Exploring Energy Efficient State Retention in Transiently-Powered Computing Systems

Extended Abstract

Theodoros D. Verykios, Domenico Balsamo, Geoff V. Merrett
School of Electronics and Computer Science
University of Southampton
Southampton, UK
{T.Verykios,D.Balsamo,gvm}@soton.ac.uk

ABSTRACT

Batteries have traditionally been used to power embedded electronic devices. However, requirements such as a long lifetime, low cost, and weight, pose significant challenges to battery-powered systems. Energy harvesting offers the potential for embedded systems to operate without batteries. Nonetheless, harvesting has been traditionally coupled with large energy buffers such as supercapacitors to tackle the instability of the source. Transiently-powered computing systems enable computation to be sustained despite the source variability, without the need for additional energy storage. To make this feasible, the system state (e.g. registers and RAM) needs to be saved to Non-Volatile Memory (NVM) before a power outage, and restored once power is available again. Existing transient systems save the entire state of the system upon power failure and do not consider the properties of different NVM technologies, leading into a sub-optimal state retention process. As a consequence, the time and energy spent towards useful computation are decreased significantly, affecting the forward progress that the system can achieve. The aim of this research is to introduce novel methods to reduce the time and energy overhead of the state retention process, exploring solutions both in the software and hardware domain.

CCS CONCEPTS

• **Computer systems organization** → *Embedded software*; Embedded hardware; • **Hardware** → Memory and dense storage;

KEYWORDS

Transient Computing; State Retention; Embedded Systems; Energy Harvesting

1 INTRODUCTION

The number of battery-powered electronic systems has increased dramatically with the vast deployment of mobile, autonomous and wearable devices. Even though batteries act as a virtually unlimited power source for a certain period of time before becoming exhausted, they come at a dimensional, financial, and environmental cost which cannot be neglected. The nature of some applications such as implantable bio-sensors [7] and underground WSNs [5] implies limited access and, consequently, maintenance becomes a serious challenge. In addition, even though CMOS devices have been downscaled significantly over the last years, the energy density of batteries has not been increased accordingly. The realisation of some devices such as implantable bio-sensors (which need to be miniaturised) was challenging due to the size of batteries. Therefore, the need for systems that can operate without batteries has emerged [1].

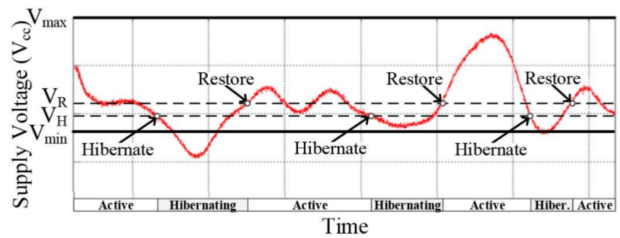


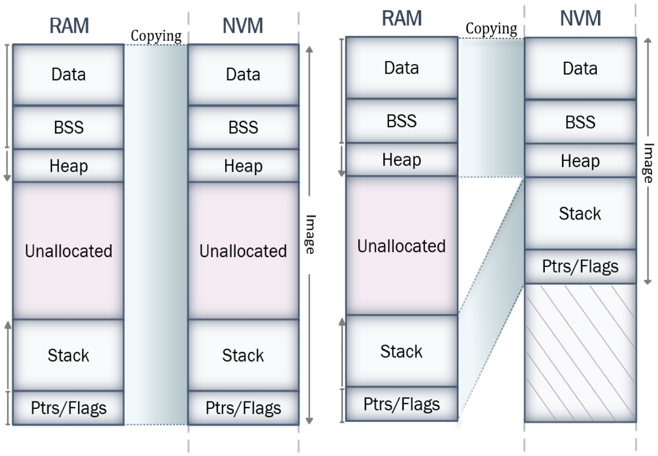
Figure 1: Operation of a typical transient system [2]

2 TRANSIENT SYSTEMS

Energy harvesting (EH) systems scavenge energy from environmental sources such as light, vibration, motion or temperature to power themselves, instead of relying on batteries [3]. Nevertheless, factors such as the weather condition, availability of light, or the intensity of vibration can significantly affect the energy availability. Relying solely on these sources can therefore result in the system being unable to sustain computation.

Transiently-powered computing systems are storage-less systems that enable computation to be sustained, despite the variable and unstable energy harvested from the environment [6]. Transient systems retain their state in Non-Volatile Memory (NVM) upon a power failure to cope with frequent power interruptions. This implies that the main memory and registers are saved before a power outage, and restored when the power is available again, as shown in Figure 1 which shows the operation of a typical transient system. Several software-based approaches such as Hibernus [2] have been proposed for transient computing; however, they all save the entire volatile state without considering what is actually required. Furthermore, using a universal policy, without regard for the NVM technology, results in considerable time and energy overhead for the state retention process.

Recently, Bhatti et al. [4] proposed a selective policy for efficient state retention which dynamically identifies the unallocated space and only saves to Flash memory the parts of the main memory being used by the main application. We implemented this "Allocated State" policy on two platforms with different NVM technologies (FRAM and Flash). Figure 3 shows the Allocated State policy applied to different applications, where the time and energy required to save



(a) Complete State [2] **(b) Allocated State [4]**

Figure 2: Existing state retention approaches illustrating a) Complete State and b) Allocated State

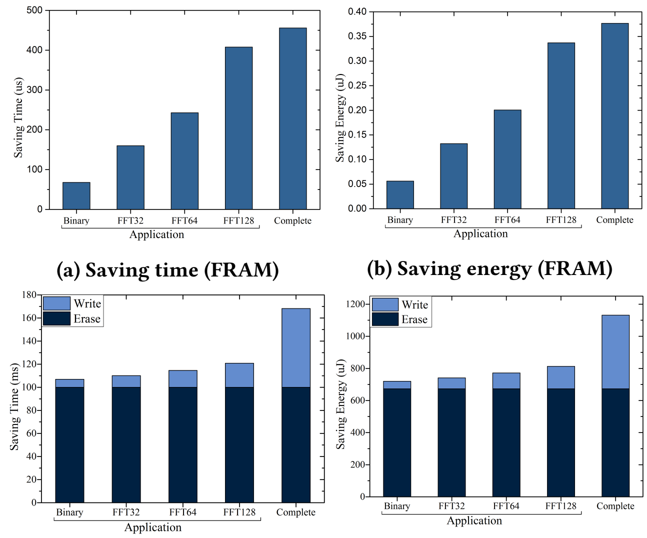
the system state are experimentally measured. This policy allows for substantial energy time and energy savings when used with NVM technologies that do not require erasing, such as FRAM. Figures 2a and 2b demonstrate that the cost for saving is proportionally reduced with the size of allocated memory, when compared to saving the entire memory (up to 85.1% reduction when the memory usage is 18%). However, when this policy is implemented using Flash memory, it is far less effective as shown in Figures 2c and 2d. This is because the overhead due to the erasing process (typical for Flash memory) that is needed before saving the system state, is not taken into account.

3 SELECTIVE POLICIES

Our current research focuses on devising novel selective policies for efficient state retention to improve the energy efficiency of transiently-powered computing systems. These policies exploit the characteristics of different NVM technologies, to ensure that the state retention is an energy and time efficient operation. In addition, the memory usage by the main application is taken into account when designing these policies. Therefore, the policies are tailored to the specific characteristics of each NVM type and the memory usage and consequently, the energy savings can be maximised.

A fundamental challenge is the integration of these policies in state-of-the-art transient systems in order to show the benefits against a universal policy. Using these policies, the overhead for saving/restoring the system state can be reduced which can lead into more energy efficient transient systems.

One of the most important parameters of transient systems is the threshold voltages at which the system starts its state retention process or restores the system state in order to continue its operation. Our future work will be concerned with optimising these thresholds by integrating these policies, in order to dynamically maximise the useful computation time and therefore, ensure that transient systems become more energy efficient.



(a) Saving time (FRAM) **(b) Saving energy (FRAM)**

(c) Saving time (Flash) **(d) Saving energy (Flash)**

Figure 3: Time and energy overhead of Allocated State with FRAM and Flash memories

4 CONCLUSIONS

In this extended abstract, the need for energy efficient state retention policies was highlighted. Software based policies are investigated which are targeted for different NVM technologies, exploiting their individual properties and the way the memory is used by the main application. Future work will focus on predicting the energy needs before saving the system state in order to maximise the time and energy spent on useful computation.

ACKNOWLEDGMENTS

This work was supported in part by UK Engineering and Physical Sciences Research Council (EPSRC) Platform Grant EP/P010164/1. Experimental data accompanying this work can be found at DOI:10.5258/SOTON/D0276 (http://doi.org/10.5258/SOTON/D0276).

REFERENCES

[1] D. Balsamo, A. Das, A. S. Weddell, D. Brunelli, B. M. Al-Hashimi, G. V. Merrett, and L. Benini. 2016. Graceful performance modulation for power-neutral transient computing systems. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 35, 5 (2016), 738–749.

[2] D. Balsamo, A. S. Weddell, A. Das, A. R. Arreola, D. Brunelli, B. M. Al-Hashimi, G. V. Merrett, and L. Benini. 2016. Hibernus++: a self-calibrating and adaptive system for transiently-powered embedded devices. *IEEE Trans. Comp.-Aided Des. Integr. Circuits Syst.* 35, 12 (2016), 1968–1980.

[3] S. Beeby and N. White. 2010. *Energy harvesting for autonomous systems.* Artech House.

[4] N. A. Bhatti and L. Mottola. 2016. Efficient State Retention for Transiently-powered Embedded Sensing.. In *EWSN 2016.* Graz, Austria, 137–148.

[5] S. Kahrobaee and M. C. Vuran. 2013. Vibration energy harvesting for wireless underground sensor networks. In *IEEE Int'l Conf. on Communications (ICC) 2013.* IEEE, 1543–1548.

[6] K. Ma, Y. Zheng, S. Li, K. Swaminathan, X. Li, Y. Liu, J. Sampson, Y. Xie, and V. Narayanan. 2015. Architecture exploration for ambient energy harvesting nonvolatile processors. In *IEEE 21st HPCA 2015.* IEEE, San Francisco, CA, USA, 526–537.

[7] J. Olivo, S. Carrara, and G. De Micheli. 2011. Energy harvesting and remote powering for implantable biosensors. *IEEE Sensors Journal* 11, 7 (2011), 1573–1586.