Bottom-up Memory Design Techniques for Energy-Efficient and Resilient Computing

Pi-Feng Chiu

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2018-156
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2018/EECS-2018-156.html

December 1, 2018

Bottom-up Memory Design Techniques for Energy-Efficient and Resilient Computing

by

Pi Feng Chiu

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Borivoje Nikolić, Chair
Professor Krste Asanović
Professor Lane Martin

Spring 2018

Bottom-up Memory Design Techniques for Energy-Efficient and Resilient Computing

Abstract

Bottom-up Memory Design Techniques for Energy-Efficient and Resilient Computing

by

Pi Feng Chiu

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Borivoje Nikolić, Chair

Energy-efficient computing is critical for a wide range of electronic devices, from personal mobile devices that have limited battery capacity to cloud servers that have costly electricity bills. The increasing number of IoT devices has resulted in a growing demand for energy-efficient computing for battery-powered sensor nodes. The energy spent on memory access is a major contributor of total energy consumption, especially for new, data-intensive applications. Improving memory energy efficiency helps build energy-efficent computing systems. One effective way to achieve better energy efficiency is to lower the operating voltage by using the dynamic voltage and frequency scaling (DVFS). However, further reductions in voltage are limited by SRAM-based caches. The aggressive implementation of SRAM bit cells to achieve high density causes larger variation than in logic cells. In order to operate the processor at the optimal energy-efficient point, the SRAM needs to reliably work at a lower voltage.

The sense amplifier of the memory circuit detects the small signal from the bit cell to enable high-speed and low-power read operation. The mismatch between the transistors due to process variation causes an offset voltage in the sense amplifier, which could lead to incorrect results when the sensing signal is smaller than the offset. The double-tail sense amplifier (DTSA) is proposed as a drop-in replacement for a conventional SRAM sense amplifier to enable robust sensing at low voltages. The dual-stage design reduces the offset voltage with a pre-amplification stage. The self-timed regenerative stage simplifies the timing logic and reduces the area. By simply replacing the conventional sense amplifier with DTSA, SRAM can operate with a 50mV Vmin reduction at faster timing.

Memory resiliency can be achieved through architecture-level assist techniques, which enable low-voltage operation by avoiding failing cells. The line disable (LD) scheme deactivates faulty cache lines in a set-associative cache to prevent bitcell with errors from being accessed. The Vmin reduction of LD is limited by the allowable capacity loss with minimum performance degradation. The line recycling (LR) technique is proposed to reuse two disabled faulty cache lines to repair a third line. By recycling the faulty lines, 1/3 of the capacity

loss due to LD can be avoided for the same Vmin, or one-third as many faulty cache lines can be ignored.

Emerging nonvolatile memory (NVM) technologies, such as STT-MRAM, RRAM, and PCM, offer a tremendous opportunity to improve energy efficiency in the memory system while continuously scaling. The new technologies are faster and more durable than NAND flash, therefore, can be placed closer to the processing unit to save the energy by powering off. However, reliable access to NVM cells faces several challenges, such as shifting of cell resistance distributions, small read margins, and wear-out.

The proposed differential 2R crosspoint resistive random access memory (RRAM) with array segmentation and sense-before-write techniques significantly improves read margin and removes data-dependent IR drop by composing one bit with two complementary cells. The proposed array architecture ensures large read margin (>100mV) even with a small resistance ratio (R_H/R_L=2).

In summary, this dissertation introduces techniques at different levels of memory design (device, circuit, and micro-architecture) that can work in concert to build a resilient and energy-efficient memory system. The proposed techniques are demonstrated on several chips fabricated in a 28nm CMOS process.

To my husband, Wei-Han.

Contents

Contents **ii**

List of Figures **iv**

List of Tables **vii**

1 Introduction **1**
1.1 Energy-efficient computing 1
1.2 Semiconductor memory technologies 2
1.2.1 Memory 2
1.2.2 Storage 4
1.2.3 Storage class memory (SCM) 6
1.3 Limitation on energy-efficient memory 6
1.4 Techniques at different levels 7
1.5 Scope of the dissertation 8
1.6 Thesis outline 9

2 Sense Amplifiers for Low-Voltage SRAM **11**
2.1 Basic SRAM operations 11
2.2 Challenges for low-voltage sensing 13
2.3 Conventional Sense Amplifiers and Pior Work 15
2.4 Double-Tail Sense Amplifier 16
2.5 Silicon Measurement Results 20
2.6 Comparison 23
2.7 Conclusion 25

3 Cache Resiliency with Architecture-level Assist Techniques **26**
3.1 Introduction 26
3.1.1 ECC-based techniques 28
3.1.2 Redundancy-based techniques 29
3.1.3 Disable-based techniques 30
3.2 Line Recycling (LR) 31

3.2.1 Implementation 33
3.3 Bit Bypass with SRAM implementation (BB-S) 35
3.3.1 Implementation 37
3.4 Error Model 38
3.5 Resilient Out-of-Order processor 40
3.5.1 Berkeley Out-of-Order Machine (BOOM) 41
3.5.2 Implementation of the Resiliency Techniques 43
3.6 Conclusion 47

4 Implementation of Resilient Out-of-Order Processor 48
4.1 Overview 48
4.2 Preparing the design 49
4.3 Synthesis 50
4.4 Placement and routing 51
4.5 Sign-off 52
4.6 Case Study: Implementation of the Register File 53
4.7 Measurement 54
4.7.1 Measurement setup 56
4.7.2 Measurement Results 57
4.8 Conclusion 60

5 Emerging Nonvolatile Memory 61
5.1 Introduction 61
5.2 Array architecture 62
5.2.1 RRAM bitcell structure 62
5.2.2 Switching behavior 64
5.2.3 Analysis of leakage current 65
5.3 Differential 2R Crosspoint Array 67
5.3.1 Circuit Implementation 68
5.3.2 Simulation Results 71
5.4 Differential RRAM in memory hierarchy 73
5.5 Conclusion 74

6 Conclusion 75
6.1 Summary of Contributions 75
6.2 Future Work 76

Bibliography 78

List of Figures

1.1 Trade-offs of dynamic energy consumption and leakage energy consumption and the definition of the optimal energy enfficient point. 2
1.2 The schematic of (a) a 6T SRAM bit cell, (b) an 8T SRAM bit cell, and (c) a 1T-1C DRAM bit cell. 4
1.3 Different levels of memory design, including semiconductor devices, circuit, and architecture. 7

2.1 Block diagram of an SRAM macro with an array, decoder, control logic, column multiplexer, and sense amplifiers. The bitcell is composed of six transistors. . . . 12
2.2 Waveforms of SRAM read/write operation. 13
2.3 A longer WL pulse and WL-SAE delay result in a larger ΔV for a more reliable sensing but degrade the read speed and power consumption. 14
2.4 Histogram of bitline swing (ΔV) at 0.9V and 0.6V. 14
2.5 The schematic of (a) a conventional SA and (b) a StrongArm SA. 16
2.6 Circuit diagrams of the double-tail sense amplifier, consisting of the preamplification stage and the regeneration stage. 17
2.7 Operation waveforms of the DTSA at 0.6V and ΔV = 50mV. 18
2.8 Simulated error rate with various ΔV for the conventional SA and the DTSA. . 18
2.9 Offset voltages at different process corners (VDD=0.45V). 19
2.10 The offset voltage of the conventional SA and the DTSA at different supply voltages. 20
2.11 Trade-off between error rate and SAE-Q delay for various sizings of $\mathrm{M_{TAIL}}$. . . . 21
2.13 Die photo and measurement setup. 21
2.12 Dimensions of the conventional SA and the DTSA (normalized to width). 22
2.14 Measured error rate for various tunable timing settings at 0.44V (averaged across 6 chips). 23
2.15 Measured error rate at different VDD and tunable timing settings (averaged across 6 chips). 24
2.16 Shmoo plots of the SRAM with the conventional SA and the DTSA. 24

3.1 Circuit-level techniques lower Vmin by improving bitcells while architecture-level techniques lower Vmin by tolerating bitcell failures. [1] 28

3.2 Dynamic column redundancy exploits cache architectures to repair a single bit per set in cache data arrays. 30
3.3 Way replacement mechanism for the disabled way. 31
3.4 The measured distribution of the number of errors in (a) a cache line and (b) a set. 32
3.5 The concept of line recycling: reusing two disabled cache lines to repair a third one via majority vote. 33
3.6 Block diagram of the implementation of line recycling. 34
3.7 The number of PAT entries required for different BER. 35
3.8 Block diagram of Bit Bypass (BB). 35
3.9 Block diagram of Bit Bypass with SRAM implementation (BB-S). 36
3.10 Four cases when the error bits are at different locations. (R: Repair bit, V: Valid bit, COL: Error column location). 37
3.11 Block diagram of the implementation of BB-S. 38
3.12 Description of the notation used in the error model framework that translates bitcell failure rate to system yield. 39
3.13 Block diagram and layout of BROOM. 41
3.14 The nine-stage pipeline BOOM. 42
3.15 Resiliency schemes implemented in BROOM. 43
3.16 The BIST control state machine allows a wide variety of programmable March tests. [2] 44
3.17 Data path of the BIST in the memory system. 45
3.18 Flow chart of reprogramming the resiliency techniques and the pseudo codes of the failure analysis. 46
3.19 Block diagram of the data path in the L2 cache. Blue blocks highlight the ECC overheads. Red blocks highlight the DCR overheads. Purple blocks highlight the LR overheads. 47

4.1 The flow of chip implementation from Chisel to fabrication. Verification and physical design are both essential to a working chip. 50
4.2 The register file implemented with flip-flops is a bottleneck for routing, resulting in shorts. 53
4.3 The schematic of a full-custom cell for the register file and the implementation with standard cells. 55
4.4 The array of customized unit cell with hierarchical read wires and the results of the guided physical design. 55
4.5 The wire-bonded BROOM chip. 56
4.6 (a) A photo of the measurement setup. (b) Interface bridges between the chip and FPGA. 57
4.7 Operating voltage and frequency range with/without LR. 58
4.8 (a) Measured SRAM bitcell failure rate versus voltage and (b) Vmin reduction of various resiliency techniques. 58
4.9 Access latencies of the cache hierarchy measured with ccbench. 59

4.10 The effect of increasing (a) miss rate and (b) CPI with 5% and 10% capacity loss on different benchmarks. 59

5.1 The pyramid of memory hierarchy for a fast, massive, and nonvolatile memory. . 62
5.2 1T1R RRAM array architecture and the cross-sectional view of the cell. 63
5.3 The crosspoint RRAM array architecture and two stacked layer. 63
5.4 Time required for setting the resistance from HRS to LRS under different V_{SET} and R_L. 64
5.5 Write energy for setting the resistance from HRS to LRS under different V_{SET} and R_L. 65
5.6 The bias scheme for the crosspoint array. (a) V/2 bias scheme. (b) Floating Wordline Half-voltage Bitline (FWHB) scheme. 66
5.7 Worst case of reading HRS in current sensing scheme. (m: BL length, n: WL length). 67
5.8 The architecture of a differential 2R crosspoint array and the table of operating conditions in form/write/read mode. 68
5.9 Cross-sectional view of the differential 2R array with array segmentation. 69
5.10 Block diagram of a 64KB crosspoint RRAM circuit. 70
5.11 Waveforms of read and write operation in the differential 2R crosspoint array. . 71
5.12 Read margin of the differential 2R crosspoint array with different R ratios. . . . 72

List of Tables

2.1 Comparisons between various published sense amplifiers. 25

3.1 Summary of the circuit-level assist techniques for SRAMs. 27
3.2 Summary of the SRAM macros in the L2 system. 34
3.3 Summary of the testable SRAMs in the chip, with corresponding size and BIST macro address. 45

5.1 Parameters in D-2R circuit simulation. 72
5.2 Comparisons between various memory technologies for cache usage. 73

Acknowledgments

I am in debt of a lot of people for their help and support throughout my graduate career. First, my sincere gratitude goes to my advisor, Borivoje Nikolić, for his support and guidance. He has always been looking out for my best interest and offering advice for achieving my goals. Although Professor Krste Asanović was not officially my co-advisor, the insightful discussions and constant encouragement inspired me. Brian Zimmer was not just a co-worker but my mentor on the projects. He is knowledgeable, humble and passionate on researches. I learned a lot from him. The days of tape-outs were tough but could have been worse without my comrades, Ben Keller and Stevo Bailey. I would like to thank all the people who dedicated in building the infrastructure of RISC-V processors, especially Yunsup Lee, Colin Schmidt and Howie Mao, who answered all my questions patiently. The collaboration with Chris Celio on BROOM was truly enjoyable. He was more than willing to share his expertise in computer architecture that helped expand my reasearch area.

I would like to thank all of the members and alumni of the COMIC group, including Rachel Hochman, Amy Whitcombe, Angie Wang, Keertana Settaluri, Paul Rigge, John Wright, Alon Amid, Katerina Papadopoulou, Martin Cochet, Miki Blagojevic, Dajana Danilovic, Natalija Javanovic, Sharon Xiao, Sameet Ramakrishnan, Matthew Weiner, Milos Jorgovanovic, Amanda Pratt, and Ruzica Jevtic. I would also like to thank my friends outside the research group, especially Pengpeng Lu, who understands all my struggles and frustrations. The Berkeley Wireless Research Center has been an incredible place to work at. Brian Richards, James Dunn and Anita Flynn, have offered tremendous help on the tape-outs. Candy Corpus is the sweetest person that I've ever met. Her energetic laughter lighted up the life in BWRC.

My research was funded by DARPA PERFECT Award Number HR0011-12-2-0016, Berkeley Wireless Research Center sponsors, ASPIRE/ADEPT sponsors. TSMC donated two 28nm tape-out for the SWERVE and BROOM project.

Being 6500 miles away from home is hard. I am really fortunate to have a group of Taiwanese friends who are like my second family and keep me company in this foreign country. The late-night pep talks motivated me and supported me through my ups and downs. I have an amazing mother who raised three girls up all by herself. I couldn't imagine how she made it. I am so grateful that she respected my decision of seeking my dream aborad. My sisters, Judy and Abby, have also been really supportive and caring.

Finally and most importantly, I would like to thank my husband, Wayne, for being my best friend and my mentor and offering endless love during the toughest time of my life. I would never have completed this dissertation without him.

Chapter 1

Introduction

1.1 Energy-efficient computing

An energy-efficient computing system is essential for a wide variety of electronic devices. Personal mobile devices have limited battery capacity constrained by the size and the weight of the product. Power-hungry devices would require undesirable, frequent recharging. The Internet of Things (IoT) connects more and more electronic devices, such as vehicles, home appliances, sensors, and health monitors. Moving computation to edge devices could reduce network load by transmitting only relevant data to the cloud. However, more intensive computing tasks of edge devices consume more energy. Data centers in the United State consume about 70 billion kWh per year to power their servers, storage, network, and cooling [3]. Improving energy efficiency in the data center not only reduces costs, but also reduces strain on the power grid. The energy consumption of memory, including cache, main memory, and storage, makes up a significant proportion of the total energy consumption of a computing system. For example, in warehouse scale computers, CPUs (including caches) consume 30%∼60% of total power, DRAM consumes 30%∼48% of power, and storage consumes 5%∼12% of total power [3]. Therefore, memory design must be optimized for energy efficiency.

Dynamic voltage and frequency scaling (DVFS) is an effective way to achieve energy efficiency. At a higher voltage, dynamic energy consumption dominates the total energy according to the formula, $E = CV^2$, where C is the total capacitance switched and V is the operating voltage. Half of the energy consumption can be reduced by reducing supply voltage by just 30%. However, it takes longer to finish a task at a lower voltage, which increases energy consumption from leakage current. Figure 1.1 shows the trade-offs of energy consumption at different voltage levels. Operating at the lowest point of the summation of dynamic and leakage energy consumption achieves the optimal energy-efficient point.

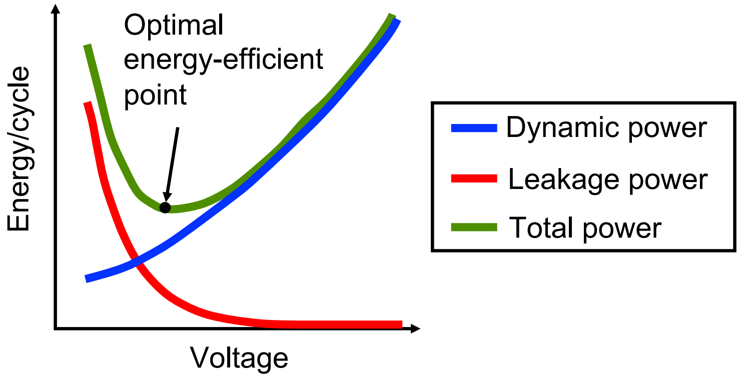


Figure 1.1: Trade-offs of dynamic energy consumption and leakage energy consumption and the definition of the optimal energy enfficient point.

For many applications, such as mobile devices and edge devices in IoT, the devices are idle most of the time. Instead of keeping such devices at minimum voltage to retain data in SRAM or DRAM, data can be copied to nonvolatile memory and power can be shut off. Depending on the ratio of the on/off period, powering off and transferring data in this way may be more energy efficient. However, moving data back and forth is expensive in terms of time and power. Hence, backing up data to flash memories every time the system switches from active state to idle state may not be beneficial due to the timing and power overhead of off-chip accesses. Emerging memory technologies, such as spin-transfer torque memory, resistive memory, and phase change memory, open up new opportunities in this scenario. Compared to the flash memory, these memories operate at a higher speed and greater endurance, allowing more read/write cycles before the cells wear out. Moreover, the CMOS-logic compatibility of the new technologies allows the memory to move closer to the processing, making data transactions cheaper.

1.2 Semiconductor memory technologies

Memory hierarchy creates an illusion of a high-capacity, high-speed and cost-effective memory by utilizing different memory technologies at different levels. The memory technologies can be categorized into two classes, memory and storage, by speed, cost, and nonvolatility.

1.2.1 Memory

Memory for computing systems includes the cache memory and the main memory. Cache memory hides the long latency of the main memory access by storing a copy of data closer to the processing units. Data in the cache are likely to be used again since access patterns typically exhibit temporal or spatial locality. Multiple levels of cache hierarchy optimize the timing and the area of a processor. The level-1 (L1) cache has a small capacity but a short access time while the last level cache (LLC) has a large capacity but a longer access time. Caches are normally implemented with SRAM, which has the fastest speed among all memory technologies.

Although SRAM provides the lowest latency, the budget for power and area limits the maximum capacity of caches. Main memory implemented with DRAM provides a much higher memory density but has a slower speed than SRAM. The size of cache and main memory significantly affects the performance of the processor because a large fast memory reduces the occurrence of slow off-chip storage accesses.

Static random access memory (SRAM)

The most common SRAM bitcell is a six-transistor (6T) structure, as shown in Figure 1.2. The aggressive design rule of SRAM helps achieve a higher density but is more volunerable to the process variation. Lowering the voltage exacerbates the poor stability and writability problems. Therefore, SRAM serves as a bottleneck in deciding the minimum operating voltage (Vmin).

The 8T sturcutre [4], as shown in Figure 1.2(b), introduces a seperate read port to elminate the read disturbance and improve the stability. Some publications also proposed structures that contain more than 8 transistors [5] in order to achieve a lower Vmin, but are not widely used due to a larger cell size. The cost of power and area limits the capacity of caches.

Dynamic random access memory (DRAM)

The DRAM bitcell is composed of one transistor and one capacitor (1T1C), as shown in Figure 1.2. To write the DRAM cell, the access transistor (M1) is activated by the wordline (WL) and the bitline (BL) charges or discharges the storage capacitor ($\mathrm{C_s}$) according to the input value. To read a DRAM cell, the bitlines is normally precharged to VDD/2. When the WL is activated, the charge redistribution takes place between the parasitic capacitance ($\mathrm{C_{BL}}$) and $\mathrm{C_s}$ that increases or decreases the BL voltage. Reading DRAM cell is a destructive operation that changes the charges stored in $\mathrm{C_s}$. Therefore, it needs to write the data back after read operations. Unlike SRAM that can hold the data as long as the power is on, the DRAM cell is subject to leakage current and requires refreshing every 64ms.

The read margin depends on the ratio of $\mathrm{C_s}$ and $\mathrm{C_{BL}}$. To increase the capacitance of $\mathrm{C_s}$ while reducing the cell size, the storage capacitors are built in a stack or trench strcuture. The DRAM fabrication process is not inherently compatible with logic process. Hence, the DRAM chips are integrated on a dual-in-line memory module (DIMM) separated from the processor chip.

Embedded DRAM (eDRAM)

Embedding DRAM [6] on processors allows for wider buses and higher speed. Although eDRAM requires additional process steps that raises fabrication cost, a large on-chip eDRAM still costs less than SRAM with the same capacity. However, eDRAM, like DRAM, requires refreshing operation that complicates the timing control. It is also challenging to increase the capacitance of C_s and read margin.

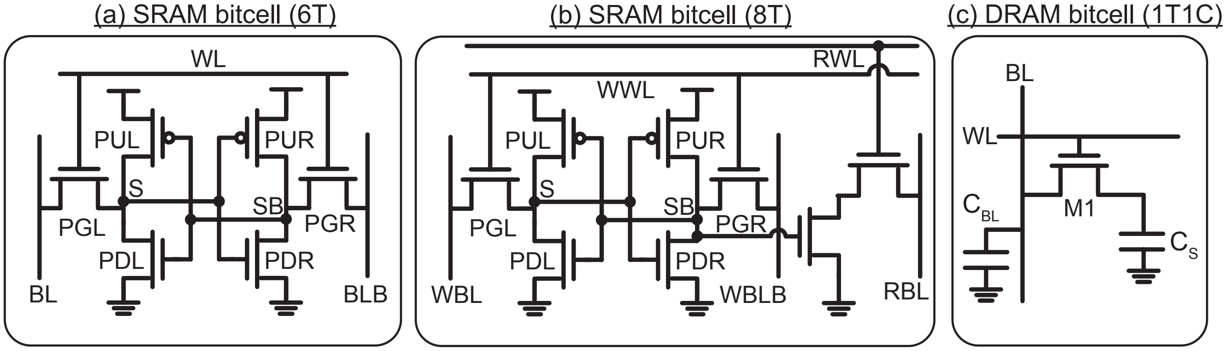


Figure 1.2: The schematic of (a) a 6T SRAM bit cell, (b) an 8T SRAM bit cell, and (c) a 1T-1C DRAM bit cell.

1.2.2 Storage

Hard disk drive (HDD) uses a thin film of ferromagnetic materials on a disk to record data, and solid-state drive (SSD) is based on semiconductor materials. Unlike HDD, SSD devices do not have moving parts so that they don't vibrate and they are more difficult to break. Moreover, SSD consumes less power and has faster speed than HDD. Nevertheless, HDD is still a major part in the storage market due to low cost. Emerging nonvolatile memories, including ferroelectric memory, magnetoresistive memory, phase change memory, and resistive memory, are developed as flash memory replacement that can operate at a higher speed and have a better endurance.

Flash memory

Flash memory is the technology used in an SSD or USB flash drive. The memory cell has a floating gate interposed between a control gate and the MOSFET channel. Changing the amount of electrons trapped in the floating gate shifts the threshold voltage of the transistor. Removing the charges from the floating gate erases the cell and sets the cell to a logical ”1”, while placing the charges to the floating gate programs the cell and sets the cell to a logical ”0”.

In NOR flash, each cell is connected in parallel with one terminal connects to a wordline, one terminal connects to a bitline, and one terminal connects to ground. NOR flash offers faster read speed and random access capabilities that is suitable for code storage. However, the large cell size makes it more expensive than NAND flash. In NAND flash, the cells are connected in series to reduce the cell size. Two selection transistors are placed at the edges of the stack to ensure the connections to ground and to the bitline. NAND flash has a slower read speed but is cost-effective for massive data storage. Vertical NAND (V-NAND) flash stacks the NAND strings vertically to further increase memory density.

By having more threshold voltage states, a single cell can store more than one bit to boost memory density. For example, a multi-level cell (MLC) stores 2 bit per cell and a triple-level cell stores 3 bit per cell. However, one of the challenges of flash memory is the poor endurance due to the program/erase mechanism. While SLC device may endure 10^5 erasures per block, TLC device could wear out after 1000 erasures.

Ferroelectric random-access memory (FeRAM)

FeRAM [7] has the same 1T-1C structure as DRAM with the ferroelectric layer replacing the dielectric layer to gain nonvolatility. The write operation applies a field across the ferroelectric layer polarizing the electric dipoles of the material. The polarization of the dipoles remains after removing the electric field. The read operation forces the cell to ”0” state. If the prior state was a ”1”, the re-orientation of atoms causes a brief pulse of current. Otherwise, no pulse is detected when the prior state was a ”0”. The read process is destructive and requires write-after-read. Compared to flash memory, FeRAM consumes lower power, operates at higher speed and has higher endurance. However, it has a much lower density than flash memory that leads to a higher cost.

Magnetoresistive random-access memory (MRAM)

MRAM cell is composed of one selection transistor and one magnetic tunnel junction (MTJ). MTJ is a structure with an insulating layer sandwiched by a fixed layer and a free layer of ferromagnetic material. When the magnetization of the free layer aligns with the fixed layer, the resistance of the cell is lower than that when the magnetization of the two layers is antiparallel. Spin-transfer torque (STT) [8, 9] is a technique that can switch direction of magnetization by passing current through the layer. STT-MRAM has high endurance and high switching speed, but the scaling of selection transistors is limited by the required amount of write current. The small resistance ratio of the device also poses a challenge on sensing.

Phase change memory (PCM/PRAM)

PCM [10, 11] uses a chalcogenide alloy called $Ge_2Sb_2Te_5$ (GST) as a storage device. By heating and cooling GST, the cell is able to switch between the crystalline and the amorphous states. The crystalline state (low resistance state) represents a logical ”1” and the amorphous state (high resistance state) represents a logical ”0”. PCM is more appealing than flash memory because of scalability and high speed. However, it requires high programming current density and is subject resistance drift.

Resistive random access memory (RRAM/ReRAM)

RRAM [12, 13] is often referred to as memristor. A wide range of materials can be used for RRAM, including HfO_x, TiO_x, and more complex compounds. RRAM cell is set or reset to different resistance states by building or breaking the filament of oxygen vacancies. Most RRAM materials are bipolar, that is, it can be set/reset to different resistance states by applying a voltage with different polarity. Some materials also require a forming operation that uses a sufficiently high voltage to form the filament for the first time.

RRAM has many attractive characteristics, including the back-end-of-line (BEOL) compatibility, high resistance ratio, and high speed. Although it is more durable than flash memory, it still suffers limited endurance.

1.2.3 Storage class memory (SCM)

There is a performance and capacity gap between memory and storage. SCM is a new class of memory technology that has performance and density that falls between DRAM and flash memory. The benefit of SCM is that it can plug into JEDEC standard DDR DIMM socket and it is much faster than flash memory placed on the PCIe bus. Moreover, it is a persistent memory so it can mitigate the risk of data loss from a power failure. Unlike flash memory that only allows block-wise accesses, SCM is byte-addressable.

However, this memory technology is expensive and has a niche market. For example, NVDIMM-N has a higher price since there are flash memory and DRAM on the same module. Also, it is not made in large quantities; therefore, the manufacture cost is relatively high. High-speed and high-density emerging nonvolatile memories are promising candidates for a low-cost SCM since it requires only one type of memory. As the hardware cost comes down and storage layer software comprehends SCM better, it will be more widely applicable and beneficial to the computing system.

1.3 Limitation on energy-efficient memory

The process, voltage and temperature (PVT) variation and the disturbance from coupled noise sources cause SRAM bitcells to fail at low voltages. With SRAM as a bottleneck, the processing unit needs to operate at a higher voltage than logic Vmin to prevent failures. Cache resliency techniques can reduce the minimum operating voltage of caches and improve energy efficiency.

On the other hand, nonvolatile memories suffer more than just PVT variations. As flash memory packs more bits into a single cell, its read margin decreases. The wide distribution of the threshold voltage of memory cells makes it even more challenging to access the data reliably. Emerging nonvolatile memories define 0s and 1s with different resistance states. The distributions of different resistance states are close to or even overlapping with each other due to the variation from manufacturing and read/write conditions. Some researchers have also reported resistance drift over time [10] that could exacerbate yield issues. Moreover, the finite endurance of nonvolatile memories limits the possibility of moving them closer to the processing units that require frequent accesses. Memory techniques are necessary to handle different problems and ensure that new memory technologies can be used reliably in computing systems for a better energy efficiency.

1.4 Techniques at different levels

A wide range of design efforts at different levels (device, circuit, and micro-architecture) has been explored on the topic of energy efficiency. (Figure 1.3) Memory resiliency is the key to allow reliable computing while reducing the energy consumption.

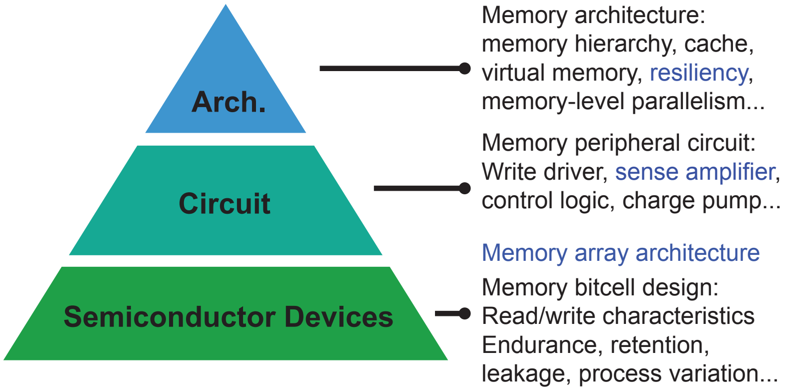

Arch.
Circuit
Semiconductor Devices
Memory architecture:
memory hierarchy, cache,
virtual memory, resiliency,
memory-level parallelism...
Memory peripheral circuit:
Write driver, sense amplifier,
control logic, charge pump...
Memory array architecture
Memory bitcell design:
Read/write characteristics
Endurance, retention,
leakage, process variation...

Figure 1.3: Different levels of memory design, including semiconductor devices, circuit, and architecture.

Circuit-level approach

Specifically for SRAM, the circuit-level assist techniques, such as boosting the wordline voltage [14], pulling the bitline to a negative voltage [15, 16], and collapsing the cell supply voltage [17, 18], are able to reduce the bit error rate (BER) and increase the write ability or the read stability to allow low-voltage operation.

Sense amplifiers are crucial to all types of memory. Reducing the offset voltage of the sense amplifiers helps to detect small signal differences. Hence, the memory can be read reliably at low voltages. Offset cancellation is normally done by auto-zeroing capaitors [19]. However, the capacitors are difficult to fit into the tight layout of the sense amplifier. The sense amplifier redundancy [20] is able to combine the offset voltages of sub-unit SAs and choose the best configuration. The hidden cost of the approach is the time/area/energy of programming the fuses for the configuration. RazorSRAM [21] improves the throughput by eliminating the guard band of the sensing margin and uses error detection and correction to fix the failing bits. Microarchitectural modification is required to realize this scheme.

Architecture-level approach

Instead of reducing the BER, the architecture-level assist techniques allow memory to operate under a lower voltage by tolerating failing bitcells and preventing the errors from propagating through. Conventional error detection and correction codes, such as Hsiao code [22], detect and correct soft errors on the fly. More complex codes, such as BCH [23], can detect three errors and correct two errors with higher timing and area overhead. Multiple-bit segmented ECC (MS-ECC) [24] sacrifices a large portion of cache (25%∼50%) to store ECC check bits, which leads to performance loss at low-voltage mode. Variable-Strength ECC (VS-ECC) [25] handles the rare multi-bit error with a small extended ECC field of 4EC5ED. Memory redundancies fix the errors at known locations by replacing them with redundant columns/rows. Column-redundancy schemes [26, 27] handle hard faults by adding spare columns (SC) to an SRAM array and configure the multiplexer to switch to a redundant column while there are faults presented in the column.

Device-level approach

Exploiting emerging memory technologies can help improve energy efficiency. However, many reliability issues surface as they are placed closer to the processing units. Different techniques are proposed to solve the intrinsic problems of the emerging nonvolatile memories. To handle the resistance drifting problem, the calibration technique [10] can compensate the differences and enlarge the margin between the states. Wear leveling [28] is the technique that arranges the data so that writes are evenly distributed to each cell. Therefore, it can prevent fails due to a high concentration of write cycles to a few blocks. Various cell structures and array architectures [29] are proposed to improve the read margin and achieve a higher speed.

1.5 Scope of the dissertation

This dissertation proposes approaches at several design levels that can improve energy efficiency in computing systems.

Memory peripheral circuits enable reliable read and write operations of memory arrays. Sense amplifiers (SA) are essential in many different kinds of memory to allow the detection of a small voltage difference between the bit cell and the reference. When reducing the voltage for better energy efficiency, the increasing offset voltage and smaller read margin lead to a sensing problem. Different SA topologies are discussed in this dissertation to evaluate the offset voltage at a lower supply voltage. SA design with low offset voltage helps memory to deliver correct output data at higher speed and lower power.

Architecture-level assist techniques prevent a small amount of faults in the memory from being seen by the system. Various techniques for the ability to tolerate errors at the microarchitectural level are explored, and the corresponding timing and area overheads are evaluated. By allowing a higher bit error rate, the processor can operate under a lower operating voltage to improve the energy effiency.

RISC-V, an open-source instruction set architecture, and Rocket Chip, an open-source SoC generator, provide an ecosystem for easy evaluation of microarchitectural research concepts in a realistic processor system, ultimately allowing silicon measurement that can convincingly demonstrate these techniques. The flow of chip implementation is also described in this dissertation.

New semiconductor memory devices, such as resistive random access memory (RRAM), have the potential to change the current memory hierarchy and reduce energy consumption by storing data in nonvolatile memories while idle. In order to closely integrate the new technology with the processor, the characteristics of the memory device need to be carefully evaluated. The characteristics of the emerging nonvolatile memory technologies include write/read performance, stability, endurance, retention, cost and yield. The crosspoint array structure of RRAM exhibits high density, but has severe leakage problems due to the absence of access transistors that could lead to read failure. This dissertation investigates the leakage issue and proposes a new array architecture that ensures a reliable read operation while retaining high memory density.

1.6 Thesis outline

This dissertation involves work at different levels of memory design. With the common goal of energy-efficiency and resiliency, the proposed schemes aim at reducing Vmin and improving reliability. The ideas are verified with simulations and two fabricated RISC-V processors.

Chapter 2 investigates the sense amplifier in the memory. The double-tail sense amplifier (DTSA) is proposed to improve the offset voltage so that the SRAM can operate under a lower Vmin compared to a conventional SRAM. The benefit of Vmin reduction and the performance improvement of DTSA has been measured in a 28nm test chip.

Chapter 3 explores various architecture-level assist techniques that could mask errors and allow an SRAM-based cache to operate at a lower voltage. Architecture-level assist techniques, including line disable, dynamic column redundancy, bit bypass, and line recycling, are implemented in an out-of-order processor and a memory hierchary with L1 and L2 cache in a 28nm process.

Chapter 4 describes the flow of the physical implementation of the processor utilizing Chisel and the Rocket Chip SoC generator. A case study of implementing a register file for the out-of-order processor with gate-level description and guided place-and-route is discussed.

Chapter 5 introduces emerging nonvolatile memories (NVM) and the advantages and challenges of the crosspoint array architecture. The differential 2R crosspoint Resistive Random Access Memory (RRAM) is proposed to enlarge the read margin and eliminate data-dependent IR drop. Leakage and endurance issues are analyzed for the crosspoint array.

Chapter 6 concludes the dissertation and lists directions for future research of energy-efficient and resilient memory design.

Chapter 2

Sense Amplifiers for Low-Voltage SRAM

This chapter explores one of the key components of the memory circuit – the sense amplifier. A double-tail sense amplifier (DTSA) is designed as a drop-in replacement for a conventional SRAM sense amplifier (SA), to enable a robust read operation at low voltages. A pre-amplification stage helps reduce the offset voltage of the sense amplifier by magnifying the input of the regeneration stage. The self-timed regenerative latch simplifies the timing logic so the DTSA can replace the SA with no area overhead. A test chip in 28nm technology achieves 56% error rate reduction at 0.44V. The proposed scheme achieves 50mV of VDDmin reduction compared to commercial SRAM with a faster timing option that demonstrates a smaller bitline swing.

2.1 Basic SRAM operations

Static random access memory, SRAM, is an on-chip memory that has the lowest latency but the largest bitcell size. SRAM is normally used to build caches in the processor due to the high speed and the CMOS process. Figure 2.1 shows the block diagram of an SRAM macro and the schematic of one bitcell. The SRAM consists of an array of bitcells and peripheral circuits, including control logic, decoder, column multiplexer and sense amplifier. The 6-transistor (6T) structure of an SRAM bitcell is composed of six CMOS transistors: the pull-up transistors (PUL, PUR) and the pull-down transistors (PDL, PDR) form a cross-coupled inverter, and the pass-gate transistors (PGL, PGR) connect to the complement bitlines (BL, BLB).

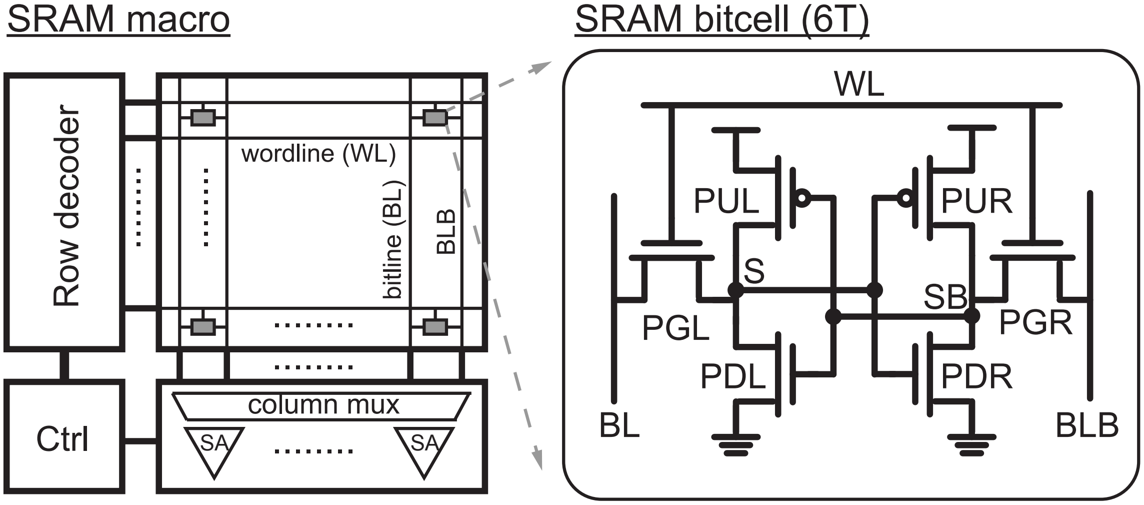


Figure 2.1: Block diagram of an SRAM macro with an array, decoder, control logic, column multiplexer, and sense amplifiers. The bitcell is composed of six transistors.

Figure 2.2 illustrates the write and read operations. When writing the bitcell, the bitlines are pulled differentially to VDD and ground according to the input data, e.g. when writing a 1 to the cell, BL is pulled up to VDD and BLB is pulled down to low. The wordline (WL) pulse activates PGL and PGR so that one of the storage nodes (S or SB) is decreased down to V_{DD}-V_T and a positive feedback of the cross-coupled inverter begins to pull S and SB to VDD and ground.

In the write-1 operation, PGR is fighting over PUR to lower the voltage at SB. Therefore, PG is normally designed to be stronger than PU. Mismatches within the bitcell could weaken the pull-down strength. To enhance the write ability, some assist techniques use negative bitline (NBL) [15, 16] or wordline boosting [14] to strengthen the PG, or collapse the cell supply [17, 18] to weaken the PU.

When reading the bitcell, the bitlines are precharged to VDD. Depending on the stored data, one side of the BL will be pulled down while the other side will be kept high. The small voltage difference, ΔV, or bitline swing can be detected by the sense amplifier at an early stage to increase the read speed and reduce the power.

In the read-1 operation, a voltage bump at SB, due to a voltage divider formed by PGR and PDR, can be viewed as a noise source and flip the cell state. To prevent the read disturb, assist techniques aim at reducing the level of the voltage bump, which is related to the PGR and PDR ratio. Thus, PD is designed to be stronger than PG. To improve the read stability, we can either weaken the PG by underdriving the wordline [30] and lower the bitline [31], or strengthen the PD by reducing the cell VSS.

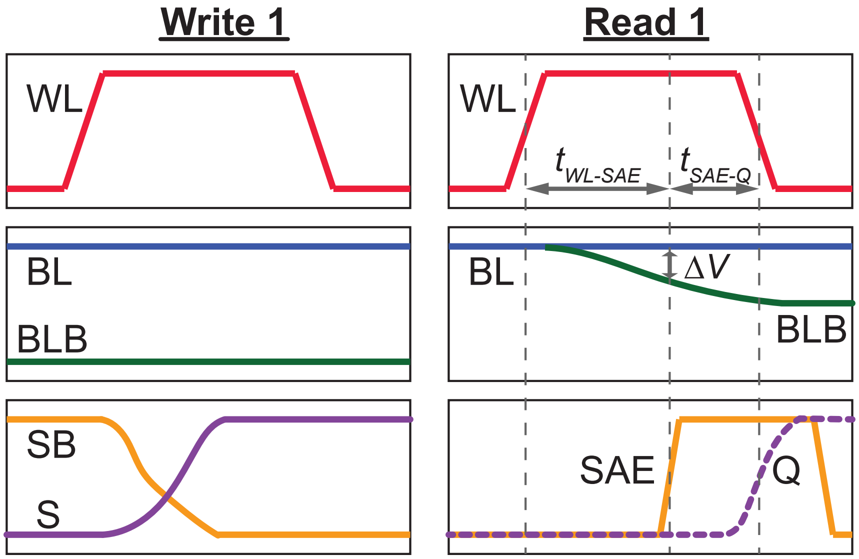


Figure 2.2: Waveforms of SRAM read/write operation.

2.2 Challenges for low-voltage sensing

The aggressively compact size of the bitcells and the massive SRAM capacity along with the process, temperature and voltage (PVT) variation cause a wide distribution in the performance and Vmin between bitcells. To cover the worst bitcells, the operating voltages need to be elevated for a robust computing, which sacrifices the energy efficiency. Many researches has focused on improving the read noise margin (RNM) and write margin (WM) of the SRAM bitcell so that it is able to operate under a lower voltage. The circuit-level assist techniques are highly dependent on the device parameters in different technology nodes. With limited accesses to the transistor-level design of the SRAM IP, it takes significant effort to modify and re-characterize the SRAM circuit.

Variations associated with technology scaling together with increased on-die memory capacities cause a wide distribution of bitline swings (ΔV), requiring a tight control of the sense amplifier offset voltage in SRAMs. To achieve the targeted sensing yield, the conventional sensing scheme margins the bitline swing with a longer sensing time (WL-SAE delay), which degrades the performance and consumes more energy, as shown in Figure 2.3. A sense amplifier with a lower offset voltage is able to resolve the output at a shorter WL-SAE delay and improves the performance and power.

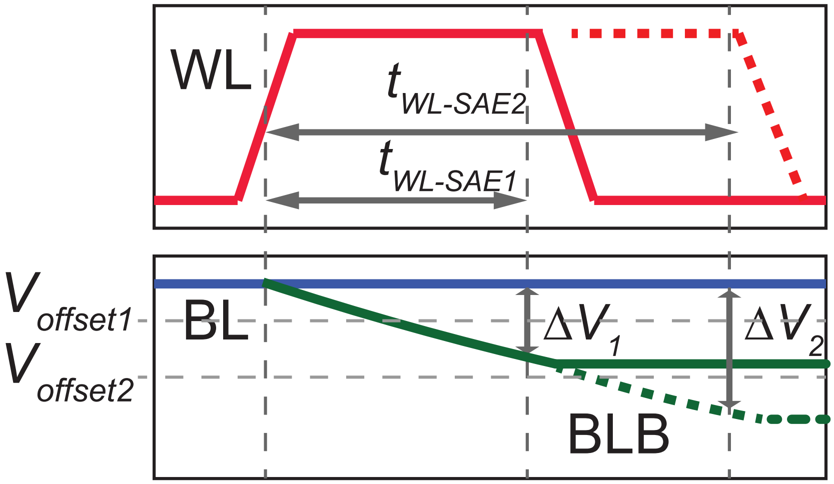


Figure 2.3: A longer WL pulse and WL-SAE delay result in a larger ΔV for a more reliable sensing but degrade the read speed and power consumption.

Figure 2.4 shows the Monte Carlo simulation results of ΔV in a commercial SRAM in a 28nm process, indicating that the distribution of ΔV at 0.6V is twice as wide as the distribution at 0.9V. This is mainly because that the V_{TH} variation has more impact at a lower voltage. Therefore, a more robust sensing scheme is needed to be able to retrieve correct output data for a wide range of operating voltage.

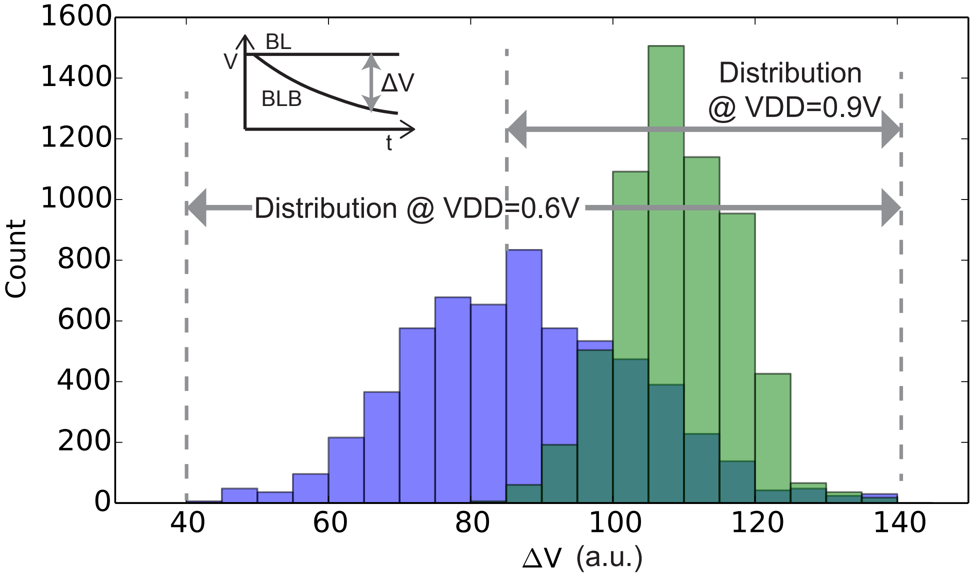


Figure 2.4: Histogram of bitline swing (ΔV) at 0.9V and 0.6V.

2.3 Conventional Sense Amplifiers and Pior Work

The two widely-used sense amplifiers are the conventional latch-type sense amplifier [32] and the StrongArm sense amplifier shown in Figure 2.5(a) and Figure 2.5(b), respectively. The conventional SA is the most common SA in the commercial SRAMs due to its compact structure. It is composed of a 6-transistor structure resembling an SRAM bitcell with a tail transistor (M_{TAIL}) to trigger the SA, a pair of precharage transistors (M_{PRE}) to reset the internal states, and a equalization transistor (M_{EQ}) to guarantee a fair comparison. Some local timing signals are needed to carefully separate the operations of realeasing the precharged bitline ($PREB$), turning off the equalization ($BLEQ$), connecting to the bitline voltage (PGB), and triggering the sense amplifier (SAE). The offset voltage of a conventional SA gets worse at a lower voltage because of the impact on the threshold voltage variation and mismatches is higer. Therefore, the trend of smaller bitline swing and higher the offset voltage limits the robustness of the conventional SA.

The StrongArm [33] also has a compact structure with the input transistors (M_{IN}) stacked below the cross-coupled inverters. Since the bitline signals are connected to the gate of M_{IN}s, the output signals are decoupled from the parasitic capacitance of the bitlines and no need to carefully control passgate timing to isolate the parasitic capacitance. However, the topology stacks four transistors, which suppresses operational regions of the transistors at low voltages.

Various sensing techniques have been proposed to improve the robustness of SAs, but the techniques require additional complexity. Capacitors are commonly used to calibrate the offset voltage in the sense amplifier [19]. However, the SA in the SRAM macro needs to be compact to fit in to the column pitch. The capacitors usually take up an area of multiple columns and the metal layers for the metal-oxide-metal (MOM) capacitor restricts the availability of signal/power routing. The reconfiguration switches also require careful timing control to change between offset storage, bitcell read, and output mode. The SA redundancy [20] [34] adds an extra set of SA to choose a better offset voltage among the two SAs or reconfigure the connections between the two SAs so that the offset voltage could be cancelled out. The SA redundancy scheme requires extra steps to test all configurations and program the best configuration with fuses. Razor-style [35] error detection and correction (EDAC) technique can also be applied to the SRAM arrays [21] to eliminate the guard band and increase the read throughput. However, microarchitectural modifications are needed to handle the error occurrence.

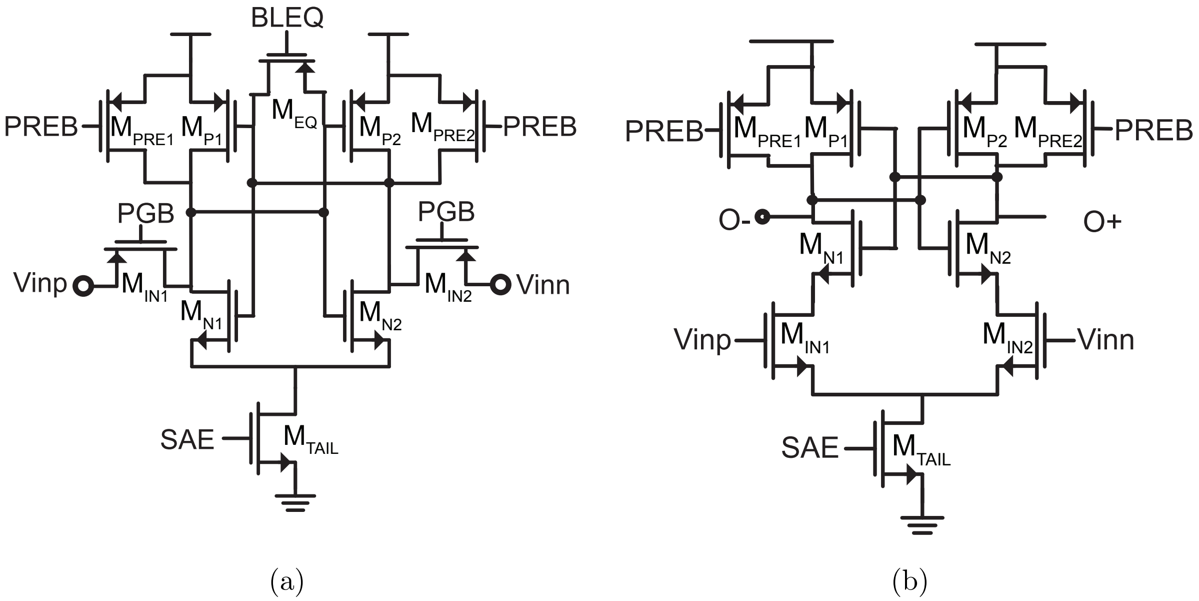


Figure 2.5: The schematic of (a) a conventional SA and (b) a StrongArm SA.

2.4 Double-Tail Sense Amplifier

The double-tail sense amplifier (DTSA) was first proposed to reduce the offset voltage and the noise in the analog-to-digital converters (ADC) [36][37]. As the offset requirement of sense amplifier in SRAM gets more stringent, the topology can also be adopted for offset reduction in low-voltage SRAMs. The schematic of the DTSA is shown in Figure 2.6. It consists of two stages: the dynamic pre-amplification stage provides a small gain through the input pair, $\mathrm{M_{IN1,\ IN2}}$ and the regeneration stage, consisting of cross-coupled inverters ($\mathrm{M_{U1,U2,D1,D1}}$) completes the comparison.

Figure 2.7 shows the simulated waveforms of the DTSA at 0.6V VDD and 50mV ΔV. Vinp and Vinn are set to VDD and VDD-ΔV, respectively. During the reset phase, DN and DP are precharged to VDD by $\mathrm{M_{PRE1}}$ and $\mathrm{M_{PRE2}}$. All the internal nodes in the second stage are discharged to 0 by $\mathrm{M_{N1\text{-}N4}}$. After the first stage is activated by SAE, DN and DP are discharged at different rates depending on the input voltage levels (Vinn, Vinp), the parasitic capacitance, and the tail current. The output signal of the first stage, DN and DP, serves as both the enabling and the input signal for the second stage. As the voltage of DN and DP drop below VDD-$\mathrm{V_{TP}}$, the second stage is turned on by $\mathrm{M_{H1,2}}$. There is a short period of time in which only one side of the latch is enabled so that the side that is pulled high can develop the signal without contention with the other branch. Three pairs of input transistors, $\mathrm{M_{H1,2}}$ and $\mathrm{M_{N1\text{-}4}}$, increase the pre-amplification gain and reduce the input-referred offset voltage. Finally, the latch completes the signal regeneration and the output signals (OP, ON) are fed to an SR latch.

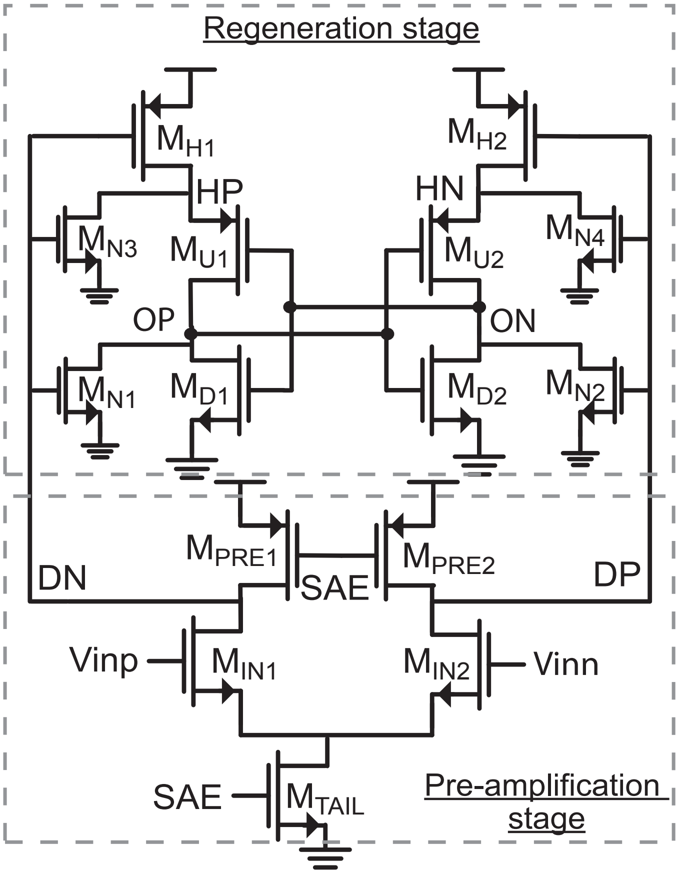


Figure 2.6: Circuit diagrams of the double-tail sense amplifier, consisting of the preamplification stage and the regeneration stage.

Figure 2.8 shows the simulated error rate of the DTSA and the conventional SA at different ΔV and supply voltages. The error rate of the DTSA is slightly worse than that of the conventional SA at 0.9V, but normally it is not a problem as the bitline swing ΔV is large enough to overcome the offset voltage. The DTSA is more robust at lower supply voltages because of the integration in the pre-amplification stage. The error rate is reduced by 85% at 0.45V with a fixed ΔV of 20mV and the offset voltage under the iso-robustness condition (99.9% yield) is 22% lower than the conventional SA. The conventional SA shows the worst case at SF corner with 8.1x larger error rate than that at TT corner, while a mix of PMOS and NMOS input pairs make the DTSA equally resilient across all corners, as shown in Figure 2.9. The improvement is more prominent when considering all corner cases.

The wider ΔV distribution at low VDD requires a sense amplifier that has lower offset at low voltage to compensate for the low read current in the weakest bitcells on the chip. The

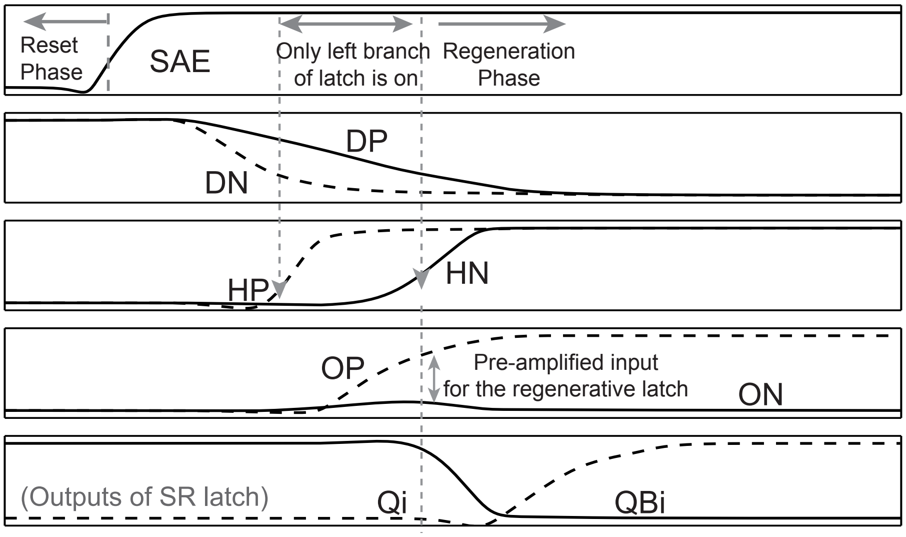


Figure 2.7: Operation waveforms of the DTSA at 0.6V and ΔV = 50mV.

current ratio of M_{IN1} and M_{IN2} in the DTSA under the same ΔV becomes larger as VDD is decreased. A higher gain for the pre-amplification stage in the DTSA at low voltage is achieved due to a larger current ratio in the input pairs and a longer integration time. The Monte Carlo simulation result in Figure 2.10 shows that the offset voltage of the conventional

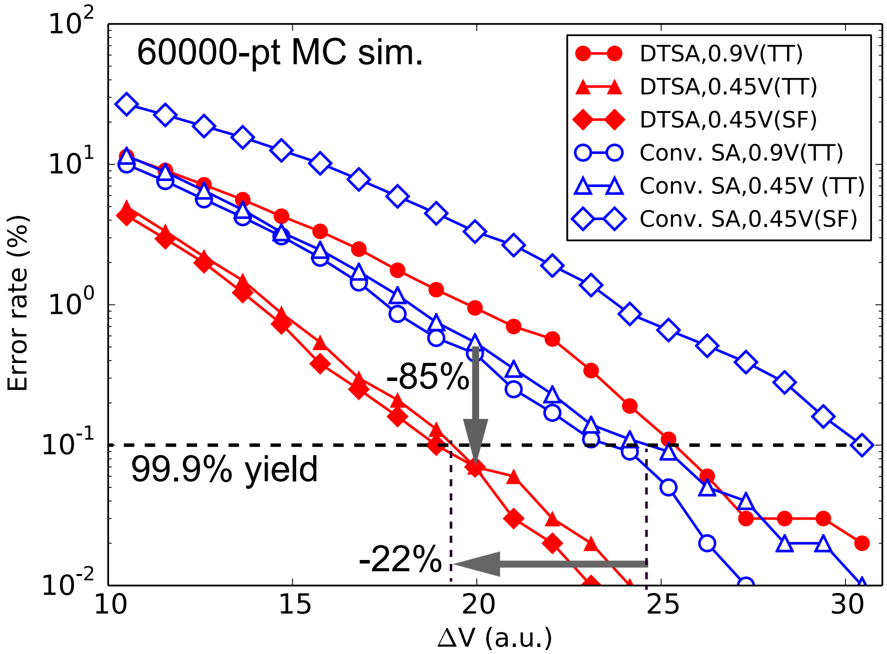


Figure 2.8: Simulated error rate with various ΔV for the conventional SA and the DTSA.

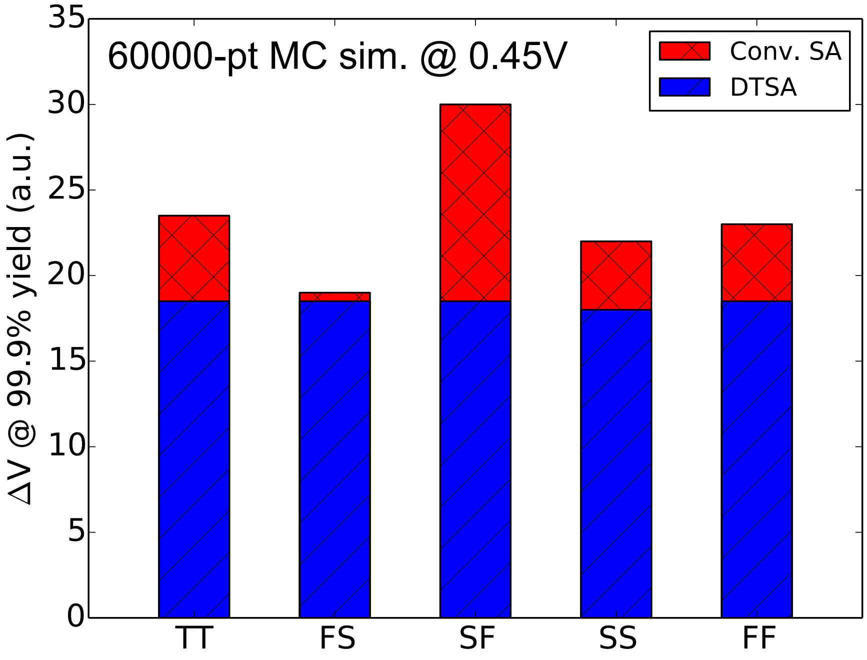


Figure 2.9: Offset voltages at different process corners (VDD=0.45V).

SA increases for lower VDD, while the offset voltage of DTSA actually decreases at low voltages. The DTSA shows better robustness when VDD is lower than 0.7V. Therefore, the DTSA is favored for low-voltage applications.

The offset voltage is largely determined by the sizing of the input pairs (M_{IN1} and M_{IN2}) and the tail transistor (M_{TAIL}). The sizing of the regeneration stage is relaxed because of the pre-amplified signal. The offset voltage is reduced by using a longer gate length for the input pairs, which also increases the area and the load capacitance of the first stage. By decreasing the size of the tail transistor, the integration time for the pre-amplification stage becomes longer, which results in a larger ΔV for the regeneration stage. However, this change degrades the sensing time. Figure 2.11 shows the trade-off between speed and offset voltage. The size of M_{TAIL} is chosen to have a low offset voltage with acceptable speed degradation.

Figure 2.12 illustrates the dimensions of the conventional SA and the DTSA with the timing logic and the output latch. The DTSA has 50% more transistors than the conventional SA, resulting in a 23% area overhead. However, the sense amplifier itself occupies only half of the area of the read-out circuit in the traditional design, as timing logic is required to generate the PREB, PGB, and BLEQ inputs. Since the regenerative latch of the DTSA is self-timed, it only requires a single-phase clock. Furthermore, the regeneration stage is released from the initial stage when the pre-amplified signal is ready. Therefore, no PREB, PGB and BLEQ signals are needed and the area required to generate the timing logic is reduced. As a result, although the DTSA consists of more transistors, the entire read-out circuit fits in the same area footprint. Similarly, although the additional pre-amplification

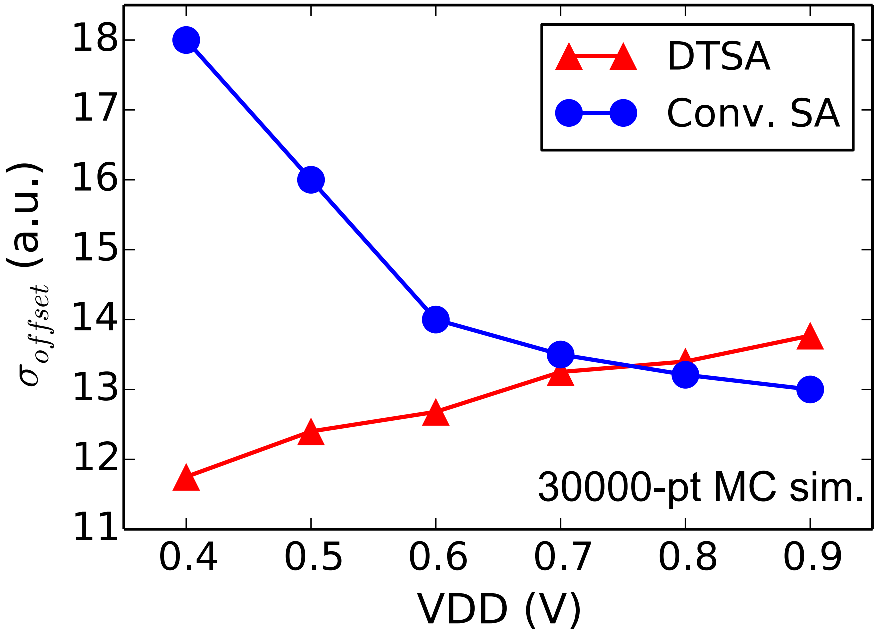


Figure 2.10: The offset voltage of the conventional SA and the DTSA at different supply voltages.

stage consumes 74% more power than the conventional SA, the overall power of the memory circuit is reduced due to a lower bitline swing in the read operation.

2.5 Silicon Measurement Results

Four 72kb SRAM macros were fabricated in the TSMC 28nm HPM process, two with the conventional SA and two with the DTSA. For the two macros that have the same SA, one is with the original timing circuit in the commercial SRAM and the other is with the tunable timing circuit to evaluate different ΔV and WL-SAE delay options. The die photo and the measurement setup are shown in Figure 2.13. The test SRAM macros are placed within a RISC-V processor and are tested by running the built-in self-test (BIST) to identify and count the errors while sweeping voltages and frequencies. The tunable timing circuit is constructed by a 3-bit delay chain, which gives 7 different WL-SAE delay and ΔV. The tunable timing code (TTC) is set to a lower value to enable early SAE triggering for a smaller ΔV.

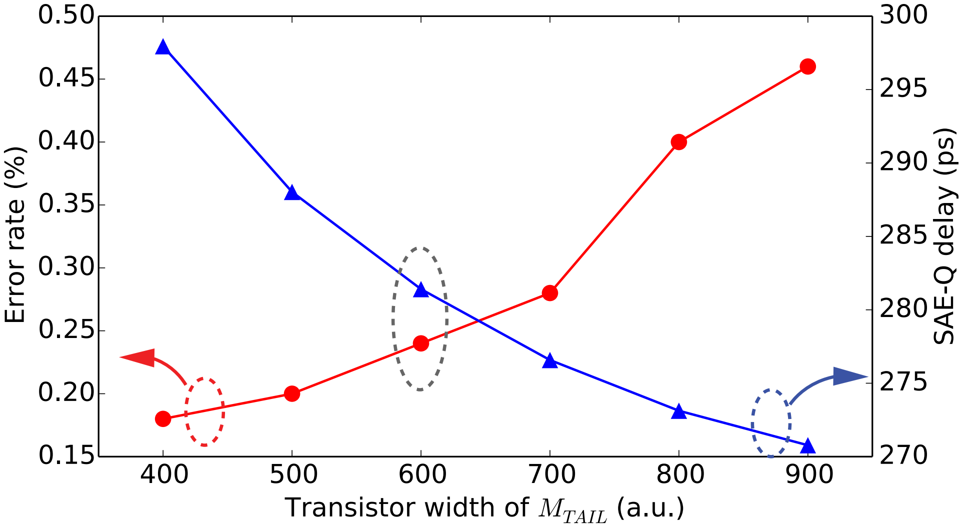


Figure 2.11: Trade-off between error rate and SAE-Q delay for various sizings of $\mathrm{M_{TAIL}}$.

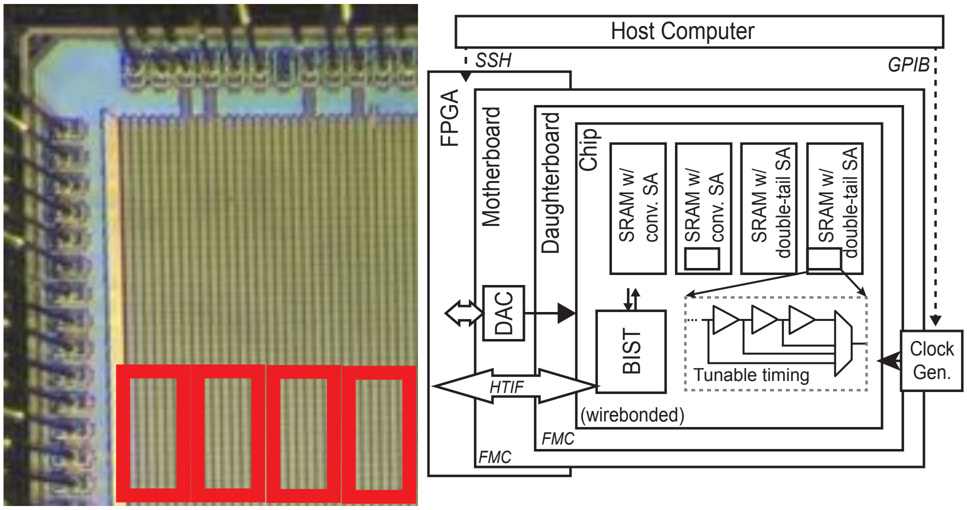


Figure 2.13: Die photo and measurement setup.

The measured error rate of the SRAM with the DTSA and different tunable timing codes are shown in Figure 2.14. The error rate is averaged over 6 chips with 72 SAs per chip (432 SAs in total). By simply replacing the conventional SA with the DTSA, the SRAM with the DTSA achieves a 56% error rate reduction compared to the commercial SRAM with the same timing settings. The red line shows that the error rate of the DTSA is increasing when operating under smaller TTC, i.e., shorter CLK-SAE delay and smaller ΔV. The relationship between TTC, CLK-SAE delay and ΔV is also shown in Figure 2.14. The

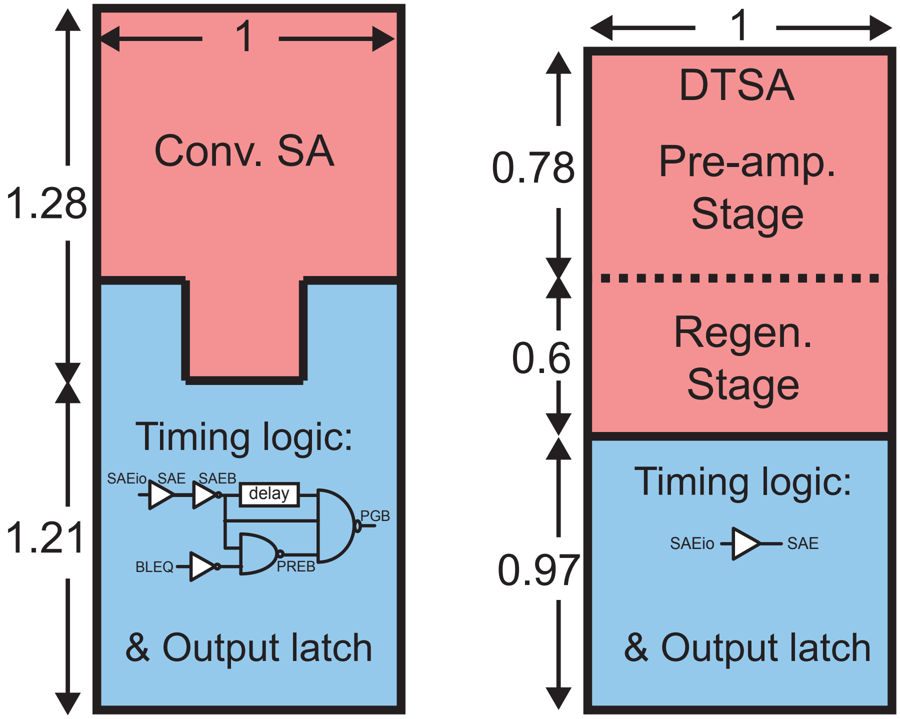


Figure 2.12: Dimensions of the conventional SA and the DTSA (normalized to width).

DTSA achieves comparable error rate to the commercial SRAM around tunable timing code 3, which corresponds to 41% shorter sensing time (CLK-SAE delay) and 46% smaller ΔV in post-layout simulation. However, the distribution for 432 SAs does not capture the worst-case process variation. Therefore, to guarantee high yields, extra margin should be added to cover the worst case. If a 10mV margin is added based on the worst-case corner (SF corner) simulations shown in Figure 2.8, the ΔV is still improved by 30%.

Adjusting the tunable timing effectively shifts the ΔV distribution. Figure 2.15 shows the error rate at different VDD and CLK-SAE delay. A longer CLK-SAE delay (larger TTC) allows larger ΔV and ensures enough read margin. However, it reduces the error rate at the expense of performance and energy efficiency. With the most aggressive timing settings, for example, TTC=1, the error rate at lower voltages increases rapidly because of a lower mean value for ΔV and the long tail in the ΔV distribution. At TTC=3, with 46% smaller ΔV DTSA achieves the same robustness as the conventional SA in the commercial SRAM.

The shmoo plots in Figure 4.7 show the operational range of the supply voltage and the frequency of the SRAM with the conventional SA and the DTSA, respectively. The SRAM with the DTSA operates at TTC=3 with 46% smaller ΔV. The operational frequencies are the same for the two SRAMs with different SAs at higher voltages, When the voltage is dropped down below 0.5V, the operational frequency starts to decrease and eventually

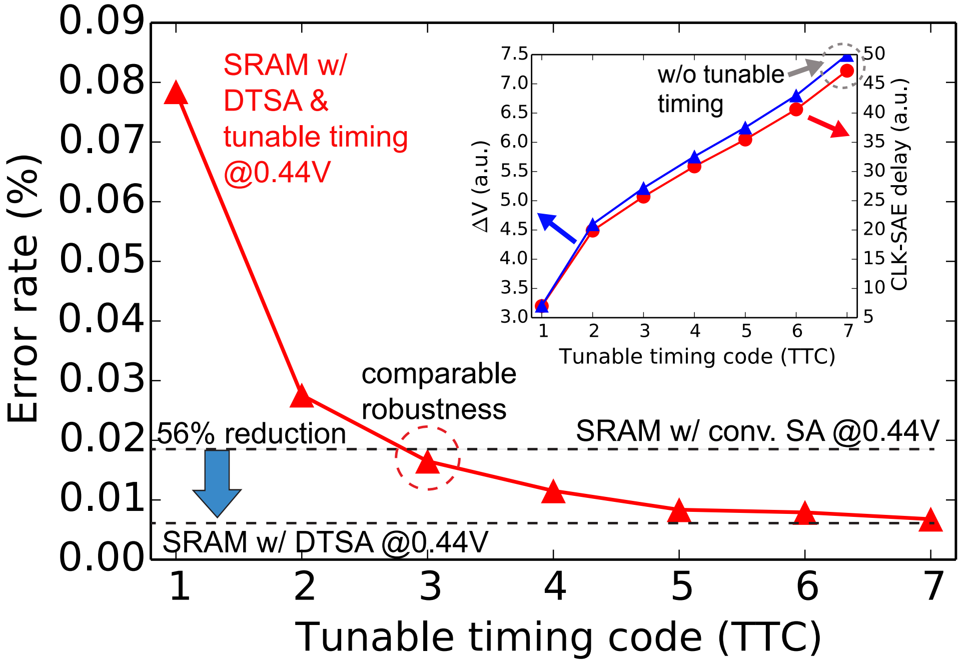


Figure 2.14: Measured error rate for various tunable timing settings at 0.44V (averaged across 6 chips).

the SRAM with the conventional SA stops working while the SRAM with the DTSA keeps functional down to 0.45V. The SRAM with the DTSA achieves 50mV of VDDmin reduction compared to the commercial SRAM. The DTSA enables CLK-SAE delay reduction, which allows the SRAM to operate at a higher frequency and a lower voltage.

2.6 Comparison

Table 2.1 compares the conventional SA and the DTSA with previously published SAs. VTS-SA [19] utilizes the same cross-coupled latch for the static pre-amplification and offset compensation with reconfiguration. It induces short-circuit current in the pre-amplification phase, requires extra auto-zeroing capacitors and precise array-scale timing circuits in the offset sampling phase. The Reconfigurable SA [20] is able to combine the offset voltages of sub-unit SAs and choose the best configuration. The hidden cost of the approach is the time/area/energy of programming the fuses for the configuration. RazorSRAM [21] improves the throughput by eliminating the guard band of the sensing margin and uses error detection and correction to fix the failing bits. Microarchitectural modification is required to realize this scheme.

All approaches report large improvements in robustness, performance, or offset voltage reduction. However, the hidden costs of design complexity make adoption of these approaches

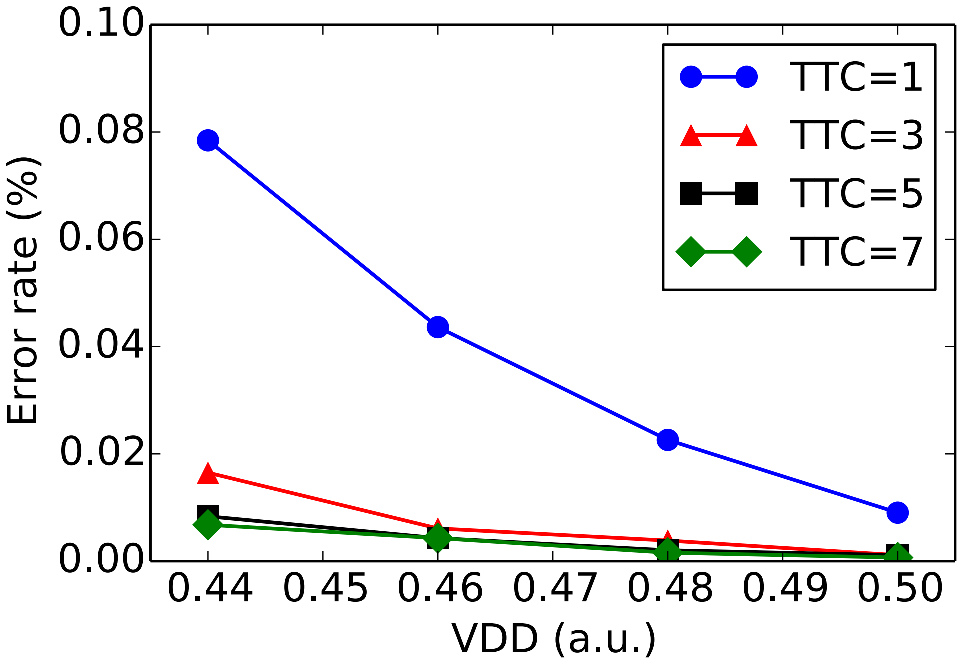


Figure 2.15: Measured error rate at different VDD and tunable timing settings (averaged across 6 chips).

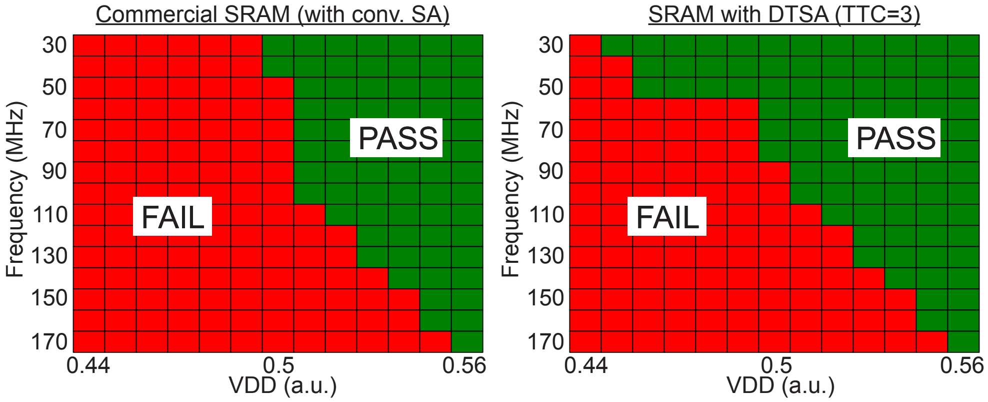


Figure 2.16: Shmoo plots of the SRAM with the conventional SA and the DTSA.

cumbersome. The DTSA improves the offset voltage and the sensing speed at low voltage by changing the circuit topology with minimum design effort. The comparable area and compatible timing signals make the DTSA attractive as a drop-in replacement of the conventional SA in commercial memories.

Table 2.1: Comparisons between various published sense amplifiers.

Source	Conventional	DTSA(this work)	VTS-SA [19]	Reconfig. SA [20]	RazorSRAM [21]
Technology	28nm	28nm	28nm	28nm FDSOI	28nm FDSOI
Circuit area	1X	1X	1X+MOM cap	1X+fuse	-
Features	-	Dynamic preamplifier, self-enabled latch	Reconfigurable static preamplifier and offset calibration	Reconfigurable SA, redundancy	High throughput, error correction and detection
Design effort overhead	-	None	Extra capacitors, complex timing	Run BIST Test, configure with fuses	Error control in memory controller
Sensing speed improvement	-	13.3% @0.6V	34% @1V	-	79% @0.6V
Offset voltage reduction	-	22% @0.45V	-	58.8%	-

2.7 Conclusion

For SRAMs aiming at a low-voltage operation, the conventional SA could not meet the offset requirement as the distribution of the bitline swing at lower voltages gets wider. The DTSA, consisting of one preamplification stage and one regeneration stage, enables robust SRAM sensing at low voltages, which allows for smaller bitline swings with early SAE timing. Preventing excessive bitline discharge improves both the performance and the energy efficiency of SRAM. Although small ΔV leads to a longer SAE-Q delay, the early SAE timing still contributes to a 13.3% reduction of the overall CLK-Q delay in post-layout simulation at 0.6V. The silicon measurement results show 56% error rate reduction at 0.44V and 50mV VDDmin reduction by replacing the conventional SA in commercial SRAM with the DTSA design. The self-timed latch saves the area for additional timing logic and allows a compact design that completely fits into the footprint of a conventional SA. This design is a direct drop-in replacement for the existing sensing circuit in the commercial SRAM.

Chapter 3

Cache Resiliency with Architecture-level Assist Techniques[1]

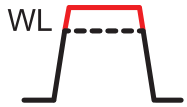
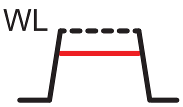
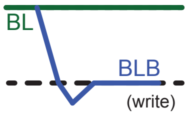
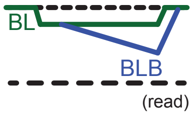
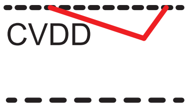
3.1 Introduction

Chapter 2 discusses that different topologies of sensing circuits can improve the read reliability at low voltages, reduce the error rate, lower the operating voltage and improve energy efficiency. The process variation of transistors not only causes the offset voltage in sense amplifiers, but also causes bitcells to suffer from worse writability and read stability. The mismatches between the identical transistors make the SRAM bitcell more volunerable to noises and the state of the bitcell could be flipped. Some weak bitcells have smaller bitline swings that can lead to read failures. The process variation also causes write failures when the pass transistor becomes weaker than the pull-up transistor and unable to pull the internal node to ground. These weak cells failed at low voltages can be overcome with circuit assist techniques. Table 3.1 shows a summary of the common circuit assist techniques that change wordline (Wordline Boosting Write Assist [14], Wordline Underdrive Read Assist [30]), bitline (Negative Bitline Write Assist [15, 16], Lower Bitline Read Assist [31]), and cell supply (Supply Collapse Write Assist [17, 18]) on a cycle-by-cycle basis to strengthen or weaken particular devices during each operation. These techniques have been shown to significantly reduce Vmin, but have their own disadvantages. The circuit-level assist techniques need to be re-evaluated for each new technology node, since the process parameters and device variations are different. Therefore, the design cycle time increases due to the redevelopment and re-evaluation of various assist techniques. Moreover, these techniques cannot be targeted only at the weak bitcells and must be enabled for all cells, which incurs high area and power overhead for all bitcells in the design. For example an 8T SRAM has a better read margin by adding a dedicated read port but 50% more area.

To avoid the SRAM Vmin to constrain the rest of the SoC logic, a separate independent voltage rail can be applied to the SRAM macros. However, this approach requires careful

[1] The content in this chapter is derived from a publication in Symp. VLSI 2018 [38]

Table 3.1: Summary of the circuit-level assist techniques for SRAMs.

Assist techniques	Wordline Boosting	Wordline Underdrive	Negative Bitline	Low Bitline	Collapse Supply
Read Stability	⇓	⇑	-	⇑	-
Write Ability	⇑	⇓	⇑	-	⇑
Related work	[14]	[30]	[15, 16]	[31]	[17, 18]
Waveform					

verification over multiple corners and distribution of two separate power rails through the chip that decreases the effectiveness of the power delivery network and requires additional voltage margins that cost power. Moreover, the SRAM macros, which take a major portion of area and power in an SoC, still have a large energy consumption at a higher voltage domain. A hybrid dual-rail [39] architecture was proposed to put the peripheral circuits into a separate domain to reduce the power. In this case, level shifters need to be inserted at the boundary of two voltage domains within the SRAM macro.

Figure 3.1 shows an alternative way of Vmin reduction. While circuit-level assist techniques lower Vmin by improving the bitcells and reducing the error rate, architecture-level assist techniques tolerate failing bitcells to allow caches to operate under a lower voltage without the errors propagating through. The architecture-level assist techniques are evaluated based on their ability to increase the maximum allowable p_{bit} fails, which is translated into voltage reduction based on the failure slope. A gradual failure slope improves the effectiveness of architecture-level techniques, as the same p_{bit} fails difference translates to a larger voltage difference. Typical SRAM arrays in modern processes exhibit a measured failure slope of around 50 mV per decade, i.e., a 50 mV reduction in VDD increases the number of failures by ten times. Therefore, a cache architecture that can tolerate a small number of faulty bits in a large array can lower the Vmin by >100 mV. The architecture-level techniques do not conflict with the circuit-level techniques, i.e., they can be appied in parallel in pursuit of the lowest Vmin.

Conventional resiliency techniques like redundancy and Error Correction Codes (ECC) are originally used for protecting SRAMs from hard defects and correcting the random soft errors, respectively. The conventional techniques can be used for the failing bits at low voltages, however, the area and timing overhead would be intolerable at the high BER under low voltages. In this section, the advantages and disadvantages of the ECC-based, redundancy-based and disable-based techniques will be reviewed.

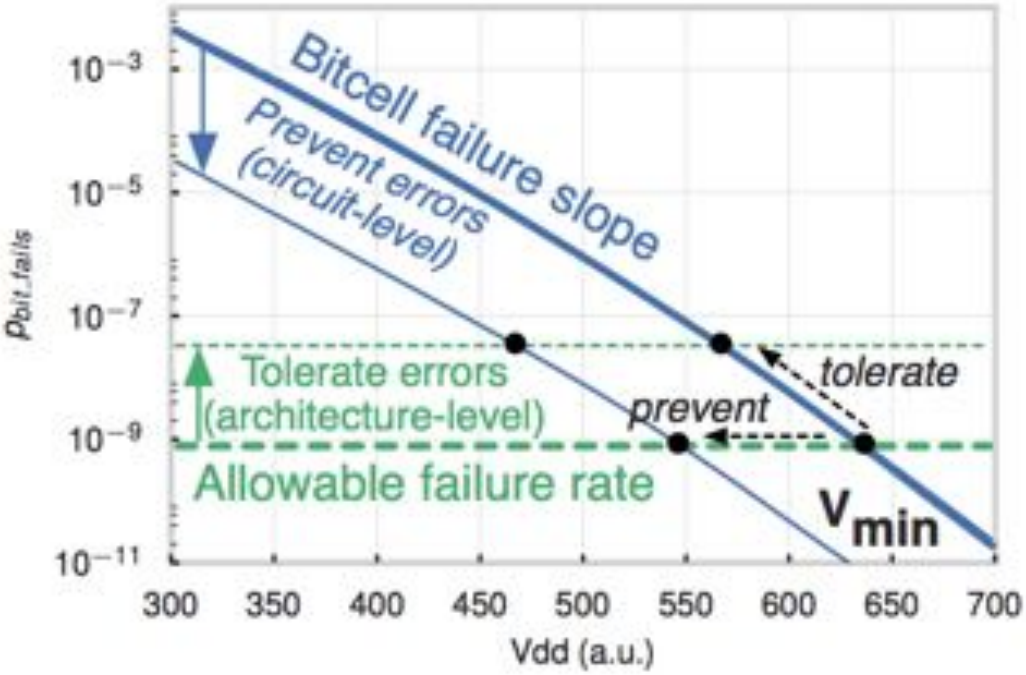


Figure 3.1: Circuit-level techniques lower Vmin by improving bitcells while architecture-level techniques lower Vmin by tolerating bitcell failures. [1]

3.1.1 ECC-based techniques

Highly reliable systems use error detection and correction codes to protect against soft errors on the fly. Soft errors in memory occur when the alpha particles from package decay or cosmic rays strike the memory cell(s) and the positive charge causes the cell to flip its state. ECC is able to detect or correct random events of soft errors with algebraic codes (BCH codes, Hamming codes) or graph-based codes (LDPC codes) by adding parity bits. ECC is also commonly used in high-error-rate nonvolatile memories, e.g. multi-level-cell (MLC) NAND flash [40]. The raw BER of MLC flash memory has increased significantly because of smaller process nodes, worse cell-to-cell interference and the small margin between each state. The limited endurance and the trend of wrapping more bits in one cell (Triple-level-cell, TLC) make multi-bit-correction ECC, such as BCH, inevitable.

However, the storage overhead, decoding and encoding of a multi-bit ECC is too expensive to be implemented to protect the SRAM-based cache from errors at low voltages. The delay for encoding and decoding of ECC adds to the short latency of cache and affect the performance. Single-Error-Correction-Double-Error-Detection (SECDED) codes such as Hsiao code [22] correct errors in words. For a 64-bit word, it requires 8-bit check bits, which causes 12.5% storage overhead. The overhead is smaller with longer word length (11-bit check bits for 512-bit word). To guarantee recovery for soft errors, SECDEC is not able to solve persistent errors at low voltages.

Double-Error-Correction-Triple-Error-Detection (DECTED) codes, such as BCH [23], can cover one persistent low-voltage error and one soft error with 22% of storage penalty for 64-bit word. Multiple-bit segmented ECC (MS-ECC) [24] sacrifices a large portion of cache (25%∼50%) to store ECC check bits, which leads to performance loss at low-voltage mode. Variable-Strength ECC (VS-ECC) [25] handles the rare multi-bit error with a small extended ECC field of 4EC5ED, which can be reconfigured to recover the multi-bit-error cache line. The rest of the cache lines are still protected from soft errors with a simple and fast SECDED. VS-ECC with 12-SECDED+4-4EC5ED in a 16-way cache requires less check bits than a full DECTED ECC solution. The technique was demonstrated with 512-bit cache line and will still cause a large overhead if it was operated on a 64-bit word.

3.1.2 Redundancy-based techniques

For known errors at fixed locations, ECC is too expensive in terms of area. Redundancy is a common scheme to fix the hard defects by replacing a faulty column/row with another redundant column/row. Column-redundancy schemes [26, 27] handle hard faults by adding spare columns (SC) to an SRAM array, with a two-way multiplexer at the bottom of each column to shift columns over to map out a failing column. Traditional static column-redundancy schemes are configured with fuses and correct one column per array. The randomness of the error location limits its effectiveness to single-bit errors spread over different columns. It takes a unrealistically large number of SC to repair all the low-voltage errors with static column redundancy.

Dynamic Column Redundancy (DCR)[2]

Dynamic Column Redundancy (DCR) [2] reconfigures the multiplexers for each set in the cache by using a redundancy address (RA) to dynamically select a faulty column. Therefore, it efficiently corrects one bit per set instead of one column per array.

Figure 3.2 describes the dynamic column redundancy technique. The DCR scheme [1] associates a redundancy address (RA) with each row of the SRAM to dynamically select a different multiplexer shift position on each access. The RA is stored inside the cache tag array, and is shared between all lines in a set. In this implementation, shifting occurs outside the array to repair one bit per logical address in a set regardless of physical interleaving. The timing overhead is small, because the RA access occurs in parallel with the tag access and the shifting operations add a single multiplexer delay to the data setup and access time. DCR offers similar resiliency to ECC, but at much lower cost. Unlike ECC, which generally requires codeword granularity to reflect access word size (to avoid read-modify-write operations), DCR granularity can be adjusted independently of access size.

The prototype repairs one bit per all 8 lines in a L2 cache set (4096 bits), requiring only a single 7-bit RA per 4096 protected bits. This technique is even more attractive for L1

[2] The content in this section is derived from a publication in JSSC 2017 [1]

caches, where ECC would require 8 checkbits for every 64 bit word, while DCR can use a 7-bit RA to protect 2048 bits. DCR enables a larger design space of resiliency versus area overhead trade-offs. ECC is generally already required for soft error tolerance, and DCR can be easily supplemented by a SEC-DED code to protect against intermittent errors. In comparison, adding soft error protection to a design that already uses single-bit ECC for voltage reduction by adding double-bit ECC is very expensive.

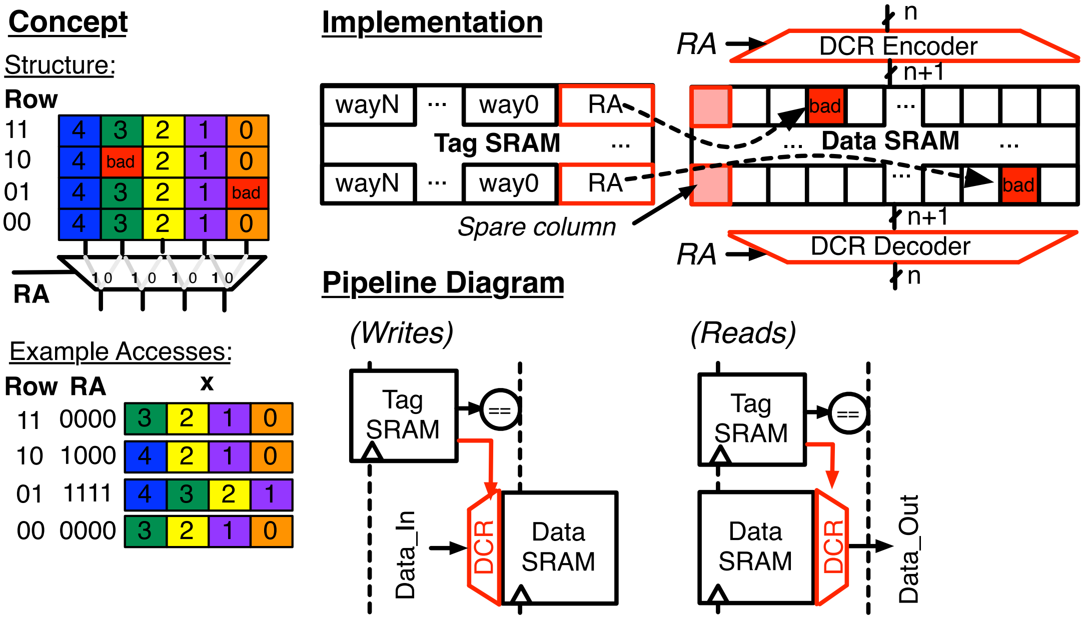


Figure 3.2: Dynamic column redundancy exploits cache architectures to repair a single bit per set in cache data arrays.

3.1.3 Disable-based techniques

Vmin is limited by a few multi-bit faults, instead of an overwhelming number of single-bit faults. Using redundancy techniques like DCR alone only repair single faulty bit in a set, One way to avoid the propagation of multi-bit errors in caches is to skip the faulty cache lines. In a set-associative cache, disabling ways in a set to avoid the errors does not harm the functionality, as long as there is one working way left. Dynamic cache line disable (LD) is a technique that takes advantage of the set associativity of caches and improve resiliency by disabling the cache lines with faulty bits. The error location can be derived from accumulative results of ECC [41] or built-in-self-test (BIST) [1].

The LD requires only an one-bit flag stored with each tag, indicating a disabled line. The area overhead is very small (one bit per way in the tag array). For example, if the tag length is 20, the 1-bit disable flag would cause 5% more bits in the tag array, which is about 3%

increase in area of the SRAM macro considering the array efficiency. The total area penalty in L2 is only 0.3% under the assumption of 1:9 tag-to-data ratio.

The only timing overhead comes from increased complexity in the way-replacement algorithm. The only time disable bits need to be checked is during refills, as the processor is not allowed to allocate data into a disabled way. A pseudo-random algorithm shown in Figure 3.3 selects the replacement way from among the enabled ways. During tag reads, the disable bit is not necessary, because the valid bit cannot be high when the line is disabled, so tag comparison will never match a disabled way.

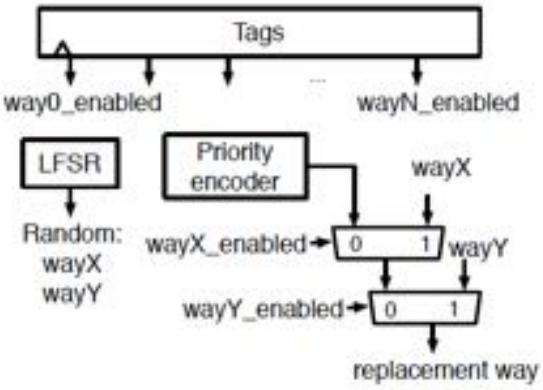


Figure 3.3: Way replacement mechanism for the disabled way.

However, the maximum number of disabled cache lines is limited in order to minimize the performance impact due to a loss of the capacity. The maximum number of disabled lines then sets the Vmin. To remove the need for analysis of the cost of increased misses due to decreased capacity, the maximum disabled cache capacity is set at 99% where only a few lines are disabled.

3.2 Line Recycling (LR)

The LD technique protects the errors from being accessed by sacrificing the faulty cache lines. Figure 3.4(a) and Figure 3.4(b) show the distribution of the number of error in a cache line or a set at 0.43V. While most of the cache lines contain only one error, 64% of sets get multi-bit errors. However, DCR can only deal with 3% of sets that have single-bit errors. The rest of the sets with multi-bit errors need to be handled by LD. The faulty cache line are disabled just to mask one or two errors, but waste 99.8% of functioning bits (one error bit in a 512b cache line).

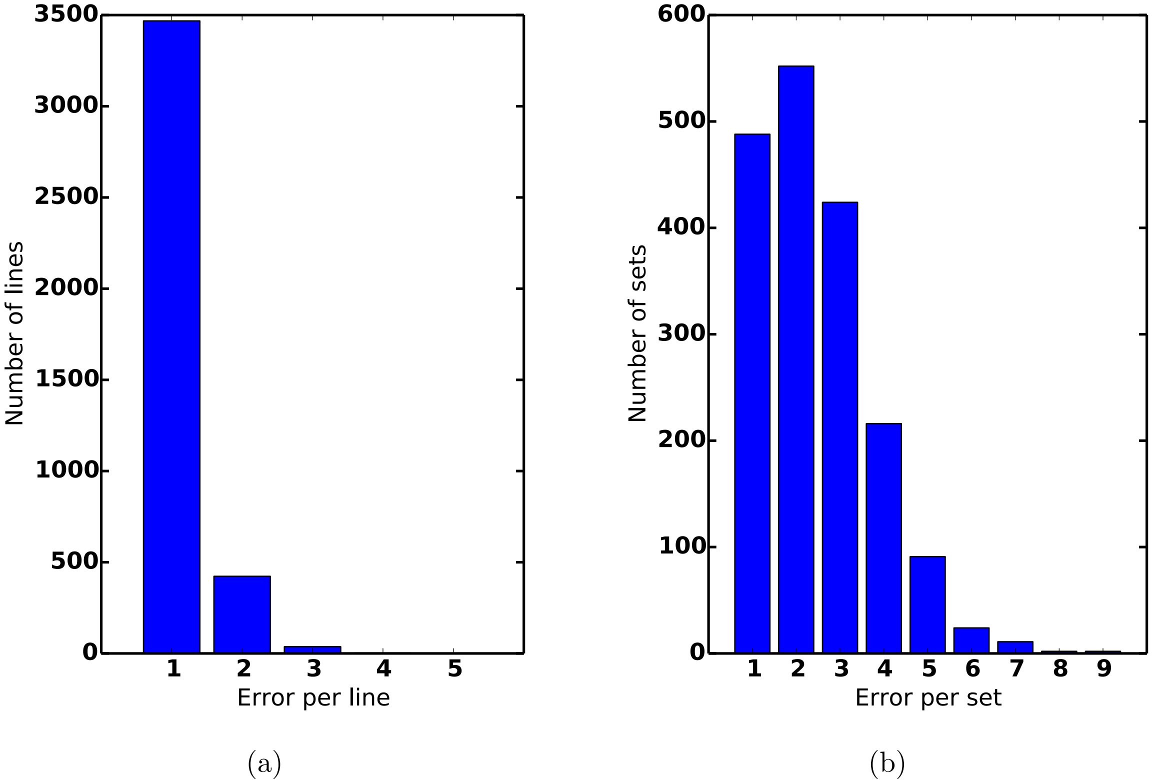


Figure 3.4: The measured distribution of the number of errors in (a) a cache line and (b) a set.

Line Recycling (LR) is proposed in this dissertation [38] to reuse two disabled faulty cache lines to repair a third faulty line. Instead of simply disabling the faulty cache lines, they can be recycled and used to minimize the capacity loss. Figure 3.5 demonstrates the LR technique. Since the probability of more than one error at the same column for three single-bit-error cache line is only 0.2%, three faulty cache lines with no errors at the same column can be easily grouped together and recycled as one one functional cache line. Two lines (P1 and P2) marked as disabled are used store data identical to that in the recycled line (line R). Therefore, the recycled line owns three copies of data and is able to recover from the errors by making a majority vote since no more than one error will be at the same bit location.

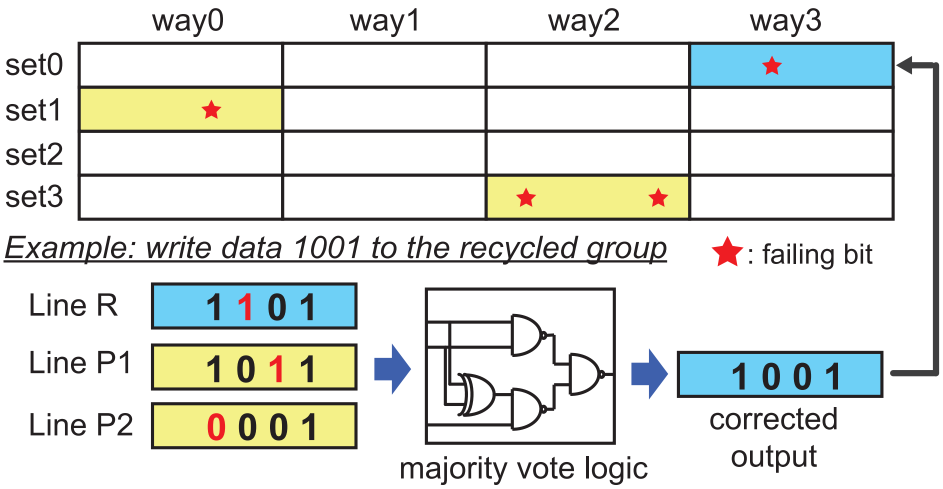


Figure 3.5: The concept of line recycling: reusing two disabled cache lines to repair a third one via majority vote.

3.2.1 Implementation

Figure 3.6 shows the block diagram of LR. Each entry in the recycle index table (RIT) indicates whether the accessed cache line is a recycled line and contains the recycle index. The recycle index selects an entry in the patch address table (PAT) containing the addresses for line P1 and P2. The patch lines are read and written in subsequent cycles after the patch addresses are acquired. To fix errors in the recycled line, the correct value is determined by a majority vote of the recycled line and two patch lines. Since no column has more than one error, the correct value prevails in the majority vote circuit. Since the number of disabled lines in L1 cache is less than that in L2 cache, the area overhead of implenting LR in L1 cache is intolerable. Moreover, the additional cycles for patch line access is comparable to L1 miss penalty, which hides the benefit of LR. Therefore, LR is only adopted in L2 cache in the test chip.

Table 3.2: Summary of the SRAM macros in the L2 system.

name	size	port	number of macros	area (umˆ2)	percentage in L2
DataArray	4096x73	1	32	2043496.61	84.62%
MetadataArray	512x57	2	16	258994.46	10.72%
RecycleIndexArray	512x16	1	4	12229.82	0.51%
PatchAddressArray	256x28	2	1	5646.12	0.23%

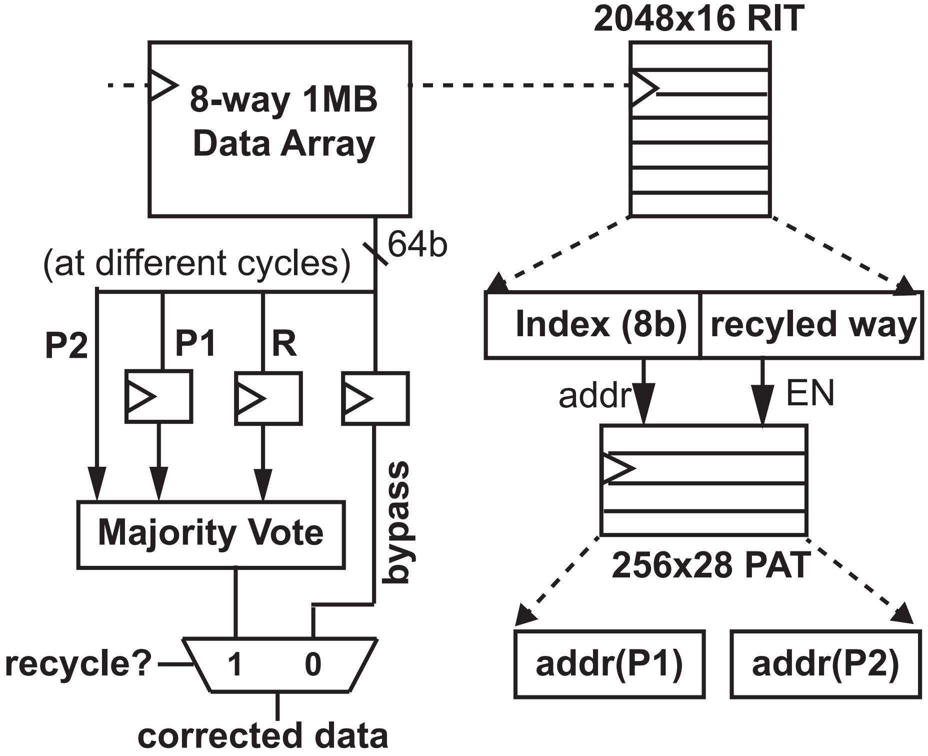


Figure 3.6: Block diagram of the implementation of line recycling.

LR has 33% smaller cache capacity loss than LD for the same BER, reducing degradation of compute performance. Access to recycled lines takes only three additional cycles to complete, while each additional L2 cache miss caused by reduced capacity could result in over a hundred cycles of miss penalty to access memory. Table 3.2 lists all the SRAM macros used in the L2 system. The area overhead in the L2 system from the tables (4KB RIT and 0.875KB PAT), registers, and majority vote circuit is only 0.77%. Figure 3.7 shows the number of entries of PAT required for different BER. This test chip chose 256 entries of PAT to reduce the area overhead. 256 out of 768 faulty cache lines can be recovered. The faulty cache lines will be disabled if the PAT is full.

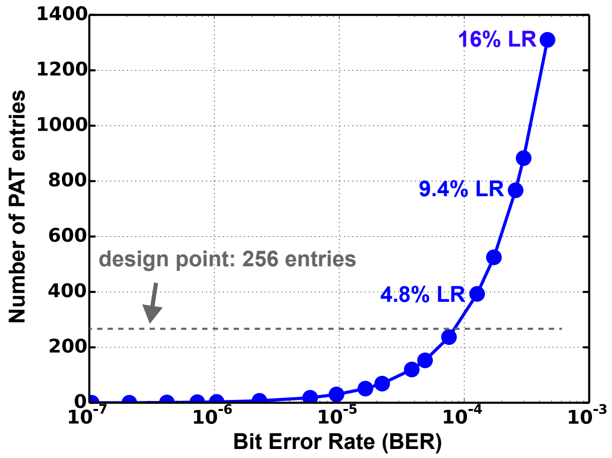


Figure 3.7: The number of PAT entries required for different BER.

3.3 Bit Bypass with SRAM implementation (BB-S)

The resiliency for the tag arrays in caches is extremely important since it guards the data arrays. Moreover, DCR and LR both rely on tag arrays to store the reprogrammable redundancy information for the data arrays. Therefore, additional protection is needed for the tag arrays. Bit Bypass (BB), as shown in Figure 3.8, uses flip-flops to form entries that log error locations and the repair bits. If the address matches any of the logged row location in the BB table, the error bit at the logged column location would be replaced by the repair bit. BB provides the ability to fix two bits for every entry. However, the number of entries in the BB table is limited in order to minimize the area overhead. Line disable needs to take action when more than 2 errors at the same row or the number of faulty rows exceeds the number of BB entries. It suppresses the effectiveness of LD for fixing the data array.

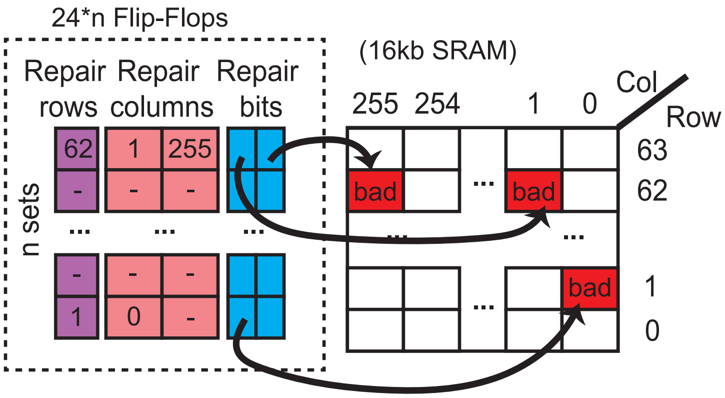


Figure 3.8: Block diagram of Bit Bypass (BB).

The standard cell of a D flip-flop is 6 times larger than an 8T SRAM bitcell (the SRAM macro used for the tag array), which results in a large area overhead for BB. The address decoder implemented with standard cells also takes way more area than the compact peripheral circuit in SRAM macros.

To provide a better resiliency under the same area constraint, the proposed BB-S (Figure 3.9) expands the tag arrays for logging error locations. In this case, every row has the ability to repair two error bits. Moreover, it saves the effort of searching through the BB table to find a match entry. BB-S contains a lot more repair entries (one for each row) but has less area overhead due to shorter width per entry (no need to log the row), smaller cell size, and compact periphery circuit.

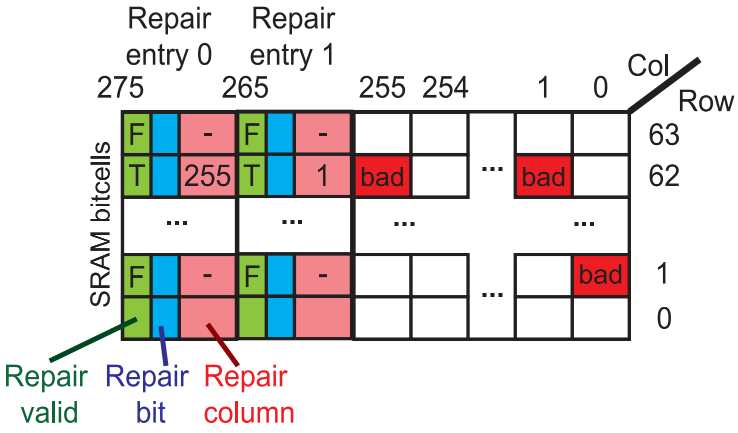


Figure 3.9: Block diagram of Bit Bypass with SRAM implementation (BB-S).

Although the SRAM bit cell is much smaller than a flip-flop, it is more volunerable than flip-flops at a low voltage. However, BB-S is error-tolerant even when the error is at the BB entries. Figure 3.10 shows four cases when the errors are at different locations. Case (a) is the normal case when the error(s) happens to be in the tag. Case (b) is when any bit in the error bit locator (COL) is failing. It results in an unnecessary but unharmful replacement with the repair bit. The valid bit can also be set to 0 to deactivate this entry. Case (c) is when the valid bit is failing. If the valid bit is stuck at 0 (case c1), the BB entry is simply deactivated. If the valid bit is stuck at 1 (case c2), the repair bit will replace one correct bit that does not affect the functionality. When the error is at the repair bit as in Case (d), the valid bit is set to 0 to deactivate the entry and prevent the bit from being replaced by the faulty bit. The second entry can still repair one error in the tag. The cases demonstrate that BB-S can fix up to two errors no matter the error is in the tag or in the BB entries.

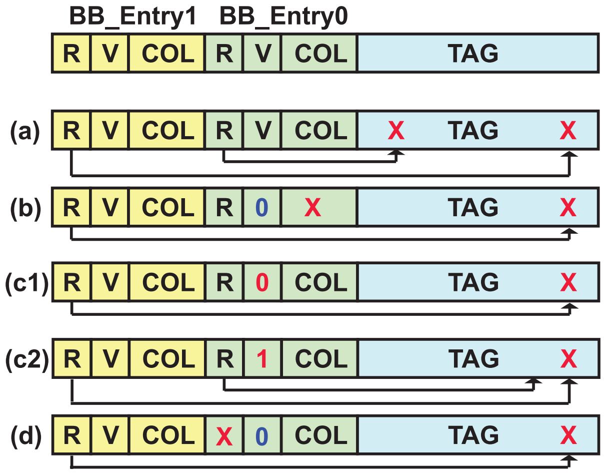


Figure 3.10: Four cases when the error bits are at different locations. (R: Repair bit, V: Valid bit, COL: Error column location).

3.3.1 Implementation

The block diagram for the implementation of BB-S is illustrated in Figure 3.11. The metadata array is composed of 8T SRAM macros with one write port and one read port. When a write operation is performed in the metadata array at cycle 0 (C0), the write mask is set to only allow to write in the designated way. In the mean time, the BB entries are read. The multiplexers select the value to write to the repair bits according to the logged column location. The second write to the metadata array at cycle 1 (C1) happens only when the previous write operation accessed to the error location and the BB entry is valid. The repair bits is then updated with the value of the new tag. The read access to the metadata array retrieves BB entries and n-ways of tags. The output of the n-way tags needs to go through two level of multiplexers to derive the correct data. One BB entry controls one level of multiplexers. The valid bit and the logged column location trigger the multiplexer to replace the faulty bit with the repair bit.

The extended columns of the SRAM macros, multiplexer and the logic to decode the error location have an area overhead of 8.6% in the tag array. BB-S has a lower area overhead than the previous BB (10%) implemented by flip-flops and is able to fix more errors. The decoding for error location and the 2-level multiplexer add some gate delays to the timing path. However, The tag access is usually not the critical path due to smaller SRAM macros. The cycle time is not affected by BB-S.

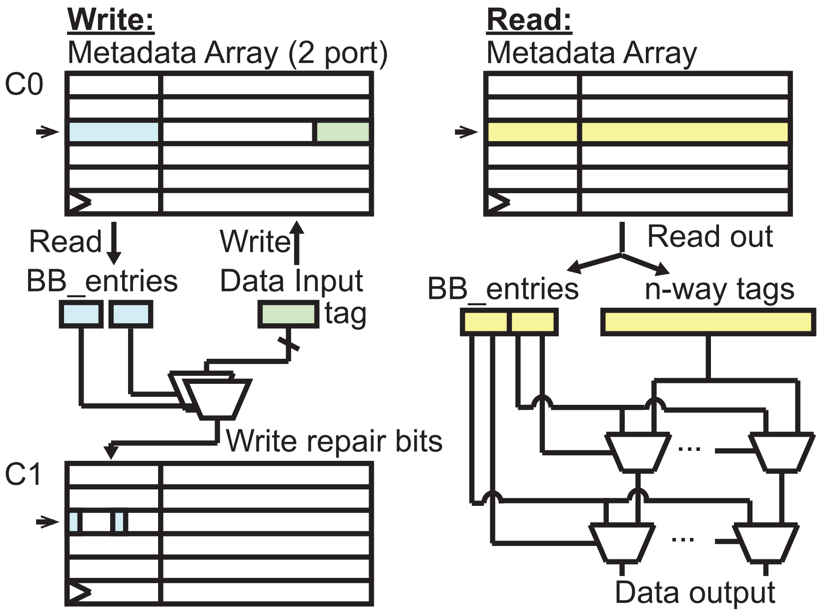


Figure 3.11: Block diagram of the implementation of BB-S.

3.4 Error Model[3]

This section use a generic framework developed in the previous work [1] that calculates cache yield as a function of bitcell error rate (or equivalently, voltage) to compare the effectiveness of the proposed schemes under a common set of assumptions. This generic framework uses a hierarchy of binomial distributions defined in Equations 3.1-3.8 to translate bitcell failure probability to system failure probability for the cache structure described in Figure 3.12. The distribution of failures in each level of the hierarchy can be represented by a binomial sample of the level below. For example, the number of failing lines in a set is a binomial sample of n total lines in a set with a given probability p of line failure. The probability that a given level in the hierarchy fails can be determined by evaluating the cumulative density function (CDF) of that level. For example, the probability that a set does not fail is the CDF evaluated at 0 (assuming no lines can fail in a set).

[3] The content in this section is derived from a publication in JSSC 2017 [1].

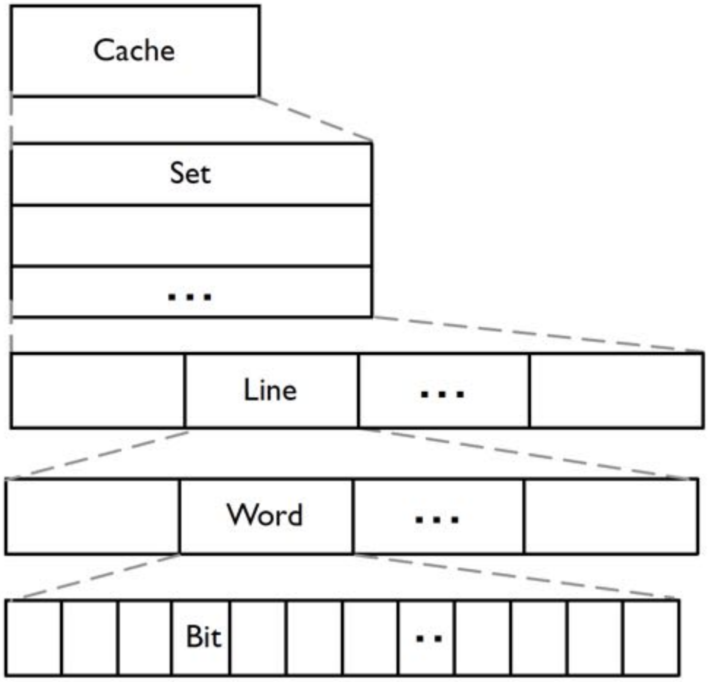


Figure 3.12: Description of the notation used in the error model framework that translates bitcell failure rate to system yield.

While using a binomial distribution to translate failure probabilities is common practice, new error model introduces a hierarchical combination of these distributions in a configuration that enables the evaluation of many different schemes by simply changing a few parameters. For example, SECDED is described by the parameters $a_{bw} = 1$, $a_{wl} = 0$, $a_{ls} = 0$, $a_{sc} = 0$, which represent the idea that one bit in every word can fail, no words can fail in a line, no lines can fail in a set, and no sets can fail in a cache. Vmin is determined by finding which p_{bit_fails} corresponds to a target probability of cache failure (p_{cache_fails}). These two different p_{cache_fails} are especially relevant, 0.5 and $1{\times}10^3$, where the former reflects the voltage at which an average cache fails, and the latter reflects the voltage at which 99.9% of all caches work. Both assumptions account for the probability that a collection of all bitcells with the same probability of failure at the same voltage create a condition that causes chip failure, but do not account for global process skew, which will cause the same p_{bit_fails} to occur at different voltages.

$$
\begin{aligned}
p_{bit\_fails} &= \text{Defined for given } V_{DD} \\
X_{word} &\sim Binomial(n_{b-w}, p_{bit\_fails}) && (3.1) \\
p_{word\_fails} &= 1 - \mathbb{P}(X_{word} \leq a_{b-w}) && (3.2) \\
X_{line} &\sim Binomial(n_{w-l}, p_{word\_fails}) && (3.3) \\
p_{line\_fails} &= 1 - \mathbb{P}(X_{line} \leq a_{w-l}) && (3.4) \\
X_{set} &\sim Binomial(n_{l-s}, p_{line\_fails}) && (3.5) \\
p_{set\_fails} &= 1 - \mathbb{P}(X_{set} \leq a_{l-s}) && (3.6) \\
X_{cache} &\sim Binomial(n_{s-c}, p_{set\_fails}) && (3.7) \\
p_{cache\_fails} &= 1 - \mathbb{P}(X_{cache} \leq a_{s-c}) && (3.8)
\end{aligned}
$$

$$
\text{where} \quad
\begin{aligned}
a_{b-w} &= \text{Allowable number of bit failures in word} \\
a_{w-l} &= \text{Allowable number of word failures in line} \\
a_{l-s} &= \text{Allowable number of line failures in set} \\
a_{s-c} &= \text{Allowable number of set failures in cache} \\
n_{b-w} &= \text{Number of bits in word} \\
n_{w-l} &= \text{Number of words in line} \\
n_{l-s} &= \text{Number of lines in set} \\
n_{s-c} &= \text{Number of sets in cache}
\end{aligned}
$$

3.5 Resilient Out-of-Order processor

The techniques of cache resiliency mentioned above are integrated in the Berkeley Out-of-Order processor (BOOM) [42]. Out-of-order processors have higher performance and burn more power than the in-order processors. The cache resiliency techniques enable BOOM to operate under lower voltages and achieve both high performance and better energy efficiency by operating in a wider voltage range. Moreover, BOOM enables instruction reordering, which hides the miss penalty by executing some independent instructions first while waiting for memory accesses. Therefore, it alleviates the performance loss caused by the reduced cache capacity.

The Berkeley Resilient Out-of-Order Machine (BROOM) chip is based on Berkeley's RISC-V Rocket Chip SoC generator [43]. By changing the configurations, we can easily create the target system with the parameterized chip-building libraries, including cores (in-order/out-of-order), caches, interconnections, and perypherals. To add testability and programmability to the chip, the system control registers (SCR) map different control registers to different memory spaces. Writing to the SCR file reprograms the settings of the resiliency schemes and reading from the SCR file peeks in some internal states. Details of physical implementation are discussed in the next chapter.

The block diagram and the layout of BROOM is shown in Figure 3.13. The BROOM chip has one BOOM core on its own clock and voltage domain and 1MB L2 cache on the uncore clock and voltage domain. To reduce the number of I/O pin count, a bidirectional SERDES module translates TileLink transactions to 8-bit serial IO.

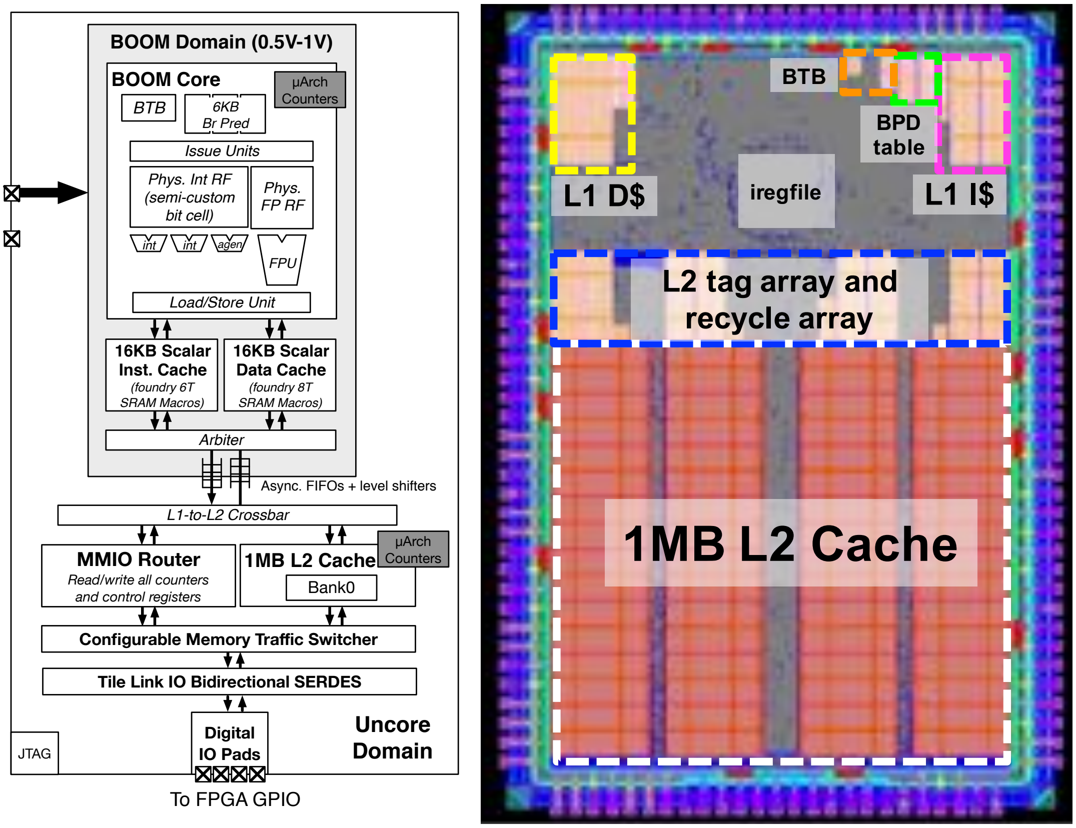

BOOM Domain (0.5V-1V)
BOOM Core
μArch Counters
BTB
6KB Br Pred
Issue Units
Phys. Int RF (semi-custom bit cell)
Phys. FP RF
int
int
agen
FPU
Load/Store Unit
16KB Scalar Inst. Cache (foundry 6T SRAM Macros)
16KB Scalar Data Cache (foundry 8T SRAM Macros)
Arbiter
Async. FIFOs + level shifters
VDDCORE
VDD
L1-to-L2 Crossbar
MMIO Router
Read/write all counters and control registers
1MB L2 Cache
μArch Counters
Bank0
Configurable Memory Traffic Switcher
Tile Link IO Bidirectional SERDES
JTAG
Digital IO Pads
Uncore Domain
To FPGA GPIO
BTB
BPD table
L1 D$
iregfile
L1 I$
L2 tag array and recycle array
1MB L2 Cache

Figure 3.13: Block diagram and layout of BROOM.

3.5.1 Berkeley Out-of-Order Machine (BOOM)

BOOM, an open-source out-of-order processor, implements the RV64G RISC-V Instruction Set Architecture (ISA) with Sv39 supervisor support and includes a 3-stage instruction fetch unit; a set-associative branch target buffer (BTB) and SRAM-based conditional branch predictor; split floating-point and integer register files; a dedicated floating-point pipeline; separate issue queues for floating-point, integer, and memory micro-operations; and separate stages for issue-select and register read (Figure 3.14). The chosen two-wide, nine-stage pipeline micro-architecture can issue up to four micro-ops in a single cycle (two integer, one FMA, and one memory operation), requiring a 6-read, 3-write (6R3W) integer register file and a 3-read, 2-write (3R2W) floating-point register file. The processor core is embedded in a tile with a memory hierarchy, including L1 and L2 caches, and standard external interfaces.

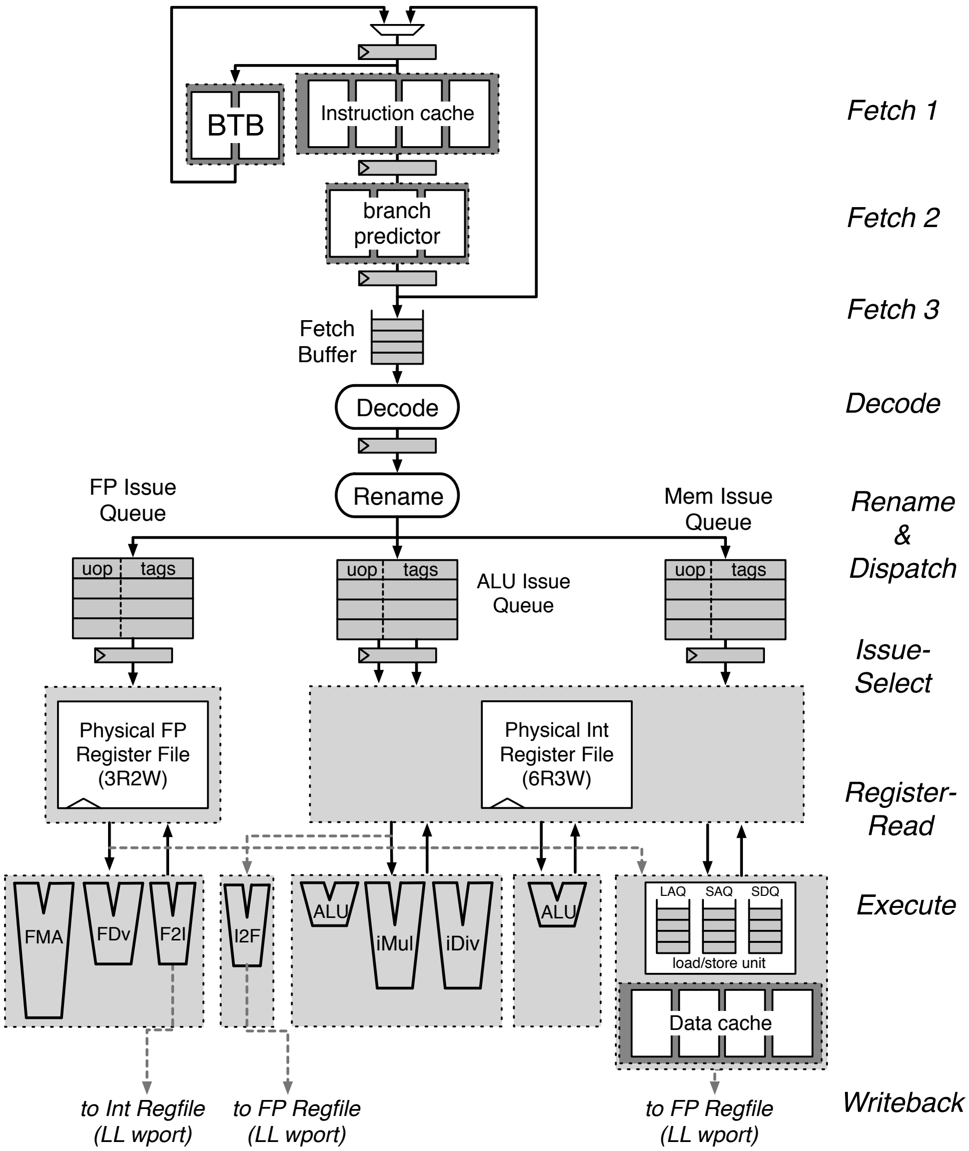


Figure 3.14: The nine-stage pipeline BOOM.

The Chisel hardware construction language [44] was used for the design, which enables rapid design-space exploration by providing features of modern programming languages, yet guarantees a synthesizable design. The Chisel design was mapped to standard cells and foundry-provided memory compilers, and optimized for performance and energy efficiency. The main challenge in achieving high performance and energy efficiency from a synthesized processor design is the implementation of fast multiport register files, which are generally not supported by standard foundry offerings of single-and dual-ported memory compilers.

3.5.2 Implementation of the Resiliency Techniques

The resiliency schemes implemented in the BROOM chip is shown in Figure 3.15. After the fault locations are detected by built-in-self-test (BIST), A set of SCR files are used to reprogram the resiliency information accordingly. The 16KB instruction cache and data cache are protected by BB-S (tag array), DCR, and LD (data array). The 1MB L2 cache is protected by BB-S (tag array), DCR, LD, and LR (data array).

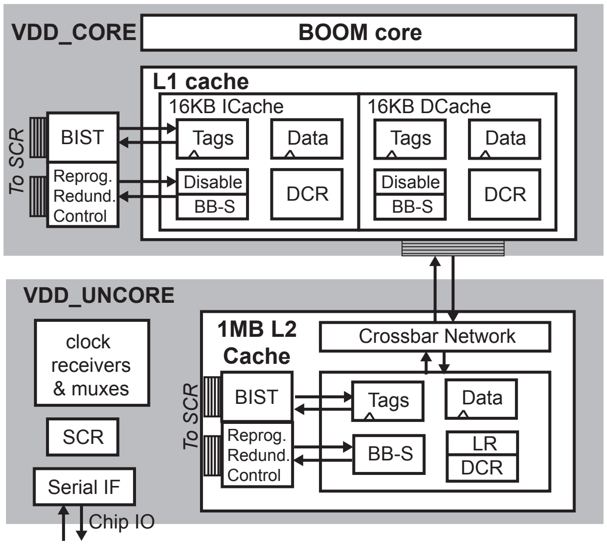


Figure 3.15: Resiliency schemes implemented in BROOM.

Built-In Self Test (BIST)

In order to detect the location of the failing cells in the memory array, the built-in-self-test (BIST) is performed before programming the resiliency. The BIST infrastructure was carried over from the previous version, which was partially implemented in Verilog. To make the BIST module reusable and easier to integrate in the Rocket Chip, it was re-written with Chisel as a parameterized library. The BIST control state machine is shown in Figure 3.16. There are six programmable March elements (TEST0 through TEST5) that can be programmed to be a write or read operation. The extra SAFEWRITE and SAFEREAD states operates at a high voltage to rule out different failure modes.

When an error is detected, the machine moves to the HANDLE ERROR state, which allows the off-chip host to retrieve the error entry from the tested SRAM. After the host received the error entry (set entry_received=1), the flags are cleared and the state goes back the next state of the one that detected the error.

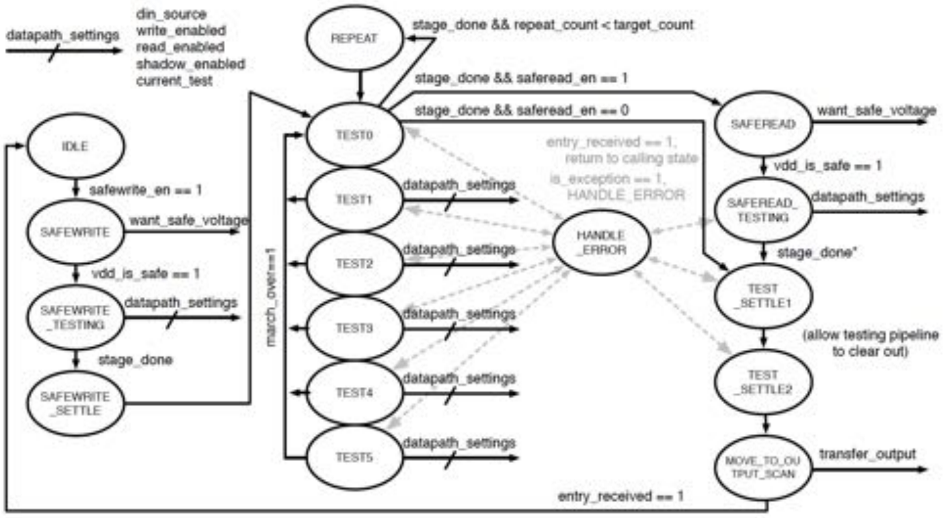


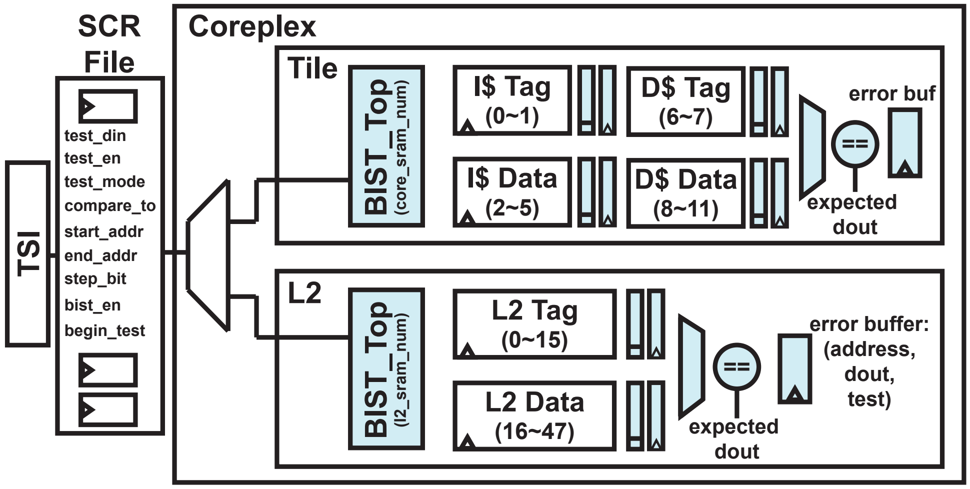
Figure 3.16: The BIST control state machine allows a wide variety of programmable March tests. [2]

The block diagram of the BIST data path is shown in Figure 3.17. The core and L2 have separate BIST control circuit to prevent the timing problem crossing two voltage domain. The BIST module is parameterized with the number of SRAM macros to instantiate different number of enable signals, latch, registers. Therefore, it's easy to have multiple instances of BIST with different number of SRAM macros under test.

The March test settings are programmed through the SCR file, which is able to control the write/comparison data, read/write operation, start/end address of the test, the stride of the test, and the signal to begin the test. In a read test, output is first latched by transparent-high latches before being sent to a normal positive-edge flip flop to make sure the SRAM test is on the critical path. The BIST only tests one macro at a time. The output of the tested macro is selected by the multiplexer and compared with the expected output. The error buffer then logs the address of the access that failed, the entire output data word, and the current test number. The error entry is then accessed off-chip through SCR. Table 3.3 summarizes the list of SRAMs that are testable by BIST. The 73-bit data width includes 64-bit data, 8-bit ECC parity bits and 1-bit redundancy. Tags are wider than 73 bits, therefore, are split into separate physical SRAM macros.

Table 3.3: Summary of the testable SRAMs in the chip, with corresponding size and BIST macro address.

Structure	Size	SRAM name	SRAM #	SRAM Size
L1 ICache Tags	64x111 (888B)	BOOMTile.icache.icache.L1I_tag_array.L1I_tag_array_ext.sram[0,1]	0, 1	64x56
L1 ICache Data (4 ways)	16KB (18688B)	BOOMTile.icache.icache.I1I_data_array[0-3].L1I_data_array_ext.sram0	2∼5	512x73
L1 DCache Tags	64x140 (1120B)	BOOMTile.HellaCache.meta.L1D_tag_array.L1D_tag_array_ext.sram[0,1]	6,7	64x70
L1 DCache Data (4 ways)	16KB (18688B)	BOOMTile.HellaCache.data.L1D_data_array[0-3].L1D_tag_array_ext.sram0	8∼11	512x73
L2 Cache Tag	2048x226 (58368B)	L2HellaCacheBank.meta.L2Bank_tag_array.L2Bank_tag_array_ext.sram[0-15]	0∼15	512x57
L2 Cache Data	1MB (1168KB)	L2HellaCacheBank.data.L2Bank_data_array.L2Bank_data_array_ext.sram[0-31]	16∼47	4096x73

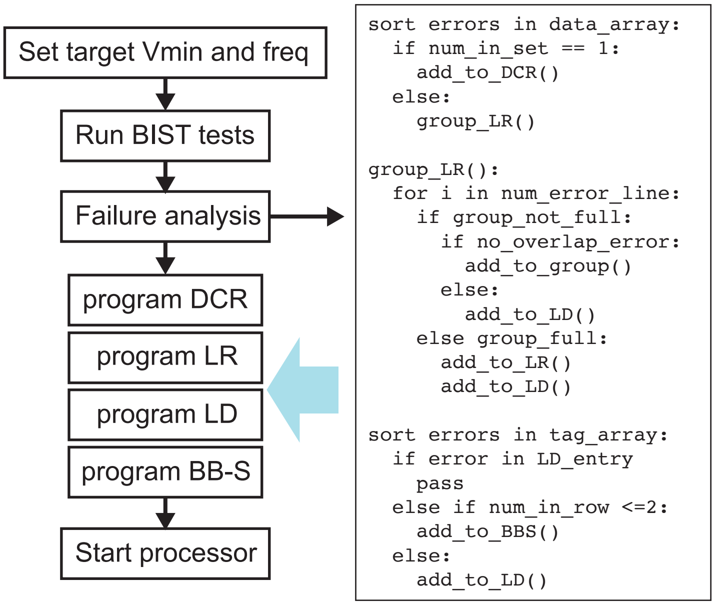


*: Blue highlighted blocks are included in BIST_Top

Figure 3.17: Data path of the BIST in the memory system.

Programming the resiliency techniques

The procedure of reprogramming the resiliency techniques is illustrated in Figure 3.18. The BIST is first running to get the failure report at a target Vmin and frequency. The failure report is then analyzed off-chip. The failure analysis sorts the errors with respect to the location and generate the lists of DCR/LD/LR/BB-S programming entries. The pseudo codes describe the tasks of the failure analysis. For errors in the data arrays, if there is only one error in the same set, it can be solved by DCR. Otherwise, the failing cache lines are divided into groups of 3 that has no overlapped error. The two patch lines in the groups are added to the LD programming entries and the recycled lines in the groups are added to the LR programming entries. For errors in the tag arrays, there is no need to fix the error with BB-S, if the error happens to be at the tag of the disabled way. Otherwise, BB-S can fix the rows with at most 2 errors, while the LD is able to deal with the rows with more than 2 errors. After all the resiliency techniques are reprogrammed, the processor can start operating without being affected by the errors at Vmin. The resiliency programming entries can be stored in a nonvolatile memory to save the time of testing and fault analyzing on every boot. The system can be set to run BIST test and reprogram the resiliency entries once in a while to update some aged or worn-out cells.



```
sort errors in data_array:
  if num_in_set == 1:
    add_to_DCR()
  else:
    group_LR()

group_LR():
  for i in num_error_line:
    if group_not_full:
      if no_overlap_error:
        add_to_group()
      else:
        add_to_LD()
    else group_full:
      add_to_LR()
      add_to_LD()

sort errors in tag_array:
  if error in LD_entry
    pass
  else if num_in_row <=2:
    add_to_BBS()
  else:
    add_to_LD()
```

Figure 3.18: Flow chart of reprogramming the resiliency techniques and the pseudo codes of the failure analysis.

ECC

BB, DCR, LD, and LR enables Vmin reduction by avoiding the cells with hard faults at low voltages. However, those techniques are not able to handle the soft errors or intermittent errors since the random events cannot be detected by the BIST. SECDED [45] is applied to the tag arrays and the data arrays to correct the intermittent errors. SECDED has the ability to fix a single-bit error and detect double-bit errors.

Applying the SECDED adds long encoding and decoding delays to the critical path. In general, the decoding delays can be alleviated by pipelining it to the following cycles. In the BROOM chip, one of the goals is to evaluate the performance of BOOM. Therefore, the decoding operation of SECDED is decoupled from the critical path to minimize the impact on the performance. Figure 3.19 shows the data path in the L2 cache. The real data path takes the uncorrected data and the decoding operation is executed off the critical path. The address and the syndrome of the error can be accessed through the SCR File to compare with the error patterns derived from the BIST. Although the decoding of SECDED is removed from the data path, both DCR and LR are adding multiplexer delays to the critical path in outputting data.

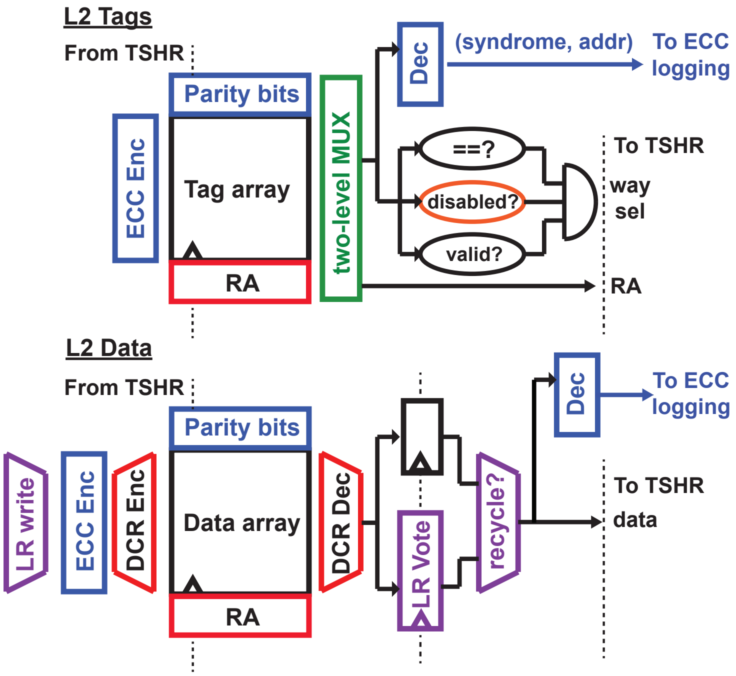


Figure 3.19: Block diagram of the data path in the L2 cache. Blue blocks highlight the ECC overheads. Red blocks highlight the DCR overheads. Purple blocks highlight the LR overheads.

3.6 Conclusion

Other than circuit-level assist techniques, the cache resiliency can be achieved by applying architecture-level assist techniques. The line recycling (LR) technique is proposed to reuse faulty cache lines that fail at low voltages to correct errors with only 0.77% L2 area overhead. LR can save 33& of cache capacity loss from line disable (LD) or allow further reduction in Vmin. BB-S implements BB with extended SRAM capacity to reduce the area overhead and has more error entries to handle multi-bit errors. Various cache resiliency techniques, including LD, LR, DCR, and BB-S, are integrated on an open-source out-of-order superscalar processor that implements the 64-bit RISC-V instruction set architecture (ISA).

Chapter 4

Implementation of Resilient Out-of-Order Processor

This chapter describes the flow of implementing a RISC-V processor utilizing Chisel and RocketChip SoC generator. The flow describes the tapeout process from building/modifying the design in Chisel to generating the verified Graphic Data System (GDSII) file for fabrication. With BROOM as an example, one special case is the implementation of the multiport register file in the high-performance out-of-order processor. Chisel's blackbox feature allows us to integrate the customized gate-level description in Chisel and optimize the area and performance of the register file.

4.1 Overview

Hardware development has been notoriously viewed more difficult than software development in terms of the length of design cycle and the required manpower and resources. The Agile hardware design is an ongoing evolution that expedite hardware development process by building a framework with reusable hardware libraries and physical design methodology. More innovations on hardware can be enabled with the help of agile hardware design.

Chisel hardware construction language and RocketChip SoC generator are two good practices of agile hardware design. Chisel enables rapid design-space exploration by providing features of modern programming languages, yet guarantees a synthesizable design. RocketChip is written with Chisel and includes extensive libraries to enable a plug-and-play design environment. For example, BOOM leverages the existing RocketChip SoC generator by replacing the in-order RocketTile with a BOOMTile. The OoO processor is highly parametrized with varying configurations: fetch width, issue width, the mix and number of functional units, the size and set-associativity of caches and BTB, sizing parameters to the branch predictors, size of the reorder buffer (ROB), and size of the register files for integer and floating point. The parametrization of BOOM allows us to optimize the design for different system requirement of timing, area, computing capability and prediction accuracy.

The infrastructure of RocketChip and RISC-V tool suite make verification over various benchmarks easier. The generator not only produce the RTL of the processor, but also the test harness, the information of different time domains and memory configurations. The generated configuration files can be easily transformed to constraint files fed to the physical design tools.

Figure 4.1 illustrates the complete chip deisgn flow from Chisel design description to chip fabrication. To guarantee the performance and functionality of the chip, verification and physical implementation are as critical as the design itself. The verification of the functionality needs to be done for every step of implementation (behavioral model, systhesis and place-and-route) to make sure the implemented design does not distort the original function. The physical verification includes design rule check (DRC), layout versus schematic (LVS), electrical rule check (ERC), and antenna checks. A clean result from the physical verification confirms the design meets various criteria for fabrication. Finally, the static timing analysis (STA) needs to be performed to get the specifications for timing and power.

The physical implementation consists of multiple steps: synthesis, I/O planning, floor planning, power planning, placement, physical placement optimization, clock tree synthesis (CTS), post-CTS optimization, signal routing, routing optimization and chip finishing. The detail of the steps will be discussed in the following sections. Although the physical implementation is not completely ”agile” so far, many scripts have been developed to make the task easier. Multiple iterations of the physical design are normally required to improve the Quality of Results (QoR).

4.2 Preparing the design

Before starting implementing the design, many constraint files need to be prepared to inform the tools the specific ways of synthesizing and arranging the design.

- Library setting: linking the tool to the paths of standard cell library and technology files.

- Power domain specification: Unified Power Format (UPF) file for Synopsys or Common Power Format (CPF) file for Cadence to specify power domains and the template for planning power mesh.

- Floorplan annotation: scripts for relative or absolute placement for SRAM macros and other hard macros.

- Pad frame instantiation and placement: instantiating the corresponding IO cells (digital, analog, power, ground) for the ports from the foundry-provided IO library and arranging the locations for the IOs, the wirebond pads, and the terminals.

- Timing constraint: specifying the targeted clock cycle, and different clock groups.

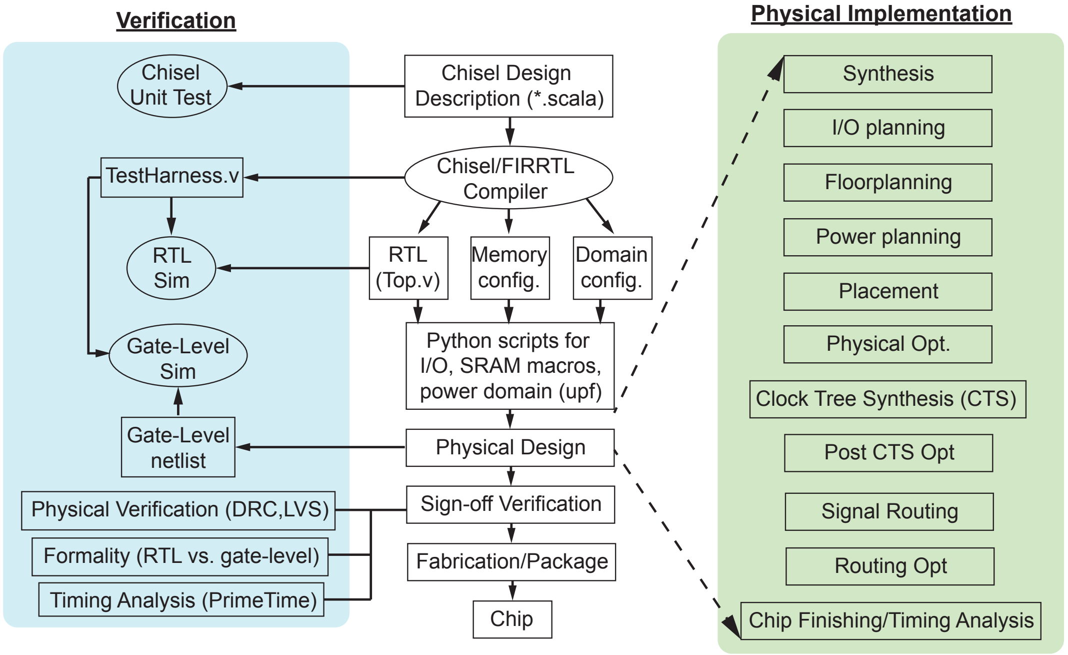


Figure 4.1: The flow of chip implementation from Chisel to fabrication. Verification and physical design are both essential to a working chip.

- Multicorner-Multimode (MCMM) scenarios: setting up various operating conditions for corners, temperatures and voltages.

Many of the scripts mentioned above, like UPF file, macro placement, and pad frame instantiation and placement, can be generated by python scripts post-processing the configuration files created by Chisel and FIRRTL.

4.3 Synthesis

Logic synthesis transforms the behavioral description in register transfer level (RTL) to the design expressed in logic gates (gate-level netlist). The retiming step in synthesis is critical to achieve the targeted performance by moving registers and latches to balance the cycle time. The BROOM chip heavily depends on retiming to implementing multi-cycle execution unit. Therefore, it is crucial to recognize the paths that need to be retimed. The retimed blocks in the BROOM chip include the rename stage, multiple floating point instances, multiplication, branch prediction stage and ALU. Flattening level(s) of hierarchy could also help for optimization. The topographical mode in Synopsys-specific Design Compiler is able to better analyze timing, area, and power with the floorplan information. It ensures better correlation between synthesis and place-and-route. Finally, after sysnthesis is done, formal verification and get-level simulation are required to verify the synthesized design match the behavioral model. Also, make sure the level shifters are inserted between the power domains.

4.4 Placement and routing

After synthesis, the gate-level design is imported to the place-and-route tool. For the BROOM chip, Synopsys's IC Compiler (ICC) is used. The tool performs different steps of pysical design implementation (Figure4.1) to place the hard macros and the logic cells, create the clock tree and route the signals and power/ground. Before the final sign-off step, the tool can run preliminary DRC/LVS on the completed design to check for opens and shorts. It would save a lot of time to fix the LVS error in the later step. If there are errors with a pattern, scripting it to find the pattern and fix the same error all at once would be a better practice than manually going through the error one by one. Finally, the GDSII-format file of the layout is generated and handed off for signing off.

The floorplanning step requires the most attention to instruct the tool the desired arrange. A good floorplan not only improves the performance of the chip but also takes less time for the tool to converge. A flawed floorplan might require many iterations to fix the DRC error and approach the targeted performance. The following items are the critical tasks in the floorplanning.

- Floorplan initialization: specifying the dimension of the chip.
- IO pad placement: the tool places the IO pads according to the constraint file generated earlier. The wirebond pads and the terminals are placed relatively to the IO position.
- Macro placement: the gaps between macros and the pin directions should be considered in this step to ensure that there are enough routing tracks between macros and the routing path is not blocked.
- Adding well taps: well taps are the standard cell that connects to the bulk. The design rule should indicate the requirement of minimum distance of well taps.
- Coarse placement and filler insertion: The M1 (or M2) layer for power/ground of the standard cell row can be created after the chip is fully populated with cells and filler.
- Routing analog signals: any analog signal that requires special rules can be routed manually and set as "don't touch" in case it gets overwritten later in the routing step.
- Planning power straps: making sure the width and number of wires and vias are enough to carry the current and prevent serious IR drops.

4.5 Sign-off

After the placement and routing of the design, various verification procedures are required before taping out. Sign-off step includes a collection of checks to evaluate the performance and variation of the post-implementation design and make sure it does not defy the rules for fabrication. The verification steps are listed as the following.

- Design Rule Check (DRC): The physical layout of a chip layout should comply with the design rules from the foundry that list all the geometric and connectivity restrictions.

- Layout Versus Schematic check (LVS): LVS transforms the drawn shapes of the layout to the electrical components of the circuit and compares it with the ideal netlist to make sure the taped out layers match the real circuit. LVS checks for shorts, opens, missing components, component mismatches and parameter mismatches.

- Formal verification: During synthesis and place-and-route, the design will be optimized with transformation, deletion and insertion of standard cells. Formal verification compares the optimized design netlist with the ideal netlist to ensure the equivalence of the two designs.

- Staic Timing Analysis (STA): In order to evaluate the perforamance of the implemented design, the timing analysis needs to take many aspects into account, like on-chip variation, parasitic resistance/capacitance. The analysis reports setup/hold time violations, critical path, arrival/required time of a signal that can be used to calculate the required cycle time.

- Power Integrity analysis: The noise from power/ground is a major problem that could lead to the failure of the chip. The power integrity analysis checks if the signals coupled to power and ground would become a fatal noise source.

- Signal Integrity analysis: Signals traveling from one end to the other are suffered from noises from adjacent signals (crosstalk), distortion and degradation. The signal integrity analysis makes sure that the signals are not corrupted before getting to the destination.

- Electromigration analysis: With the scaling of the wire width, the probability of failure due to electromigration increases with increasing current density. The electromigration analysis are conducted to prevent the loss of wire connections and increase the reliability of the chips,

- Voltage drop analysis: A power/signal wire that carries large current and travels for a long distance is suffered from IR drop or voltage drop. The lower supply/signal voltage level could degrade the performance or even damage the functionality. The voltage drop analysis calculates the IR drops along the power and signal traces to highlight the potential risks.

4.6 Case Study: Implementation of the Register File

The register file is a memory structure between the functional units and the caches to hold data and addresses. Register files are normally implemented with SRAM-based full-custom design that optimizes for area and power consumption. However, The foundry only provides memory compilers for single-ported SRAM and dual-ported SRAM. It takes a lot of efforts to build full-custom register files with different number of read/write ports for design space exploration. Unfortunately, implementation of a register file with an array of flip-flops generally causes a large area and routing congestions. The one-line code in Chisel instantiates an array of flip-flops for the register file with num_registers entries and width of register_width.

```
val regfile = Mem(num_registers, UInt(width=register_width))
```

With no constraint to the design, every flip-flop is wired to multiplexers for multiple sources and destinations. The large amount of wires uses up all the routing resources and results in routing congestions. Figure4.2 shows that the block diagram of a synthesized register file and the implementation of it. Without any placement guidance, the location of the flip-flops (shown as yellow dots) are calculated for optimization. The tool fails to solve the routing problem with limited routing tracks and lead to shorts between the wires (shown as white Xs).

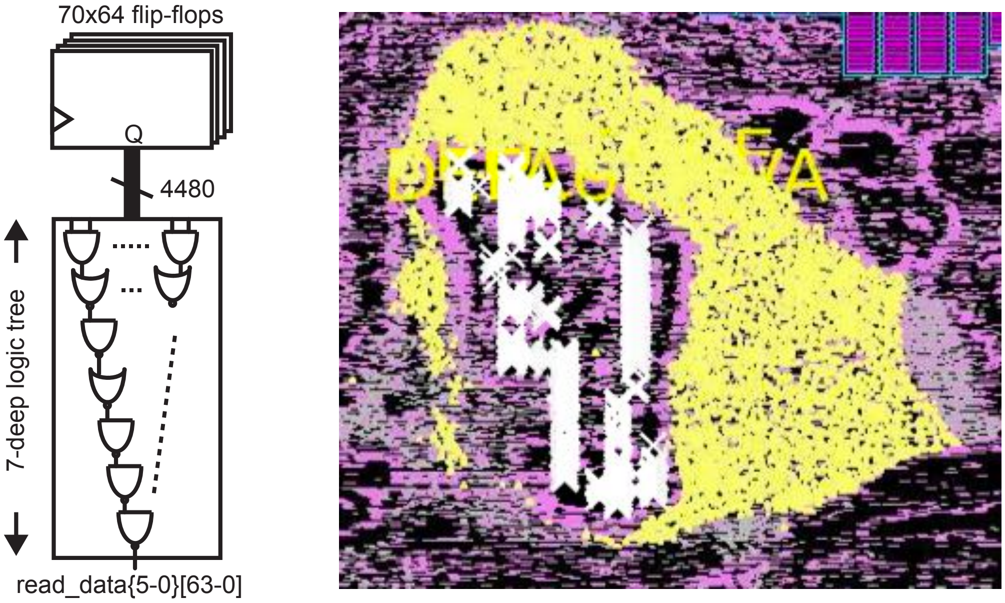


Figure 4.2: The register file implemented with flip-flops is a bottleneck for routing, resulting in shorts.

The 6-read 3-write register file in the BROOM chip was designed as a customized macro composed of standard cells to aid design convergence and improve performance. Figure 4.3

shows the schematic of a customized register file cell and the similar implementation with standard cells. The customized register file has dedicated read/write passgates to connect to the storage nodes of the cross-coupled latch. The wire for the same read/write port is shared among the same column. The register file design in BROOM mimics a customized cell with a 3-input multiplexer, a flip-flop and six tri-state buffers. The tri-state buffers allow different entries to share the read wires thus relieving routing congestion. To reduce parasitic capacitance on the read wire, it is divided into a hierarchy of shorter local wires and connected to the global read wire through a multiplexer. The register files are placed as an array macro. Using a gate-level description of a register file and manually guided place-and-route, a high-port-count register file is feasible without need for custom IP.

To integrate the gate-level register file into Chisel language, a block box is instantiated with the corresponding ports as shown in the following codes.

```
class RegisterFileArray(
  num_registers: Int,
  num_read_ports: Int,
  num_write_ports: Int,
  register_width: Int)
  extends BlackBox
{
  val io = new Bundle {
    val clock = Clock(INPUT)
    val WE = Vec(num_registers, Bool()).asInput
    val WD = Vec(num_write_ports, UInt(width = register_width)).asInput
    val RD = Vec(num_read_ports, UInt(width = register_width)).asOutput
    val WS = Vec(num_registers, UInt(width = 2)).asInput
    val OE = Vec(num_registers, UInt(width = num_read_ports)).asInput
    val RDSEL = Vec(num_read_ports, UInt(width = 4)).asInput
  }
}
```

Multiple constraints are applied to the physical design to force it to place and route in the array fashion. The standard cells are constrained to place within the soft boundries created for each cell. The boundary boxes form an array so the routing wires can be connected vertically and horizontally without congestions as shown in Figure4.4. The unit cell of the register file and the shared read wires need to set as ”don’t touch” so the tool won’t make any effort to optimize it or adding buffers right after the tri-state buffer.

4.7 Measurement

The photograph of the wire-bonded BROOM chip for chip-on-board packaging is shown in Figure 4.5. The 124 wire-bonding pads include serial IOs, high-speed clock inputs, clock

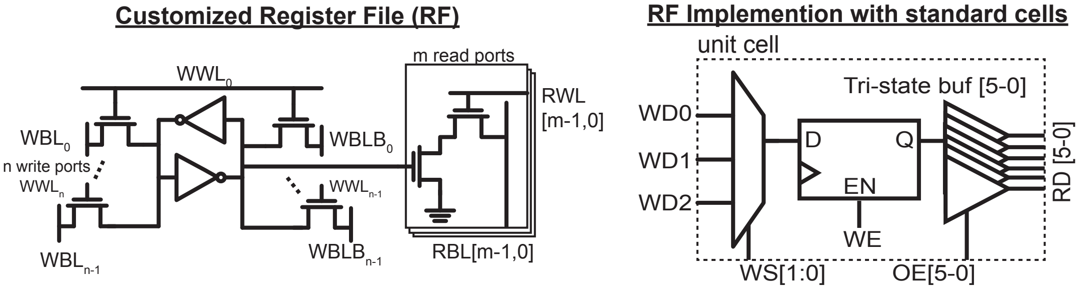


Figure 4.3: The schematic of a full-custom cell for the register file and the implementation with standard cells.

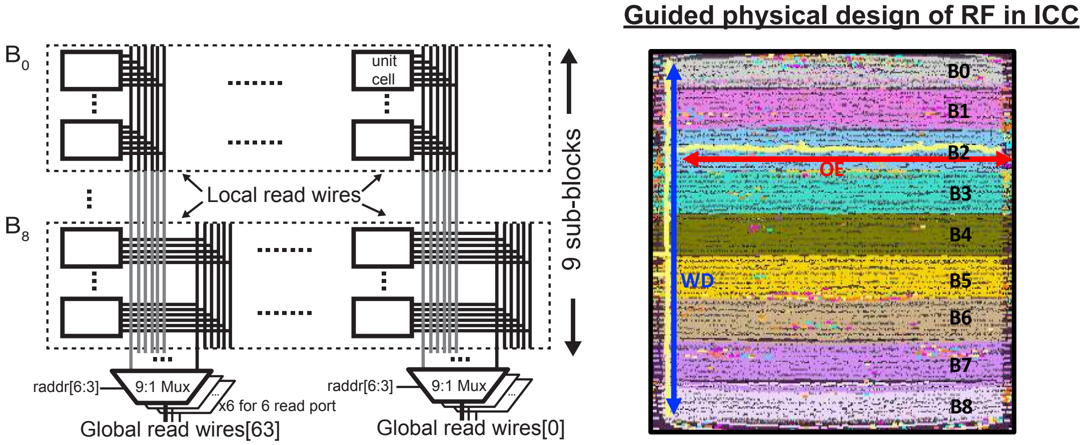


Figure 4.4: The array of customized unit cell with hierarchical read wires and the results of the guided physical design.

input selection for uncore clock, JTAG, and power/ground for different domains (IO, core, and uncore).

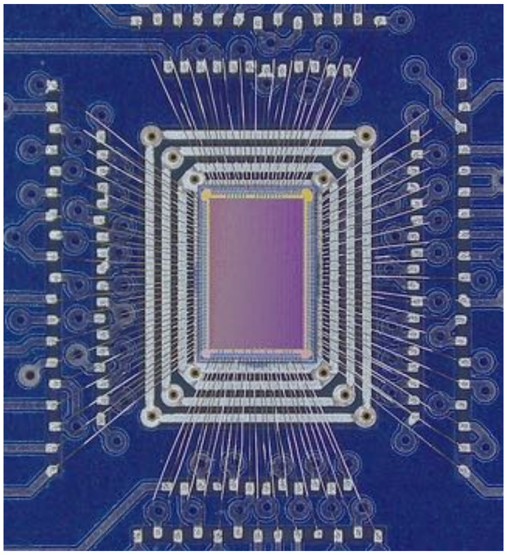


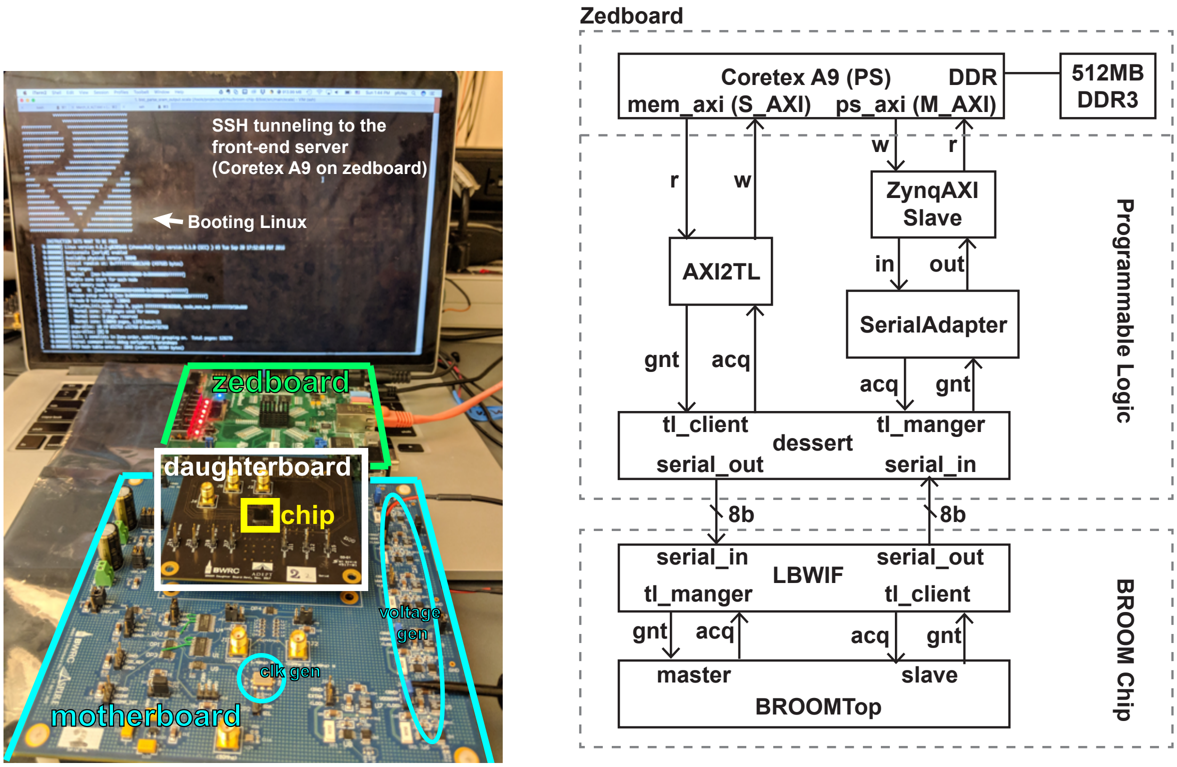
Figure 4.5: The wire-bonded BROOM chip.

4.7.1 Measurement setup

Figure 4.6(a) shows the setup for the measurement. The daughterboard, where the chip is mounted on, is connected through FMC to the motherboard, which provides voltage and clock generation controlled by I2C. Different capacitors are populated on the daughterboard to filter the noise to power/ground. The probe pins on the daughterboard allows us to take a peek at some signals for debugging. The high-speed clocks (one single-ended and one differential) are connected through SMA connectors.

The jumpers of the voltage generation islands on the motherboarad provide options to use on-board voltage sources or external voltage sources. The clock sources include the slow clock from the FPGA, the clock generated from Si570 programmable oscillator, and the clock from external sources.

The motherboard is then connected to the FPGA evaluation board (Xillinx Zedboard), which has a Cortex-A9 ARM processor as the front-end server and provides 512MB memory. The interfaces between the chip, FPGA, and ARM core is shown in Figure 4.6(b). Since the ARM processor uses AXI4 and the chip uses SerialIO and TileLink [46] for data transactions, bridges need to be built in FPGA to convert the protocols.



(a)

(b)

Figure 4.6: (a) A photo of the measurement setup. (b) Interface bridges between the chip and FPGA.

4.7.2 Measurement Results

The open-source out-of-order processor with different cache resilient techniques, BROOM, is able to execute all the benchmarks in the RISC-V test suite and boot Linux. The performance of BROOM is evaluated with CoreMark benchmark [47]. BROOM achieves 3.77 CoreMark/MHz, which is better than a commercial processor Cortex-A9 with 3.71 CoreMark/MHz. The out-of-order nature allows the processor to run CoreMark at 0.9 CPI.

The shmoo plot (Figure 4.7) shows that the processor is able to operate up to 1 GHz at 0.9 V and down to 0.6 V without any assist techniques. LR with 5% loss of L2 cache capacity allows the processor to work at 70 MHz under 0.47 V. The bit error rate (BER) of the SRAM macros at different voltages, as shown in Figure 4.8(a), is measured by running the BIST. The Vmin reduction enabled by different resiliency techniques are shown in Figure 4.8(b). DCR alone is limited by multi-bit errors in the same set and needs to work with LD or LR to achieve the lowest Vmin with less impact on cache capacity. For the same disabled cache capacity (1% or 5%), the LR scheme drops Vmin by 10 mV more than LD.

For 5% cache capacity loss with LR, there is a 2.3% increase in L2 cache misses and 0.2% degradation in CPI by running a cache-intensive benchmark (SPEC CPU2006 `bzip2`).

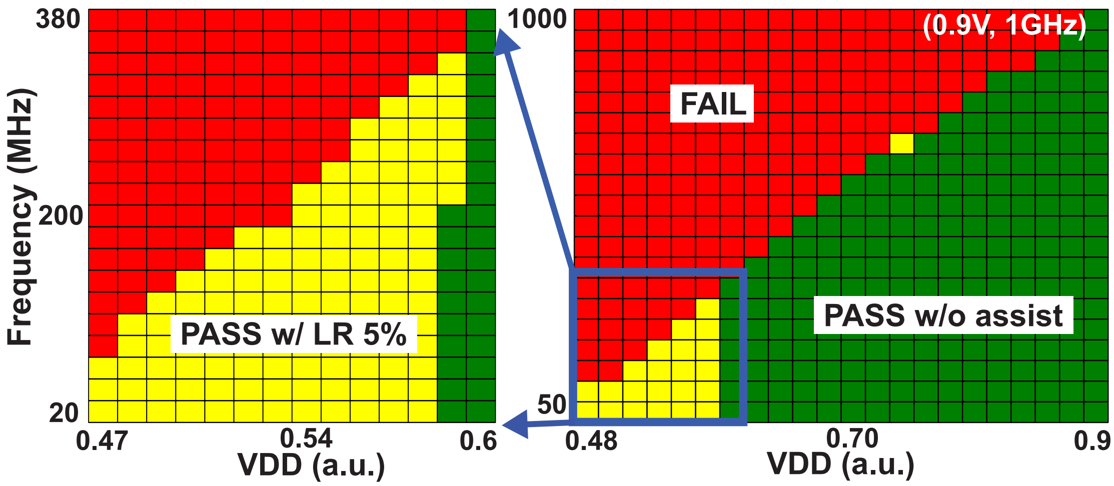


Figure 4.7: Operating voltage and frequency range with/without LR.

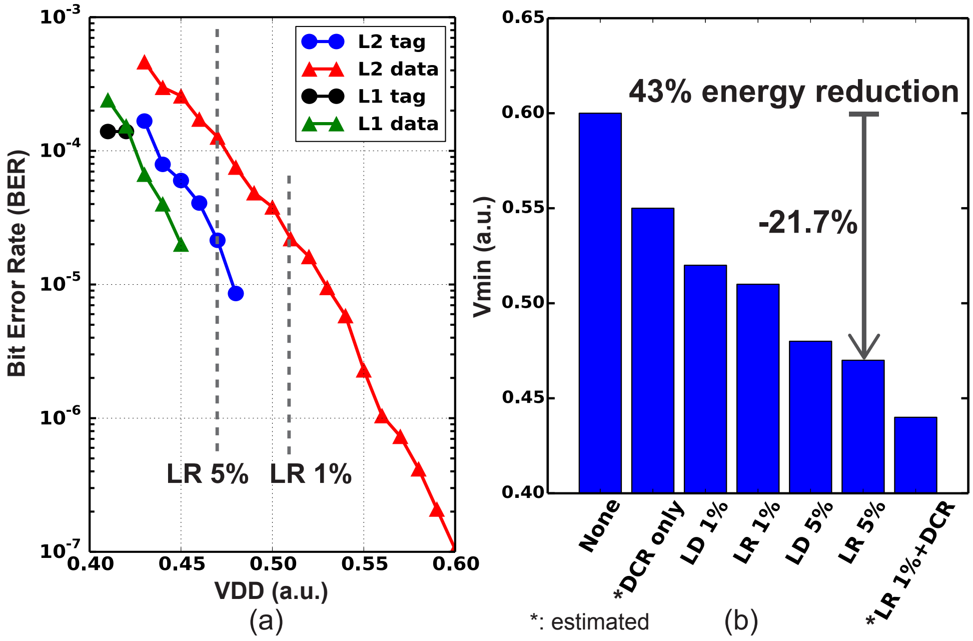


Figure 4.8: (a) Measured SRAM bitcell failure rate versus voltage and (b) Vmin reduction of various resiliency techniques.

ccbench [48] is a small collection of micro-benchmarks designed to empirically characterize some of the interesting parameters of a processor and its memory system. The micro-benchmark, *caches*, determines the number of levels in the memory hierarchy, the capacity of each cache, and the access latency at every level of the memory hierarchy. Figure 4.9 shows that three phases of cycle per interation with different array size. The switching points, happening when the array is too big to fit entirely within the L1 or L2 cache, reveal the L1 and L2 cache size of 16KB and 1MB. The L2 system is operate at half of the core frequency so that it takes about 100 cycles to fetch L2 data. When the array size is larger than 1MB, the latency is significantly increased due to serial accesses to the off-chip DRAM in the BROOM measurement setup.

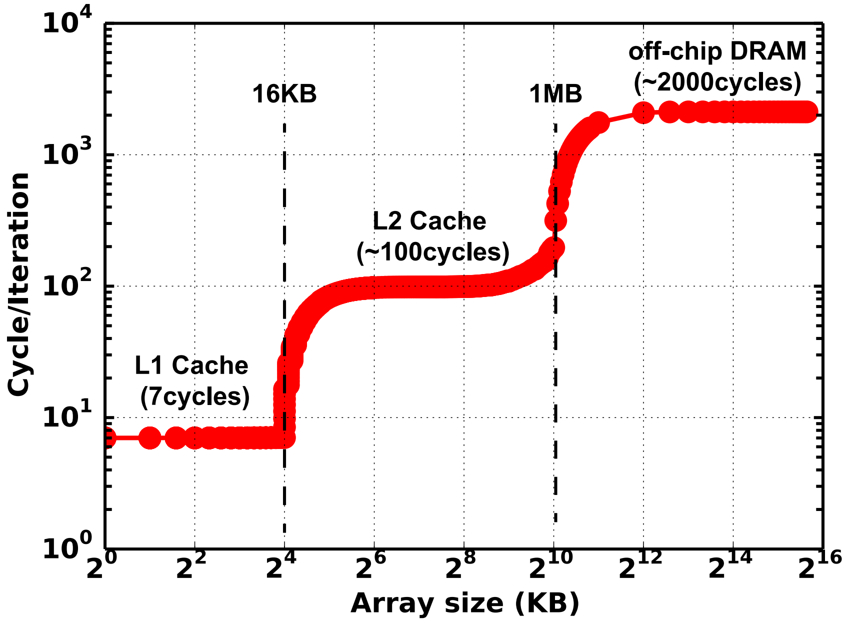
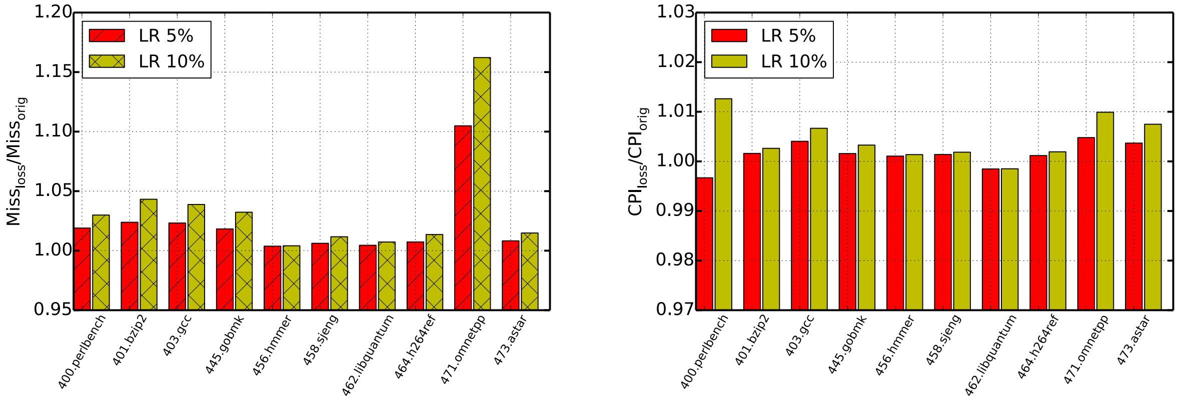


Figure 4.9: Access latencies of the cache hierarchy measured with ccbench.



(a)

(b)

Figure 4.10: The effect of increasing (a) miss rate and (b) CPI with 5% and 10% capacity loss on different benchmarks.

4.8 Conclusion

The BROOM chip is one good practice of the agile hardware design leveraging Chisel and RocketChip. This design contains 124 I/Os, 72 million transistor and 72 SRAM macros. Although the physical design flow is not completely automated for now, FIRRTL passes and python scripts significantly save the time that was needed for manually scripting the constraint file. Future effort can be made in generalizing the scripts for auto-generation of the constraint files, the floorplanning and power planning scripts. Integrating them into a framework could make the physical design flow more agile. The implementation of the register file is highly parametrized and can be easily reconfigured while design space exploration to avoid taking long design cycles to build full-custom register file macros.

An open-source out-of-order superscalar processor implements the 64-bit RISC-V instruction set architecture (ISA) and achieves 3.77 CoreMark/MHz. The 2.7mm1.8mm chip includes one core operating at 1.0 GHz at nominal 0.9V with 1MB of level-2 (L2) cache in a 28 nm HPM process. A line recycling (LR) technique reuses faulty cache lines that fail at low voltages to correct errors with only 0.77overhead. LR reduces minimum operating voltage to 0.47 V, improving energy efficiency by 43on CPI.

Chapter 5

Emerging Nonvolatile Memory

This chapter discusses the applications of emerging nonvolatile memories. Fast nonvolatile memory devices (NVMs) offer a tremendous opportunity to eliminate memory leakage current during standby mode. Resistive random access memory (RRAM) in a crosspoint structure is considered to be one of the most promising emerging NVMs. However, the absence of access transistors puts significant challenges on the write/read operation. The differential 2R crosspoint RRAM sctructure has been proposed to improve the read margin, alleviate the endurance constraint and solve the data dependency of the leakage current.

5.1 Introduction

An energy-efficient memory system is necessary for continued scaling of mobile systems into nanometer technologies. Mobile devices and some IoT edge devices are idle more than 90% of the time, highlighting the need to minimize standby energy consumption. As the technology scaling trend continues, leakage current in SRAM-based cache memories will dominate energy consumption in standby mode. Nonvolatile memories can be powered down completely without loss of states, eliminating the leakage current. Flash memory [49], the most mature nonvolatile memory technology, has a large storage density and small cell size. However, slow program/erase (P/E) speeds make it too impractical for caches, and physical limitations associated with oxide thickness prevent flash memory from continued scaling. Therefore, there is a perceived need for a high-speed nonvolatile memory that can be used as a universal memory, replacing both flash memory and SRAM.

A universal memory should combine all the features of different level of memory hierarchy: low latency, high density, non-volatility, and high endurance (Figure 5.1). Although no existing memory technologies satisfy all of the requirements perfectly, the emerging NVMs have been developed toward the ultimate goal of being a universal memory. Emerging memory technologies include ferroelectric memory (FeRAM), spin-transfer torque memory (STT-RAM), phase-change memory (PCM), and resistive memory (RRAM). FeRAM [7] has a similar structure as the DRAM with the ferroelectric layer to achieve nonvolatility. FeRAM has difficulties to reach high density due to the limitation of the scaling process. It also requires write-after-read to solve the destructive read problem. STT-RAM [8, 9] has high endurance and high switching speed, and is being evaluated as a successor to DRAM. However, the resistance ratio between two states is low, which poses a big challenge on sensing. PCM [10, 11] is a thermally driven process, which suffers high programming current and resistance drift. RRAM [12, 13], which often referred to as a memristor, is one of the promising candidates for a universal memory. RRAM has a higher resistance ratio than STT-RAM but still suffers from limited endurance.

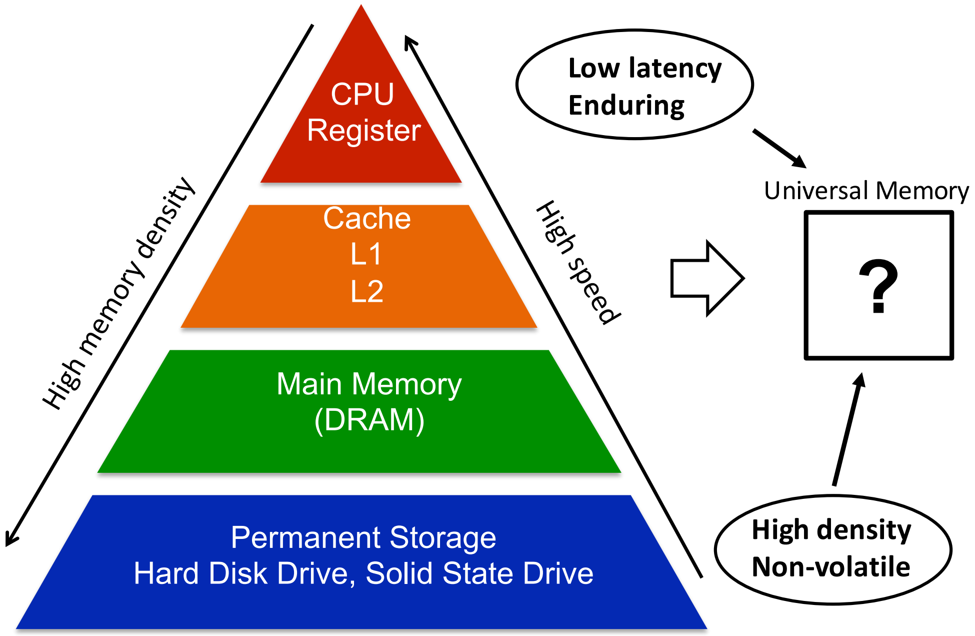


Figure 5.1: The pyramid of memory hierarchy for a fast, massive, and nonvolatile memory.

Although the emerging NVMs cannot completely replace SRAM and DRAM due to the constraints of speed and endurance, the technologies are able to bring the storage closer to the processor. RRAM is highly compatible with the CMOS process and can be implemented on chip to avoid long off-chip communication. This chapter explores the advantages and challenges of making RRAM the last-level cache.

5.2 Array architecture

5.2.1 RRAM bitcell structure

RRAM features a simple structure, small cell area, low switching voltage, and fast switching speed. The resistive memory cell has a sandwiched structure with two metal electrodes above and below, and a metal oxide in the middle. To SET a cell, a positive voltage is applied across the device, increasing its conductance, i.e. switching to a low resistance state (LRS). To RESET a cell, a negative voltage is applied and the cell switches to a high resistance state (HRS). The cell retains the same resistance state even with no power supplied. Although the endurance is approaching 10^{10} cycles, it remains RRAM's primary challenge.

Conventionally, an RRAM cell is constructed of one transistor and one programmable resistive device (1T1R), as shown in Figure 5.2. The transistor not only works as a switch for accessing the selected cell and isolating unselected ones, but also constrains the write current and limits cell disruption. However, in order to provide sufficient write current, the transistor needs to be large, which would dominate the cell area.

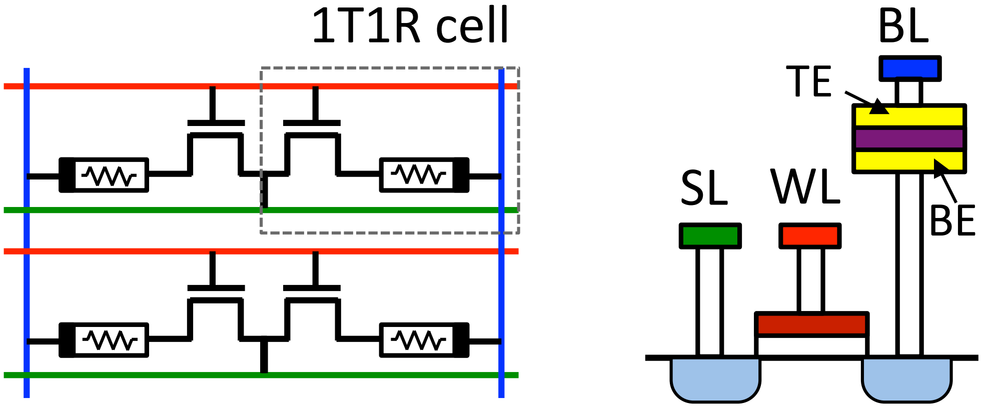


Figure 5.2: 1T1R RRAM array architecture and the cross-sectional view of the cell.

An alternative approach is the crosspoint architecture [50], shown in Figure 5.3. In a crosspoint array, RRAM cells are sandwiched between wordlines (WLs) and bitlines (BLs), which could achieve the ideal cell size of $4F^2$. Moreover, the resistive memory cells are fabricated in the back-end of the line (BEOL) process, which enables peripheral circuits to be hidden underneath the crosspoint array. Using a multi-layer structure [51] could further reduce the effective cell area, as shown in Figure 5.3. However, the absence of access transistors in a crosspoint array complicates write and read operations.

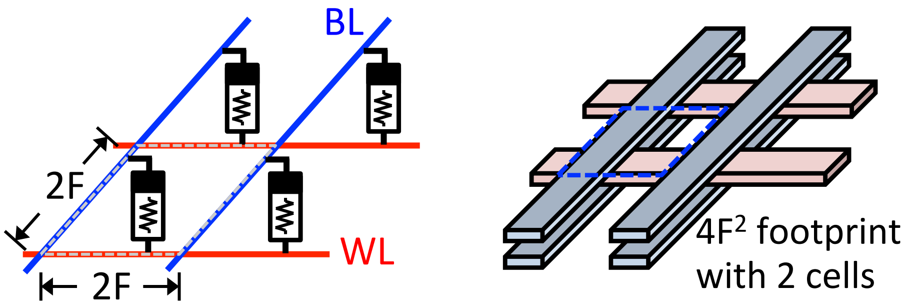


Figure 5.3: The crosspoint RRAM array architecture and two stacked layer.

5.2.2 Switching behavior

The switching behavior of an RRAM cell depends on the write voltage (V_{SET}, V_{RESET}), the duration of write pulses (T_{SET}, T_{RESET}), and the high/low resistance values (R_H, R_L). Figure 5.4 shows the tradeoff between the required time (T_{SET}) and voltage (V_{SET}) for programming a cell from the HRS to the LRS under different target RL values. A higher R_L requires less time and energy to program and also suppresses the overall leakage current. However, to maintain a sufficient read margin, a smaller R_L is preferred so that the R_H vs. R_L ratio is larger. Figure 5.5 shows the relationship between write energy and R_L under different V_{SET} values. Writing the cell with a higher voltage and a shorter pulse is more energy efficient. However, variations in the pulse duration widen the distribution of cell resistances.

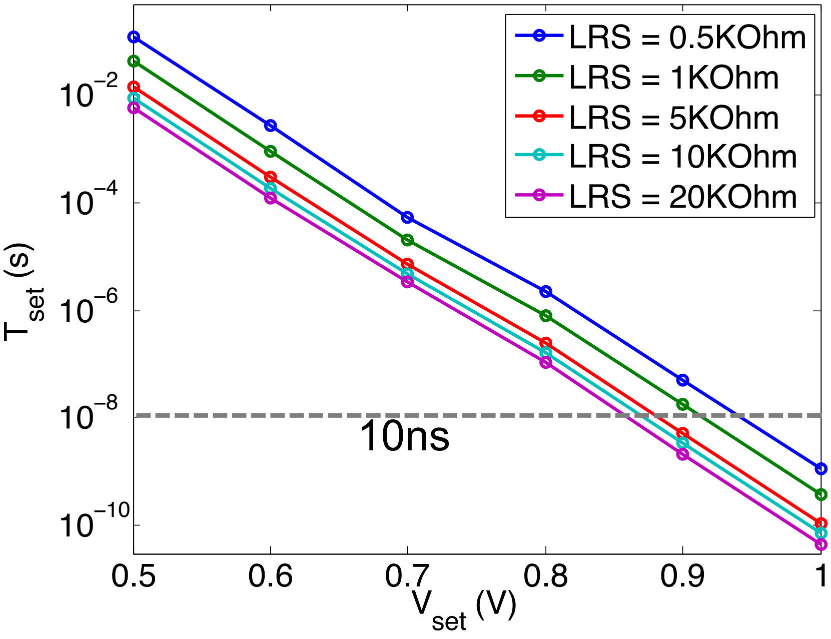


Figure 5.4: Time required for setting the resistance from HRS to LRS under different V_{SET} and R_L.

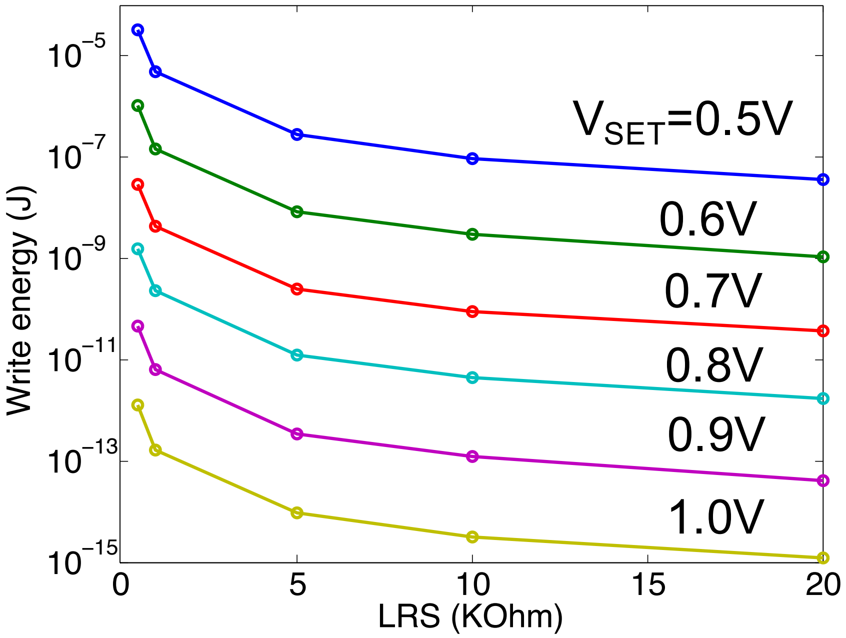


Figure 5.5: Write energy for setting the resistance from HRS to LRS under different V_{SET} and R_L.

5.2.3 Analysis of leakage current

While a crosspoint array achieves high density by avoiding access transistors, it loses the ability to isolate unselected cells. To relax the requirements for minimizing write disturbance in crosspoint arrays, unselected WLs and BLs must be biased precisely. Figure 5.6(a) shows the V/2 bias scheme, which limits the voltage disruption along the selected WL and BL to V/2. Another option is the floating wordline half-voltage bitline (FWHB) scheme shown in Figure 5.6(b), which applies V/2 to the unselected BLs and floats the unselected WLs.

In this case, the voltage drop across the cell (V_{DROP}) is generally less than V/2, but it disturbs more cells. The write voltage should be large enough to successfully switch the cell but not so large as to cause a write disturbance. Undesired disruption voltages also induce leakage currents through unselected cells. The amount of leakage current is data-dependent, and the worst case occurs when all the unselected cells are in the LRS. Since the wire/switch resistance in an array is not negligible, variable IR drop amounts change the voltage applied across the cell, expand the cell variability distribution, and may even result in a write failure.

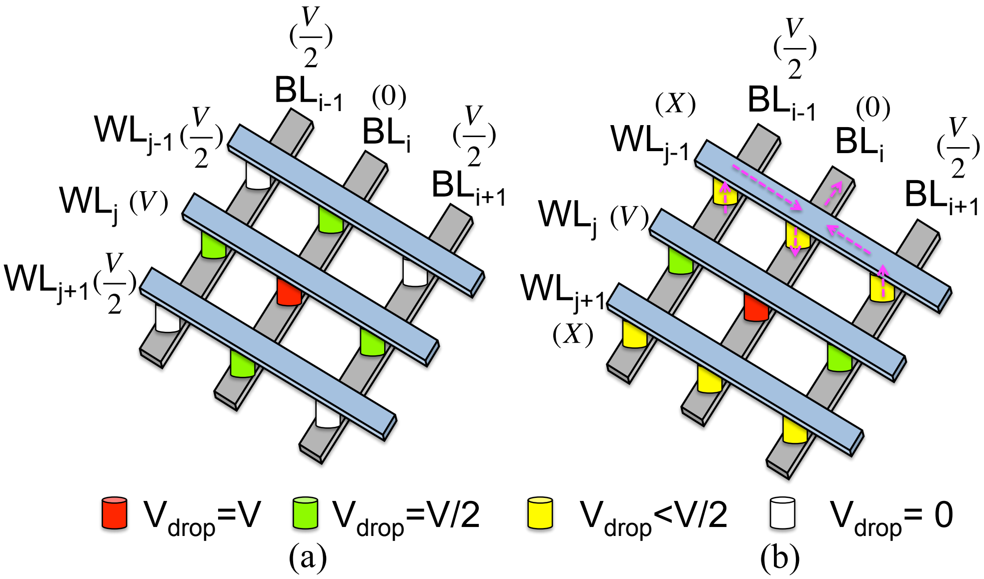
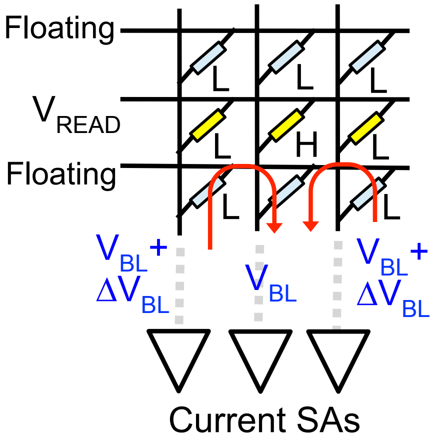


Figure 5.6: The bias scheme for the crosspoint array. (a) V/2 bias scheme. (b) Floating Wordline Half-voltage Bitline (FWHB) scheme.

A common approach to detect the resistance state is by current sensing, which mirrors the current flowing through the selected cell and compares it with a reference current (I_{REF}). However, the BL current (I_{BL}) in a crosspoint array includes both the selected cell current (I_{CELL}) and the total leakage current (I_{LEAK}). Figure 5.7 illustrates the worst-case situation, when the selected cell is in a HRS and the other cells in the same array are all in the LRS. In this case, the read operation would fail when the BL current becomes larger than the reference current. Since the total leakage current depends on the number of cells, this situation constrains the array dimension. Also, the BL voltage fluctuation (ΔV_{BL}) and the leakage current are both data-dependent. Therefore, it is challenging to design a robust sensing circuit, under all cell variability distributions, data patterns, leakage currents, and PVT variations.



For read-HRS worst case:

$$I_{REF} > I_{BL}$$

$$I_{BL} = I_{CELL} + I_{LEAK}$$

$$I_{CELL} = \frac{V_{READ} - V_{BL}}{R_H}$$

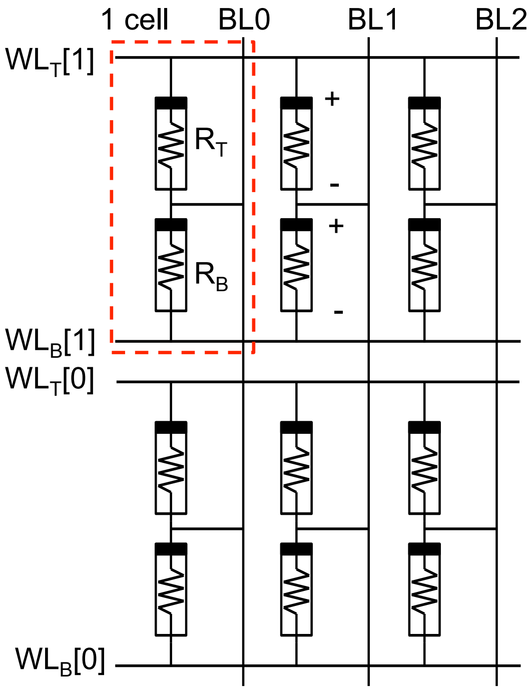
$$I_{LEAK} = \frac{\Delta V_{BL}}{R_L / (n-1) + R_L} \cdot (m-1)$$

Figure 5.7: Worst case of reading HRS in current sensing scheme. (m: BL length, n: WL length).

5.3 Differential 2R Crosspoint Array

We propose a differential 2R crosspoint structure, shown in Figure 5.8, which can be read by using voltage sensing. The goal is to trade density for speed and robustness, thus to make it applicable for use in memory hierarchy. In this structure, two resistive devices with complementary resistance states are used to represent a 1-bit datum. To write a 1, SET R_T to a low resistance state and RESET R_B to a high resistance state; to write a 0, RESET R_T to a high resistance state and SET R_B to a low resistance state. The cell state can be readily determined by sensing the intermediate node X while applying V_{READ} to WL_T and ground to WL_B. The voltage on node X depends on the voltage divider formed by R_T and R_B. For evaluation purposes, BLs are connected to a StrongARM sense amplifier with a reference voltage of $V_{READ}/2$. Therefore, the read operation is immune to the leakage current flowing from neighboring BLs, which greatly increases the read margin without limiting the block size. The differential 2R cell always contains one HRS and one LRS, which solves the data dependency issue. Furthermore, the stacked resistors suppress leakage consumption during the read operation.

It is possible to design a 2R crosspoint array in a single layer of RRAM. However, thanks to the ability of stacking multiple RRAM layers, the differential 2R cell can be constructed between different metal layers with minimal area penalty. Since R_T and R_B have opposite electrodes connected to WL_T and WL_B, the same voltage can be applied to WL_T and WL_B to set one device and reset the other. The write operation is illustrated in Figure 5.8. In the write-1 operation, both WL_T and WL_B are connected to a write voltage, V_{WRITE}, and the BL is connected to ground. A positive V_{WRITE} drops across R_T, which sets R_T to the LRS. In the meantime, a negative V_{WRITE} drops across R_B, which resets R_B to the HRS. In contrast, to write a zero, the BL is connected to V_{WRITE}, and WL_T and WL_B are connected to ground.



	Form-R_T	Form-R_B
R_T	FORM	X
R_B	X	FORM
WL_T	V_{FORM}	V_{FORM}
WL_B	0	0
BL	0	V_{FORM}

	Write-1	Write-0
R_T	SET	RESET
R_B	RESET	SET
$WL_{T/B}$	V_{WRITE}	0
BL	0	V_{WRITE}

Read operation:
$WL_T = V_{READ,}$ $WL_B = 0$

$$\boxed{V_{BL}=V_{READ}\times R_B/(R_T+R_B)}$$

Figure 5.8: The architecture of a differential 2R crosspoint array and the table of operating conditions in form/write/read mode.

The forming operation in the initialization step is required to construct the conductive filament in each resistive device after fabrication. A forming operation is similar to a set operation with a higher voltage and a longer duration. Two sequential phases are applied to initialize R_T and R_B separately. In the first phase, the selected WL_T is connected to V_{FORM} while the selected BL and WL_B are held at ground. In the second phase, WL_T and the BL are connected to V_{FORM} and WL_B is connected to ground. In the two phases, R_T and R_B are applied to V_{FORM} and switched to the LRS respectively.

5.3.1 Circuit Implementation

Array Segmentation

There are twice as many cells in the D-2R array as in the conventional crosspoint array. During operation, half of the cells are in the HRS and half of them are in the LRS. Therefore, the leakage current would be 8% larger than in the worst case of a conventional crosspoint array. However, the leakage current is a constant value in the D-2R scheme, and the data-dependent variable IR drop issue does not exist. The write current (I_{WRITE}) in the D-2R scheme with V/2 biasing includes the cell current ($I_{CELL} = V/R_L$) and the leakage current

($I_{LEAK}(n-1) \times V/RL$). The energy efficiency (I_{CELL}/I_{WRITE}) decreases with increasing array dimensions.

Array segmentation, similar to the divided wordline technique [52] employed in SRAM for reducing WL loading, disturbance, and power consumption, is used here to reduce the number of activated cells and mitigate the write leakage current. To keep the write current under 100μA, 4-cell wide WLs are required. Instead of building a 4x4 array with its own peripheral circuit, a large array is constructed by segmenting one WL into local WLs (LWLs). Only one LWL is active at a time to reduce the write leakage current. Switches are inserted every four columns to connect the global WL (GWL) and LWLs. Figure 5.9 shows a cross-sectional view of the D-2R array with array segmentation. Although placing transistors under the array minimizes their overhead, additional area is consumed for routing transistors to the GWL and LWL metal layers. There is a tradeoff between area penalty and leakage current. For a LWL of 4 cells wide, the area would be twice the size of that without array segmentation. Compared to a 140 F^2 SRAM bitcell, an RRAM cell in a crosspoint array of $4F^2$ cell area is 35x smaller. However, the area penalty due to array segmentation increases the equivalent bit cell area to $10F^2$, which is still much smaller than an SRAM bit cell and a 1T1R RRAM cell.

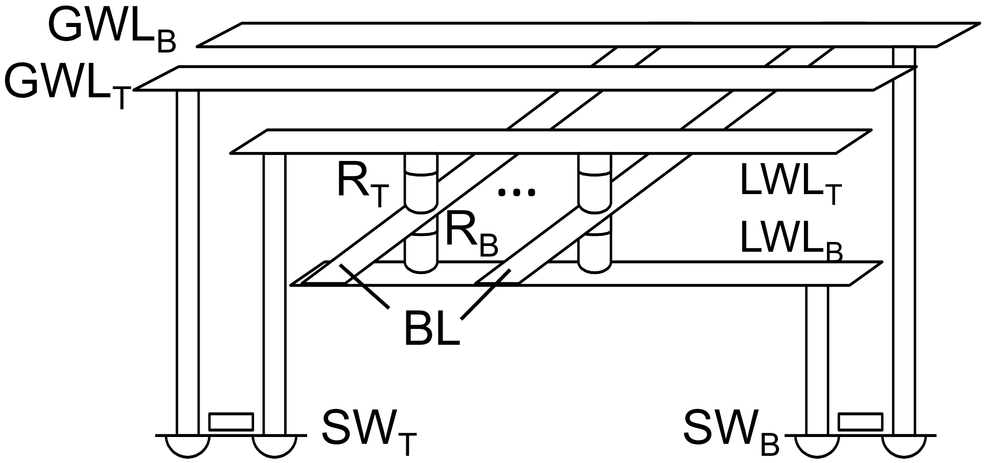


Figure 5.9: Cross-sectional view of the differential 2R array with array segmentation.

Sense-before-Write

The resistive value of the memory cell varies with the voltage and period of the write pulse. Repeated SET pulses applied to the same cell reduce its resistance value until it hits the lowest resistance level. Doing this would result in very high current consumption and a wide cell resistance distribution. To prevent the over-SET situation, the sense-before-write approach is applied [53]. At the beginning of the write cycle, a read operation is first conducted and the output is fed back to a control circuit to determine whether to write or not. The cell would not be written again unless there is a need to flip the state. By using the sense-before-write approach, the cell resistance distribution is narrower and the leakage current is suppressed by keeping each LRS at a higher resistance value. Moreover, avoiding unnecessary cell access elongates the endurance.

The block diagram of a 64KB D-2R crosspoint RRAM macro that contains 8 blocks is shown in Figure 5.10. In SRAM, WLs drive the gates of access transistors. In a crosspoint array, however, WLs are connected to V_{WRITE}, V_{READ}, or ground, depending on data input values and operational modes. Therefore, 8 bits of data need separate WL/BL drivers to provide the correct voltage to program the cell. V_{WRITE}, V_{READ} and the unselected BL voltage (V_{HALF}) are provided by the voltage generator, which is not shown in the block diagram.

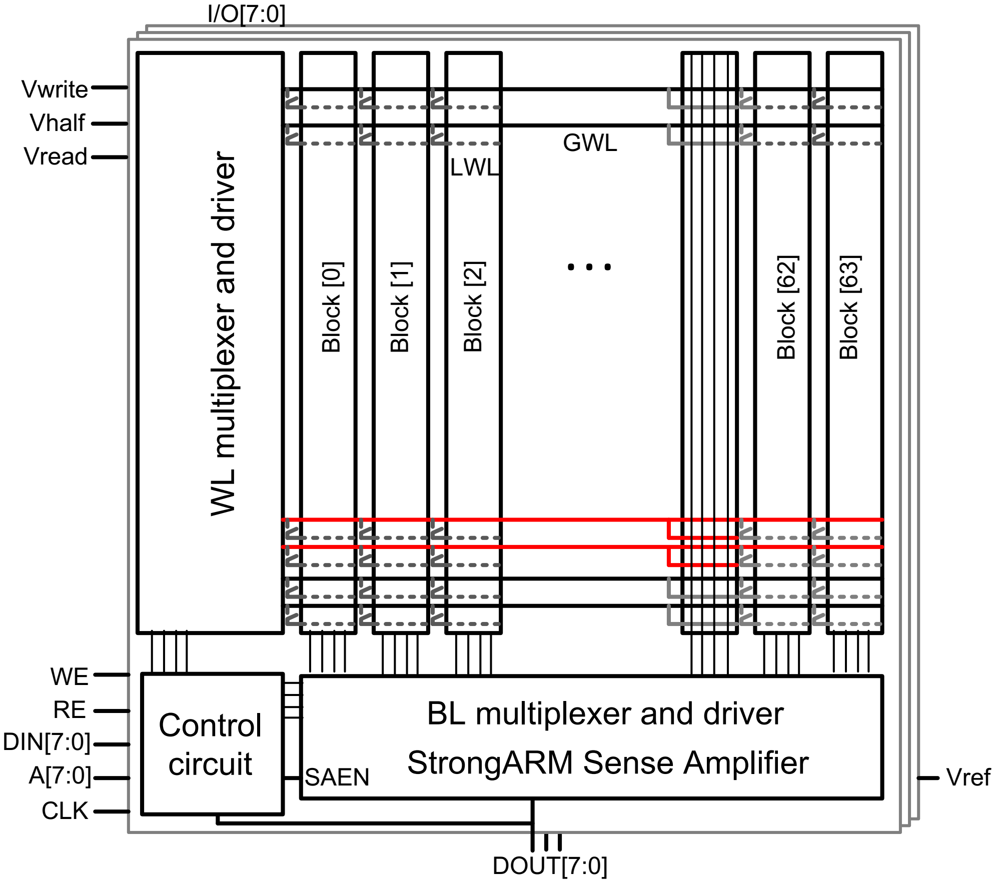


Figure 5.10: Block diagram of a 64KB crosspoint RRAM circuit.

The control circuit generates all the input control signals, such as write enable (WE), read enable (RE), input data (DIN), addresses (A) and output data (DOUT), to determine the operational mode and corresponding control signals to read/write circuits. WL/BL multiplexers and drivers deliver different voltage levels (V_{FORM}, V_{WRITE}, V_{HALF}, V_{READ} and ground) to WLs and BLs according to the control signals. The read voltage (V_{READ}) is set to a low value of 0.3V to prevent disturbance. Thus, the StrongARM sense amplifiers with PMOS input transistors are used to sense the inputs with low common mode. It compares the BL voltage to the reference voltage (V_{REF}) and outputs the result. The voltage-sensing scheme in D-2R crosspoint array is less susceptible to cell distribution and data pattern variability than the conventional current-sensing scheme in a 1R crosspoint array.

5.3.2 Simulation Results

Simulation of one block and its peripheral circuits is conducted using Eldo with a 28/32nm predictive technology model (PTM) and a Verilog-A RRAM model. The RRAM model illustrates the physical behavior of SET/RESET processes to fit the measurement results [54]. Figure 5.11 shows the simulation waveform. In the highlighted period, $\mathrm{WL_T}[2]$ and $\mathrm{WL_B}[2]$ are connected to ground and BL[3] is connected to V_{WRITE} to SET $\mathrm{cell_{23B}}$ and RESET $\mathrm{cell_{23T}}$. The unselected WLs are kept at $V_{WRITE}/2$ to prevent disturbance. The switching behavior of $\mathrm{cell_{23T}}$ and $\mathrm{cell_{23B}}$ is confirmed by noting the increase in current of $\mathrm{cell_{23B}}$ (SET operation) and the decrease in current of $\mathrm{cell_{23T}}$ (RESET operation).

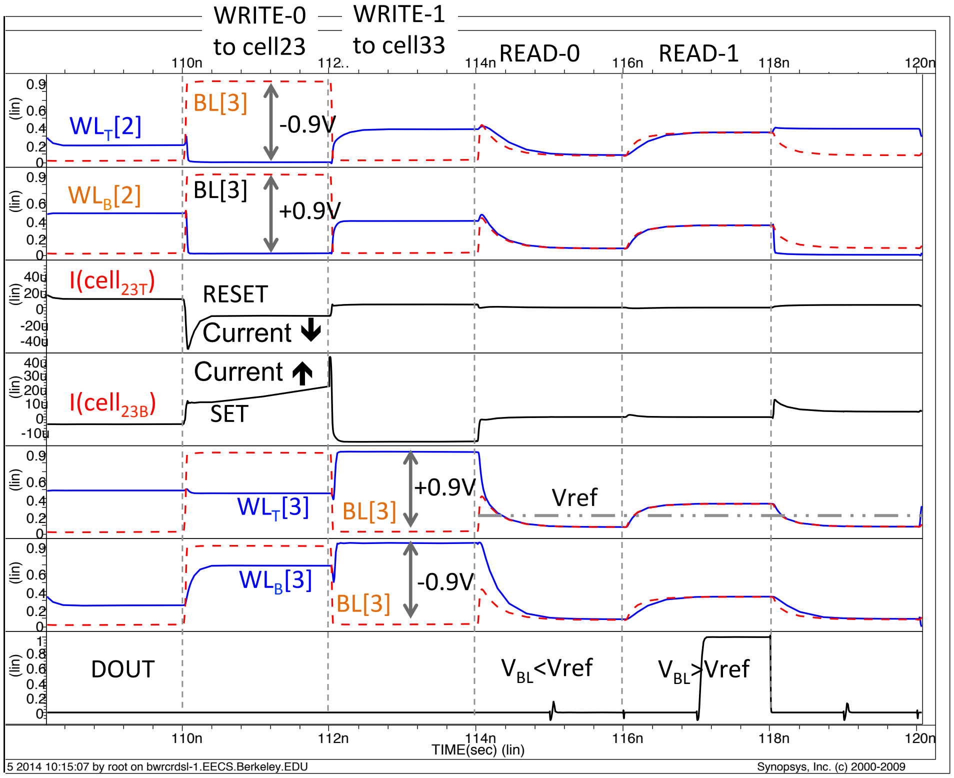


Figure 5.11: Waveforms of read and write operation in the differential 2R crosspoint array.

Table 5.1: Parameters in D-2R circuit simulation.

Clock Frequency	500MHz
Capacity	64KB
Power supply	1.0V
Write voltage (V_{WRITE})	0.95V
Read voltage (V_{READ})	0.3V
Reference voltage	0.2V
R ratio (R_H/R_L)	9KΩ/8KΩ
Write current (one block)	140μA
Read current (one block)	17μA
Standby current	~0μA

During a read operation, V_{READ} is applied to the selected $\mathrm{WL_T}$ and $\mathrm{WL_B}$ is connected to ground. Thus, the BL voltage is proportional to the resistance ratio of R_T and R_B. The sense amplifier compares this BL voltage to V_{REF} to determine the ouput (DOUT). The sensing enable (SAEN) signal is triggered after the voltage difference is fully developed. Table 5.1 shows the parameters used for simulating the D-2R crosspoint RRAM circuit. The average current during a write cycle in each block is 140μA, and the average current during a read cycle in each block is 17μA. The switches are designed for a maximum voltage drop of 50mV during read/write.

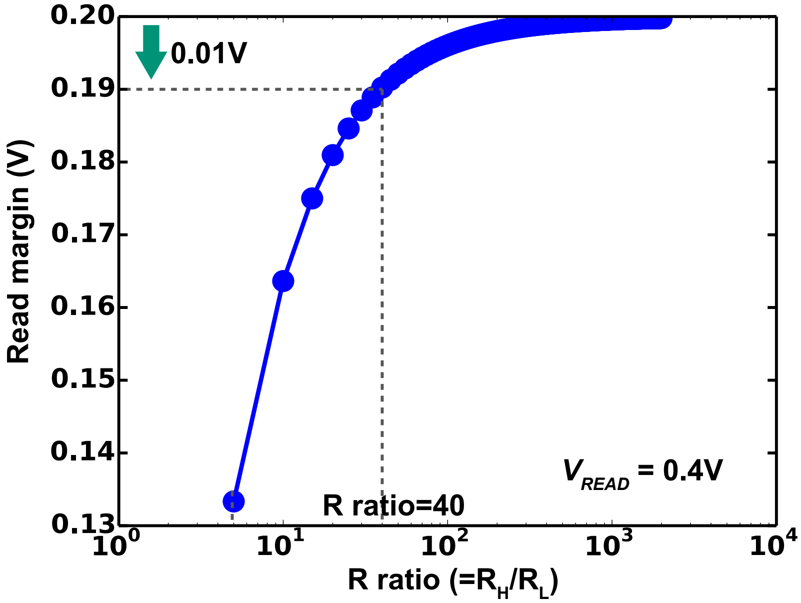


Figure 5.12: Read margin of the differential 2R crosspoint array with different R ratios.

Table 5.2: Comparisons between various memory technologies for cache usage.

Memory	SRAM	eDRAM [6]	STT-RAM [8]	1R crosspoint [50]	D-2R
Cell size (F^2)	100-200	20-50	6-50	4	10
Write energy	Low	Low	High	High	High
Read/write speed	High	Medium	Low	Low	Low
Standby leakage	High	Low	None	None	None
Endurance	High	High	High	Medium	Medium
Retention time	-	$<100\mu$s	Nonvolatile	Nonvolatile	Nonvolatile
Features	High speed	Relatively high speed	High endurance	Small cell size	Higher read margin
Challenge	Leakage	Refresh	Read yield	Sensing error	Power consumption

The read margin is defined as the difference between output voltage and the reference voltage. The D-2R crosspoint structure can maitain the read margin to be above 190mV with R ratiolarger than 40. Even with extremely low R ratio (for example, R_H/R_L=2), the read margin is still large enough to overcome the offset voltage of the sense amplifier.

5.4 Differential RRAM in memory hierarchy

The process variation in advanced technologies prevents the scaling of SRAM bit cells. The area overhead and leakage energy consumption are significant for on-chip last-level cache. To reduce miss penalty by increasing memory capacity, eDRAM provides an option of high-density cache memory. The bit cell area is 20-50 F^2 [6], which is about 4x smaller than an SRAM bitcell. However, an extra process steps to add the capacitors and the need of refresh cycles increase cost and energy. RRAM is another approach to reach high density. Ideally, the bit cell size is $4F^2$ in crosspoint array and $10F^2$ in D-2R crosspoint structure. Moreover, the non-volatility eliminates the leakage current of high-capacity last-level cache. Therefore, nonvolatile cache is attractive for long-standby battery-driven consumer devices. Aside from non-volatility, the potential of stacked layers enables even larger memory capacity. A comparison table of various memory technologies for cache usage is provided in Table 5.2. RRAM endurance of 10^{10} is below the 10^{16} requirement for the conventional L3 cache. However, it meets the needs of a context-switching memory in mobile systems [55]. Contexts of idle applications, which reside in storage to mitigate power consumption, take a long time to recall while users switch over different applications. It requires orders of magnitude more reads than writes and is of growing importance in mobile computing. Parallel read is feasible to further increase the read throughput, which greatly improves the performance for context switch purpose. The endurance can be improved by system or circuit approaches. In addition to the sense-before-write scheme, wear-leveling spreads the write operations evenly across the memory and the built-in test circuit monitors the worn cell status.

5.5 Conclusion

In this chapter, we have proposed a voltage-sensing D-2R crosspoint structure. It enhances the read margin and solves the sensing error due to leakage in a current-sensing scheme. In addition, having the same number of HRS and LRS cells prevents data pattern problems and avoids variable IR drop. To avoid disturbance and limit the leakage current during a write operation, an array segmentation scheme with WL length of four cells wide is adopted. This constrains the write current to below 200μA. The sense-before-write approach prevents cells from having variable LRS resistance values and constrains the cell variability distribution.

A 64-KB D-2R crosspoint RRAM memory can operate at 500 MHz with an average write current of 140μA and an average read current of 16.6μA. The sense-before-write scheme requires two cycles to complete a write operation. In addition, the array segmentation scheme suffers 2× area penalty but effectively reduces the leakage current. An envisioned application as a context memory device presents an attractive application for the D-2R crosspoint RRAM. The equivalent cell size is 10 F^2, much smaller than an SRAM bit cell. Elimination of the standby current outweighs the higher write energy.

Chapter 6

Conclusion

The three topics in the dissertation explore energy-efficient computing in three different aspects: device, circuit, and micro-architecture. By improving sensing robustness with DTSA and increasing the maximum allowable BER with architecture-level assist techniques, an SRAM-based cache is able to operate at a lower voltage enhancing the overall energy efficiency of the processor. The proposed techniques are verified in 28nm test chips, demonstrating low-voltage functionality and high energy efficiency. Emerging nonvolatile memory create new possibilities for memory hierarchy for mostly-off computing applications. Simulations validate the proposed D2R crosspoint RRAM array for reliable accesses.

6.1 Summary of Contributions

The main contributions of this work are:

- A double-tail sense amplifier that operates at a lower voltage than the conventional sense amplifier in SRAM. (Chapter 2)
- A novel architecture-level assist technique, line recycling (LR), that can save 33% of cache capacity loss from line disable (LD) or allow further reduction in Vmin. (Section 3.2)
- The first silicon for the Berkeley Out-of-Order Machine (BOOM), which is also the first high-speed RISC-V OoO processor. (Section 3.5.1)
- A many-port register file implementation with gate-level description and guided placement to avoid routing congestion. (Section 4.6)
- A differential 2R crosspoint RRAM array architecture that enhances the read margin and eliminates data-dependent IR drop to mitigate write variation. (Chapter 5)

6.2 Future Work

The work at different levels of memory design in this dessertation address only a small portion of possible research into resilient and energy-efficient memory. Device, circuit, and architecture co-design is essential for future memory systems to optimize the performance and energy effiency.

- The DTSA allows sensing at a smaller voltage difference (a shorter CLK-SAE delay). To realize the energy savings from sensing at a lower swing, the wordline pulse needs to be shortened to stop the BL from discharing. More benifits could be revealed after optimizing the control circuit of the DTSA.

- Circuit design is dependent on accurate characterization of the CMOS devices. The sense amplifier topologies and circuit-assist techniques should be re-evaluated in sub-10nm nodes. Different I-V characterization and variation may have an impact on the effectiveness of the techinques.

- In this dissertation, the circuit effort and the architecture schemes are assessed separately. Joint development of circuit and architecture techniques could minimize overheads and optimize performance with more degrees of freedom. Moreover, effectiveness of the circuit and architecture approaches can be examined together.

- The architecture-level assist techniques can be adopted more easily if they are implemented as features in the Rocket Chip and can be included simply by changing the configuration. With the help of Chisel library structures, it is possible to make the assist techniques highly parameterized options of the SoC generator. The gate-level description of the register file can also be integrated in Chisel so that it could take parameters, such as number of read/write ports, and generate the configuration file to assist the place-and-route tool.

- The implementation of BROOM described in this dissertation included only two levels of cache hierarchy. The serial accesses to the off-chip DRAM slowed memory transactions. To better estimate performance and energy, both the architecture-level assist techniques and the last-level nonvolatile cache be included in the memory system to estimate the benefit of the performance and energy savings.

- The D2R crosspoint RRAM array has only been validated in simulation. However, fabrication with real devices is extremely important for evaluating new technologies since the device model might not match the real behavior. More information will be revealed after the measurement, such as the resistance distribution of the cells and the read disturb threshold. In addition, challenges such as 3D stacking of multi-layer arrays can only be addressed via silicon implementation.

- Some emerging applications feature unique memory access patterns, leading to new opportunities for acceleration computaton and improving energy efficiency. New non-Von Neumann architectures that move processing closer to memory could significantly reduce the power consumption of data movement.

Energy-efficient and reliable computing will continue to dominate the electronics market. In advanced technology nodes, it is important to save energy while guaranteeing reliable operation. As Moore's law slows, new devices and memory technologies will play an important role in moving beyond the physical limitations of CMOS technology and the current memory hierarchy.

Bibliography

[1] B. Zimmer, P. F. Chiu, B. Nikolić, and K. Asanović, "Reprogrammable redundancy for sram-based cache v_{min} reduction in a 28-nm risc-v processor," *IEEE Journal of Solid-State Circuits*, vol. 52, pp. 2589–2600, Oct 2017.

[2] B. Zimmer, *Resilient Design Techniques for Improving Cache Energy Efficiency.* PhD thesis, EECS Department, University of California, Berkeley, Dec 2016.

[3] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, 2016.

[4] L. Chang, Y. Nakamura, R. Montoye, J. Sawada, A. Martin, K. Kinoshita, F. Gebara, K. Agarwal, D. Acharyya, W. Haensch, *et al.*, "A 5.3 GHz 8T-SRAM with operation down to 0.41 V in 65nm CMOS," in *VLSI Circuits, 2007 IEEE Symposium on*, pp. 252–253, IEEE, 2007.

[5] I. J. Chang, J. J. Kim, S. P. Park, and K. Roy, "A 32 kb 10t sub-threshold sram array with bit-interleaving and differential read scheme in 90 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 44, pp. 650–658, Feb 2009.

[6] F. Hamzaoglu, U. Arslan, N. Bisnik, S. Ghosh, M. B. Lal, N. Lindert, M. Meterelliyoz, R. B. Osborne, J. Park, S. Tomishima, *et al.*, "13.1 a 1gb 2ghz embedded dram in 22nm tri-gate cmos technology," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, pp. 230–231, IEEE, 2014.

[7] D. Takashima, Y. Nagadomi, and T. Ozaki, "A 100 mhz ladder feram design with capacitance-coupled-bitline (ccb) cell," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 3, pp. 681–689, 2011.

[8] K. C. Chun, H. Zhao, J. D. Harms, T. H. Kim, J. P. Wang, and C. H. Kim, "A scaling roadmap and performance evaluation of in-plane and perpendicular mtj based stt-mrams for high-density cache memory," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 598–610, Feb 2013.

[9] Q. Dong, Z. Wang, J. Lim, Y. Zhang, Y. C. Shih, Y. D. Chih, J. Chang, D. Blaauw, and D. Sylvester, "A 1mb 28nm stt-mram with 2.8ns read access time at 1.2v vdd

using single-cap offset-cancelled sense amplifier and in-situ self-write-termination," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, pp. 480–482, Feb 2018.

[10] W. S. Khwa, M. F. Chang, J. Y. Wu, M. H. Lee, T. H. Su, K. H. Yang, T. F. Chen, T. Y. Wang, H. P. Li, M. Brightsky, S. Kim, H. L. Lung, and C. Lam, "A resistance drift compensation scheme to reduce mlc pcm raw ber by over 100*times* for storage class memory applications," *IEEE Journal of Solid-State Circuits*, vol. 52, pp. 218–228, Jan 2017.

[11] R. Simpson, M. Krbal, P. Fons, A. Kolobov, J. Tominaga, T. Uruga, and H. Tanida, "Toward the ultimate limit of phase change in ge2sb2te5," *Nano letters*, vol. 10, no. 2, pp. 414–419, 2009.

[12] R. S. Williams, "How we found the missing memristor," in *Memristors and Memristive Systems*, pp. 3–16, Springer, 2014.

[13] C. C. Chou, Z. J. Lin, P. L. Tseng, C. F. Li, C. Y. Chang, W. C. Chen, Y. D. Chih, and T. Y. J. Chang, "An n40 256k embedded rram macro with sl-precharge sa and low-voltage current limiter to improve read and write performance," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, pp. 478–480, Feb 2018.

[14] M. Sinangil, H. Mair, and A. Chandrakasan, "A 28nm high-density 6T SRAM with optimized peripheral-assist circuits for operation down to 0.6V," in *Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pp. 260–262, 2011.

[15] M. Yabuuchi, K. Nii, Y. Tsukamoto, S. Ohbayashi, Y. Nakase, and H. Shinohara, "A 45nm 0.6V cross-point 8T SRAM with negative biased read/write assist," *IEEE Symp. VLSI Circuits Dig.*, 2009.

[16] J. Chang, Y. H. Chen, W. M. Chan, S. P. Singh, H. Cheng, H. Fujiwara, J. Y. Lin, K. C. Lin, J. Hung, R. Lee, H. J. Liao, J. J. Liaw, Q. Li, C. Y. Lin, M. C. Chiang, and S. Y. Wu, "12.1 a 7nm 256mb sram in high-k metal-gate finfet technology with write-assist circuitry for low-vmin applications," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 206–207, Feb 2017.

[17] E. Karl, Z. Guo, J. Conary, J. Miller, Y.-G. Ng, S. Nalam, D. Kim, J. Keane, U. Bhattacharya, and K. Zhang, "A 0.6V 1.5GHz 84Mb SRAM design in 14nm FinFET CMOS technology," in *Solid- State Circuits Conference - (ISSCC), 2015 IEEE International*, pp. 1–3, Feb 2015.

[18] E. Karl, Y. Wang, Y.-G. Ng, Z. Guo, F. Hamzaoglu, U. Bhattacharya, K. Zhang, K. Mistry, and M. Bohr, "A 4.6 GHz 162Mb SRAM design in 22nm tri-gate CMOS technology with integrated active V MIN-enhancing assist circuitry," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pp. 230–232, IEEE, 2012.

[19] B. Giridhar, N. Pinckney, D. Sylvester, and D. Blaauw, "A reconfigurable sense amplifier with auto-zero calibration and pre-amplification in 28nm cmos," in *Proc. IEEE International Solid-State Circuits Conference*, pp. 242–243, Feb 2014.

[20] M. Khayatzadeh, F. Frustaci, D. Blaauw, D. Sylvester, and M. Alioto, "A reconfigurable sense amplifier with 3x offset reduction in 28nm fdsoi cmos," in *Proc. IEEE Symposium on VLSI Circuits*, pp. 270–271, June 2015.

[21] M. Khayatzadeh, M. Saligane, J. Wang, M. Alioto, D. Blaauw, and D. Sylvester, "A reconfigurable dual-port memory with error detection and correction in 28nm fdsoi," in *Proc. IEEE International Solid-State Circuits Conference*, pp. 310–312, Jan 2016.

[22] M.-Y. Hsiao, "A class of optimal minimum odd-weight-column SEC-DED codes," *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 395–401, 1970.

[23] T. R. Rao and E. Fujiwara, *Error-control coding for computer systems*. Prentice-Hall, Inc., 1989.

[24] Z. Chishti, A. R. Alameldeen, C. Wilkerson, W. Wu, and S.-L. Lu, "Improving cache lifetime reliability at ultra-low voltages," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 42, pp. 89–99, 2009.

[25] A. R. Alameldeen, I. Wagner, Z. Chishti, W. Wu, C. Wilkerson, and S.-L. Lu, "Energy-efficient cache design using variable-strength error-correcting codes," in *Proceedings of the 38th annual international symposium on Computer architecture*, ISCA '11, pp. 461–472, 2011.

[26] W. Chen, S. L. Chen, S. Chiu, R. Ganesan, V. Lukka, W. W. Mar, and S. Rusu, "A 22nm 2.5mb slice on-die l3 cache for the next generation xeon® processor," in *2013 Symposium on VLSI Technology*, pp. C132–C133, June 2013.

[27] M. Huang, M. Mehalel, R. Arvapalli, and S. He, "An energy efficient 32-nm 20-mb shared on-die l3 cache for intel® xeon® processor e5 family," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 1954–1962, Aug 2013.

[28] S. W. Cheng, Y. H. Chang, T. Y. Chen, Y. F. Chang, H. W. Wei, and W. K. Shih, "Efficient warranty-aware wear leveling for embedded systems with pcm main memory," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, pp. 2535–2547, July 2016.

[29] H. Noguchi, K. Ikegami, S. Takaya, E. Arima, K. Kushida, A. Kawasumi, H. Hara, K. Abe, N. Shimomura, J. Ito, S. Fujita, T. Nakada, and H. Nakamura, "7.2 4mb stt-mram-based cache with memory-access-aware power optimization and write-verify-write / read-modify-write scheme," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 132–133, Jan 2016.

[30] K. Nii, M. Yabuuchi, Y. Tsukamoto, S. Ohbayashi, S. Imaoka, H. Makino, Y. Yamagami, S. Ishikura, T. Terano, T. Oashi, K. Hashimoto, A. Sebe, G. Okazaki, K. Satomi, H. Akamatsu, and H. Shinohara, “A 45-nm bulk cmos embedded sram with improved immunity against process and temperature variations,” *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 180–191, Jan 2008.

[31] M. Khellah, Y. Ye, N. Kim, D. Somasekhar, G. Pandya, A. Farhang, K. Zhang, C. Webb, and V. De, “Wordline bitline pulsing schemes for improving sram cell stability in low-vcc 65nm cmos designs,” in *2006 Symposium on VLSI Circuits, 2006. Digest of Technical Papers.*, pp. 9–10, June 2006.

[32] A. Abidi and H. Xu, “Understanding the regenerative comparator circuit,” in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 1–8, Sept 2014.

[33] M. Abu-Rahma, Y. Chen, W. Sy, W. L. Ong, L. Y. Ting, S. S. Yoon, M. Han, and E. Terzioglu, “Characterization of sram sense amplifier input offset for yield prediction in 28nm cmos,” in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 1–4, Sept 2011.

[34] N. Verma and A. P. Chandrakasan, “A 256 kb 65 nm 8t subthreshold sram employing sense-amplifier redundancy,” *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 141–149, Jan 2008.

[35] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, “Razor: a low-power pipeline based on circuit-level timing speculation,” in *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36.*, pp. 7–18, Dec 2003.

[36] D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta, “A double-tail latch-type voltage sense amplifier with 18ps setup+hold time,” in *Proc. IEEE International Solid-State Circuits Conference*, pp. 314–605, Feb 2007.

[37] M. Miyahara, Y. Asada, D. Paik, and A. Matsuzawa, “A low-noise self-calibrating dynamic comparator for high-speed adcs,” in *IEEE Asian Solid-State Circuits Conference*, pp. 269–272, Nov 2008.

[38] P.-F. Chiu, C. Celio, K. Asanovic, D. Patterson, and B. Nilolic, “An Out-of-Order RISC-V Processor with Resilient Low-Voltage Operation in 28nm CMOS,” in *VLSI Circuits (VLSIC), 2018 Symposium on*, June 2018.

[39] M. Clinton, H. Cheng, H. Liao, R. Lee, C. W. Wu, J. Yang, H. T. Hsieh, F. Wu, J. P. Yang, A. Katoch, A. Achyuthan, D. Mikan, B. Sheffield, and J. Chang, “12.3 a low-power and high-performance 10nm sram architecture for mobile applications,” in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 210–211, Feb 2017.

[40] Y. Lee, H. Yoo, J. Jung, J. Jo, and I. C. Park, "A 2.74-pj/bit, 17.7-gb/s iterative concatenated-bch decoder in 65-nm cmos for nand flash memory," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 2531–2540, Oct 2013.

[41] J. Chang, M. Huang, J. Shoemaker, J. Benoit, S.-L. 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 Journal of Solid-State Circuits*, vol. 42, pp. 846–852, April 2007.

[42] C. Celio, P.-F. Chiu, B. Nikolić, D. A. Patterson, and K. Asanović, "Boomv2: an open-source out-of-order risc-v core," in *First Workshop on Computer Architecture Research with RISC-V (CARRV)*, 2017.

[43] K. Asanovic, R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, D. Dabbelt, J. Hauser, A. Izraelevitz, *et al.*, "The rocket chip generator," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17*, 2016.

[44] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avižienis, J. Wawrzynek, and K. Asanović, "Chisel: constructing hardware in a scala embedded language," in *Proceedings of the 49th Annual Design Automation Conference*, pp. 1216–1225, ACM, 2012.

[45] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[46] H. M. Cook, A. S. Waterman, and Y. Lee, "Tilelink cache coherence protocol implementation," *White Paper*, 2015.

[47] S. Gal-On and M. Levy, "Exploring coremarka benchmark maximizing simplicity and efficacy," *The Embedded Microprocessor Benchmark Consortium*, 2012.

[48] C. Celio, "Characterizing memory hierarchies of multicore processors," 2011.

[49] Y. Li, S. Lee, K. Oowada, H. Nguyen, Q. Nguyen, N. Mokhlesi, C. Hsu, J. Li, V. Ramachandra, T. Kamei, M. Higashitani, T. Pham, M. Honma, Y. Watanabe, K. Ino, B. Le, B. Woo, K. Htoo, T. Y. Tseng, L. Pham, F. Tsai, K. h. Kim, Y. C. Chen, M. She, J. Yuh, A. Chu, C. Chen, R. Puri, H. S. Lin, Y. F. Chen, W. Mak, J. Huynh, J. Chan, M. Watanabe, D. Yang, G. Shah, P. Souriraj, D. Tadepalli, S. Tenugu, R. Gao, V. Popuri, B. Azarbayjani, R. Madpur, J. Lan, E. Yero, F. Pan, P. Hong, J. Y. Kang, F. Moogat, Y. Fong, R. Cernea, S. Huynh, C. Trinh, M. Mofidi, R. Shrivastava, and K. Quader, "128gb 3b/cell nand flash memory in 19nm technology with 18mb/s write rate and 400mb/s toggle mode," in *2012 IEEE International Solid-State Circuits Conference*, pp. 436–437, Feb 2012.

[50] E. Ou and S. S. Wong, "Array architecture for a nonvolatile 3-dimensional cross-point resistance-change memory," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 9, pp. 2158–2170, 2011.

[51] A. Kawahara, R. Azuma, Y. Ikeda, K. Kawai, Y. Katoh, Y. Hayakawa, K. Tsuji, S. Yoneda, A. Himeno, K. Shimakawa, *et al.*, "An 8 mb multi-layered cross-point reram macro with 443 mb/s write throughput," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 178–185, 2013.

[52] M. Yoshimoto, K. Anami, H. Shinohara, T. Yoshihara, H. Takagi, S. Nagao, S. Kayano, and T. Nakano, "A divided word-line structure in the static ram and its application to a 64k full cmos ram," *IEEE Journal of Solid-State Circuits*, vol. 18, no. 5, pp. 479–485, 1983.

[53] J. Ahn and K. Choi, "Lower-bits cache for low power stt-ram caches," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, pp. 480–483, IEEE, 2012.

[54] C. Cagli, J. Buckley, V. Jousseaume, T. Cabout, A. Salaun, H. Grampeix, J. Nodin, H. Feldis, A. Persico, J. Cluzel, *et al.*, "Experimental and theoretical study of electrode effects in hfo 2 based rram," in *Electron Devices Meeting (IEDM), 2011 IEEE International*, pp. 28–7, IEEE, 2011.

[55] H. Kim, N. Agrawal, and C. Ungureanu, "Revisiting storage for smartphones," *ACM Transactions on Storage (TOS)*, vol. 8, no. 4, p. 14, 2012.