Exploring the Effect of Energy Storage Sizing on Intermittent Computing System Performance

Jie Zhan, Geoff V. Merrett, *Senior Member, IEEE,* and Alex S. Weddell, *Member, IEEE*

***Abstract*—Batteryless energy-harvesting devices promise to deliver a sustainable Internet of Things. Intermittent computing is an emerging area, where application forward progress, i.e. computation beneficial to the progress of the active application, is maintained by saving volatile computing state into non-volatile memory before power interruptions, and restored afterwards. Conventional intermittent computing approaches typically minimize energy storage to reduce device dimensions and interruption periods, but this can result in high state-saving and -restoring overheads and impede forward progress. In this paper, we argue that adding a small amount of energy storage can significantly improve forward progress. We develop an intermittent computing model that accurately estimates forward progress, with an experimentally validated mean error of 0.5%. Using this model, we show that sizing energy storage can improve forward progress by up to 65% with a constant current supply, and 43% with real-world photovoltaic sources. An extension to this approach, which uses a cost function to trade-off the energy storage size against forward progress, can save 83% of capacitor volume and 91% of interruption periods while maintaining 93% of the maximum forward progress.**

***Index Terms*—Intermittent computing, energy harvesting, energy storage, forward progress, batteryless, internet of things.**

I. Introduction

INTERNET of Things (IoT) devices are becoming ubiquitous, with forecasts of hundreds of billions being installed in the near future [1]. They are conventionally battery-powered, thus have constrained lifespans, necessitating inconvenient periodic battery replacement. Energy-harvesting is a potential solution. Environmentally harvested power is, however, intrinsically variable and intermittent [2]. Traditionally, large energy storage devices such as rechargeable batteries or supercapacitors are used to smooth out supply variability [3]. Unfortunately, these increase cost and device dimensions [4], raise pollution concerns [5], and still limit lifespans [6].

Recently, *intermittent computing systems* (ICSs) have been proposed as an alternative [7]. Instead of using large energy storage devices to sustain execution, they tolerate power interruptions by saving the state of the system into non-volatile memory (NVM) so that computation can continue when power is restored. They may save this state (e.g. CPU registers and RAM contents) either *statically* at pre-defined points, or *reactively* by detecting when the supply is about to fail [7].

The authors are with the School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK (email: {j.zhan, gvm, asw}@ecs.soton.ac.uk).

This work was supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) under EP/P010164/1 and China Scholarship Council (CSC) under 201706310132.

All data associated with this paper are openly available at https://doi.org/10.5258/SOTON/D1785.

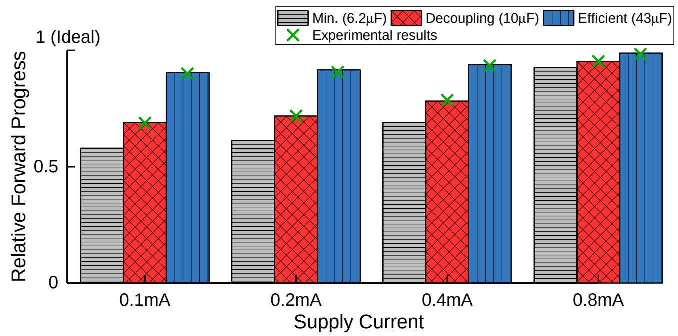


Fig. 1. The relationship between energy storage capacitance and ICS forward progress, for various supply currents.

Static approaches save state at points determined at design or compile time, either by inserting checkpoints [8], [9] or decomposing a program into atomic tasks[1] [10], [11]. After a power interruption, progress rolls back and resumes from the last saved checkpoint or task boundary. This can introduce issues such as violation of data memory consistency, along with wasting energy on lost and re-executed progress.

Conversely, reactive approaches monitor the supply voltage and only save state when it falls below a threshold [12]–[14], which is set high enough to reliably save state even with a total and immediate drop-off in harvested energy. They then enter a low-power mode, in many cases preserving their volatile memory and avoiding re-execution. These typically make more forward progress than static approaches, e.g. a 2.5× mean computational speedup [15].

In ICSs, *forward progress* denotes the computation beneficial to the progress of the active application, excluding lost progress due to power failures and state-saving and -restoring operations [16]. The amount of forward progress directly determines application performance, e.g. program iteration rate or task completion time. In this paper, to allow fair comparison, we define normalized forward progress as *the ratio of the effective execution time to the total elapsed time*, without being restricted to a specific workload.

With the goal of minimizing device dimensions and interruption periods, most ICS approaches adopt a minimum amount of energy storage [12], [17]–[20]. This is typically just sufficient for the most energy-expensive atomic operation. However, our assertion is that this can be *inherently inefficient*

[1] Atomic operations in ICSs denote operations that should be completed in one continuous period. If an atomic operation is interrupted by a power failure, it should be re-executed rather than resumed. Examples of atomic operations include saving and restoring volatile state, transmitting and receiving packets, and sampling sequences of data from sensors.

in terms of time and energy. We show that a system with minimum energy storage frequently goes through a cycle of: wake up; restore state; execute program; save state; halt. We propose that provisioning *slightly more* energy storage can prolong the operating cycles, reduce the frequency of interruptions, and hence improve forward progress. We show with modelled and experimental results that (Fig. 1) using efficiently-sized energy storage capacitance (in this case, 43 µF) achieves up to a 55% improvement in forward progress compared to using the theoretical minimum amount of capacitance (6.2 µF). This improvement is more significant with a weaker supply.

The relationship between ICS energy storage capacitance and forward progress has not previously been defined. However, current tools for ICSs (Section II) are not practical for estimation of forward progress in a long-term deployment, and lack a method of sizing energy storage to improve forward progress while moderating the physical size and interruption periods. We present an ICS model for estimating forward progress, along with an approach for sizing energy storage in ICSs, quantifying and trading-off forward progress, capacitor volume, and interruption periods.

The main contributions of this paper are:

- A reactive ICS model which accurately estimates forward progress; experimental validation shows a 0.5% mean error (Section III).
- A model-based sizing approach that recommends appropriate energy storage capacitance in ICSs (Section IV).
- An exploration based on the model, where we analyze the energy storage sizing effect on forward progress, showing up to 65% forward progress improvement (Section V).
- An evaluation of the impact of sizing in real-world conditions using real energy availability data (Section VI). This includes a cost function-based method for trading-off parameters. In an example, this reduced capacitor volume and interruption periods by 83% and 91% respectively, while sacrificing 7% of forward progress.

The associated simulation tool is available open-source at https://git.soton.ac.uk/energy-driven/energy-storage-sizing.

II. Related Work

To explore forward progress of ICSs with regard to energy storage, simulation tools need to represent transient operation (timescales of µs-ms) as well as long-term overall performance (from days up to years).

Su *et al.* [21] modelled a dual-channel solar-powered non-volatile sensor node, and Jackson *et al.* [22] provided a model to explore battery usage in ICSs. Both were configured for long-term simulations and large energy storage (from mF-scale supercapacitors to batteries), thus cannot respond to frequent power interruptions and accurately estimate forward progress when using minimized energy storage (e.g. 4.7 µF [19]).

In contrast, a set of fine-grained models have been proposed to accurately simulate the frequent micro-operations in ICSs. NVPsim [23] is a gem5-based simulator for nonvolatile processors. Fused [24] is a closed-loop simulator which allows interaction between power consumption, power supply, and forward progress. EH model [25] can compare a range of ICS approaches in a single active period with the same energy budget, quantifying forward progress by the energy spent on effective execution. These fine-grained models are inefficient for processing long-term energy data, especially when iterative tests are needed for various system configurations.

TABLE I
Model parameters of reactive ICS

Input Parameters	
I_{harv}	Energy harvester current supply
C	Energy storage capacitance
Configuration Parameters	
I_{exe}	Execution current draw
I_{lpm}	Low-power mode current draw
I_r	Restore current draw
I_s	Save current draw
I_{leak}	Leakage current draw
V_r	Restore voltage threshold
V_s	Save voltage threshold
T_r	Restore time overhead
T_s	Save time overhead
Output Parameter	
α_{exe}	Normalized forward progress

Besides models and simulators, hardware emulators of energy harvesters can provide replicate power profiles recorded from energy harvesters for experimental comparisons [26], [27]. Though they provide practical results, hardware emulations are limited by hardware options and are generally impractical for performing long-term trials.

To address the above problems, we provide a reactive ICS model to estimate forward progress even with minimized energy storage, and a simulation tool that enables exploration with long-term real-world environmental conditions. Together, they enable the exploration of the effect of energy storage size on the forward progress of ICSs. Further, we provide a sizing approach which recommends appropriate energy storage capacitance when deploying ICSs.

III. Reactive ICS Modelling

To facilitate the understanding and exploration of reactive ICSs, we present a model which outputs the normalized forward progress α_{exe}. Parameters of this model are listed in Table I. We assume that all input and configuration parameters remain constant in this model derivation, but later provide a dynamic process for cases where parameters change dynamically.

For brevity, I_{in} denotes the usable input current as expressed in (1). The effect of capacitor leakage current, I_{leak}, is discussed at the end of Section III-B.

$$I_{in} = I_{harv} - I_{leak} \tag{1}$$

A. Operating Modes of Reactive ICS

The behavior of reactive ICSs can be classified into three operating modes depending on the supply current, as shown in Fig. 2. These are differentiated by the relationship between

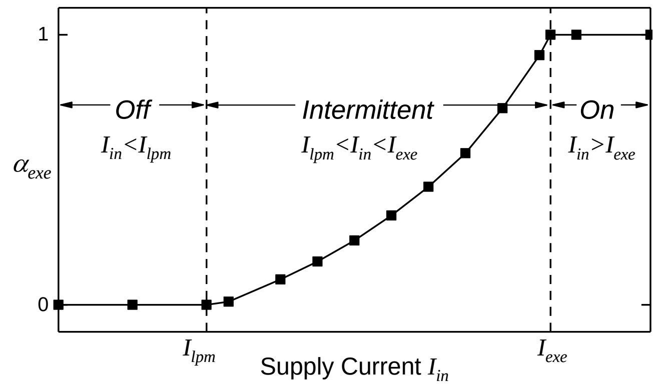


Fig. 2. Operating modes of reactive ICSs, and achieved forward progress against supply current.

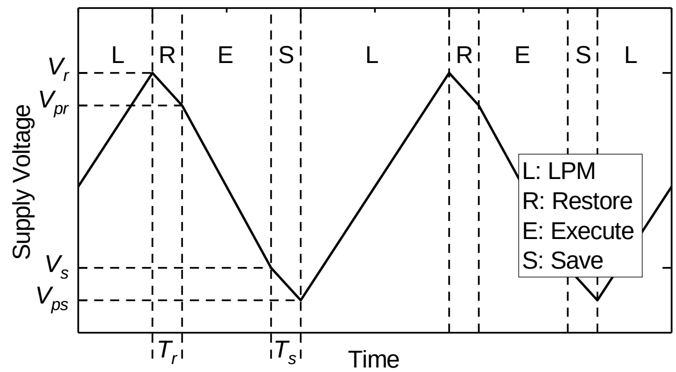


Fig. 3. Operating cycles in the *Intermittent* mode.

input current I_{in} and the system's current draw in its low-power mode (LPM) or active modes, i.e. I_{lpm} and I_{exe}. We define the three modes as:

- *Off* mode: When $I_{in} < I_{lpm}$, the system stays inactive. The supply voltage V_{cc} cannot rise above the restore threshold V_r to wake the system and start execution. The LPM current I_{lpm} includes the consumption of voltage monitoring circuits and system idle current.
- *On* mode: When $I_{in} > I_{exe}$, the system executes constantly as the supply voltage V_{cc} never drops below V_s. V_{cc} grows until I_{in} and I_{exe} are in equilibrium, which may result from I_{in} decreasing due to poor impedance matching, or I_{exe} increasing due to either greater current draw at higher voltage or dissipation through overvoltage protection circuits.
- *Intermittent* mode: When $I_{lpm} < I_{in} < I_{exe}$, the system executes intermittently after $V_{cc} > V_r$ and before $V_{cc} < V_s$. V_{cc} can rise above V_r and the system starts execution. However, the stored energy is then consumed by the load as $I_{in} < I_{exe}$, causing V_{cc} to eventually drop below the save threshold V_s, where the system saves its state and enters LPM. The system stays in LPM until V_{cc} rises to V_r again and then resumes execution. In general, a higher I_{in} leads to more forward progress in this mode, but the exact relationship between I_{in} and forward progress requires further analysis.

B. Formulating Forward Progress

Next, we derive formulations to calculate α_{exe} from I_{in} and energy storage capacitance C. We then explore the effect of capacitor leakage on maximum forward progress.

In the *On* and *Off* modes, the normalized forward progress is trivial to find (simply 1 and 0 respectively). In the *Intermittent* mode, as shown in Fig. 3, the system goes through four intervals in turn, i.e. charging, restoring, executing, and saving, with current consumption of I_{lpm}, I_r, I_{exe}, and I_s in each interval respectively. The normalized forward progress, i.e. effective execution time ratio, is indicated as T_{exe}/T_{cycle}, where T_{exe} is the time spent on effective execution in one operating cycle and T_{cycle} is the period of operating cycles. Hence, the forward progress given all supply levels is expressed as:

$$\alpha_{exe} = \begin{cases} 0 & , \ \textit{Off}\,(I_{in} < I_{lpm}) \\ \dfrac{T_{exe}}{T_{cycle}} & , \ \textit{Intermittent}\,(I_{lpm} < I_{in} < I_{exe}) \\ 1 & , \ \textit{On}\,(I_{in} > I_{exe}) \end{cases} \quad (2)$$

In the following analysis, we focus on deriving T_{exe}/T_{cycle} in the *Intermittent* mode. Let V_{pr} (post-restore) and V_{ps} (post-save) denote the voltage after restoring and saving operations. Referring to Table I and Fig. 3, V_{pr} and V_{ps} can be calculated as:

$$V_{pr} = V_r + \frac{T_r(I_{in} - I_r)}{C} \quad (3)$$

$$V_{ps} = V_s + \frac{T_s(I_{in} - I_s)}{C} \quad (4)$$

With (3), the time spent on effective execution T_{exe} in one operating cycle can be expressed as:

$$T_{exe} = \frac{C(V_{pr} - V_s)}{I_{exe} - I_{in}} \quad (5)$$

Analogously, with (4), the charging interval can be described as:

$$T_{charge} = \frac{C(V_r - V_{ps})}{I_{in} - I_{lpm}} \quad (6)$$

With (5) and (6), the period of an operating cycle is:

$$T_{cycle} = \quad T_{charge} + T_r + T_{exe} + T_s \quad (7)$$

Finally, combining (3)–(7), we obtain normalized forward progress α_{exe} in the *Intermittent* mode as:

$$\begin{aligned} \alpha_{exe} &= \frac{T_{exe}}{T_{cycle}} \\ &= \frac{\frac{C(V_r - V_s) + T_r(I_{in} - I_r)}{I_{exe} - I_{in}}}{\frac{C(V_r - V_s) + T_s(I_s - I_{lpm})}{I_{in} - I_{lpm}} + \frac{C(V_r - V_s) + T_r(I_{exe} - I_r)}{I_{exe} - I_{in}}} \end{aligned} \quad (8)$$

In the numerator T_{exe}, $C(V_r - V_s)$ represents the amount of charge in the capacitor available for restoring and executing. $T_r(I_{in} - I_r)$ represents the charge used by a restore operation. $I_{exe} - I_{in}$ is the rate of charge consumption from the energy storage during execution.

To explore the effect of energy storage on forward progress, we need to analyze $d\alpha_{exe}/dC$. Here, if we assume that I_{leak} remains constant, α_{exe} keeps increasing and approaches $(I_{in}-I_{lpm})/(I_{exe}-I_{lpm})$ when energy storage capacitance C increases. Defining $(I_{in}-I_{lpm})/(I_{exe}-I_{lpm})$ as α_{exe_ideal}, $\alpha_{exe} = \alpha_{exe\_ideal}$ is an ideal case, where restore and save overheads are absent.

In an electrolytic capacitor, however, I_{leak} typically increases with C with the following relationship [28]:

$$I_{leak} = kCV_{cc} \tag{9}$$

where k is a constant normally in a range 0.01 to 0.03 ($\frac{A}{F \cdot V}$). Combining (9) with (1), dI_{in}/dC is $-kV_{cc}$, meaning I_{in} decreases linearly as C increases. Thus, when C increases, α_{exe} keeps approaching α_{exe_ideal} while α_{exe_ideal} decreases. Hence, we believe that there is a capacitance value that leads to the maximum α_{exe} considering I_{leak} increases with C.

C. Dynamic process

The above model assumes all parameters are constant, which is useful for fast exploration in cases where this can be considered to approximately hold true (this is used for the analysis of principal sizing effects presented in Section V). For dynamically-varying parameters (e.g. a dynamic harvesting profile), we also implement a dynamic process, where the supply voltage is calculated with current flows across small time steps, hence updating system state accurately. This is used for the exploration of real-world energy conditions in Section VI.

D. Model Validation

We implemented and parameterized a reactive ICS [29] on a TI MSP430FR6989 microcontroller to validate our model. The on-board decoupling capacitance was measured as 10.0 µF, and hence was the minimum capacitance that could be tested. Further capacitance was added to provide extra energy storage up to a maximum of 43 µF, as forward progress with this capacitance can approximate α_{exe_ideal} (an upper bound), which is linear to supply current I_{in} when $I_{lpm} < I_{in} < I_{exe}$ (mentioned in Section III-B). The parameters are profiled with the MCU running a Dijkstra path finding algorithm with 1696 B RAM usage at 8 MHz. The supply voltage monitoring circuits use the MCU's internal comparator and an external 3 MΩ voltage divider. The restore and save voltage thresholds are set as V_r = 2.4 V and V_s = 2.1 V respectively. The MCU shutdown voltage V_{off} is 1.8 V.

To validate the accuracy of our model, we powered the device with a range of supply currents (0–0.9 mA) to operate the device in *Intermittent* mode, and repeated the tests with three energy storage capacities: a) 10.0 µF decoupling capacitance; b) 21.5 µF (11.5 µF added); c) 43.0 µF (33.0 µF added). We compared the actual forward progress against predictions generated from our model. As shown in Fig. 4, the model-generated output matches closely with the experimental results with only 0.5% mean absolute percentage error.

In the next section, we extend this approach with a cost function for trading-off forward progress against various design factors.

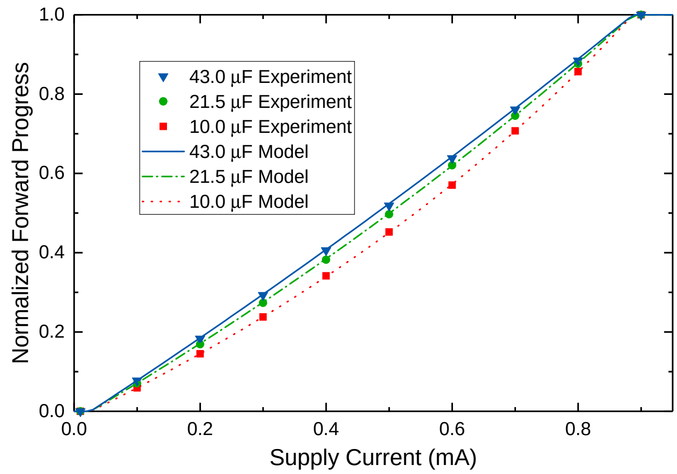


Fig. 4. Model validation with experimental and modelled forward progress.

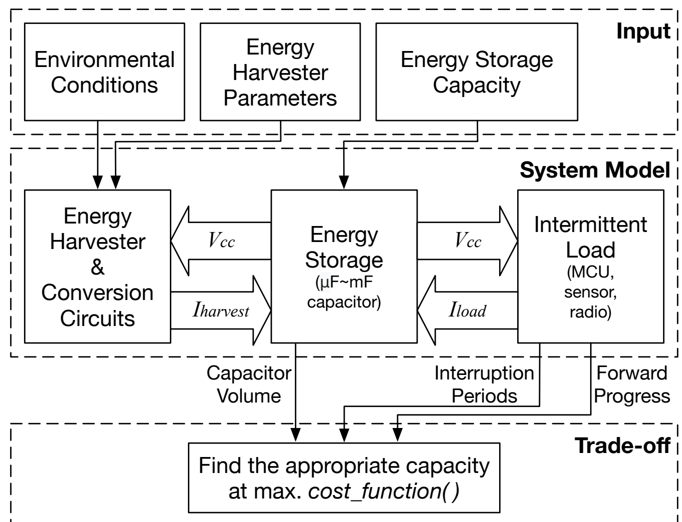


Fig. 5. Structure of the proposed system model and sizing approach.

IV. Energy Storage Sizing Approach

We propose a sizing approach which recommends appropriate energy storage capacitance for an ICS, trading-off forward progress against capacitor volume and interruption periods. We present a system model which accepts real long-term data on environmental energy conditions. The three inputs can be swept for design exploration, but we focus on energy storage in this paper. The model outputs forward progress, capacitor volume, and interruption periods (defined in Section VI-B). These are subsequently traded-off in a cost function to obtain the appropriate energy storage capacitance. This process is summarized in Fig. 5 with details explained as follows.

A. Input

A time trace of representative environmental energy conditions in the intended deployment location is provided as an input, along with the energy harvester size; for design exploration, these can optionally be changed to explore variations and scales of harvested power. A pre-defined set of energy storage capacitance values are swept through.

B. System Model

This contains three modules:

- *Energy Harvester and Conversion Circuits*: The energy harvester module transduces environmental energy into electricity. In ICSs, conversion circuits may simply be a diode to inhibit backflow of current. The energy harvester and conversion circuits can be modelled together as a module because they are usually coupled or integrated.
- *Energy Storage*: Energy storage in ICSs is usually in the form of a µF- to mF-scale capacitor. It must be sufficient to complete the most energy-expensive atomic operation, and may be formed only of the decoupling capacitor(s). This also includes an empirical model relating capacitance to capacitor volume (discussed in Section VI-C).
- *Intermittent Load*: Includes all the power consumers in an ICS, such as a microcontroller, sensors, and a radio. This module outputs forward progress and interruption periods using the model presented in Section III.

C. Trade-off

The appropriate capacitance is then found through a cost function (an example of which is presented in Section VI-C). This may trade-off forward progress against capacitor volume and interruption periods.

V. Exploration of Energy Storage Sizing

In this section, we configure the reactive ICS model presented in Section III to approximate a real ICS platform, and then present an exploration of the relationship between α_{exe} and C with respect to I_{harv} and volatile state size.

A. Model Configuration

1) Energy Storage: The energy storage is represented as an ideal capacitor with leakage current. Its terminal voltage is directly applied to the load, so is modelled as:

$$C\frac{dV_{cc}}{dt} = I_{harv} - I_{load} - I_{leak} \tag{10}$$

where I_{load} is the current consumption of the load. In this exploration, we refer to the empirical I_{leak} of AVX TAJ low-profile series tantalum capacitors, which depends on capacitance C, rated voltage V_{rated}, and terminal voltage V_{cc} [28]:

$$I_{leak} = 0.01\lambda C V_{rated} \quad (A) \tag{11}$$

where λ denotes the ratio of the actual current leakage at V_{cc} to the current leakage at V_{rated}, and λ is approximated as:

$$\lambda = 0.05 \times 20^{\frac{V_{cc}}{V_{rated}}} \tag{12}$$

We assume a typical load of < 4.0 V so, to minimize leakage, we select a device with $V_{rated} = 10$ V so as to operate between 25-40% of its rated voltage [28].

TABLE II
PROFILED MCU PARAMETERS

Parameter	Value
I_{exe}	887 µA
I_{lpm}	26 µA
I_r	971 µA
I_s	811 µA
T_r	1.903 ms
T_s	1.880 ms

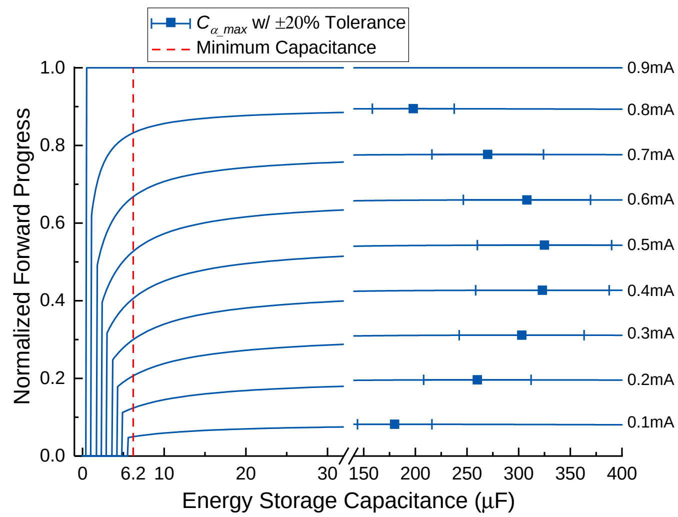


Fig. 6. Forward progress against energy storage capacitance at different levels of constant supply current. Error bars around C_{α_max} denote the impact of typical ±20% capacitance tolerance. The 30–150 µF range is omitted as forward progress in that range increases monotonically.

2) Intermittent Load: The load parameters of current draws and time overheads, as listed in Table II, were profiled with the experimental settings explained in Section III-D. We only consider computational loads in this study, as handling of peripherals in intermittent systems is still an ongoing research topic [15], [30]. The current draw was profiled with experimental measurements at a range of supply voltages. The variation of I_{lpm} between V_{off} (1.8 V) and V_r (2.4 V) is 2%, and for I_{exe} between V_s (2.1 V) and 3.3 V is 1.5%. I_{exe} also has a run-time variation of 2.8% due to a variable memory access rate. We omit these minor variations and use the mean of I_{exe} and I_{lpm} in the model. I_r and I_s are measured at V_r and V_s respectively. Given the voltage thresholds and the current consumption, the minimum energy storage capacitance is 6.2 µF. This guarantees that a save and restore operation can complete even if the incoming supply current drops instantaneously to zero. The model parameters in Table II are given as an example, and can be changed for different load characteristics. For example, T_r and T_s can be tuned for different volatile state sizes.

B. Sizing Energy Storage to Improve Forward Progress

1) Impact of Supply Current: Increasing energy storage capacitance above the minimum can improve forward progress by reducing the frequency of power interruptions, but this improvement may be offset by increased leakage. Fig. 6 shows

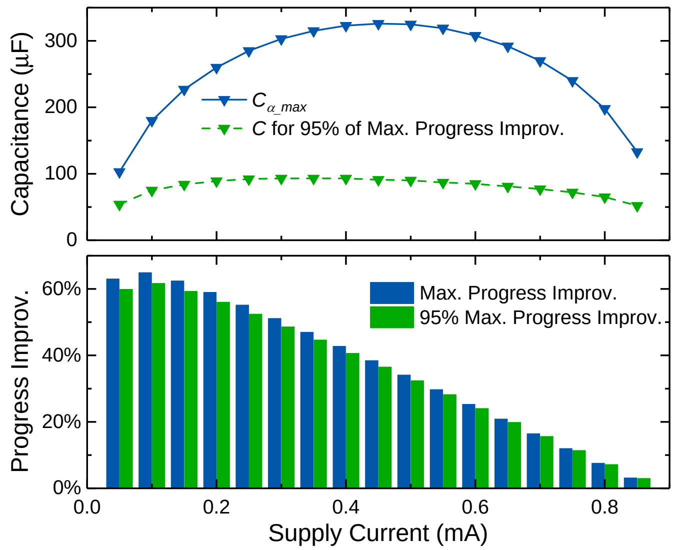


Fig. 7. Maximum forward progress improvement by sizing energy storage given a spectrum of supply current (normalized by the minimum capacitance case), with the corresponding maximum and sub-maximum (95% of maximum) capacitance.

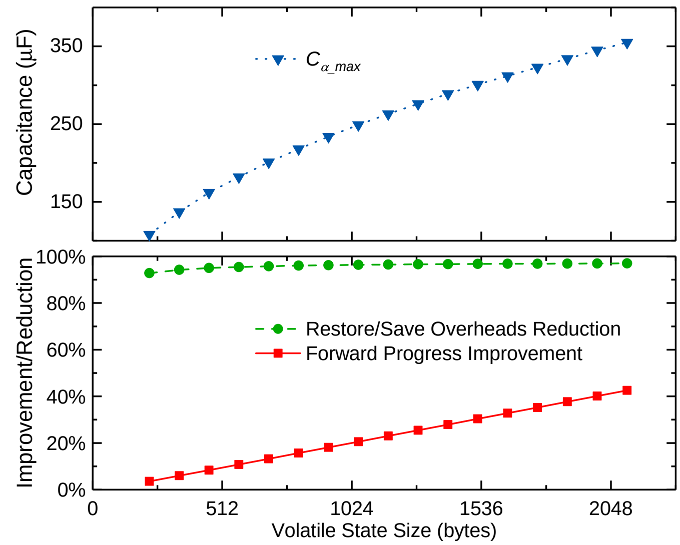


Fig. 8. Impact of RAM usage (linear to restore/save overheads) on sizing energy storage with 0.4 mA current supply. Improvement and reduction are normalized by the minimum capacitance case.

TABLE III
LINEAR SCALING RANGE OF VOLATILE STATE SIZE AND RESTORE/SAVE TIME OVERHEADS

State Size (Registers + SRAM)	Restore Time	Save Time
64B + 160B (lower bound)	232 µs	208 µs
64B + 2048B (upper bound)	2.298 ms	2.274 ms

the relationship between forward progress and energy storage capacitance for a range of constant supply currents. In this section, we denote the capacitance that leads to the maximum forward progress α_{exe} as C_{α_max}.

The minimum capacitance (dashed line in Fig. 6) is calculated to deliver correct operation even if the supply current instantaneously drops to zero. If it does not drop to zero, this means that correct operation could have continued even with a smaller capacitance, though designing a system in this way would be inadvisable owing to unpredictability of the supply. This property is illustrated in Fig. 6, in the area on the left of the dashed line. It may be observed that, for each of the current values, there is a sudden drop-off towards zero forward progress. This illustrates the hazard of setting the capacitance too small: the stored energy is too low to allow a restore and save to be undertaken.

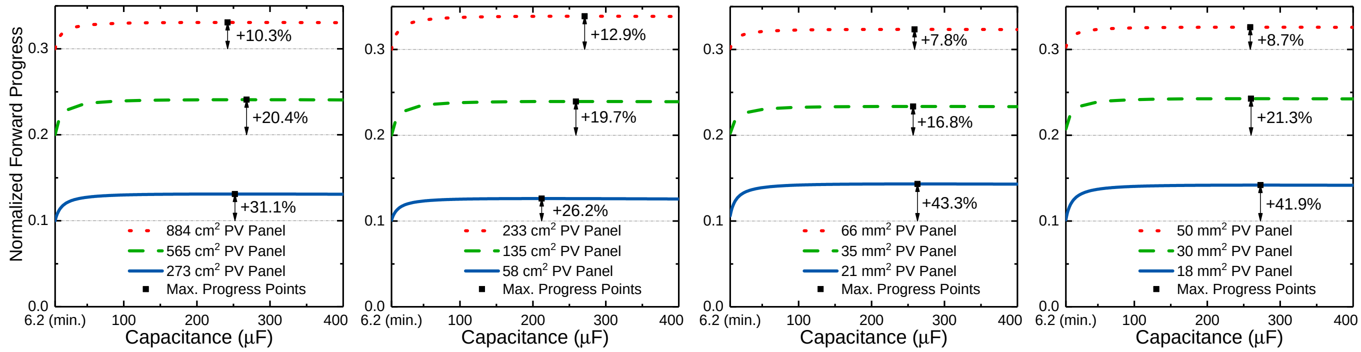
Typically, commercially-available capacitors have a $\pm20\%$ tolerance. The effect of this variation on maximum forward progress is shown to be negligible ($< 0.23\%$) in Fig. 6. However, it must be pointed out that the effect would be much more pronounced if operating at the minimum capacitance as the variation of forward progress is larger with smaller capacitance values. Thus, it is recommended that a tolerance is considered when designing ICSs with minimum capacitance.

Fig. 7 shows that an improvement in forward progress of up to 65% can be achieved when using C_{α_max} instead of the minimum. However, it may not be desirable to set the capacitance solely for maximizing forward progress, because there are often trade-offs with other factors including increased interruption periods and dimensions. While a large improvement can be delivered with C_{α_max}, as shown in Fig. 7, 95% of this gain can still be obtained with significantly smaller capacitances (mean 31% of C_{α_max}). For example, reducing from 325 µF to 90 µF gives 95% of the maximum improvement with a 0.5 mA supply.

2) Impact of Volatile State Size: The size of volatile state differs across applications with different amounts of RAM usage, and hence incurs varying time and energy overheads for restore and save operations. We measured time overheads of restore and save operations in the minimum case (64B register data and a 160B stack) and the maximum case (64B register data and a full 2048B RAM) respectively as shown in Table III. As these time overheads are expected to be linear to the state size [31], the model can be tuned for various volatile state sizes by linearly scaling the profiled values.

An example of this is plotted in Fig. 8. The forward progress improvement by sizing energy storage increases with the volatile state size, and C_{α_max} grows accordingly. The improvement becomes insignificant when the volatile state size is small because the restore and save overheads are already negligible. For example, when the workload uses the least volatile state (the leftmost point), the maximum progress improvement is only 3.6% although the restore and save overheads are reduced by 93%.

Where the size of the volatile state may vary at run time, a different capacitor size within the range 108–355 µF may have been recommended (Fig. 8). However, as can be seen from Fig. 6, there is a minimal difference in forward progress across this range. In the worst case, a 2.7% reduction results from setting C_{α_max} for the minimum state size, while running with the largest state size.



(a) EnHANTs Setup A (b) EnHANTs Setup D (c) NREL Denver 2018 (d) NREL Hawaii 2018

Fig. 9. Improvement of average forward progress by sizing energy storage given different PV panel areas under real-world energy source conditions. The model is able to find the PV panel area required for achieving the target mean forward progress.

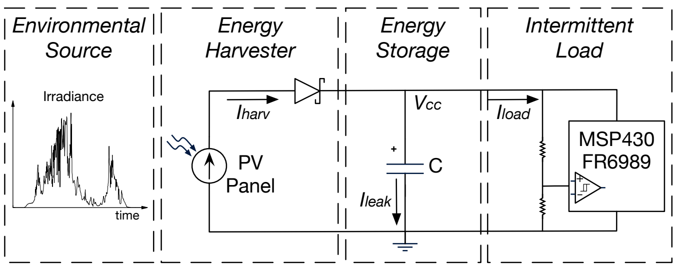


Fig. 10. System model of a PV-based ICS.

C. *Validation of Sizing Effects*

As previously shown in Fig. 1, the efficiently-sized energy storage capacitance (43 μF) improves forward progress by up to 55% and 30% compared to the minimum and decoupling capacitance respectively. We notice that this improvement becomes significant when the supply current attenuates because the save and restore overheads consume a larger proportion of the available energy. Also, this achieves at least 90% of the ideal forward progress mentioned in Section III-B. These results illustrate the importance of this technique, in particular for conditions where the supply current is low.

VI. Sizing under Real-World Energy Conditions

In this section, we model an ICS with a photovoltaic (PV) energy harvester to explore the energy storage sizing effect in real-world energy conditions, and demonstrate use of the proposed sizing approach.

A. *Simulation Configuration*

We integrate the validated reactive ICS model into a system model with a PV energy-harvesting supply as shown in Fig. 10. The energy storage model and the intermittent load model are as presented in Section V.

We use a converter-less supply circuit where only a Schottky diode is connected to the energy harvester output in order to prevent current backflow. The energy source conditions are imported from NREL outdoor solar irradiance data [32] and EnHANTs indoor irradiance data [33]. Four sets of light

TABLE IV
PV cell properties under a 1000 W/cm^2, AM-1.5 light source

Parameter	Value
Open-Circuit Voltage	0.89 V/cell
Short-Circuit Current	14.8 mA/cm^2
Maximum Power Voltage	0.65 V/cell
Maximum Power Current	12.1 mA/cm^2

conditions are used to encompass different energy environments. To convert irradiance into harvested power, we adopt a PV cell model [34] which uses the parameters available in common datasheets, so it can easily be reconfigured to suit various devices. We refer to Panasonic Amorton glass type solar cells [35] for PV cell properties as shown in Table IV. We set four cells in series (with V_{oc} = 3.56V) to match the operating voltage of the MCU (maximum 3.6V), and model energy harvester sizing by scaling the cell area.

B. *Exploration with Real-World Energy Source Conditions*

In real-world deployments, ambient energy source conditions are dependent on time and location. The energy harvester and storage need to be sized to achieve the desired forward progress across the range of expected conditions.

1) Sizing the Energy Harvester: For the purposes of this exploration, three levels of baseline mean forward progress (α_{exe}) are set as 0.1, 0.2, and 0.3. We use the system model to find the PV panel area that achieves the expected forward progress under the different energy source conditions with minimum energy storage. We scale the PV panel area to find that which achieves each baseline α_{exe}. As shown in Fig. 9, the energy harvester sizes that achieve the desired α_{exe} may span orders of magnitude given different energy source conditions from mm^2 for outdoor sources ((c) and (d)) to cm^2 for indoor sources ((a) and (b)).

2) Sizing the Energy Storage: Having obtained the energy harvester sizes for the baseline forward progress, we then use the modelling approach to size energy storage. We analyze the sizing effect of energy storage on forward progress given real-world energy conditions. Fig. 9 shows a 7.8-43.3% improvement in forward progress by sizing energy storage

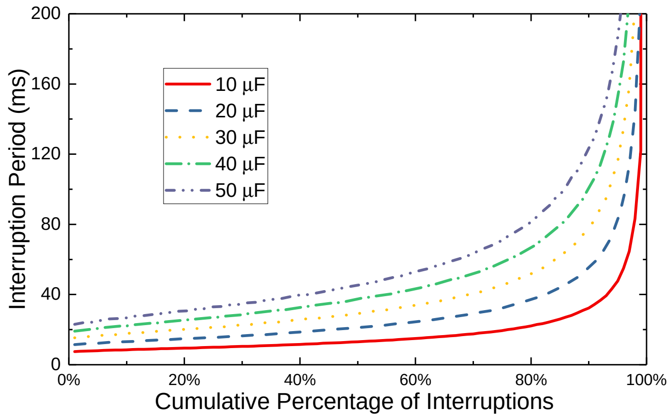


Fig. 11. Distribution of interruption periods.

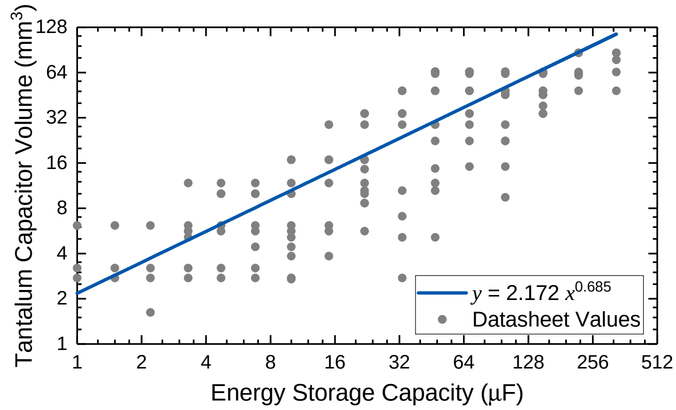


Fig. 12. Tantalum capacitor volume against capacitance for the six series of capacitors analyzed.

under the given real-world energy conditions and baseline energy harvester sizes. It can also be inferred that optimizing energy storage can either improve forward progress for a given energy harvester size, or reduce the energy harvester size that achieves the target forward progress. Given higher-power energy sources (e.g. Denver 2018 and Hawaii 2018 outdoor solar), increasing the harvester size efficiently improves forward progress with minor dimensional overheads, e.g. tens of mm^2; however, given lower-power sources (e.g. EnHANTs Setup A and Setup D indoor light), optimizing energy storage capacitance can save tens of cm^2 of PV panel area to achieve the same forward progress.

3) Interruption Period: Besides forward progress, we also explore how the capacitance can change the interruption periods. When interrupted by insufficient power supply, an ICS enters an interruption period where it saves its volatile state, waits for supply voltage to recover, and restores the state to resume execution, without making any forward progress. Applications that require frequent sensing may be negatively affected by long interruption periods. We measure an interruption period as *the period between two successive execution periods*, e.g. a consecutive 'SLR' period in Fig. 3 forms an interruption period. We record all the interruption periods during a one-year simulation with 10–50 µF capacitors, the Denver 2018 dataset, and an 80 mm^2 PV panel. Fig. 11 presents the distribution of all the interruption periods. With increased energy storage, the interruption period is prolonged. For example, the 90th percentile of interruption periods increases from 32.2 ms at 10 µF to 123.4 ms at 50 µF at an approximate rate of 23 ms per 10 µF. Facilitated by the simulator, developers are enabled to estimate whether the distribution of interruption periods meet their application requirement.

C. Trading Forward Progress, Dimensions, and Interruption Period

Although increasing energy storage capacitance improves forward progress, larger capacitance increases both dimensions and interruption periods. We evaluate the overheads of increased capacitor dimensions and interruption periods, and then trade them off against forward progress using a cost function to suggest an optimal capacitance value.

1) Metric of Dimensions: The overhead of capacitor dimensions is evaluated by characteristics of off-the-shelf tantalum capacitors. We narrow down the range of sample capacitors within a set of characteristics: low-profile, 10V rated voltage, and surface-mount package, and select six series of capacitors[2]. The volume and capacitance of these devices are plotted in Fig. 12. We use the regression of these data to approximate a capacitance-volume relationship.

2) Metric of Interruption Periods: Applications may have various requirements on interruption periods. To demonstrate the usage of our sizing approach, we consider a designer requests the 90th percentile of all interruption periods as an example metric of interruption periods, denoted as T_{int}. This metric indicates 90% of interruption periods are shorter than T_{int}. This metric can be adapted for particular application requirements.

3) Cost Function: From the previous observations (Fig. 7) we can see that achieving the optimal progress improvement costs much more capacitance (mean 3.2×) than to achieve 95% improvement. A trade-off is necessary to improve forward progress while restricting the overheads of increased capacitor volume and interruption periods. This involves a problem of multi-criteria decision making [36], which is outside the scope of this paper. Nevertheless, we provide a cost function in (13) as an example to illustrate how these three factors could be traded-off, but designers are expected to customize a cost function with parameters of importance to specific application requirements. Note that the function (13) is to be maximized to find the recommended capacitance.

$$f = \frac{\alpha_{exe}}{k_1} - \left(\frac{v_{cap}}{k_2}\right)^2 - \left(\frac{T_{int}}{k_3}\right)^2 \quad (13)$$

α_{exe} denotes normalized forward progress as defined in Section I, v_{cap} denotes capacitor volume, and T_{int} denotes application interruption periods as mentioned in Section VI-C2. α_{exe}, v_{cap}, and T_{int} can be generated from the simulation tool given C as an input. k_1, k_2, and k_3 are coefficients for normalizing each metric, and they are empirically determined

[2]The series of capacitor considered were: AVX TAJ, AVX TACmicrochip, AVX F92, Vishay 572D, Vishay 591D, and Vishay 592D.

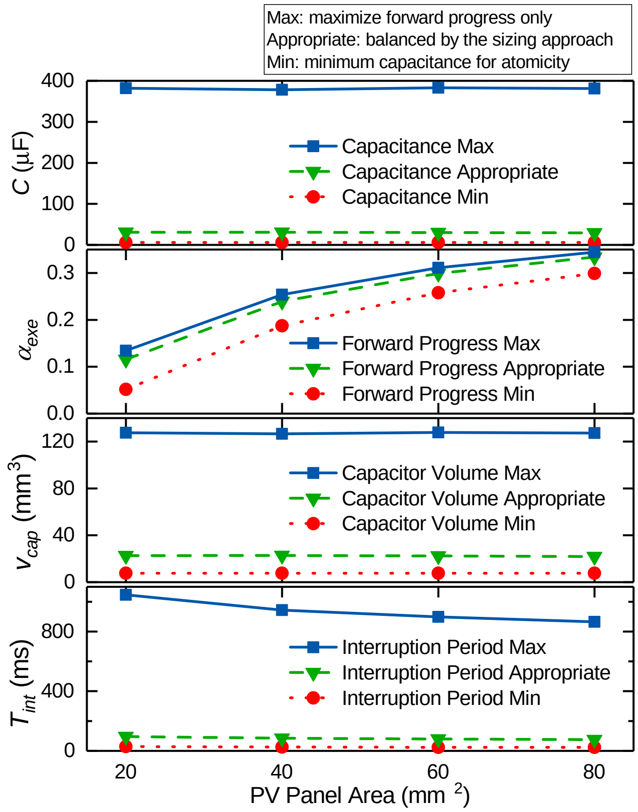


Fig. 13. The sizing approach trades-off forward progress, capacitor volume, and interruption periods. The results are plotted against a range of PV panel area, given Denver 2018 energy source dataset.

according to applications. In this example, the undesirable parameters are expressed as quadratic and negative terms to give an increasing cost to higher values. While only three parameters are considered here, others (such as the energy harvester size) could be included for a system-wise sizing scenario. As an example to demonstrate its usage, we arbitrarily configure the function by setting $k_1 = 0.2$, $k_2 = 200\,\text{mm}^3$, and $k_3 = 500\,\text{ms}$.

The effect of the trade-off is plotted in Fig. 13 using the Denver 2018 energy source dataset. Compared to the capacitor size that solely maximizes forward progress, on average, an appropriately-sized capacitor achieves 93% of the maximum forward progress, while saving 83% of capacitor volume and 91% of interruption periods. This also demonstrates the efficacy of the cost function and the chosen coefficients. Compared to the minimum storage case, the appropriately-sized capacitor improves forward progress by 12-124% with energy storage increased from 6.2 μF to 30 μF.

As shown in Fig. 12, the closest available capacitance that satisfies the 6.2 μF minimum capacitance is 6.8 μF, whereas the closest available capacitance to the appropriate 30 μF is 33 μF. The minimum volumes of 6.8 μF and 33 μF capacitors are both 2.75 mm^3, which means using the appropriate capacitance, instead of the minimum one, may not incur dimensional overhead. The regressed volume of the above two capacitance values are 8.1 mm^3 and 23.8 mm^3 respectively. However, the selection of capacitors can be dependent on factors other than physical volume, such as reliability, operation temperature, and more specific application needs. These factors can also be added into the cost function if necessary.

VII. Conclusions

While conventional ICSs have used minimal levels of capacitance, this paper has shown that increasing the amount of energy storage can improve forward progress by up to 65% with a constant current supply and 43% with real-world PV sources. The work includes a simulation tool which is available to download, enabling researchers to experiment with energy storage sizes to optimize ICS designs. A cost function can be incorporated, allowing various properties of the system to be traded-off. Our conclusion is that energy storage should be carefully designed, rather than minimized or indiscriminately picked, to efficiently operate ICSs. Future work will include a further investigation into cost functions for meeting multiple design objectives, and extensions to the simulation tool, e.g. models of additional energy storage devices and peripheral workloads.

References

[1] P. Sparks, "The route to a trillion devices," ARM Ltd., Tech. Rep., 2017.

[2] S. Chalasani and J. M. Conrad, "A survey of energy harvesting sources for embedded systems," in *IEEE SoutheastCon*, Apr. 2008, pp. 442–447.

[3] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, Sep. 2007.

[4] F. I. Simjee and P. H. Chou, "Efficient charging of supercapacitors for extended lifetime of wireless sensor nodes," *IEEE Trans. Power Electron.*, vol. 23, no. 3, pp. 1526–1536, May 2008.

[5] G. Liu, Y. Yu, J. Hou, W. Xue, X. Liu, Y. Liu, W. Wang, A. Alsaedi, T. Hayat, and Z. Liu, "An ecological risk assessment of heavy metal pollution of the agricultural ecosystem near a lead-acid battery factory," *Ecological Indicators*, vol. 47, pp. 210 – 218, 2014.

[6] F. Akhtar and M. H. Rehmani, "Energy replenishment using renewable and traditional energy resources for sustainable wireless sensor networks: A review," *Renewable Sustainable Energy Rev.*, vol. 45, pp. 769 – 784, 2015.

[7] S. T. Sliper, O. Cetinkaya, A. S. Weddell, B. Al-Hashimi, and G. V. Merrett, "Energy-driven computing," *Philosophical Transactions of the Royal Society A*, vol. 378, no. 2164, 2020.

[8] B. Ransford, J. Sorber, and K. Fu, "Mementos: System support for long-running computation on RFID-scale devices," in *Proc. ASPLOS XVI*, Newport Beach, CA, USA, Mar. 2011, pp. 159–170.

[9] N. A. Bhatti and L. Mottola, "HarvOS: Efficient code instrumentation for transiently-powered embedded sensing," in *Proc. 16th IPSN*, Pittsburgh, PA, USA, Apr. 2017, pp. 209–220.

[10] A. Y. Majid, C. D. Donne, K. Maeng, A. Colin, K. S. Yildirim, B. Lucia, and P. Pawełczak, "Dynamic task-based intermittent execution for energy-harvesting devices," *ACM Trans. Sen. Netw.*, vol. 16, no. 1, Feb. 2020.

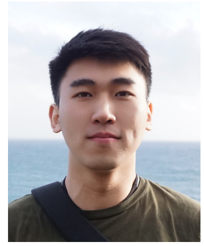
[11] K. Maeng, A. Colin, and B. Lucia, "Alpaca: Intermittent execution without checkpoints," *Proc. ACM Program. Lang.*, vol. 1, OOPSLA, Oct. 2017.

[12] 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 Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 12, pp. 1968–1980, 2016.

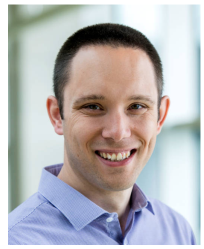
[13] M. Zhao, C. Fu, Z. Li, Q. Li, M. Xie, Y. Liu, J. Hu, Z. Jia, and C. J. Xue, "Stack-size sensitive on-chip memory backup for self-powered nonvolatile processors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 11, pp. 1804–1816, 2017.

[14] Y. Liu, J. Yue, H. Li, Q. Zhao, M. Zhao, C. J. Xue, G. Sun, M. Chang, and H. Yang, "Data backup optimization for nonvolatile sram in energy harvesting sensor nodes," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 10, pp. 1660–1673, 2017.

[15] K. Maeng and B. Lucia, "Supporting peripherals in intermittent systems with just-in-time checkpoints," in *Proc. 40th PLDI*, Phoenix, AZ, USA, 2019, pp. 1101–1116.
[16] K. Ma, X. Li, K. Swaminathan, Y. Zheng, S. Li, Y. Liu, Y. Xie, J. J. Sampson, and V. Narayanan, "Nonvolatile processor architectures: Efficient, reliable progress with unstable power," *IEEE Micro*, vol. 36, no. 3, pp. 72–83, May 2016.
[17] H. Jayakumar, A. Raha, W. S. Lee, and V. Raghunathan, "Quickrecall: A HW/SW approach for computing across power cycles in transiently powered computers," *J. Emerg. Technol. Comput. Syst.*, vol. 12, no. 1, Aug. 2015.
[18] J. Hester, L. Sitanayah, and J. Sorber, "Tragedy of the coulombs: Federating energy storage for tiny, intermittently-powered sensors," in *Proc. 13th SenSys*, Seoul, South Korea, 2015, p. 5–16.
[19] D. Balsamo, B. J. Fletcher, A. S. Weddell, G. Karatziolas, B. M. Al-Hashimi, and G. V. Merrett, "Momentum: Power-neutral performance scaling with intrinsic mppt for energy harvesting computing systems," *ACM Trans. Embed. Comput. Syst.*, vol. 17, no. 6, Jan. 2019.
[20] K. Maeng and B. Lucia, "Adaptive dynamic checkpointing for safe efficient intermittent computing," in *Proc. 13th OSDI*, Carlsbad, CA, Oct. 2018, pp. 129–144.
[21] F. Su, Y. Liu, X. Sheng, H. G. Lee, N. Chang, and H. Yang, "A task failure rate aware dual-channel solar power system for nonvolatile sensor nodes," *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 4, Jul. 2019.
[22] N. Jackson, J. Adkins, and P. Dutta, "Capacity over capacitance for reliable energy harvesting sensors," in *Proc. 18th IPSN*, Montreal, Quebec, Canada, Apr. 2019, pp. 193–204.
[23] Y. Gu, Y. Liu, Y. Wang, H. Li, and H. Yang, "NVPsim: A simulator for architecture explorations of nonvolatile processors," in *Proc. 21st ASP-DAC*, Jan. 2016, pp. 147–152.
[24] S. T. Sliper, W. Wang, N. Nikoleris, A. Weddell, and G. Merrett, "Fused: closed-loop performance and energy simulation of embedded systems," in *Proc. ISPASS*, Boston, MA, USA, 2020.
[25] J. San Miguel, K. Ganesan, M. Badr, C. Xia, R. Li, H. Hsiao, and N. Enright Jerger, "The EH model: Early design space exploration of intermittent processor architectures," in *Proc. 51st IEEE/ACM MICRO*, Fukuoka, Japan, Oct 2018, pp. 600–612.
[26] J. Hester, T. Scott, and J. Sorber, "Ekho: Realistic and repeatable experimentation for tiny energy-harvesting sensors," in *Proc. 12th SenSys*, Memphis, Tennessee, 2014, p. 1–15.
[27] K. Geissdoerfer, M. Chwalisz, and M. Zimmerling, "Shepherd: A portable testbed for the batteryless iot," in *Proc. 17th SenSys*, New York, NY, USA, 2019, p. 83–95.
[28] R. Faltus, "Low leakage current aspect of designing with tantalum and niobium oxide capacitors," AVX, Tech. Rep., 2012.
[29] 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 Syst. Lett.*, vol. 7, no. 1, pp. 15–18, Mar. 2015.
[30] A. Rodriguez Arreola, D. Balsamo, G. V. Merrett, and A. S. Weddell, "Restop: Retaining external peripheral state in intermittently-powered sensor systems," *Sensors*, vol. 18, no. 1, p. 172, 2018.
[31] S. T. Sliper, D. Balsamo, N. Nikoleris, W. Wang, A. S. Weddell, and G. V. Merrett, "Efficient state retention through paged memory management for reactive transient computing," in *Proc. 56th DAC*, Jun. 2019, pp. 26:1–26:6.
[32] A. Andreas and T. Stoffel, "NREL solar radiation research laboratory (SRRL): Baseline measurement system (BMS); Golden, Colorado (Data)," *NREL Report NO.DA-5500-56488*, 1981.
[33] M. Gorlatova, A. Wallwater, and G. Zussman, "Networking low-power energy harvesting devices: Measurements and algorithms," *IEEE Trans. Mobile Comput.*, vol. 12, no. 9, pp. 1853–1865, Sep. 2013.
[34] S. Vergura, "A complete and simplified datasheet-based model of PV cells in variable environmental conditions for circuit simulation," *Energies*, vol. 9, no. 5, 2016.
[35] *Panasonic Amorton Solar Cells*, Accessed: Jul. 2020. [Online]. Available: https://www.panasonic-electric-works.com/cps/rde/xbcr/pew_eu_en/ca_amorton_solar_cells_en.pdf
[36] E. Triantaphyllou, *Multi-criteria decision making methods: A comparative study*. Springer, 2000.



Jie Zhan received the B.Eng. degree in electrical engineering from Xiamen University, China, in 2017. He is currently pursuing a Ph.D. in electronic engineering with the School of Electronics and Computer Science, University of Southampton, U.K. His research is focused on energy-harvesting intermittently-powered systems for batteryless devices and internet of things applications.



Geoff V. Merrett (GSM'06-M'09-SM'19) received the B.Eng. (Hons) and Ph.D. degrees from the University of Southampton, U.K., in 2004 and 2008, respectively. He currently holds a Personal Chair in Electronic and Software Systems at the School of Electronics and Computer Science, University of Southampton, U.K., where he is Head of the Centre for Internet of Things and Pervasive Systems.

Prior to becoming a Professor in 2019, he was a Lecturer and Associate Professor at Southampton from 2009 and 2014 respectively. He is Co-Director of the Arm-ECS Research Centre, an award winning collaboration between the university and Arm Research, Cambridge. His current research interests are in energy management of mobile/embedded systems and self-powered devices, and he has published over 200 journal and conference articles on these topics.

Alex S. Weddell (GSM'06-M'10) received the M.Eng. degree (1st class honors) and Ph.D. in electronic engineering from the University of Southampton, U.K., in 2005 and 2010. His main research focus is in the areas of energy harvesting and energy management for future Internet of Things devices. He has over 15 years' experience in energy harvesting systems, and has published over 65 papers in the area. He is now a Lecturer at the University of Southampton, involved with three projects funded by EPSRC, EU Horizon 2020 and Clean Sky 2.