

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

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

increases recharging time and physical dimensions. define application

Abstract—Batteryless energy-harvesting devices are promising for a sustainable IoT. Forward progress of application execution in such devices is maintained by intermittent computing, where volatile computing state is saved into, and restored from, non-volatile memory before and after power interruptions. Current intermittent computing approaches typically minimise energy storage to reduce device dimensions and recharging time. However, this results in energy-expensive state-saving and -restoring overheads and impedes forward progress, whereas increasing energy storage requires more space and longer recharging time. In this paper, we propose a model-based approach for identifying an appropriate energy storage capacity which trades off forward progress, dimensions, and recharging time. We develop an intermittent computing model that accurately estimates forward progress, with an experimentally validated mean error of only 0.5 %. Using this model, we show that appropriately sizing energy storage can improve forward progress by up to 65 % with constant current supply and 43 % with real-world solar energy. Finally, we demonstrate the sizing approach, which achieves 98.3 % of the maximum forward progress while reasonably saves capacitor volume (71.7 %) and recharging time (83.8 %) from the maximised case.

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

I. INTRODUCTION

INTERNET of Things (IoT) devices are becoming ubiquitous, with tens of billions of devices to be installed, many of which will be in potentially hard-to-reach locations [1]–[3]. Conventionally, these devices are battery-powered, and thus have constrained lifespans, necessitating impractical replacement. To circumvent the limited battery lifespan, energy-harvesting IoT devices become a solution.

Environmentally harvested power is intrinsically variable and intermittent [4]. Traditionally, large energy storage devices in the form of rechargeable batteries or supercapacitors are adopted to buffer the variability and provide a continuous supply of power that is free from interruptions [5]–[10]. Unfortunately, such types of energy storage raise pollution concerns [11], increase cost and device dimensions [12], and still have limited lifespans [12], [13].

Recently, *intermittent computing systems* (ICSs) have been proposed to remove large energy storage devices while maintaining execution across frequent power interruptions [14]–[19]. ICSs save volatile system computing state, e.g. CPU registers and RAM contents, into nonvolatile memory (NVM) either at pre-installed points (*static methods*) or when the supply is about to fail (*reactive methods*) [20], [21]. In

The authors are with School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK.

Software of this work will be open-source upon acceptance of this paper.

ICSs, *forward progress* denotes the effective program progress, excluding re-executed progress, lost progress, and state-saving and -restoring operations [22], [23]. The amount of forward progress directly determines application performance, e.g. program iteration rate or task completion time. In this paper, we describe normalised forward progress as the ratio of the effective execution time to the total time, without being restricted to a specific workload.

Current intermittent computing approaches typically adopt only a minimum amount of energy storage, which is just enough for the most energy-expensive atomic operation¹, with the goal of minimising device dimensions and recharging time [15], [16], [24]–[27]. However, in this paper we show that a system with minimum energy storage has to frequently go through the cycle of wake up, restore state, execute program, save state, and halt, thereby causing high overheads in time and energy. We propose that provisioning more energy storage can prolong the operating cycles and reduce the frequency of interruptions, hence reducing overheads and improving forward progress. However, larger energy storage also increases leakage current, occupies greater volume, and requires longer recharging time. Such a relationship between energy storage capacity and forward progress remains unexplored in ICSs, and the challenge of sizing energy storage to improve forward progress while moderating the physical size and recharging time is unaddressed.

In this paper, we present a model-based approach for sizing energy storage in ICSs, which quantifies and trades off forward progress, capacitor volume, and recharging time. In summary, the main contributions of this paper are:

- A model-based sizing approach that identifies an appropriate energy storage capacity in ICSs (Section III), which is demonstrated with real-world energy conditions to achieve 98.3 % of the maximum forward progress while saves 71.7 % capacitor volume and 83.8 % recharging time from the maximised case (Section VII).
- A model of reactive intermittent computing to accurately estimate forward progress (Section IV), which is experimentally validated with 0.5 % mean absolute percentage error (Section VI). This shows a
- An exploration based on the model, where we analyse the energy storage sizing effect on forward progress with respect to supply current and volatile state size, showing up to 64.9 % forward progress improvement (Section V).

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

Goes 1, 2, 1... merge 1st & 2nd... and we argue that

adding a relatively small amount of additional energy storage can significantly improve FP?

Something about
how hard to
size / spec ICS?

II. BACKGROUND AND RELATED WORK

A. Intermittent Computing *Approaches to ICS?*

Intermittent computing aims at preserving forward progress across frequent power interruptions, such that the device can operate even directly by energy-harvesting power. The basic mechanism is to back up volatile computing state (lost if the supply fails) as nonvolatile state (preserved if the supply fails) during active periods; when the supply recovers, the lost volatile state is restored from the saved nonvolatile state to resume execution. Approaches in intermittent computing can be classified as *static* and *reactive*. *ICS?*

1) *Static Intermittent Computing*: Static approaches save volatile state into NVM at design-time or compile-time ~~re-ins~~ points, either by inserting checkpoints or decomposing a program into atomic tasks. After power interruptions, the progress rolls back and resumes from the last saved checkpoint or task boundary. Advantages of static approaches include minimising hardware dependency and ensuring operation atomicity. However, the rollback of progress intrinsically introduces violation of memory consistency due to code re-execution, and wastes energy on lost and re-executed progress. Also, if the consumption between two successive checkpoints or task boundaries exceeds the amount that the energy storage can guarantee, the progress may never proceed due to insufficient power input. *ICS?* *only*

2) *Reactive Intermittent Computing*: In contrast to static approaches, reactive approaches save volatile state in NVM when supply is about to fail by monitoring supply voltage. Specifically, reactive intermittent computing saves volatile state and enters a low-power mode (execution halted) if supply voltage falls below a save threshold. This save threshold should be set high enough to successfully save volatile state before power fails. By entering the low-power mode, reactive approaches avoid re-execution, and hence typically make more forward progress than static approaches. In a comparison between static [28] and reactive [15] approaches, the reactive approach demonstrates a $2.5 \times$ mean speedup on computational workloads [29]. Therefore, we focus on reactive intermittent computing for modelling and validation in this paper. *peripherals?*

B. Modelling Intermittent Computing *ICS*

A few models have been proposed for exploring design space in intermittent computing. Su et al. [30] provide a model to explore the impact of sizing energy harvester and energy storage on a dual-channel solar-powered nonvolatile sensor node, but their exploring range of energy storage is $10\text{--}10^4 \text{ F}$ supercapacitors, which is significantly larger than a typical scale of energy storage (μF to mF scale) in intermittent computing. EH model [31], [32] explores forward progress with different parameter settings, but the model only considers a single active period without considering inactive periods that scale down the effective forward progress. Jackson et al. [33] also present a numerical model to explore energy storage effect, where they suggest that novel batteries are preferable in order to offer improve the completion rate of scheduled tasks. However, the model lacks a detailed state-saving and -restoring mechanism in intermittent computing, *ICS*.

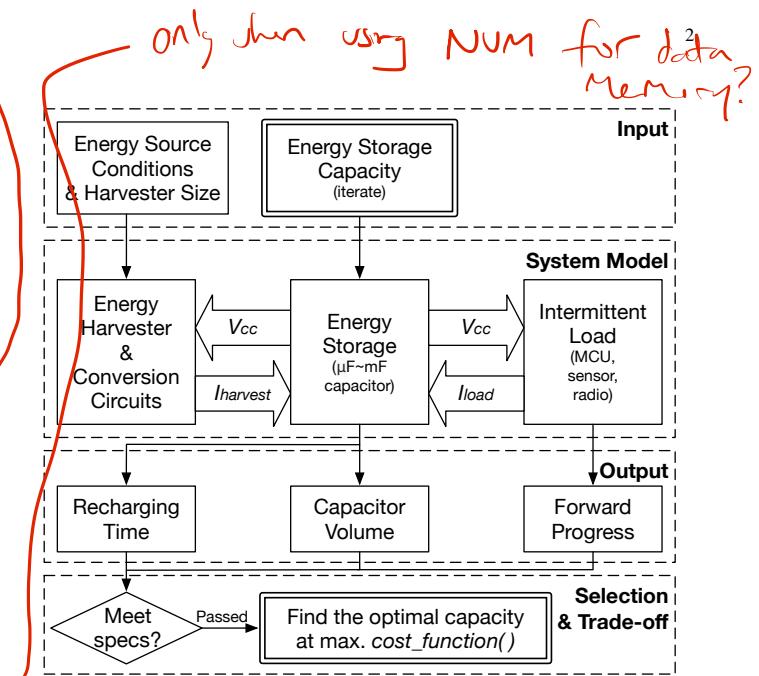


Fig. 1. Overview of the proposed sizing approach and the system model structure.

Furthermore,
III. A. *Compos*
but all work
refers.
III. B. *Expans*
from your design
tools for non-IC;
Chartres; Exico;
Exico; U; rea
only; since PP
in ICS.

III. ENERGY STORAGE SIZING APPROACH

The proposed sizing approach aims at identifying an appropriate energy storage capacity that improves forward progress and moderates the side effects, i.e. increased capacitor volume and recharging time.

*written as "this is what I've done,"
"I have this 'thing', rather than party/
crying on cracks."*

A. Overview

Essentially, the sizing approach inputs a range of energy storage capacities into an ICS model to output forward progress, capacitor volume, and recharging time, and trades off them in a cost function to obtain the optimal energy storage capacity. This approach is summarised in Fig. 1 with details explained as follows:

- Input:** A time trace of energy source conditions in the deployed location should be obtained and imported. Assuming the energy source is equally distributed in the deployed space, the energy harvester size can be optionally changed to explore different harvested power scales.
- System model and Output:** The sizing approach relies on a system model, which enables design exploration by changing parameters, e.g. energy storage and energy harvester sizes, and observing the effect on forward progress, capacitor volume, and recharging time. We provide a system model as further explained in Section III-B.
- Selection and Trade-off:** The outputs are filtered to obtain the energy storage capacities that meet design specifications, e.g. maximum volume. The appropriate capacity is then suggested by evaluating forward progress

*Could be
constant
current
or variable
though?*

*So not for ICS? Is it for EN systems?
Not strong argument considering words are the same as your wish..*

and the side effects with a cost function. We provide a cost function in Equation (1) as an example:

$$f = \frac{\alpha_{exe}}{k_1} - \left(\frac{v_{cap}}{k_2} \right)^2 - \left(\frac{T_{recharge}}{k_3} \right)^2 \quad (1)$$

where v_{cap} represents capacitor volume and $T_{recharge}$ represents recharging time. k_1 , k_2 , and k_3 are linear scalers, which are empirically determined according to applications. The negative side effects are calculated in quadratic forms so as to punish high values more heavily. We only consider the above three common factors in sizing energy storage, but other particular factors can also be included if necessary.

such as?

B. System Model

The system model is shown as a part in Fig. 1. The system model contains three modules, i.e. *Energy Harvester and Conversion Circuits*, *Energy Storage*, and *Intermittent Load*. The three modules communicate by their voltage and current flows. Due to the variety in each module, the three module should be individually specified to represent the target platform according to the techniques implemented.

1) *Energy Harvester and Conversion Circuits*: The energy harvester module transduces environmental energy into electric power. Typically, the power harvested from the energy harvester should be conditioned to provide a suitable voltage for charging the energy storage and supplying the load efficiently. However, in intermittent computing, such conversion circuits may be removed to increase power efficiency, using only a diode to prevent current backflow. The energy harvester and conversion circuits can be modelled together as a module because they are usually coupled and integrated.

2) *Energy Storage*: Energy storage in ICSs is usually in the form of a μ F- to mF-scale capacitor. The energy storage provides the minimum length of an execution period, which should be enough to complete the most energy-expensive atomic operation.

3) *Intermittent Load*: The load module includes all the power consumers in an ICS, such as a microcontroller, sensors, and a radio. As mentioned, intermittent computing approaches can be classified as *static* and *reactive*. These two types of approaches fundamentally differs in how the load consumes power and makes forward progress, and hence, require different models for estimating forward progress.

III. a bit poor at present. Loses momentum. Needs work.

IV. MODELLING REACTIVE INTERMITTENT COMPUTING

We present a model of reactive intermittent computing in respect of forward progress to facilitate understanding and exploration of ICSs. This model accounts for *Energy Storage* and *Intermittent Load* modules in the system model. Parameters of this model are listed in Table I. The model outputs α_{exe} , i.e. the mean forward progress, with a constant current supply I_{harv} . In the model we assume that all the configuration parameters maintain constant mean values.

For equation brevity, we use I_{in} to denote usable current input as expressed in Equation (2), with the capacitor leakage effect to be discussed at the end of this section.

$$I_{in} = I_{harv} - I_{leak} \quad (2)$$

does this capture power continuity?

No, not always true

You said real time varying date in III?

TABLE I
MODEL PARAMETERS OF REACTIVE INTERMITTENT COMPUTING

Input Parameters	
I_{harv}	Energy harvester current supply
C	Energy storage capacitance
Configuration Parameters	
I_{exe}	Execution current consumption
I_{sleep}	Sleep current consumption
I_r	Restore current consumption
I_s	Save current consumption
I_{leak}	Leakage current consumption
V_r	Restore voltage threshold
V_s	Save voltage threshold
T_r	Restore time overhead
T_s	Save time overhead
Output Parameter	
α_{exe}	Effective execution time ratio (forward progress)

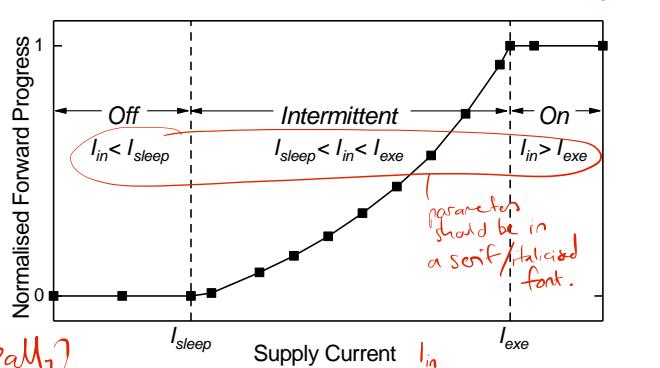


Fig. 2. Operating modes of reactive intermittent computing according to supply current and forward progress.

A. Operating Modes of Reactive Intermittent Computing

The behaviours of reactive intermittent computing can be classified as three operating modes given a spectrum of supply current, as shown in Fig. 2. An example experiment demonstrates the three operating modes. These three modes are divided by the relationship between the current input and current consumption. We denote the three modes as *Off*, *Intermittent*, and *On*.

Off mode: When $I_{in} < I_{sleep}$, the system stays inactive. The supply voltage V_{cc} cannot be charged up to the restore threshold V_r to wake up the system and start execution. The sleep current I_{sleep} denotes the current consumption when the system waits for supply voltage to recover above the restore threshold, and hence, includes the consumption of voltage monitoring circuits and system idle current.

Intermittent mode: When $I_{sleep} < I_{in} < I_{exe}$, the system executes when supply voltage is available intermittently. V_{cc} can be charged up to V_r and the system starts execution. However, the energy in energy storage is then consumed by the load as $I_{in} < I_{exe}$, causing V_{cc} to drop below the save threshold V_s , where the system saves its state and sleeps. The system sleeps until V_{cc} charges to V_r again and then resumes execution. In general, higher I_{in} leads to more forward progress in this mode, but the exact relationship between I_{in}

I'd remove table, & instead make sure explicit explain how or for what use.

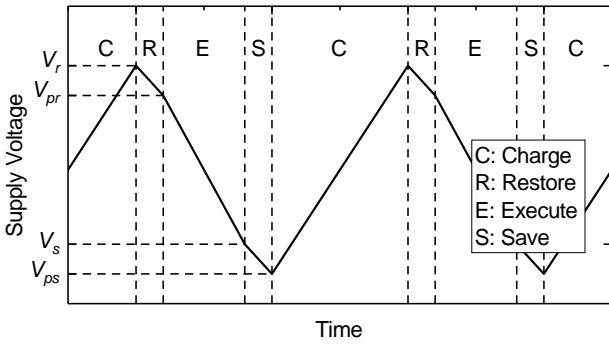


Fig. 3. Operating cycles in the Intermittent mode.

and forward progress requires further analysis.

On mode: When $I_{in} > I_{exe}$, the system executes constantly as the supply voltage V_{cc} never drops. The excess power either dissipates through circuits or overcharges V_{cc} . An overcharged V_{cc} may affect harvesting efficiency due to poor impedance matching and reduce I_{harv} , such that current input and consumption are in equilibrium.

B. Formulating Forward Progress

We provide formulations to calculate forward progress in relation to energy storage capacity and current input through a theoretical analysis. In the following analysis, we focus on the Intermittent mode ($I_{sleep} < I_{in} < I_{exe}$), as the forward progress of the On mode and the Off mode are straightforward (1 in the On mode, 0 in the Off mode).

In the Intermittent mode, the system goes through four intervals in turn, i.e. charging, restoring, executing, and saving as shown in Fig. 3, with current consumption of I_{sleep} , I_r , I_{exe} , and I_s in each interval respectively.

Let V_{pr} (post-restore) and V_{ps} (post-save) denote the voltage after restoring and saving operations. 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 Equation (3), the time spent on useful execution T_{exe} in one operating cycle can be expressed as:

$$\begin{aligned} T_{exe} &= \frac{C(V_{pr} - V_s)}{I_{exe} - I_{in}} \\ &= \frac{C(V_r - V_s) + T_r(I_{in} - I_r)}{I_{exe} - I_{in}} \end{aligned} \quad (5)$$

In Equation (5), $C(V_r - V_s)$ represents the amount of energy in the storage capacitor for restoring and executing. $T_r(I_{in} - I_r)$ represents the restoring cost. $I_{exe} - I_{in}$ is the current consumption for execution.

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

$$\begin{aligned} T_{charge} &= \frac{C(V_r - V_{ps})}{I_{in} - I_{sleep}} \\ &= \frac{C(V_r - V_s) - T_s(I_{in} - I_s)}{I_{in} - I_{sleep}} \end{aligned} \quad (6)$$

*This section is stronger, but.. so short.
Any general analysis of (6)? figure to illustrate more about Intermittent.

Then, with Equation (5) and (6), the period of an operating cycle is:

$$\begin{aligned} T_{period} &= T_{charge} + T_r + T_{exe} + T_s \\ &= \frac{C(V_r - V_s) + T_s(I_{in} - I_s)}{I_{in} - I_{sleep}} + \\ &\quad \frac{C(V_r - V_s) + T_r(I_{exe} - I_r)}{I_{exe} - I_{in}} \end{aligned} \quad (7)$$

Finally, with T_{exe} and T_{period} calculated in Equation (5) and (7), we obtain the percentage of time spent on effective execution (forward progress) in the Intermittent mode:

$$\alpha_{exe} = \frac{T_{exe}}{T_{period}}, \quad I_{sleep} < I_{in} < I_{exe} \quad (8)$$

As $d\alpha_{exe}/dI_{in}$ is positive, higher harvested current I_{harv} leads to more forward progress. To find out the energy storage effect on α_{exe} , we need to analyze $d\alpha_{exe}/dC$. Here, if we assume the leakage current maintains constant when storage capacity increases, α_{exe} keeps increasing with C to approach $(I_{in} - I_{sleep})/(I_{exe} - I_{sleep})$, which is an ideal case where restore and save overheads are zero. However, in an electrolytic capacitor, the leakage current typically increases with storage capacity C in the following relationship:

$$I_{leak} = kCV_{cc} \quad (9)$$

where k is a constant normally in a range from 0.01 to 0.03 ($\frac{A}{F.V}$). Therefore, referring to Equation (2), dI_{in}/dC is $-kV_{cc}$, which is negative. Hence, considering capacitor leakage increases with capacitance, there is an optimal capacity that leads to the maximum forward progress α_{exe} .

To include the Off and On modes, the forward progress given all supply levels is presented as:

$$\alpha_{exe} = \begin{cases} 0 & , \text{Off } (I_{in} < I_{sleep}) \\ \frac{T_{exe}}{T_{period}} & , \text{Intermittent } (I_{sleep} < I_{in} < I_{exe}) \\ 1 & , \text{On } (I_{in} > I_{exe}) \end{cases} \quad (10)$$

where T_{exe} and T_{period} are calculated by Equation (5) and (7) respectively.

on 10/ March.

V. EXPLORATION OF ENERGY STORAGE SIZING

In this section, we configure the reactive intermittent computing model, and then present an exploration of the relationship between forward progress and energy storage capacity with respect to current input and volatile state size.

A. Model Configuration

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

$$\frac{dV_{cc}}{dt} = I_{harv} - I_{load} - I_{leak} \quad (11)$$

where I_{load} is the current consumption of the load. The leakage current I_{leak} in electrolytic capacitors demonstrates

TABLE II
PROFILED MCU PARAMETERS

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

complex physical phenomenon. In the exploration, we refer to an off-the-shelf tantalum capacitor [34], with an empirical I_{leak} in relation to capacitance C , rated voltage V_{rated} , and terminal voltage V_{cc} [35]:

$$I_{leak} = 0.01\lambda CV_{rated} \quad (A) \quad (12)$$

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:

$$\lambda = 0.05 \times 20^{\frac{V_{cc}}{V_{rated}}} \quad (13)$$

V_{rated} is chosen to be 10 V so as to operate the load (typically < 4.0 V) at 25–40 % of the rated voltage for low leakage design [35].

2) *Intermittent Load*: We implemented and parameterized a reactive intermittent computing approach [16] on a TI MSP430FR6989 microcontroller. The parameters are profiled with the MCU running Dijkstra path finding algorithm with 1696 B RAM usage at 8 MHz. The supply voltage monitoring circuits use the MCU internal comparator and an external 3 M Ω 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 is $V_{off} = 1.8$ V.

The profiled parameters are shown in Table II. We measured the current consumption with a range of supply voltage. In experimental measurements, the variation of I_{sleep} between V_{off} and V_r is 2 %, and the variation of I_{exe} between V_s 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 such minor variations and use the mean of I_{exe} and I_{sleep} 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 is 6.2 μF to guarantee save and restore operations.

Although we define the parameters in the model, however, these parameters can be changed to represent different load characteristics. For example, T_r and T_s can be tuned to model different volatile state sizes.

B. Sizing Energy Storage to Improve Forward Progress

1) *Impact of Supply Current*: Increasing energy storage capacity improves forward progress by reducing the frequency of power interruptions compared to the minimum energy storage. In Fig. 4, the relationship between forward progress and energy storage capacity is plotted given a range of constant current supply. Forward progress increases as energy storage capacity increases beyond the minimum level (6.2 μF

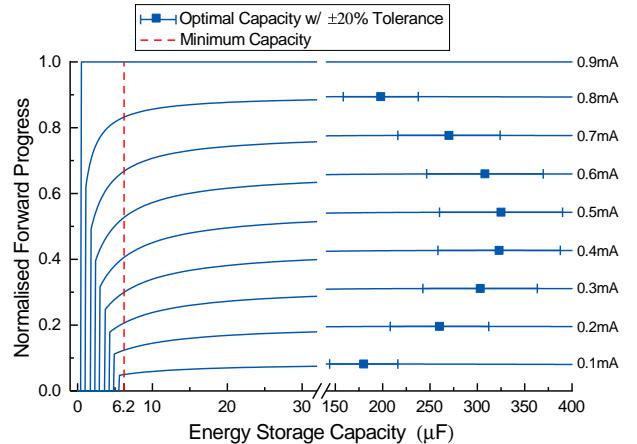


Fig. 4. Forward progress against energy storage capacity at different levels of constant supply current.

in this case). However, this improvement is offset by increased capacitor leakage, which leads to an optimal storage capacity.

When capacity is less than the minimum (in Fig. 4, the area on the left of the dashed line), the execution may still progress given that the current supply keeps providing energy during execution. However, if the restore or save operation cannot be completed even with the energy input during execution, the forward progress directly falls to zero as the volatile state cannot be preserved successfully. This steep fall is because the implemented control algorithm enters the low-power mode with volatile state retained after a save operation, and hence, the energy used for restoring state in the first operating cycle is then used for effective execution in the following operating cycles as long as the supply voltage recovers to the restore threshold without a power interruption. Typically, capacitors in manufacture has ± 20 % tolerance of capacitance. This effect of capacitance variations on forward progress is also plotted in Fig. 4, where it is shown to be trivial (< 0.23 %).

As a further illustration shown in Fig. 5, the maximum forward progress improvement can achieve up to 64.9 % compared to using the minimum energy storage. Correspondingly, the optimal energy storage capacity is also plotted against supply current. It may not be preferable to optimise the energy storage size solely for maximising forward progress because side effects exist in increasing capacity, e.g. recharging time and dimensions (later explained in Section VII-C). As shown that the progress changes trivially around the optimum, we also plot in Fig. 5 the storage capacity that achieves 95 % of the maximum improvement, observing a significant capacity reduction, e.g. from 325 μF to 90 μF at 0.5 mA supply and 68.8 % mean capacity reduction for all supply levels.

2) *Impact of Volatile State Size*: The size of volatile state differs across applications with different amounts of RAM usage, and hence, incurs application-specific time and energy overheads on restore and save operations. Although the overheads are profiled for one application, the model can be tuned to explore various volatile state sizes. We modelled the volatile state size variations by linearly scaling restore and save time overheads because they are experimented to be linear [20]. We measured time overheads of restore and save operations

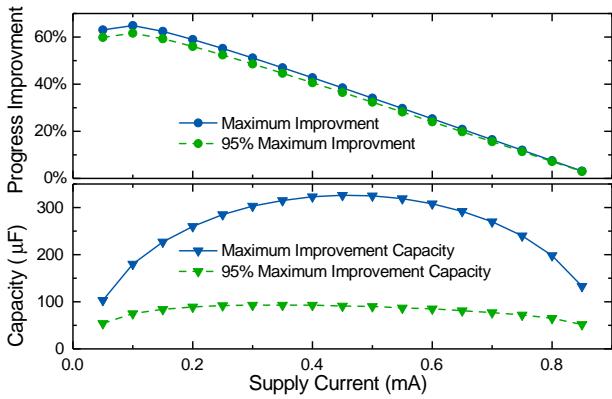


Fig. 5. Maximum forward progress improvement by sizing energy storage given a spectrum of supply current (normalised by the minimum capacity case), with the corresponding maximum and sub-maximum (95 % of maximum) capacitance.

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

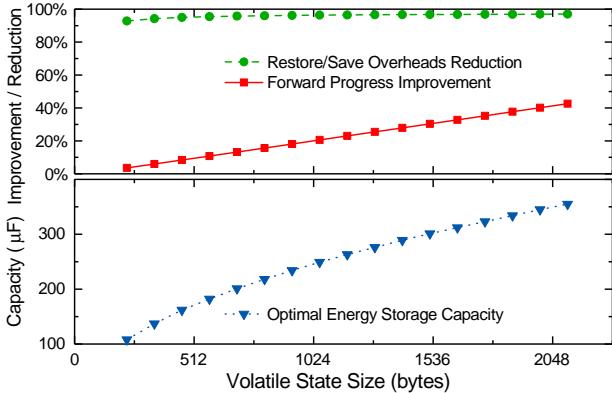


Fig. 6. Impact of RAM usage (linear to restore/save overheads) on sizing energy storage with 0.4 mA current supply. Improvement and reduction are normalised by the minimum capacity case.

in the minimum case (64B register data and a 160B stack) and the maximum case (64B register data and 2048B full RAM) respectively as shown in Table III.

An example of this exploration on volatile state size is plotted in Fig. 6 with 0.4 mA supply current. The forward progress improvement by sizing energy storage increases with the volatile state size, and the optimal capacity 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 left end point in Fig. 6), the maximum progress improvement is only 3.6 % although the restore and save overheads are reduced by 92.9 %.

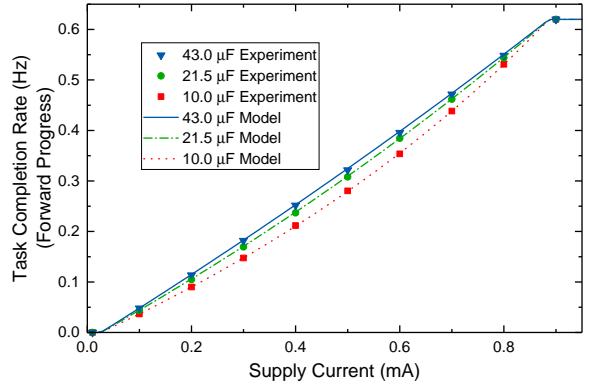


Fig. 7. Model validation with the experimented and modelled task completion rates (i.e. forward progress).

VI. MODEL VALIDATION

This section shows the comparison between the experimental and modelled forward progress to validate the proposed reactive intermittent computing model and the exploration results.

A. Experiment Setup

With the identical platform properties and workload settings in the model configuration (Section V-A2), we used the TI MSP430FR6989 development board as the experiment platform. The on-board capacitance is measured to be 10.0 μF . This is the minimum capacitance that can be tested with the platform. Additional capacitors were added to provide extra energy storage. The leakage current of these capacitors were measured to be less than 10 nA at 3 V, which is negligible compared to the current consumption of the platform (μA current draw), and hence, we omit the capacitor leakage in the following experiment and model output.

In this practical experiment, the task completion rate (i.e. tasks completed per second) is used as the metric of forward progress rather than the effective execution time ratio. To gain the task completion rate from the model, we multiply the normalised forward progress (execution time ratio) generated from our model by the completion rate when the system executes constantly since the task completion rate is linear to the effective execution time.

B. Results

1) *Model Accuracy:* To validate the accuracy of our model, we powered the device with a range of supply currents to operate the device in the Intermittent mode (0.1–0.8 mA, 0.1 mA per step), and repeated the tests with three energy storage capacities: a) on-board 10.0 μF capacitance only; b) 21.5 μF measured in total (11.5 μF added); c) 43.0 μF measured in total (33.0 μF added). We compared the actual forward progress with the one that our model generated. As shown in Fig. 7, the model generated output matches closely with the experimental results with only 0.5 % mean absolute percentage error. Hence, the model is able to accurately estimate forward progress for design exploration.

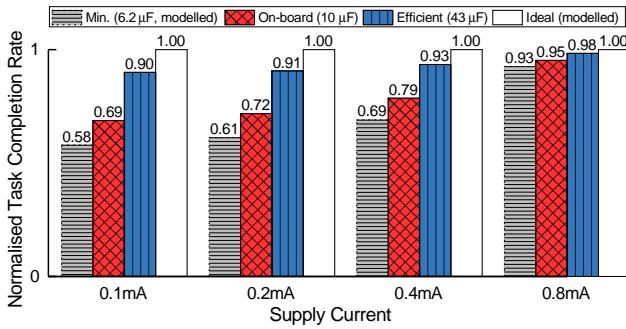


Fig. 8. Experimental comparison of task completion rates (forward progress) given different energy storage capacities.

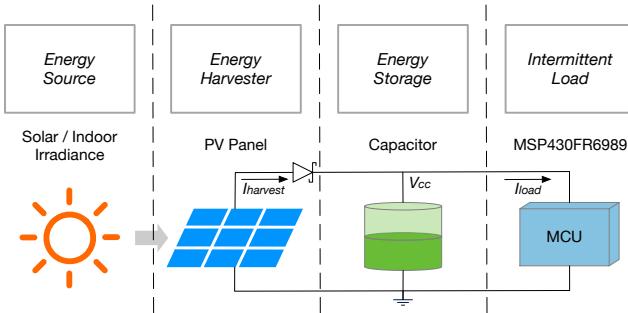


Fig. 9. System model configuration of a PV-based ICS.

2) *Forward Progress Optimisation:* We compared the tested completion speed among the minimum capacity ($6.2\ \mu\text{F}$, model generated), the on-board $10\ \mu\text{F}$ case, the efficient $43\ \mu\text{F}$ case, and an ideal case. The ideal case switches between sleeping and executing without restoring and saving overheads (mentioned in Section IV-B). As shown in Fig. 8, the efficiently-sized energy storage capacity ($43\ \mu\text{F}$) improves up to 55.2 % and 30.4 % more progress compared to the theoretical minimum one and the on-board capacitance one, and also achieves at least 90 % of the ideal forward progress. Relatively, the forward progress improvement by sizing energy storage is significant when supply is weak and vice versa.

VII. DEMONSTRATION OF SIZING APPROACH

A. System Model Configuration

We integrate the validated reactive intermittent computing model into a system model with photovoltaic (PV) energy-harvesting supply as shown in Fig. 9. The energy storage model and the intermittent load model are identical to the ones in Section V. We use a converter-less supply circuit with only a diode at the energy harvester output for preventing current backflow.

The energy source conditions are imported from NREL outdoor solar irradiance data [36] and EnHANTs indoor irradiance data [37]. To encompass different energy environments, four light source datasets across different environments are used in the following tests.

To convert irradiance into system power, we adopt a datasheet-based PV cell model [38]. This model takes the parameters available in common PV cell datasheets, so the

TABLE IV
PV CELL MODEL PROPERTIES

Parameter	Value
Open-Circuit Voltage	$0.89\ \text{V}/\text{cell}$
Short-Circuit Current	$14.8\ \text{mA}/\text{cm}^2$
MPP Voltage	$0.65\ \text{V}/\text{cell}$
MPP Current	$12.1\ \text{mA}/\text{cm}^2$

model can be easily reconfigured to suit various PV cells. The output current I_o of the PV cell model can be described as:

$$I_o = \frac{G}{G_{ref}} I_{sc} \left(1 - \left(1 - \frac{I_{mpp}}{I_{sc}} \right)^{\frac{V_o - V_{oc}}{V_{mpp} - V_{oc}}} \right) \quad (14)$$

where V_o is the output voltage of the PV cell, G is the ambient irradiance, G_{ref} is the reference irradiance (normally 1000W/m^2), and I_{sc} , V_{oc} , I_{mpp} , V_{mpp} are respectively short-circuit current, open-circuit voltage, and the current and voltage at maximum power point (MPP) given the reference irradiance. V_o and G are dynamic at run time, while other parameters in this model are constant.

A PV panel is an array of PV cells, which amplifies voltage and current output by connecting PV cells in series or parallel. In a PV panel, the open-circuit voltage is proportional to the number of cells in series, and the short-circuit current is proportional to the area of each cell and the number of cells in parallel. We refer to a commercial solar cell [39] for PV cell properties as shown in Table IV. We set four cells in series (with $V_{oc} = 3.56\text{V}$) to match the operating voltage of the MCU (maximum 3.6V), and model energy harvester sizing by scaling the cell area. A Schottky diode is connected to the energy harvester output in order to prevent current backflow.

Before the simulation, we developed and explored two simulation processes: (a) simulate system state chronologically with a fine-grained time step, and (b) sort energy source conditions into a distribution in time lengths and process the distribution. Process (b) shows only 0.892 % mean absolute percentage error compared to Process (a), while reducing simulation time significantly (e.g. from several hours to several seconds in a one-year test). Hence, we use Process (b) in the following tests.

B. Exploration with Real-World Energy Source Conditions

In the practical deployment of ICSs, ambient energy source conditions change over time and locations. The energy harvester and energy storage should be configured accordingly in order to achieve desired forward progress across various energy source conditions.

To explore different forward progress targets (namely target α_{exe}), three levels of baseline mean forward progress are assumed to be 0.1, 0.2, and 0.3. We use the system model to find the PV panel area that achieves expected forward progress under different energy source conditions. We scale the PV panel area with the minimum energy storage to find the PV panel area that achieves each target α_{exe} . As specified in Fig. 10, the energy harvester size that achieves desired forward progress may span over orders of magnitude given different

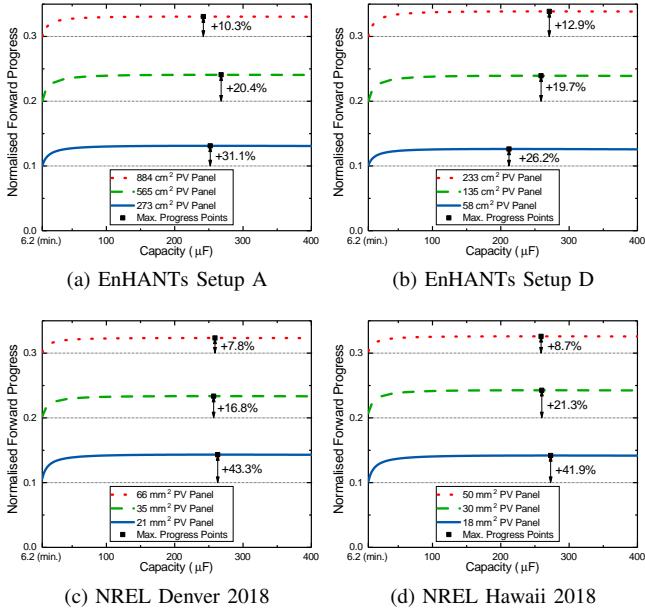


Fig. 10. Forward progress improvement by sizing energy storage given different PV panel areas under real-world energy source conditions. The model is able to find the PV panel area required for achieving the target mean forward progress.

energy source conditions (from mm^2 for outdoor sources to cm^2 for indoor sources).

We analyse the the sizing effect of energy storage on forward progress given real-world energy conditions. Fig. 10 shows 7.8–43.3 % mean forward progress improvement by sizing energy storage under the given real-world energy conditions and the baseline energy harvester sizes. Another takeaway is optimising energy storage can either improve forward progress upon a given energy harvester size, or reduce the energy harvester size that achieves the target forward progress. Given strong energy sources (e.g. Denver 2018 and Hawaii 2018 outdoor solar sources), increasing energy harvester size efficiently improves forward progress with minor dimensional overheads, e.g. tens of mm^2 ; however, given weak energy sources (e.g. EnHANTs Setup A and Setup D indoor light sources), optimising energy storage capacity save tens of cm^2 of PV panel area to achieve the same forward progress.

C. Trading off Forward Progress, Dimensions, and Recharging Time

Although increasing the storage capacity improves forward progress, larger capacitance may impact both dimensions and recharging time. We evaluate the overheads of increased capacitor dimensions and recharging time, and then trade off them with forward progress using a cost function to suggest an optimal capacitor size.

1) *Metric of Dimensions:* The overhead of capacitor dimensions is evaluated by characteristics of various 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 Device (SMD) package, and select six series of capacitors [34], [40]–[44]. The volume and

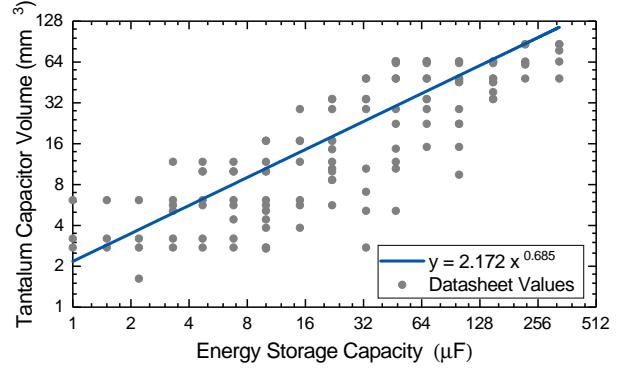


Fig. 11. Tantalum capacitor volume against capacitance [34], [40]–[44].

capacitance of these capacitors are plotted in Fig. 11 according to their datasheets. We use the regression of these data to represent the general capacitance-volume relationship.

2) *Metric of Recharging Time:* There is a recharging period between two consecutive execution periods in a intermittent computing device. During the recharging period, the device saves a volatile state, wait for supply voltage to recover, and restore the state to resume execution, without making forward progress. Applications that require frequent sensing may be negatively affected by such delay. We evaluate and consider this recharging time in our sizing approach.

Technically, the actual recharging time between two active periods depends on capacitance, the start and end voltage of recharging, and the current input and consumption. While the end voltage (restore voltage) and current consumption can be set or predicted, the start voltage and current flows vary with energy source conditions. Instead of evaluating the recharging time in complex conditions, we use the time between two successive execution intervals given 0.5 mA supply current in the Intermittent mode as a metric of recharging time (or recharging ability). Using Equation (5) and (7), we can obtain this recharging period between two successive execution intervals in the Intermittent mode as:

$$T_{\text{recharge}} = \frac{T_{\text{period}} - T_{\text{exe}}}{\frac{C(V_r - V_s) + T_s(I_{in} - I_s)}{I_{in} - I_{sleep}} + T_r} \quad (15)$$

3) *Trade-offs:* From the previous observations (Fig. 5) we can see that to achieve the optimal progress improvement costs much more storage capacity ($\text{mean } 3.2\times$) than to achieve 95 % improvement. A trade-off is necessary to improve progress while restricts the overheads of increased capacitor volume and recharging time.

As a part of the proposed sizing approach, the cost function Equation (1) is used to trade off forward progress, capacitor volume, recharging time. We configure the function by setting $k_1 = 0.2$, $k_2 = 200$, and $k_3 = 1$, i.e.:

$$f = \frac{\alpha_{\text{exe}}}{0.2} - \left(\frac{v_{\text{cap}}}{200} \right)^2 - T_{\text{recharge}}^2 \quad (16)$$

where v_{cap} is in mm^3 and T_{recharge} is in second. Here, $\frac{1}{k_1}/\frac{1}{k_2}$ equals 1000, but this does not mean that forward progress is 1000 times more important than capacitor volume.

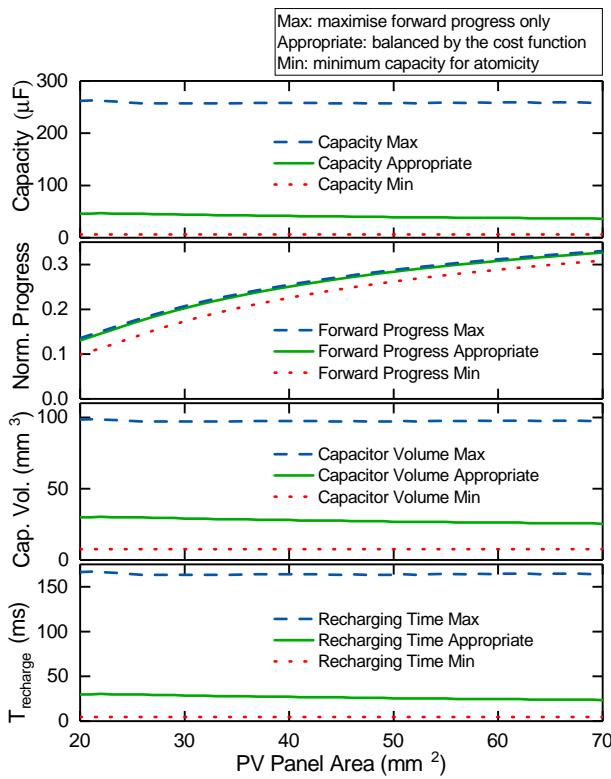


Fig. 12. The sizing approach trades off forward progress, capacitor volume, and recharging time. The results are plotted against a range of PV panel area, given Denver 2018 energy source dataset.

The effect of the trade-off is plotted in Fig. 12 using Denver 2018 energy source dataset as an example. Compared to the capacitor size that solely maximises forward progress, on average, an appropriately-sized capacitor achieves 98.3 % of the maximum forward progress, while saves 71.7 % capacitor volume and 83.8 % recharging time.

Compared to the minimum storage case, the appropriately-sized capacitor improves mean forward progress by 5.7–25.3 % with energy storage increased from 6.2 μF to 36–46 μF (depending on PV panel area). As shown in Fig. 11, the closest available capacitance that ensures the minimum storage is 6.8 μF , whereas the closest available capacitance around the appropriate capacitance are 33 μF or 47 μF . The minimum volume for these three capacitance are 2.75 mm^3 , 2.75 mm^3 , and 5.12 mm^3 , which means using the optimal capacitance, instead of the minimum one, may not incur dimensional overhead (for 33 μF). For 47 μF , the absolute volume (5.12 mm^3) is insignificant compared to the device as a whole, e.g. an MSP430FR6989 MCU chip alone occupies 274.4 mm^3 (14 × 14 × 1.4). The regressed volume of the above three capacitance values are 8.1 mm^3 , 23.8 mm^3 , 30.4 mm^3 respectively. Again, such a volume scale is still insignificant in a whole device.

VIII. CONCLUSION

In this paper, we presented a model of reactive intermittent computing that accurately estimates forward progress. Using this model, we explored the sizing effect of energy

storage on forward progress with respect to supply current and volatile state size, showing up to 64.9 % progress improvement under constant current supply and 7.8–43.3 % improvement on annual mean forward progress under various real-world energy conditions. We proposed an approach for sizing energy storage, which improves forward progress with insignificant overheads on capacitor volume and recharging time. We believe that the problem of frequent power interruptions in energy-harvesting supply can be easily alleviated by using an appropriately-sized capacitor. Energy storage should be carefully designed, rather than minimised or indiscriminately picked, to efficiently operate ICSs.

REFERENCES

- [1] T. Adegbija, A. Rogacs, C. Patel, and A. Gordon-Ross, “Microprocessor optimizations for the internet of things: A survey,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 7–20, Jan 2018.
- [2] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, “Vision and challenges for realising the internet of things,” *Cluster of European Research Projects on the Internet of Things, European Commission*, vol. 3, no. 3, pp. 34–36, 2010.
- [3] D. Evans, “The internet of things: How the next evolution of the internet is changing everything.” [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf
- [4] Sravanthi Chalasani and J. M. Conrad, “A survey of energy harvesting sources for embedded systems,” in *IEEE SoutheastCon 2008*, April 2008, pp. 442–447.
- [5] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *IEEE Communications Surveys Tutorials*, vol. 13, no. 3, pp. 443–461, Third 2011.
- [6] 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.
- [7] B. Buchli, F. Sutton, J. Beutel, and L. Thiele, “Dynamic power management for long-term energy neutral operation of solar energy harvesting systems,” in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys ’14. New York, NY, USA: ACM, 2014, pp. 31–45.
- [8] P. Wägemann, T. Distler, H. Janker, P. Raffeck, V. Sieh, and W. Schröder-Preikschat, “Operating energy-neutral real-time systems,” *ACM Trans. Embed. Comput. Syst.*, vol. 17, no. 1, pp. 11:1–11:25, Aug. 2017.
- [9] X. Jiang, J. Polastre, and D. Culler, “Perpetual environmentally powered sensor networks,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, ser. IPSN ’05. Piscataway, NJ, USA: IEEE Press, 2005.
- [10] F. Simjee and P. H. Chou, “Everlast: Long-life, supercapacitor-operated wireless sensor node,” in *Proceedings of the 2006 International Symposium on Low Power Electronics and Design*, ser. ISLPED ’06. New York, NY, USA: ACM, 2006, pp. 197–202.
- [11] 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, integrated Ecological Indicators for Sustainable Urban Ecosystem Evaluation and Management.
- [12] F. I. Simjee and P. H. Chou, “Efficient charging of supercapacitors for extended lifetime of wireless sensor nodes,” *IEEE Transactions on Power Electronics*, vol. 23, no. 3, pp. 1526–1536, May 2008.
- [13] F. Akhtar and M. H. Rehmani, “Energy replenishment using renewable and traditional energy resources for sustainable wireless sensor networks: A review,” *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 769 – 784, 2015.
- [14] B. Ransford, J. Sorber, and K. Fu, “Mementos: System support for long-running computation on RFID-scale devices,” in *Proceedings of the Sixteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS XVI. New York, NY, USA: ACM, 2011, pp. 159–170.

this section should be conclusion, not a survey of ICS in the paper...

- [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 *2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*, Jan 2014, pp. 330–335.
- [16] 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, March 2015.
- [17] B. Lucia and B. Ransford, "A simpler, safer programming and execution model for intermittent systems," in *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI '15. New York, NY, USA: ACM, 2015, pp. 575–585.
- [18] Y. Wang, Y. Liu, S. Li, D. Zhang, B. Zhao, M. Chiang, Y. Yan, B. Sai, and H. Yang, "A 3us wake-up time nonvolatile processor based on ferroelectric flip-flops," in *2012 Proceedings of the ESSCIRC (ESSCIRC)*, Sep. 2012, pp. 149–152.
- [19] 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.
- [20] 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 *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC '19. New York, NY, USA: ACM, 2019, pp. 26:1–26:6.
- [21] 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: Mathematical, Physical and Engineering Sciences*, vol. 378, no. 2164, p. 20190158, 2020.
- [22] 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.
- [23] 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 *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2015, pp. 526–537.
- [24] 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.
- [25] J. Hester, L. Sitanayah, and J. Sorber, "Tragedy of the coulombs: Federating energy storage for tiny, intermittently-powered sensors," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 5–16.
- [26] D. Balsamo, B. J. Fletcher, A. S. Weddell, G. Karatzolas, B. M. Al-Hashimi, and G. V. Merrett, "Momentum: Power-neutral performance scaling with intrinsic mppt for energy harvesting computing systems," *IEEE Trans. Embed. Comput. Syst.*, vol. 17, no. 6, Jan. 2019.
- [27] 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.
- [28] K. Maeng, A. Colin, and B. Lucia, "Alpaca: Intermittent execution without checkpoints," *Proc. ACM Program. Lang.*, vol. 1, no. OOPSLA, pp. 96:1–96:30, Oct. 2017.
- [29] 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: ACM, 2019, pp. 1101–1116.
- [30] 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, pp. 33:1–33:21, Jul. 2019.
- [31] J. San Miguel, K. Ganesan, M. Badr, and N. E. Jerger, "The EH model: Analytical exploration of energy-harvesting architectures," *IEEE Computer Architecture Letters*, vol. 17, no. 1, pp. 76–79, Jan 2018. *International Symposium on Microarchitecture (MICRO)*, Oct 2018, pp. 600–612.
- [32] N. Jackson, J. Adkins, and P. Dutta, "Capacity over capacitance for reliable energy harvesting sensors," in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, ser. IPSN '19. New York, NY, USA: ACM, 2019, pp. 193–204.
- [33] AVX, "AVX TAJ low profile series tantalum capacitors." [Online]. Available: http://datasheets.avx.com/TAJ_LOW_PROFILE.pdf
- [34] R. Faltus, "Low leakage current aspect of designing with tantalum and niobium oxide capacitors." [Online]. Available: https://www.avx.com/docs/techinfo/Low_Leakage_Current_Aspect_Designing_Tantalum_Niobium_Oxide_Capacitors.pdf
- [35] 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.
- [36] M. Gorlatova, A. Wallwater, and G. Zussman, "Networking low-power energy harvesting devices: Measurements and algorithms," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1853–1865, Sep. 2013.
- [37] 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.
- [38] Panasonic, "Amorton amorphous silicon solar cells." [Online]. Available: https://www.panasonic-electric-works.com/cps/rde/xbscr/pew_eu_en/ca_amorton_solar_cells_en.pdf
- [39] AVX, "AVX TACmicrochip low profile series tantalum capacitors." [Online]. Available: http://datasheets.avx.com/TAC_LP.pdf
- [40] ———, "AVX F92 series tantalum capacitors." [Online]. Available: <http://datasheets.avx.com/F92.pdf>
- [41] Vishay, "Vishay 572D series tantalum capacitors." [Online]. Available: <http://www.vishay.com/docs/40064/572d.pdf>
- [42] ———, "Vishay 591D series tantalum capacitors." [Online]. Available: <https://www.vishay.com/docs/40012/591d.pdf>
- [43] ———, "Vishay 592D series tantalum capacitors." [Online]. Available: <http://www.vishay.com/docs/40004/592d.pdf>

Jie Zhan Biography text here.

Alex S. Weddell Biography text here.

Geoff V. Merrett Biography text here.