Stateful Energy Management for Multi-Source Energy Harvesting Transient Computing Systems

Sergey Mileiko[1], Oktay Cetinkaya[2], Rishad Shafik[1], Domenico Balsamo[1*]
[1]MicroSystems Research Group, School of Engineering, Newcastle University, Newcastle NE1 7RU, UK.
[2]Oxford e-Research Centre, Department of Engineering Science, University of Oxford, Oxford OX1 3QG, UK.
[*]domenico.balsamo@newcastle.ac.uk

***Abstract*—The intermittent and varying nature of energy harvesting (EH) entails dedicated energy management with large energy storage, which is a limiting factor for low-power/cost systems with small form factors. Transient computing allows system operations to be performed in the presence of power outages by saving the system state into a non-volatile memory (NVM), thereby reducing the size of this storage. These systems are often designed with a *task-based* strategy, which requires the storage to be sized for the most energy consuming task. That is, however, not ideal for most systems since their tasks have varying energy requirements, i.e., energy storage size and operating voltage. Hence, to overcome this issue, this paper proposes a novel energy management unit (EMU), tailored for multi-source EH transient systems, that allows selecting the storage size and operating voltage for the next task to be performed at run-time, thereby optimizing task-specific energy needs and startup times based on application requirements. For the first time in literature, we adopted a hybrid *NVM+VM* approach allowing our EMU to reliably and efficiently retain its internal state, i.e., *stateful* EMU, under even the most severe EH conditions. Extensive empirical evaluations validated the operation of the proposed stateful EMU at a small overhead ($0.07m$J of energy to update the EMU state and a $\simeq 4\mu$A of static current consumption of the EMU).**

***Index Terms*—Energy management unit, energy harvesting, transient computing, energy efficiency, low-power design.**

I. Introduction

Energy harvesting (EH) can help operate low-power systems by harnessing electrical energy from ambient sources [1]. However, the spatio-temporal variance of these sources makes EH unreliable for the perpetual execution of sensing, processing, and transmission operations. Therefore, EH systems usually require dedicated energy management with large energy storage to flatten the mismatch between EH output and systems' consumption. However, this storage increases the volume, weight, and cost of systems and takes longer to charge, delaying the system startup [2].

Here, *transient computing* steps in, allowing operations to span across power outages, thereby minimizing the size of this storage [3]. This paradigm is often designed with a *task-based* strategy, which breaks operations into small tasks, each of which must be completed within one EH cycle. The results of a completed task and a snapshot of the system state, i.e., the contents in core registers and volatile memory (VM), are saved into non-volatile memory (NVM) to guarantee the intermittent operation of systems [4]. Task-based schemes ensure reliability in task execution but require the energy storage to be sized for the most energy-consuming task.

In practice, however, the energy required for different tasks, and even for the same task, can significantly change over time, which is not ideal when a fixed storage is used. In addition, the systems have microcontrollers (MCUs), memories, and peripherals, which typically operate at a wide range of supply voltages. However, system designers avoid using multiple voltage domains and choose the highest minimum voltage required to supply the entire system as a sub-optimal solution.

To overcome these issues, we propose an Energy Management Unit (EMU) tailored for task-based transient computing systems that manages energy extraction from various resources, i.e., *multi-source EH*, and allows selecting the size of the energy storage as well as the operating voltage for each task at run-time, thereby optimizing task-specific energy needs and startup times based on application requirements. Furthermore, unlike the related existing works, in Sec. II, this EMU is capable of handling the most severe harvesting conditions (minimum or no input power) without losing its internal state, i.e., the energy storage size and the operating voltage required to perform the next task, which refers to *stateful EMU*. That is achieved by equipping the EMU with NVM elements that are able to maintain this state.

The main bottleneck of integrating NVM elements is that their access time and power consumption are higher than that of volatile counterparts, such as flip-flops (FFs) and latches. That is a significant barrier, especially when energy is scarce [5]. In addition, emerging technologies for NVMs, such as FRAM, PCM, RRAM, *etc.*, aim for high-density storage to replace existing memories, e.g., Flash or SRAM, rather than offering single-bit storage to maintain the internal circuit states. Therefore, it is evident that the field is lacking low-power and fast-access single-bit NVMs. To address this issue, we propose a hybrid *NVM+VM* approach to retain the internal state of the EMU. This state is primarily maintained on VM elements, whilst the NVM elements are activated sporadically, i.e., to update the state or when a power outage occurs. Thus, this state can be retained at a lower energy cost thanks to the minimal use of NVM elements but more reliably, thanks to the combination of *NVM+VM* elements.

Following are the novel contributions of this paper:

- An EMU tailored for multi-source EH transient systems that can select the energy storage size and operating voltage for the next task based on application requirements;
- A hybrid *NVM+VM* approach for reliable retainment of the internal EMU state across power outages;

978-3-9819263-7-8/DATE23/© 2023 EDAA

- A task-based transient computing system embodying the proposed EMU to evaluate its performance under varying EH conditions and applications requirements.

The remainder of this paper is organized as follows. We start with the related works in Sec. II to draw attention to the gap in the field, motivating our proposal. Then, in Sec. III, we explain our thinking behind the design of the proposed EMU and the transient system. That is followed by EMU architecture in Sec. IV, where we introduce each component of the EMU in detail. Afterwards, we provide the EMU and transient system evaluation in Sec. V, highlighting the capabilities offered by our proposal. Finally, we draw our conclusions with some insights on future research directions.

II. Energy Management Designs

In theory, the load, e.g., the MCU, of a transient computing system can be directly connected to an EH source without any intermediate energy storage if there is a compatible voltage-current curve between the source-load pair [6, 7]. However, energy harvesters often cannot extract enough energy to power the load directly; therefore, transient systems require a dedicated EMU with intermediate storage. Reliable and efficient energy management becomes even more critical when harvesting from multiple sources [8], which is typically achieved via a Schottky diode-based power OR-ing that allows unidirectional current flow [9]. However, this method has additional power consumption due to the forward voltage drop across the diodes. That can be solved by active switches with negligible voltage losses in their ON states (MOSFETs and control ICs) [10, 11].

Most EH systems exploit a few sources (two-three) and charge a single storage, e.g., a (super)capacitor. For those, OR-ing can occur before or after the power conversion phase. The first option allows a single power converter for all sources but requires a similar internal impedance. That is not always achievable, as each harvester is designed for a specific source; hence, they have a different impedance. To exemplify, for the harvesters considered in this work, i.e., a thermoelectric generator (TEG) and a photovoltaic (PV) cell, the internal impedances differ by orders of magnitude (tens of Ωs for TEGs and several KΩs for PVs). The second option needs multiple converters, equivalent to the number of sources exploited.

For EH transient systems, dedicated power converters are used when harvesting microwatts (μW) to milliwatts (mW) to increase voltage or current levels and accumulate charge in the intermediate storage until the systems perform their tasks, i.e., the harvested energy is enough to supply the load. These may comprise maximum power point tracking (MPPT) circuits (e.g., charge regulators) that adjust the converter's input impedance for varying ambient conditions and extract the maximum power possible from each source over time [12]. However, most energy harvesters (e.g., microturbines or TEGs) do not necessarily require MPPT as their internal impedance does not significantly change as ambient conditions change.

When powering the load via the intermediate storage, a common approach to ensure a stable output voltage is to use buck converters or low dropout (LDO) regulators. However, fixing the voltage, so using an additional converter, may not be necessary considering that the system components operate on a wide range of supply voltages. Yet, the task-specific voltage/storage requirements have to be carefully selected and maintained for reliable and efficient operations. For example, in [13], the authors proposed a low-power EMU for a single EH source (PV cell) that uses both boost and buck converters, where the harvesting part is based on the commercial BQ25505 from Texas Instruments. This unit builds up charge with minimum start-up costs and allows selecting the operating voltage for the next task at run-time via a dedicated control interface. However, this interface still requires minimum power to maintain the operating voltage information and is unable to select the energy storage size. In [14], an EMU managing multiple harvesters (PV cell and TEG) to extract energy from human body and power a wearable device (along with a main battery) was presented. That is also based on commercial BQ25505 from Texas Instrument and LTC3108 from Linear Technology, where the passive OR-ing occurs after the power conversion. However, the authors did not explore any run-time strategy to select and maintain the operating voltage and/or the energy storage size for the next task.

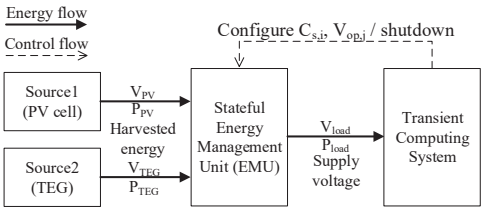

Figure 1: Energy and control flows between the main components.

What is missing? Although we give only two examples above, the issues mentioned are valid field-wide. Hence, we propose the EMU shown in Fig. 1 to overcome the everlasting challenges of energy management with task-based transient systems. Our EMU can manage energy extraction from two sources, without loss of generality, select the storage size ($C_{s,i}$) and operating voltage level ($V_{op,j}$) for the next task, and keeps its internal state under severe EH conditions. The first harvester, TEG, uses a step-up transformer, i.e., a voltage booster, which allows the harvesting process to be started with low input power, while the second (PV cell), requires a boost converter with MPPT. The EMU uses no buck or LDO converters so that the system can operate over a range of supply voltages. Moreover, it adopts both VM and NVM elements for reliable and low-cost retainment of the internal state. Details of each part are provided in the following section.

III. EMU and the Transient System Model

This section explains our thinking behind the design of the proposed stateful EMU and the accompanying task-based transient system. First, we focus on how to select $C_{s,i}$ and $V_{op,j}$ (for the next task to be performed), referring to the internal state of the EMU, and then discuss the hybrid *NVM+VM* approach to maintain this state in the event of a power outage.

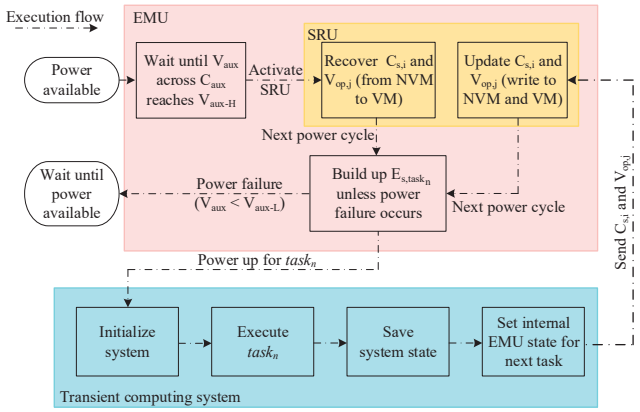

Figure 2: Execution flow between the EMU and transient system.

Fig. 2 shows the execution flow of the task-based transient system and the stateful EMU considered. While the EMU builds up energy, $E_{s,task_n}$, required for $task_n$, the transient system is disabled/off. When the system powers up after this period of energy unavailability, the MCU performs some basic initialization and then starts executing $task_n$, which may require using some peripherals, e.g., sensors and transceivers. Upon completion, the results of the task and the snapshot of the system state are saved in the NVM of the MCU (e.g., FRAM). Finally, the internal state of the EMU is set, configuring the next energy level required. That involves activating the NVM elements of the EMU and updating $C_{s,i}$ and $V_{op,j}$, which is duplicated on the VM elements (further discussed in Sec. IV).

To ensure reliability in task execution, the energy stored by the EMU, $E_{s,task_n}$, at each power cycle must be equal to or larger than the total energy required to perform the task, $E_{tot,task_n}$. That is given by the sum of the energy required for system initialization, E_{init}, for task execution, $E_{e,task_n}$, for saving the system state, E_{save}, and for setting the internal EMU state, E_{update} (actual overhead). The value of $E_{e,task_n}$ may significantly vary for different tasks, such as sensing, processing, and transmission. That can also change between different iterations of the same task. For example, when transmitting, the energy required may vary over time based on various parameters, such as transmit power, bandwidth, and spreading factor (SF) [15]. Therefore, the values of $C_{s,i}$ and $V_{op,j}$ are selected at run-time based on $E_{e,task_n}$ per each task that is determined by profiling the application requirements.

Ideally, the selection space for the internal EMU state should consist of a broad set of energy storage sizes ($C_{s,1}$, $C_{s,2}$, ..., $C_{s,N}$) and operating voltages ($V_{op,1}$, $V_{op,2}$, ..., $V_{op,M}$), where choosing a pair of $C_{s,i}$-$V_{op,j}$ provides a different $E_{s,task_n}$, i.e.:

$$E_{s,task_n} = C_{s,i}(V_{op,j}^2 - V_{sys,min}^2)/2. \quad (1)$$

Here, $V_{sys,min}$ is the minimum voltage of the transient system. In practice, we limit this selection to a few pairs since we do not want to excessively increase the EMU's design complexity while proposing a scalable solution for it. Above all, we want to limit the energy consumption due to the NVM and VM elements, which increases with the increasing number of pairs. Thus, as a proof of concept, our EMU considers two voltage levels ($V_{op,1}$ and $V_{op,2}$) and two storage sizes ($C_{s,1}$ and $C_{s,2}$), whose values are provided in Sec. V, resulting in four pairs to be selected as the internal EMU state.

The EMU uses a dedicated state retention unit (SRU) to update and maintain this state and recover it in the event of a power outage. During the recovery, the EMU activates the NVM elements, copies the previously saved state into the VM elements, and immediately after that it deactivates the NVM elements. To determine if a power outage occurred, the EMU monitors the voltage, V_{aux}, through an auxiliary capacitor, C_{aux}, and activates the SRU only when it reaches a given threshold, V_{aux-H}. At the same time, it deactivates when V_{aux} falls below a threshold, V_{aux-L}, thus providing a certain level of hysteresis when powering the SRU. The values of V_{aux-H} and C_{aux} are selected as ensuring the correct operating range for the NVM elements (between V_{aux-H} and the minimum voltage, $V_{NVM,min}$) during the period while restoring the internal EMU state, t_r. Hence, C_{aux} can be defined as:

$$C_{aux} = \frac{2P_{SRU}t_r}{V_{aux-H}^2 - V_{NVM,min}^2}, \quad (2)$$

where P_{SRU} is the average power consumption of the SRU during restore, including both NVM and VM elements and the additional memory control logic. At the same time, V_{aux-L} is chosen considering the minimum operating voltage of the VM elements to limit the number of restores from NVM to VM elements. That can be quite low depending on the type of the NV elements. For example, when using latches or FFs, this voltage can go as low as $0.8 - 1$V[1] , which is typically lower than $V_{sys,min}$, i.e., $2 - 2.2$V.

Next section presents the close-up of each EMU part, detailing their architecture and how they contribute to updating and recovering the internal EMU state and the consequent build-up of energy through $C_{s,i}$ at $V_{op,j}$, besides providing the EMU with the capability of harvesting energy from multiple sources.

IV. EMU Architecture

Fig. 3 shows the energy and control flows between the three main components, namely *the harvesters*, i.e., the TEG and PV cell, *the EMU* managing the harvested energy and providing the supply voltage for *the transient system*. Depending on the next task, the system configures the internal EMU state, i.e., $C_{s,i}$ and $V_{op,j}$, before turning off, and thus controls the EMU, as shown with the dashed line (Configure $C_{s,i}$, $V_{op,j}$) in Fig. 3.

The EMU accommodates the following sub-components to build up the energy required to drive the whole system:

- **Voltage Booster**: This component boosts the input voltage from the TEG, V_{TEG}, typically tens of mV, to the sufficient levels for driving the charge controller. Fig. 4 shows the schematic of the voltage booster, which is a resonant DC-to-DC converter with a step-up transformer of $1:100$ ratio, allowing to start up the system from voltages as low as 20mV. The AC voltage produced on the secondary winding

[1]Nexperia, "Low-power D-type transparent latch; 3-state," 74AUP1G373 datasheet, Nov. 2006 [Revised Jan. 2022].

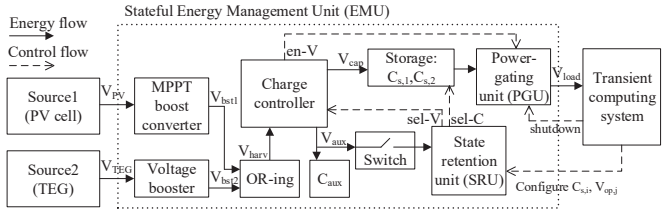

Figure 3: Close-up of the stateful EMU.

of the transformer is boosted and rectified by the charge pump capacitors (C_{cp1} and C_{cp2}) and the rectifier circuit (based on rectifiers D_1 and D_2 and a sync rectify element) that feeds current to the OR-ing circuit.

- **MPPT Boost Converter**: For the PV cell, we used an off-the-shelf DC-to-DC step-up converter[2] with an MPPT. The MPPT allows charge controller to be driven from a voltage, V_{PV}, as low as the minimum operating voltage of 225mV. That enables direct operation from the PV cell and, more generally, from any other low-voltage power source with high series impedance, e.g., fuel cells.
- **Active OR-ing**: To combine the two harvesters into a single output power, P_{harv}, we used two commercial controllers[3], each of which is paired with an NMOS to emulate a low forward voltage diode.
- **Charge Controller + Switch**: The charge controller shown in Fig. 5 receives P_{harv} from OR-ing, powers the SRU that is responsible for retaining the internal EMU state, and provides charge to the selected energy storage ($C_{s,1}$ or $C_{s,2}$). Before the last step, it ensures that V_{aux} through C_{aux} reaches V_{aux-H} for the reliable start up and operation of the SRU, which is achieved by the comparator, $Comp_1$. According to sel–V signal provided by the SRU, the controller verifies that the voltage across the selected storage, V_{cap}, reaches the required level before enabling the power-gating unit (PGU) (via the en–V signal) that is responsible for supplying the system. This is achieved using a programmable voltage divider, driven by sel–V, and a second comparator, $Comp_2$. The storage is selected based on sel–C, also provided by the SRU. This drives the two low-leakage NMOS connected in series with $C_{s,1}$ and $C_{s,2}$, where only one of these storages can be selected at a time. Finally, for the reliable operation of the SRU when powered,

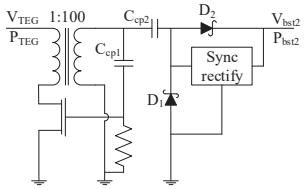

Figure 4: Schematic of the Voltage Booster.

[2]Linear Technology, "400mA Step-Up DC/DC Converter with Maximum Power Point Control and 250mV Start-Up," LTC3105 datasheet, May 2010 [Revised Nov. 2015].

[3]Texas Instruments, "TPS241x N+1 and ORing Power Rail Controller," TPS241x datasheet, Jan. 2007 [Revised Oct. 2019].

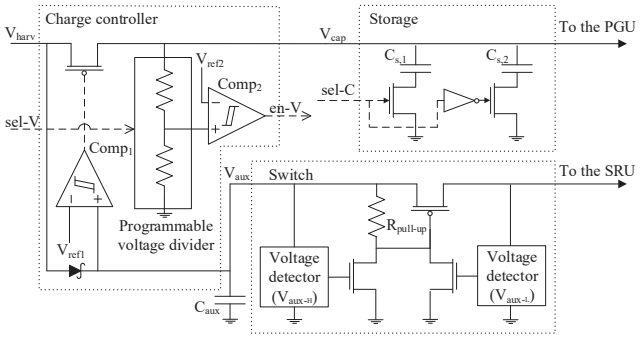

Figure 5: Schematic of the Charge Controller + Switch arrangement.

a switch with internal hysteresis is used to enable and disable the SRU. This switch monitors V_{aux} across C_{aux} and activates the SRU when it exceeds V_{aux-H} and deactivates the SRU when V_{aux} falls below V_{aux-L} (see Sec. III). The hysteresis is achieved by a pair of micro-current voltage monitors configured in a MOSFET-latch arrangement.

- **State Retention Unit (SRU)**: This component maintains the internal EMU state via the VM elements when power is available ($V_{aux} \geq V_{aux-L}$) and uses the NVM elements to restore the state in the VM elements upon a power outage.

As shown in Fig. 6, to implement this hybrid *NVM+VM* approach, the SRU is equipped with three NVM variable digital resistors that function as digital switches (the 3-bit NVM in Fig. 6)[4]. These are programmable through an inter-integrated circuit (I2C) interface. Additionally, the SRU uses two low-power D-latches as NV elements. During the system update phase, the transient system activates the 3-bit NVM and updates the internal EMU state via the I2C interface. The first bit, NVM-sel–V, is used to store the selected operating voltage (0 = $V_{op,1}$ & 1 = $V_{op,2}$), the second, NVM-sel-C, is used for the energy storage size (0 = $C_{s,1}$ & 1 = $C_{s,2}$), and the third, NVM_{ready}, is used as a control signal to indicate when the NVM elements can be read by the memory control logic, thus avoiding the use of additional power-hungry delay elements. The memory control logic receives these bits, copies the NVM-sel–V and NVM-sel–C bits onto the D-latches as soon as the NVM_{ready} bit is activated, and deactivates the 3-bit NVM. The same read/copy process happens when recovering from a power outage, i.e., when V_{aux} reaches V_{aux-H} after falling below V_{aux-L}. The 3-bit NVM is activated by the memory control logic to copy NVM-sel–V and NVM-sel–C into

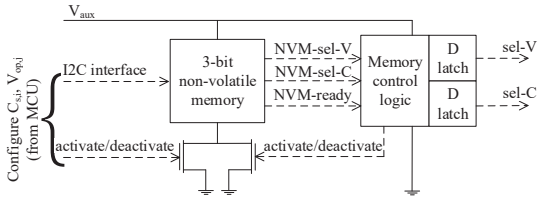

Figure 6: Schematic of the SRU.

[4]Maxim Integrated, "Triple 128-Position Nonvolatile Digital Variable Resistor/Switch," DS3904/DS3905 datasheet, Jan. 2007 [Revised Mar. 2007]

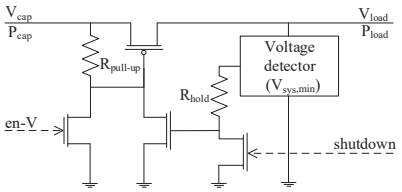

Figure 7: Schematic of the PGU.

the D-latches, and soon after, it is deactivated.

- **Power-gating Unit (PGU)**: This component, shown in Fig. 7, switches the transient system (load) on and off depending on the voltage across the storage (V_{cap}). When the selected storage is fully charged, the system is activated by the enable signal, en–V, of the charge controller. The voltage detector keeps the system on until V_{cap} drops below the minimum operating voltage of the system, $V_{sys,min}$. Another way of turning the system off rather than dying due to power outage is sending a *shutdown* request to the EMU. When the system is off, the PGU prevents the load consume the stored energy. Hence, the stored energy reaches its turn-on voltage in quick succession.

V. EMU and Transient System Evaluation

In this section, we first describe the *transient system* used for EMU evaluation. Then, we focus on measuring the *SRU consumption* when restoring the internal EMU state, which is used to calculate C_{aux}. Finally, we analyze the *EMU performance* when the system is powered by only one source (TEG) and then a combination of two, i.e., TEG and PV cell.

Transient system: The system used in our evaluations is a Texas Instrument MSP430FR MCU that measures the rate of water flow in pipes using ultrasonic sensors ($task_1$). This task requires an $E_{tot,task_1}$ of $1.32mJ$, measured experimentally. The measured flow rate values are saved into the NVM of the MCU, i.e., FRAM, to guarantee the intermittent behaviour among tasks. The MCU uses the serial peripheral interface (SPI) to communicate with a LoRa transceiver[5] and send the measurements to a LoRa gateway ($task_2$). The required $E_{tot,task_2}$ for this varies depending on transmit power, bandwidth, and SF of the transceiver. To understand their effect, we consider two SF values, 7 (lowest) and 12 (highest), which respectively require an $E_{tot,task_2}$ of $4.96mJ$ and $92.62mJ$, measured experimentally. The energy needs of sub-operations contributing to the total energy spending for each task are shown in Table I. The system also uses the I2C interface to set

Table I: Requirements of each task.

Task	E_{init} (mJ)	E_e (mJ)	E_{save} (mJ)	E_{update} (mJ)	E_{tot} (mJ)	$C_{s,i}$ (mF)	$V_{op,j}$ (V)	E_s (mJ)
$task_1$	0.69	0.52	0.04	0.07	1.32	1.33	2.5	1.49
$task_2$ (SF 7)	0.69	4.16	0.04	0.07	4.96	1.33	3.6	5.90
$task_2$ (SF 12)	0.69	91.82	0.04	0.07	92.62	22	3.6	98.56

[5]Hoperf, "Low Power Long Range Transceiver Module," RFM95W datasheet, Oct. 2013 [Revised Sept. 2019].

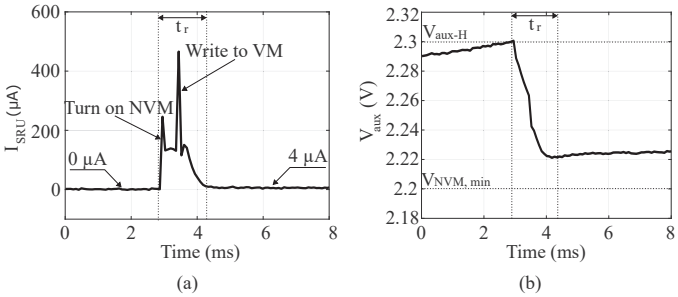

Figure 8: Changes in: (a) I_{SRU}; (b) V_{aux} across C_{aux} during the recovery of internal EMU state.

the internal EMU state, $C_{s,i}$-$V_{op,j}$, and an output that requests the shutdown when tasks are completed.

SRU consumption: Fig. 8(a) shows the current consumption of the SRU, I_{SRU}, during the recovery period (t_r), which lasts 1.6ms. The first peak is due to turning the 3-bit NVM on. That is followed by a fairly stable line, which denotes NVM initialization. The second peak is due to copying the values onto D-latches using the memory control logic. After this, the 3-bit NVM is disabled, which yields to a static current consumption of $\cong 4\mu$A (actual overhead when power is available). Here, V_{aux-H} is set to 2.3V, knowing that $V_{NVM,min}$ is 2.2V and considering a maximum voltage drop of 0.1V during recovery. Therefore, C_{aux} is calculated as 2μF using (1) and selected as 2.2μF. The drop in V_{aux} after reaching V_{aux-H} in Fig. 8(b) confirms that the recovery process ends before reaching $V_{NVM,min}$ with the selected C_{aux}. After that, V_{aux} rises to a value of 2.5V.

EMU performance: Fig. 9 shows the operations of the EMU and transient system when powered by the TEG for 15 minutes. Here, P_{TEG} changes over time based on the temperature difference, ΔT, between the two surfaces of the TEG. To perform $task_1$, $C_{s,1} = 1.33mF$ and $V_{op,1} = 2.5$V (see Table I) are selected ($E_{s,task_1} = 1.49mJ$) while $C_{s,1}$ and $V_{op,2} = 3.6$V

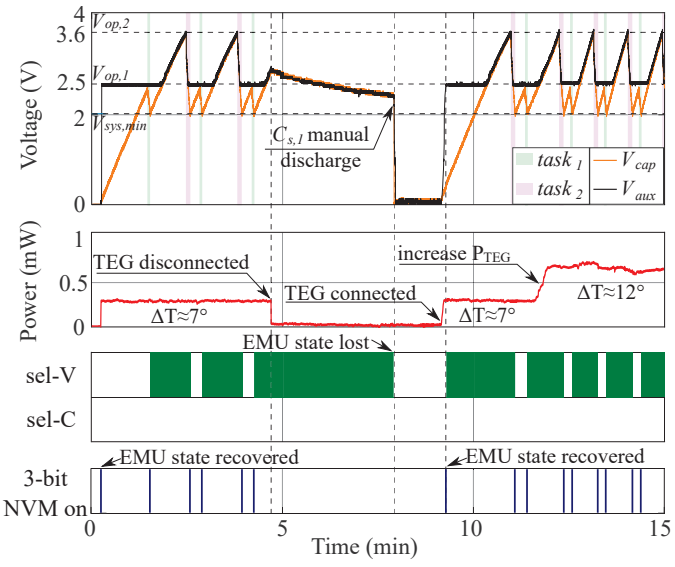

Figure 9: Experimental results showing the transient operation of the system with the EMU when powered by the TEG.

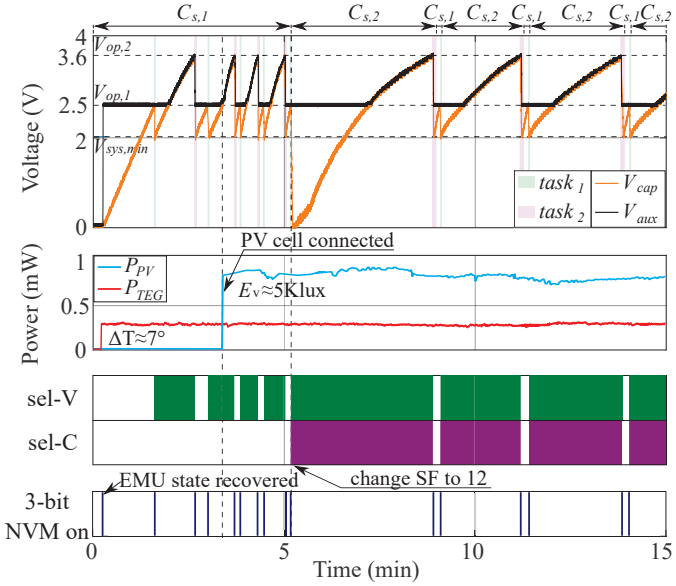

Figure 10: Experimental results showing the transient operation of the system with the EMU when powered by the TEG and PV cell.

($E_{s,task_2} = 5.9mJ$) are used for $task_2$ at SF7. Initially, the EMU recovers its internal state and alternately performs $task_1$ and $task_2$. When TEG is disconnected (a power outage occurs), P_{TEG} drops to 0 and both V_{aux} and V_{cap} discharge slowly. To speed up operations, we manually discharge $C_{s,1}$, causing the loss of internal EMU state on the VM elements. Nevertheless, whenever the power is available again (TEG connected), the recovery occurs, and the correct $C_{s,1}$-$V_{op,2}$ is restored for $task_2$. As can be seen, *sel–V* changes after the state on the 3-bit NVM has been updated or recovered, while *sel–C* is 0 because only $C_{s,1}$ is used. Over time, the 3-bit NVM is only sporadically active during restore and update. This confirms that the EMU can recover its internal state after a power outage. It can also be observed that the charging times of $C_{s,1}$ vary depending on P_{TEG}. For example, increasing ΔT from 7°C to 12°C increases P_{TEG}, reducing charging times and the time between performing tasks.

Fig. 10 shows the operations of the transient system and EMU when powered by both TEG and PV cell for 15 minutes. To perform $task_1$, $C_{s,1}$-$V_{op,1}$ is selected while $C_{s,1}$-$V_{op,2}$ is used for $task_2$ until the fifth minute where the SF changes from 7 to 12. Since then, $C_{s,2} = 22mF$ and $V_{op,2}$ ($E_{s,task_2} = 98.56mJ$) are used instead, which increase the charging time required to perform $task_2$. Initially, only the TEG harvests energy with a constant ΔT of 7°C and P_{TEG} of $0.3mW$. From the fourth minute, the PV cell is connected, thereby increasing the harvested power based on the illuminance (E_v) it intercepts (in Klux). Thus, the charging time of $C_{s,1}$ and (afterwards) $C_{s,2}$ are reduced, decreasing the time between tasks. It can be seen that both *sel–V* and *sel–C* change after the state on the 3-bit NVM has been updated; for *sel–C*, that only happens when SF is moving to 12. Over time, 3-bit NVM becomes active sporadically, only during the updates. This experiment confirms the EMU's ability to adjust the energy requirements at run-time, not only between tasks but also between different iterations of the same task.

Future studies will explore adopting a broader set of energy storage sizes and operating voltages to increase the selection space for the internal EMU state. Furthermore, we will investigate how to determine the energy requirements of each task automatically, without knowing them in advance, using various learning mechanisms, e.g., reinforcement learning. This will make the current EMU and transient system completely independent of the application (application-agnostic).

VI. Conclusions

This paper proposes a stateful EMU for task-based transient systems powered by multiple EH sources. The EMU, operating on the energy collected by a TEG and a PV cell, allows selecting the energy storage size and the operating voltage for the next task, i.e., the internal EMU state, at run-time. In this way, it optimizes task-specific energy needs and startup times based on application requirements. After practical implementation, the proposed EMU and the accompanying transient system were experimentally tested. The results demonstrated that, thanks to the hybrid *NVM+VM* approach adopted, it could reliably maintain the EMU state during a power outage and recover it efficiently once the power is available again.

Acknowledgment

This work was supported by EPSRC Grant EP/R511584/1 (IAA). The authors would like to thank David Scott, Darren Mackie, and Carl Samuel for their invaluable support.

References

[1] T. Becker *et al.*, "Energy harvesting for a green internet of things," PSMA, NJ, USA, White Paper, 2021.
[2] A. S. Weddell *et al.*, "A survey of multi-source energy harvesting systems," *DATE'13*, pp. 905–908, 2013.
[3] D. Balsamo *et al.*, "Hibernus: Sustaining computation during intermittent supply for energy-harvesting systems," *IEEE Embedded Syst. Lett.*, vol. 7, no. 1, pp. 15–18, 2014.
[4] D. Balsamo *et al.*, "A control flow for transiently powered energy harvesting sensor systems," *IEEE Sens. J.*, vol. 20, no. 18, 2020.
[5] J. Boukhobza *et al.*, "Emerging nvm: A survey on architectural integration and research challenges," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 23, no. 2, nov 2017.
[6] B. J. Fletcher *et al.*, "Power neutral performance scaling for energy harvesting mp-socs," *DATE'17*, pp. 1516–1521, 2017.
[7] D. Balsamo *et al.*, "Graceful performance modulation for power-neutral transient computing systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 5, pp. 738–749, 2016.
[8] O. B. Akan *et al.*, "Internet of hybrid energy harvesting things," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 736–746, 2017.
[9] D. Carli *et al.*, "An effective multi-source energy harvester for low power applications," *DATE'11*, pp. 1–6, 2011.
[10] A. C. R. Ferreira *et al.*, "A low-power two-stage active rectifier for energy harvesting applications," *IEEE MeMeA'20*, pp. 1–5, 2020.
[11] K. Ali, "Active or-ing solution with lm74610-q1 smart diode controller," in *Texas Instruments Application Report*, 2017.
[12] A. Eltamaly and A. Abdelaziz, *Modern Maximum Power Point Tracking Techniques for Photovoltaic Energy Systems*. Springer, 2020.
[13] A. Gomez *et al.*, "Efficient, long-term logging of rich data sensors using transient sensor nodes," *ACM Trans. Embed. Comput. Syst.*, vol. 17, no. 1, 2017.
[14] M. Magno *et al.*, "Infinitime: Multi-sensor wearable bracelet with human body harvesting," *SUSCOM*, vol. 11, pp. 38–49, 2016.
[15] S. Kim *et al.*, "An adaptive spreading factor selection scheme for a single channel lora modem," *Sensors*, vol. 20, no. 4, 2020.