

# Ambient Energy Harvesting Nonvolatile Processors: From Circuit to System \*

Yongpan Liu<sup>1</sup>, Zewei Li<sup>1</sup>, Hehe Li<sup>1</sup>, Yiqun Wang<sup>1</sup>, Xueqing Li<sup>2</sup>, Kaisheng Ma<sup>2</sup>, Shuangchen Li<sup>3</sup>

Meng-Fan Chang<sup>4</sup>, Sampson John<sup>2</sup>, Yuan Xie<sup>3</sup>, Jiwu Shu<sup>1</sup>, Huazhong Yang<sup>1</sup>

<sup>1</sup>Department of Electronic Engineering / Computer Science and technology, Tsinghua University, Beijing 100084

<sup>2</sup>Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802

<sup>3</sup>Department of Electronic and Computer Engineering, University of California, Santa Barbara, CA 93106

<sup>4</sup>Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013

ypliu@tsinghua.edu.cn, mfchang@ee.nthu.edu.tw, yuanxie@ece.ucsb.edu

(Invited)

## ABSTRACT

Energy harvesting is gaining more and more attentions due to its characteristics of ultra-long operation time without maintenance. However, frequent unpredictable power failures from energy harvesters bring performance and reliability challenges to traditional processors. Nonvolatile processors are promising to solve such a problem due to their advantage of zero leakage and efficient backup and restore operations. To optimize the nonvolatile processor design, this paper proposes new metrics of nonvolatile processors to consider energy harvesting factors for the first time. Furthermore, we explore the nonvolatile processor design from circuit to system level. A prototype of energy harvesting nonvolatile processor is set up and experimental results show that the proposed performance metric meets the measured results by less than 6.27% average errors. Finally, the energy consumption of nonvolatile processor is analyzed under different benchmarks.

## Keywords

Nonvolatile Processor, Energy Harvesting, Design Metrics

## 1. INTRODUCTION

With the increase in popularity of Internet of Things (IoT), implantable and wearable devices, more and more computational ability are required from those energy-limited systems, where lifetime becomes the most critical design issues. Given the battery weight or volume as the limiter, people had developed plenty of low-power techniques to extend the operating time. However, the rather slow increasing battery capacity and fast increasing demands on the computational power requires more effective solutions. Furthermore, it is predicted that the era of trillion sensors will come, which means the number of sensor will increase very fast in the near future. To power a huge amount of sensors with batteries will lead to expensive maintenance and polluted environment.

\*This work was supported in part by High-Tech Research and Development Program 2013AA01320 and the Importation and Development of High-Caliber Talents Project YETP0102 and the Center for Low Energy Systems Technology (LEAST) sponsored by MARCO and DARPA, and by the NSF awards 1160483 (ASSIST), 1205618, 1213052, 1461698, and 1500848. Thanks to the suggestion and assistance of Xiao Sheng, Daming Zhang, Prof. Suman Datta and Vijaykrishnan Narayanan.

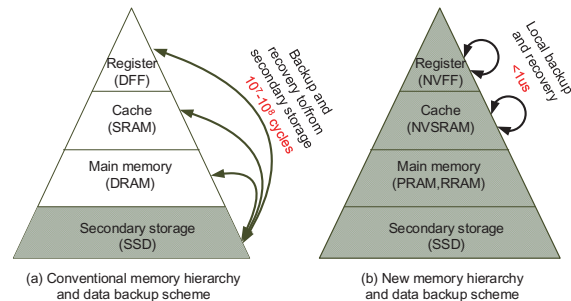
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

DAC '15, June 07 - 11, 2015, San Francisco, CA, USA  
Copyright 2015 ACM 978-1-4503-3520-1/15/06...\$15.00  
<http://dx.doi.org/10.1145/2744769.2747910>.

Therefore, battery-less systems which harvest ambient energy have been proposed to be the next step in the evolution of IoT.

Battery-less energy harvesting systems manifest strong vitality recently due to their characteristics of ultra-long operation times without maintenance. Moreover, they are environmentally friendly. Therefore, energy harvesting systems have been extensively used in various fields such as habitat monitoring, wireless health and structural monitoring [1]. A typical energy harvesting system consists of an energy harvester to collect the ambient energy and workloads including a processor, peripheral sensors and wireless transceivers. According to [2], solar, thermal, wireless and vibration energy are four commonly used harvesting sources. Different from the conventional battery powered scenarios, there are several challenging characteristics for those systems as follows.

1) *Low power supplement*: the output power ranges from several to hundreds of microwatt due to the small size and limited conversion efficiency of energy harvesters; 2) *Unstable power output*: the power failures frequently happen and the amplitude and power level vary significantly; 3) *Hard to predict*: the harvested power trace depends on many factors, such as the vibration pattern, environmental condition and temperature difference, etc. It is quite hard to accurately predict the harvested energy in future.



**Figure 1: Comparison of memory hierarchy between volatile and nonvolatile processors**

The frequent unpredictable power failures make traditional processors suffer from either many operating rollbacks or large backup overheads in energy harvesting environments. As Figure 1 has shown, the state backup in volatile processor will cause slow and energy-consuming data movements to the nonvolatile secondary storage. Therefore, an efficient processor for energy harvesting is badly needed.

## 2. OVERVIEW

In this section, nonvolatile processors are proposed to handle the challenges arising in energy harvesting applications. We first explain the concept of nonvolatile processor and its advantages compared with the traditional ones. After that, we show the roadmap to explore the usage of nonvolatile processor in

energy harvesting applications. Finally, novel design metrics for nonvolatile processors are given out.

## 2.1 What Is Nonvolatile Processor

As Figure 1 has shown, nonvolatile processors realize in place backup by adopting nonvolatile registers and nonvolatile SRAM. Thus, they have the following advantages: 1) *Nearly Zero Leakage*: The traditional volatile processors have to keep the power supply to sustain the memory state with nontrivial leakage power. On the contrary, the nonvolatile processor can be shut down to achieve zero leakage as soon as the computation is completed, while the state is kept in the nonvolatile memory; 2) *Efficient Backup/Restore Operation*: In the traditional processors, the system state has to be stored in the cross-layer secondary storage, which is slow and power hungry. The nonvolatile processor can realize in place backup, which is  $2\text{--}4 \times$  magnitudes better than the up-to-date commercial processors [3]; 3) *Resilience to Power Failures*: Nonvolatile processors have rather fast backup/restore process, which can be easily powered with a quite small capacitor; 4) *High Performance and Low Energy*: The nonvolatile processor can backup intermediate results and keep continuous forward progress, while the volatile processor needs rollback due to the state loss. In a word, nonvolatile processors provide several advantages compared with traditional ones and they are a promising candidate in energy harvesting systems.

## 2.2 Systematical Exploration of Energy Harvesting Nonvolatile Processor

In order to mitigate the performance problems caused by volatile systems, we explore the design of nonvolatile processor in circuit, architecture and system levels shown in Figure 2.

In the circuit level, we need to design efficient backup circuits for the nonvolatile processor. Since the nonvolatile memory faces limited endurance and asymmetric read/write operations, nonvolatile flip-flops and SRAM with hybrid structures are proposed. Furthermore, high-performance and area-efficient nonvolatile controller is designed to synchronize wakeup/sleep operations. Finally, voltage detection circuits are investigated to identify the power failures by considering the tradeoff between speed and reliability.

In the architecture level, we investigate the nonvolatile backup units in different cases, where various operating and control data are needed in the backup. In those architectures, the key is to balance the backup overhead and performance savings. Second, the power supply architecture is reconsidered for better efficiency, to meet the situation that the backup/restore overhead is quite small.

In the system level, we explore the possibility to design nonvolatile processors with different circuits, architecture and process. To obtain the best performance improvement, we discuss the proper software techniques and scheduling methods to make the nonvolatile processor operate efficiently.

## 2.3 Design Metrics

Compared with the volatile processors, the design metrics of nonvolatile processor should take the factor of energy harvesting into consideration, which makes them different from conventional ones. We define the new metrics of performance, power and reliability as follows.

### 2.3.1 Performance Metrics

We define the nonvolatile processor (NVP) CPU time to model the performance of nonvolatile processors. We assume that the power input is a square waveform with frequency  $F_p$  and duty cycle  $D_p$ . Usually, the shape of power supply in practice can be approximated as square waveforms under different periods.

**DEFINITION 1. NVP CPU time:** *The run time of a program on a nonvolatile processor with an intermittent power supply modeled as a  $(F_p, D_p)$  is denoted as  $T_{NVP}$ .*

For a nonvolatile processor with operating frequency  $f$ , backup

time  $T_b$  and recovery time  $T_r$ ,  $T_{NVP}$  can be calculated as follows:

$$T_{NVP} = \frac{CPI \times I}{f(D_p - F_p(T_b + T_r))} \quad (1)$$

where  $CPI$  represents the cycles per instruction and  $I$  is the instructions number. We assume that  $D_p > F_p(T_b + T_r)$  is satisfied, which means the duty cycle length is larger than the state transition time. The correctness of Equation 1 is validated by the measured results in Section 6. Besides conventional performance optimization techniques, such as instruction level parallelism, overclocking, the performance of nonvolatile processors can be improved in two new aspects. From the perspective of energy harvesting, we can increase the duty cycle  $D_p$  or decrease the frequency  $F_p$  for better  $T_{NVP}$  by optimizing the design of energy harvesters. From the hardware perspective, we can shorten the transition time  $T_b$  and  $T_r$  by improving the read/write speed of nonvolatile flip-flops.

### 2.3.2 Energy Efficiency Metrics

Different from the previous energy efficiency of volatile processor defined as the energy consumption per operation, we define a new metric for nonvolatile processor as follows:

**DEFINITION 2. NV energy efficiency:** *The ratio of the amount of energy used for normal execution to the total energy collected by the energy harvester with intermittent power supplement, is denoted as  $\eta$ .*

As the definition has shown, the NV energy efficiency depends on both energy harvesting efficiency  $\eta_1$  and the execution efficiency  $\eta_2$  of nonvolatile processor. The former is related to the capacitor size, voltage regulator efficiency and the charging/discharging policy. Intuitively, large capacitor usually leads to a lower  $\eta_1$  due to low capacitor voltage and larger regulator loss. The latter can be expressed as follows:

$$\eta_2 = \frac{E_{exe}}{E_{exe} + (E_b + E_r)N_b} \quad (2)$$

where  $E_{exe}$  is the total execution energy of program,  $E_b$  ( $E_r$ ) is the backup (recovery) energy and  $N_b$  is the backup times. It is obvious that  $\eta_2$  becomes better if  $N_b$  decreases. Therefore, we tend to use a large capacitor to reduce  $N_b$  for better  $\eta_2$ . However, the previous analysis on  $\eta_1$  shows that the smaller the better. Therefore, from the point of  $\eta$ , a tradeoff design should consider the effects of both parts. Moreover,  $E_b$  and  $E_r$  varies under different benchmarks and backup points because of backup data volume and inputs. Those effects are shown in Section 6.

### 2.3.3 Reliability Metrics

Considering the failures from energy harvesting, we define the mean time to failure (MTTF) as the reliability metric as follows:

**DEFINITION 3. (MTTF of NVPs):** *The mean time before a failure occurs caused by hardware malfunctions or backup (recovery) faults, denotes as  $MTTF_{nvp}$ .*

$$\frac{1}{MTTF_{nvp}} = \frac{1}{MTTF_{system}} + \frac{1}{MTTF_{b/r}} \quad (3)$$

where  $MTTF_{b/r}$  is the MTTF induced by backup or recovery failure and  $MTTF_{system}$  is the MTTF of traditional computing system.  $MTTF_{b/r}$  is related to the power trace distribution, backup strategies and capacitor parameters. Given a reliability constraint, the MTTF can be satisfied by tuning the above factors.

## 3. EFFICIENT BACKUP CIRCUITS

Before discussing the critical backup circuits for nonvolatile processor, we first illustrate the typical backup and restore diagram of a nonvolatile processor in Figure 3. The backup sequence is as follows: The reset IC detects the voltage of the bulk capacitor. When the voltage is lower than a threshold, the reset IC generates a reset signal to the nonvolatile controller, which gates the clock to hold the state. The nonvolatile controller then generates a sequence of control signals to perform write actions to NVFFs and nvSRAM.

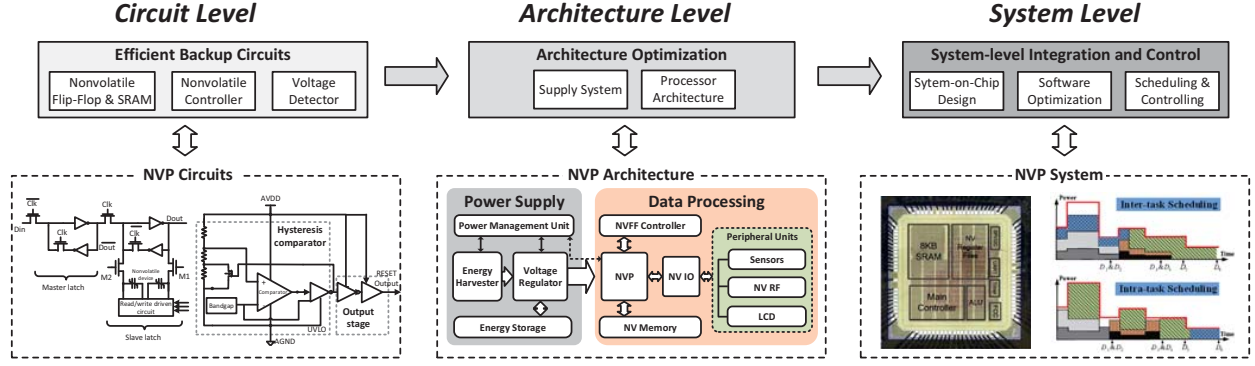


Figure 2: A holistic approach for nonvolatile processor: circuit-level, architecture-level, and system-level design space exploration

Finally, NVFFs and nvSRAMs drive internal nonvolatile devices to memorize the system states.

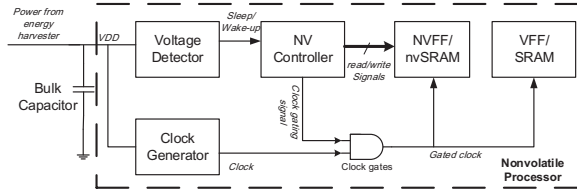


Figure 3: The typical backup and restore diagram of a nonvolatile processor

Three kinds of circuits play important roles in the backup sequence. The first is nonvolatile units, including NVFFs and nvSRAM cells. They provide fast and low power backup/recovery to the nonvolatile processor. The second is the nonvolatile controller, which generates control signals for NVFFs and nvSRAM. The third is the voltage detector, which detects voltage changing in a fast and reliable way. We discuss these circuits as follows.

### 3.1 Nonvolatile Flip-Flop

Since the nonvolatile devices suffer from writing performance loss and limited endurance, most nonvolatile processors adopt the hybrid structure of nonvolatile flip-flops in Figure 4. The main concept is to isolate the nonvolatile device from the CMOS flip-flop with switches (M1 and M2), and the nonvolatile devices perform store/recall operations only when power failures happen.

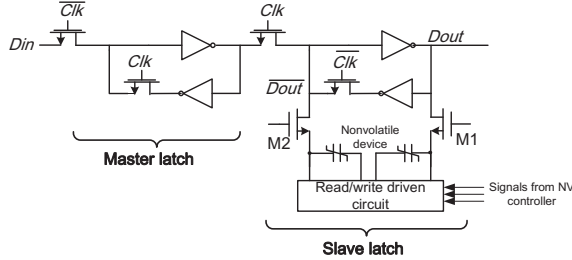


Figure 4: The typical structure of a hybrid NVFF

Various types of emerging memory devices have been employed in the NVFF, including FeRAM, STT-MRAM, RRAM, IGZO [3, 4, 5, 6, 7, 8]. Table 1 compares their performance of data store and recall. The fastest store and recall time is reduced to several nanoseconds and the energy is below  $10\text{pJ/bit}$ .

### 3.2 Nonvolatile SRAM

Figure 5(b) illustrates the concept of nonvolatile-SRAM (nvSRAM), which forms a direct bit-to-bit connection between a SRAM cell and NVM devices within a single cell. The nvSRAM achieves fast parallel data transfer and fast store/restore than 2-macro schemes (see in Figure 5(a)).

Previous work adopted various types of emerging memory devices, such as FeRAM, STT-MRAM (MTJ), PCRAM, and

Table 1: Comparison of NVFFs using different nonvolatile devices

NV device	Feature Size	Store time	Recall time	Store energy	Recall energy
FeRAM[6]	130nm	40ns	48ns	2.2pJ/bit	0.66pJ/bit
STT-MRAM[5]	65nm	4ns	5ns	6pJ/bit	0.3pJ/bit
RRAM[7]	45nm	10ns	3.2ns	0.83pJ/bit	N.A.
CAAC-IGZO[8]	1μm	40ns	8ns	1.6pJ/bit	17.4pJ/bit

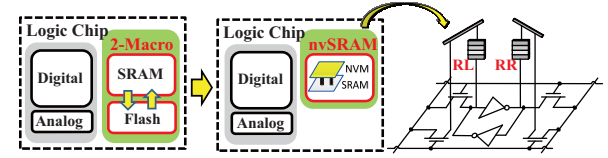


Figure 5: Structure of (a) 2-macro and (b) nvSRAM

RRAM (memristor) in nvSRAM cells [9, 10, 7, 11, 12, 13, 14, 15]. The 4T2R and 7T2R nvSRAMs [9, 11, 12, 14] achieve small cell area at the expense of significant DC-short current at storage nodes (Q and QB). To cut off this DC-short current at storage nodes, extra transistors are required at the expense of larger cell area for nvSRAMs[7, 13, 15]. Figure 6 compares the cell structure and performance of selected nvSRAM works using the same NVM device. Since each structure has its advantages and disadvantages, there are challenges remained to achieve low energy, compact area and robust store/restore operation for nvSRAMs.

### 3.3 Nonvolatile Controller

The nonvolatile controller provides the read and write signals to the nonvolatile devices, and controls the sequence of backup and recovery. In a nonvolatile processor, the direct way to perform the fastest backup is to drive all the nonvolatile devices in parallel. The all-in-parallel (AIP) method always employs a centralized nonvolatile controller. However, it leads to a large peak backup current and area overhead when the number of NVFFs is large. Moreover, it induces high fan-out of nonvolatile controller and difficulties to test. Therefore, some works were proposed to alleviate such problems.

A parallel compare and compression (PaCC) control scheme is proposed [16] to save chip area. This scheme compresses the system states before backup, and reduces the number of NVFFs by over 70%. However, the PaCC causes more than 50% backup time overhead. Another compression based control scheme, called SPaC, is proposed to speed up the compression of PaCC [17]. It uses a block-level parallel compression architecture to achieve up to 76% compressing speed with only 16% area overhead. One more block-level parallel NVFF controlling is proposed in [6]. It uses a NVL-Array based nonvolatile storage architecture and

Cell Structure	6T2C[9]	6T4C [10]	8T2R[7]	4T2R[11]	7T2R[12]	7T1R[13]	6T2R [14]
Cell Schematic							
SRAM-mode DC Short Current	No	No	No	Yes	Yes	No	Yes
Cell Area (A)	1.17x	1.77x	1.26x	0.67x	1.12x	1.12x	1x
Store Energy ( $E_s$ )	2x	4x	2x	2x	2x	1x	2x
Technology	0.25um+FRAM	0.35um+FRAM	0.18um + RRAM	0.18um + MTJ	0.18um + RRAM	90nm + RRAM	90nm + RRAM

Figure 6: Cell structure and performance of selected nvSRAM works

this architecture simplifies the control circuit and enables NVFF testability by centralizing the NVFF placement.

The future work of nonvolatile controller will focus on the tradeoff between backup speed, peak power and reliability. Moreover, the co-optimization of both NVFFs and nvSRAM controlling will be an interesting topic.

### 3.4 Voltage Detector

Figure 7 shows the wakeup sequence and its breakdown of a nonvolatile processor under real measurements. The delay of reset IC introduces up to 34% of the total wakeup time. The commercial reset IC [18] always needs a delay to prevent detection fault to happen when there are noises on the power line. However, this part can be eliminated if we use a concrete voltage detector design for the energy harvesting applications. The wakeup time can be further reduced by designing proper nonvolatile controller, reset IC and sizing the capacitor at the expense of reliability issues.

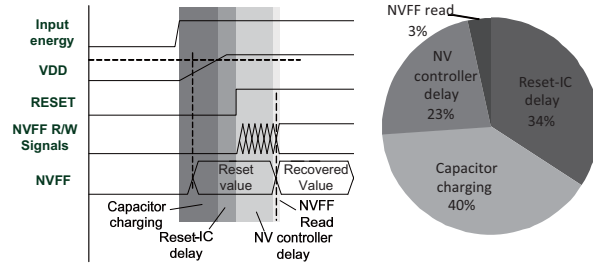


Figure 7: Breakdown of wake-up time

## 4. ARCHITECTURE OPTIMIZATION

Architecture exploration of nonvolatile processor includes the energy harvesting supply system and the processor architecture dedicated for the energy harvesting applications.

### 4.1 Supply System

Figure 8 shows a typical supply system in energy harvesting applications. Generally, the widely used ambient energy sources include the radio-frequency (RF) signal, piezoelectric energy, photovoltaic cells, and thermoelectric devices [19, 20, 21]. These energy sources require different power conversion techniques to get the DC power. For example, RF and piezoelectric energy require a rectifier [19, 22] for the AC-DC conversion, while photovoltaic and thermoelectric power is direct current. Additional DC-DC converters and low-dropout regulators (LDO) could also be used for more voltage levels.

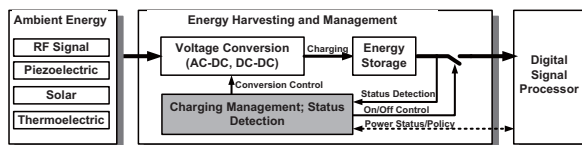


Figure 8: A typical supply system for energy harvesting applications

One challenge in using the ambient energy is its erratic and unreliable nature. Even with nonvolatile processors, an intermediate energy storage element, i.e. a capacitor, should be used to mitigate the effect of temporary power failures [23, 2, 24, 25, 26, 3, 5]. The capacitor affects the system performance and should be optimized based on the input power trace characteristics, the processor power and the applications [2]. Another challenge is how to alleviate the efficiency degradation when the environment or the load changes. In this case, various maximum power point tracking techniques (MPPT) have been applied by explicitly or implicitly configuring the power converter input impedance [23, 27, 28, 29, 30].

In previous designs, emphasized efforts in lowering the power of digital processors to meet the weak ambient energy sources results in the waste of extra input power. An additional path or capacitor is proposed to store the extra energy for future usage [19, 23]. However, this does not guarantee the complete usage of the otherwise wasted energy due to the inevitable non-ideal power conversion. In fact, if the processor could be dynamically tuned to meet the input power trace, no such additional path is necessary while more computation could be carried out with less energy loss.

### 4.2 Processor Architecture

Energy harvesting demands architectural-level redesign to provide continuous computation despite power interruptions. Significantly different from traditional processors targeting at low power, energy harvesting nonvolatile processor focuses mainly on the goal of maximum forward progress, because unused energy will be wasted by leakage. Three different types of optimizations have been proposed recently [2].

1) *The selection of backup and recovery data.* For the non-pipelined structure with a fixed amount of backup data, a volatile flag could be used to omit redundant backups. For a pipelined structure, the tradeoff is to backup more data for less rollbacks at the cost of more backup overhead. For a more complex out-of-order (OoO) processor, there is a similar tradeoff between the rollbacks and the backup overhead. It has been revealed that an optimum selection of backup data exists while taking both backup and recovery energy consumption into account.

2) *The backup frequency.* As backup and recovery operations consume energy, checkpointing at a fixed frequency guarantees less worst-case rollbacks at the cost of power. On-demand backup with voltage detector is power efficient because it is performed only when there is a power outage. However, checkpointing is better when the power failures are frequent and periodic.

3) *Adaptive architecture under varying power profiles.* As different processor architectures achieve the best forward progress under different power traces, an adaptive architecture based on the power trace is a promising solution to achieve the maximum forward progress and the highest energy usage efficiency. For example, a simple non-pipelined architecture is suitable for weak power with frequent power failures, while a fast OoO processor may achieve the maximum forward progress with a higher input power and less frequent power failures, even though it requires the highest power threshold.



## 5. SYSTEM INTEGRATION AND CONTROL

### 5.1 System-on-Chip Design

Recently, some chips of nonvolatile processors are designed based on different nonvolatile technologies, including FeRAM, STT-MRAM and RRAM [3, 4, 5, 6, 7]. The first nonvolatile processor chip, THU-1010N, was based on ROHM's ferroelectric technology [3]. It achieves  $7\mu s$  data backup time, 100x faster than the up-to-date commercial microprocessors. After that, researchers in MIT and TI fabricated FeRAM based nonvolatile processors [4, 6] and reduced the backup and restore time to hundreds of nanoseconds. Tohoku university and NEC fabricated an STT-MRAM based nonvolatile processor [5], which achieved  $4ns$  backup time and  $5ns$  restore time.

All of above chips dedicate to shorten the backup and restore time of nonvolatile processor, which is suitable for fine-grained power gating. However, their designs assume that the peripheral circuits, such as clock, power converter, etc., are powered on. Therefore, those chips are inefficient for energy harvesting applications, where both processor and peripheral circuits are powered off. In that case, the wakeup time of peripheral circuits dominates that of NVFFs (see in Figure 7). Thus, future research work should be investigated on how to reduce the wakeup time of peripheral circuits besides the nonvolatile processor itself.

### 5.2 Software Optimization

To fully unveil the advantage of nonvolatile processors, the software should be redesigned to leverage considerable area overheads, introduced by the nonvolatile memory. [31] provides a novel register allocation algorithm to minimize the critical data overflows in a hybrid nonvolatile register architecture. [32] analyzes the program execution path and identifies the reachable positions where a much smaller state should be saved. By sharing the corresponding address space of the caller function and the callee function's frames, [33] proposes a compiler directed stack trimming strategy to reduce the size of program state.

On the other hand, the traditional software may also cause nontrivial energy and runtime overheads under the unstable ambient power. New software rules should be developed. For instance, the conventional programs on the volatile processor re-initialize their peripheral devices every time, which is unnecessary for nonvolatile processors. The software on nonvolatile processors should avoid this by analyzing the data flow pattern and suitable checkpointing. Moreover, if the power failures happen during data transmission between different nonvolatile devices, they may cause data inconsistency and lead to irreversible computation errors. Systematic consistency-aware checkpointing mechanism [34] and new software resetting technique are investigated to correct these errors while optimizing the performance and energy efficiency.

### 5.3 Scheduling and Controlling

To achieve good quality of service (QoS) for real-time tasks running on nonvolatile processors, proper scheduling algorithm should be developed. Different from the conventional power supply system, the nonvolatile sensor nodes are powered by the storage-less and converter-less power supply system (e.g., [28], [23]). Present algorithms (e.g., LSA [35], DVFS [36], etc.) are based on inter-task scheduling and focus on the single period, which are not suitable for the NVP-based sensor nodes.

Without energy buffers and long term QoS considerations, these algorithms suffer from quite uncertain execution delays and lower QoS. [37, 38] proposes a long term intra-task scheduling algorithm, which supports task scheduling at any time during the execution with positive energy migration. In the algorithms, trigger mechanisms are developed to select scheduling points. Artificial neural networks (ANNs) based task priority calculation are performed for the online task scheduling, whose parameters are offline trained by static optimal scheduling samples.

## 6. CASE STUDY: ENERGY HARVESTING NONVOLATILE SENSING PLATFORM

### 6.1 Platform Setup

We design an ambient energy harvesting sensing platform based on an actual fabricated processor, as shown in Figure 9. Table 2 illustrates the system specifications.

Table 2: The parameters of prototype

Parameter	Value	Parameter	Value
Energy harvester	Solar	Nonvolatile Processor	THU1010N
Process Technology	$0.13\mu m$	Core Architecture	8051-based
Nonvolatile technology	Ferroelectric	Nonvolatile Memory	NVFF and FeRAM
Nonvolatile RegFile	128 bytes	FRAM Capacity	2M bits
Max. clock	25MHz	MCU power	$160\mu W @ 1MHz$
Backup Energy	$23.1nJ$	Recovery Energy	$8.1nJ$
Backup Time	$7\mu s$	Recovery Time	$3\mu s$

On this platform, the sensor node is powered by a solar panel. The nonvolatile processor (THU1010N) adopts an 8051-based CISC-like architecture. The instructions are stored in an off-chip Flash and the register data is stored in the on-chip ferroelectric flip-flops. Besides, an FeRAM chip is connected to the processor through the SPI interface. It is used to store the sensing data and intermediate computation data, which is too large for the on-chip memory to store. The power failure detecting circuit generates a pulse when the voltage falls below a threshold to trigger the backup process. We adopt the I2C bus interface to connect the processor and the sensors.

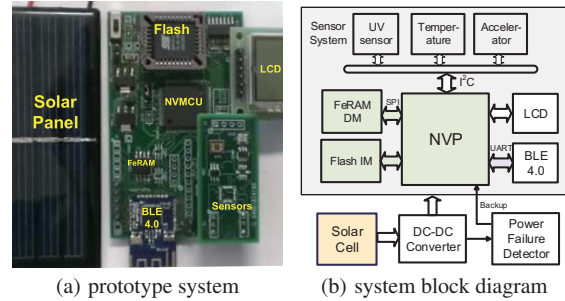


Figure 9: The prototype and platform architecture

### 6.2 Experimental Results

We implemented six real-life sensing applications on this platform to verify the design metrics discussed in Section 2. Without losing generality, we use a FPGA board to generate a 16kHz square waveform to model the intermittent power supply with tunable duty cycles. The system operates at a 1MHz clock frequency. Moreover, we also created a nonvolatile processor simulator based on the GEM5 platform to explore the influence of different power traces on system performance and energy efficiency.

#### 6.2.1 Performance Evaluation

We demonstrate that the nonvolatile sensing platform works correctly under very fast power failure conditions. As shown in Table 3,  $D_p$  is the duty cycle of the square waveform. When  $D_p$  reaches 100%, the system operates with no power failures. The average difference between the executing time calculated based on Equation 1 and the measurements on the prototype system is 6.27%, and the maximum error is 10.4%. As we can see, the maximum error comes from the case when the duty cycle becomes shorter. We believe that the errors come from the less accurate modeling. Based on the experiments, more factors in the backup and restore process, such as clock jitters and power traces

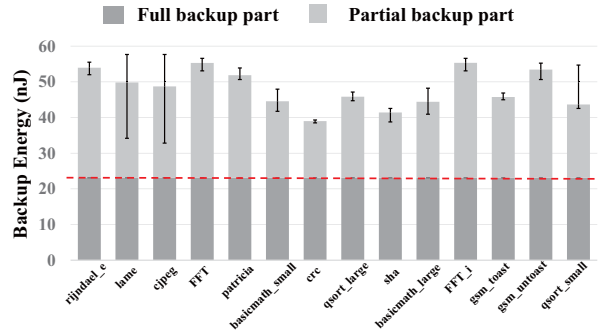
deviations, should be taken into account. Those factors have a larger influence when the duty cycle is short.

**Table 3: Performance metrics comparison between analytical and measured results under a 16kHz square waveform power supply with different duty cycles**

	FFT-8 /ms		FIR-11 /ms		KMP /ms		Matrix /s		Sort /ms		Sqrt /ms	
$D_p$	Sim. Mea.	Sim. Mea.	Sim. Mea.	Sim. Mea.	Sim. Mea.	Sim. Mea.	Sim. Mea.	Sim. Mea.	Sim. Mea.	Sim. Mea.	Sim. Mea.	Sim. Mea.
10%	239	264	17.6	19.6	201	223	6.52	7.23	1587	1760	147	164
20%	81.6	87.9	6.03	6.51	68.7	74.3	2.23	2.41	543	585	50.3	54.6
30%	49.2	49.4	3.64	3.67	41.4	41.8	1.35	1.36	327	330	30.4	30.7
40%	35.2	35.9	2.61	2.67	29.7	30.4	0.96	0.98	234	239	21.7	22.3
50%	27.4	27.3	2.03	2.02	23.1	23.1	0.75	0.75	183	182	16.9	16.9
60%	22.5	22.6	1.66	1.68	18.9	19.1	0.61	0.62	149	151	13.9	14.0
70%	19.0	19.3	1.41	1.43	16.0	16.3	0.52	0.53	127	129	11.7	12.0
80%	16.5	16.5	1.22	1.22	13.9	13.9	0.45	0.45	110	110	10.2	10.2
90%	14.6	14.6	1.08	1.09	12.3	12.4	0.40	0.40	96.8	97.6	8.98	9.10
100%	12.4	12.4	0.92	0.92	10.4	10.4	0.34	0.34	82.5	82.5	7.65	7.65

### 6.2.2 Energy Consumption

The nonvolatile processor simulator is used to study the energy consumption under different power traces. Figure 10 shows the backup energy of different benchmarks from Mibench [39]. We forward 10M instructions for cache warmup and execute 50M instructions for evaluation. Twenty backup points are uniformly selected for each benchmark. Each bar represents the average backup energy. The backup energy consists of two parts: the fixed energy part consumed by the full backup hardware region (all NVFF) and the alterable energy part consumed by the partial backup hardware region (nvSRAM). In this case, we assume that the nonvolatile RegFile adopts a full backup strategy and the nvSRAM uses a partial backup policy [40]. We observe that the average backup energy varies a lot among different benchmarks. Moreover, the backup energy also varies inside a single benchmark, as shown by the variation bars. These variations provide us with the potential of both intra-task and inter-task backup point adjustments so as to improve the energy efficiency.



**Figure 10: Backup energy for different benchmarks in Mibench [39]**

## 7. CONCLUSIONS

In this paper, we define new metrics of performance, power and reliability for nonvolatile processors for the first time, and we explore the design space of nonvolatile processor from circuit to system. Various efficient backup circuits, including NVFF, nvSRAM, nonvolatile controller and voltage detector, are discussed. The architectures of processor and supply system are further investigated. Finally, system-on-chip of nonvolatile processors and related software optimization are presented. Based on a fabricated nonvolatile processor, we build a prototype to validate

the performance metric with the measured results, as well as the energy efficiency. Future work includes system-level design space exploration for nonvolatile processor and software optimizations, such as nonvolatile operating system and communication protocol and so on.

## 8. REFERENCES

- [1] S. Sudevalayam and P. Kulkarni. Energy harvesting sensor nodes: survey and implications. *CST*, 13(3):443–461, 2011.
- [2] K. Ma, Y. Zheng, et al. Architecture exploration for ambient energy harvesting nonvolatile processors. In *HPCA*, pages 526–537, 2015.
- [3] Y. Wang, Y. Liu, et al. In *A 3us wake-up time nonvolatile processor based on ferroelectric flip-flops*, pages 149–152, 2012.
- [4] M. Qazi, A. Amerasekera, and A. P. Chandrakasan. In *A 3.4pJ FeRAM-enabled D flip-flop in 0.13 um CMOS for nonvolatile processing in digital systems*, pages 192–193, 2013.
- [5] N. Sakimura, Y. Tsuji, et al. A 90nm 20mhz fully nonvolatile microcontroller for standby-power-critical applications. In *ISSCC*, pages 184–185, 2014.
- [6] B. C. Bartling, S. Khanna, et al. An 8mhz 75pA/mhz zero-leakage non-volatile logic-based cortex-m0 mcu soc exhibiting 100% digital state retention at vdd=0v with <400ns wakeup and sleep transitions. In *ISSCC*, pages 432–433, 2013.
- [7] P. Chiu, M. Chang, et al. Low store energy, low vddmin, 8t2r nonvolatile latch and sram with vertical-stacked resistive memory (memristor) devices for low power mobile applications. *JSSC*, 47(6):1483–1496, 2012.
- [8] T. Aoki, Y. Okamoto, et al. 30.9 normally-off computing with crystalline ingazno-based fpga. In *ISSCC*, pages 502–503, 2014.
- [9] T. Miwa, J. Yamada, et al. Nv-sram: a nonvolatile sram with backup ferroelectric capacitors. *JSSC*, 36(3):522–527, 2001.
- [10] S. Masui, W. Yokozeki, et al. Design and applications of ferroelectric nonvolatile sram and flip-flop with unlimited read/program cycles and stable recall. In *CICC*, pages 403–406, 2003.
- [11] T. Ohsawa, H. Koike, et al. 1mb 4t-2mtj nonvolatile stt-ram for embedded memories using 32b fine-grained power gating technique with 1.0 ns/200ps wake-up/power-off times. In *VLSIC*, pages 46–47, 2012.
- [12] S. Sheu, C. Kuo, et al. A reram integrated 7t2r non-volatile sram for normally-off computing application. In *ASSCC*, pages 245–248, 2013.
- [13] A. Lee et al. Rram-based 7t1r nonvolatile sram with 2x reduction in store energy and 94x reduction in restore energy for frequent-off instant-on applications. In *VLSIC*, 2015.
- [14] W. Wang, A. Gibby, et al. Nonvolatile sram cell. *IEDM*, 2006.
- [15] S. Yamamoto, Y. Shuto, and S. Sugahara. Nonvolatile sram (nv-sram) using functional mosfet merged with resistive switching devices. In *CICC*, pages 531–534, 2009.
- [16] Y. Wang, Y. Liu, et al. Pacc: a parallel compare and compress codec for area reduction in nonvolatile processors. *TVLSI*, 22(7):1491–1505, 2014.
- [17] X. Sheng, Y. Wang, et al. Spac: a segment-based parallel compression for backup acceleration in nonvolatile processors. In *DATE*, pages 865–868, 2013.
- [18] ROHM. Batasheet of BD5xxx free delay time setting CMOS voltage detector IC series.
- [19] S. Roundy, D. Steingart, et al. Power sources for wireless sensor networks. *Springer: Wireless Sensor Networks*, 2014.
- [20] S. Kim, R. Vyas, et al. Ambient rf energy-harvesting technologies for self-sustainable standalone wireless sensor platforms. *Proceedings of the IEEE*, 102(11):1649–1666, 2014.
- [21] X. Li, H. Liu, et al. Rf-powered systems using steep-slope devices. In *NEWCAS*, pages 73–76, 2014.
- [22] H. Liu, X. Li, et al. Tunnel fet rf rectifier design for energy harvesting applications. *JESTCS*, 4(4):400–411, 2014.
- [23] X. Sheng, C. Wang, et al. A high-efficiency dual-channel photovoltaic power system for nonvolatile sensor nodes. In *NVMSA*, pages 1–2, 2014.
- [24] J. Christmann, E. Beigne, et al. An innovative and efficient energy harvesting platform architecture for autonomous microsystems. In *NEWCAS*, pages 173–176, 2010.
- [25] D. Porcarelli, D. Brunelli, et al. A multi-harvester architecture with hybrid storage devices and smart capabilities for low power systems. In *SPEEDAM*, pages 946–951, 2012.
- [26] V. Boicea. Energy storage technologies: The past and the present. *Proceedings of the IEEE*, 102(11):1777–1794, 2014.
- [27] S. Bandyopadhyay and A. P. Chandrakasan. Platform architecture for solar, thermal, and vibration energy combining with mppt and single inductor. *JSSC*, 47(9):2199–2215, 2012.
- [28] W. Cong, N. Chang, et al. Storage-less and converter-less maximum power point tracking of photovoltaic cells for a nonvolatile microprocessor. In *ASPDAC*, pages 379–384, 2014.
- [29] A. K. Abdelsalam, A. M. Massoud, et al. Platform architecture for solar, thermal, and vibration energy combining with mppt and single inductor. *TPE*, 26(4):1010–1021, 2011.
- [30] M. A. G. de Brito, L. Galotto, et al. Evaluation of the main mppt techniques for photovoltaic applications. *TIE*, 60(3):1156–1167, 2013.
- [31] Y. Wang, H. Jia, et al. Register allocation for hybrid register architecture in nonvolatile processors. In *ISCAS*, pages 1050–1053, 2014.
- [32] M. Zhao, Q. Li, et al. Software assisted nonvolatile register reduction for energy harvesting based cyber-physical system. In *DATE*, 2015.
- [33] Q. Li, M. Zhao, et al. Compiler directed automatic stack trimming for efficient nonvolatile processors. In *DAC*, 2015.
- [34] M. Xie, M. Zhao, et al. Fixing the broken time machine: consistency-aware checkpointing for energy harvesting powered nonvolatile processor. In *DAC*, 2015.
- [35] C. Moser, J. Chen, et al. Reward maximization for embedded systems with renewable energies. In *RTCSA*, pages 247–256, 2008.
- [36] X. Lin, Y. Wang, et al. A framework of concurrent task scheduling and dynamic voltage and frequency scaling in real-time embedded systems with energy harvesting. In *ISLPED*, pages 70–75, 2013.
- [37] D. Zhang, S. Li, et al. Intra-task scheduling for storage-less and converter-less solar-powered nonvolatile sensor nodes. In *ICCD*, pages 348–354, 2014.
- [38] D. Zhang, Y. Liu, et al. Deadline-aware task scheduling for solar-powered nonvolatile sensor nodes with global energy migration. In *DAC*, 2015.
- [39] M. R. Guthaus, J. S. Ringenberg, et al. Mibench: a free, commercially representative embedded benchmark suite. In *DAC*, pages 3–14, 2001.
- [40] H. Li, Y. Liu, et al. An energy efficient backup scheme with low inrush current for nonvolatile sram in energy harvesting sensor nodes. In *DATE*, 2015.