cNV SRAM: CMOS Technology Compatible Non-volatile SRAM based Ultra-low Leakage Energy Hybrid Memory System

Jinhui Wang, *Member, IEEE,* Lina Wang, Haibin Yin, Zikui Wei, Zezhong Yang, Na Gong, *Member, IEEE*

Abstract—A CMOS technology compatible non-volatile SRAM (cNV SRAM) is proposed in this paper to achieve energy efficient on-chip memory. cNV SRAM works as conventional 8T SRAM to keep high speed in work mode; in sleep mode, it backs up the data in its NV component and switches off the power supply, thereby minimizing the leakage energy without data loss. The circuit- and architectural- level implementation schemes of cNV SRAM are developed considering multiple key performance parameters including energy dissipation, access time, write time, noise margin, layout area, restoration time, and injection charges. Simulation results on SPEC 2000 benchmark suite demonstrate that cNV SRAM realizes 86% energy savings on average with negligible performance impact and small hardware overhead as compared to conventional SRAM. Finally, the impact of the sleep time and memory size on the effectiveness of cNV SRAM is analyzed in detail and it shows that cNV SRAM is particularly effective to implement large on-chip memories with long idle time.

Index Terms—Non-volatile SRAM, leakage energy, hybrid memory, sleep mode

1 Introduction

SRAM have been the predominant and universal technologies used to implement storage in computer systems [1]. Fast on-chip memories are vital to increase the speed of the data flows and, hence, the speed of the entire system. With the high access speed and robustness characteristics, eight transistors (8T) SRAM cells have been widely applied to on-chip memories (including register files and cache memories) in state-of-the-art microprocessors [2], [3], [4]. However, on-chip memories take over 50% of the die area according to 2012 ITRS [5], and dissipate a dominant amount of the leakage energy [6]. This issue is expected to aggravate with continuous technology scaling, especially in battery-powered computing devices, such as laptop computers, smart phones, and medical sensors [7]. There have been many efforts to reduce leakage energy of on-chip memory, such as voltage and frequency scaling [8], body biasing [44], variable keeper [45], sleep transistor sizing [46], cache decay [47], drowsy cache [48]. For each technique, there is a trade-off between energy efficiency, performance, and implementation cost. In addition to these techniques, hybrid memories, commonly combining fast and low power partners with time-division satisfaction of high performance and low leakage energy requirements, have been proved to be extremely effective in suppressing leakage energy. Significant amount of researches on such hybrid memories have been reported in the literature, which can be divided into two categories. One type is volatile cell based hybrid memories which are as following.

1) Various-T SRAM cell based hybrid memories. 6T SRAM storages are suitable for area efficient design [9]. 8T SRAM with 30% area penalty can achieve reliable low voltage operation [10]. 10T SRAM cell design provides built-in feedback mechanism with 45% area penalty for ultra-low power due to robust operation at ultra-low voltage [11]. In [12], [13], [14], various-T SRAM cells are utilized in different blocks of one memory, even in different bit cells of one storage word to facilitate aggressive scaling of supply voltage for power saving and performance trade-off.
2) DRAM/SRAM based hybrid memories. Several hybrid structures, mixing the high speed SRAM and low leakage DRAM without refreshing periodically, are presented in [1], [15], [16], [17].

However, both DRAM and various-T SRAM are volatile. In sleep mode, although the power supply can be scaled down, it is difficult to predict the minimum V_{dd} [18]. Additionally, V_{dd} cannot be cut off as it is necessary to restore data for relayed working operation. Because the power supply is the source of the leakage energy, the efficiencies of volatile cell based hybrid memories are limited.

The other type of hybrid memories is the emerging SRAM combined non-volatile memories (NVMs), such as

- *J. Wang is with the Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND, 58102,USA, and the College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing, 100124, China. E-mail: wangjinhui@bjut.edu.cn*
- *L. Wang, H. Yin, Z. Wei, and Z. Yang are with the College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing, 100124, China.*
- *N. Gong is with the Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND, 58102,USA. E-mail: na.gong@ndsu.edu*

Manuscript received 3 Feb. 2014

*Please note that all acknowledgments should be placed at the end of the paper, before the bibliography (**note that corresponding authorship is not noted in affiliation box, but in acknowledgment section**).*

SRAM with NEM (Nanoelectro-mechanical) [55], MRAM (Magnetic RAM) [56], FeRAM (Ferroelectric RAM) [19], [54], NCs (Nano-crystals) [57], RRAM (Resistive RAM) [58], STTRAM (Spin Torque Transfer RAM) [20], and PRAM (Phase Change RAM) [21]. They are promising memory technologies for modern microprocessors due to their attractive features. Recently, SRAM with FET [49] and SRAM with split-gate transistor [50] are proposed. However, all of above hybrid structures are heterogeneous semiconductor technology based or unconventional device based [23], [24]. As a result, they cannot be manufactured through traditional CMOS technology, which significantly increases the manufacturing cost [22].

In this paper, a CMOS technology compatible non-volatile SRAM based ultra-low leakage energy hybrid memory is proposed, and that is referred to as cNV SRAM. cNV SRAM cell is implemented based on the Fowler-Nordheim tunneling mechanism such as EEPROM to back up data, while the entire hybrid memory system employs existing high efficient power gating techniques [25], [26] to realize ultra-low leakage energy. cNV SRAM works as conventional 8T SRAM with normal power supply to keep high speed and high noise margin; in sleep mode, it backs up the data in the NV component and switches off the power supply, and thus the leakage energy is significantly suppressed without data loss.

This paper makes the following six major contributions.

- It proposes the circuit level implementations and architectural control mechanisms of cNV SRAM (Section 3).
- It provides a detailed analysis on cNV SRAM cell characteristics considering tunneling voltage, size, and area. In particular, the relation between coupling/tunneling capacitance ratio in NV part of the proposed cell and amount of injection charges as well as restoration time is investigated (Section 4).
- It implements the layout of cNV SRAM and analyzes the access time, write time, and noise margin based on 65 nm SMIC Technology design rules (Section 5).
- It designs a low power and area efficient charge pump which enables the cNV SRAM to quickly enter into sleep mode, and therefore further increases the power off opportunities of cNV SRAM based memories (Section 6).
- It discusses the impact of memory size and sleep time on the effectiveness of cNV SRAM and provides the general guidelines to apply cNV SRAM in practice. Architecture-level simulation results show that the proposed cNV SRAM can achieve noticeable energy reduction with minimal area overhead and little performance degradation (Section 7).
- The cNV SRAM is compared with the five existing NV SRAMs (Section 8).

The rest of the paper is organized as follows. In Section 2, the motivation for cNV SRAM is introduced. The circuit and architectural control implementation of cNV SRAM is presented in Section 3. The design parameters including optimum tunneling voltage and size of devices for cNV SRAM are discussed in Section 4. Section 5 shows the layout implementations, read and write time, noise margin of cNV SRAM and Section 6 presents the charge pump design. Architecture-level evaluations and general guidelines for applying cNV SRAM are given in Section 7, followed by the comparison of different NV SRAM in Section 8 and by the conclusion in Section 9.

2 MOTIVATION

In this section, the register files are used to present the motivation of this paper.

In modern microprocessors, register rename technique (RRT) is applied to enhance instruction level parallelism by mapping architectural registers to physical registers. RRT uses a register alias table (RAT) to keep track of the state of each physical register [27]. During the instruction execution, the lifetime of a physical register can be divided into two states: work and standby, as illustrated in Fig. 1 (a). In the work state which is from obtaining the executed result to the read of its last consumer, the physical register stores useful information for the following instructions.

The physical register P1 in the code is taken as an example in Fig. 1 (b). When $Inst_A$ is renamed, P1 is mapped to architectural register r1 and the flags of P1 are updated in RAT; as $Inst_A$ executes, P1 enters the work state to contain the valid data; after its last consumer (LastUser) $Inst_B$ reads the value, P1 becomes standby for recovery until $Inst_L$ redefining its architectural register (Redefiner) commits; and then P1 is unmapped to architectural register in RAT. In the worst case scenario, long latency events such as L2 cache misses occur between the LastUser ($Inst_B$) and the Redefiner ($Inst_L$). However, L2 cache misses may take hundreds of cycles to resolve [28], [29]. During such long service time, P1 stays in the standby state, consuming large leakage energy.

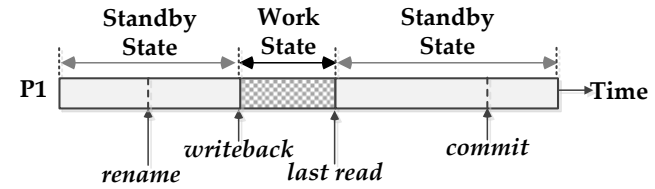
Therefore, the motivation for this paper arises from the following two observations:

1) As instructions pass through the pipeline, registers only spend a small fraction of its lifetime in work state and therefore there is a large opportunity to reduce energy dissipation of on-chip memory. Fig. 1 (c) shows the simulation results of average register lifetime distribution on the integer SPEC2000 benchmarks. It is shown that the average work state of registers is only 20%, and the standby state takes about 80% of the entire lifetime of registers. Such short work time is due to two reasons: i) most registers are read at most once; and ii) some registers are never read because their values are not needed or their consumers obtain the result through bypass logic [30].

2) The contribution of cells in on-chip memory to leakage energy is much larger than that to dynamic energy. Using a modified version of CACTI 5.3 [31], a 1.5 KB 2-read/1-write ported register file is modeled and the breakdown of dynamic energy and leakage energy of register file components is shown in Fig. 2. It can be seen that bit-lines and cells only consume 11% dynamic energy, but they contribute to 63% leakage energy. Since most of the leakage current of bit-lines flows from the cells, reduc-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2014.2375187, IEEE Transactions on Computers

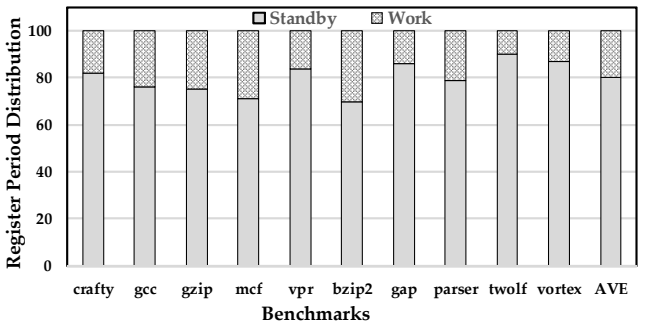
ing the leakage in cells can eliminate the bit-line leakage [32]. Accordingly, low leakage energy cNV SRAM cell can suppress the leakage energy of the entire memory effectively.



(a)

inst	Original code	Rename code	ROB
--	----------	-------	-
A	AND ->r1	AND ->P1	P1
B (LastUser)	SUB<- r1	SUB<- P1	
Many instructions do not need r1 Worst scenario: long latency events happen such as L2 cache misses			
L (Redefiner)	OR ->r1	OR ->P2	P2
M	JMP		-

(b)



(c)

Fig.1. (a) Different states of registers (b) Example code sequence (c) Register period distribution.

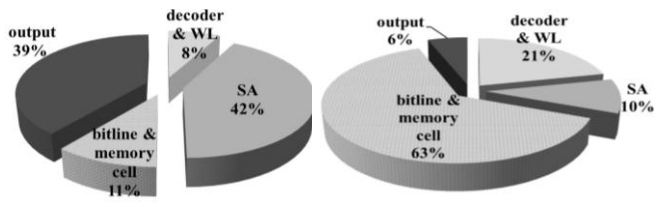


Fig. 2. (a) Dynamic power and (b) Leakage power breakdown of 1.5KB 2-read/1-write ported register file.

Fig. 3 presents the architecture of the proposed cNV SRAM based memory. The fundamental idea is that, cNV SRAM cell consists of conventional SRAM cell with NV component. The conventional SRAM cell is used as working memory with fast access speed and the NV component is used to back up the data with minimized leakage energy. Based on the state of a register, the control logic generates signals and then passes them to power management unit (PMU), charge pump, and memory, placing the corresponding cNV SRAM cells into the appropriate power mode (Power on/off) and enabling back-up/restore process. The circuit-level implementations and architectural mechanisms of cNV SRAM will be discussed in the following section.

3 Proposed cNV SRAM

3.1 Circuit Implementation

Fig. 4 (a) shows the schematic of proposed cNV SRAM. The cell is designed with a conventional 8T-SRAM cell along with coupling capacitor (made by transistor Tc2) and tunneling capacitor (made by transistor Tc1) as NV part. Two additional transistors Nt and Pt form the transmission gate to control the data restoration.

In working mode (WAK="1" and SLP="0"), the state control logic generate control signals to set WAK (wake-up) and clear SLP (sleep). Accordingly, the transmission gate is cut off and the cNV SRAM cell works the same as a conventional 8T-SRAM cell, achieving fast access time, high noise margin, and high robustness to PVT variations [10].

In sleep mode (short low WAK pulse and SLP = high voltage), at the beginning, the microprocessor system enters into standby state while the state control logic enables the charge pump to heighten SLP at a high voltage to make transistor Tc1 Fowler-Nordheim tunneling. Floating gate (FG) is deprived of electrons by tunneling current, and the FG potential is constantly increasing until a steady high voltage [34]. If Q is "1", QB is "0", Nc is turned off, FG keeps this voltage and data in cNV SRAM is backed up as non-volatile "1"; otherwise, Nc is turned on, FG is discharged to ground voltage and data in cNV SRAM is backed up as non-volatile "0". Once the backup process is done, it requests the PMU (see Fig. 3) to switch off the power supply of cNV SRAM and the memory enters into ultra-low leakage energy state.

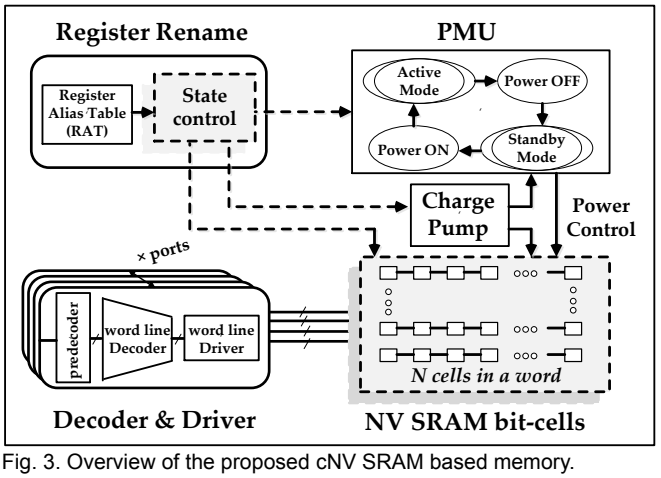


Fig. 3. Overview of the proposed cNV SRAM based memory.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2014.2375187, IEEE Transactions on Computers

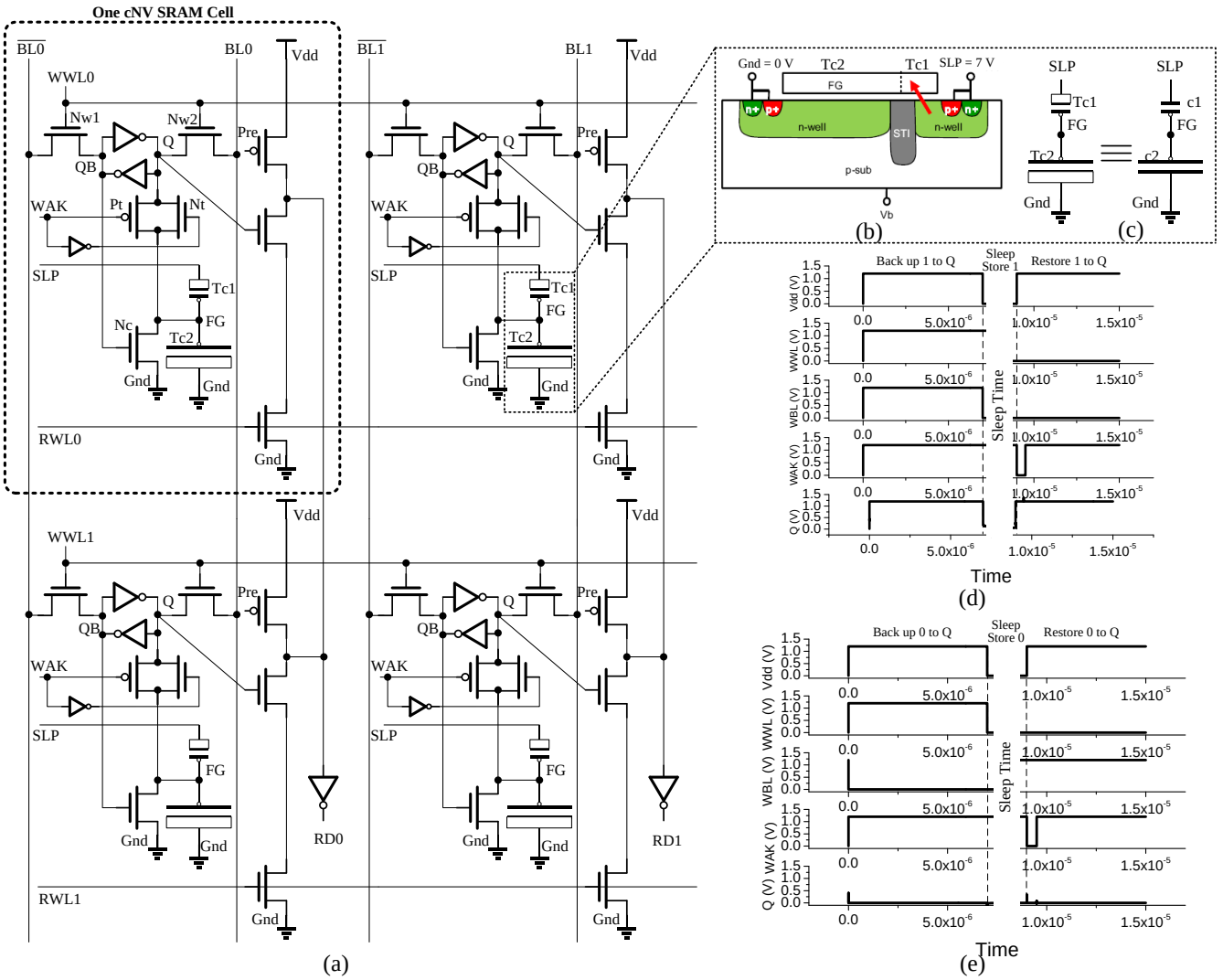


Fig. 4. (a) Schematic of proposed cNV SRAM (b) Structure of planar cNV part of the proposed cell (c) Circuit model for the Fowler-Nordheim tunneling mechanism (d) Process of backing up and restoring "1" for cNV SRAM (e) Process of backing up and restoring "0" for cNV SRAM.

Finally, the processor system is ready to return to the work state while the control logic clears WAK to open the transmission gate and makes the hybrid memories power-on. If FG is at a steady high voltage, Q is pulled up to "1", cNV SRAM is restored to "1"; otherwise, Q is pulled down to "0" and therefore "0" is restored to cNV SRAM cell.

In order to evaluate the characteristics of cNV SRAM, HSPICE simulations are performed based on 65 nm SMIC Technology with 1.2 V power supply. Fig. 4 (d) and (e) shows the backing up and restoring "1/0" processes of cNV SRAM cell: 1) as the write word line WWL is enabled, "1"/"0" is written to Q; 2) cNV SRAM enters its sleep state and the stored data is transferred into its NV part; 3) during the sleep period, the power supply is switched off to minimize the leakage energy dissipation; and 4) when cNV SRAM returns to its work state, the power supply is *ON*, a short low WAK pulse is applied to open the transmission gate, and the data in NV part is restored to the cNV SRAM cell.

Note that, before cNV SRAM enters its sleep mode, the response time of data back-up operation mainly comes from feedback time of charge pump which up-converters the voltage. Accordingly, the sleep time of cNV SRAM (T_{sleep}) does not equal the standby period of the memory ($T_{standby}$) and their relationship can be expressed as

$$T_{sleep} = T_{standby} - T_{cp} \quad (1)$$

where T_{cp} is feedback time of charge pump (see details in Section 6). If the standby period of the memory is larger than the feedback time of charge bump, the energy savings will be achieved with the proposed technique.

Moreover, to avoid the performance penalty the restoration time of cNV SRAM, should be short enough to return the memory from the standby state to the work state. The detailed analysis will be provided in Section 4.

3.2 Architectural Control Implementation

The working process of cNV SRAM depends on the architectural state control logic. Taking a cNV SRAM based register file as an example, the developed cNV SRAM state control logic is set by the rename logic, as the register renaming logic tracks the state of physical registers.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2014.2375187, IEEE Transactions on Computers

Since most renaming logic is placed close to register files such as Alpha 21264, the power consumed by the signal transmission from renaming logic to register files can be negligible.

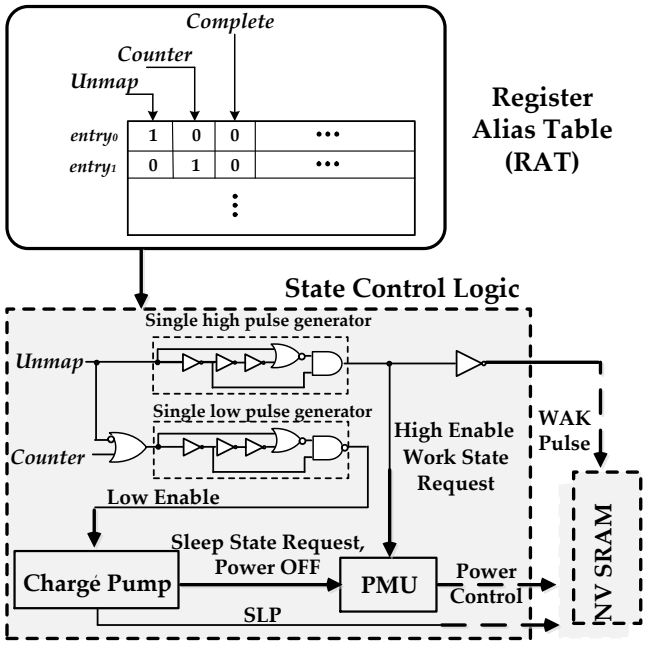


Fig. 5. State Control Logic for cNV SRAM.

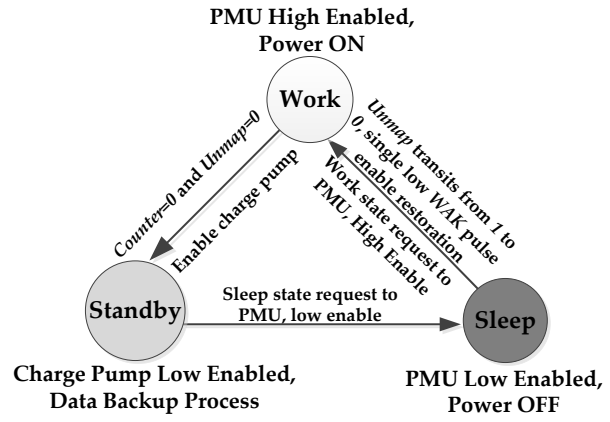


Fig. 6. cNV SRAM State Machine.

Fig. 5 shows the state control logic for cNV SRAM. In a conventional superscalar microprocessor, the renaming logic has one entry for each physical register to record its status [27]. Typical renaming logic contains which contain *Unmap* flag, *Complete* flag, and *Counter*: *Unmap* flag indicates whether a register is mapped to an architecture register; *Counter* is used to record the number of consumers that have not read the information of a register; and *Complete* flag denotes if a register has been redefined [27]. As discussed in Section 3.1, the working process of cNV SRAM is based on the state (work/standby) of registers (see Fig. 1). Such register state information has been extensively studied in late register allocation and early register release [28, 36, 59-61, 65], dynamic register renaming [62], dynamic voltage scaling [63], and register-caching [64]. Those techniques are orthogonal to cNV SRAM and they could be applied in conjunction to further optimize the power efficiency of register files. To detect the register state, researchers have developed two major mechanisms: compiler-assist and hardware-only. The compiler-assist schemes use the compiler analysis to determine the state of each register and pass the information to microarchitecture. However, to make the information available to hardware, a change of ISA is required by defining extra instruction bits or adding new instructions. Alternatively, the hardware-only schemes detect the register state without compiler support. Note that, all of those register state detection mechanisms can be used to implement cNV SRAM.

Here, to reduce the implementation complexity, we present a simple hardware scheme, as shown in Fig. 5. Based on the conventional register rename logic, *Unmap* and *Counter* flags are used to generate the signals to control charge pump, PMU, thereby enabling work/sleep modes of cNV SRAM cells. As will be discussed in Section 4.2, the restoration time from the sleep state to the work state of cNV SRAM is much shorter than the clock cycle of typical microprocessors. In a real microprocessor, typically there is more than a cycle between register renaming of an instruction and its register access, and therefore *Unmap* flag can be used to enable the restoration process and switch the register from the sleep state to the work state. Note that, in the case of a branch misprediction, interrupt, or exception, the data could be copied from its NV part into 8T SRAM cells for recovery.

As also shown in Fig. 5, PMU is responsible for providing power control to register files based on its state. Recently, a supply switching technique with ground collapse (SSGC) [26] realizes faster switching speed and lower implementation overhead as compared to earlier approaches. The proposed cNV SRAM could utilize SSGC or other proposed PMU techniques to power on/off memories [40], [41]. Additionally, a charge pump is enabled as a register enters standby state and the content of 8T SRAM cell can be stored in its NV part. The charge pump design will be presented in Section 6.

As shown in Fig. 6, the working process of the state control logic is detailed as follows.

- **From work state to standby state**: *Counter* = "0" and *Unmap* = "0", which indicates the last read is completed and this register becomes standby state. Therefore, a signal is generated to enable charge pump for back-up process. The designed charge pump is low enabled, which will be discussed in Section 6.
- **From standby state to sleep state**: After the back-up is successful, it requests sleep state to PMU and the power supply of cNV SRAM is off.
- **From sleep state to work state**: When *Unmap* transits from "1" to "0", the control logic requires the work state to PMU which will switch on the power supply of cNV SRAM. At the same time, a low WAK pulse will be generated to enable restoration process and the content of NV is copied to 8T SRAM cell for the normal program execution with fast access speed.

To avoid performance penalty, the state control logic are operated in parallel with the word-line decoder of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2014.2375187, IEEE Transactions on Computers

register file. Due to the small size of the state control logic, its access delay is much smaller and can be overlapped with the word-line decoding. Therefore, the state control logic will not create new critical paths in the register file. At the same time, the control logic also introduces a small amount of energy overhead. However, the energy savings achieved by the proposed technique can offset this overhead, as shown in Section 7.

4 Implementation Considerations of cNV SRAM Cell

4.1 Optimum Tunneling Voltage

As been discussed, Fowler-Nordheim tunneling mechanism is critical for data retention in cNV SRAM. Therefore, the tunneling behavior of 2.35 nm thick gate oxide (65 nm SMIC Technology) is analyzed to obtain the optimum tunneling voltage for cNV SRAM. Fig. 4 (b) shows the cNV part of the proposed cell designed in planar. Both oxides in coupling capacitor (Tc2) and tunneling capacitor (Tc1) have the same thickness and can be manufactured in the same process step. Neither special structures nor special process steps, differing from the conventional CMOS process, are involved in planar cNV SRAM fabrication.

The tunneling current through gate oxide of tunneling capacitor (transistor Tc1) in Fig. 4 (c) follows the Fowler-Nordheim tunneling mechanism, as given by [34]

$$J_{F-N} = AE_{ox}^2 exp(-B/E_{ox}) \quad (2)$$

where $J_{F\text{-}N}$ is current density, E_{ox} is electrical field, and both A and B are constants. Researchers have shown that Fowler-Nordheim tunneling is a safe way to inject charges repeatedly through a thin dielectric film and 6-8 V is desired to be applied across the tunneling oxide for charge injection [33], [34]. Considering the capacitive division by C1 and C2 in Fig. 4 (c), the voltage between SLP and ground at 7-10 V with C2/C1> 10 is set in the simulation. Fig. 7 shows the simulation results based on 65 nm SMIC Technology. As observed, when SLP is set at 7V, after Fowler-Nordheim tunneling, the voltage of FG keeps at 1.081 V which is higher than that with 8-10 V SLPs. This is because for PMOS transistors in 65 nm SMIC Technology, Fowler-Nordheim tunneling is about 6.4 V as shown in Fig. 7. As the voltage of SLP is over 8V, more electrons escape, reaching a higher electron pressure. As a result, as the voltage SLP becomes "0", excess electrons leak out quickly and the FG voltage is reduced to a lower level. As also observed in Fig. 7, with a 10 V SLP, the FG voltage drops down to 1.028 V. Additionally, as compared with 8-10 V SLP, 7 V SLP is larger than Fowler-Nordheim tunneling voltage (6.4 V) and it is beneficial to save the power consumption related to the production and consumption of the boosted internal power supply [34]. Therefore, 7 V SLP is used to drive cNV SRAM in the implementation.

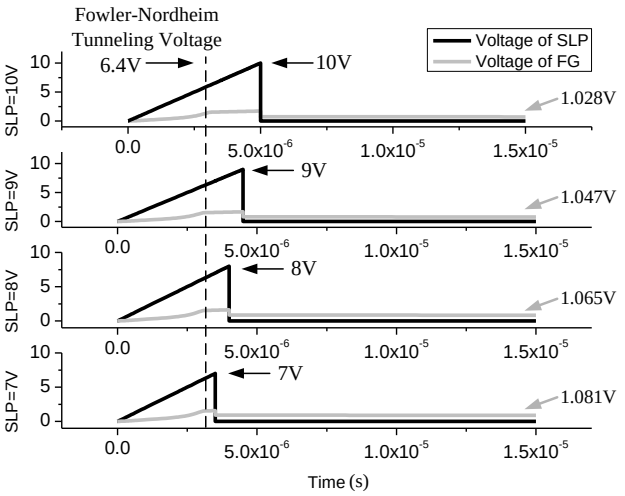


Fig. 7. Optimum tunneling voltage of cNV SRAM.

4.2 Optimum Size of Transmission Gate

As discussed in Section 3.1, the restoration time of cNV SRAM from the sleep mode to the work mode may cause performance penalty to the entire system. In particular, in an cNV SRAM cell as shown in Fig. 4 (a), Nt and Pt form the transmission gate which controls the data restoration and significantly influences the restoration time. In order to achieve similar transmission speed of NMOS (Nt) and PMOS (Pt) transistors, the size (W/L) of PMOS is set as 2 times as that of NMOS. Note that the restoration time of "0" from FG to Q is much smaller than the restoration time of "1". This is because, if a "0"is restored to Q, an enhanced QB voltage will turn on Nc to pull Q down to "0", which speed up the restoration process as a positive feedback. Therefore, the restoration time of "1" is longer and it determines the overall restoration performance. Fig. 8 shows the delay time of restoring "1"with various size of Nt (the smallest W/L for NMOS in 65nm SMIC is 120 nm/60 nm). It shows that the delay time increase is ~8% as Nt size increases from 120 nm/60 nm to 540 nm/60 nm.

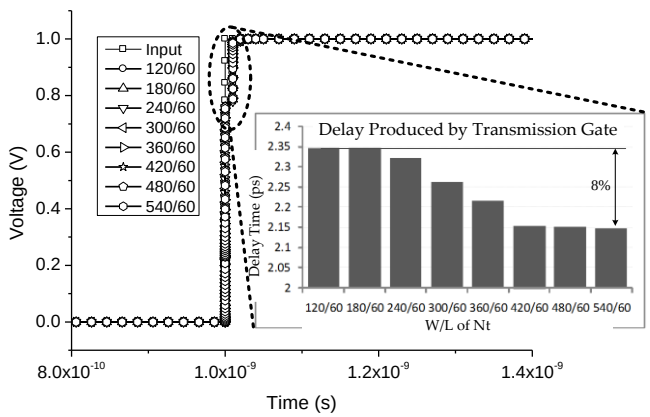


Fig. 8. Delay produced by transmission gate with the increase of W/L of Nt when the data "1" is restored from FG to Q.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2014.2375187, IEEE Transactions on Computers

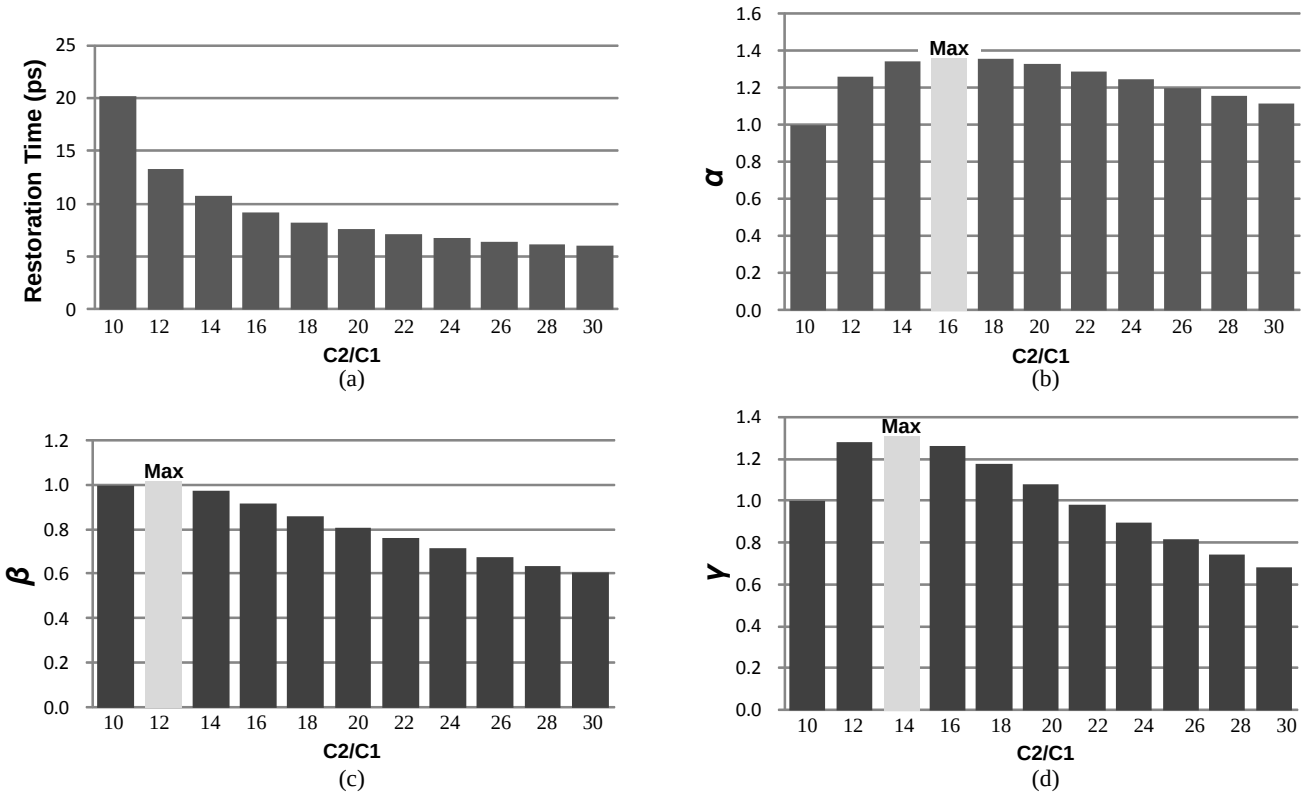


Fig. 9. (a) Restoration time of "1" from FG to Q for cNV SRAM (b) Normalized per unit C2/C1 contributes to the speed (c) Normalized per unit C2/C1 contributes to the amount of injection charges (d) Normalized per unit C2/C1 contributes to both of the speed and the amount of injection charges.

With the minimum sized Nt (120 nm/60 nm), the delay time is still less than 2.35 ps, which is negligible. However, the layout area significantly increases with a large Nt, inducing considerable overhead [5]. Thus, in the design, the minimum sized Nt (120 nm/60 nm) is used and the size of Pt is 240 nm/60 nm.

4.3 Tc1 and Tc2 Size

As discussed in Section 4.1, as the coupling capacitor (C2) becomes larger than the tunneling capacitor (C1), the voltage between SLP and ground drops mostly across the tunneling. In cNV SRAM, as C2 is enlarged, it significantly influences the restoration time, layout area and injection charges on FG.

As mentioned earlier, the restoration time of "1" determines the wake-up speed of cNV SRAM. Fig. 9 (a) shows that with the increase of C2/C1 from 10 and 30, the restoration time decreases from 20 ps to 6 ps. Such fast wake-up speed enables cNV SRAM works well with modern high performance microprocessors. Considering area overhead, C1 is set as the minimum size 120 nm×60 nm (W=120 nm and L=60 nm for transistor Tc1).

Determining C2/C1 requires tradeoff between the restoration time and injection charges on FG. Accordingly, three parameters α, β, and γ are defined as

$$\alpha = C1/C2 \cdot (1/t_{BK}) \quad (3)$$

$$\beta = C1/C2 \cdot V_{FG} \quad (4)$$

$$\gamma = \alpha \cdot \beta \quad (5)$$

where t_{BK}, V_{FG}, α, β, and γ represent the back-up time, voltage of FG, contribution of unit C2/C1 to the speed (the reciprocal of restoration time), to the amount of injection charges, and to both of speed and charges (product of α and β), respectively.

The normalized values of α, β, and γ with various C2/C1 are shown in Fig. 9 (b), (c), and (d). As C2/C1 is 16, it achieves maximum average utility in terms of speed. When C2/C1 is 12, it can result in maximum inject electrons from n-well to FG. While considering both speed and charges, 14 of C2/C1 realizes the optimized γ in cNV SRAM design with 11 ps restoration time (Fig. 9 (a)). Accordingly, the restoration time of cNV SRAM is much shorter than the clock cycle of typical microprocessors and therefore 11 ps restoration time is expected to be hidden easily in the pipelines based on the architectural control mechanism as discussed in Section 3.2., incurring no impact on the performance, Therefore, C2/C1 is set as 14 and the restoration time of the proposed cNV SRAM imposes little performance penalty.

When C2/C1 is 14, the gate area of Tc2 is as 14 times as that of Tc1 and Tc2 with various W and L are listed in Table 1. With various sizes of Tc2, the normalized amount of injection charges on FG is shown in Fig. 10. It can be seen that when Tc2 is sized as W=420 nm and L=240 nm, C2 is most effective for Fowler-Nordheim tunneling. Accordingly, in the proposed cNV SRAM cell, sizes Tc1 and Tc2 are as follows: W_{c1}=120nm, L_{c1}=60nm, W_{c2}=420nm, and L_{c2}=240nm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2014.2375187, IEEE Transactions on Computers

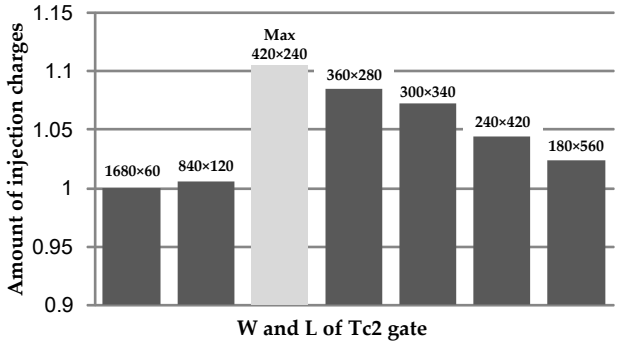


Fig. 10. Amount of the injection charges on FG with various W and L of Tc2.

TABLE 1
Size of Tc2 gate

Tc2 gate	Size						
W (nm)	1680	840	420	360	300	240	180
L (nm)	60	120	240	280	340	420	560
Area (nm²)	C2 area=C1 area×14=120×60×14=100800						

5 Characteristic of cNV SRAM Cell

5.1 Speed of cNV SRAM Cell

TABLE 2
Characteristic of cNV SRAM Cell

Cell	Access Time (s)	Write 1 Time (s)	Write 0 Time (s)	SNM
cNV SRAM	6.7E-11	1.1E-11	1.4E-11	0.31 V
Conventional 8T	6.7E-11	1.0E-11	1.2E-11	0.31 V
Penalty	0	10%	17%	0

As shown in Fig. 4 (a), a cNV SRAM Cell is a conventional 8T-SRAM cell along with its NV part. When a cNV SRAM cell is working (WAK="1" and SLP="0"), its NV part is turned off. However, the NV part, including the coupling capacitor, the tunneling capacitor, and one transmission gate, would add the additional capacitance to Q/QB and degrade the write speed of the cell. As listed in Table 2, as compared to conventional 8T-SRAM cell, the write 1 time and the write 0 time, respectively, increase by 10%, and 17%. Alternatively, during read operation, since the data stored at Q/QB has been kept by bi-stable structure, the access time therefore is not influenced by the additional capacitance. As also shown in Table 2, similar to typical memory in modern processors, the access time of cNV SRAM is much longer than its write time and it determines the performance of cNV SRAM. Accordingly, the proposed technique does not induce performance penalty as compared to the conventional SRAM.

5.2 Noise Margin of cNV SRAM Cell

In this subsection, the noise margin of cNV SRAM is discussed. As shown in Fig. 4 (a), in the read operation, low WWL turns off Nw1 and Nw2 to isolate the read path and Q/QB, thus eliminating read disturbance. Due to such isolation, the read SNM (static noise margin) of cNV SRAM is nearly the same as its hold SNM. What is more, SNM is dependent on the supply voltage, threshold voltage of transistors in the bi-stable structure, cell ratio, and pull-up radio, but it does not depend on the capacitance load at Q/QB [53]. Accordingly, the additional capacitance induced by the NV part of cNV SRAM has little impact on its SNM. Fig. 11 compares the SNM of conventional 8T cell and cNV SRAM Cell and it shows that they have the same SNM (0.31 V).

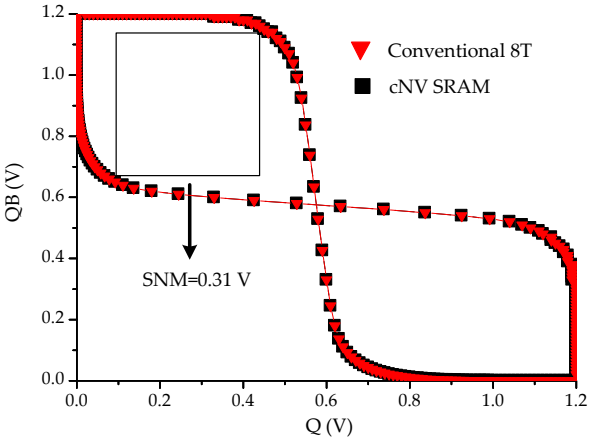


Fig. 11. SNM of conventional 8T cell and cNV SRAM cell.

5.3 Layout Implementations of cNV SRAM

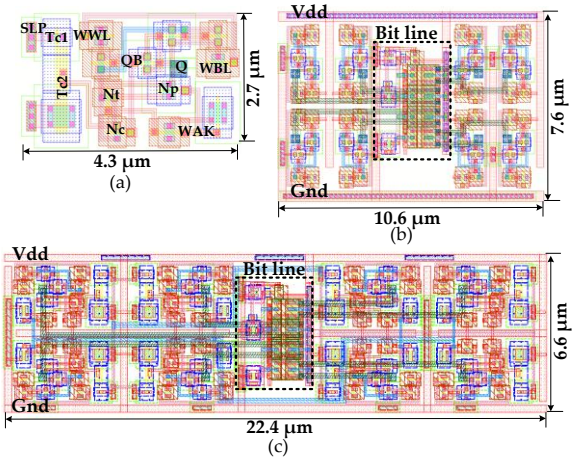


Fig. 12. (a) Layout of cNV SRAM cell (b) Layout of typical 8-bit 8T SRAM array (c) Layout of 8-bit cNV SRAM array.

Based on 65 nm SMIC Technology design rules, the layout design of a cNV SRAM cell, an 8-bit cNV SRAM array, and a typical 8-bit 8T SRAM array are implemented, as shown in Fig. 12. The cNV SRAM cell has a similar topology to that of the conventional 8T cell except the routing of the cNV part including Tc1, Tc2, transmission gate, and Nc, which lead to a larger cell layout area. The additional signals "SLP" and "WAK" result in higher interconnection complexity. To reduce the area overhead, Tc1, Tc2,

transmission gate, and Nc are routed in compact style, as shown in Fig. 12 (a). As compared to conventional 8T SRAM array, 8-bit cNV SRAM array consumes 45% more area. It should be noted that, similar to the cell design in [36], the area overhead of cNV SRAM is also influenced by the number of memory ports. Adding cNV part to a heavily multi-ported memory is expected to induce only a small amount of area overhead. For example, for a 6-read/6-write ported cNV SRAM based memory, its area overhead can be reduced below 25%. Given the energy savings discussed as Section 7, this area increase is tolerable.

6 Charge Pump Design for cNV SRAM

To implement cNV SRAM based on-chip memories, the charge pump is required to produce high voltage for Fowler-Nordheim tunneling with small feedback delay and efficient layout area. Since 7 V *SLP* signal is optimal for the proposed cNV SRAM cell, a simple NMOS-type Dickson's charge pump with high SLP/V_{dd} ratio [37], [38] is designed to produce 7 V boosted tunneling voltage. The schematic is shown in Fig. 13. Without considering the body effect, the generated *SLP* voltage can be expressed as [39]:

$$SLP=V_{dd}+N\left(\frac{C_{pmp}V_{dd}}{C_{pmp}+C_p}-\frac{I_{pp}/f_{osc}}{C_{pmp}+C_p}\right)-\sum_{i=1}^{N}V_{thi} \quad (6)$$

where C_{pmp}, C_p, N, and f_{osc} represent pumping capacitor, parasitic capacitor, number of pumping stages, and pumping frequency, respectively.

With the reduced number of the stages and larger pumping capacitors, the feedback delay grows. After careful design consideration, the charge pump is implemented for cNV SRAM and the parameters are listed in Table 3. The charge pump uses 10 stages with 100 fF pumping capacitor and achieves 7 V *SLP* and 1 µA current (I_{pp}) with 353 ns feedback delay, enabling a fast back-up process and further enhances the energy efficiency of cNV SRAM. The power consumption of the designed charge pump is about 56 µw.

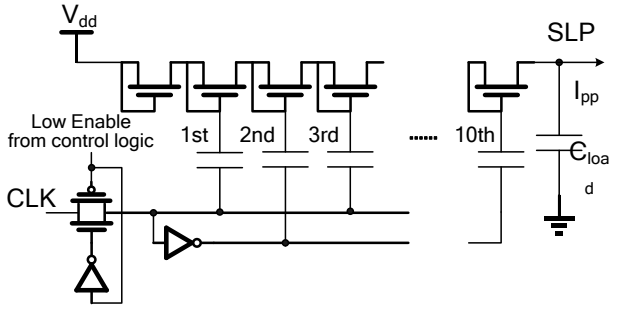


Fig. 13. 10 stages NMOS-type Dickson's charge pump.

As also shown in Table 3, the layout area of the designed charge pump is 32 µm×35 µm which approximately equals the area of 64 cNV SRAM cells (see Fig. 12). In terms of memory driving ability, the designed charge pump can provide 7 V *SLP* and 1 µA current, which are typically required by 1K-bit memory [38]. Since one charge pump is needed for 1K-bit cNV SRAM, the layout area overhead of charge pumps is about 1/16 (1K/64) of cNV SRAM cell area. According to CACTI 5.3 [31], in 65 nm 1-read, 1-write ported 1.5 KB SRAM and 8 MB SRAM, the percentages of cells area are about 27% and 52%, respectively. Thus, the area overheads of the designed charge pumps in cNV SRAM are only 1.7% and 3.3%.

TABLE 3
Parameters of Charge Pump

SLP	Stages	Pumping capacitor	Feedback delay	Layout area	Power
7 V	10	100 fF	353 ns	32 ×35 µm²	56 µw

7 Energy Dissipation of cNV SRAM System

In this section, the energy efficiency of the proposed cNV SRAM is evaluated firstly based on architecture-level simulations. Then, the impact of memory size and sleep time are analyzed on the effectiveness of cNV SRAM and provide general guidelines to apply cNV SRAM in practice.

7.1 Architecture-level Evaluations

To investigate the energy efficiency of cNV SRAM, architecture-level evaluations are carried out by execution-driven simulations using an extensively modified version of the Simple Scalar simulator [43]. The microprocessor is assumed to work at 1-GHz frequency and 1.2V supply voltage. Table 4 describes the microprocessor architecture. The register file configurations are similar as Intel architecture: 128 integer register files and 128 floating-point register files are included in the microprocessor. Table 4 describes the processor architecture. Ten integer SPEC 2000 benchmark suite compiled are used for the Alpha 21264 processor, based on the reference input set. The benchmarks are simulated after 20 million fast-forward initialization phase.

Fig. 14 shows the leakage energy reduction achieved by cNV SRAM as compared to the conventional SRAM design. Here, the dissipations of additional logic such as architectural control logic and charge pump are considered. For the dissipations in the auxiliary structures, the state control logic introduces a very small energy overhead and thus is not shown; considering one additional bit for RAT, it is modeled as traditional SRAM structure and then the energy dissipation is included. As shown in Fig. 14, with the energy overhead of additional logic, there are still average leakage energy reductions of 86% in register file, and because the leakage energy of register files attributes to 60% of core leakage without cache memory and to 15%-36% of full system-on-chip leakage, [51], [52], [59] at least 13% leakage energy reduction is achieved in overall processor. This suggests that by using cNV SRAM, the proposed technique is very effective in implementing low energy register files.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2014.2375187, IEEE Transactions on Computers

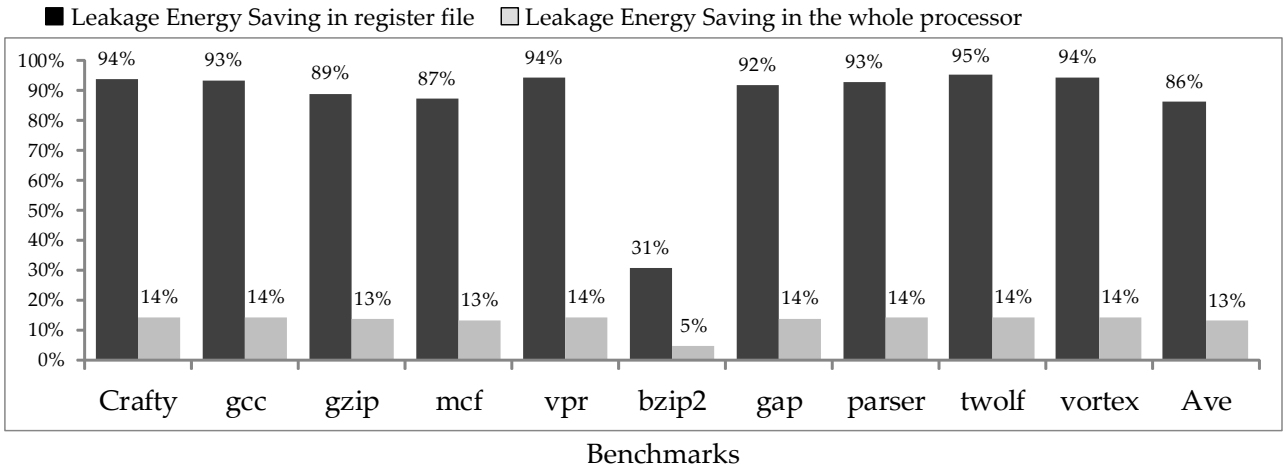


Fig. 14. Architecture-level evaluations of the leakage energy saving of cNV SRAM over the conventional SRAM.

As also observed in Fig. 14, the energy reduction achieved by different applications is not uniform. This is because different applications have different register state distribution, which affect the percentage of operation time when cNV SRAM can stay in sleep mode. At the same time, different microprocessors achieve different register state distribution, which further influences the energy efficiency of cNV SRAM. While running different applications (SPEC 2000, SPEC 95, Mibench and DSP stone) on different microprocessors (Blackfin and out-of-order processors with various architectural parameters), researchers have shown that 39.3%-80% of operation time that registers stay in sleep mode [36], [59], [60], [61]. Accordingly, significant opportunities exist in modern microprocessors to apply cNV SRAM to achieve leakage energy savings.

It's worth mentioning that, the proposed cNV SRAM is expected to achieve more energy savings for advanced microprocessors. For example, in multi-core multi-thread microprocessors, when one thread gets blocked due to memory stall [61], the corresponding cells of this thread can be placed in sleep mode for power reduction. Therefore, the cNV SRAM is expected to yield noticeable power savings.

TABLE 4
Processor Organization

Reorder buffer	128 entry
Register File	128 integer and 128 floating-point
Machine Width	4-wide fetch, 4-wide issue, 4-wide commit
Load/Store Queue	48 entry load and 48 entry store
Function Units	4 IntALU, 4 FP ALU, 1 Int MULT/DIV, 1 FP MULT/DIV
BTB	2048 entry, 4-way set-associative
Branch Predictor	Combined with 1024 entries 2-level global predictor with 8 bits history width
L1 I/D Cache	32 KB, 4-way set-associative, 1 cycle hit time
L2 Cache Unified	512 KB, 4-way set-associative, 6 cycles hit time
Memory	64 bit wide, 100 cycles

7.2 General Guidelines for using cNV SRAM

The sensitivity of cNV SRAM is further analyzed and finally guidelines are provided for using cNV SRAM in different conditions. As discussed in Section 3, in working state, cNV SRAM based memory works as a conventional SRAM and they consume similar dynamic energy; in sleep mode, the leakage energy dissipation of cNV SRAM is reduced and therefore the effectiveness of cNV SRAM strongly depend on the sleep time as well as the memory size. As cNV SRAM based on-chip memories (@NV) switch between the work state and the sleep state, the transition process including control signal generation, voltage upconverter using charge pump, data back-up, date restoration, causes energy overhead. Note that, once cNV SRAM enters the sleep mode, the power supply is turned off and the energy consumed does not increase with the sleep time. However, the transition energy overhead will increase with the size of memories.

Using a modified version of CACTI 5.3 [31] and HSPICE simulations, the energy dissipations of cNV SRAM is compared to the conventional SRAM with various memory sizes and sleep time in 65 nm SMIC Technology. The typical sizes of four different on-chip memories in Sandy Bridge architecture based Intel CPU i5 and i7 [42] are considered: register file = 1.5 KB, L1 cache = 32 KB, L2 cache = 256 KB, and L3 cache = 8 MB. Fig. 15 shows the leakage energy curves of four different memories implemented with cNV SRAM (@NV) and conventional memories (@Con.) technologies. It is shown that the required sleep time is longer for larger memories to show the effective of cNV SRAM. For instance, as the sleep time is longer than 132 ns, cNV SRAM based register file (RF) is able to achieve energy savings. In terms of L3 cache, the sleep time of cNV SRAM is required to be longer than 281 ns to realize lower energy dissipation. The reason is that larger memories employ more charge pumps and complex control logic for data back-up and restoration, which increases the energy overhead and influences the effectiveness of cNV SRAM. Another important observation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2014.2375187, IEEE Transactions on Computers

obtained in Fig. 15 is that more energy savings can be achieved for larger memories with the proposed technique. As the sleep time is 1 µs, cNV SRAM based register file, L1 cache, L2 cache, and L3 cache enable 1 nW, 14 nW, 81 nW, and 1.7 µW energy savings, respectively. Furthermore, the achieved energy savings increase with the sleep time. As a consequence, the energy efficiency of cNV SRAM is more pronounced for large on-chip memories with long idle time.

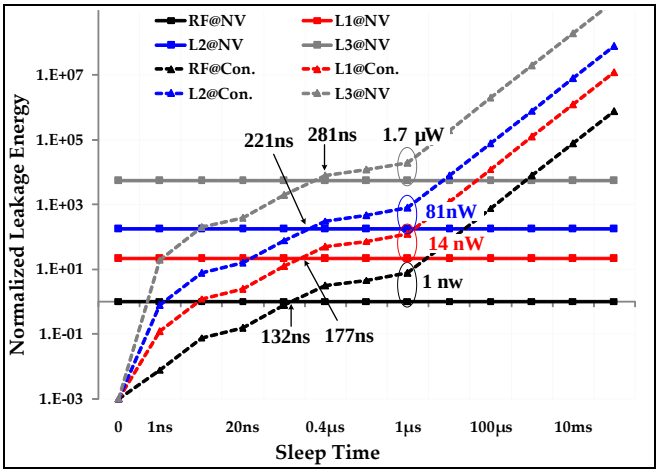


Fig. 15. Energy dissipation comparison of cNV (@NV) SRAM based and conventional (@Con.) SRAM based memory with increasing sleep time and memory sizes.

8 Comparison of Different NV SRAM

In this section, the proposed cNV SRAM is compared with the five existing NV SRAMs: FeRAM/SRAM [54], NEM/SRAM [55], MRAM/SRAM [56], NCs/SRAM [57], and RRAM/SRAM [58]. The results are shown in Table 5. Since the existing NV SRAM technologies are not compatible with traditional CMOS technology, the proposed cNV SRAM possesses lowest cost at all. As also shown in Table 5, all of the NV SRAM designs trade off among silicon area, performance, noise margin, and power efficiency. For example, as compared to conventional SRAM, NEM/SRAM improves the read and write performance with higher noise margin, but its cell employs much more silicon area (~10X of cNV SRAM cell). Although the proposed cNV SRAM needs more write time (up to 2ps), the write path is usually a noncritical path and the write time is much smaller than the access time of a memory. Accordingly, the write speed penalty of cNV SRAM would not influence the performance of the entire system. However, cNV SRAM achieves significant power savings and efficient silicon area, without inducing penalty in its access time and robustness. Therefore, the proposed cNV SRAM can be a preferable solution for implementing power-efficient memory in microprocessors, especially for applications with area considerations such as embedded systems.

9 Conclusion

A CMOS compatible Non-volatile SRAM has been presented in this paper. The technique uses conventional 8T SRAM in the work state for fast operation; in sleep state, it backs up the data with non-volatile parts and switch off the power supply to eliminate leakage energy. The schematic and architectural schemes are developed to implement cNV SRAM, while considering multiple key performance parameters including energy dissipation, layout area, and speed. Specifically, 7 V tunneling voltage, coupling/tunneling capacitors with W_{c1}=120 nm, L_{c1}=60 nm, W_{c2}=420 nm, and L_{c2}=240 nm, and minimize size of transmission gate are used in cNV SRAM cell. Additionally, a 10 stages NMOS-type Dickson's charge pump with 100 fF pumping capacitor, 7 V *SLP*, and 1 µA I_{pp} is designed to speed up the data back-up process, thereby achieving a fast transition to sleep state and further enhancing the energy efficiency of cNV SRAM. With ten integer SPEC 2000 benchmark suite compiled for the Alpha 21264 processor, the architecture-level evaluations are performed and experiment results show that even with the energy overhead of additional logics, there are still leakage energy reductions of 86% on average, as compared to conventional SRAM. Additionally, based on the typical on-chip memories in Sandy Bridge architecture, the impact of the sleep time and memory size is analyzed in detail and general guidelines for using cNV SRAM is provided: with the reasonable idle time and size, cNV SRAM is preferred in larger on-chip memories with longer idle time. Finally, the proposed cNV SRAM is compared with the five existing NV SRAMs and it shows that the proposed cNV SRAM can be a preferable solution for implementing power-efficient memory for applications with area considerations such as embedded systems. The future investigations would include extension of the proposed NV SRAM to deal with multi-core and multi-thread workloads and tape out memory chip for validation.

Acknowledgment

This work was supported in part by the ND EPSCoR under Grant FAR0021960, the National Natural Science Foundation of China under Grant 61204040, the Ph.D. Programs Foundation of Ministry of Education of China under Grant 20121103120018, and the Plan Program of Beijing Education Science and Technology Committee under Grant JC002999201301.

References

[1] A. Valero, S. Petit, J. Sahuquillo, P. L. Pez, and J. Duato "Design, Performance, and Energy Consumption of eDRAM/SRAM Macrocells for L1 Data Caches," *IEEE Transactions on Computers*, vol. 61, No. 9, pp. 1231-1242, Sep. 2012.

[2] J.P. Kulkarni, A. Goel, P. Ndai, and K. Roy, "A Read-Disturb-Free, Differential Sensing 1R/1W Port, 8T Bitcell Array," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19 , no. 9, pp. 1727 – 1730, Sep. 2011.

TABLE 5
Comparison of the different NV SRAM

	2001 [54]	2009 [55]	2009 [56]	2010 [57]	2014 [58]	Our Work
Cell	[1]FeRAM/SRAM	[2]NEM/SRAM	[3]MRAM/SRAM	[4]NCs/SRAM	[5]RRAM/SRAM	cNV SRAM
Cell Area	[6]1.22 μm²+FV	120 μm²	28.6 μm²	----	12.5 μm²	11.6 μm²
Leakage Energy Saving	[7]~	Up to 85%	[7]~	[7]~	[7]~	[8]86%
CMOS Compatible	×	×	×	×	×	√
Average Access Time Influence	0	-10%	----	+6%	+26% +2.35 ps	0
Write Time Influence (1/0)	+1.2 ns/+1.2 ns	-60%/-60%	----	+6%	+1.54 ps/+0.31 ps +4%/+1%	+1 ps/+2 ps +10%/+17%
SNM Influence (hold/read)	0/0	+112%/+254%	----	----	-27%/0	0/0
Technology Node	250 nm	65 nm	150 nm	70 nm	32 nm	65 nm

[1]FeRAM: Ferroelectric RAM; [2]NEM: Nanoelectro-mechanical; [3]MRAM: Magnetic RAM; [4]NCs: Nano-crystals; [5]RRAM: Resistive RAM; [6]FV: Ferroelectric capacitors and via-stacked-plug; [7]Depending on size and idle time; [8]For the processor in Table 4 is 86%, otherwise depending on size and idle time.

[3] R. Jotwani, S. Sundaram, S. Kosonocky, A. Schaefer, V. Andrade, A. Novak, and S. Naffziger, "An x86-64 Core in 32 nm SOI CMOS," *IEEE J. Solid-State Circuits,* vol. 46, no. 1, pp. 162–172, Jan. 2011.

[4] H. McIntyre, S. Arekapudi, E. Busta, T. Fischer, M. Golden, A. Horiuchi, T. Meneghini, S. Naffziger, and J. Vinh, "Design of the Two-Core x86-64 AMD "Bulldozer" Module in 32 nm SOI CMOS ," *IEEE Journal of Solid-State Circuits*, vol. 47 , no. 1, pp. 164 – 176, Jan. 2012.

[5] ITRS, http://www.itrs.net/Links/2012ITRS/Home2012.htm

[6] A. Do, Z. Kong, and K. Yeo, "Hybrid-Mode SRAM Sense Amplifiers: New Approach on Transistor Sizing," *IEEE Transactions on Circuits and Systems II*, vol. 55, No. 10, pp. 986-990, Oct. 2008.

[7] B. Maric, J. Abella, and M. Valero, "APPLE: Adaptive Performance-Predictable Low-Energy Caches for Reliable Hybrid Voltage Operation," *Proc. 50th ACM/EDAC/IEEE Design Automation Conference (DAC),* pp. 1-8, 2013.

[8] D. Marculescu, "On the Use of Micro Architecture-driven dynamic Voltage Scaling," *Proc. Workshop on Complexity-Effective Design,* 2000.

[9] M. Ishida, T. Kawakami, A. Tsuji, N. Kawamoto, M. Motoyoshi, and N. Ouchi, "A Novel 6T-SRAM Cell Technology Designed with Rectangular Patterns Scalable Beyond 0.18 μm Generation and Desirable for Ultra High Speed Operation," *Proc. Electron Devices Meeting,* pp. 201-204, 1998.

[10] S. Jain and P. Agarwal, "A Low Leakage and SNM Free SRAM Cell Design in Deep Sub Micron CMOS Technology," *Proc. 19th International Conference on Embedded Systems and Design Held jointly with 5th International Conference on,* 2006.

[11] J. Kulkarni, K. Kim, and K. Roy, "A 160 mV, Fully Differential, Robust Schmitt Trigger Based Sub-threshold SRAM," *Proc. International Symposium on Low Power Electronics and Design (ISLPED),* pp. 171-176, 2007.

[12] B. Maric, J. Abella, F. Cazorla, and M. Valero, "Hybrid High-Performance Low-Power and Ultra-Low Energy Reliable Caches," *Proc. 8th ACM International Conference on Computing Frontiers Article,* 2011.

[13] I.J. Chang, D. Mohapatra, and K. Roy, "A Priority-Based 6T/8T Hybrid SRAM Architecture for Aggressive Voltage Scaling in Video Applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21 , no. 2, pp. 101-112, Feb. 2011.

[14] N. Gong, S. Jiang, A. Challapalli, S. Fernandes, and R. Sridhar, "Ultra-Low Voltage Split-data-aware Embedded SRAM for Mobile Video Applications," *IEEE Transactions on Circuits and Systems II*, vol. 59, no. 12, pp. 883-887, Dec. 2012.

[15] Y. Gong, H. Jang, and S. Chung, "Performance and Cache Access Time of SRAM-eDRAM Hybrid Caches Considering Wire Delay," *Proc. 14th International Symposium on Quality Electronic Design (ISQED),* pp. 524-530, 2013.

[16] F. Hameed, L. Bauer, and J. Henkel, "Adaptive Cache Management for A Combined SRAM and DRAM Cache Hierarchy for Multi-Cores," *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 77-82, 2013.

[17] W. Yu, R. Huang, S. Xu, S. Wang, E. Kan, and G. Suh, "SRAM-DRAM Hybrid Memory with Applications to Efficient Register Files in Fine-Grained Multi-Threading," *Proc. 38th Annual International Symposium on Computer Architecture (ISCA),* pp. 247-258, 2011.

[18] A. Makosiej, O. Thomas, A. Vladimirescu, and A. Amara, "Stability and Yield-oriented Ultra-low-power Embedded 6T SRAM cell Design Optimization," *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 93-98, 2012.

[19] K. Kim and Y. Song, "Current and Future High-density FRAM Technology," *Integr. Ferroelectr.*, vol. 61, pp. 3–15, 2004.

[20] T. Andre, J. Nahas, C. Subramanian, B. Garni, H. Lin, A. Omair, and W. Martino, "A 4-Mb 0.18-μm 1T1MTJ toggle MRAM with Balanced Three Input Sensing Scheme and Locally Mirrored Unidirectional Write Drivers," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 301–309, Jan. 2005.

[21] W. Cho, B. Cho, B. Choi, H. Oh, S. Kang, K. Kim, K. H. Kim, D. Kim, C. Kwak, H. Byun, Y. Hwang, S. Ahn, G. Koh, G. Jeong, H. Jeong, and K. Kim, "A 0.18-μm 3.0-V 64-Mb Non-volatile Phase Transition Random Access Memory (PRAM)," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 293–300, Jan. 2005.

[22] J. Li, L. Shi, C. Xue, C. Yang, and Y. Xu, "Exploiting Set-Level Write Non-Uniformity for Energy-Efficient NVM-Based Hybrid Cache," *9th IEEE Symposium on Embedded Systems for Real-Time Multimedia (ESTIMedia)*, pp. 19-28, 2011.

[23] S. Paul, S. Mukhopadhyay, and S. Bhunia, "A Circuit and Architecture Codesign Approach for a Hybrid CMOS–STTRAM Nonvolatile FPGA," *IEEE Transactions on Nanotechnology*, vol. 10, no. 3, pp. 385-394, Mar. 2011.

[24] H. Sun, C. Liu, W. Xu, J. Zhao, N. Zheng, and T. Zhang, "Using Magnetic RAM to Build Low-Power and Soft Error-Resilient L1 Cache," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20 , no. 1, pp. 19-28, Jan. 2012.

[25] J. Chang, M. Huang, J. Shoemaker, J. Benoit, S. Chen, W. Chen, S. Chiu, R. Ganesan, G. Leong, V. Lukka, S. Rusu, and D. Srivastava, "The 65-nm 16-MB Shared On-Die L3 Cache for the Dual-Core Intel Xeon Processor 7100 Series," *IEEE J. Solid-State Circuits*, vol. 42, no. 4, pp. 846–852, Apr. 2007.

[26] H. Kim, B. Lee, J. Kim, J. Choi, K. Choi, and Y. Shin, "Supply Switching with Ground Collapse for Low-leakage Register Files in 65-nm CMOS," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 3, pp. 505–509, Mar. 2010.

[27] M. Moudgill, K. Pingali, and S. Vassiliadis, "Register Renaming and Dynamic Speculation: An Alternative Approach," *Proc. 26th Annual International Symposium on Micro-architecture*, pp. 202-213, 1993.

[28] T. Jones, M. O'Boyle, J. Abella, A. González, and O. Ergin, "Exploring the Limits of Early Register Release: Exploiting Compiler Analysis," *ACM Trans. on Architecture and Code Optimization*, vol. 6, no. 3, pp. 12-30, Sep. 2009.

[29] S. Roy, N. Ranganathan, and S. Katkoori, "State-Retentive Power Gating of Register Files in Multicore Processors Featuring Multithreaded In-Order Cores," *IEEE Trans. on Computers*, vol. 60, no. 11, pp. 1547-1560, Nov. 2011.

[30] R. Sangireddy and A.K. Somani, "Exploiting Quiescent States in Register Lifetime," *Proc. IEEE International Conference on Computer Design,* pp. 368-374, 2004

[31] Hewlett-Packard Company, P. Alto, CA, "CACTI5," [Online].Available: http://quid.hpl.hp.com:9081/cacti.

[32] H. Homayoun, A. Sasan, J. Gaudiot, and A.V. Veidenbaum, "Reducing Power in All Major CAM and SRAM-Based Processor Units via Centralized, Dynamic Resource Size Management," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 11, pp. 2081-2094, Nov. 2011.

[33] N. Ravindra and J. Zhao, "Fowler-Nordheim Tunneling in Thin SiO Films," *Smart Mater. Struct.*, vol. 1, no. 3, pp. 197–201, Jun. 1992.

[34] K. Lee, J. Chun, and K. Kwon, "A Low Power CMOS Compatible Embedded EEPROM for Passive RFID Tag," *Microelectronics Journal* , vol. 41, no. 10, pp. 662-668, Oct. 2010.

[35] K. Nomura, K. Abe, H. Yoda, and S. Fujita, "Ultra Low Power Processor Using Perpendicular-STT-MRAM/SRAM Based Hybrid Cache Toward Next Generation Normally-off Computers," *Journal of Applied Physics*, vol. 111 , No. 7, pp. 07E330 - 07E330-3, Jul. 2012.

[36] O. Ergin, D. Balkan, D. Ponomarev, and K. Ghose, "Early Register Deallocation Mechanisms Using Checkpointed Register Files", *IEEE Trans. on Computers,* vol. 55, no. 9, pp. 1153-1164, Sep. 2006.

[37] J. Dickson, "On-chip High-voltage Generation in MNOS Integrated Circuits Using an Improved Voltage Multiplier Technique," *IEEE J. Solid-State Circuits,* vol. SSC-11, no. 3, pp. 374–378, Jun. 1976.

[38] J. Baek, J. Chun, and K. Kwon, "A Power-Efficient Voltage Upconverter for Embedded EEPROM Application," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 6, pp. 435 – 439, Jun. 2010.

[39] F. Pan and T. Samaddar, Charge Pump Circuit Design. New York: McGraw-Hill, 2006, pp. 45–57.

[40] E. Pakbaznia and M. Pedram, "Design of a Tri-modal Multithreshold CMOS Switch with Application to Data Retentive Power Gating," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 2, pp. 380–385, Feb. 2012.

[41] S. Yang, S. Khursheed, B. Al-Hashimi, D. Flynn, and S. Idgunji, "Reliable State Retention-based Embedded Processors through Monitoring and Recovery," *IEEE Trans. Comput., Aided Des. Integr. Circuits Syst.*, vol. 30, no. 12, pp. 1773–1785, Dec. 2011.

[42] T. Damien, D. Franck , and L. Guillaume, "Intel Core i7 and Core i5," http://www.behardware.com/articles/815-2/intel-core-i7-and-core-i5-lga-1155-sandy-bridge.html

[43] D. Burger and T. Austin, "The SimpleScalar tool set: Version 2.0," *Tech. Report, Dept. of CS, Univ. of Wisconsin-Madison,* 1997.

[44] C. Kim, J. Kim, S. Mukhopadhyay, and K. Roy, "A forward Body Biased Low-leakage SRAM Cache: Device, Circuit and Architecture Considerations," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 25, pp. 349–357, Aug. 2005.

[45] S. Mutoh, S. Shigematsu, Y. Gotoh, and S. Konaka, "Design method of MTCMOS power switch for lowvoltage high-speed LSIs," *Proc. Asian South Pacific Design Autom. Conf.*, pp. 113–116, 1999.

[46] N. Gong, J. Wang, S. Jiang and R. Sridhar, "Clock-biased local bit line for high performance register files," *Electron. Lett.*, vol. 48, no. 18, pp. 1104–1105, Aug. 2012.

[47] K. Flautner, N. Kim, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: Simple techniques for reducing leakage power," *Proc. Int. Symp. Comput. Arch.*, pp. 148–157, 2002.

[48] S. Kaxiras, Z. Hu, and M. Martonosi, "Cache decay: Exploiting generational behavior to reduce cache leakage power," *Proc. Int. Symp. Comput. Arch. Conf.*, pp. 240–251, 2001.

[49] Non-volatile, static random access memory with high speed store capability, by D. Dietrich, P. Ruths, and C. Herdt. (2000, Aug. 1). *Patent US6097629 A*.

[50] Non-volatile SRAM cell having split-gate transistors, by S. Liaw and H. Yang. (2006, May 30). *Patent US7054194 B2*.

[51] D. Atienza, P. Raghavan, J. Ayala, G. De Micheli, F. Catthoor, D. Verkest, M. Lopez-Vallejo, "Joint hardware–software leakage minimization approach for the register file of VLIW embedded architectures," *Integr. J. VLSI*, vol. 41, no. 1, pp. 38–48, Jan. 2008.

[52] H. Tabkhi and G. Schirner, "AFReP: Application guided Function-level Register file power-gating for embedded processors" *Proc. IEEE/ACM Int. Conf. on Computer-Aided Design*, pp. 302- 308, 2012.

[53] I. Chang, J. Kim, S. Park, and K. Roy, "A 32 kb 10T Sub-Threshold SRAM Array With Bit-Interleaving and Differential Read Scheme in 90 nm CMOS," *IEEE J. Solid-State Circuits,* vol. 44, No. 2, pp. 650-658, Feb. 2009.

[54] T. Miwa, J. Yamada, H. Koike, H. Toyoshima, K. Amanuma, S. Kobayashi, T. Tatsumi, Y. Maejima, H. Hada, and T. Kunio, "NV-SRAM: A Nonvolatile SRAM with Backup Ferroelectric Capacitors," *IEEE J. Solid-State Circuits,* vol. 36, No. 3, pp. 522-527, Mar. 2001.

[55] S. Chong, K. Akarvardar, R. Parsa, J. Yoon, R. Howe, S. Mitra, H. Wong, "Nanoelectromechanical (NEM) Relays Integrated with CMOS SRAM for Improved Stability and Low Leakage," *Proc. Int. Conf. on Computer-Aided Design,* pp. 478- 484, 2009.

[56] N. Sakimura, T. Sugibayashi, R. Nebashi, and N. Kasai, "Nonvolatile Magnetic Flip-Flop for Standby-Power-Free SoCs," *IEEE J. Solid-State Circuits,* vol. 44, No. 8, pp. 2244-2250, Aug. 2009.

[57] W. Yu, S. Xu, T. Huo, G. Suh, E. Kan, "Low power nonvolatile SRAM circuit with integrated low voltage nanocrystal PMOS Flash," *Proc. IEEE Int. SOC Conf.*, pp. 461-466, 2010.

[58] W. Wei, K. Namba, J. Han, and F. Lombardi, "Design of a Nonvolatile 7T1R SRAM Cell for Instant-on Operation," *IEEE Trans. Nanotechnology*, vol. 13, No. 5, pp. 905-916, Sep. 2009.

[59] H. Tabkhi and G Schirner, "Application-Guided Power Gating Reducing Register File Static Power,"to appear in *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2014.

[60] T. Monreal, V. Vinals, J. Gonzalez, A. Gonzalez, and M. Valero, "Late Allocation and Early Release of Physical Registers," *IEEE Trans. Very on Computers,* vol. 53, no. 10, pp. 1244-1259, Oct. 2004.

[61] S. Roy, N. Ranganathan, and S. Katkoori, "State-Retentive Power Gating of Register Files in Multicore Processors Featuring Multithreaded In-Order Cores," *IEEE Trans. on Computers,* vol. 60, no. 11, pp. 1547-1560, Nov. 2011.

[62] N. Zingirian and M. Maresca, "Selective Register Renaming: A Compiler-Driven Approach to Dynamic Register Renaming,"

High-Performance Computing and Networking. Lecture Notes in Computer Science, 2001.

[63] G. Magklis, M. Scott, G. Semeraro, D. Albonesi, and S. Dropsho, "Profile-based Dynamic Voltage and Frequency Scaling for A Multiple Clock Domain Microprocessor," *Proc. the 30th International Symposium on Computer Architecture.*, pp. 14-25 , 2003.

[64] T. Jones, M. O'Boyle, J. Abella, A. González, and O. Ergin, "Energy-efficient Register Caching with Compiler Assistance," *ACM Trans. on Architecture and Code Optimization*, vol. 6, no. 4, pp. 13-23, 2009.

[65] M. Gebhart, D. R. Johnson, D. Targan, S. W. Keckler, W. J. Dally, E. Lindholm, and K. Skadron, "A Hierarchical Thread Scheduler and Register File for Energy-Efficient Throughput Processors," A*CM Trans. on Architecture and Code Optimization*, vol. 30, no. 2, pp. 8-38, Apr. 2012.

Jinhui Wang received the B.E. degree in electrical engineering from Hebei University, Hebei, China, in 2004, and the Ph.D. degree in electrical engineering through a joint USA/China program between University of Rochester and Beijing University of Technology, in 2010. Dr. Wang is currently an Assistant Professor with the Department of Electrical and Computer Engineering at the North Dakota State University, Fargo, ND, USA. His research interests include low-power, high-performance, and variation-tolerant integrated circuit design, 3-D IC and EDA methodologies, and thermal issue solution in VLSI. He has more than 80 publications and 6 patents in the emerging semiconductor technologies.

Na Gong received the B.E. degree in electrical engineering, the M.E. degree in microelectronics from Hebei University, Hebei, China, and the Ph.D. degree in computer science and engineering from the State University of New York, Buffalo, in 2004, 2007, and 2013, respectively. Currently, Dr. Gong is an Assistant Professor with the Department of Electrical and Computer Engineering at the North Dakota State University, Fargo, ND, USA. Her research interests include device-circuit-architecture co-design for nano-scale VLSI circuit and system, power efficient and reliable electronics for mobile computing and high performance computing, and emerging memory technologies in computer systems.