# Design Exploration of Inrush Current Aware Controller for Nonvolatile Processor

Yongpan Liu*, Fang Su†, Zhibo Wang†, Huazhong Yang*
Dept. of Electronic Engineering, Tsinghua University, Beijing, 100084, China
{ypliu,yanghz}@tsinghua.edu.cn*, {suf15,wangzb13}@mails.tsinghua.edu.cn†
**(Invited)**

***Abstract*—Leakage power consumption has become a critical limitation in the normally off low-power systems. Power gating provides a promising solution to reduce leakage energy but cannot avoid data loss. Nonvolatile processor has paved the way to achieve zero leakage power while maintaining data. However, nonvolatile processor faces severe inrush current problem when all nonvolatile memories are backed up in parallel. A large inrush current will occur and induce IR drop, which deteriorates the stability of the entire system. This paper proposes a distributed backup control architecture for nonvolatile processors to cope with the inrush current problem. Furthermore, we devise corresponding algorithms to accelerate backup operations under given maximum tolerable current constraints. The proposed techniques are evaluated on a simulation platform and a prototype chip. Experimental results demonstrate up to 26.3% reduction in backup time compared with the sequential backup strategy under the same inrush current constraint.**

## I. Introduction

Advances in device, circuit, architecture and compiler over the last three decades have made us surrounded by a world of smart electronic systems. However, all these systems, including laptops, smart phones and wearable devices, face severe energy challenge. It is predicted that the power budget at present can only support 10% transistors on a chip to work simultaneously in 2020 [1]. That is, the rest 90% transistors have to be switched off, which leads to the "Dark Silicon" challenge [2]. Moreover, inline with the decreasing process dimension, leakage power has exceeded dynamic power. Therefore, it is critical to minimize leakage energy at the design stage of VLSI systems.

Power gating [3] is a frequently-used technique to achieve leakage power reduction. It works well with combinational logic circuits. However, the circuits with memories inside, such as processors, still suffer from nontrivial leakage power, because they have to keep the power supply to maintain data and system states. Moreover, data loss will still occur in traditional memories when power failure arrives in energy harvesting systems.

Recently, nonvolatile processor [4] (NVP) has drawn great attentions, and provides an effective solution to overcome the leakage power problem. Wang et al. [5] proposed the first fabricated NVP with ferroelectric flip-flops (FeFF). Sakimura et al. [6] achieved nanosecond data backup and recovery time in a spintronics-based NVP. Bartling et al. [7] adopted nonvolatile logic arrays to back up the system status, achieving considerable testability and area savings. These NVPs integrate distributed nonvolatile memories (NVM), and store all system states into NVMs when the power is off. In this way, NVPs achieve zero leakage power as well as data retention.

However, conventional NVP faces severe inrush current problem when it backs up all NVMs in parallel. On one hand, a large surge current probably burns surrounding circuits, which is prohibitive from a safety perspective. On the other hand, it induces IR drop, and harms the reliability of the entire system. To overcome the surge current problem, this paper explores the design space of inrush current aware NVPs. We divide the NVMs inside a NVP into clusters to reduce the degree of parallelism in store operations, and schedule the backup sequence of all clusters to minimize backup time under current constraints. Our specific contributions are summarized as follows:

- we perform analysis on the inrush backup current of NVPs;
- we propose a 3-layer distributed backup control architecture (DBCA) for inrush current reduction;
- we devise both a current prediction algorithm and a scheduling algorithm based on DBCA to minimize backup time under inrush current constraints;
- we implement and evaluate the proposed DBCA technique on gem5 simulator [8] with benchmarks from mibench [9], and experimental results illustrate that DBCA and the proposed algorithms reduce 26.3% backup time against the sequential backup strategy, while satisfying the current restriction.

The remaining of this paper is organized as follows. Section II discusses the fundamental background of the proposed technique. Section III elaborates the design of DBCA and Section IV shows the experimental results. Section V concludes the paper.

## II. Background

This section provides the necessary background on the design exploration of inrush current aware controller. Firstly, we present the working mechanism and advantages of NVPs. After that, we analyze the inrush current problem in a NVP's backup operation. Finally, related works and design challenges on inrush current reduction are discussed.

### *II.A. Nonvolatile Processors*

NVPs adopt distributed nonvolatile flip-flops (NVFF) and nonvolatile SRAMs (NVSRAM) to store system state. Fig. 1 shows the block diagram of a typical NVP. When a power failure arrives, a nonvolatile controller drives all NVFFs and NVSRAMs to back up data within several cycles. Compared with traditional volatile processors, NVPs have the following advantages.

*II.A.1) Zero Leakage Power Consumption:* When power is off, the distributed NVMs in a NVP can sustain data without leakage power. On the contrary, volatile processors must always keep power on to prevent data loss.

This work was supported in part by the NSFC under grant 61271269 and High-Tech Research and Development (863) Program under contract 2013AA01320 and the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions under contract YETP0102.

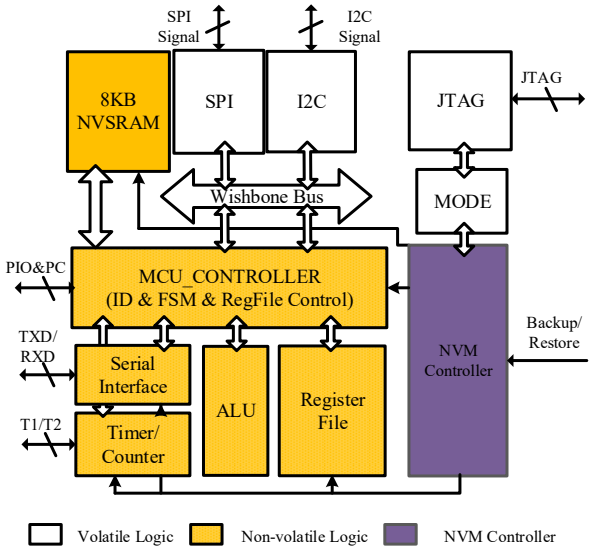


Fig. 1. Block diagram of a typical NVP [5] with NVSRAM

*II.A.2) Fast Backup/Restore Operation:* Due to the read/write speed limitation of secondary memories, the backup operation of traditional volatile processors is slow and energy-inefficient. A NVP realizes in-place backup and finishes the backup/restore operation within nanoseconds.

*II.A.3) High Efficiency:* After power failures, traditional volatile processors have to re-execute instructions due to the state loss. This will induce time penalty and extra power consumption. NVPs can store intermediate data and continue working after power supply is recovered.

Benefiting from above advantages, NVPs provide a promising way to build high energy-efficiency systems with low leakage power consumption compared to volatile processors.

### *II.B. Inrush Current Problem*

To clearly illustrate the inrush current problem in NVP's backup operation, we analyze the store current of NVFF and NVSRAM in NVPs.

Table I lists the backup current and the required amount of NVFF and NVSRAM in a NVP. We have two observations from the table. 1) NVSRAM has far larger backup current per bit than NVFF does. 2) The amount of NVSRAM is two orders of magnitude larger than that of NVFF. This is because NVSRAMs are used as instruction or data caches in a NVP, while NVFFs are used as registers.

TABLE I
BACKUP CURRENT ANALYSIS OF NVMS

| NVM type | Backup current per bit | Quantity | Total current |
|---|---|---|---|
| NVFF | ~2uA | $10^3 \sim 10^4$ | $< 20$mA |
| NVSRAM | ~40uA | $10^5$ | $> 2$A |

At 70 °C, the maximum tolerable current of 130nm process is about 150mA [10]. Obviously, the parallel backup operation of NVSRAM is unacceptable. Therefore, as shown in Fig. 2, a dilemma exists for a NVP at the backup point. The y-axes represents the total backup current magnitude, and the x-axes indicates time. The red

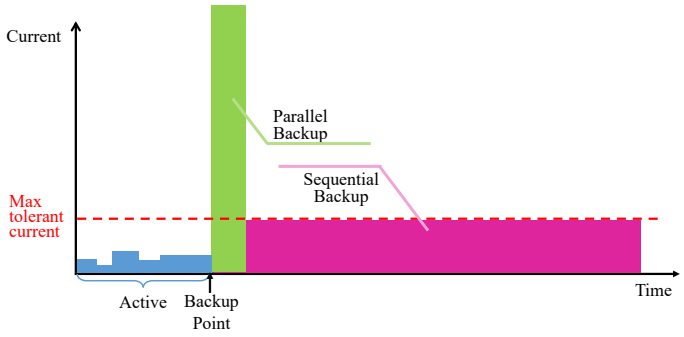


Fig. 2. Parallel and sequential backup strategies

dash line represents the maximum tolerable current. There are two choices at the backup point. The parallel backup strategy achieves extremely short backup time, but cannot guarantee the reliability of the circuit. The sequential approach satisfies the current constraint. However, it deteriorates the backup speed of NVP, since only a small number of NVSRAMs can back up at one time. In a word, it is critical to achieve an optimal tradeoff between current constraint and backup speed.

### *II.C. Related Works and Design Challenges*

Many researches have attempted to reduce the inrush backup current. These works can be classified into two groups. One approach is to reduce the amount of the backup bits in parallel. For example, some compression methods were proposed to reduce the total number of bits to be stored, including parallel compare and compress codec [11] and segment-based parallel compression [12]. These works optimize the backup operation of NVFFs, but may not be suitable for NVSRAMs, where circuit density does matter. Recently, Li et al. [13] proposed a dead block prediction method for NVSRAM based on the statistics of data. Tsai et al. [14] provided a bit-level redundancy elimination method to discard needless store operations in NVSRAM. Another approach is to minimize the average backup current of NVMs. For example, Chiu et al. [10] designed a low store energy, low VDD NVSRAM cell, which reduces the backup current from the device-level. All above works on NVSRAM significantly relieve the inrush current problem, but they have not thought the architectural-level optimization.

This paper focuses on the architectural exploration for NVPs, to cope with the inrush backup current problem. The design challenge mainly comes from the following aspects. On one hand, the redundant bits in backup operation should be eliminated with trivial overheads in area and energy. On the other hand, how to schedule the backup order of all NVMs is inconspicuous.

## III. ARCHITECTURAL EXPLORATION

In this section, we propose DBCA to cope with the inrush backup current problem, which is discussed in the previous section. Furthermore, we devise corresponding algorithms for the proposed 3-layer architecture to achieve the minimum backup time under inrush current constraints.

Fig. 3 shows an overview of DBCA. The lowermost layer is the NVMs, including NVFFs and NVSRAMs. They are segmented into clusters to avoid the surge current brought by parallel backup. The central layer consists of distributed control units, which predict the backup current and generate backup control signals. This layer plays a

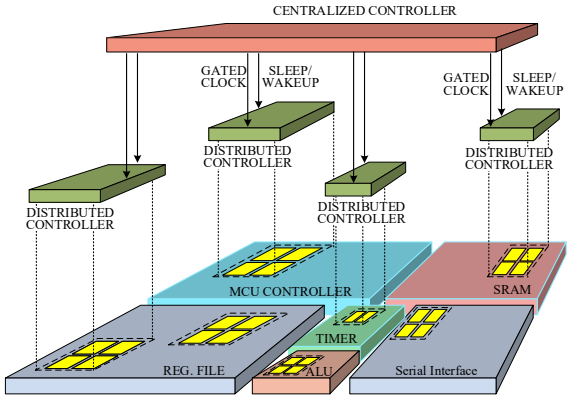


Fig. 3. DBCA: distributed backup control architecture

critical role in the entire architecture. Finally, the uppermost layer of the architecture is a centralized scheduler. It schedules the backup sequence of all the distributed controllers. Under certain current constraints, it backs up as many NVM clusters as possible, in order to minimize the total backup time. In this way, we realize a partially parallel backup scheme. Inside a cluster, all the bits are backed up simultaneously. Meanwhile, different clusters are backed up in a non-parallel manner, to prevent inrush current. We will give the detailed elaboration of the proposed DBCA in the following subsections.

### *III.A. Layer 1: Nonvolatile Memories*

Our analysis in Section II has proved that the amount of NVSRAMs is far larger than that of NVFFs. Furthermore, NVFFs act as processor registers, while NVSRAMs are used as data cache. We design the backup strategy based on the following observation. The data stored in NVFFs plays a more important role than that in NVSRAMs. If data loss occurs in registers, the processor has to roll back and recalculate the lost bits, which will greatly deteriorate the processing efficiency. On the contrary, data loss in cache only costs a few cycles to fetch the data from main memory. Therefore, we put all NVFFs into one group and divide NVSRAMs into different groups. The backup operation first backs up all NVFFs in parallel, which consume less current. After that, we estimate the backup current in each NVSRAM group, and arrange their backup orders.

### *III.B. Layer 2: Distributed Controllers*

The central layer of DBCA consists of distributed controllers. Each controller manages a group of NVFFs or NVSRAMs. Since the backup operation of NVFFs is quite simple, the rest of this paper will focus on NVSRAM's backup strategy. The distributed controllers have two major functions.

*III.B.1) Generating Backup Control Signals:* Conventional NVP employs a centralized controller to generate the backup control signals for all NVMs. To accomplish the same functionality, a control signal generator is also necessary in DBCA. We implement a FSM in each distributed controller to provide a series of backup control signals. Fig. 4 shows an example of backup control signals generated by the distributed controller. Firstly, the controller drives all NVFFs to backup. After that, the FSM generates control signals to back up NVSRAMs. Note that NVSRAM1 needs backup and NVSRAM2 does not. Similarly, when power supply recovers, the FSM successively controls NVFFs and NVSRAMs to restore.

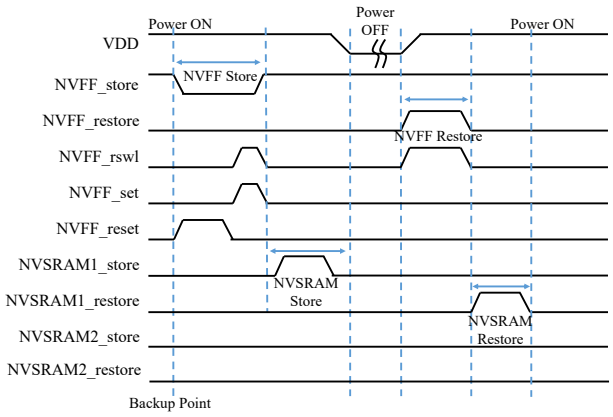


Fig. 4. Backup control signals

*III.B.2) Predicting Current Magnitude:* Inside the data cache of a CPU, there are some cache blocks which will never be visited again before they are replaced by other blocks. We call them "dead blocks". A single-core processor with a 16KB 4-way association L1 data cache is tested under 9 frequently-used benchmarks. The dead block rate of data cache under each benchmark is presented in Fig. 5.

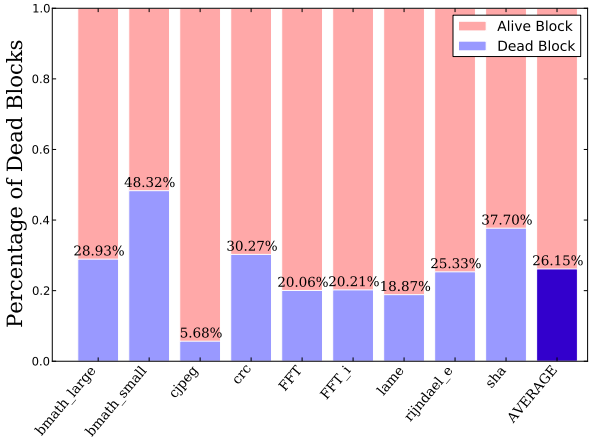


Fig. 5. Dead block rate in data cache under different benchmarks

As we can see, the average dead block rate is nearly 30%. We have the following observation from this result. At NVP's backup point, a large number of bits in NVSRAM can be classified into dead blocks. That is to say, the dead bits are useless in the future, thus we can discard them and only back up the alive ones. In this way, both backup time and current are reduced, since this approach backs up less bits than the full backup strategy does.

Therefore, the second function of distributed controller is to estimate the dead blocks in NVSRAM. Since the backup current of NVSRAM is in proportion to the number of stored bits, the distributed controllers are also able to forecast the backup current.

Many researches have focused on the prediction of dead blocks in data cache. For example, Lai [15] provided a trace-based predictor and Li et al. [13] proposed a dead-block prediction method on the basis of statistics. Liu et al. [16] designed a predictor based on the bursts of accesses to a cache block.

TABLE II
COMPARISON OF DEAD-BLOCK PREDICTION METHODS

| Approach | Area overhead/Byte | Energy overhead/nJ | Accuracy |
|---|---|---|---|
| SBDP | 64 | 0.429 | 91% |
| Reftrace | 1K | 63 | 88% |
| Cache burst | 0.7K | 33 | 96% |

Table II compares the performance of several prevalent dead block prediction methods. Among the three methods listed in the table, the Statistics Based Dead-block Prediction (SBDP) approach achieves relatively high accuracy with trivial area and energy overheads. Moreover, SBDP only requires simple calculation, which is more friendly to the integrated circuits. Taking all these factors into consideration, we select SBDP as the inrush current prediction algorithm. The SBDP algorithm is built on the positive correlation between RUB values and dead block rates. Due to space limitations, we do not go into details about SBDP algorithm.

How to realize the dead block predictor with minimum area and energy overheads is a critical issue at the design stage of our proposed architecture. In [13], Li only presented a block diagram of the circuit implementation of SBDP. This paper implements the distributed controller with integrated circuits. We minimize the area of the predictor by reusing combinational logics. The details of the circuit is given in Section IV.

### *III.C. Layer 3: Centralized Scheduler*

The top layer of DBCA is a centralized scheduler. In layer 2, the distributed controller predicts the backup current of one cluster of NVMs. According to the predicted current magnitude of each cluster, the scheduler decides the backup order of all distributed controllers. In the precise of guaranteeing the backup current does not exceed the maximum tolerant current restriction, the scheduler decides which cluster(s) to back up at the present cycle. The ultimate aim of the centralized scheduler is to back up all alive blocks as quickly as possible.

**Algorithm 1** Backup Scheduling Algorithm

**Input:** $I_{constraint}, \{I_j\}$
**Output:** Backup decisions $\{T_{Backup,i}\}$
**Initialization:** $\{T_{Backup,i}\} \leftarrow 0,\ t \leftarrow 0,\ peak\_current \leftarrow 0$
**Scheduling:**

```
while ∃i not scheduled do
    update {T_Backup,i} for all i
    for i not scheduled do
        if peak_current + I_i < I_constraint
        T_Backup,i ← t
        peak_current ← peak_current + I_i
    end for
    t ← t + t_interval, peak_current ← 0
end while
return T_Backup,i for all i
```

This scheduling task can boil down to the bin packing problem. However, the bin packing problem is NP-Hard, which is difficult to get the optimal result in a short time. Taking the scheduler's implementability into consideration, an approximation algorithm is preferable for the scheduler. We devise a backup scheduling algorithm on the basis of "First Fit" [17], which is an approximation algorithm for bin packing problem. The description of the proposed backup scheduling algorithm is given in Algorithm 1.

By far, we have finished the design of DBCA and the corresponding algorithms. Fig. 6 shows a block diagram of DBCA's working mechanism. In the first place, the dead block predictor reads the flag bits in cache blocks, and decides whether the block is dead or alive. Then the predictor adds up all alive blocks in a cluster, and outputs a estimated backup current value. After that, the centralized scheduler decides the backup order of all distributed controllers according to the predicted current magnitudes. Finally, when a distributed controller decides to start backup operation, it generates and sends control signals to all alive blocks in its cache cluster to drive NVSRAMs to backup.

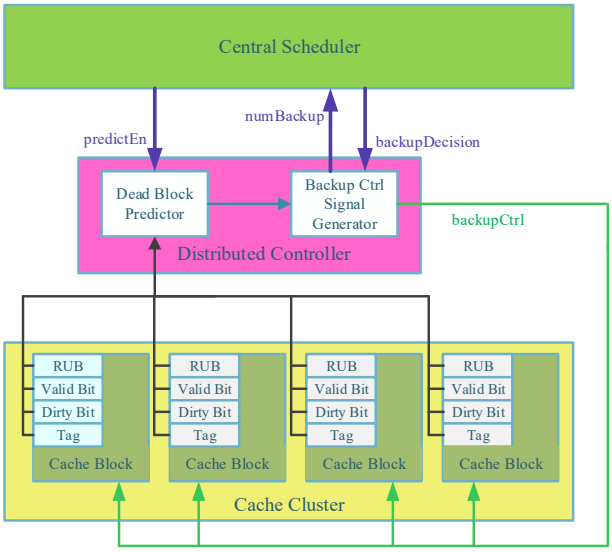


Fig. 6. The working mechanism of DBCA

## IV. EVALUATION

This section evaluates the overall performance of the proposed architecture and algorithms. Firstly, we validate our proposal by implementing a NVP with DBCA on a simulation platform. We select backup speed and energy under inrush current constraints as two major performance indicators. After that, we realize the circuit of distributed controller, which contains a signal generator and a predictor with SBDP algorithm. Finally, we estimate the area and energy overheads introduced by DBCA.

### *IV.A. Experiment Setup*

Gem5 is a highly configurable CPU architecture simulator. We implement a NVP with DBCA on gem5. All the simulation parameters are set according to a fabricated NVP. Table III lists some important configurations of the simulator.

### *IV.B. Performance Evaluation*

Fig. 7 depicts the simulation results of SBDP algorithm under different benchmarks. The left vertical axes indicates the dead block rate of data cache, which is obtained by SBDP algorithm. The right vertical axes is the prediction error rate. The horizontal axes

TABLE III
SIMULATION SETUP

| | | |
|---|---|---|
| Simulator configuration | Processor | Single core, 25MHz |
| | NVSRAM (data cache) | Size: 16KB, 4-way association |
| | | Cache block: 32 Bytes |
| | | Backup current: 40 uA/bit |
| | | Backup time: 10 ns/bit |
| | Current constraint | 160 mA |
| Benchmark | 9 benchmarks from mibench | |
| Test condition | Instruction number | $10^6$ warm-up instructions |
| | | $2 \times 10^6$ test instructions |
| | Backup point | 10 random points |

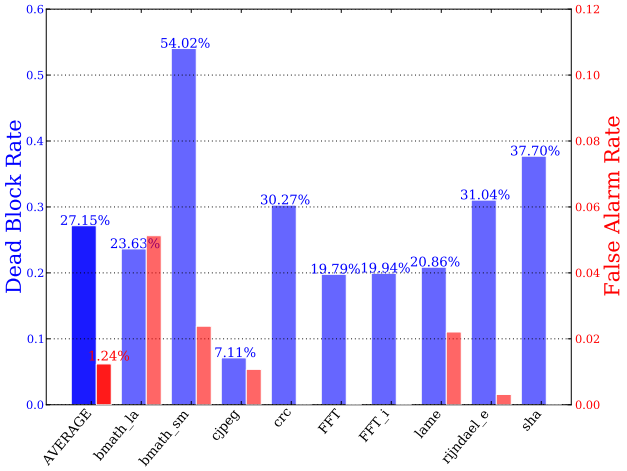


Fig. 7. Predicted dead block rates

represents different programs running in NVP, including 9 benchmarks from mibench. And the average values of all benchmarks are shown in the leftmost bars. As shown in Fig. 7, the benchmark "basicmath_small" has over 50% dead blocks. On average, the dead block rate in data cache reaches 27.15%, and the prediction error rate is 1.24%.

Fig. 8 compares the backup speed of the sequential backup approach and the proposed DBCA. The uppermost red bar represents the time consumption of sequential backup strategy. Due to the inrush current limitation, NVP cannot back up all NVMs simultaneously. The sequential approach backs up as many bits as possible at one time, on the premise of keeping the backup current at a safe level. However, the sequential backup strategy still backs up all NVSRAMs, without discarding dead or no-use ones. Therefore, the backup speed of the sequential approach is relatively slow. The dark blue bar denotes the average backup time of DBCA under different benchmarks. As the figure has shown, DBCA accelerates NVP's backup operation by 26.3% on average. Especially, DBCA can reduce the backup time by half under the benchmark "basicmath_small". This is because the benchmark "basicmath_small" has a larger dead block rate than other benchmarks. This result validates the effectiveness of the proposed DBCA approach.

Compared with the full-parallel backup strategy, DBCA restricts the backup current at a safe level. Moreover, DBCA can save over

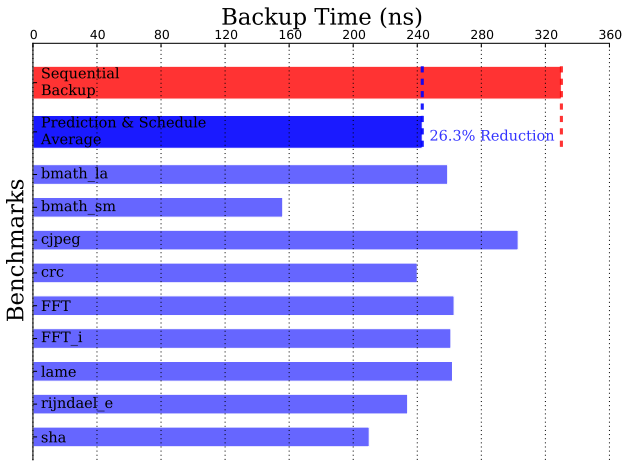


Fig. 8. Simulation result of DBCA & the scheduling algorithm

one-fourth backup time, with respect to the sequential backup strategy under the same current constraint. Table IV presents a performance comparison between DBCA and other approaches.

TABLE IV
PERFORMANCE OF DBCA AND OTHER BACKUP SCHEMES

| Backup scheme | Full-parallel | Sequential backup with equal-division | **DBCA (our proposal)** |
|---|---|---|---|
| Current safety | × | √ | √ |
| Backup speed | Very fast | Slow | **Fast** |

Energy consumption is another important performance indicator to evaluate the proposed architecture and algorithms. We estimate the backup energy by the following equation:

$$E_{tot} = E_{nvff} \cdot N_{nvff} + E_{nvsram} \cdot N_{nvsram} \quad (1)$$

where $E_{nvsram}$ and $E_{nvff}$ indicate the per-bit backup energy of NVSRAM and NVFF, respectively. Similarly, $N_{nvsram}$ and $N_{nvff}$ denote the number of NVSRAMs and NVFFs. The total energy $E_{tot}$ consists of the backup energy of both NVMs. DBCA limits backup current by predicting and discarding the dead blocks in NVSRAM, thus $N_{nvsram}$ will decrease significantly. In this evaluation, DBCA saves 21.8% backup energy on average, compared to full backup approaches.

### *IV.C. Hardware Implementation*

In Section III, we've illustrated that the distributed controller, to estimate dead blocks and generate control signals, is the key module of the entire architecture. We implement the distributed controller to verify its functionality, and design a prototype chip under 65nm RRAM process. The prototype chip consists of four 4KB RRAM-based NVSRAM macros and 16 distributed controllers. The layout of the prototype chip is shown in Fig. 9.

### *IV.D. Overhead Estimation*

Although DBCA can significantly save time and energy in NVP's backup operation, we also have to take area and energy overheads of DBCA into account. Both dead block prediction and backup

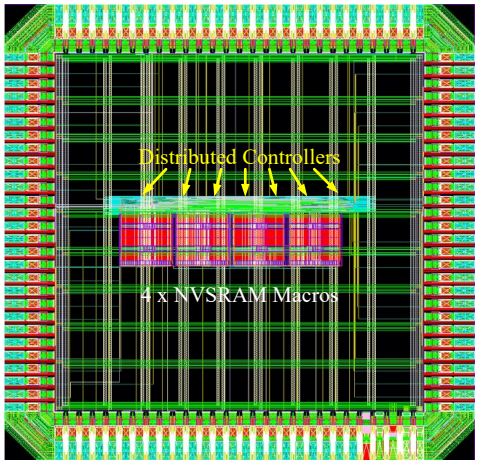


Fig. 9. Layout of the prototype chip

control signal generation require additional area and extra energy. To precisely estimate the overheads of the proposed DBCA approach, we extract the area and energy information from the layout. We compare the area and power consumption of the distributed controllers with a conventional NVP. Table V shows the overhead estimation results.

TABLE V
AREA & ENERGY OVERHEADS OF DISTRIBUTED CONTROLLERS

| | Area/um$^2$ | Energy consumption/nJ |
|---|---|---|
| Conventional NVP | 177594.72 | 109.70 |
| NVP with DBCA | 198692.80 | 118.66 |
| Overheads | 11.9% | 8.17% |

In Table V, DBCA brings 8.17% extra energy consumption and 11.9% additional area. This result proves the validity and feasibility of the proposed DBCA approach. Our future work will focus on integrating the DBCA into a real-world NVP.

## V. CONCLUSION

Nonvolatile processor has become a popular topic in recent years, due to its zero leakage power and rapid backup/restore characteristics. However, nonvolatile processor faces severe inrush current problems when it backs up all nonvolatile memories in parallel. To address these concerns, we propose DBCA: a 3-layer distributed backup control architecture. Based on the proposed architecture, we devise a current prediction and a backup scheduling algorithm to achieve tradeoff between backup speed and inrush current reduction. We validate the proposed architecture and algorithms on gem5. Experimental results illustrate that the proposal accelerates the backup operation by nearly 30% against the sequential backup strategy under maximum tolerable current constraint. Furthermore, we realize the circuit of the distributed controllers. The result of place&route indicates that DBCA brings only 11.9% area penalty and 8.17% energy overhead.

## REFERENCES

[1] "International technology roadmap for semiconductors, 2011 edition," 2011.

[2] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*. IEEE, 2011, pp. 365–376.

[3] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-v power supply high-speed digital circuit technology with multithreshold-voltage cmos," *Solid-State Circuits, IEEE Journal of*, vol. 30, no. 8, pp. 847–854, 1995.

[4] Y. Liu, Z. Li, H. Li, Y. Wang, X. Li, K. Ma, S. Li, M.-F. Chang, S. John, Y. Xie *et al.*, "Ambient energy harvesting nonvolatile processors: from circuit to system," in *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, p. 150.

[5] Y. Wang, Y. Liu, S. Li, D. Zhang, B. Zhao, M.-F. Chiang, Y. Yan, B. Sai, and H. Yang, "A 3us wake-up time nonvolatile processor based on ferroelectric flip-flops," in *ESSCIRC (ESSCIRC), 2012 Proceedings of the*. IEEE, 2012, pp. 149–152.

[6] N. Sakimura, T. Sugibayashi, R. Nebashi, and N. Kasai, "Nonvolatile magnetic flip-flop for standby-power-free socs," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 8, pp. 2244–2250, 2009.

[7] S. C. Bartling, S. Khanna, M. P. Clinton, S. R. Summerfelt, J. A. Rodriguez, and H. P. McAdams, "An 8mhz 75$\mu$a/mhz zero-leakage non-volatile logic-based cortex-m0 mcu soc exhibiting 100% digital state retention at v dd= 0v with¡ 400ns wakeup and sleep transitions," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*. IEEE, 2013, pp. 432–433.

[8] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti *et al.*, "The gem5 simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1–7, 2011.

[9] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "Mibench: A free, commercially representative embedded benchmark suite," in *Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop on*. IEEE, 2001, pp. 3–14.

[10] P.-F. Chiu, M.-F. Chang, S.-S. Sheu, K.-F. Lin, P.-C. Chiang, C.-W. Wu, W.-P. Lin, C.-H. Lin, C.-C. Hsu, F. T. Chen *et al.*, "A low store energy, low vddmin, nonvolatile 8t2r sram with 3d stacked rram devices for low power mobile applications," in *VLSI Circuits (VLSIC), 2010 IEEE Symposium on*. IEEE, 2010, pp. 229–230.

[11] Y. Wang, Y. Liu, S. Li, X. Sheng, D. Zhang, M.-F. Chiang, B. Sai, X. S. Hu, and H. Yang, "Pacc: A parallel compare and compress codec for area reduction in nonvolatile processors," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 22, no. 7, pp. 1491–1505, 2014.

[12] X. Sheng, Y. Wang, Y. Liu, and H. Yang, "Spac: A segment-based parallel compression for backup acceleration in nonvolatile processors," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013*. IEEE, 2013, pp. 865–868.

[13] H. Li, Y. Liu, Q. Zhao, Y. Gu, X. Sheng, G. Sun, C. Zhang, M.-F. Chang, R. Luo, and H. Yang, "An energy efficient backup scheme with low inrush current for nonvolatile sram in energy harvesting sensor nodes," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*. EDA Consortium, 2015, pp. 7–12.

[14] H.-J. Tsai, C.-C. Chen, K.-H. Yang, T.-C. Yang, L.-Y. Huang, C.-H. Chung, M.-F. Chang, and T.-F. Chen, "Leveraging data lifetime for energy-aware last level non-volatile sram caches using redundant store elimination," in *Proceedings of the 51st Annual Design Automation Conference*. ACM, 2014, pp. 1–6.

[15] A.-C. Lai, C. Fide, and B. Falsafi, "Dead-block prediction & dead-block correlating prefetchers," in *Computer Architecture, 2001. Proceedings. 28th Annual International Symposium on*. IEEE, 2001, pp. 144–154.

[16] H. Liu, M. Ferdman, J. Huh, and D. Burger, "Cache bursts: A new approach for eliminating dead blocks and increasing cache efficiency," in *Proceedings of the 41st annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2008, pp. 222–233.

[17] B. S. Baker, "A new proof for the first-fit decreasing bin-packing algorithm," *Journal of Algorithms*, vol. 6, no. 1, pp. 49–70, 1985.