* it [14], [24], [32], [33]. Since checkpoints are one of the drivers of run time overhead (on-par with voltage monitoring), one way to improve performance is to eliminate superfluous checkpoints. Chinchilla [32] is a timer-augmented continuous checkpointing system that improves performance through energy-guided checkpointing. Chinchilla dynamically tunes a timer to the expected on time and skips checkpoints that occur before the timer expires. Despite the challenges of representing energy in a timer value, Chinchilla yields a 2x–4x performance boost over similar checkpointing systems. Chinchilla must be overly pessimistic on available energy and energy usage to maintain correctness. With a practical voltage monitoring solution, Chinchilla is able dynamically query available energy and remove its guard bands; this increases performance, while also increasing system reliability.

Work beyond checkpointing presents a variety of energy-efficient techniques tailored for energy harvesting. PHASE [34] makes the case for single-workload heterogeneous architectures, switching between high-performance and high-efficiency systems for the same workload depending on the availability of ambient power. HarvOS [35] profiles the energy requirements of each section of code and schedules software execution or non-volatile checkpoints accordingly. Dewdrop [19] similarly balances task execution and sleeping depending on available energy to make the most of changing ambient energy conditions. These systems all promise significant performance boosts but depend principally on *low cost, on-demand measurements of remaining energy*. While ADCs can fulfill this role, their high resolution and high sample rate are overkill and steal up to 50% of energy from software. The goal of this paper is to *enable these and other technologies* through a low power, all digital (i.e., scalable), on-chip voltage monitor that provides just enough resolution and sample rate; in doing so, we make energy availability a first-class hardware abstraction.

III. *Failure Sentinels* DESIGN

We design *Failure Sentinels*: a low power, fully digital, software-programmable voltage monitor optimized to minimize power consumption while meeting the resolution and sample rate needs of current and future intermittent computation systems. Two intermittent-computation-focused design goals drive our approach:

- ***Failure Sentinels* must minimize power consumption and provide *just enough resolution and sample rate* to serve software's needs.** Section II shows that existing voltage monitors fail at this, whereas *Failure Sentinels* provides enough resolution to support frequent, voltage measurements, while avoiding the design and power concerns associated with DSP-focused ADCs.

- ***Failure Sentinels* must be *compact and fully-digital* to enable ubiquity and scalability:** As energy harvesters find their way into micro-healthcare and smart dust applications, miniaturization of every component in the system is a primary concern. Implementing *Failure Sentinels* with solely the CMOS gates used for digital logic ensures that it can be incorporated into any device and scales with process technology.

A. Ring Oscillators

Digital systems operate correctly at a wide range of voltages using well-chosen clock frequency guard bands, which hide the extreme sensitivity of the underlying circuits to voltage changes. Removing these guard bands from an otherwise digital circuit reveals analog-domain latency changes, which in turn reveal the system voltage. Desktop-class systems use this effect by measuring the propagation of a signal through lines of digital delay elements to support dynamic voltage and frequency scaling [36]–[38], but their narrow voltage range—beyond which the input either propagates entirely or not at all—makes them ill-matched for the voltage monitoring required by intermittent systems. Feeding the output of the delay line into the input such that that the output changes each time it passes through the entire delay line (i.e., it is self-oscillating) forms a Ring Oscillator (RO) with an output frequency that is primarily a function of supply voltage and a dynamic range covering nearly the entire voltage at which the RO oscillates. **This paper leverages the voltage-dependent nature of RO frequency to measure supply voltage**.

The RO is a common circuit with applications in clock generation [39], process tuning and characterization [40]–[42], and performance monitoring [43]. Ring oscillators are attractive options for these applications for their ease of integration into IC designs, low power consumption, and electrical tunability [44]. The basic RO structure is an odd-numbered ring of digital inverters as shown at the bottom of Figure 2. Because the output of an odd-numbered chain of inverters is the inverse of the input, feeding the chain output back to the input produces a circuit that oscillates as long as power is applied. RO length is largely application-dependent, but is typically prime to reduce potential harmonic oscillations [45].

The frequency of oscillation depends on the length n of the chain and the gate delay of each inverter τ_d as shown in Equation 1 [44].

$$f_o = \frac{1}{2n\tau_d} \tag{1}$$

With a constant chain length, the RO output frequency is entirely dependent on average gate delay. Several factors affect the gate delay: the designer tunes gate delay by changing the transistor size or supply voltage, while temperature and manufacturing variations also play a role. Among these, voltage is the dominant factor [46], [47].[3]

[3]Ring oscillators also have potential as fully-fledged ADCs [48], [49], but the signal-processing focus of these designs precludes them for efficient supply monitoring. However, research in this area to linearize the RO frequency-voltage curve and reduce sensitivity to process/temperature variations [50], [51] has potential to improve *Failure Sentinels*.

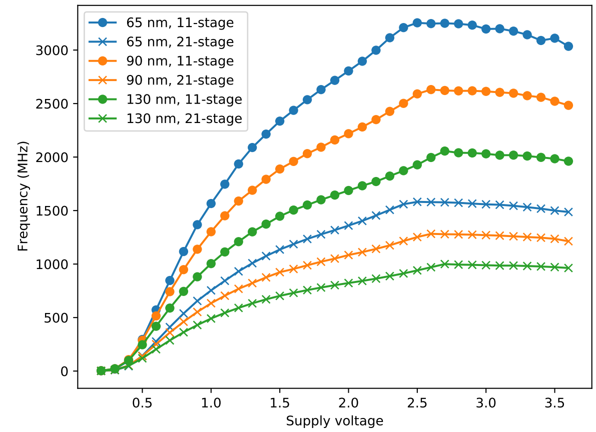

Fig. 1: RO frequency vs. supply voltage at different feature sizes.

B. Voltage-Frequency Relationship

To characterize supply voltage's effect on RO frequency, we run a comprehensive set of SPICE simulations on ROs of varying length, operating at a range of supply voltages using the Predictive Technology Models [52] for the 130nm, 90nm, and 65nm technology nodes. We choose these feature sizes because they are representative of the technology currently used on energy harvesting platforms [53] as well as the logical next feature size for future systems. We sweep the supply voltages from 0.2 V (below which the rings do not oscillate) in 100 mV steps up to 3.6 V, the maximum supply voltage for typical energy-harvesting-class devices [25]–[27].

Figure 1 illustrates the results of these simulations using 11- and 21-stage ROs in each technology. We use these results to make three key observations motivating and informing the design of *Failure Sentinels*:

- The high sensitivity of frequency to voltage makes ROs viable supply voltage sensor, and the sensitivity increase from moving to smaller processes means that *Failure Sentinels* improves as technology scales.
- Decreasing RO chain length magnifies the effects of supply voltage changes, increasing sensitivity.[4]
- Regardless of RO length or feature size, the output frequency becomes less sensitive as supply voltage increases, eventually decreasing at higher supply voltages. The RO must operate in the low-voltage, high-sensitivity region to reduce error.

C. System Overview

Figure 2 shows the high-level organization of *Failure Sentinels*. The voltage divider sets the operating range for the RO, allowing us to tune the RO to operate in the most-sensitive voltage region. The level shifter makes the output signal from the RO compatible with the voltage level used by digital logic,

[4]Decreasing RO length also increases the frequency and therefore current consumption of supporting circuitry; we explore the tradeoffs of different RO lengths in more detail in Section V-A.

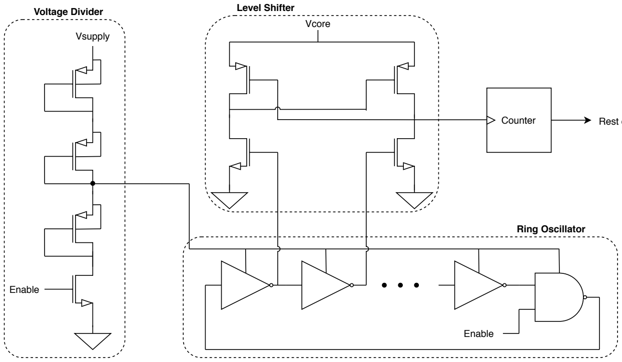

Fig. 2: *Failure Sentinels* block diagram with a divide-by-3 voltage divider. Not shown is a level shifter for interfacing the enable signal to the RO.

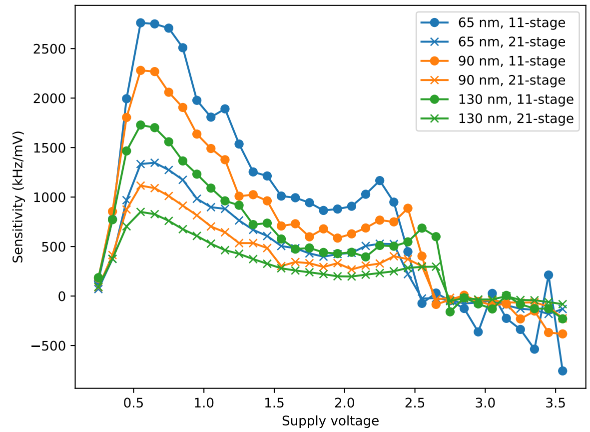

Fig. 3: Frequency-voltage sensitivity for ROs across length and technology.

reducing power consumption and ensuring reliable operation. The enable signal drives both an input to the NAND gate closing the RO loop and an N-type MOS device (NMOS) at the bottom of the voltage divider, allowing the designer to change the duty cycle of the RO, reducing dynamic power consumption. Breaking the RO chain with an enable sets each gate to a known state before it begins oscillating to prevent higher harmonic output frequencies [45]. Finally, the counter makes the output of the RO available to the rest of the system in the form of an edge count accumulated during the sampling period. Software maps the resulting counter values to supply voltage values using enrollment data stored in the NVM.

D. Choosing RO Length

Per Equation 1, *Failure Sentinels*'s voltage sensitivity scales proportionally to $1/n$ where n is the length of the RO—a given change in supply voltage produces a corresponding frequency change that is larger in shorter ring oscillators because a smaller n magnifies the impact of a change in the gate delay τ_d on the oscillation frequency. For ROs implemented in the same technology a given voltage change produces the same *proportional* frequency change regardless of the number of RO stages, but *Failure Sentinels* measures the *absolute* change in frequency. A higher change in frequency requires a shorter enable period to detect; a shorter enable period allows *Failure Sentinels* to run either at a lower duty cycle (consuming less power, because the ring spends less time enabled) or at a higher sampling rate. We distinguish between the *enable period*—the amount of time the RO is powered to produce a single sample—and the *sample period*—the time between distinct samples—and discuss their impact on *Failure Sentinels* in more detail in Section III-E.

Note that the dynamic power consumed by an RO is not dependent on its length, as only one inverter is ever switching at a time.[5] Increasing the size of the RO increases area overhead and static power; however, our evaluation in Section IV-B shows that *Failure Sentinels* consumes negligible power and area compared to the rest of the microcontroller. However, an RO that oscillates too fast for a given sampling period will overflow the counter. Thus, the counter bit-width, sampling period, and RO length are interconnected, a design space that we explore in Section V. From these constraints, we analyze the RO length primarily to the extent that it affects accuracy and power draw by setting a minimum duty cycle.

[5] *Failure Sentinels*'s total dynamic power is weakly dependent on RO size because the counter and level shifter power draw increase with frequency. However, the RO consumes the majority of *Failure Sentinels*'s power (Section V-A).

E. Duty Cycling

The accuracy of *Failure Sentinels* depends largely on the sampling rate and duty cycle $D = T_{en}/T_{sample} \leq 1$, where T_{en} is the time per sample during which *Failure Sentinels* is enabled and T_{sample} is the sampling period. A higher T_{en} enables *Failure Sentinels* to discriminate between finer RO frequency, and thus voltage, changes. The output of *Failure Sentinels* is in the form of the count $C = f_{ro} * T_{en}$; the edge-sensitive nature of the counter means that decimal values of C are effectively truncated. Therefore, the minimum detectable RO frequency change is $1/T_{en}$. The bit-width n of the counter limits the maximum value of C to $2^n - 1$; all possible values of $f_{ro} * T_{en}$ must be below this maximum to prevent counter overflow. Increasing T_{en} increases both accuracy and power consumption, which scales directly with duty cycle: given that low-resolution and low-frequency (relative to ADCs) measurements of the supply voltage are sufficient for current and near-future energy harvesters, operating *Failure Sentinels* with a low duty cycle enables significant power savings at little practical cost. A sufficiently low duty cycle also reduces counter size and its power. We evaluate the relationship between duty cycle, power, and accuracy in more detail in Section V-A.

F. Maximizing Voltage Sensitivity

a) Inverter cell choice: Past work provides a variety of options for the design of the inverter used to build the RO. Most ROs used in communications, clock generation, and other applications are current-starved [54]: the charge/discharge time of each inverter is limited by a voltage-controlled current source using a separate variable biasing voltage. An important property of the current-starved RO for these applications is that the current source *isolates the inverter from supply voltage noise*, minimizing uncontrollable variation and enabling the designer to produce a frequency output that is primarily a function of only the bias voltage. The crucial difference in *Failure Sentinels* is that the change in the supply voltage is the quantity of interest. Instead, we maximize sensitivity to changes in the supply voltage by using the simplest inverter available consisting of single PMOS and NMOS transistors connected directly to the supply voltage and ground, respectively. This basic inverter design has additional benefits, as it reduces the total transistor count and is implementable using digital-only standard cell libraries.

b) RO operating voltage: Figure 1 shows that the frequency-voltage curve for each RO is steepest at lower voltages, leveling off around 2.5 V and decreasing at higher voltages. The recommended operating voltage for microcontrollers used in recent energy harvesting work is 1.8V–3.6V [25], [26], [55]; for these platforms, connecting the ROs directly to the supply voltage means that they would operate primarily in the less-sensitive region. Furthermore, the voltage-frequency relationship at high voltages is non-monotonic—complicating the mapping in software from RO frequency back to supply voltage. To maximize *Failure Sentinels*'s sensitivity to supply voltage changes and keep the voltage-frequency relationship monotonic, the RO operates at a reduced voltage produced by the transistor-based voltage divider shown in Figure 2. This has the added benefit of reducing power consumption. The trade-off is that reducing the RO operating voltage adds complexity to the design because the output must be integrated back into the digital system, which Section III-G explores in detail.

Assuming a standard n-well process, which exposes the bulk connection of PMOS transistors,[6] the voltage divider consists of diode-connected PMOS devices with the bulk terminal connected to the source to ensure that each device is biased identically even as the gate voltage with respect to ground of successive transistors drops. V_{gs} for each individual transistor is small, limiting the current draw of the divider. This design parallels a resistive voltage divider, but the use of transistors makes it applicable to wholly-digital ICs. The drawback of the transistor version is that they become non-linear at extremely low and high voltages, but these voltages are well beyond the specified operational voltage range of microcontrollers.

The RO draws power from a node n PMOS transistors away from ground; in a divider consisting of m diode-connected devices, the RO supply voltage is $V_{ro} = V_{supply} * n/m$. The best voltage division ratio depends on the sensitivity curve of the RO, shown in Figure 3 for several RO lengths and technologies. Reducing the voltage seen by the RO tends to increase sensitivity; however, it also reduces the voltage change seen by the RO for a corresponding change at the supply rail. We define the *sensitivity gain* G using Equation 2.

$$G = \frac{\overline{S_{new}}}{\overline{S_{old}}} * \frac{n}{m} \tag{2}$$

The $\overline{S_{new}}$ and $\overline{S_{old}}$ terms reflect the average sensitivity in the new and old operating regions, respectively. The best division ratio is the one that maximizes G and is technology-dependent. We find that the best ratios implementable in a small number of transistors are $n/m = 1/3$ or $1/2$; each of these division ratios produces a sensitivity gain of $G \approx 2$. Between division ratios that produce the same sensitivity gain for a given process, the smallest one reduces power consumption by reducing the operating voltage of the RO. Thus, we select $n/m = 1/3$.

Assuming the transistors are well-matched, the unloaded output of the voltage divider is a reliable fractional value of the supply voltage. However, enabling the RO to draw power from the voltage divider reduces the effective resistance between the divider output and ground—resulting in a voltage drop and a V_{ro} below the nominal value. We compensate for this voltage drop by increasing the width of certain transistors. We widen the transistors between the voltage divider output and V_{supply} to increase current delivered to the RO and reduce the magnitude of the voltage drop. Appropriate transistor sizing reduces, but does not eliminate, the voltage drop seen at V_{ro} because the proportional error depends on the value of V_{supply}. However, the enrollment process described in Section III-H accounts for any remaining error, because the voltage offset is predictable at each supply voltage.

G. Logic Interfacing

Digital CMOS gates depend on well-defined input signals to achieve high speed and low power: an input must either be close to the supply voltage or close to ground to fully and rapidly switch the component transistors. Operating the RO at a fraction of the system's supply voltage increases sensitivity and decreases power consumption, but means that applying the logical 1 output of the RO directly to the counter input (operating at the normal supply voltage) violates this fundamental assumption of digital CMOS logic. The low-voltage RO logical 1 at best leaves little margin for noise and at worst is consistently below the core's logical 1 level, producing a signal that is unrecognizable to the core. Even if the RO output is reliably interpreted as a logical 1 by the core, driving CMOS gates with a low-voltage 1 increases power consumption due to ohmic losses from partially-on transistors and current in the low-impedance path to ground. We resolve the voltage difference using the level shifter shown in Figure 2, a self-reinforcing circuit leveraging the common ground of both voltage domains to boost the RO output voltage to the core voltage.

Ultimately, software needs to measure the frequency of the RO to make decisions based on supply voltage. We measure the output of the RO using a digital counter configured as shown in Figure 2 to increment on every positive edge of the level shifter output. The measurement is sent to a digital comparator for interrupt generation and made available to software

[6] In a p-well process, the voltage divider consists of NMOS devices and works equally well.

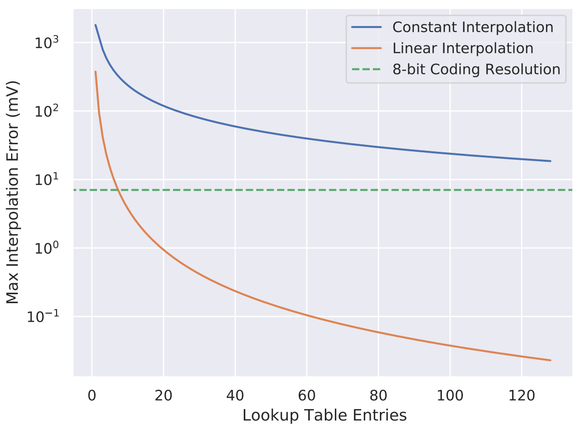

Fig. 4: Maximum interpolation error for a 21-stage RO in 130nm. The dashed line indicates minimum error possible using 8-bit calibration table entries.

by the addition of an instruction to the microcontroller's ISA, making energy availability a first-class hardware abstraction.

H. Voltage-Frequency Memoization

The counter maps RO frequency to a count value; the final step is mapping the count value to supply voltage. While the slope of the frequency-voltage relationship is predictable across all ROs, manufacturing-time process variation mean that identical ROs on different chips produce different frequencies under the same conditions. Microcontroller manufacturers already address process variation in sensitive circuits such as clock oscillators and sensors [26], [27], [55] using a post-manufacture enrollment step, testing the device with known inputs and writing device-specific calibration data to the Flash/ROM before deployment. We extend this enrollment process to increase *Failure Sentinels*'s precision by recording the RO frequency using several known supply voltages. [7] Once deployed, software uses these calibration values to determine supply voltage with reduced error.

The choice of both what and how much data to store is important. In general, designers can increase run-time performance by increasing memory consumption and enrollment effort. We identify and evaluate several enrollment strategies that occupy different points in that trade space:

- **Full enrollment:** A simple but impractical solution is to store a voltage value for every possible *Failure Sentinels* output; this maximizes accuracy (the voltage-count curve is fully characterized and stored) and speed (mapping a count to a voltage is a simple indexing operation). However, it also maximizes memory overhead and enrollment effort for each device.
- **Piecewise-constant interpolation:** Instead of storing every possible counter output, we can trade accuracy for memory overhead by reducing the number of data points stored in NVM. When the counter produces a value not stored in the lookup table, *Failure Sentinels* pessimistically assumes the supply voltage is at whatever level is associated with the closest stored count value below the measured value. Designers can tune *Failure Sentinels*'s accuracy by changing the number of stored data points, while a runtime count-voltage conversion in this case is slightly slower than with a full table (requiring a comparison followed by indexing).
- **Piecewise-linear interpolation:** Piecewise-linear interpolation enables the same accuracy-memory tradeoff as the piecewise-constant design but instead calculates a linear interpolation between the nearest two points when a count value is not stored. This increases accuracy for the same memory footprint at the cost of increased runtime overhead evaluating the interpolation function.
- **Polynomial interpolation:** To minimize memory overhead, the enrollment system can characterize *Failure Sentinels* at a few supply voltage points and place coefficients for an arbitrary-degree polynomial regression function in the device's memory. This makes space overhead negligible at the cost of runtime performance—evaluating the polynomial function requires numerous floating-point multiplication operations, which can be both time- and energy-intensive on typical energy harvesting hardware.

We explore the piecewise-constant and piecewise-linear interpolation designs in more detail because they are the most flexible and best suited to the performance and NVM constraints of current energy harvesters. For a continuous function $f(x)$ with lower and upper bounds a and b, respectively, Equations 3 and 4 describe the respective maximum error for piecewise-linear and piecewise-constant interpolation [56].

$$E_{const} \leq h * \max_{x\in[a,b]} \left| \frac{df(x)}{dx} \right| \tag{3}$$

$$E_{lin} \leq \frac{h^2}{8} * \max_{x\in[a,b]} \left| \frac{d^2 f(x)}{dx^2} \right| \tag{4}$$

$f(x)$ is the mapping from frequency to voltage for a given RO, the inverse of the relationship shown in Figure 1. h is the distance between known frequency datapoints and decreases with higher NVM consumption; for the frequency-voltage transfer function with minimum frequency L, maximum frequency H, and c evenly-spaced datapoints[8], $h = (H - L)/c$.

Figure 4 shows the maximum error introduced by both types of interpolation as a function of NVM overhead, assuming that each voltage entry in the table is stored in a single byte. By operating the RO at a low voltage using the divider described in Section III-F, we maximize the linearity of the voltage-frequency transfer function and enable highly accurate interpolation with a relatively small NVM footprint. Linear interpolation scales better than constant interpolation

[7] On devices with ADCs, an alternative to manufacture-time enrollment is a one-time characterization of the RO frequency-supply voltage relationship using the ADC for ground truth.

[8] One way to increase interpolation accuracy is to locally reduce h by taking more data points in areas where $\left|\frac{df(x)}{dx}\right|$ or $\left|\frac{d^2f(x)}{dx^2}\right|$ are highest, but for simplicity we use evenly spaced points.

with increasing NVM overhead, but both eventually achieve diminishing returns as other sources of error such as temperature begin to dominate *Failure Sentinels*'s total error. The precision of the recorded data points also limits interpolation accuracy, as shown in Figure 4: assuming a 1.8 V supply range, interpolating between 8-bit values cannot reduce the total error below $\frac{1.8V}{2^8} \approx 7mV$.

IV. *Failure Sentinels* IMPLEMENTATION

We evaluate *Failure Sentinels* using two implementations, each targeting different aspects of the design: (1) a SPICE implementation and (2) a FPGA implementation. We use SPICE to drive our design space exploration and evaluate the effects of supply voltage, feature size, and the analog circuit components on *Failure Sentinels*'s performance. To explore the effects of run time variation such as temperature and to demonstrate *Failure Sentinels* on real hardware, we integrate *Failure Sentinels* into a RISC-V processor running on a FPGA.

A. *SPICE Modeling*

We model *Failure Sentinels* using LTspice [57] to explore its behavior at a wide variety of supply voltages across different feature sizes. This enables us to practically explore *Failure Sentinels*'s design space. To match deployed and near-future real-world energy harvesting microcontrollers, we implement each RO using the 130nm, 90nm, and 65nm process Predictive Technology Model (PTM) SPICE cards [52]. We also include the provided parasitic resistance and capacitance estimates for local interconnects in those technologies between components. These SPICE simulations also offer insight into the effects of the analog circuitry (the voltage divider and level shifters), which is not available on the FPGA. Finally, SPICE includes power consumption information for each component of the design—enabling a direct comparison between *Failure Sentinels* and currently available alternatives such as ADCs.

B. *FPGA Implementation*

While the SPICE PTM models make it possible to perform a design-space exploration across process nodes and voltages, they do not accurately model the effects of thermal variation [58] and do not capture the ability to incorporate *Failure Sentinels* into a full system. To validate the SPICE-based design space exploration, understand temperature's impact on *Failure Sentinels*, and to show how architects can add *Failure Sentinels* to an existing System-on-Chip (SoC) to make energy availability a first-class hardware abstraction, we implement *Failure Sentinels* inside a RISC-V RocketChip SoC [59] on top of a Xilinx Artix-7 FPGA [60]. On top of this SoC, we run software that communicates with *Failure Sentinels* via two instructions added to the ISA: (1) an instruction that stores a 64-bit value representing the available energy to a user-specified destination register and (2) an instruction that the library-level recovery routine uses to enable *Failure Sentinels* as well as set the energy interrupt threshold. Similar to previous work that requires ADC support [16], we link unmodified software against a library-level interrupt handler that saves software state as a checkpoint when *Failure Sentinels*'s interrupt fires.

	area (LUTs)	timing (MHz)	power (W)
Base SoC	53664	30	1.105
+*Failure Sentinels*	53687 (+0.04%)	30 (+0.0%)	1.104 (-.09%)

TABLE II: *Failure Sentinels* hardware overheads when added to a RISC-V SoC [59]. Note that power is within the noise margin of the tools.

Design			**Performance**		
Parameter	Min.	Max.	Parameter	Min.	Max.
RO Length	3	73	Mean Current (μA)	0	5
F_s (kHz)	1	10	F_s (kHz)	1	10
Counter Size (bits)	1	16	Granularity (mV)	0	50
Enable Time	1 μs	1 ms	NVM Overhead (B)	0	128
NVM Entries	1	128	Transistor Count	0	1000
Entry Size (bits)	1	16			

TABLE III: *Failure Sentinels* design and performance parameters bounding our exploration.

Following the design goals of the ideal voltage monitor, Table II shows that adding *Failure Sentinels* to an existing SoC is low cost: *Failure Sentinels* maintains the maximum frequency of the SoC, minimally increases area, and, in accordance with the SPICE evaluation (Section V-A), steals very little energy from software computation. The implemented variant of *Failure Sentinels* has a 21-stage RO and an 8-bit counter. The fixed nature of the FPGA fabric precludes implementing the transistor-based voltage divider and level shifter, but those minimally contribute to *Failure Sentinels* area and removing them actually increases *Failure Sentinels*'s power.

V. EVALUATION

We take a two-level approach in evaluating *Failure Sentinels* in order to support comparison to currently available alternatives. We first perform a comprehensive evaluation of the *Failure Sentinels* design space using SPICE simulations to explore the trade space between different design parameters and their effect on *Failure Sentinels*. Then we demonstrate *Failure Sentinels* on real hardware and evaluate the impact of thermal variation by implementing *Failure Sentinels* as part of a RISC-V System-on-Chip using a FPGA. The results of this evaluation answer the following questions:

1) How does building *Failure Sentinels* to satisfy certain design constraints impact its performance in other areas?
2) How do typical sources of run-time variation such as temperature affect *Failure Sentinels*?
3) How well does *Failure Sentinels* fit the needs of energy harvesting applications compared to existing alternatives?

A. *Failure Sentinels Design Space*

Failure Sentinels is designed for flexibility; each application places unique demands on the system in terms of power consumption, resolution, and other parameters. Rather than hand-design and evaluate one *Failure Sentinels* implementation for a given deployment, we explore the *Failure Sentinels* design space by finding a set of Pareto-optimal implementations

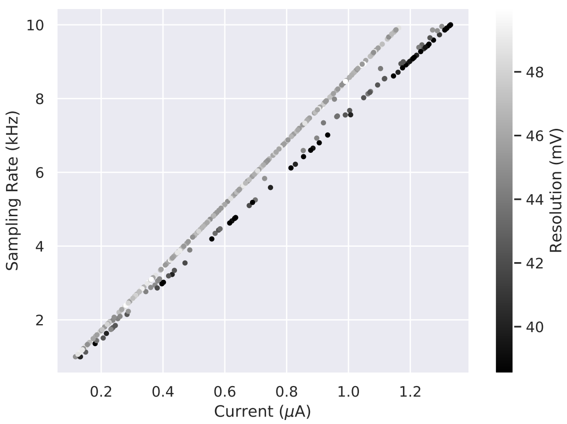

Fig. 5: Objective space exploration for *Failure Sentinels* in 90nm.

within performance constraints suitable for a range of energy-harvester deployments. We model *Failure Sentinels* design as an optimization problem mapping six design parameters to five performance parameters as shown in Table III. We set the design parameter bounds to ease integration with today's energy harvesters—for example, we limit the counter size to 16 bits to improve performance on the 16-bit architectures common to currently deployed energy harvesters [25]. Similarly, the 1 μs minimum enable-time stems from the minimum period of the fastest (1 MHz) clock available on similar systems without increasing current consumption [25].

We explore the resulting design space using Pymoo [61], a Python optimization library. We implement *Failure Sentinels* in LTspice [57] at several design points spanning the limits shown in Table III in each of the process nodes described in Section III-B and evaluate each implementation across the typical 1.8V–3.6V operating range, in 100 mV steps. The results from these simulations form the basis for an analytical model of *Failure Sentinels*'s performance that we use to drive the optimization function. However, the SPICE simulations do not fully reflect several design choices beyond the core *Failure Sentinels* hardware. In order to accurately represent a real *Failure Sentinels* implementation, we augment the analytical model with several elements beyond the SPICE results:

- We model the number and size of NVM lookup table entries to fulfill the NVM overhead constraint, and factor in their effect on *Failure Sentinels*'s accuracy using the piecewise-linear interpolation strategy described in Section III-H.
- We include temperature as another limiting factor on *Failure Sentinels*'s accuracy and assume a maximum temperature-induced RO frequency deviation of 2% according to the FPGA-based experiments in Section V-C.
- We add a rejection filter to ensure the resulting configuration is realizable and correct (e.g., the RO is never enabled long enough to overflow the counter).

In general, the resulting Pareto frontier is five-dimensional in each of the performance parameters. Given that NVM and die space consumption have minimal impact on operational performance (as long as the code/calibration data still fit in the NVM and *Failure Sentinels* fits on the chip), we expect typical *Failure Sentinels* deployments to be constrained primarily by sampling frequency, power, or resolution. For visualization, we reduce the dimensionality of the frontier by only plotting the first three performance parameters in Table III with the knowledge that each solution satisfies the limit on NVM overhead and transistor count. Figure 5 shows the trade space for *Failure Sentinels* in 90nm technology; each point denotes the performance of a different Pareto-optimal configuration.

Failure Sentinels's flexibility enables designers to precisely tune performance to the needs of their specific application by compromising on each of the three performance parameters shown. Sampling frequency is the primary driver of current consumption in the design space we explore because temperature variations rather than current consumption set the limit on *Failure Sentinels*'s resolution (see Section V-C). Figure 5 shows the current-resolution-sample rate trade space accounting for the temperature-induced limit; reducing sampling granularity (e.g., from 38 mV to 48 mV) reduces mean current consumption by 14% at the highest sampling rate of 10 kHz. This tradeoff becomes more favorable at both lower sampling rates and smaller process nodes; at a 10 kHz sample rate, there is an 8% difference in current consumption between the finest (27 mV) and coarsest (50 mV) granularities for the 65nm implementation of *Failure Sentinels*. For all configurations and all technologies, the RO represents over 90% of *Failure Sentinels*'s total current consumption. Given that RO length only affects the power consumption of the supporting components (see Section III-D), this indicates that duty cycle—a function of sampling frequency and enable-time—is the primary determinant of current consumption.

In each case, *Failure Sentinels* dramatically reduces the power budget required for voltage monitoring hardware and provides the resolution and speed performance needed to enable the most sophisticated energy harvesting runtimes available. Assuming a 1.8V dynamic range, Figure 6 shows that *Failure Sentinels* offers between 5 and 6 bits of resolution depending on feature size while consuming, in total, less than 1 μA—enabling sophisticated energy harvesting systems with negligible power overhead. *Failure Sentinels* eliminates between 59%–77% of the system's energy overhead while enabling the same power-based intermittent runtimes as an ADC. Even compared to single-bit analog comparators supporting a simple just-in-time checkpointing system, *Failure Sentinels* increases energy available to software by 24%–45%.

B. Failure Sentinels Scales with Technology

Failure Sentinels's fully-digital design enables it to scale down with the rest of the processor to maximize performance. First, smaller process nodes enable lower-power operation, all other parameters being equal: switching from 130nm to the 90nm process, we observe a 14% reduction in power consumption—with a similar reduction from 90nm to 65nm.

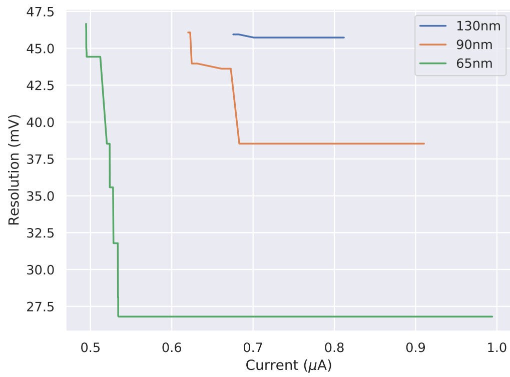

Fig. 6: Pareto-optimal configurations for each technology with F_s = 5 kHz.

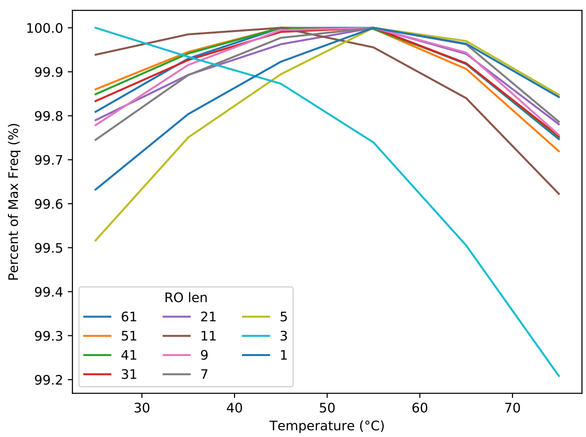

Fig. 7: RO frequency variation with temperature on Xilinx Virtex-7 FPGA.

Second, transistor delays in smaller technologies are also more sensitive to supply voltage variations [62]: our experiments show that RO frequency in the 65nm process is approximately 2% more sensitive to supply voltage than in the 90nm process and 14% more sensitive than the 130nm process.

To predict *Failure Sentinels*'s performance trends from current energy harvester feature sizes to near-future ones, we explore the trade space in each technology discussed in Section III-B around the $F_s = 5kHz$ operating point. Figure 6 shows that at the same sample rate, smaller feature sizes enable both lower current and finer resolution operation for *Failure Sentinels*. These results indicate that *Failure Sentinels* effectively removes the power/size bottleneck of highly-analog circuits and enables energy harvesters to better leverage the advantages of transistor scaling.

C. Temperature Variation

To build a picture of how operational conditions affect *Failure Sentinels*'s performance, we need to examine the impact of thermal fluctuations on RO frequency. Environments where the temperature changes dramatically have the potential to reduce the system's accuracy because *Failure Sentinels* misinterprets temperature-induced frequency changes as voltage changes. Temperature affects digital circuits by changing gate delay,[9] which in turn affects the frequency of the RO. For *Failure Sentinels*, the circuitry supporting the RO is largely temperature-independent: the voltage divider depends only on the *relative* differences between each device, and temperature affects each device equally. Temperature changes the maximum operating frequency of the level shifter by changing transistor drive strength, but our results indicate that RO frequency is always well below the level shifter's maximum.

Monitor	Sys. Current (μA)	Res. (mV)	F_s (kHz)	V_{ckpt} (V)
Ideal	112.3	Infinite	Infinite	1.82
FS (LP)	112.5	50	1	1.87
FS (HP)	113.6	38	10	1.86
Comparator	147.3	30	3030*	1.86
ADC	377.3	0.293	200	1.87

TABLE IV: Voltage monitors we evaluate within a full system. FS (LP) uses a 67-stage RO with a 49-entry LUT of 8-bit values, while FS (HP) uses a 7-stage RO with a 52-entry LUT of 10-bit values. Both versions uses a 6-bit counter and a 1 μs enable time. *Comparator response time is 330 ns.

Given that the RO is the primary factor in *Failure Sentinels*'s temperature sensitivity, we implement a range of RO sizes on a Xilinx Artix-7 FPGA. Using a TestEquity 123H temperature chamber [64], we vary the operating environment from room temperature (25°C) up to 75°C to encompass the typical operating range of energy harvesting devices. We let the device stabilize at the target temperature for an hour before measurements. For each configuration and temperature, we report the average of 1000 RO count measurements. Figure 7 illustrates the relative change in frequency across temperatures for all implemented ROs. We consider the temperature-induced error to be the largest frequency change between any two frequencies. Much like voltage-induced changes, temperature-induced changes are similar across RO sizes, because only one gate switches at a time. We double the 1% maximum effect shown in Figure 7 to create a conservative, worst-case 2% thermal error. This error fits past work measuring RO and delay line sensitivity to temperature [43], [65].

This thermal error serves as an upper bound on *Failure Sentinels*'s resolution. Our analytical model indicates that temperature-induced frequency changes approximately double *Failure Sentinels*'s error, motivating future work reducing *Failure Sentinels*'s temperature sensitivity. One potential approach is to increase the interconnect length between each inverter; because transistors are significantly more sensitive than interconnects to temperature changes [66], increasing the RO delay due to interconnect reduces *Failure Sentinels*'s overall temperature sensitivity. Because longer interconnects may affect *Failure Sentinels*'s voltage sensitivity, we leave a detailed exploration of this area for future work.

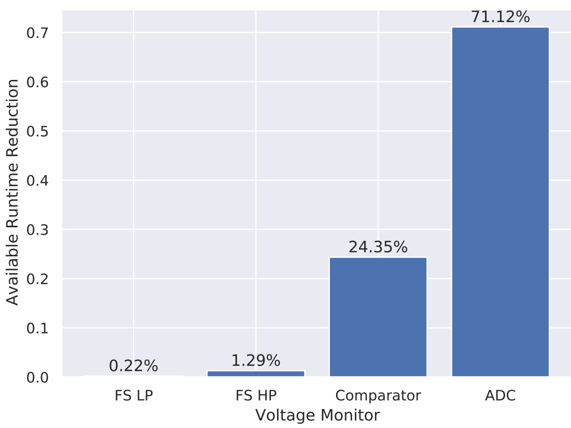

Fig. 8: Reduction in available time to process application code, normalized to ideal monitoring.

D. System-level Impact

In order to determine the impact *Failure Sentinels* has on a typical intermittent system in an energy-scarce environment, we compare it to existing solutions in the context of a simulated solar-powered energy harvester using the EnHANTs irradiance dataset [67] for a pedestrian in New York City at night. Similar to past energy harvesting architectural exploration [34], we use this simulation framework to explore the effect of different *Failure Sentinels* configurations on system performance, measured in time available for executing application code.

a) Evaluation Parameters: We compare *Failure Sentinels* to the analog alternatives on the MSP430FR5969 [25] detailed in Table I. We evaluate one 90nm *Failure Sentinels* implementation optimized for high performance (HP) and one for low power (LP), taken from opposite extremes of the objective space exploration in Figure 5. We model a typical energy harvesting sensor using a 5 cm^2, 15% efficient solar panel to charge a 47 μF storage capacitor; when the capacitor reaches the enable voltage of 3.5V, the microcontroller and a peripheral accelerometer [68] begin consuming power. Both devices operate until the supply capacitor reaches a checkpoint voltage detailed below, at which point the microcontroller stops application code and stores a checkpoint in NVM. We model the current consumption of the microcontroller core, accelerometer, and voltage monitor when the device is executing, and a leakage current of 0.5 μA at all times.

b) Checkpointing Mechanics: We model the worst-case checkpoint behavior as writing all volatile data to non-volatile FRAM, which takes 8.192 ms at a clock frequency of 1 MHz on our microcontroller. This execution time combined with total current draw and supply capacitance sets the ideal minimum voltage at which the microcontroller has just enough energy to complete the checkpoint. However, the limited accuracy of each system prevents us from achieving this minimum voltage. We add the measurement resolution of each device to the theoretical minimum to ensure the checkpoint will always complete despite worst-case measurement error and show the final checkpoint voltage for each system in Table IV. The similar checkpoint voltages across each monitor, despite dramatic differences in resolution, show how current monitors are over-optimized for resolution because the additional energy drawn from the capacitor is consumed by the monitor itself. Finally, we consider the effect of monitor sampling frequency (because capacitor voltage changes over the course of a sample). The effect of sampling frequency on accuracy is small for this scenario—2mV in the worst case using FS (LP)—showing that reducing sampling frequency is an effective way to reduce power consumption without sacrificing performance.

c) Performance Comparison: Table IV and Figure 8 illustrate the performance improvement of *Failure Sentinels* over analog-based alternatives for our checkpointing system. We normalize all runtime results to performance using the ideal voltage monitor, representing perfect sampling and zero overhead from monitoring hardware. Both implementations of *Failure Sentinels* achieve near-ideal runtime, compared to the 24% and 70% runtime penalties of the analog solutions—illustrating *Failure Sentinels*'s ability to maximize the time and energy available for application code.

d) Discussion: While we evaluate *Failure Sentinels* here based on a typical batteryless sensor mote, different system-level design choices place different demands on the voltage monitoring hardware. Systems with smaller supply capacitors require a monitor with a higher sampling frequency because the supply capacitor discharges more per unit of time, but designers must balance higher sampling frequency with the corresponding current draw. Conversely, monitor resolution becomes more important as the size of the supply capacitor increases because the voltage offset represents increasingly more energy that could have otherwise been used for computation. Broadly, we expect small sensor motes to favor a low-current, low-resolution implementation of *Failure Sentinels* while platforms with comparatively large supply capacitors and active power draws (e.g., energy harvesting satellites [4]) benefit more from a high-resolution implementation of *Failure Sentinels* when the additional energy extracted from the capacitor outweighs the increased draw of the monitor itself. Emerging energy-aware systems beyond checkpointing (Section II-C) will further exercise the voltage monitoring trade space we explore, highlighting the value of *Failure Sentinels*'s flexibility.

VI. Conclusion

Failure Sentinels leverages the voltage-dependent gate delay of CMOS devices to eliminate the need for ill-suited, high-power analog hardware to monitor available energy. We design *Failure Sentinels* to provide *just enough* performance using only the lowest power, most scalable hardware available to designers: the transistor. A focus of our design is identifying and

[9] Temperature affects digital gates by reducing carrier mobility (increasing propagation delay) and reducing threshold voltage (decreasing propagation delay) [63]; each of these effects dominates in different circumstances.

operating at the sweet spot of the transistor delay and voltage relationship, where dynamic power is reduced and sensitivity is most linear. We incorporate *Failure Sentinels* into a RISC-V system-on-chip and provide a software-queriable register for energy availability, making energy availability a first-class abstraction of the hardware. Our evaluation shows that *Failure Sentinels* reduces power consumption by between 59% and 77% compared to conventional analog-to-digital converters—without compromising system performance. Replacing one-bit voltage comparators with *Failure Sentinels* reduces power consumption by between 24% and 45%, while also enabling a myriad of new power-responsive techniques to improve whole-system efficiency and performance.

These results show that enabling sophisticated intermittent computing support on even the smallest, lowest-power devices is possible without a substantial increase in price or power consumption. *Failure Sentinels*'s low power and space overhead implies that it could even be integrated onto devices smaller than microcontrollers to support sophisticated intermittently operated peripherals, expanding the horizons for future energy harvesting designs and deployments.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their feedback and suggestions that enhanced the quality of this work. The project depicted is sponsored by the Defense Advanced Research Projects Agency. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. Approved for public release; distribution is unlimited.

REFERENCES

[1] X. Wu, I. Lee, Q. Dong, K. Yang, D. Kim, J. Wang, Y. Peng, Y. Zhang, M. Saliganc, M. Yasuda, K. Kumeno, F. Ohno, S. Miyoshi, M. Kawaminami, D. Sylvester, and D. Blaauw, "A 0.04mm316nw wireless and batteryless sensor system with integrated cortex-m0+ processor and optical communication for cellular temperature measurement," in *2018 IEEE Symposium on VLSI Circuits*, 2018, pp. 191–192.

[2] M. Buettner, R. Prasad, A. Sample, D. Yeager, B. Greenstein, J. R. Smith, and D. Wetherall, "Rfid sensor networks with the intel wisp," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 393–394. [Online]. Available: https://doi.org/10.1145/1460412.1460468

[3] IBM, "Combating fraud with blockchain and crypto-anchors," 2018. [Online]. Available: https://www.research.ibm.com/5-in-5/crypto-anchors-and-blockchain/

[4] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 939–954. [Online]. Available: https://doi.org/10.1145/3373376.3378473

[5] R. Jurdak, A. G. Ruzzelli, G. M. P. O'hare, and C. V. Lopes, "Mote-based underwater sensor networks: Opportunities, challenges, and guidelines," 2008.

[6] M. Alhaideri, M. Rushanan, D. F. Kune, and K. Fu, "The moo and cement shoes: Future directions of a practical sense-control-actuate application," September 2013, presented at First International Workshop on the Swarm at the Edge of the Cloud (SEC'13 @ ESWeek), Montreal. [Online]. Available: http://terraswarm.org/pubs/111.html

[7] Y. K. Ramadass and A. P. Chandrakasan, "An efficient piezoelectric energy harvesting interface circuit using a bias-flip rectifier and shared inductor," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 189–204, 2010.

[8] U. Karthaus and M. Fischer, "Fully Integrated Passive UHF RFID Transponder IC With 16.7-μ W Minimum RF Input Power," in *IEEE Journal of Solid-State Circuits, Volume 38*, 2003.

[9] K. Ma, Y. Zheng, S. Li, K. Swaminathan, X. Li, Y. Liu, J. Sampson, Y. Xie, and V. Narayanan, "Architecture exploration for ambient energy harvesting nonvolatile processors," in *IEEE International Symposium on High Performance Computer Architecture*, ser. HPCA, Feb 2015, pp. 526–537.

[10] J. Hester, T. Scott, and J. Sorber, "Ekho: Realistic and repeatable experimentation for tiny energy-harvesting sensors," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, 2014.

[11] H. Sodano, G. Park, and D. Inman, "Estimation of Electric Charge Output for Piezoelectric Energy Harvesting," in *Strain, Volume 40*, 2004.

[12] B. Ransford, J. Sorber, and K. Fu, "Mementos: System Support for Long-Running Computation on RFID-Scale Devices," in *Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2011.

[13] D. Balsamo, A. S. Weddell, G. V. Merrett, B. M. Al-Hashimi, D. Brunelli, and L. Benini, "Hibernus: Sustaining computation during intermittent supply for energy-harvesting systems," *IEEE Embedded Systems Letters*, vol. 7, no. 1, pp. 15–18, 2015.

[14] D. Balsamo, A. S. Weddell, A. Das, A. R. Arreola, D. Brunelli, B. M. Al-Hashimi, G. V. Merrett, and L. Benini, "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, pp. 1968–1980, 2016.

[15] H. Jayakumar, A. Raha, and V. Raghunathan, "QUICKRECALL: A Low Overhead HW/SW Approach for Enabling Computations across Power Cycles in Transiently Powered Computers," in *International Conference on VLSI Design and International Conference on Embedded Systems*, 2014.

[16] H. Williams, X. Jian, and M. Hicks, "Forget failure: Exploiting sram data remanence for low-overhead intermittent computation," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 69–84. [Online]. Available: https://doi.org/10.1145/3373376.3378478

[17] K. Maeng and B. Lucia, "Supporting peripherals in intermittent systems with just-in-time checkpoints," in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 1101–1116. [Online]. Available: https://doi.org/10.1145/3314221.3314613

[18] Xilinx, "7 series fpgas data sheet: Overview," 2018. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf

[19] M. Buettner, B. Greenstein, and D. Wetherall, "Dewdrop: An energy-aware runtime for computational rfid," in *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'11. USA: USENIX Association, 2011, p. 197–210.

[20] B. Lucia and B. Ransford, "A simpler, safer programming and execution model for intermittent systems," in *Conference on Programming Language Design and Implementation*, ser. PLDI, 2015, pp. 575–585.

[21] J. V. D. Woude and M. Hicks, "Intermittent computation without hardware support or programmer intervention," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, Nov. 2016, pp. 17–32. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/vanderwoude

[22] M. Hicks, "Clank: Architectural support for intermittent computation," in *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, 2017, pp. 228–240.

[23] A. Colin and B. Lucia, "Chain: Tasks and channels for reliable intermittent programs," in *International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ser. OOPSLA, Oct. 2016, pp. 514–530.

[24] K. Maeng, A. Colin, and B. Lucia, "Alpaca: Intermittent execution without checkpoints," in *International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ser. OOPSLA, Oct. 2017, pp. 96:1–96:30.

[25] T. Instruments, "MSP430FR596x, MSP430FR594x Mixed-Signal Microcontrollers," 2018, https://www.ti.com/lit/ds/symlink/msp430fr5969.pdf.

[26] Microchip, "PIC16(L)F15356/75/76/85/86," 2016, http://ww1.microchip.com/downloads/en/devicedoc/40001866a.pdf.

[27] T. Instruments, "MIXED SIGNAL MICROCONTROLLER," 2013, https://www.ti.com/lit/ds/symlink/msp430g2553.pdf.

[28] G. Ge, C. Zhang, G. Hoogzaad, and K. A. A. Makinwa, "A single-trim cmos bandgap reference with a 3σ inaccuracy of $\pm 0.15\%$ from $-40°$c to $125°$c," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 11, pp. 2693–2701, 2011.

[29] T. Instruments, "ADS7042 Ultra-LowPower, Ultra-Small Size, 12-Bit, 1-MSPS, SAR ADC," 2015, https://www.ti.com/lit/ds/sbas608c/sbas608c.pdf.

[30] R. H. Dennard, F. H. Gaensslen, H. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of ion-implanted mosfet's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974.

[31] B. E. Jonsson, "A survey of A/D-converter performance evolution," in *International Conference on Electronics, Circuits, and Systems*, ser. ICECS, Dec 2010, pp. 766–769.

[32] K. Maeng and B. Lucia, "Adaptive dynamic checkpointing for safe efficient intermittent computing," in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. Carlsbad, CA: USENIX Association, Oct. 2018, pp. 129–144. [Online]. Available: https://www.usenix.org/conference/osdi18/presentation/maeng

[33] G. Gobieski, A. Nagi, N. Serafin, M. M. Isgenc, N. Beckmann, and B. Lucia, "Manic: A vector-dataflow architecture for ultra-low-power embedded systems," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '52. New York, NY, USA: Association for Computing Machinery, 2019, p. 670–684. [Online]. Available: https://doi.org/10.1145/3352460.3358277

[34] H. Desai and B. Lucia, "A power-aware heterogeneous architecture scaling model for energy-harvesting computers," *IEEE Computer Architecture Letters*, vol. 19, no. 1, pp. 68–71, 2020.

[35] N. A. Bhatti and L. Mottola, "Harvos: Efficient code instrumentation for transiently-powered embedded sensing," in *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2017, pp. 209–220.

[36] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE Journal of solid-state circuits*, vol. 35, no. 11, pp. 1571–1580, 2000.

[37] V. Gutnik and A. Chandrakasan, "An efficient controller for variable supply-voltage low power processing," in *1996 Symposium on VLSI Circuits. Digest of Technical Papers*, 1996, pp. 158–159.

[38] J. Park and J. A. Abraham, "A fast, accurate and simple critical path monitor for improving energy-delay product in dvs systems," in *IEEE/ACM International Symposium on Low Power Electronics and Design*, 2011, pp. 391–396.

[39] M. R. Halesh, K. R. Rasane, and H. Rohini, "Design and implementation of voltage control oscillator (vco) using 180nm technology," in *Advances in Computing, Communication and Control*, S. Unnikrishnan, S. Surve, and D. Bhoir, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 472–478.

[40] M. Bhushan, A. Gattiker, M. B. Ketchen, and K. K. Das, "Ring oscillators for cmos process tuning and variability control," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 1, pp. 10–18, 2006.

[41] M. Hempstead, N. Tripathi, P. Mauro, Gu-Yeon Wei, and D. Brooks, "An ultra low power system architecture for sensor network applications," in *32nd International Symposium on Computer Architecture (ISCA'05)*, 2005, pp. 208–219.

[42] M. Hempstead, G.-Y. Wei, and D. Brooks, "Architecture and circuit techniques for low-throughput, energy-constrained systems across technology generations," in *Proceedings of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, ser. CASES '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 368–378. [Online]. Available: https://doi.org/10.1145/1176760.1176805

[43] T. Chan, P. Gupta, A. B. Kahng, and L. Lai, "Ddro: A novel performance monitoring methodology based on design-dependent ring oscillators," in *Thirteenth International Symposium on Quality Electronic Design (ISQED)*, 2012, pp. 633–640.

[44] M. K. Mandal and B. C. Sarkar, "Ring oscillators: Characteristics and applications," *Indian Journal of Pure & Applied Physics*, vol. 48, no. 1, pp. 136–145, 2010.

[45] M. Bhushan and M. B. Ketchen, "Generation, elimination and utilization of harmonics in ring oscillators," in *2010 International Conference on Microelectronic Test Structures (ICMTS)*, 2010, pp. 108–113.

[46] G. Jovanović, M. Stojcev, and Z. Stamenkovic, "A cmos voltage controlled ring oscillator with improved frequency stability," *Scientific Publications of the State University of Novi Pazar: Applied Mathematics, Informatics, and Mechanics*, vol. 2, pp. 1–9, 06 2010.

[47] W. Xu, X. Chen, and J. Wu, "An overview of theory and techniques for reducing ring oscillator supply voltage sensitivity in mixed-signal soc," in *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, 2011, pp. 2039–2042.

[48] M. Danesh, S. T. Chandrasekaran, and A. Sanyal, "Ring oscillator based delta-sigma adcs," in *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2018, pp. 113–116.

[49] A. Sanyal, S. Li, and N. Sun, "Low-power scaling-friendly ring oscillator based $\sigma\delta$ adc," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5.

[50] A. Jayaraj, M. Danesh, S. T. Chandrasekaran, and A. Sanyal, "Highly digital second-order $\delta\sigma$ vco adc," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 7, pp. 2415–2425, 2019.

[51] P. Prabha, S. J. Kim, K. Reddy, S. Rao, N. Griesert, A. Rao, G. Winter, and P. K. Hanumolu, "A highly digital vco-based adc architecture for current sensing applications," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 8, pp. 1785–1795, 2015.

[52] Y. Cao, T. Sato, M. Orshansky, D. Sylvester, and C. Hu, "New paradigm of predictive mosfet and interconnect modeling for early circuit simulation," in *Proceedings of the IEEE 2000 Custom Integrated Circuits Conference (Cat. No.00CH37044)*, 2000, pp. 201–204.

[53] L. R. Wolfgang Lutsch, "Energy optimization tools for ultra-low-power microcontrollers," 2020. [Online]. Available: https://www.mouser.com/applications/low-power-ewc-design/

[54] M. Kulkarni and K. Hosur, "Design of a linear and wide range current starved voltage controlled oscillator for pll," *International Journal on Cybernetics & Informatics*, vol. 2, pp. 23–30, 02 2013.

[55] T. Instruments, "Msp432p401r, msp432p401m simplelink mixed-signal microcontrollers," June 2019, https://www.ti.com/lit/ds/symlink/msp432p401r.pdf.

[56] M. Yano, J. D. Penn, G. Konidaris, and A. T. Patera, "Interpolation," August 2013, https://ocw.mit.edu/courses/mechanical-engineering/2-086-numerical-computation-for-mechanical-engineers-fall-2014/nutshells-guis/MIT2_086F14_Interpolation.pdf.

[57] A. Devices, "Ltspice," September 2020, https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html.

[58] R. Kumar and V. Kursun, "Modeling of temperature effects on nano-cmos devices with the predictive technologies," in *2007 50th Midwest Symposium on Circuits and Systems*, 2007, pp. 694–697.

[59] K. Asanović, R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, D. Dabbelt, J. Hauser, A. Izraelevitz, S. Karandikar, B. Keller, D. Kim, J. Koenig, Y. Lee, E. Love, M. Maas, A. Magyar, H. Mao, M. Moreto, A. Ou, D. A. Patterson, B. Richards, C. Schmidt, S. Twigg, H. Vo, and A. Waterman, "The Rocket chip generator," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17, Apr 2016.

[60] Xilinx, "Artix-7 fpgas data sheet: Dc and ac switching characteristics," 2018. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds181_Artix_7_Data_Sheet.pdf

[61] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020.

[62] M. Alioto and G. Palumbo, "Impact of supply voltage variations on full adder delay: Analysis and comparison," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 12, pp. 1322–1335, 2006.

[63] S. Park, C. Min, and S. Cho, "A 95nw ring oscillator-based temperature sensor for rfid tags in 0.13um cmos," in *2009 IEEE International Symposium on Circuits and Systems*, 2009, pp. 1153–1156.

[64] TestEquity, "Model 123H Temperature/Humidity Chamber With F4 Controller and EZ-Zone Limit Operation and Service Manual," 2010, https://www.testequity.com/UserFiles/documents/pdfs/123Hman.pdf.

[65] J. Tschanz, K. Bowman, S. Walstra, M. Agostinelli, T. Karnik, and V. De, "Tunable replica circuits and adaptive voltage-frequency tech-

niques for dynamic voltage, temperature, and aging variation tolerance," in *2009 Symposium on VLSI Circuits*, 2009, pp. 112–113.

[66] K. M. Zick and J. P. Hayes, "Low-cost sensing with ring oscillator arrays for healthier reconfigurable systems," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 5, no. 1, Mar. 2012. [Online]. Available: https://doi.org/10.1145/2133352.2133353

[67] M. Gorlatova, A. Wallwater, and G. Zussman, "Networking low-power energy harvesting devices: Measurements and algorithms," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 1602–1610.

[68] A. Devices, "Micropower, 3-Axis, ±2g/±4g/±8gDigitalOutput MEMS Accelerometer," 2015, https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL362.pdf.