

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/282178750>

Nonvolatile Processor Architecture Exploration for Energy Harvesting Applications

Article in IEEE Micro · September 2015

DOI: 10.1109/MM.2015.88

CITATIONS

43

READS

386

7 authors, including:



[Kaisheng Ma](#)

Pennsylvania State University

38 PUBLICATIONS 641 CITATIONS

[SEE PROFILE](#)



[Xueqing Li](#)

Pennsylvania State University

74 PUBLICATIONS 928 CITATIONS

[SEE PROFILE](#)



[Shuangchen Li](#)

University of California, Santa Barbara

48 PUBLICATIONS 1,222 CITATIONS

[SEE PROFILE](#)



[Yongpan Liu](#)

Tsinghua University

158 PUBLICATIONS 2,178 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Complex Oxide Transistors [View project](#)



Energy Harvesting Wireless Sensor Network [View project](#)

Architecture Exploration for Ambient Energy Harvesting Nonvolatile Processors

Kaisheng Ma*, Yang Zheng*, Shuangchen Li*, Karthik Swaminathan*, Xueqing Li*,
Yongpan Liu†, Jack Sampson*, Yuan Xie‡ and Vijaykrishnan Narayanan*

*Pennsylvania State University †Tsinghua University ‡University of California, Santa Barbara
kxm505,yxz184,sul263,kvs120,lixueq,sampson,vijay@cse.psu.edu, ypliu@tsinghua.edu.cn, yuanxie@ece.ucsb.edu

Abstract— Energy harvesting has been widely investigated as a promising method of providing power for ultra-low-power applications. Such energy sources include solar energy, radio-frequency (RF) radiation, piezoelectricity, thermal gradients, etc. However, the power supplied by these sources is highly unreliable and dependent upon ambient environment factors. Hence, it is necessary to develop specialized systems that are tolerant to this power variation, and also capable of making forward progress on the computation tasks. The simulation platform in this paper is calibrated using measured results from a fabricated nonvolatile processor and used to explore the design space for a nonvolatile processor with different architectures, different input power sources, and policies for maximizing forward progress.

I. INTRODUCTION

Battery-less systems have been proposed to be the next step in the evolution of computing. It is predicted that in the near future, a number of systems will be powered by technologies that harvest ambient energy sources, enabling exciting new applications such as medical monitoring, toxic gas sensors, and next-generation portable video gadgets [1]. Consequently, there is a great impetus to devise battery-free systems which harvest ambient energy such as solar energy, Wi-Fi, and Radio Frequency (RF) energy from mobile base-stations or even motion energy using piezoelectric devices [2], [3]. These include wireless-powered smart contact lenses for diabetic patients [4], RF-powered devices on the carrier of dragonflies [5], and solar-powered low power processor chips operating in the near-threshold voltage domain [6].

With the increase in popularity of Body-Area-Networks and the Internet-of-Things, energy harvesting systems are being adopted to run a host of applications on these platforms. With increasing complexity, throughput constraints, and computational demands, these applications can be characterized according to their need for nonvolatility, as shown below:

- 1) **Signal detection and sensing.** This comprises of simple applications which require detecting and relaying signals such as UV radiation, blood pressure or blood sugar level, temperature and other atmospheric parameters. The system emits a warning if the signal crosses a threshold.
- 2) **Signal detection and analysis.** This includes applications like wearable EEG/ECG meters. Here, in addition to basic sensing, there is some computation carried out for analyzing the signal for the purpose of diagnosis.
- 3) **Signal prediction.** In addition to sensing the signal, the system needs to predict its future patterns. Exam-

ples include wearable systems that predict and warn against seizures or those that predict the exact ovulation time for women in order to maximize chances of pregnancy. These require a relatively continuous notion of prior history in order to maintain high prediction accuracy.

There are, however, several drawbacks in relying on ambient sources of energy for such computing purposes. Most of these energy sources operate at relatively low conversion efficiencies, since only a small fraction of the total transmitted power can be tapped. In addition, they are not reliable energy sources, since external factors could cause a disruption in the supply. For instance, ambient RF or WiFi power can vary arbitrarily, according to power source, frequency, distance from the transmitter, height, obstacles, external electromagnetic signals and other factors [7].

On account of these limitations, most current energy harvesting platforms tend to restrict themselves to applications from category 1, that require relatively simple signal capturing mechanisms involving minimal computation and processing. While best-effort processing under intermittent power supply conditions may be sufficient for devices that carry out memoryless sensing operations, it would not work for more complex state-dependent processing engines. For instance, applications such as electrocardiogram (ECG) analysis, which require uninterrupted monitoring capabilities would require a more reliable source of energy. Further, several applications demand a Quality-of-Service (QoS) requirement, in that all computation should be completed within a fixed amount of time. In such scenarios, it is mandatory to augment these battery-less systems with some techniques to ensure forward progress, or in the very least, save its current state in case of a power loss.

In this paper, we attempt to address a whole range of application scenarios with varying complexity, primarily from categories 2 and 3. Several different techniques can be adopted while designing the systems. For instance, it would be possible to use a temporary energy storage device like a capacitor in order to provide an alternate source of energy in case the ambient source fails. Further, the state of the system could be checkpointed and restored, using nonvolatile memory technologies [8]–[11]. Finally, the entire processor could be designed using these nonvolatile technologies, as *Non-Volatile Processors (NVPs)*. [12]–[15]. This eliminates the need for explicit checkpointing mechanisms.

The aim of this paper is to analyze the design space be-

tween volatile and nonvolatile processors to determine optimal configurations for applications running on energy-harvesting platforms. With this in mind, this paper makes the following contributions:

- We explore architectures that optimize energy-harvesting processors with different complexities, depending on the nature of the energy source and application characteristics.
- We demonstrate a simulation infrastructure combining Register-Transfer-Level (RTL) and analytical models to evaluate the optimal architecture from a performance and an energy perspective.
- We carry out an evaluation of a fabricated NVP chip to calibrate our simulation model.
- We propose several policies that trade off between performance and the utilization of available energy by choosing which data to save, and when to save it.

The rest of the paper is organized as follows. Section II provides a brief overview of typical energy harvesting systems, ambient power sources that could potentially be harvested as well as the factors involved in the designing the processing element. Section III examines the various architectural considerations that arise when we extrapolate the existing system to use-case scenarios that require more complex, faster and energy-efficient designs. Section IV describes the simulation infrastructure. Section V describes the fabricated NVP. Section VI provides the design guideline. We discuss the prior work in the field in section VII and conclude with section VIII.

II. BACKGROUND

In this section, we introduce a general system powered by ambient energy and characterize possible energy sources in terms of signal magnitude, variability, and granularity of variation. Finally, we focus on the digital signal processing module and motivate the need for nonvolatile logic.

A. Typical energy-harvesting system structures

Figure 1 shows a typical system powered by ambient energy sources. It consists of three blocks: (a) the energy harvesting and management block, (b) the digital signal processor, and (c) the I/O interface including the analog/RF front-end. The energy harvesting and management block determines the entire power that could be used for signal sensing, processing and transmission, and will be discussed in subsection II-B. The signal processor is the main focus of this work and will be discussed in detail. The I/O interface may include digital interfaces like I²C and serial-to-parallel interfaces with peripherals like sensors, display, etc., as well as analog/RF interfaces with electrodes, antennas, etc. Its design aims at reducing the power consumption while satisfying the system requirements. For example, a low-power backscatter modulation technique could be employed to design ultra-low-power wireless transceivers [16]–[18]. The clock generator design is also important in that it affects the recovery time from power failures because it takes time for the output of the clock generator to become stable [19].

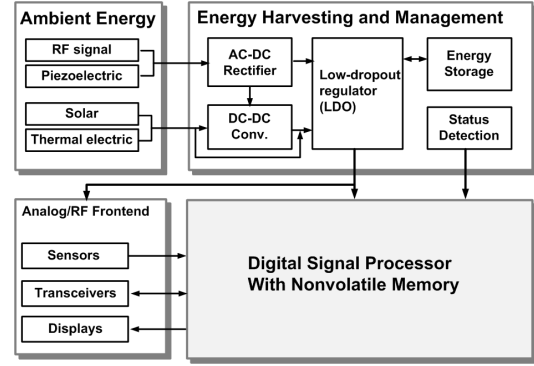


Fig. 1. Energy harvesting system structure

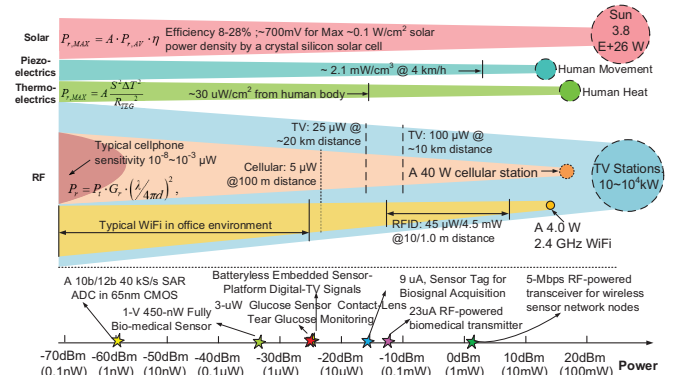


Fig. 2. Typical power ranges of ambient sources

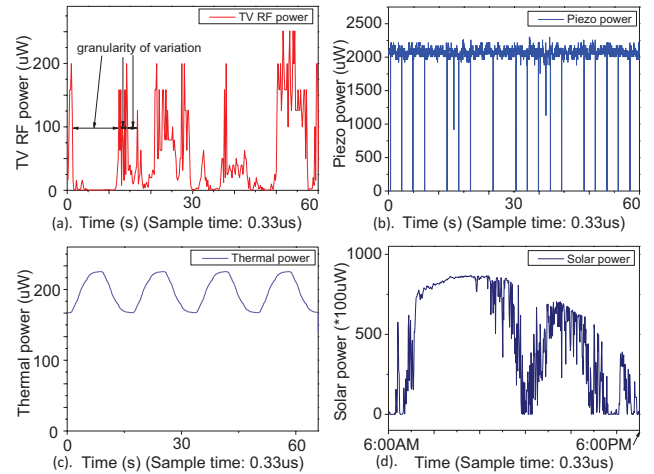


Fig. 3. Power traces a)TV RF b) Piezo c) Thermal d) Solar

B. Ambient power sources and harvesting techniques

Typical ambient energy sources that could be harvested to power an embedded system include solar energy, radio-frequency (RF) radiation, piezoelectric effect and thermal gradients [20]. These sources can be classified according to three characteristics: signal magnitude, variability in signal strength, and granularity of variation/intermittency frequency. Figure 2 illustrates the power harvested in comparison to the typical circuits that can be powered at that power range. The magnitude of harvested power determines the complexity and frequency at which a battery-less system can operate.

Figure 3a) shows power traces for four typical ambient energy sources. The RF energy is obtained by measuring the power of the frequency spectrum from a TV station, the piezo energy is measured through devices fixed on a bike, the thermal energy is generated from characterizations described in [21]–[23] and the solar trace is obtained using data from MIDC [24]. We observe substantial variation in power, even over a few milliseconds for RF in Figure 3a) with the ratio between the maximum and minimum power over this period around $250\times$ [20], [25], [26]. Piezo power is more stable than RF with just some short power loss in Figure 3b). Thermal power, shown in Figure 3c), is even more stable, due to the gradual nature of temperature variation. Variation in solar power, seen in Figure 3d) is contingent on the weather conditions and orientation of the solar cell.

Another feature is the intermittency frequency that influences how soon the power drops below a given threshold as shown in Figure 3a). The intermittency frequency decides the backup and recovery overheads. Sources with periodic behavior, like Figure 3b, facilitate prediction of power loss and enable efficient scheduling of tasks.

While the different energy sources and the associated conversion circuitry (such as rectifiers, DC-DC convertors, voltage boosters) influence the effective power supplied to the processor, these considerations are not the focus of this work. Joint optimization of the conversion circuitry and the processor design will be our future focus.

C. Processor design: Volatile or nonvolatile

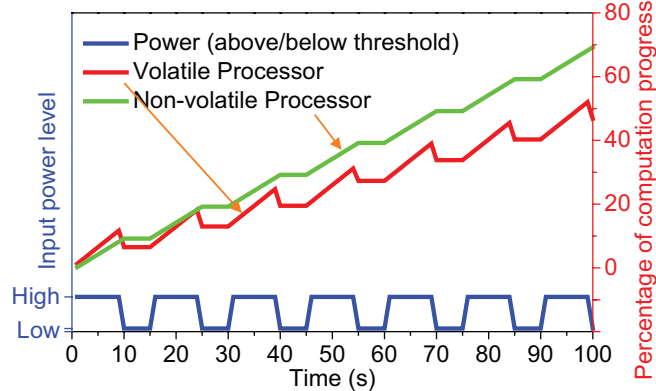


Fig. 4. VP vs. NVP processing progress comparison

Due to the limited and intermittent nature of the ambient power that can be harvested, existing energy-harvesting systems with volatile processors have limited computation capability. To enable more complex state-dependent signal processing that tolerates such power source insufficiency and unreliability, a nonvolatile (NV) processor is essential to provide high-efficiency forward computation progress.

Figure 4 illustrates differences in the behavior of a volatile processor with periodic checkpointing to an external NV memory and a completely NV processor when working under variable power source conditions. While both processors can only run when the input power exceeds a certain threshold, the volatile processor does not retain the instantaneous state of the system when the power drops below the threshold,

resulting in forced rollback to the previously checkpointed state. This could limit forward progress from being made. On the other hand, the non-volatile processor may consume more power than the volatile processor due to the inherently higher power required for a non-volatile read and write operation. Consequently, determining the degree of non-volatility to ensure efficient forward progress is challenging and the focus of this paper. Several factors such as input power profile, processor architecture, and application characteristics influence the design. We explore how they influence the design space of NV processors.

III. ARCHITECTURAL EXPLORATION

This section focuses on figuring out which architectural configurations are best suited to optimally utilize available power and energy by maximizing processor performance under different energy constraints. Hence, depending on the energy that is harvested, we analyze various parameters such as the number of pipeline stages, the data to be backed up and the frequency of backups.

The configuration assumptions for these structures are:

- 1) MIPS ISA.
- 2) 8KHz Clock frequency for all configurations in section III. Selection of clock frequency is driven by the limited strength of the WiFi signal used, rather than limits of the microarchitectures.
- 3) Instruction Memory and ICache: Instruction memory is assumed to be ROM. The ICache can be SRAM, hybrid [27], or NVM [14], [27]. Here ICache is designed using NVMs.
- 4) Data memory and DCache: The Data memory is assumed to be nonvolatile. An SRAM-based DCache employing a write-through strategy does not require any backup policy, while a write-back strategy necessitates writing dirty data back to memory. Our system assumes a NV write-back DCache which preserves dirty data even during periods of power down.

A. Non-Pipelined configuration (NP)

In the absence of any pipeline stages, the entire state of the processor can be characterized by a single instruction state. Hence it is sufficient to focus on the following structures for retrieving architectural state.

- 1) Program counter (PC): The PC address relates to the instruction being executed and needs to be stored.
- 2) Register file (RegFile): Due to frequent usage, the RegFile undergoes large number of writes, hence a volatile RegFile is more energy efficient than an NVM based one. However, all the volatile RegFiles need to be moved to a non-volatile memory on power failures to save state.

In addition to the architecture, there are also tradeoffs between the energy consumed in backing up and recovering the data and the overall performance. These tradeoffs are explored, by choosing which data to save, and when to save it, as demonstrated by the following policies.

Backup Every Cycle (BEC)

In spite of the significant energy penalty, this solution employs

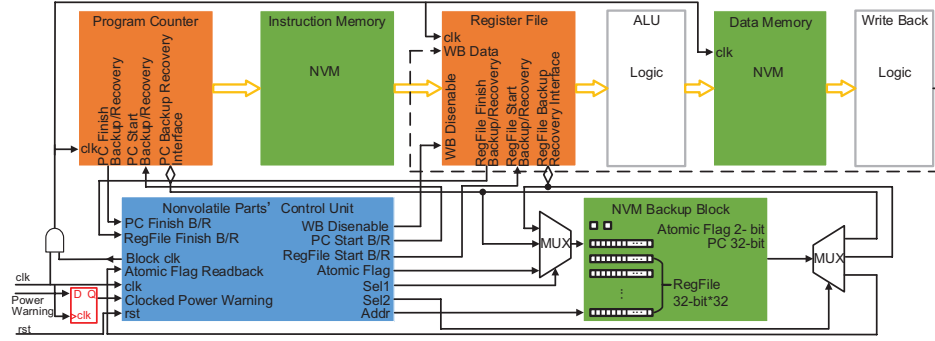


Fig. 5. NP On Demand All Backup structure

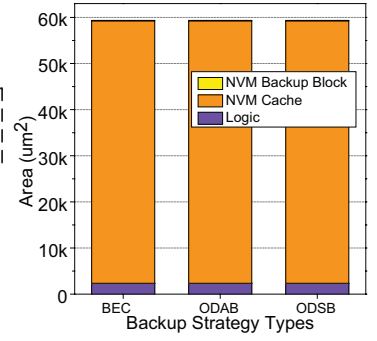


Fig. 6. Non-Pipelined area

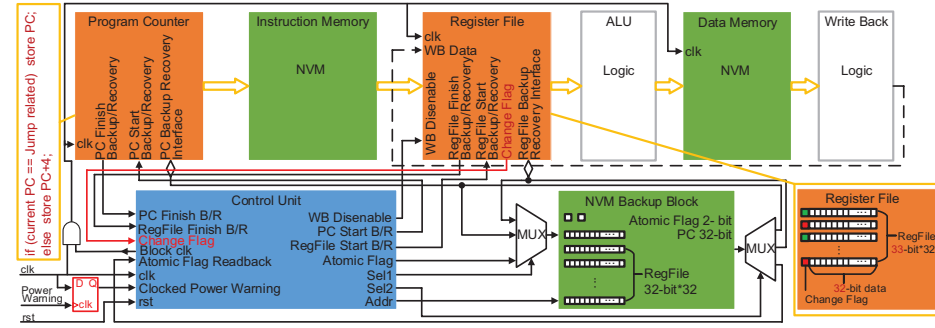


Fig. 7. Non-Pipelined On Demand Selective Backup structure

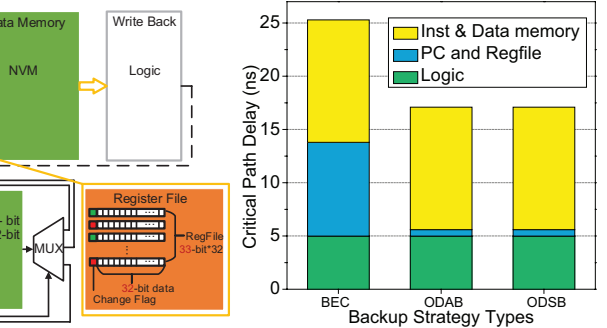


Fig. 8. Non-Pipelined critical path delay ($V_{DD} = 0.95V$)

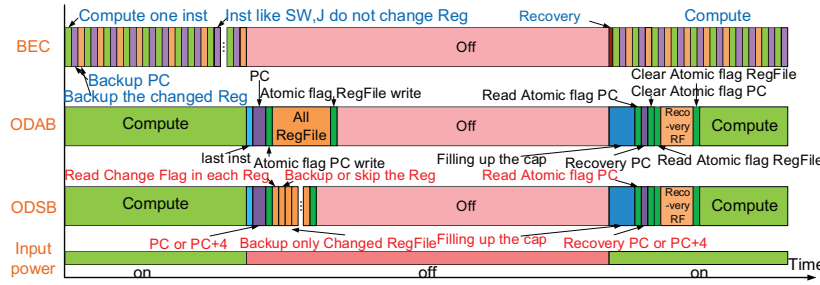


Fig. 9. Individual runtime components for BEC, ODAB and ODSB schemes

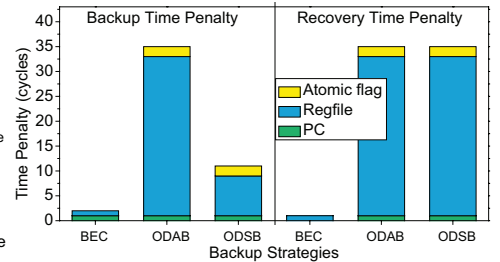


Fig. 10. NP time penalty comparison

an NVM register file, or else both the contents of a volatile Regfile and its counterpart non-volatile structure need to be updated every cycle. As shown in Figure 9, only the PC and few registers are written into the Regfile every cycle. Instructions like *StoreWord* and *Jump* do not require any further Regfile write. Thus, the power increase due to the use of a power hungry NV memory is moderate.

On Demand All Backup (ODAB)

This differs from the previous solution in that all RegFile entries need to be backed up only in the event of a reduced power state. We develop a control structure shown in Figure 5, in which there is an NVM backup block to back up the PC and RegFiles. If input power drops below a preset threshold, a power warning signal is activated. The control unit then starts to back up the PC and resets the atomic flag to indicate that the PC has been successfully backed up. A similar procedure is carried out for the RegFile. When power is restored, we first need to accumulate energy in the capacitor to ensure enough energy for the next backup/recovery operation before continuing execution.

On Demand Selective Backup (ODSB)

In order to reduce the backup time and energy penalty, we develop an *On-Demand Selective Backup* solution. Here, a synchronous power warning signal is used, which may delay the power warning signal a little, but can guarantee that the current PC finishes executing and writing back. To avoid re-executing the instruction corresponding to the current PC, we store $PC + 4$ except in case of jump or branch instructions. This solution can save one clock cycle. Since the frequency of this system is very low, even a single clock cycle may be very significant if power down happens frequently. In the volatile RegFile, we add a change flag to each register to identify if a register has been written into between two backup operations. If the register has not been changed during the interval, the control unit would not need to generate addresses for the unchanged data, as shown in Figure 7.

Simulation results and comparison

Figure 6 shows the component area for the above schemes. We observe that total area is similar, since the NVM Cache and Backup Blocks are much larger than the logic components. The critical path delay shown in Figure 8 indicates that the BEC

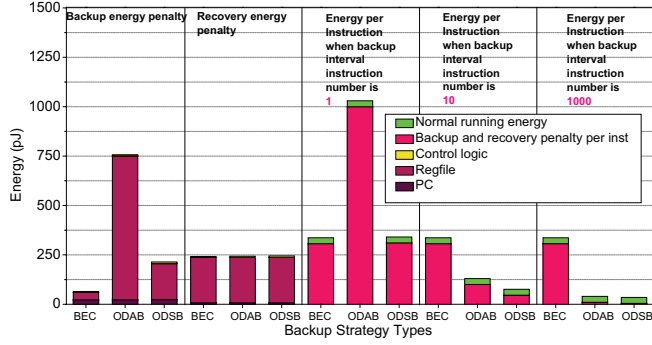


Fig. 11. Energy overheads for each NP scheme. For high frequency of backups, ODAB has the highest overhead, while BEC consumes maximum energy when the backup interval exceeds 10

has lowest peak frequency due to frequent backups. However, the overheads in the other schemes also prevent them from running at peak performance. These overheads are illustrated in Figure 9, which shows compute, backup, recovery and off times for each scheme. BEC distributes the backup energy penalty to every cycle. Thus these penalties are the smallest, as shown in Figure 10 and Figure 11. The recovery time is defined as the time from the activation of the *Energy OK* signal to the time all backup operations are completed. The recovery times are similar across all schemes, but BEC does not need to accumulate energy for backup. Consequently, this scheme can restore the system the fastest. The ODAB scheme needs to back up the PC and the entire RegFile, thus the time and energy penalties are the largest. ODSB reduces the number of RegFile entries to be copied, by detecting if the RegFile has changed during two backup intervals, thus requiring less backup time and energy than ODAB.

In order to determine the best NP scheme, optimizing power and energy is more important than timing, due to the low frequency. In BEC, if the interval time between two power losses is short, the energy per instruction is low because at most only one RegFile entry is backed up, while ODAB needs to back up all RegFile entries. ODSB backs up only one entry at a time, but it is more complex in design. As the backup interval is increased, ODAB and ODSB are more energy efficient, as observed in Figure 11, since backups happen only in the event of a power warning.

In order to avoid a large peak power which can result in system instability, we choose to back up and recover data serially. Although a parallel approach can reduce the back up and recovery time, it increases the peak power requirement. From this point of view, the ODSB is better than ODAB.

- *ODSB is most energy efficient strategy when the source is relatively stable like solar energy. Compared to ODAB, ODSB can reduce the backup energy penalty by 69% with only 0.002% area overhead.*

- *While BEC is not the most energy efficient with very weak sources like WiFi, it does not require the time to accumulate energy in the capacitor to ensure sufficient backup energy is available, as shown in Figure 9. Hence it is viable when the power failures are extremely frequent (less than 1 in 10 cycles), which rarely happens even in WiFi sources.*

B. N-Stage-Pipeline:

In contrast to the MIPS non-pipelined case, a MIPS *N-Stage Pipeline* is traditionally used to improve the clock frequency. Due to the increase in circuit complexity and the activity factor of the processor, the power threshold of this design in energy harvesting systems is higher than that of the non-pipelined case. In this subsection, we assume a Five-Stage-Pipeline structure (5SP) and propose two backup schemes.

Shifted PC & Volatile Flip-flops (SPC/VFF)

The main differences between NP and 5SP configurations are the pipelined data flow with bypass and forward and the complex control flow to handle hazards. In the SPC/VFF scheme, a shifter buffer stores the PC value in each pipeline stage, as shown in Figure 14. This means the PC no longer needs to pass through all pipeline stages to be stored. When the power is down, the clocked power warning signal can guarantee that the PC in the write back stage will be finished. The unfinished PC to be backed up would then be in the data memory stage. We use a shifter instead of simply rolling back the PC since a different PC would need to be backed up for *jump* or *branch* instructions. In case of a *store (SW)* instruction in the MEM stage, it will be guaranteed to finish by the clocked power warning signal. We then back up the PC in EX stage in the shifter instead of at the MEM stage. Once the power is on again, the first instruction will be SW. In this case, we run SW actually twice: the first time during the back up operation, and again as the first instruction after recovery in case the former has not completed.

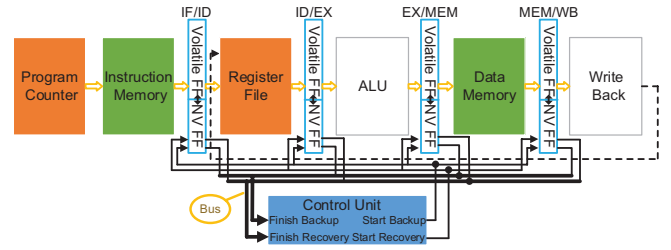


Fig. 12. Five-Stage-Pipeline NVM Flip-flops backup

Nonvolatile Flip-flops Solution (NVFF)

This solution involves the use of NVFF flip-flops (Figure 12). Here, the PC and the RegFile are automatically backed up through NVFF flip-flops in the IF/ID pipeline stages.

Simulation results and comparison

SPC/VFF requires 11% less time and 57% less energy than NVFF in Figure 15. However, an extra 4 clock cycles are needed to re-execute the last 4 instructions lost from the latter pipeline stages after recovery, which we regard as part of the recovery time penalty.

- *Counter to intuition, we show that SPC/VFF is more energy efficient than NVFF. Instead of backing up all data in the pipeline latches, SPC/VFF only backs up one PC with a small shifter. Hence, a smaller backup capacitor with lower leakage is sufficient for SPC/VFF, which, in turn, will affect the power threshold. In this case, SPC/VFF will also be able to outperform NVFF after several repeated instructions.*

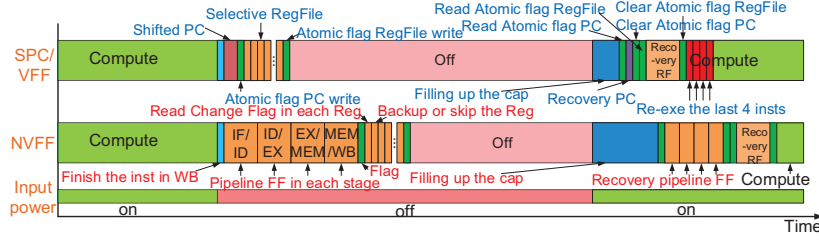


Fig. 13. Comparison of individual runtime components for SPC/VFF and NVFF

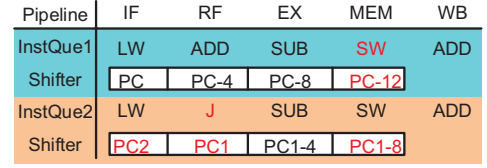


Fig. 14. Illustration of Shifted PC

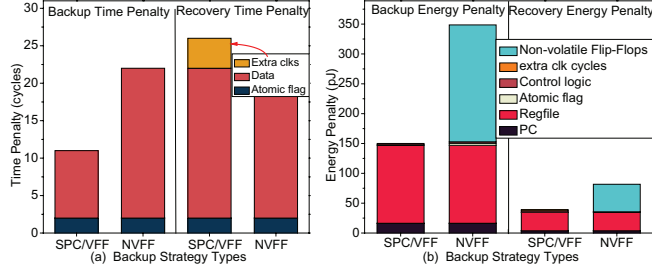


Fig. 15. Comparison of 5SP a) time & b) energy overheads

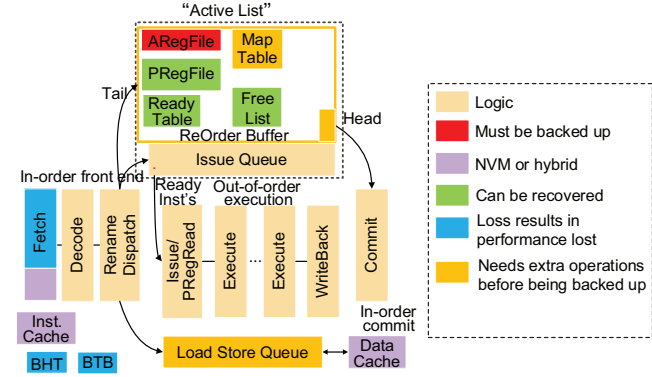


Fig. 16. OoO volatile/non-volatile structure

C. Out-of-Order Processor (OoO)

Our evaluations also included examining a range of issue widths for the 5SP configuration. An average improvement of around 10% was observed when the issue width was increased from 1 to 4. The reason for this limited speedup was due to the in-order nature of the processor.

Compared to the MIPS 5SP configuration, our MIPS out-of-order (OoO) processor configuration, described in Table I, is much more complex. Figure 16 indicates the key blocks we consider in our OoO processor model derived from Fab-Scalar [28]. Conceptually, system state, unlike in the previous two examples, is broadly distributed across several structures such as the PC, ROB, RegFile, Map Table, Issue Queue, Load

Parameter	OoO	Parameter	OoO
Fetch width	4	Map Table	32
Issue width	4	PRegFile	128
ROB size	32	Ready Table	128
IQ size	32	BHT/BBT	128
LSQ size	32/32	ARegFile	32
ICache/DCache	32kB/32kB	Free List	128

TABLE I. PARAMETERS FOR OoO PROCESSOR

Store Queue as well as the Branch History Table and Branch Target Buffer. Some structures are essential to maintain the integrity of the state of the system, while others contribute toward optimizing the performance and/or energy of execution in the presence of frequent backups and recoveries.

Due to the relatively larger power requirements of an OoO processor, there are both fewer periods where the input power exceeds the minimum threshold, as compared to the previous cases, and more state to consider saving during power emergencies. Hence it is imperative to judiciously select the structures to be backed up, in order to ensure a comparable performance to the no-pipeline and n-stage pipeline designs. On the other hand, when there is sufficient power available, the OoO processor can yield a speedup of around $3\times$ over a comparable in-order configuration. Hence it is imperative to judiciously select the structures to be backed up, in order to ensure a comparable performance to the no-pipeline and n-stage pipeline designs.

We propose several resource selection strategies for this purpose, as illustrated in Figure 17.

OoO solutions	ROB	IQ	ARegFile	Map Table	PRegFile	Ready Table	Free List	BHT	BTB
MinR	★	★	★	★	★	★	★	★	★
LLB	★	★	★	★	★	★	★	★	★
MLB	★	★	★	★	★	★	★	★	★
MPL	★	★	★	★	★	★	★	★	★

★ Back up ★ Last uncommitted PC ★ Pseudo-misprediction

Fig. 17. Backup schemes for OoO configuration

Minimum State Resource backup solution (MinR)

MinR backs up the minimal number of bits required to preserve functionality across power interruptions, as shown in Figure 17 and Figure 18. Fundamentally, this approach piggybacks on the branch misprediction mechanism to minimize the number of valid/relevant state bits prior to initiating backup, at the cost of some time and effort being required to enact the misprediction logic prior to checkpointing.

- 1) ROB and PC: To minimize state storage, we only back up the first uncommitted PC at the head of ROB. This means all other instructions in the ROB will be abandoned regardless of status.
- 2) IQ: IQ does not need to be backed up as all the instructions in IQ are uncommitted.
- 3) ARegFile: We either choose to backup ARegFile or PRegFile. The ARegFile is preferred since it is usually smaller.
- 4) Map Table: It is possible that uncommitted instructions following the ROB head could have modified

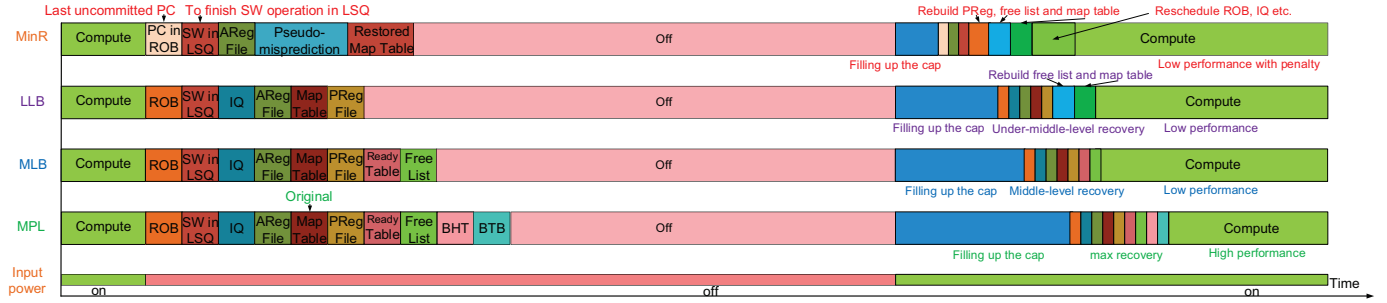


Fig. 18. Out of order structure backup design trade-off

the Map Table. However, since we need to restore the state to the instruction at the ROB head, the Map Table should also be correspondingly restored. To achieve this, we trigger an instruction flush identical to that following a branch misprediction on the ROB head. Since no actual branch prediction occurs, we term this operation *Pseudo-Misprediction*.

- 5) PRegFile, Ready Table, Free List, BHT, and BTB can be recovered.

Low-latency Backup Solution (LLB)

While MinR minimizes bits pushed to nonvolatile storage, it does so at the expense of requiring additional work before backup can begin. We next consider a backup solution that aims to minimize the number of bits to store if backup begins immediately. Rather than back up only the first uncommitted PC, the LLB solution backs up the entire ROB, IQ, ARegFile, Map Table, and PRegFile. Compared to MinR, structures such as the Ready Table and Free List (Figure 21 and Figure 22) can be more easily reconstructed, resulting in a penalty of only a few recovery cycles. While LLB stores more state than MinR, it can sometimes nonetheless be more energy-efficient, due to the extra work required of MinR on both backup and recovery.

Middle-level Backup Solution (MLB)

Instead of using extra recovery time and energy to restore the Ready Table and Free List in the low-level backup solution, MLB backs up Ready Table and Free List as well (Figure 17).

Min-state-lost Backup Solution (MPL)

In this solution, all the structures are backed up including the BHT and BTB as shown in Figure 17.

Integrated Flexible Atomic Backup Solution (IFA)

All previous solutions save and restore a fixed amount of state determined by the structures in question. However, one key feature of the backup process is that it must necessarily be triggered conservatively: The backup signal must be issued at a point where the processor can guarantee sufficient energy to complete the backup **even assuming zero additional input power during backup**. However, in practice, when a power emergency occurs in an energy-harvesting system, it is not usually because input power has dropped to zero, but because it has fallen below some threshold for some period of time. Thus, there may frequently be additional energy available during the backup period that, while insufficient to continue operation, would allow for optional state, such as the BHT, to be subject to optimistic attempts at backup.

We propose a flexible backup mechanism that integrates

aspects of the previous solutions together to exploit the conservative nature of the backup trigger. The key idea of the solution is to regard each backup operation as an atomic operation. A backup operation has only two states: success or failure. Figure 19 shows the systematic structure of this solution. Figure 20 shows how the power may be dropping at different pace to zero and can execute more or less backup.

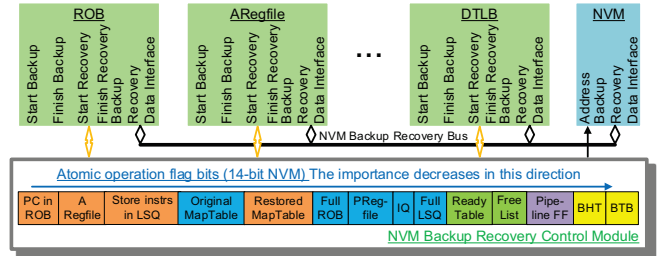


Fig. 19. OoO integrated flexible atomic backup solution

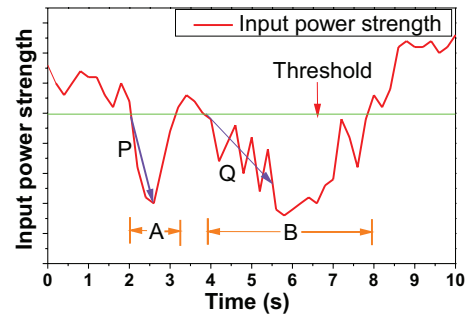


Fig. 20. Scenarios in which IFA can be applied

Simulation results and comparison

For MinR, the pseudo-misprediction operation for the Map Table requires extra backup clock cycles as shown in Figure 18. When recovering, we also need extra cycles to restore PRegFile, Ready Table, and Free List. Further, since we discard all instructions in the ROB following the head, we need to re-execute these instructions, resulting in the timing and energy penalties, shown in Figures 21 and 22 respectively. In the case of LLB, the ROB and PRegFile are relatively large and significantly increase the backup time and energy. On the other hand, the recovery energy penalty is smaller than MinR, because all the instructions and their information in the ROB are backed up, eliminating the need to re-execute these instructions. The backup time and energy penalty of MLB are larger than those of LLB. This MLB strategy can be used when the system is optimizing the time to resume execution

after a power failure. MPL incurs the largest backup and recovery penalties, but backing up all the additional structures incurs the minimum latency to return to peak performance after a power failure. Results show a 29 cycle gain for MinR, but not backing up the BHT and BTB negatively affects IPC. This loss in performance depends on the frequency of interrupts. When the interrupt frequency is low (1 intpt/10s), the prediction accuracy continues to remain at over 90%. However, for higher interrupt frequencies (10 intpt/s), the accuracy drops to around 50%.

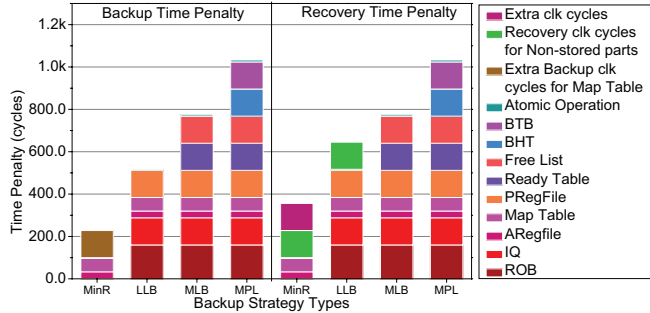


Fig. 21. OoO time penalty

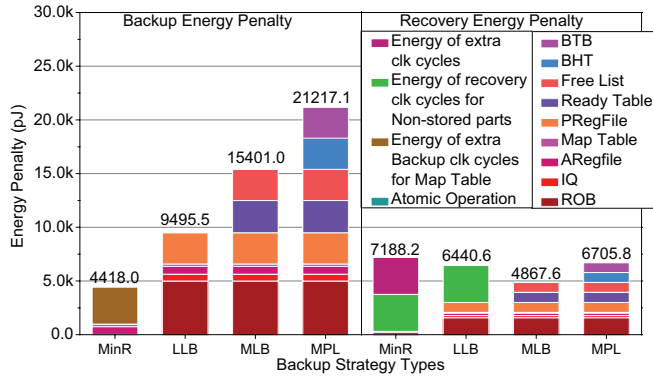


Fig. 22. OoO energy penalty

On account of OoO being thought to be too complex for energy harvesting systems, prior work has seldom considered OoO platforms. *Since OoO needs a much higher threshold than NP and N-SP, the percentage of time OoO can run is much smaller than NP and N-SP. However, it remains a favored option in several test scenarios because the periods of sufficient power are common enough to allow superior performance to pay for lost cycles.* In summary, storing the minimum number of bits (MinR) does not always provide the best backup solution, while MLB has the shortest time to execution after power failure. Thus, the conservative nature of backup initiation offers sizeable potential for opportunistic backup of optional, performance enhancing bits with a flexible backup policy.

IV. SIMULATION INFRASTRUCTURE, BENCHMARKS, AND RESULTS

Simulation results in section III are based on designs generated from Synthesizable Verilog. Timing results are obtained from Modelsim, and logic area and critical path delay from Synopsys Design Compiler using a 45 nm TSMC LP Library.

The non-volatile technology is based on an STT-RAM block for which NVSim [29] is used to derive performance/power numbers. We use a combination of testbenches from the MiBench suite [30], along with some real-world applications. The baseline OoO modules are derived from Fabscale [28]. The power trace is home/office WiFi. Due to the extremely low scavenged power available, the clock frequency is fixed at 8kHz for NP, NSP, and OoO configurations.

These configurations are evaluated against a baseline non-pipelined volatile processor (without checkpointing or data backup) with a measured RF signal as input power. (See Figure 23). Since the volatile processor has the lowest power-on threshold, it is operational for most of the time in the tested 1 minute. However, due to its volatile nature, the processing progress returns to zero when power drops below threshold and it ends up re-executing a majority of the instructions. The non-volatile Non-Pipelined (NP) and Five-Stage Pipeline (5SP), on the other hand, have relatively higher power thresholds than the volatile processor, thus the percentage of operational time is smaller. Although the OoO processor runs only for a small fraction of the time, its performance can be up to 4× faster than NP and 5SP. Hence, for some applications, the OoO processor has the best processing progress at the end.

V. VALIDATION

While the primary focus of this paper has been on an simulation-based exploration, we have explored the non-pipelined on-demand-back up strategy using an actual fabricated processor. In addition to demonstrating the execution of real workloads on the processor, this effort enabled us to gain insights to approximations in initial simulation models and helped refine the simulation model used in this work.

A. System overview

The nonvolatile THU1010N processor is an Intel 8051-based CISC-like architecture, in contrast to the MIPS-like ISA used in the rest of this paper. Hence, we extended our simulation platform to model the 8051 processor for carrying out comparisons with measured data. Further details regarding its fabrication and characterization are provided in [12].

In the design of this prototype chip, the saved state includes the state machine that captures the exact cycle in which the instruction was carried out currently. The NV processor-based system is interfaced to a solar power panel and a UV sensor, as shown in the Figure 24. The processor is based on a 0.13 μm ROHM CMOS-ferroelectric hybrid process. The PC and all RegFiles are FeRAM-based Flip-Flops. The Flip-Flops are realized using an additional backup ferroelectric capacitor (FeCap) for each D flip-flop (DFF) used in the design. When a power failure is detected, the NV control logic backs up the DFFs to the FeCaps. When power is resumed, data is restored from FeCaps to DFFs. All FeCaps are distributed and connected close to their own DFFs, thus the data backup and recovery can proceed in parallel to reduce the operation time. Table II shows the chip specifications. The total power decides the power threshold, the backup energy decides the energy storage capacitor volume. The capacitor used in the system is 470nF.

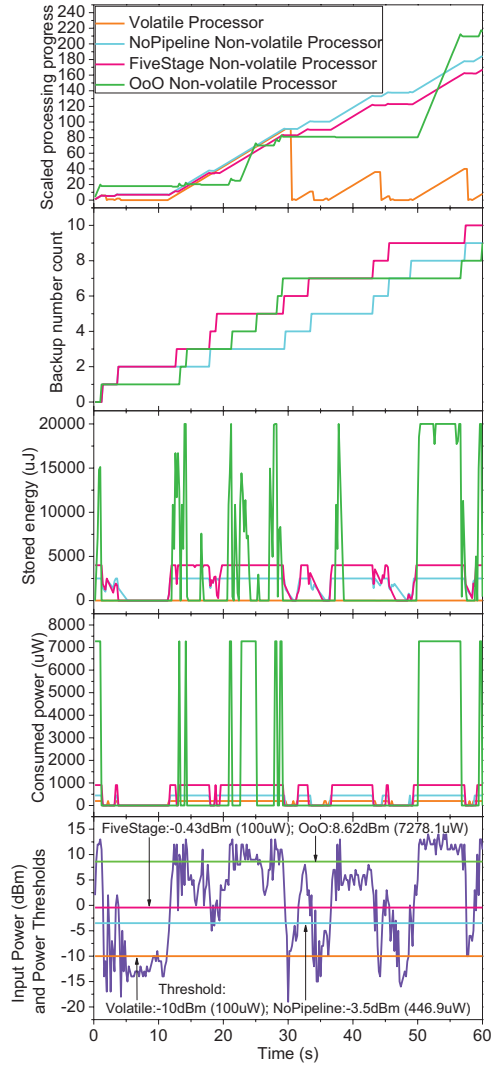


Fig. 23. Simulation results for power, energy, and processing progress etc.

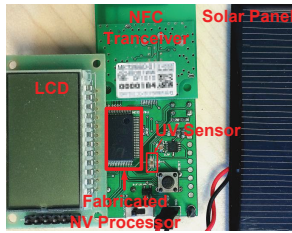


Fig. 24. Prototype system

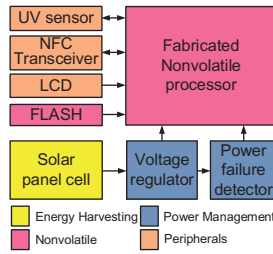


Fig. 25. System block diagram

The design process revealed insights to modeling key aspects in the simulation environment. The clocking network is switched to a lower frequency to transition clock generation from an external oscillator to an internal RC circuit. The external oscillator could then become unstable or may not have sufficient power to operate. Further, a lower frequency increases the reliability of the FeRAM writes and reduces peak power consumption. The slower clock impacts the overall back-up time as compared to using estimates based on a faster operational clock. Similarly, the recovery time should not only

include the time required to restore architectural state but also the time for the clock generators and power supply grid to become stable.

B. Simulator Calibration

Several kernels were executed on both the platform and the simulator (See Table III). To model an intermittent power supply, a 1KHz square waveform power input was fed to the processor and the processor frequency was limited to 3MHz (the maximum frequency at which it could operate based on power supply when connected to the solar panel). Each kernel was executed 1000 times to obtain overall completion time shown in Table III. For the stable power case, the simulator and platform mismatch is negligible. For unstable power, the simulator and the platform measurements differ less than 5%. The simulator averages the energy consumed by an instruction to estimate remaining energy for triggers. However, the actual instruction execution exhibits non-uniform activity. Further, the energy storage capacitance models used in the simulation add and decrease in discrete portions unlike the actual design, which is the reason for the small deviation in the simulation results. This validation process for the simulator based on a real design indicates that our simulation-based models are fair representations of a whole range of real-life systems.

VI. DESIGN GUIDELINES

The complexity of the non-volatile architecture selected for a particular application scenario depends on a variety of factors. These include input power and the stability of the power supply, as well as the computational complexity of the application and its performance requirements. In this section, we attempt to define guidelines for such as selection, based on the considerations described above.

A. Dependence on input power characteristics

The input signal characteristics play a major role in determining the optimal design, as is evident from our experiments with Wi-Fi power trails under different environment conditions. Figures 26 and 27 demonstrate the performance of the various backup schemes discussed in Section III when home and office Wifi sources are used for harvesting energy. For the home environment, a non-pipelined ODSB architecture is the

Parameter	Result	Parameter	Result
Max. clock	25MHz	Total power	160μW@1MHz
Process Technology	0.13μm	Backup energy	23.1 nJ
V _{DD} for core	0.9V-1.5V	Recovery energy	8.1 nJ
Total area	1.015 mm ²	Backup time	7μs
Energy/Inst	347pJ	Recovery time	3μs

TABLE II. MEASURED PARAMETERS

Testbench	Stable/ms Measured	Interrupted/ms Measured	Model	error
FIR-11	0.626	1.260	1.209	-1.59%
Sqrt	2.620	5.280	5.190	0.81%
KMP	3.573	7.184	7.059	0.77%
FFT-8	4.207	8.460	8.238	-0.13%
Matrix	5.826	11.740	12.021	2.39%
Bubble sort	27.23	54.705	57.236	4.63%

TABLE III. EXECUTION TIME ON SIMULATOR AND ACTUAL PLATFORM WHEN USING AN INTERRUPTED POWER SUPPLY GENERATED AS A SQUARE WAVEFORM.

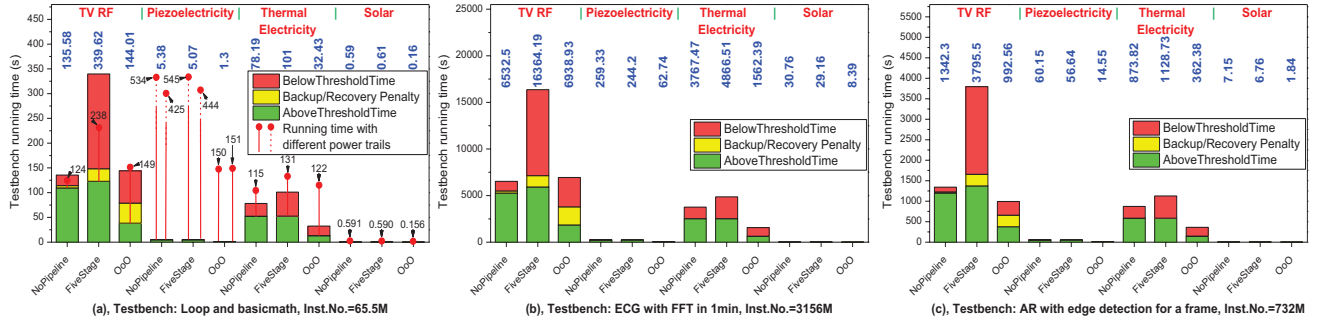


Fig. 28. Execution time for different testbenches under different power sources and trails

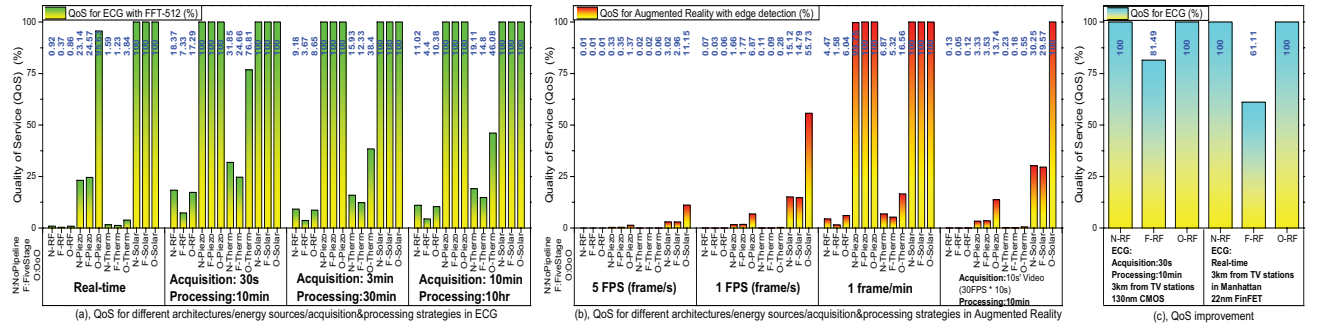


Fig. 29. QoS for ECG and Augmented Reality (AR) applications

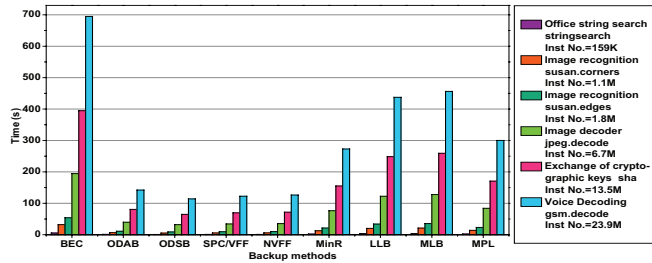


Fig. 26. Execution time with energy scavenged from WiFi home environment

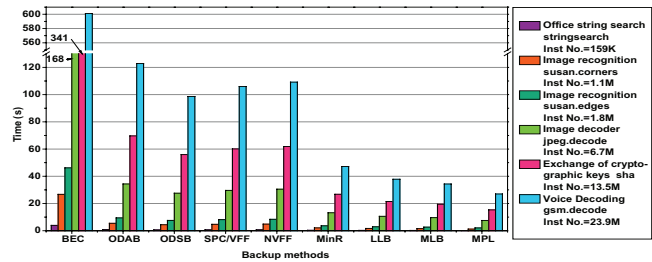


Fig. 27. Execution time with energy scavenged from WiFi office environment

best performing. On the other hand, in the office environment, the more complex OoO processor with the minimum performance loss scheme is desirable. The reason for this behavior is that, the home WiFi signal comprises of a single router, while the office environment consists of several routers of similar signal strengths. A disturbance in the signal would result in input power going to almost zero in the home environment, hence the simplest design with the lowest power threshold is preferred. In contrast, in the office environment, the additional routers continue to supply input power at a relatively similar strength in an uninterrupted fashion, thus allowing for more complex architectures.

B. Dependence on nature of input source

Input energy sources differ both in the magnitude of the input power as well as its variation. Figure 28 demonstrates the behavior of different architectures under these conditions, by testing multiple power traces for each configuration. In each case, the best performing backup policy is adopted. Since the power traces have different ratios between the on and off states, the backup/recovery penalties and thus the running times are also different. We observe that, for the same input power source, the actual execution time of NP and 5SP are roughly the same. However, the higher power threshold in the 5SP configuration results in the below-threshold or off-time being much higher. The OoO configuration is nearly $3\times$ faster than NP and 5SP when it executes and hence the overall running time is proportionately smaller. This behavior is consistent across all input sources with the actual execution time determined by the magnitude of the power source.

C. Meeting Performance/QoS requirements

A large number of applications such as motion sensing and medical monitoring require periodic outputs within fixed time periods, resulting in Quality-of-Service (QoS) constraints. When these systems run on harvesting ambient energy, the unreliable nature of the input source could prevent the QoS demands from being met in some instances.

Figure 29 shows the percentage of instances that meet the QoS demands specified, for two different applications - measurement of ECG and an edge detection algorithm used in vision sensors. It also provides an illustration of the different architectures that are feasible for different QoS constraints for these two applications. For example, in Figure 29a), the probability of achieving real-time ECG processing for an RF non-

volatile processor (N-RF) is only 0.92%. Consequently for RF and Thermal sources, achieving reliable real-time processing is very difficult. On the other hand, most solar and piezo powered architectures can meet even real-time QoS requirements close to 100% of the time.

Table IV shows the various parameters used in defining the energy harvesting platforms and their relation with the harvesting efficiency. For instance, in densely populated areas such as Manhattan, average TV station distances are as low as 3 km. In such cases, the RF power improves by over $11\times$, in comparison to a 10 km baseline distance. Similarly, by shrinking the technology from 130 nm to 22 nm FinFETs [31]–[33] will enable us to achieve 100% QoS for real time ECG applications. Finally, various circuit and architecture-level techniques can be applied to reduce the power: adoption of emerging technologies like Tunnel-FET [20], low power sub-threshold circuits, dark silicon-aware architectures [34], clock gating, dynamic-voltage-frequency-scaling (DVFS) and Dynamic-Adjusting Threshold-Voltage Scheme (DATS) [35] etc. are some examples.

Thus it is evident that application requirements and environmental constraints also play a major role in determining the best architecture for the energy harvesting platform and the best source to power it.

Source	Parameter	QoS Baseline	Relation to Efficiency
RF	Antenna gain	6dBi	α
	Bandwidth	539M	α
	Distance	10km	$1/\alpha^2$
Therm	Area	1cm^2	α
	ΔT	20°C	α^2
Piezo	Volume	1cm^3	α
Solar	Area	4cm^2	α
	Efficiency	28%	α
Circuit	IP matching, AC-DC, DC-DC, LDO, Cap		
Tech.	Shink Tech.	130nm	α^2
	FinFET, IG-FinFET, TFET, NC-FET	CMOS	
	DVFS, DATS	Fixed frequency	
	Voltage	0.95V	$1/\alpha^2$

TABLE IV. BASELINE AND RELATIONSHIP WITH QoS IMPROVEMENT

VII. RELATED WORK

A. NVM in energy-harvesting platforms

There have been several works demonstrating processors that harvest different sources of ambient energy. [36]–[38] demonstrate energy-harvesting microcontroller chips with FeRAM as embedded non-volatile memory. In this paper, we use one such design as our baseline and subsequently carry out detailed architecture-level explorations. There have also been several works that use other non-volatile technologies such as STT-RAMs, PCRAMs and ReRAMs at various levels of abstraction, from design of Flip-Flops [39]–[43] to realizing micro-architecture components using these technologies [15], [44]. Our models, while having being calibrated against FeRAMs, can be easily extended to most state-of-the-art non-volatile memory technologies.

B. Architectural Aspects of Energy Harvesting

Computing under unreliable power supply conditions leads to several interesting architecture and system-level issues, many of which have been dealt with in this paper. [45] have explored the possibility of concurrent programming under

intermittent energy and the various efforts required to maintain program consistency. These issues are addressed by means of atomic instructions allied with an on-chip capacitance to ensure that the processor has sufficient power to complete the ongoing instruction. [46] uses an FeRAM for quickly checkpointing the system state in case of power loss in transiently powered computers. In addition, our work also explores in detail various micro-architectures by varying the power-on threshold, thus being able to optimally run for a whole range of application complexities. In [47], the authors propose a power-management technique for a solar-powered multicore architecture. Our paper, on the other hand, extends our analysis to different energy sources with a detailed micro-architectural evaluation.

C. Checkpointing mechanisms

There is a large body of work that employs checkpointing techniques in processors. Checkpointing techniques that leverage non-volatile memories have been proposed for improving the resiliency in high performance systems [48]. In [49], the authors propose using STT-RAMs to selectively checkpoint micro-architectural structures that are vulnerable to transient errors. In [50], the authors examine transiently powered RFID systems. They use software techniques to transform the program into interruptible computation operations, thus facilitating checkpointing. The techniques proposed in our paper do not modify the program and use the NVM for hardware-level checkpointing.

VIII. CONCLUSION

In this paper, we explore the various factors involved in designing a battery-less system powered by ambient energy sources. We explore various architectural level designs and optimizations that are viable for different ambient sources such as solar, RF, thermal and piezo energy and attempt to define the design guidelines that would facilitate this selection. To counter the intermittent nature of the energy source, we evaluate several nonvolatile processor configurations along with energy-optimal techniques to conserve the state while maximizing forward progress. We examine the trade-offs between performance and energy for different architectural complexities and application requirements. Finally, we compare and validate our simulation results with a fabricated non-volatile solar energy-harvesting processor platform. This paper will be a first guideline for ambient energy harvesting system designers.

ACKNOWLEDGEMENT

This work was supported in part by Shannon Lab Huawei Technologies Co., Ltd, High-Tech Research and Development (863) Program under contract 2013AA01320, the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions under contract YETP0102, the Center for Low Energy Systems Technology (LEAST), sponsored by MARCO and DARPA, and by the NSF awards 1160483 (AS-SIST), 1205618, 1213052, 1461698, and 1500848. The authors would also like to thank our shepherd Prof. Engin Ipek and reviewers for their comments that have greatly improved this work. Thanks Xiao Sheng and YiQun Wang from Tsinghua University for their assistance with chip measurements and Nandhini Chandramoorthy from Penn State for her help with simulations.

REFERENCES

- [1] A. P. Chandrakasan, D. C. Daly, J. Kwong, and Y. K. Ramadass. Next generation micro-power systems. In *IEEE Symp. on VLSI Circuits*, 2008.
- [2] A. Parks, A. Sample, Z. Yi, and J. Smith. A wireless sensing platform utilizing ambient RF energy. In *IEEE Radio and Wireless Symposium (RWS)*, 2013.
- [3] K. Gudan, S. Chemishkian, J. J. Hull, M. S. Reynolds, and S. Thomas. Feasibility of wireless sensors using ambient 2.4GHz RF energy. In *2012 IEEE Sensors*, pages 1–4, 2012.
- [4] H. Tsukayama. Google's smart contact lens: What it does and how it works. *The Washington Post*, 2014.
- [5] S. Gollakota, M. Reynolds, J. Smith, and D. Wetherall. The Emergence of RF-Powered Computing. *IEEE Computer*, 2014.
- [6] R. Merritt. ISSCC: Intel focuses on low power, digital RF, 2012.
- [7] H. Visser, A. Reniers, and J. Theeuwes. Ambient RF energy scavenging: GSM and WLAN power density measurements. In *European Microwave Conference (EuMC)*, 2008.
- [8] J. Hursey, T. Mattox, and A. Lumsdaine. Interconnect agnostic checkpoint/restart in openMPI, 2009.
- [9] S. Kannan, A. Gavrilovska, K. Schwan, and D. Milojicic. Optimizing checkpoints using NVM as virtual memory. In *IPDPS*, 2013.
- [10] D. Sorin, M. Martin, M. Hill, and D. Wood. SafetyNet: improving the availability of shared memory multiprocessors with global checkpoint/recovery. In *ISCA*, 2002.
- [11] X. Dong, N. Muralimanohar, N. Jouppi, R. Kaufmann, and Y. Xie. Leveraging 3D PCRAM technologies to reduce checkpoint overhead for future exascale systems. In *High Performance Computing Networking, Storage and Analysis*, 2009.
- [12] Wang. Y et al. A 3us wake-up time nonvolatile processor based on ferroelectric flip-flops. In *ESSCIRC*, pages 149–152, 2012.
- [13] S. Bartling et al. An 8MHz 75uA/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*, 2013.
- [14] X. Guo, E. Ipek, and T. Soyata. Resistive computation: Avoiding the power wall with low-leakage, STT-MRAM based computing. *ISCA*, 2010.
- [15] Wang. Y et al. A compression-based area-efficient recovery architecture for nonvolatile processors. In *Design, Automation and Test in Europe Conference (DATE)*, 2012.
- [16] C. Boyer and S. Roy. Backscatter communication and RFID: Coding, energy, and MIMO analysis. *Communications, IEEE Transactions on*, 2014.
- [17] C. Boyer and S. Roy. Coded QAM Backscatter Modulation for RFID. *IEEE Transactions on Communications*, 2012.
- [18] P. Mercier et al. A 78 pW 1 b/s 2.4 GHz radio transmitter for near-zero-power sensing applications. In *ESSCIRC*, 2013.
- [19] R. Larsen. EFM32 white paper, 2014.
- [20] X. Li et al. RF-powered systems using steep-slope devices. *New Circuits and Systems Conference (NEWCAS)*, 2014.
- [21] R. Grezard and J. Willemin. A self-starting fully integrated auto-adaptive converter for battery-less thermal energy harvesting. In *NEWCAS*, pages 1–4. IEEE, 2013.
- [22] V. Leonov, T. Torfs, P. Fiorini, and C. Van Hoof. Thermoelectric converters of human warmth for self-powered wireless sensor nodes. *Sensors Journal*, 7(5):650–657, 2007.
- [23] V. Leonov, T. Torfs, R. Vullers, J. Su, and C. Van Hoof. Renewable energy microsystems integrated in maintenance-free wearable and textile-based devices: the capabilities and challenges. In *ICIT*, 2010.
- [24] Measurement and Instrumentation Data Center (MIDC). <http://www.nrel.gov/midc/>.
- [25] P. Harpe, E. Cantatore, and A. van Roermund. A 10b/12b 40 kS/s SAR ADC With Data-Driven Noise Reduction Achieving up to 10.1b ENOB at 2.2 fJ/Conversion-Step. *IEEE JSSC*, 2013.
- [26] Z. Xiaodan, X. Xiaoyuan, Y. Libin, and L. Yong. A 1-v 450-nw fully integrated programmable biomedical sensor interface chip. *IEEE JSSC*, 44(4):1067–1077, 2009.
- [27] M. Natsui et al. Nonvolatile logic-in-memory array processor in 90nm MTJ/MOS achieving 75% leakage reduction using cycle-based power gating. In *ISSCC*, 2013.
- [28] N Choudhary et al. Fabscalar: Composing synthesizable RTL designs of arbitrary cores within a canonical superscalar template. In *ISCA*, pages 11–22, 2011.
- [29] X. Dong, C. Xu, Y. Xie, and N. Jouppi. NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(7):994–1007, 2012.
- [30] M. Guthaus et al. Mibench: A free, commercially representative embedded benchmark suite. In *Workload Characterization, IEEE International Workshop on*, pages 3–14, Dec 2001.
- [31] K. Ma et al. Key characterization factors of accurate power modeling for FinFET circuits. *Science China Information Sciences*, 2014.
- [32] K. Ma et al. Independently-controlled-gate finfet 6t sram cell design for leakage current reduction and enhanced read access speed. pages 296–301, July 2014.
- [33] X. Cui et al. Research on power model of multi-mode finfet standard cell. pages 1–3, Oct 2012.
- [34] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. *Micro*, 32(3):122–134, May 2012.
- [35] Cui. X. et al. A dynamic-adjusting threshold-voltage scheme for finfets low power designs. In *ISCAS*, 2013.
- [36] M. Zwerg et al. An 8uA/MHz microcontroller with embedded FeRAM for energy-harvesting applications. In *ISSCC*, 2011.
- [37] S. Khanna et al. An FRAM-Based nonvolatile logic MCU SoC Exhibiting 100% digital state retention at VDD=0V achieving zero leakage With <400-ns wakeup time for ULP applications. *JSSC*, 2014.
- [38] A. Baumann et al. A mcu platform with embedded fram achieving 350na current consumption in real-time clock mode with full state retention and 6.5us system wakeup time. In *2013 Symposium on VLSI Circuits (VLSIC)*, pages C202–C203, 2013.
- [39] D. Chabi et al. Ultra low power magnetic flip-flop based on checkpointing/power gating and self-enable mechanisms. *IEEE Transactions on Circuits and Systems*, PP(99), 2014.
- [40] J-M Choi, C-M Jung, and K-S Min. PCRAM Flip-Flop Circuits with Sequential Sleep-in Control Scheme and Selective Write Latch. *Journal of Semiconductor Technology and Science*, 2013.
- [41] H. Kimura et al. Highly reliable non-volatile logic circuit technology and its application. In *ISMVL*, pages 212–218, May 2013.
- [42] W. Jue, L. Yongpan, Y. Huazhong, and W. Hui. A compare-and-write ferroelectric nonvolatile flip-flop for energy-harvesting applications. In *International Conference on Green Circuits and Systems*, 2010.
- [43] S. Onkaraiah et al. Bipolar ReRAM Based non-volatile flip-flops for low-power architectures. In *New Circuits and Systems Conference (NEWCAS)*, 2012.
- [44] Y. Wang et al. PaCC: A parallel compare and compress codec for area reduction in nonvolatile processors. *VLSI Systems, IEEE Transactions on*, 2014.
- [45] B. Ransford and B. Lucia. Nonvolatile memory is a broken time machine. In *MSPC*, 2014.
- [46] 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 *VLSI Design and Embedded Systems*, 2014.
- [47] C. Li, W. Zhang, C-B Cho, and T. Li. SolarCore: solar energy driven multi-core architecture power management. In *HPCA*, 2011.
- [48] X. Dong, Y. Xie, N. Muralimanohar, and N. Jouppi. Hybrid checkpointing using emerging nonvolatile memories for future exascale systems. *ACM Trans. Archit. Code Optim.*, 8(2), June 2011.
- [49] K. Swaminathan, R. Mukundarajan, N. Soundararajan, and V. Narayanan. Towards resilient micro-architectures: Datapath reliability enhancement using STT-MRAM. In *VLSI*, 2011.
- [50] B. Ransford, J. Sorber, and K. Fu. Mementos: System support for long-running computation on rfid-scale devices. In *ASLPOS*, 2011.