

Design of Register File for Negative Bias Temperature Instability

Yuanyuan Ma^a, Bai Na^{a,*}, Wei Tan^b, and Gelan Yang^c

^a*School of Electronic and Information Engineering, University of Anhui of China, Hefei, 230601, China*

^b*School of Computer Science and Technology, Dongguan University of Technology, Dongguan, 523808, China*

^c*Department of Computers, Hunan City University, Yiyang, 413000, China*

Abstract

Negative bias temperature instability (NBTI) is becoming an important reliability problem in the semiconductor industry. As time goes on, the NBTI aging impact affects microprocessors' ability to perform correct calculations. The SRAM-based register file block is one of the largest logic units, and it is affected by process deviations. SRAM is the bottleneck of the whole process deviation tolerance. Based on theoretical analysis, the connection between the SRAM static noise margin value and the bitcell probability is discussed. Moreover, this paper adopts a dynamic shifter combined with periodic bitcell inversion design to reduce the NBTI aging impact and achieve a more robust register file. Simulation results exhibit that this design improves the bitcell probability by 3.7 times and reduces the uncertainty of SNM caused by NBTI stress to 46.79%.

Keywords: negative bias temperature instability; architecture register file; static noise margin; bias probability

(Submitted on September 27, 2019; Revised on October 30, 2019; Accepted on November 10, 2019)

© 2019 Totem Publisher, Inc. All rights reserved.

1. Introduction

Negative bias temperature instability (NBTI) has emerged as one of the key reliability issues in the semiconductor industry in recent years [1-3]. This crucial reliability concern for p-channel MOSFETs has become more and more important especially in logic and memory circuits. As one kind of memory circuit, SRAM-based register files should be robust enough to operate against process variation, high switching activity, large static leakage current, high temperature, and performance degradation over their usage lifetime [1-3]. This makes them particularly vulnerable to aging by NBTI degradation of their PMOS devices, as the threshold voltages of the PMOS transistors in the SRAM bitcell are affected. Changes in the threshold voltage can cause transistor characteristics to degrade and deviate from the design value, resulting in a large deviation from the expected behavior of the circuit [3]. It is possible to compensate for the design gain by designing a follower compensation circuit.

Assume that a "0" is stored in one SRAM bitcell. The gate voltage of one PMOS transistor is low, and this PMOS transistor is in the "ON mode", in which the interface trap slowly forms. However, when the input changes from "0" to "1", this causes a high voltage on the same PMOS transistor gate, and the PMOS is in the "OFF mode". Transistor aging relies on the relative length of the "ON mode" and "OFF mode". In other words, transistor aging is closely related to the gate voltage, threshold voltage, and work time. This paper proposes a technique to sense this aging degradation.

Many references simulate the actual situation of the typical computer architecture system and analyze register file work modes by running different series of programs on different microprocessors [7-9]. Thus, a relatively accurate model of NBTI influence in register files can be obtained by this method. Unfortunately, this method is too time-consuming and lacks transparency. To solve these problems, this paper proposes a design by distributing inherent bias and bitcell pattern variability evenly throughout the whole SRAM bitcell pools of register files. At the same times, this design can achieve robust aging characteristics in a better range of scenarios and reduce the uncertainty of NBTI aging influence.

* Corresponding author.

E-mail address: bnasic@126.com

The rest of this paper is organized as follows: firstly, the aging effect of NBTI on a 6T SRAM bitcell is analyzed, including its reliability. Secondly, SRAM work time degradation is analyzed. Thirdly, this paper presents a design method to overcome the limitation of periodic inversion in practical applications and proves the effectiveness of this method.

2. NBTI Aging in SRAM

2.1. NBTI Aging Evaluation on Traditional 6T SRAM Bitcell

Reference [10-13] proposed several models to explain the mechanism of NBTI impact. Based on these evaluation models, this paper focuses on the NBTI impact of a traditional 6T SRAM bitcell. To simplify the problem, according to references [11-13], this paper also assumes that NBTI has an impact on SRAM bitcell performance by changing the transistor threshold voltage.

If PMOS transistors are under different and variable stress, the AC RD model is

$$\Delta V_{th}(t) \cong K_{AC} \times t^n \cong a(s, f) \times K_{DC} \times t^n \quad (1)$$

$$K_{DC} = A \times T_{ox} \times \sqrt{C_{ox}(V_{GS} - V_{th})} \times [1 - \frac{V_{DS}}{a(V_{GS} - V_{th})}] \times \exp(E_{ox} / E_0) \times \exp(-E_a / KT) \quad (2)$$

Where T_{ox} is the gate oxide thickness, E_{ox} is the gate electric field, k is the Boltzmann constant, and T is the temperature.

$$V_T = V_{T0} + \gamma(\sqrt{|-2\phi F + V_{SB}|} - \sqrt{|2\phi F|}) \quad (3)$$

$$\Delta V_{tp} = -\frac{q}{C_{ox}} N_{IT} \quad (4)$$

With reference to the RD model, the rate of appearance of interface traps (NIT) initially relies on the rate of bond decomposition; it is governed by the rate constant (K_p) and the local self-annealing phase, which is controlled by the rate constant (K_r). Cox is the oxide capacitance per cell area. In order to show that the generation of traps will lead to a decrease in mobility, which will lead to a decrease in threshold voltage, Equation (4) has been modified to reflect m as a measure of the additional V_{tp} degradation caused by mobility degradation.

Thus,

$$\Delta V_{tp} = -\frac{(m+1)q}{C_{ox}} N_{IT} \quad (5)$$

To exhibit SRAM bitcell's performance degradation caused by NBTI aging, 3,000 HSPICE simulations of delay time, write delay time, and static noise margin (SNM) are analyzed as shown in Table 1.

Table 1. Performance degradation caused by NBTI impact

	65nm SRAM bitcell				180nm SRAM bitcell			
	Nominal		NBTI impact		Nominal		NBTI impact	
	Mean	Deviation	Mean	Deviation	Mean	Deviation	Mean	Deviation
Read delay	183.61ps	0.27ps	185.58ps	0.32ps	195.6ps	0.0134ps	195.3ps	0.0115ps
Write delay	34.9ps	0.211ps	33.31ps	0.54ps	45.9ps	0.0012ps	44.62ps	0.0021ps
SNM	288mV		185.4mV		518mV		437mV	

Compared to the read delay mean of the SRAM bitcell under nominal situation, this parameter has a negligible effect under the NBTI aging situation: 183.61ps and 185.58ps in the 65nm process, and 195.6ps and 195.3ps in the 180nm process. At the same time, the mean of write delay has some improvements when the bitcell is under the NBTI aging situation in both the 65nm process and 180nm process, compared to the same bitcell under the nominal situation. The simulation results in Table 1 show that the mean of the write delay is decreased by 4.56% in the 65nm process and 2.79% in the 180nm process. The static noise margin (SNM) is the value of the minimum DC noise margin (including process variation, high

temperature, and performance degradation), which can be measured by the SRAM inverter transfer characteristics (butterfly curve). As shown in Table 1, SNM is 288mV in the 65nm nominal situation, which deteriorates to 185.4mV under the NBTI aging situation, while in 180nm, this parameter deteriorates to 84.36% under the NBTI aging situation. Meanwhile, the most critical influence of NBTI impact is SNM. To widen SRAM SNM and enhance the bitcell reading stability, this paper focus on the characteristics and performance degradation of SNM in SRAM-based register files, because it determines the reliable operation of embedded systems.

2.2. Static Noise Margin Degradation due to NBTI

The traditional 6T SRAM bitcell consists of two cross couple inverters (P1-N1, P2-N2) and two access NMOS transistors (N3, N4). As shown in Figure 1, "1" is stored in QL, "0" is stored in QR, and there exist three leakage paths. During the read mode, the WL line is connected to "1", and the QL voltage is determined by P1 and N1. If these transistors are affected by the NBTI impact, such as if the threshold voltages are increased, then the working current may be reduced significantly, making the access time slower. Due to the NBTI impact, the SNM is reduced, which is shown in Figure 2 and Table 2.

$$\begin{aligned} V_{tp1} &\rightarrow V_{tp1} + \Delta V_{tp1} \\ V_{tp2} &\rightarrow V_{tp2} + \Delta V_{tp2} \end{aligned} \quad (6)$$

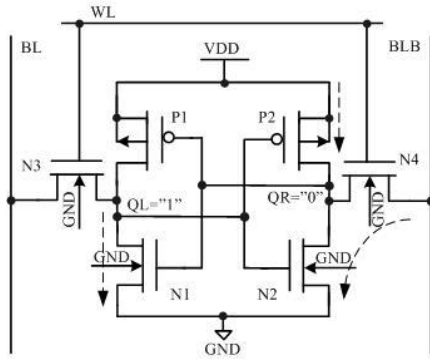


Figure 1. Traditional 6T SRAM bitcell

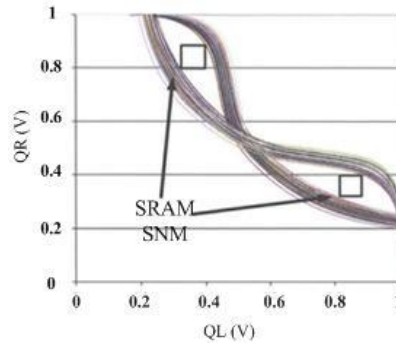


Figure 2. Static noise margin degradation due to NBTI impact

Table 2. SRAM bitcell SNM degradation in 65nm and 180nm caused by NBTI impact

Times (s)	65nm SRAM bitcell			180nm SRAM bitcell		
	Vtp (mV)		SNM	Vtp (mV)		SNM
	Mean	Deviation		Mean	Deviation	
0	-343.0	62.1	288.0	-620.0	32.4	518.0
1	-343.9	62.1	254.1	-620.8	40.3	518.0
10 ¹	-344.7	63.1	243	-621.1	40.5	510.1
10 ²	-346.0	63.3	237	-622.4	43.3	475.6
10 ³	-348.3	63.9	229	-624.0	43.7	466.1
10 ⁴	-352.5	65.2	212	-627.0	45.5	458.5
10 ⁵	-360.1	66.1	198	-633.4	47.5	443.7
10 ⁶	-373.2	66.4	188.1	-643.0	47.5	436.0
10 ⁷	-396.5	66.8	184.4	-659.1	47.9	436.5
10 ⁸	-436.7	67	185.4	-689.6	48.0	437.0

Considering that a bitcell's static noise margin equals the minimum side of the square nested between its two inverter transfer characteristics curves, its value degrades due to the NBTI aging impact. The 10⁸ SNM value is reduced to 64.37% in the 65nm process and 84.36% in the 180nm process. This may include a large deviation from the expected behavior of the circuit, especially in the register file. As a result, embedded systems may not work as expected in theory over time, and the aging impact may become more severe over time, causing the entire system to fail. Changes in the threshold voltage can cause SRAM transistor characteristics to degrade and deviate from the design value, resulting in a large deviation from the expected behavior of the circuit. It is possible to compensate for its design gain by designing a SNM compensation circuit.

2.3. Periodic Bitcell Inversion Design

This paper use bitcell probability (BP) to define the probability of a bitcell storing the value "0". Considering the NBTI aging impact, BP represents a time period when one of the PMOS transistors is in the "ON mode" and the other one is in the

"OFF mode". The decline in SNM is strongly dependent on the BP of cells [7]. To describe SRAM SNM degradation characteristics by the NBTI aging impact, this paper studies the influence of PMOS transistors' "ON mode" on the actual system in this section. Therefore, this paper runs CPU2006 standard marks in SPARC V9 architecture to generate register file values. Figure 3 shows the bitcell probability scatter of four different register groups (global, input, output, and local) in SPEC 2006 benchmarks [10]. After running 100 million instructions of this test bench, the average BP value of all eight 64-bits register file SRAM bitcells can be obtained. As shown in Figure 3, the minimum probability of the bitcell storing "0" is 0.08 in the local register file, while the maximum probability is 0.56 in the input register file, which is almost seven times the minimum BP in the local register file. In the individual group, the distribution of bitcell probability of SRAM bitcell stores "0" unstably. The probability between the maximum probability and the minimum probability increases by 5.25 times in the local register file, while the ratio is 2.3 in the global register file. In view of the huge difference in BP values in different applications, it is important to analyze the method that stabilizes the BP value and enhances SRAM SNM stability.

A processor is needed to run various applications. Therefore, the register file of a processor should have ability to work under different work modes from various program instructions. To enhance the SRAM SNM, reference [3] proposed a periodic bitcell inversion design to keep the probability of a SRAM bitcell storing value "0" at about 0.5 to reduce SRAM SNM aging degradation, with an average BP value of 0.22 for all register files. This design reverses the signal that bitcell stores after a regular interval time to ensure that the BP value of the register file is 0.5. During the inversion work mode, the input data is stored in the inversion mode. At the end of operation, the output data is returned to its inversion. Although this method is simple, it is truly effective in register file design, as these circuit does not change too much.

2.4. Periodic Bitcell Inversion Design Limitation

Unfortunately, a register file has various kind of work modes that exhibit great differences when considering the relationships of various programs. Periodic bitcell inversion design works well when BP values of a register file remain more or less stable, and it is expected to work in a single-threaded environment. However, multi-threaded environments are more popular in modern embedded system applications. In most applications, the random sequence signal that is stored in a register file may be reused by other program instructions. In order to simulate the actual situation where different programs execute respectively, this paper divides the 15 different standard test programs in the SPEC 2006 into ten different time ranges. 150 different instructions and 2,000 random sequences of the program have been selected in this way. This paper collects BP values of every register file bitcell to exhibit a real application, as shown in Figure 4. The degradation of the entire register file will be dictated by the worst case BP value among all of the register files bitcells in the processor. Then, BP and 1-BP values of a register file are close (completely negating their inherent bias) with the periodic bitcell inversion method. For work in a single-threaded environment, the register file can obtain an overall BP of 0.5. Since periodic bitcell inversion design is unable to adapt to the fluctuating program characteristics, it cannot achieve the desired average BP for several execution sequences. Considering a multi-threaded application environment, periodically reversing bitcell values does not bring the resulting BP value close to the optimal value. Because the periodic bitcell inversion of the SRAM bitcell cannot adapt to the fluctuation characteristics of the program, it cannot achieve the expected BP of multiple execution sequences.

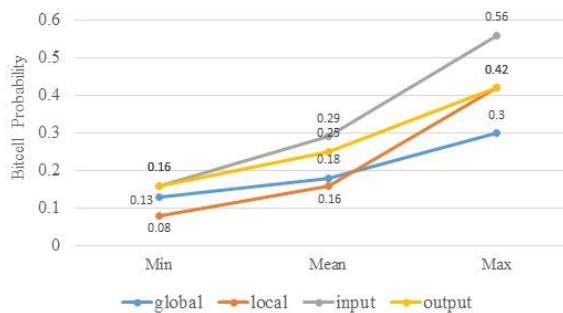


Figure 3. Distribution of bitcell probability value in SPEC 2006 benchmarks

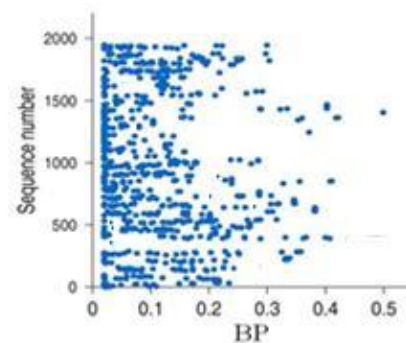


Figure 4. BP with periodic inversion on application sequences

3. Traditional Architecture Design

The architecture presented here is designed to balance the NBTI "ON mode" across register files, even in multi-threaded environments. Usually, there are several register files in a regular processor. Figure 5 shows the physical structure of the register file SRAM block. A basic register file consists of a memory cell array, I/O Buffer, address decoder (row and

column decoder), multiplexer (MUX), sensitive amplifier (SA), and operation control circuit. A 128 row- and 128 column-16 Kbit SRAM is the basic block for a processor register file. To reduce the propagation delay of word lines, the block is composed of two symmetrical arrays. 128 SRAM bitcells are connected to two complementary bit lines for the requirement of worst case performance.

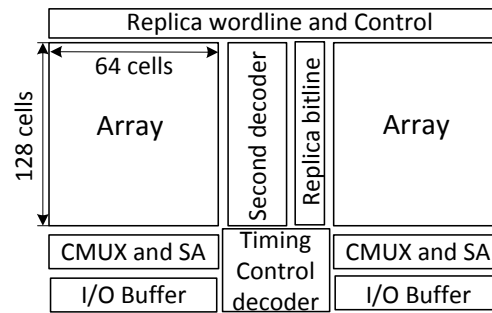


Figure 5. Physical structure of a register file

The address decoder is an important module among the peripheral circuits of the SRAM block. To obtain the exact signal stored in the SRAM bitcell, the system address is usually divided into two parts: a high-address decoder (row decoder) and low-address decoder (column decoder). In traditional processor design, the same physical SRAM unit is used to store the values of specific registers throughout the design's life. The static logic mapping technique between the architecture register file and the physical SRAM bitcell is a very important technique in processor memory arrangement. With this technique, a processor can extend its storage space with few management penalties. However, this technique also leads to a large difference in BP values across the entire register files, as shown in Figure 6. This paper proposes to change the decoding scheme of register files and use the dynamic shifter combined with periodic bitcell inversion design to eliminate static links between chip registers and SRAM bitcells.

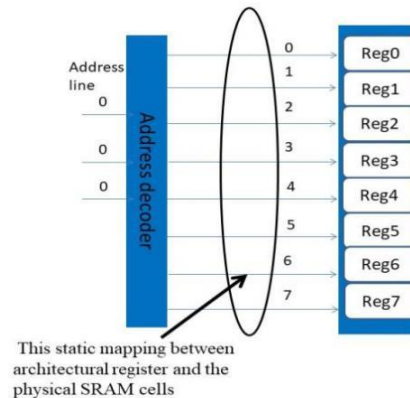


Figure 6. Static mapping of register file

4. Proposed Architecture Design

4.1. Dynamic Shifter Design

Figure 7 takes a simple 8-bit register file as an example. Eight identical registers are also listed. This paper takes a 64-bit register in SPARC V9 architecture as a testing platform. When the register file size is extended to 16Kbits, it has the same working scheme. As shown in Figure 7, a bucket shifter is included between the address decoder and the memory unit. The bucket shifter rotates the address selector line (word line in Figures 7 and 8) after a period of time. Register 0 maps to line 1 after the interval T . Then, the selection line moves to the next one, after another interval T . This means that register 0 can use different memory cell lines during a long period. Therefore, the selected SRAM bitcell under the NBTI stress mode is also reduced by this method. The bitcell probability of a certain SRAM bitcell can be balanced, and the register file SNM can be enhanced in this way. It must be noticed that, after such an operation, the shift count is incremented from "0" to "1". The bucket shifter stores this value as a flag signal. Figure 8 shows the changes in the moving count after a certain interval of time t_0 . Repeatedly moving such operations can result in a complete rotation, which ensures that all eight rows of memory units are stored in register 0. Figures 7 and 8 only take word line rotation as an example. The bit line rotation performs the same operation. The shifter in the write port moves the input data and stores it in a SRAM bitcell (which is

called the content addressable memory bitcell and includes a 6T SRAM bitcell and XOR unit). When the shift count is 1, the bitcell column 0 is saved for the first column. If the count adds, the bitcell column 0 will move to the left. After several times, the number is added to N (the width of the register), and bitcell column 0 uses each column of bitcell to save its signal.

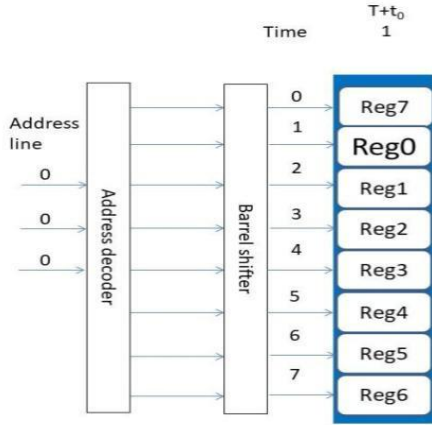


Figure 7. Dynamic shifter design

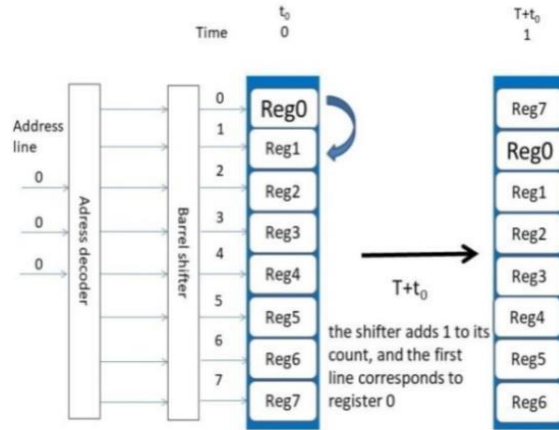


Figure 8. Barrel shifter rotates the select line by shifting count

4.2. Dynamic Shifter Design Combined with Periodic Bitcell Inversion

The row and column NBTI aging stress of the SRAM array are distributed simultaneously in the register file by dynamic shifter design. Figure 8 shows the sequence of operations. The signal stored in the register file bitcell tends to be stable. Thus, some BP values tend to be zero. Although dynamic shifter design can release certain bitcells' NBTI stress, it cannot make bitcell probability values tends to be 0.5. Therefore, to design a NBTI robust microstructure, this design combines dynamic shifter with periodic bitcell inversion, as shown in Figure 9.

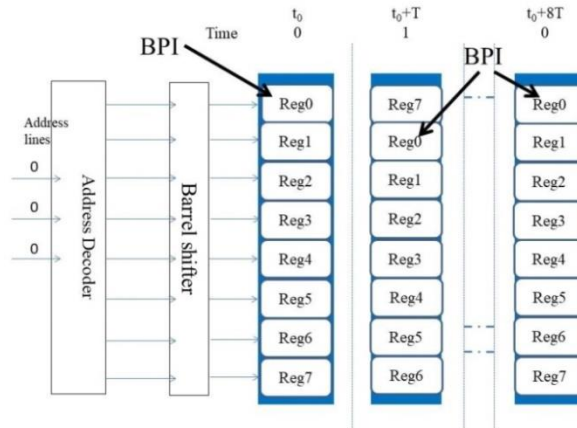


Figure 9. Dynamic shifter design combine with periodic bitcell inversion

4.3. Implementation

Generally, there are three major segments in the operation of a register file: (1) decoder, (2) read/write, and (3) input/output. For decoder, Figure 10(a) shows the structure of a two-stage distributed decoder used in this design. In the first-level decode stage, four decoders are used to decode the nine addresses into four sets of output lines. The column decoder decodes address a_{1-0} into four column select lines for column multiplexer (CMUX) by two to four decoders. Decoders RD1 and RD2 decode address bits a_{4-2} and a_{7-5} into eight row select lines by two 3-8 row decoders respectively, and the RD3 decodes the remaining address into two pulse select lines by one to two decoders. Compared to the traditional decode scheme (a seven address bit decoder that is composed of one 4-bit decoder and one 3-bit decoder), the row decoder structure decreases the delay of address decoding and releases the NBTI aging impact incurred by body effect because the largest decoder delay and threshold voltage changing is only in the 3-bit decoder. The second-level decoder contains 128 blocks (WLL_0 - WLL_{127}). A detailed circuit of a block is shown in Figure 10(b). A 6-64 NOR type decoder decodes six address bits a_{7-2} into 64 row select lines. To reduce system leakage, the virtual ground technique is used in this paper. The WLL enable signal (WLL_{CLK})

controls the address bit signal a_8 . It is the input of an inverter whose virtual ground is controlled by 64 row select lines. As a result, only one of the 128 word lines (WLL) is enabled in each operation.

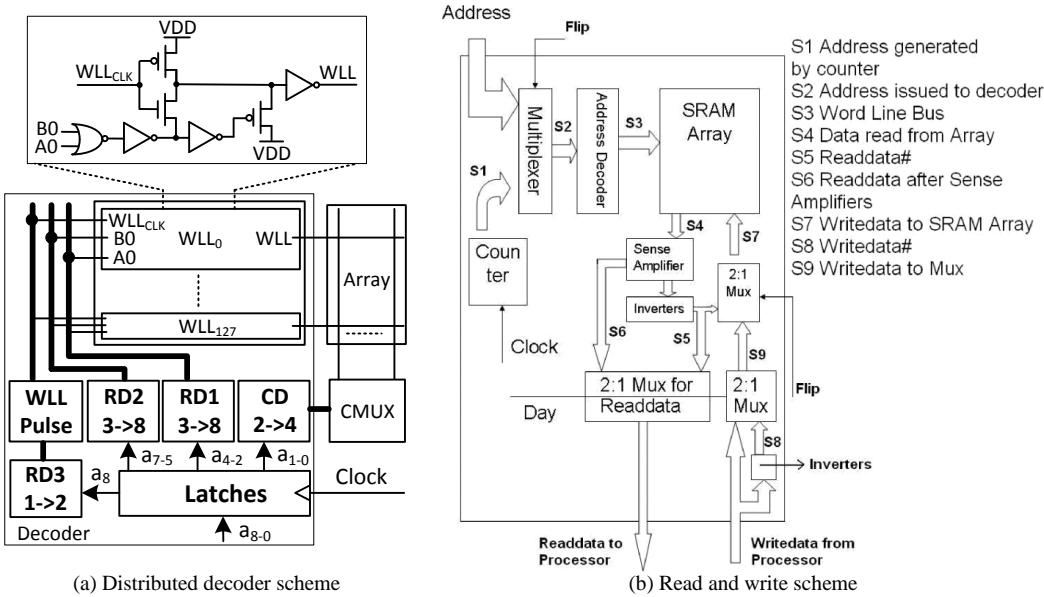


Figure 10. Circuit implementation

Take one SRAM column as example. During the read mode, two complementary signals are read out from the sense amplifier (latch type). A 2:1 multiplexer selects one of them by the control signal and sends it to the output. Similarly, during the write mode, two complementary signals are generated almost simultaneously, with only one inverter delay. The proposed design also uses a 2:1 multiplexer to select one of two write signals and send it into the SRAM array according to the control signal value. The control signal generated from counter helps the system operate exactly. Instruction execution relies on the association between the data and their address (the correct register file SRAM bitcells). Therefore, it is necessary to ensure that all of the word lines and bit lines are pre-charged when executing an instruction. As the mapping relationship between physical SRAM units and register bitcell locations are shifted, an initial state should be defined at the beginning. The values in the register files according to the decoding changes or bitcell position information should be detected every clock cycle.

5. Testing and Results

Compared to the regular design, most BP values are around 0.13 (as shown in Figure 4). Most BP values of the dynamic shifter combined with periodic bitcell inversion design are around 0.48, very close to the optimal value when considering application interleaving (as shown in Figure 11). This is because the proposed dynamic shifter combined with periodic bitcell inversion design writes all of the register file bitcells at regular intervals, and the NBTI's influence on the register file blocks is almost "0" because there is no opportunity to create interface traps; therefore, no recovery measures are required. Table 3 gives the summary of these results. Compared to the referenced design, the proposed design can achieve more robust aging characteristics and reduce the uncertainty of NBTI aging influence across a wider range of applications. The median value of bitcell probability is 3.7 times that of the referenced design. The SNM degradation is reduced to 46.79%.

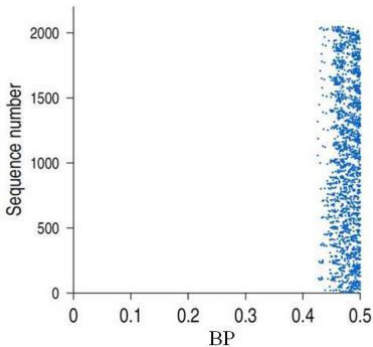


Figure 11. Scatter plot for sequences of dynamic shifter design combine with periodic bitcell inversion

Table 3. Comparative BP and SNM degradation

Design	Median (bitcell probability)	SNM degradation
Referenced [3]	0.13	31.2%
Dynamic shifter design + PBI	0.48	14.6%

Table 4 shows the leakage distribution of the proposed design in the 65nm process. The leakage current of WLL drivers with virtual ground control can be decreased by about 34.98% and 22.64% compared with the traditional full static NAND gate WLL drivers in different processes. At the same time, the total leakage power in the active mode is decreased by about 14.6% and 4.47%. The area of the distributed decoder is almost equal to that of the conventional decoder.

Table 4. Leakage current distribution of 16kb register file in 65nm process

Process corner	TT, 1.2 V, 27°C	FF, 1.32 V, 125°C
Leakage current of SRAM array (uA)	1.72	542
Leakage current of peripheral circuit (uA)	0.71	114
Leakage current of WLL modules (uA)	1.015	129.4
Leakage current of WLL modules with virtual ground control (uA)	0.66	100.1
WLL leakage current decrease	34.98%	22.64%
Total leakage current decrease	14.60%	4.47%

6. Conclusions

The reliability degradation in register file is analysed in this paper. Based on the theoretical modeling, the connection between the SRAM static noise margin value and the bitcell probability is discussed. The shortcoming of traditional periodic bit inversion design is also pointed out. To reduce the NBTI aging impact in a multi-threaded environment, a dynamic shifter is added to cater to the register file content of cyclical change. Combined with periodic bit inversion design, it can achieve more robust aging characteristics and reduce the uncertainty of NBTI aging influence across a wider range of applications. The simulation results demonstrate that this design improves the bitcell probability by 3.7 times and reduces the uncertainty of SNM caused by NBTI stress to 46.79%.

Acknowledgements

This work was supported by the National Science Foundation of Hunan Province, China (No. 2018JJ2023), the National Natural Science Foundation of China (No. 61702098), the Science and Technology Planning Project of Guangdong Province, China (No. 2014A 010103039, 2015A010103022, and 2014B090901002), and the Dongguan Science and Technology Plan Project, Guangdong Province, China (No. 2014509102211).

References

1. J. Abella, X. Vera, and A. Gonza'lez, "Penelope: The NBTI-Aware Processor," in *Proceedings of 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*, pp. 85-96, 2007
2. K. Kang, S. Gangwal, S. P. Park, and K. Roy, "NBTI Induced Performance Degradation in Logic and Memory Circuits: How Effectively Can We Approach a Reliability Solution?" in *Proceedings of 2008 Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 726-731, 2008
3. S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "Impact of NBTI on SRAM Read Stability and Design for Reliability," in *Proceedings of the 2006 7th International Symposium on Quality Electronic Design (ISQED)*, pp. 210-218, 2006
4. Y. Lin, Y. Li, X. Yin, and Z. Dou, "Multisensor Fault Diagnosis Modeling based on the Evidence Theory," *IEEE Transactions on Reliability*, Vol. 67, No. 2, pp. 513-521, 2018
5. Y. Lin, C. Wang, J. Wang, and Z. Dou, "A Novel Dynamic Spectrum Access Framework based on Reinforcement Learning for Cognitive Radio Sensor Networks," *Sensors*, Vol. 16, No. 10, pp. 1675, 2016
6. Y. Lin, X. Zhu, Z. Zheng, Z. Dou, and R. Zhou, "The Individual Identification Method of Wireless Device based on Dimensionality Reduction and Machine Learning," *Journal of Supercomputing*, Vol. 75, No. 6, pp. 3010-3027, 2019
7. K. Kang, H. Kufluoglu, K. Roy, and M. A. Alam, "Impact of Negative-Bias Temperature Instability in Nanoscale SRAM Array: Modeling and Analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 26, No. 10, pp. 1770-1781, 2007
8. G. Yang and J. Nightingale, "Special Issue on Mobile Multimedia Big Data Embedded Systems," *J. Adv. Comput. Intell. Intell. Inform.*, Vol. 22, No. 7, pp. 1071, 2018
9. W. Zhao and Y. Cao, "New Generation of Predictive Technology Model for Sub-45 nm Early Design Exploration," *IEEE Transactions on Electron Devices*, Vol. 53, No. 11, pp. 2816-2823, 2006
10. M. Marchitto, S. Benedetto, T. Baccino, and J. J. Canas, "Air Traffic Control: Ocular Metrics Reflect Cognitive Complexity," *International Journal of Industrial Ergonomics*, Vol. 54, pp. 120-130, 2016

11. C. H. Liu, M. T. Lee, C. Y. Lin, J. Chen, K. Schruefer, J. Brighten, et al., "Mechanism and Process Dependence of Negative Bias Temperature Instability (NBTI) for pMOSFETs with Ultrathin Gate Dielectrics," in *Proceedings of IEEE International Electron Devices Meeting*, pp. 39.2.1-39.2.4, 2001
12. B. Zhu, J. S. Suehle, Y. Chen, and J. B. Bernstein, "Negative Bias Temperature Instability of Deep Sub-Micron p-MOSFETs under Pulsed Bias Stress," *IEEE International Integrated Reliability Workshop Final Report*, pp. 125-129, 2002
13. M. Marchitto, S. Benedetto, T. Baccino, and J. J. Canas, "Air Traffic Control: Ocular Metrics Reflect Cognitive Complexity," *International Journal of Industrial Ergonomics*, Vol. 54, No. 4, pp. 120-130, 2016