Low-Leakage Asymmetric-cell SRAM

by

Navid Azizi

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

© Copyright by Navid Azizi 2003

Low-Leakage Asymmetrical-cell SRAM

Navid Azizi

Master of Applied Science, 2003
Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

In this work a novel family of asymmetric dual-V_t Static Random Access Memory (SRAM) cell designs that reduce leakage power in caches while maintaining low access latency is introduced. The asymmetrical cell family is built on the following premise: select a preferred stored value and weaken, by increasing the threshold voltage, only those transistors necessary to drastically reduce leakage when this value is stored. The designs exploit the strong bias towards zero at the bit level exhibited by the memory values of ordinary programs. Relative to conventional symmetric high-performance cells, the asymmetric cells offer significant leakage reduction in the zero state and in some cases also in the one state albeit to a lesser extent. A unique sense-amplifier, in combination with dummy bitlines, enables read times to be comparable to conventional symmetric cells. With one cell design, leakage is reduced by 7 times (in the zero state) with no performance degradation, but with a stability degradation of 6%. Another cell design reduces leakage by 2 times (in the zero state) with no performance or stability loss. An alternative cell design reduces leakage by 58 times (in the zero state) with a performance degradation of 1% and an area increase of 2% and no stability degradation.

Acknowledgments

I would like to gratefully acknowledge the enthusiastic supervision of my advisors Professor Farid N. Najm, and Andreas Moshovos for their continuous guidance, and inspiration.

Life during the progression of this Masters program was very interesting. Not only was the Masters itself challenging, but it was also very enjoyable to come to school everyday. The most important discovery during this time, however, was not the research that I was accomplishing for my thesis. I had always seen the Baha'i Faith as a very logical step in progressive revelation, and the principles of unity and equality were very noble to me. However, during this past year, I went through some very challenging times personally, and I discovered what faith and spirituality really is. The Baha'i Faith and the Love of God have taken a complete different type of hold in my heart, changing me for the better and opened me to a larger perspective of life.

There are a couple of people that I would like to thank individually. These people I consider my closest friends and advisors. First and foremost, I would like to thank Igor Arsovski, whose friendship has always been unparralled. Igor, buddy, you know I couldn't have done this without you keeping me sane in times of difficulties.

In the short time I have known Kamran Zamanian, he has not only become a role-model for me, but a very close friend. I would like to thank Kamran for all the guidance he has given me in the professional, spiritual, and social spheres.

Then there is Babak Pirmoradi. How can I describe my thankfulness towards Babak. I don't think I can. Even though, we are at different ends of the spectrum, the friendship we have can never be replaced or substituted by anyone else.

I am grateful to many people at the University who have assisted me in the course of this work including Kostas Pagiamtzis, who, with complete humility, has always

accepted to help when problems have arisen; and Paul Kundarewich, who has taught me to understand the need for more than just education in the education system.

I'd also like to thank all my fellow students in LP392 and now Bayen 5000 including Trevis Chandler, Joyce Wong, Lesley Shannon, Warren Gross, Ahmad Darabiha, Jason Anderson, Andy Ye, Tomasz Czajkowski, Yadollah Eslami, Rubil Ahmadi, and Anish Alex for their friendship. I truly look forward to coming to LP392 each and every day.

Thanks to my coffee-house friends Kamran Zamanian, Afsoon Houshidari, Senthil Shanmugasundaram, and Luke Chamandy for their friendship and interesting points of discussion (and actually putting on a better show than "Friends" on Thursday nights.)

Thanks to my Baha'i friends Nassim Rouhani, who has taught me how we should really treat one another, Negar Arjomand, Hooman Katirai, Navid Sabet, Shadi Katirai, Nima Sobhani, Iman Ferdowsi, Yannick Katirai, Alex Arjomand and Mahbod Mahmodani for making CABS so successful and awesome. Fridays are always great. UofT CABS rules!!

I'm running out of space, so I'll be short, but my following friends really do need more space. You guys have made the past years unforgettable: Sameh Barsoum, Samira Naraghi, Babak Vaez, Nima Ghazizadeh, Negin Shobeiri, Azadeh Kushki and Houman Namiranian.

I would also like to acknowledge financial support provided by the National Science and Engineering Research Council and the Semiconductor Research Corporation, and CAD tool access and support by Canadian Microelectronics Corporation.

Finally, my parents, Fereydoon and Shohreh, have always encouraged me and guided me to independence, wanting only the best for me. I am grateful to them for their complete love and support. Without them I would be absolutely nothing. I would also like to thank my brothers, Omid and Paymon, who have always kept me looking forward towards the next day. Without their companionship, life would not be the same.

Contents

List of Figures xi

List of Tables xv

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Objective . . . 2
- 1.3 Thesis Organization . . . 3

2 Background 5
- 2.1 Leakage Currents . . . 5
- 2.2 SRAM Cells . . . 6
- 2.3 Stability . . . 7
  - 2.3.1 Static Noise Margin . . . 7
  - 2.3.2 I_{trip}/I_{read} . . . 8
- 2.4 Circuit Level Techniques to Reduce Power in Caches . . . 10
  - 2.4.1 Voltage Scaling . . . 10
  - 2.4.2 Replica Technique . . . 11
  - 2.4.3 Dual-V_t SRAM Cells . . . 11
  - 2.4.4 Gated-V_{DD} . . . 12
  - 2.4.5 Data Retention Gated-Ground (DRG)-Cache . . . 13
- 2.5 Architectural Level Techniques to Reduce Power in Caches . . . 15
  - 2.5.1 Block Buffering . . . 15
  - 2.5.2 RAM subbanking . . . 15
  - 2.5.3 Turning off Cache Lines . . . 17
  - 2.5.4 Zero Byte Compression . . . 18
- 2.6 Conclusion . . . 19

3 Cell Level Design 21
- 3.1 Introduction . . . 21
- 3.2 Methodology . . . 22

3.2.1 Leakage . . . 23
3.2.2 Performance . . . 23
3.2.3 Stability . . . 24
3.3 Nine Asymmetrical Cells . . . 25
3.4 Stability . . . 31
3.4.1 Static Noise Margin . . . 31
3.4.2 I_{trip}/I_{read} . . . 35
3.5 Stability Improved Cells through Threshold Voltage . . . 39
3.5.1 Leakage . . . 40
3.5.2 Performance . . . 41
3.5.3 Stability . . . 41
3.6 Stability Improved Cells through Transistor Sizing . . . 42
3.6.1 Leakage . . . 44
3.6.2 Performance . . . 45
3.6.3 Stability . . . 45
3.7 Stability at Different Supply Voltages . . . 46
3.8 Supply Voltage Scaling . . . 47
3.9 Summary . . . 50

4 Sense Amplifier/SRAM 53
4.1 Introduction . . . 53
4.2 Sensing Architecture . . . 54
4.3 Circuit Operation . . . 55
4.4 Simulated SRAM . . . 58
4.4.1 Sense Times . . . 58
4.4.2 Write Times . . . 64
4.4.3 Power Consumption . . . 64
4.4.4 Sensitivity . . . 66
4.5 Summary . . . 66

5 Architectural Enhancements 69
5.1 Introduction . . . 69
5.2 General Issues . . . 70
5.2.1 Methodology . . . 70
5.2.2 Benchmarks Used . . . 72
5.3 Level 1 Caches . . . 73
5.3.1 Bit Value Distribution of Ordinary Programs . . . 73
5.3.2 Bit Values in Live Blocks . . . 76
5.3.3 Selective Inversion . . . 78

5.3.4 Leakage Power Reduction 80
5.4 SRAM Based Structures 82
5.4.1 General Purpose Register Files 84
5.4.2 Floating-Point Register Files 85
5.4.3 Branch Predictors 87
5.4.4 Branch Address Predictors 88
5.5 Selective Inversion Circuits 90
5.5.1 Majority Counter 91
5.5.2 Read Inversion 92

6 Conclusion **95**
6.1 Summary and Conclusions 95
6.2 Future Work 97

References **99**

List of Figures

2.1 SRAM cell . . . 7
2.2 Back-to-back inverters comprising SRAM cell where static-noise voltage source are included . . . 8
2.3 SNM of an SRAM cell during read access . . . 9
2.4 Benchmark for I_{trip}/I_{read} . . . 9
2.5 Benchmark for measuring V_{drop} . . . 10
2.6 Symmetric Dual-V_t cells . . . 12
2.7 Gated V_{DD} SRAM cell . . . 14
2.8 DRG SRAM cell . . . 14
2.9 2-way Set-Associate Cache with 2 subbanks per Data bank . . . 16

3.1 Transistors that dissipate leakage when an SRAM is holding a '0' . . . 23
3.2 Basic Asymmetric SRAM cell . . . 25
3.3 Leakage Improved 2 Cell . . . 26
3.4 Leakage Improved 3 Cell . . . 26
3.5 Speed Improved 1 Cell . . . 27
3.6 Speed Improved 2 Cell . . . 27
3.7 Speed Improved 3 Cell . . . 28
3.8 Special Precharge Cell . . . 28
3.9 Graphical Representation of Asymmetrical Leakage Characteristics of all cells . . . 29
3.10 SNM of the LE cell . . . 32
3.11 SNM of the SE cell . . . 33
3.12 SNM of LE cell pinching off . . . 34
3.13 SNM of SE cell pinching off . . . 34
3.14 Normal Scores for SNM . . . 36
3.15 Normal Scores for $I_{\text{trip}}/I_{\text{read}}$ . . . 38
3.16 SLE cell . . . 39
3.17 SSE cell . . . 39
3.18 SNM for SLE cell . . . 40
3.19 SNM for SSE cell . . . 41

3.20 SNM for RSE cell 43
3.21 SNM for RLE cell 44
3.22 Mean SNM for different supply voltages for LE derived cells 48
3.23 Mean SNM for different supply voltages for SE derived cells 48
3.24 Leakage as supply voltage and threshold value are scaled down 49
3.25 Read times at different Supply Voltages 50
3.26 Write times at different Supply Voltages 50

4.1 (a) Simple Sense Amplifier. (b) New Sense Amplifier 54
4.2 Dummy Column Architecture 55
4.3 Approximate Sense Amplifier Circuit when reading a '0' 57
4.4 Approximate Sense Amplifier Circuit when reading a '1' 57
4.5 Layout of Simulated SRAM 59
4.6 Layout of SRAM cell 60
4.7 Layout of Sense Amplifier 60
4.8 Layout of Decoder 61
4.9 Breakdown of Memory Access Time 62
4.10 Sense times during a read cycle, i.e., the 3rd component of Fig. 4.9. . . 62
4.11 Write Times 65
4.12 Sensitivity of New Sense Amplifier 66

5.1 Fraction of cache bits that are zeroes for L1D and L1I for the ALPHA binaries for 32-Kbyte caches 74
5.2 Memory space breakdown of zero bits for L1D 74
5.3 Fraction of cache bits that are zeroes for L1D and L1I for the ALPHA binaries for 64-Kbyte caches. 76
5.4 Comparing the bit value distribution across the ALPHA and PISA binaries for 32-Kbyte L1D caches. 77
5.5 Zero bit fraction with 32K L1 caches and for only those blocks that hold live data. 78
5.6 Memory space breakdown for live data. 79
5.7 Selective inversion at the byte level 81
5.8 Data and instruction cache leakage power with the asymmetric cells relative to the RV conventional cache 83
5.9 Fraction of GPR bits that are zeros 84
5.10 Fraction of GPR bits that are zeroes under different inversion policies . 85
5.11 Fraction of FPR bits that are zeros 86
5.12 Fraction of bits in branch predictors that are zeros 88
5.13 Fraction of bits for Branch Predictors under different codings 89

5.14 Fraction of zero bits in BTB Targets 90
5.15 Fraction of zero bits in BTB Tags 91
5.16 Tally Circuit 92
5.17 Circuit that performs inversion during read 93

List of Tables

3.1 Summary of Leakage Reduction for all asymmetric cells 30
3.2 Summary of bitline discharge times for all asymmetric cells 30
3.3 Summary of write times for all asymmetric cells 31
3.4 Nominal SNM 34
3.5 Worst-Case SNM 35
3.6 Mean and Standard Deviation during Monte-Carlo Analysis for SNM . 35
3.7 Nominal $I_{\text{trip}}/I_{\text{read}}$ 36
3.8 Worst-Case $I_{\text{trip}}/I_{\text{read}}$ 37
3.9 Mean and Standard Deviation during Monte-Carlo Analysis for $I_{\text{trip}}/I_{\text{read}}$ 38
3.10 Percent Improvement over RV cell for stability-improved cells under SNM test 42
3.11 Percent Improvement over RV cell for stability-improved cells under $I_{\text{trip}}/I_{\text{read}}$ test 42
3.12 Percent Improvement over RV cell for resized cells under SNM test . . 46
3.13 Percent Improvement over RV cell for resized cells under $I_{\text{trip}}/I_{\text{read}}$ test 46
3.14 Summary results for all the cells. 51

5.1 Base Processor Configuration 71
5.2 Applications used in the studies 72
5.3 Inversion Policies evaluated for SRAM based structures 84

6.1 Summary results for all the cells. 96
6.2 Summary results for all the cells, showing expected leakage. 97

List of Acronyms

ACC Asymmetrical Cell Cache

ALU Arithmetic and Logic Unit

BA Basic Asymmetric

BL bitline

BLB bitline-bar

BSIM Berkely Short Channel Insulated Gate Field Effect Transistor Model

BTB Branch Target Buffer

FPR Floating-Point Register File

GPR General Purpose Register File

HV High-V_t

LE Leakage Enhanced

LI Leakage Improved

L1I Level-one Instruction

L1D Level-one Data

L2 Level-two

MCML MOS Current Mode Logic

MOSFET Metal Oxide Semiconductor Field Effect Transistor

NMOS Negative-Channel Metal Oxide Semiconductor

PMOS Positive-Channel Metal Oxide Semiconductor

RISC Reduced Instruction Set Computer

RLE Resized-Leakage Enhanced

RSE Resized-Speed Enhanced

RV Regular-V_t

SI Speed Improved

SE Speed Enhanced

SNM Static Noise Margin

SLE Stability-Leakage Enhanced

SP Special Precharge

SSE Stability-Speed Enhanced

SRAM Static Random Access Memory

1 Introduction

All things must needs have a cause, a motive power, an animating principle

Baha'u'llah

1.1 Motivation

Technology trends have resulted in static power dissipation (leakage) emerging as a first-class design consideration in high-performance processor design. Historically, architectural innovations for improving performance relied on exploiting ever larger numbers of transistors operating at higher frequencies. To keep the resulting switching power dissipation at bay, successive technology generations have relied on reducing the supply voltage. In order to maintain performance, however, a corresponding reduction in the transistor threshold voltage (V_t) is required. Since the Metal Oxide Semiconductor Field Effect Transistor (MOSFET) sub-threshold leakage current increases exponentially with a reduced V_t, static power dissipation has grown to be a significant fraction of overall chip power dissipation in modern, deep-sub-micron ($< 0.18\mu$m) processes. Moreover, it is expected to grow by a factor of five every new chip generation [Bor99]. For processors it is estimated that in 0.10μm technology, static power will account for approximately 50% of the total chip power [KRK+00].

Since leakage power is proportional to the number of transistors, much of the recent work in reducing static power has focused on Static Random Access Memory (SRAM)-based structures, such as the caches, that comprise the vast majority of on-chip transistors. Existing circuit-level leakage reduction techniques are oblivious

to program behavior and trade off performance for reduced leakage where possible, e.g., [HYK+00]. Combined circuit- and architecture-level techniques reduce leakage for those parts of the on-chip caches that remain unused for long periods of time (thousands of cycles) [KHM01][YPF+01][ZTRC01], but the mechanisms that identify which cache parts will be unused and enable leakage reduction incur considerable power and performance overheads that have to be amortized over long periods of time. Furthermore, these methods are not effective when most of the cache is actively used.

1.2 Objective

The goal of this research is to develop a circuit level technique that takes advantage of program behavior to reduce leakage with no performance degradation. A new SRAM cell composed of an asymmetrical configuration of dual-V_t transistors has been designed to accomplish this goal.

The objectives of this research include:

1. Develop a novel family of low-leakage asymmetric SRAM cells.

2. Determine the merits of the asymmetric SRAM cells with respect to leakage, performance, and stability.

3. Develop a novel sense amplifier design that exploits the asymmetric nature of our cells to offer cell read times that are on par with conventional symmetric SRAM cells.

4. Evaluate a cache design that is based on Asymmetrical Cell Cache (ACC)s and demonstrate that relative to a conventional cache, it offers drastic leakage reduction while maintaining high performance and comparable noise margins and stability.

1.3 Thesis Organization

This thesis is organized as follows: chapter 2 contains background information on the sources of leakage current; an overview of SRAM cells and their stability; methods for reducing power in SRAMs; and previous work done on the dynamic distribution of bit values in cache accesses. In Chapter 3, the basic asymmetrical cell design along with other asymmetrical cells that have better leakage, performance and stability characteristics are introduced. A unique sense amplifier that allows asymmetrical cells to have symmetric read access times is presented in Chapter 4. Chapter 5 describes bit level bias distributions in various SRAM based structures in microprocessors and evaluates the leakage savings when asymmetrical cells are used. Finally, Chapter 6 concludes and suggests directions for future research.

2 Background

Truthfulness is the foundation of all human virtues

'Abdu'l Baha

2.1 Leakage Currents

The leakage current in MOSFETs depends on various process parameters, the transistor size and the quiescent state of the circuit. This sub-threshold leakage can be modeled by the Berkely Short Channel Insulated Gate Field Effect Transistor Model (BSIM) equation [CJWR98]:

$$I_{subth} = Ae^{\frac{q}{nkT}(V_{GS}-V_t)}(1 - e^{\frac{qV_{DS}}{kT}}) \tag{2.1}$$

where

$$V_t = V_{TH0} - \gamma' V_{SB} + \eta V_{DS} \tag{2.2}$$

and

$$A = \mu_0 C'_{OX} \frac{W}{L_{eff}} (\frac{kT}{q})^2 e^{1.8} \tag{2.3}$$

In these equations V_{TH0} is the zero bias threshold voltage, γ' is the linearized body effect coefficient, η represents the effect of V_{DS} on threshold voltage, μ_0 is the zero bias mobility, n is the sub-threshold swing coefficient, C'_{OX} is the gate oxide capacitance, k is Boltzmann's constant, q is the electron charge, T is the temperature in degrees Kelvin and W and L_{eff} are the width and effective length of the transistor respectively [CJWR98].

From equations 2.1 it can be seen that leakage current is exponentially dependent on the threshold voltage; therefore, the body effect and other threshold voltage variations can become very important when determining the leakage current within a circuit. While the threshold voltage has an exponential relationship to leakage current, the size of transistors only effects the leakage current linearly through the A coefficient. Leakage is also very sensitive to temperature, doubling for every 8° to 10^oK increase [CJWR98].

One method of reducing leakage currents is to stack transistors in series. The stacking of transistors can exponentially decrease sub-threshold leakage in two ways [CJWR98]. First the source nodes of the stacked transistors are no longer at ground, and, therefore, the source-bulk voltage, V_{SB} becomes larger, thus increasing the effective threshold voltage through the *body effect* and lowering the leakage current. Secondly, a slight reverse bias between the gate and source (V_{GS}) of the transistors is induced when stacked transistors are turned off, and this reduces the effective driving voltage on the gate, and again decreases the sub-threshold leakage current [CJWR98].

2.2 SRAM Cells

SRAM cells are usually used to implement memories that require short access times, low power dissipation, and tolerance to environmental conditions [Har00]. SRAM cells conventionally have 6-transistors as seen in Fig. 2.1, Transistors P2, N2, P1 and, N1 comprise a pair of cross-coupled CMOS inverters that use positive feedback to store a value. Transistors N3 and N4 are two pass transistors that allow access to the storage nodes for reading and writing.

To write a value into an SRAM cell, the new value and its complement are driven on the bitlines, and then the wordline is raised. The new value will overwrite the old value, since the bitlines are actively driven by write circuitry. To read a value from an SRAM cell, the bitlines are precharged high, and then the wordline is raised turning on the pass transistors. Because one of the internal storage nodes is low, one of the bitlines starts discharging. A sense amplifier, which is connected to the bitlines, senses which of the bitlines is discharging and reads the stored value.

2.3 Stability

The stability of SRAM cells determines its soft-error rate and its sensitivity to process tolerances and operating conditions [SLL87]. In many cases, the stability of the cell is a critical factor to obtain a desired yield and to lower the cost of the chip. Many different tests and methods exist that try to capture different aspects of the cell's stability. In this section, two of these methods, the Static Noise Margin (SNM) and the $I_{\text{trip}}/I_{\text{read}}$ method will be presented.

2.3.1 Static Noise Margin

Static noises are DC disturbances such as offsets and mismatches due to process variations and changes in operating conditions [SLL87]. The SNM of an SRAM cell is the maximum value of DC disturbances that can be tolerated before the cell's storage value is flipped. Fig. 2.2 shows the storage elements of the SRAM cell where the static noise sources, V_n are included explicitly.

The SNM of SRAM cells can be determined graphically by drawing and mirroring the inverter voltage characteristics and finding the maximum possible square between them as shown in Fig. 2.3 [SLL87]. (V_{right} and V_{left} in Fig. 2.3 are the input voltages of the bottom and top inverters in Fig. 2.2 respectively.) The SNM can be thought as the noise voltage necessary at each of the cell storage nodes to shift the curve of the two

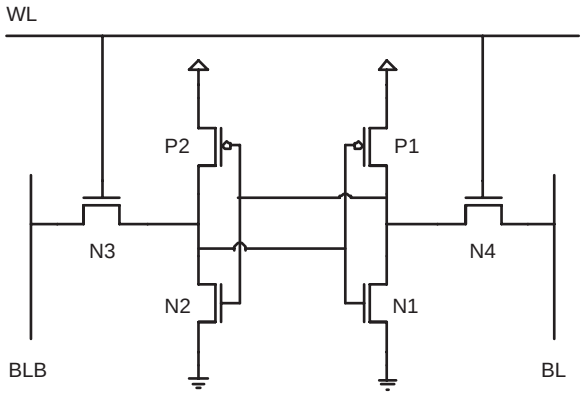


Figure 2.1: SRAM cell

cell inverters vertically or horizontally so that they intersect at only one point, where the cell no longer can reliably store a value [BTM01]. The SNM is measured during the read, since the cell is most vulnerable when the pass-transistors are conducting[1].

2.3.2 I_{trip}/I_{read}

Another measure of SRAM stability is the $I_{\text{trip}}/I_{\text{read}}$ method presented in [HYK+00]. I_{trip} is the measured current through the pull-down NMOS when the state of the cell is being reversed by injecting an external current I_{test}. I_{trip} is measured by using the testbench shown in Fig. 2.4. In the testbench V_{drop} emulates the excessive bitline leakage that can cause the bitline voltage to decrease when all cells that share the same bitlines, as the cell being read, hold a '1,' as shown in Fig. 2.5 [HYK+00]. V_{drop} is included since a reduction in voltage at the stored '1' node degrades the current sinking capacity of the pull-down NMOS, which lowers I_{trip}. Furthermore, to obtain the worst-case bitline droop, channel lengths for all pass transistors are reduced by 3σ and noise is applied to all the wordlines that are off.

I_{read} is the maximum current through the pass transistor on the stored '0' side of the cell during a read [HYK+00]. The ratio of I_{trip} to I_{read} shows how much current, that is passed through the pass-transistor NMOS, can be sunk through the pull-down NMOS before the state of the cell is flipped. Larger values of $I_{\text{trip}}/I_{\text{read}}$ indicate higher stability since more current is needed to flip the state of the cell.

[1] During a read the '0' storage node rises to a higher voltage due to the voltage division along the access and inverter pull-down Negative-Channel Metal Oxide Semiconductor (NMOS) transistors from the precharged bitlines and ground.

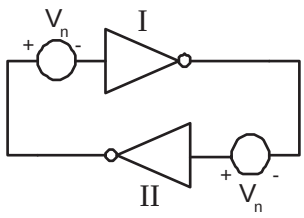


Figure 2.2: Back-to-back inverters comprising SRAM cell where static-noise voltage source are included

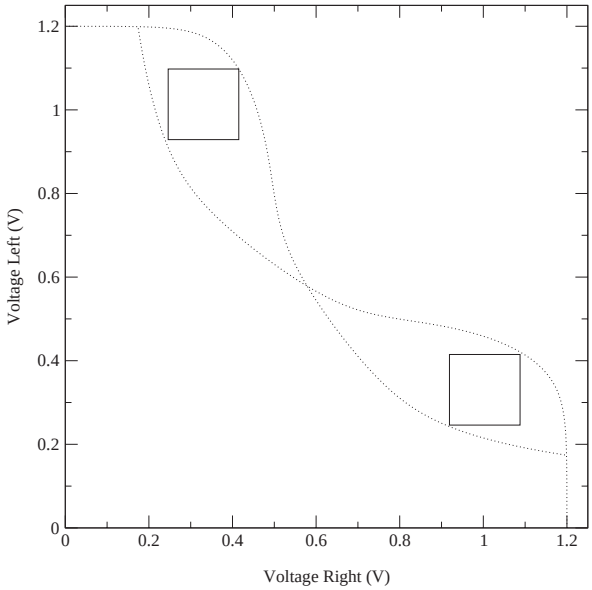


Figure 2.3: SNM of an SRAM cell during read access

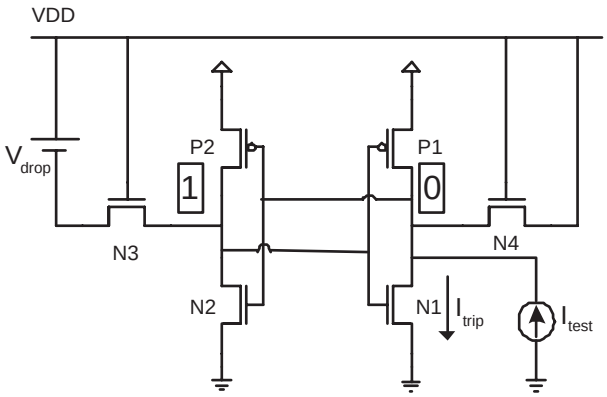


Figure 2.4: Benchmark for I_{trip}/I_{read}

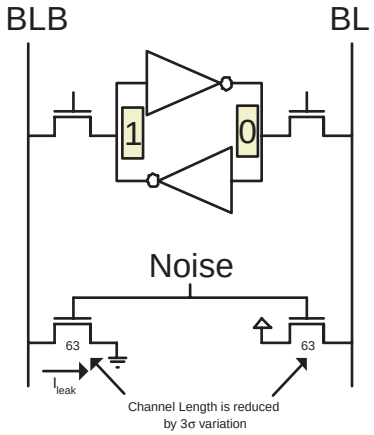


Figure 2.5: Benchmark for measuring V_{drop}

2.4 Circuit Level Techniques to Reduce Power in Caches

The following subsections will review the previous research in reducing the dynamic and static power consumption of SRAMs based on circuit-level techniques.

2.4.1 Voltage Scaling

By varying the supply voltage and clock frequency on demand, dynamic voltage scaling tries to provide high performance when it is required and low energy consumption during periods of standby [BB00]. Since dynamic energy is a function of the supply voltage squared, (i.e. $E_d \propto V_{DD}^2$) lowering the supply voltage can lower the dynamic energy consumption dramatically. A complete microprocessor with cache is developed in [BB00] where the impact of a changing supply voltage is measured on SRAM cells and on sense amplifiers. [BB00] cites that the energy consumption varied from 0.54mW/MIP in standby to 5.6mW/MIP when the processor was running at its fastest speed.

2.4.2 Replica Technique

Due to the variability in the threshold voltage of transistors due to process variations, large delay margins must be designed in SRAM arrays [AH98]. These large margins become the source of power inefficiencies along the bitline and through the sense amplifier. [AH98] tries to minimize the power consumed along the bitlines and the sense amplifiers by using a self-timed approach. Self-timing is used to pulse the wordlines to limit their bitline swing to the minimum needed by the sense amplifiers and to clock the sense amplifiers and minimize the time that they are ON. To clock the sense amplifiers, dummy columns are used to match the clock path to the data path, thus allowing the sense amplifiers to turn on only when the data has arrived and thus limiting the sense amplifiers power consumption [AH98].

The design presented in [AH98] limits the dynamic energy consumption of the periphery of the SRAM, with no degradation to the performance of the SRAM. Moreover, due to the self-timed approach, large margins are not designed into the SRAM and the fabricated SRAM is able to be perform at the fastest possible rate. The self-timed approach, however, does nothing to limit the large static power dissipation of SRAM cells.

2.4.3 Dual-V_t SRAM Cells

Due to the unavoidable usage of low threshold voltage transistors in speed critical paths of microprocessors, the divergence between the logic delay within microprocessors and memory delay is expanding [HYK+00]. This divergence has necessitated the use of low-V_t devices within the cache, but this use of low-V_t devices causes excessive leakage and a reduction in SRAM cell stability.

Different dual-V_t SRAM cells are studied for leakage, performance and stability in [HYK+00]. The SRAM cells studied, some of which are shown in Fig. 2.6, are composed of a symmetric composition of transistors. The dual-V_t cells compose a continuum between the high-performance, high static power dissipation of an SRAM cell solely composed of low-V_t transistors and the low-performance, low static power dissipation of a high-V_t cell. Thus, the design in [HYK+00] concedes low static power

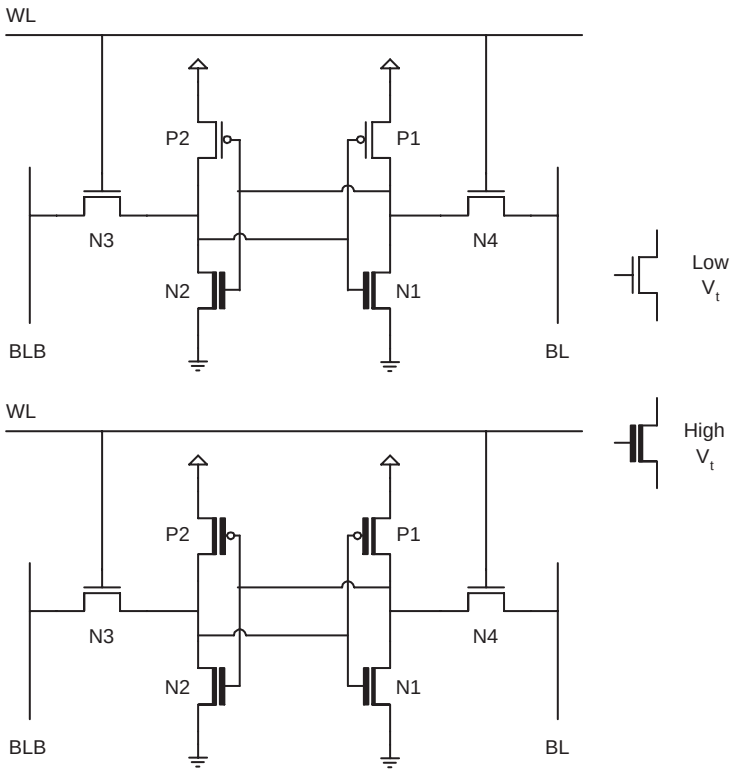


Figure 2.6: Symmetric Dual-V_t cells

dissipation for higher performance.

2.4.4 Gated-V_{DD}

As noted in section 2.1, the stacking of transistors can effectively reduce the transistor leakage current. In [PYF+00], either an additional NMOS transistor is inserted between ground and the SRAM pull-down transistor (as seen in Fig. 2.7), or an additional Positive-Channel Metal Oxide Semiconductor (PMOS) transistor is inserted between the power supply and the SRAM PMOS transistor. When the gated V_{DD} control signal turns the gating transistor off, the leakage current is reduced due to the stacking effect. The gating transistor is shared between many SRAM cells, and thus whole portions of the cache can effectively be turned off, reducing leakage power consumption. One disadvantage to this method is that all data stored in sections that are turned off will

be lost.

[PYF+00] states that an SRAM with a gated PMOS can save 86% of the static power dissipation when the gating transistor is off, with no performance degradation, and an NMOS gating transistors can save 97% of the static power dissipation when the gating transistor is off with a 8% performance penalty.

Similar to [HYK+00] and [BB00] this circuit technique trades performance for leakage power savings. Furthermore, SRAM cell gating is only useful when architecture level decisions can be made that determine when data is no longer needed so that the portions of the cache can be turned off. The process of turning cache lines off and on can incur considerable dynamic energy consumption due to the large capacitance along the Gated-V_{DD} control lines.

2.4.5 Data Retention Gated-Ground (DRG)-Cache

The gating idea in [PYF+00] has been extended in [ALR02] where the Gated-V_{DD} control is supplied by the wordline, as shown in Fig. 2.8, instead of an architectural level decision. In contrast to the notion of turning off whole cache blocks when they are not needed [PYF+00], a complete row of cells turns on just prior to their being read; all other cells in the SRAM or cache remain inactive. Furthermore, [ALR02] indicates that while the cell is inactive, it is able to retain its data with the '1' storage node remaining at V_{DD} and the '0' storage node raising to an intermediate value that remains below the threshold voltage of the NMOS transistor. When the gated transistor turns on, the '0' node returns to ground and discharges one of the bitlines. Note that the claim that the value stored in the SRAM cell can be retained when the gating transistor is off can also be applied to the Gated-V_{DD} SRAM cell in Fig. 2.7. To assure that the stored value is not lost however, the gating transistor must be sized appropriately. However, even with a resized gating transistor, many designs assume that the stored value is lost since the stability of the cells decreases quite rapidly when the cell is in its low-leakage state.

The DRG-Cache again exhibits the energy performance trade-off by reducing leakage by 46%, but increasing read times by 4.6%. Furthermore, as shown in [ALR02] the stability of the gated-V_{DD} cells reduces substantially with lower V_t transistors because

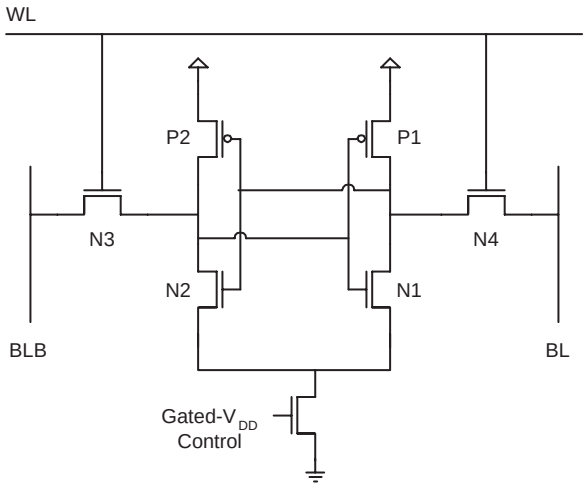


Figure 2.7: Gated V_{DD} SRAM cell

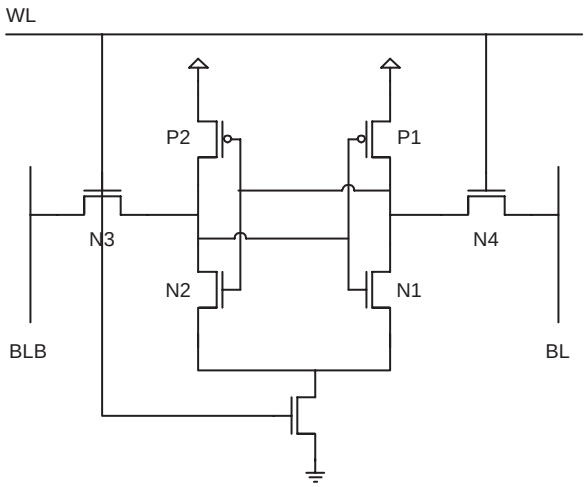


Figure 2.8: DRG SRAM cell

when the cell is inactive, the '0' storage node may rise over the threshold voltage of the low-V_t NMOS transistor, and the stored data may be deleted.

2.5 Architectural Level Techniques to Reduce Power in Caches

The circuit-level techniques that limit the power in caches have been reviewed. Now a review of architectural level techniques that reduce power consumption in caches will be presented.

2.5.1 Block Buffering

Due to spatial and temporal localities in programs, it is very likely that a requested word will be found in the block that was last accessed [KG97]. In this situation the tag and status arrays do not have to be read, which can save on precharge and readout energy.

To be able to save the power, an extra latch and comparator are needed that store and compare the set address of the last access with that of the current address. The comparison is done first, and only on a miss is the normal cache access started. This can add an additional delay to the cache access time, but there exists pipelined versions of this design which add no delay [KG97].

2.5.2 RAM subbanking

Another way of saving dynamic energy is to subbank the data arrays and add decoders that select each subbank, not allowing wordlines to go high for unselected subbanks [KG97]. The configuration is shown in Fig. 2.9. This method saves dynamic power consumption along the wordlines and bitlines by reducing the number of cells that are accessed on each read. This method, however, needs a hierarchical decoding scheme that may add to the access time.

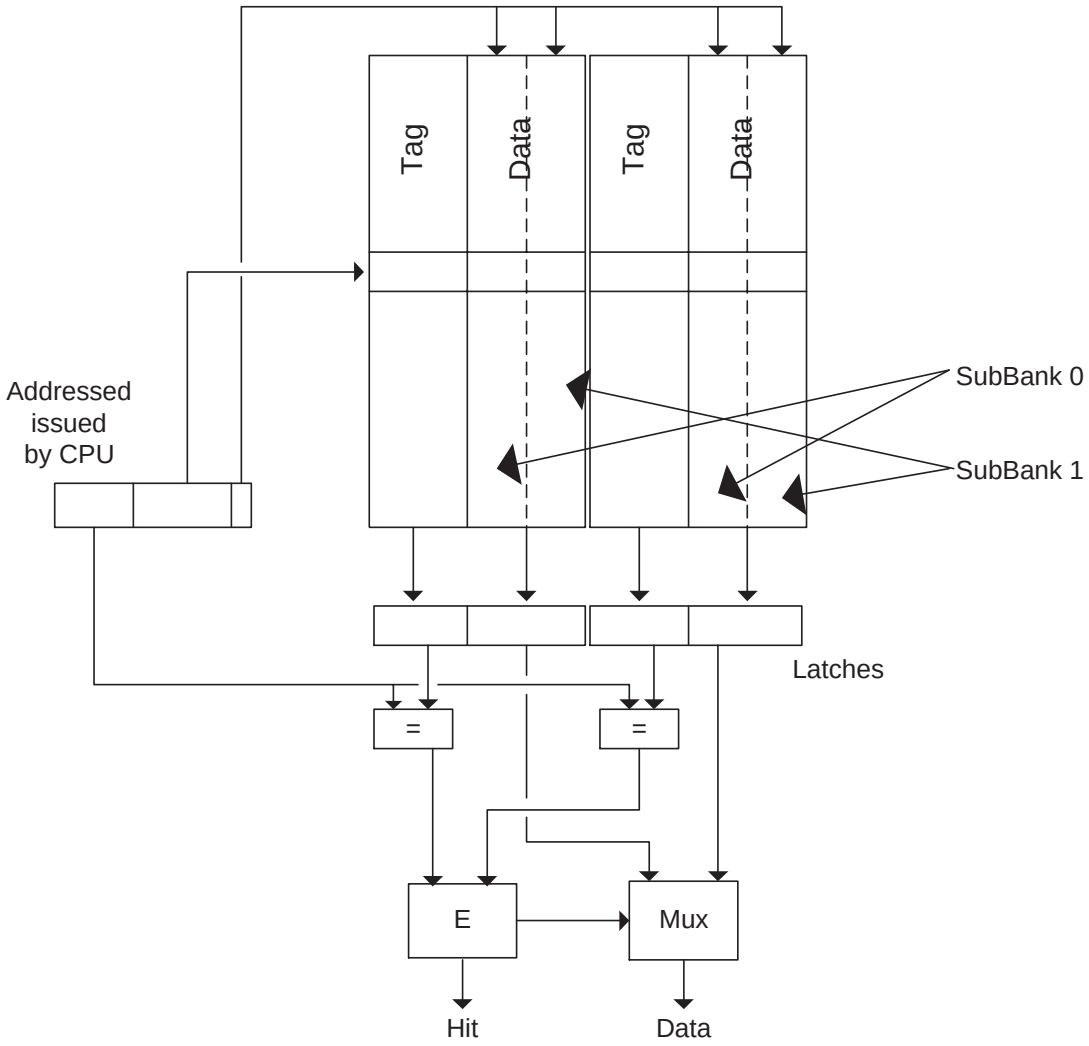


Figure 2.9: 2-way Set-Associate Cache with 2 subbanks per Data bank

2.5.3 Turning off Cache Lines

The circuit level technique presented in section 2.4.4 can only be useful if the cache lines can be turned off at appropriate times. Three different methods for deciding when cache lines should be turned off are presented below.

Cache Decay

In [KHM01], cache lines are turned off after a certain number of cycles have elapsed since its last access. This policy follows from program behavior, where cache lines usually see a flurry of activity when first brought in, and then a period of dead time between their last access and when they are replaced [KHM01]. Two different methods are described for deciding how long a cache line has to remain inactive before it is shut off. The first method uses a pre-set number of cycles to make the decision, and the second method uses an adaptive policy that tries to approach the best-case operating point in terms of leakage energy [KHM01].

The adaptive scheme tries to choose the smallest decay interval from a predefined set of intervals for each cache line [KHM01]. The scheme starts with the smallest interval, and every time it turns off a cache line before its last access, which causes a miss, the scheme selects a larger interval. [KHM01] notes that the static policy results in a leakage reduction of 4 times, and the adaptive policy results in a leakage reduction of 5 times, with negligible degradation in performance. The degradation in performance is due to the increased number of misses that are caused by turning off cache lines that still need to be accessed.

Dynamically ResIzable instruction-cache (DRI i-cache)

The DRI i-cache dynamically estimates and adapts to the required instruction-cache size and turns off a selected set of cache lines [YPF+01]. The total number of cache lines in sleep mode is controlled by periodically examining the miss rate, and comparing it to a pre-determined value [YPF+01]. If the observed miss rate is lower than the pre-determined value, another part of the instruction-cache is put to sleep [YPF+01]. However, if the miss-rate is larger than the acceptable value, more cache-lines are

activated.

[YPF+01] notes that DRI reduces leakage by 62% on average, but increases execution time by 4%. It is also noted that the same technique could be used for data caches, but they did not study data-cache implications due to complications involving dirty cache blocks (blocks that have to be written back to main memory before being replaced).

Adaptive Mode Control (AMC)

A limitation of both Cache Decay and DRI policies is that their control mechanisms depend on arbitrary parameters that must be tuned [ZTRC01]. Another option is to only deactivate the data portion of the cache, and allow the tag portion to remain active, thus allowing hardware to measure the hypothetical miss rate if the cache lines were still kept active [ZTRC01]. This way the hardware can self-adjust to variations among different applications and changes in program behavior as it executes.

More specifically AMC turns individual cache lines on and off, as in Cache Delay, but with the key difference that the turn-off period is dynamically adjusted to ensure performance closely tracks a cache without sleep mode [ZTRC01]. By using AMC, [ZTRC01] illustrates that on average 73% of instruction cache lines and 54% of data cache lines can be turned off with only a 1.7% impact on performance.

2.5.4 Zero Byte Compression

Zero byte compression takes advantage of the asymmetric distribution of '1' and '0's to reduce energy dissipation. Zero byte compression adds an extra bit per byte of memory which indicates that a byte composed of all '0's is stored. This additional bit can preemptively stop the raising of the wordline so that there is no bitline swing thus saving power [VZA00]. At the same time, the sense amplifiers are forced to read zeros so that the correct value is read out of the SRAM.

Zero byte compression also saves energy during a write since only one bit, instead of the whole bit field has to written to [VZA00]. [VZA00] states that total data and instruction cache energy is reduced by 26% and 10% respectively.

2.6 Conclusion

In many of the methods presented above there is a fundamental trade-off between performance and energy savings: a reduction in energy produces a negative impact on performance. The following chapters will introduce a new SRAM cell that dramatically reduces leakage power with no degradation in performance. This is achieved by using a circuit-level technique that takes advantage of program behavior to reduce leakage.

3 Cell Level Design

> Inasmuch as human society consists of two parts, the male and female, the happiness and stability of humanity cannot be assured unless both are perfected.
>
> Abdu'l Baha

3.1 Introduction

Evaluating the merit of different SRAM cells depends on factors, including performance, area, power, and stability. The intended use of the embedded memory further influences which factors are more critical to the design. Previously, performance and area were the two main factors involved in memory cell design, but recently power has become an increasingly more important design consideration.

Ideally, an SRAM cell is fast and dissipates low leakage power. This is increasingly at odds with a fundamental technology trade off between transistor speed and leakage: the lower the threshold voltage of a transistor, the faster it becomes and the more leakage power it dissipates. Conventional high-performance SRAM cells use a symmetric configuration of six transistors of identical threshold voltage to achieve fast access latency. As the supply voltage is scaled down, the transistor threshold voltage is also scaled to maintain performance. As a result of the low threshold voltage, leakage power increases rapidly due to the exponential relationship between leakage and V_t. Leakage can be reduced by using higher-V_t transistors, but by using an all-high-V_t

transistor cell performance degrades by an unacceptable margin. Asymmetrical SRAM cells, on the other hand, reduce leakage while maintaining performance comparable to traditional cells.

The asymmetrical cell family is built on the following premise: select a preferred stored value and weaken, by increasing the threshold voltage, only those transistors necessary to drastically reduce leakage when this value is stored. These cells exhibit asymmetric leakage and access behavior. Fortunately, their asymmetric access behavior can be exploited to maintain performance while reducing leakage.

In this chapter, the methodology for evaluating different asymmetric SRAM cells is presented, and is followed with the design and evaluation of various asymmetric dual-V_t SRAM cells. The most promising asymmetric cells are then further analyzed under process variations and different operating conditions.

3.2 Methodology

All results noted throughout this thesis are simulation results produced at 110°C using SPICE models of a commercial 0.13μm 1.2V CMOS technology. The choice of 110°C for all simulations was used since SRAM based structures frequently sit alongside active logic circuits such as microprocessors where the temperature of the die is high.

Furthermore, throughout this thesis, the following convention will be used. The basic 0.13μm 1.2V, transistor's (referred to herein as the Regular-V_t (RV) transistor) threshold voltage is artificially increased by 0.2V using the HSPICE in-line parameter DELVTO to obtain the High-V_t (HV) transistor. Although the specific choice of the 0.2V value may seem arbitrary, one can argue that the conclusions of this work, namely the feasibility and utility of using an asymmetric cell to reduce leakage, are valid irrespective of the specific value used for DELVTO. This specific value was selected because it leads to a difference of about 10X between the leakage currents of HV and RV transistors, which is typical of dual-V_t technology.

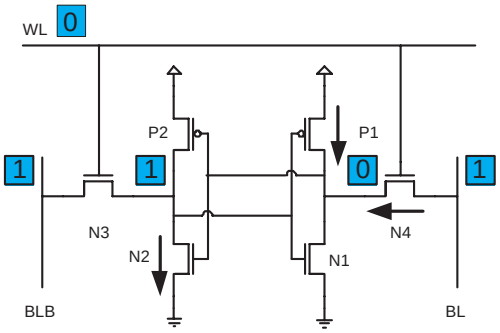


Figure 3.1: Transistors that dissipate leakage when an SRAM is holding a '0'

3.2.1 Leakage

An SRAM cell is comprised of two inverters (P2, N2) and (P1,N1) and two pass transistors N3 and N4. In the inactive state, the wordline (WL) is held low so that the two pass transistors are off, isolating the cell from bitline (BL) and bitline-bar (BLB). During this time the bitlines are also typically charged at V_{DD}. In this state, most of the leakage is dissipated by the transistors that are off while having a voltage differential across their drain and source. Furthermore, the transistors that dissipate leakage power depends on the value stored in the cell. For example when the cell is storing a '0', as in Fig. 3.1, the leaking transistors are P1, N4 and N2. If, however, the cell was storing a '1' then transistors P2, N1 and N3 would dissipate leakage power.

Since SRAM cells spend most of their time in their inactive state with their bitlines precharged high, the leakage of the cell was measured in this state. A complete column of cells (64 cells) was used to minimize precision errors when measuring leakage current, which can occur when high-V_t cells are used and very little leakage is produced. All the cells in the column were initialized to store a '0' and then a '1' and the stable current drawn from the power sources was measured as the leakage.

3.2.2 Performance

The performance of the cell is composed of two components:

1. How fast the cell can be read.

2. How fast the cell be written to.

To obtain a measure of the relative read speed of the different cells, the discharge time for each side of each cell was measured. Discharge time is defined as the time from when the wordline is raised to when one of the bitlines reduces to 90% of its precharge value. The value of 90% was chosen due to it being an appropriate differential signal to trigger the sense amplifiers.

For a measure of the write times, the cell was tested under to two states; one holding a '0' and another holding a '1'. Then the opposite value was placed on the bitlines, and the wordlines raised. The flip time was measured from the time the wordline was raised to when the latter of the two internal storage nodes reached 90% of its final value. The worst-case flip time was recorded as the write time for the cell.

3.2.3 Stability

The stability of all cells was measured through both the SNM and the $I_{\text{trip}}/I_{\text{read}}$ methods. For both stability tests, the stability was first measured under nominal conditions, assuming no process variations.

Then, to measure stability under process variations, two different sets of tests were performed. First, the SNM and $I_{\text{trip}}/I_{\text{read}}$ tests were performed on different combinations of different V_t and length variations for all six transistors in the cell. The combinations included modifying the NMOS transistors' V_t and length values and the PMOS transistors' V_t value by $\{-3\sigma, 0, +3\sigma\}$, thus giving $3^8 = 59,049$ combinations. The worst-case value for each cell was found, and compared to the worst-case value obtained for the RV cell.

Second, Monte-Carlo analysis was performed to obtain a distribution for the SNM and $I_{\text{trip}}/I_{\text{read}}$ values. For each cell, 500 scenarios for V_t and length were generated, which were consistent with their joint-distributions, and simulated. The mean of the distribution was calculated using the unbiased estimator in Equation 3.1 and the variance was calculated by using the unbiased estimator in Equation 3.2 [AN97]. Finally, the Normal Scores Method was used to graphically determine the distribution

type [AN97]. With the availability of distribution type, mean and variance the probability of failure for various cells was measured.

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{3.1}$$

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2 \tag{3.2}$$

3.3 Nine Asymmetrical Cells

As noted in section 3.2.1 transistors P1, N4 and N2 in Fig. 3.1 dissipate leakage power if the cell is storing a '0', and if the cell was storing a '1' then transistors P2, N1 and N3 would dissipate leakage power. A simple technique for reducing leakage power would be to replace all transistors with high-V_t ones, but this unacceptably degrades the bitlines discharge times by 62%. Since ordinary programs exhibit a strong bias in cache-resident bit values as will be shown in Chapter 5, another possibility to reduce leakage power, but at the same time keep read access times short, is to choose '0' as the preferred stored value and to only replace those transistors that contribute to the leakage power in that state with HV transistors, as seen in Fig. 3.2.

This Basic Asymmetric (BA) cell was simulated and it exhibited the same leakage

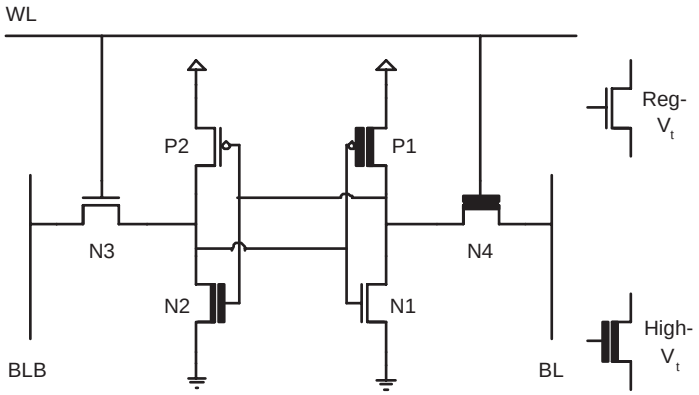


Figure 3.2: Basic Asymmetric SRAM cell

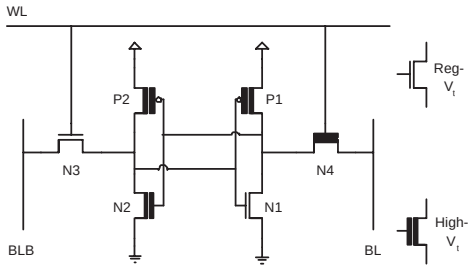


Figure 3.3: Leakage Improved 2 Cell

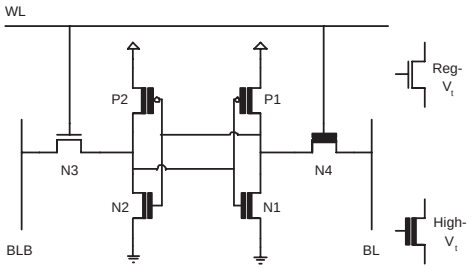


Figure 3.4: Leakage Improved 3 Cell

as the RV cell when holding a logical '1', but it decreased leakage by 70X when holding a logical '0.'

The read access time of the BA cell is, however, degraded. Due to N2's and N4's higher threshold voltage, the bitline discharge time takes longer. The discharge time for BLB and BL are 12% and 46% longer than the discharge time for the RV cell respectively. The increase in discharge time along BL is unimportant as read times can be made to match the faster read time by using a set of dummy bitlines and a unique sense amplifier, that will be discussed in Chapter 4.

Since PMOS transistors have very little effect on the cell's read access time[1], a better asymmetric cell consists of the BA cell with P2 also being high-V_t. This cell, shown in Fig. 3.3, referred to as the Leakage Improved (LI)2 cell, has the advantage of partially reducing leakage in the high leakage state. When the cell is holding a logical '1' the cell reduces leakage by 1.6X, and when it is holding a logical '0' it reduces leakage by 70X. The discharge times for BLB and BL are 12% and 46% longer than the discharge times for the RV cell respectively; the same as the BA cell's discharge times.

Another modification to BA is that N1 can be made high-V_t as the sense amplifier matches the read time on the slow side of the cell to the fast side. This leads to the cell in Fig. 3.4, which is reffered to as LI3. This cell further reduces leakage when the cell is holding a '1', with there being a leakage reduction of 7X in that state. The BL

[1] the role of pulling down the bitlines are played by the two NMOS transistors on the side of the cell storing the '0'

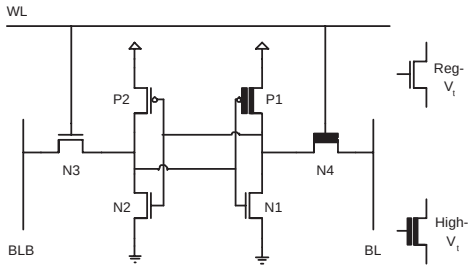


Figure 3.5: Speed Improved 1 Cell

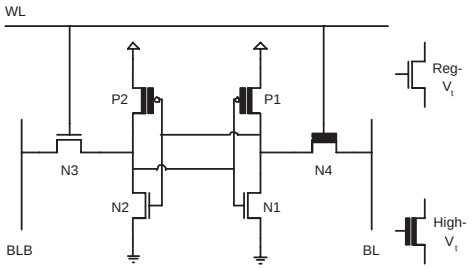


Figure 3.6: Speed Improved 2 Cell

discharge time is now 62% longer than the discharge time for the RV cell, but that is of minor importance due to the sense amplifier design, reviewed in Chapter 4.

The two asymmetric cells, LI2 and LI3, take the BA cell and improve its leakage performance while producing similar read access times. Another design front is to take the BA cell and try to improve its read access time, while keeping some of the leakage benefits found in the BA cell.

To eliminate the speed penalty incurred in the BA, cell due to both pull-down paths having one high-V_t transistor, both N2 and N3 are made low-V_t as shown in Fig. 3.5. This cell, Speed Improved (SI)1, now has discharge times for BLB and BL which are 0% and 47% respectively longer than the RV cell. Thus one side of the cell is just as fast as the RV cell. However, this cell suffers from higher leakage than the BA cell, with a reduction of 2X when holding a '0', and the same leakage power when holding a '1'.

The same transformations performed on BA to improve its leakage performance can also be performed on the SI1 cell. First P2, is made high-V_t, and then N1 is also made high-V_t. These two new cells are named SI2 and SI3 respectively. SI2 is shown in Fig. 3.6 and SI3 is shown in Fig. 3.7. SI2 has leakage reductions of 2X and 1.6X when storing a '0' and '1' respectively, while SI3 has leakage reductions of 2X and 7X[2]. These two cells have no read access time degradation compared to the RV cell along

[2]Note that SI3 reverses the preferred leakage state to the state when the cell is holding a '1'. All further references to this cell will have the '1' state as the preferred state so that the cell language remains in conformity with all other cells, but it should be noted that in practice the cell bitlines can be flipped to allow for '0' to be the preferred state without affecting any of the performance or stability results shown hereafter.

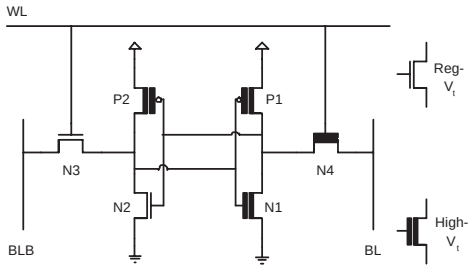


Figure 3.7: Speed Improved 3 Cell

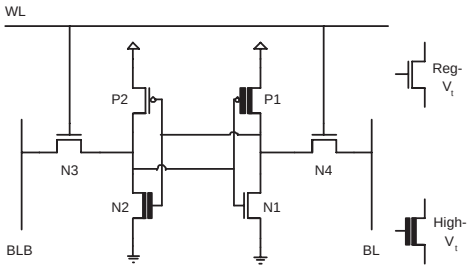


Figure 3.8: Special Precharge Cell

BLB, but have a 47% and 62% degradation along BL respectively. Once again, due to the use of the new sense amplifier, the degradation along BL is of minor importance.

One would like to get the very low leakage of the LI2 and LI3 cells with a very small read access delay. A final asymmetrical cell can meet these objectives, but it requires a different read operation. In the steady state, instead of keeping BL precharged to V_{DD}, it must be kept at ground. Now, N4 can be kept low-V_t for the preferred '0' state. This Special Precharge (SP) cell is shown in Fig. 3.8. Before a read, BL has to be raised to '1', or a new sensing scheme may have to be developed, which may be power hungry. This cell requires changes to the peripheral circuits of the SRAM array and further work is required to develop this concept. Nevertheless, the results for this cell are presented for completeness: leakage is reduced by 83X in the '0' state, while the '1' state shows no leakage reduction. Bitline discharge times are degraded by 12% and 0% on BLB and BL respectively.

A summary of the leakage reduction while holding a '0' and '1' can be seen in Table 3.1. A summary of the the bitline discharge times are in Table 3.2. This table shows the distinction between the LI and SI cells since all LI cells show a 12% increase in bitline discharge times, while the SI cells show no increase in bitline discharge times. Additionally Fig. 3.9, which shows the asymmetry in the leakage current by shading the region between the '0' and '1' leakage currents for the cells, shows that the BA and LI2 cell show no advantage to the LI3, since the LI3 cell has the same speed performance as the BA and LI2 cells but has better leakage performance. Also, SI3 is clearly the best design from within the SI cells. The LI3 cell will be referred to henceforth as the Leakage Enhanced (LE) cell and the SI3 cell will be referred to henceforth as the Speed

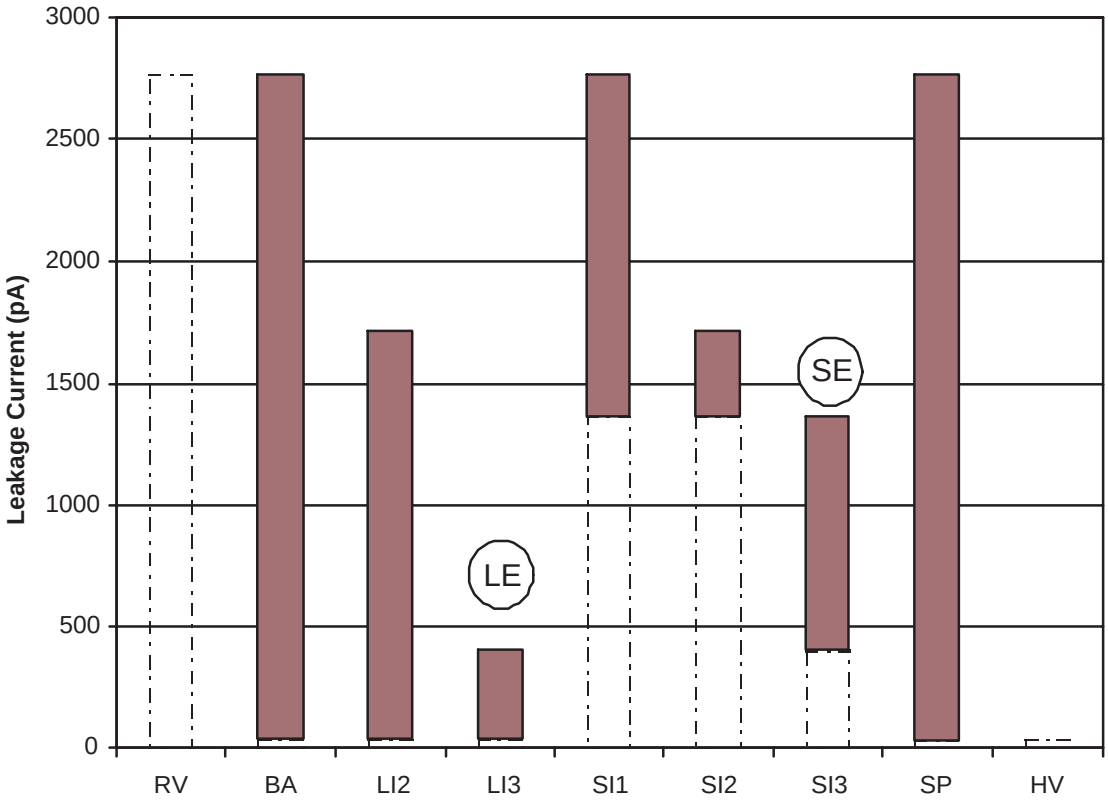


Figure 3.9: Graphical Representation of Asymmetrical Leakage Characteristics of all cells

Enhanced (SE) cell.

Until now, only the bitline discharge times of the different cells have been compared, and write times have been ignored. The write times of the cells are less important because stronger write drivers can be designed to drive the bitlines, and write drivers are a small portion of the total SRAM. The write times of the asymmetric cells all lie within the write times of the RV cell and the HV cell. The precise numbers can be seen in Table 3.3, where the LE and SE cells show the smallest increase in their respective groupings.

Since the LE (Fig. 3.4) and the SE (Fig. 3.7) are the two best designs from two

Table 3.1: Summary of Leakage Reduction for all asymmetric cells

Asymmetric Cell	Leakage Reduction when holding a '0' (times)	Leakage Reduction when holding a '1' (times)
BA	69.5X	1.0X
LI2	69.5X	1.6X
LI3	69.5X	7.0X
SI1	2.0X	1.0X
SI2	2.0X	1.6X
SI3	2.0X	7.0X
SP	83.3X	1.0X
HV	69.5X	69.5X

Table 3.2: Summary of bitline discharge times for all asymmetric cells

Asymmetric Cell	Percent Increase of BLB discharge time over RV cell	Percent Increase of BL discharge time over RV cell
BA	12.25%	46.73%
LI2	12.25%	46.50%
LI3	12.09%	61.64%
SI1	0.00%	46.73%
SI2	0.00%	46.51%
SI3	0.00%	61.69%
SP	12.26%	0.00%
HV	61.64%	61.64%

Table 3.3: Summary of write times for all asymmetric cells

Asymmetric Cell	Percent Increase in Write Times in reference to the RV cell
RV	
BA	51.9%
LI2	41.2%
LI3	35.9%
SI1	56.1%
SI2	45.5%
SI3	40.2%
SP	11.5%
HV	69.4%

sets of asymmetric cells, only these two cells will be further discussed in the following sections.

3.4 Stability

The SE and LE cells were shown in Section 3.3 to have the most leakage savings for their associated read access performance. A further check on the merit of their design is their stability, under both the SNM and $I_{\text{trip}}/I_{\text{read}}$ tests.

3.4.1 Static Noise Margin

The SNM of the LE and SE cells were computed through simulation along with the SNM of the RV cell, which was used as a reference. Under nominal conditions, the SNM of the LE and SE cells were $0.246V$ and $0.221V$ respectively, while the SNM of the RV cell was $0.250V$. Thus, the LE and SE show a decrease in SNM of 1.6% and 11.7% respectively. Although an increase in the SNM for designs using HV transistors might be expected, the asymmetry of the cells skews the lobes of the butterfly curve and decreases the SNM, as is explained in further detail below.

First, examine the SNM of the cells when the wordline is not active. During this

state, the SRAM cell is not as vulnerable as when it is being read, but a study of this case helps to understand the decrease in the SNM when the cell is being read. When the wordline is off, the only transistors that affect the SNM are the four transistors comprising the back-to-back inverters.

Since the four internal transistors of the LE cell are all high-V_t, the cell has equal low and high noise margins of $0.685V$, a 22.6% increase over the standby SNM of the RV cell. However, when the SNM of the cell is being measured during a read, as in Fig. 3.10, the cell has high stability in one state, $0.363V$, and low stability in the other, $0.246V$. The asymmetry in the LE butterfly curve is the result of the mismatch between the strength of the pass-gate (N3) and pull-down (N2) transistors. During a read, the N3 pass transistor, with its low threshold voltage, has a higher conductivity than N2 and raises the voltage at the storage node to a higher voltage than if the two NMOS were of equal strength. This increased voltage at the storage nodes makes the SRAM cell more susceptible to noise.

The SE cells internal four transistors are not symmetric. Thus the standby SNM

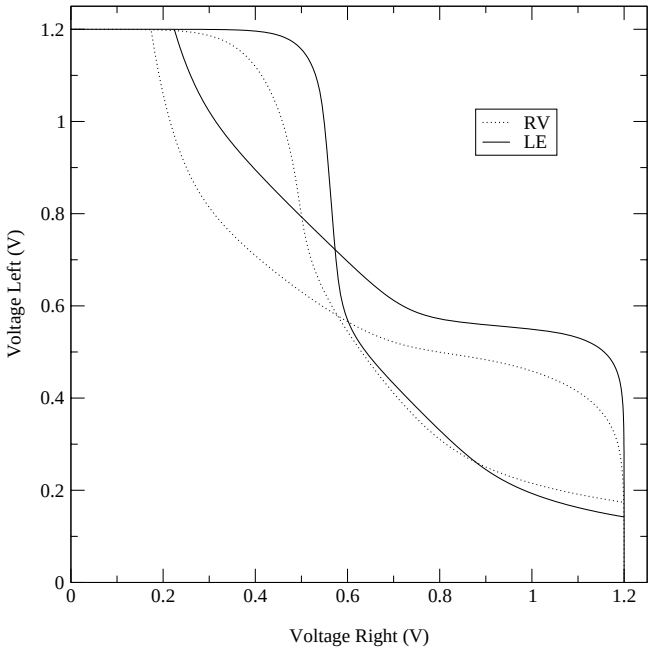


Figure 3.10: SNM of the LE cell

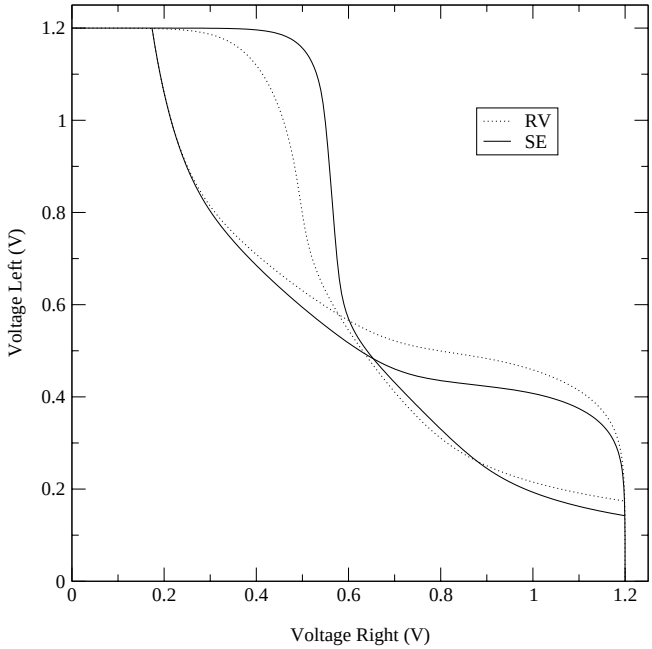


Figure 3.11: SNM of the SE cell

of the cell has asymmetric lobes with noise margins of $0.535V$ and $0.727V$, which in the worst-case is 4.2% lower than the SNM of the RV cell. The source of this mismatch is the V_t difference between N1 and N2, which causes one of the voltage transfer characteristics to commence its transition in the SNM plot from '0' to '1' later than the in RV cell. During a read, the mismatch between the size of the lobes becomes exaggerated because it is as if a constant is subtracted from the noise margin on each side of the cell since the cell has equal strength pass-gate and pull-down transistors on each side of the cell. While being read, the SE cell has low and high noise margins of $0.222V$ and $0.366V$ respectively. The SNM plot of the SE cell is shown in Fig. 3.11. The SNM of all cells is summarized in Table 3.4.

As explained in Section 3.2.3, the stability of the cells under process variations were analyzed by two methods. First, by exhaustingly sweeping over different process variations, the worst-case SNM was found for each cell. Then Monte-Carlo analysis was also performed on the cells.

Table 3.4: Nominal SNM

Cell	SNM under standby(V)	% Increase	SNM during read(V)	% Increase
RV	0.559		0.250	
LE	0.685	22.64%	0.246	-1.61%
SE	0.535	-4.18%	0.222	-11.33%

Worst-Case Process Variations

The asymmetrical cells' stability performance degrades compared to that of the RV cell under worst-case conditions. Since process variations induce an asymmetry in the butterfly curve, the original asymmetry inherit in the butterfly curves for the LE and SE allows one lobe of the butterfly curve to become pinched off even further and, therefore, lose stability. Fig. 3.12 shows the process corners where the LE cell has its worst stability; the butterfly curve becomes pinched off when N3 becomes stronger than N2 and P1 increases in strength, while N1 does not. The worst-case for the SE cell occurs at a different process corner. The butterfly curve becomes pinched off when P2 decreases in strength and N2 increases in strength, and N4 gets stronger than N1, as shown in Fig. 3.13. The worst-case SNM values are summarized in Table 3.5.

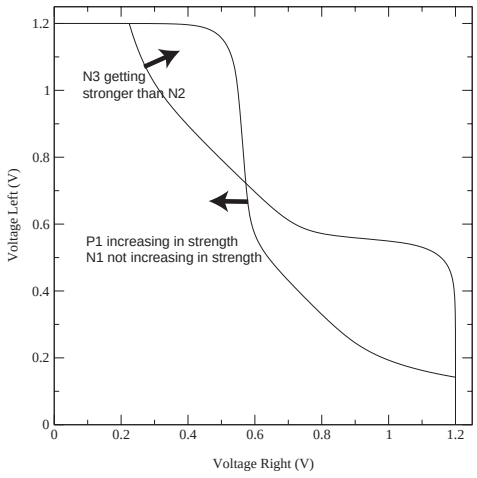


Figure 3.12: SNM of LE cell pinching off

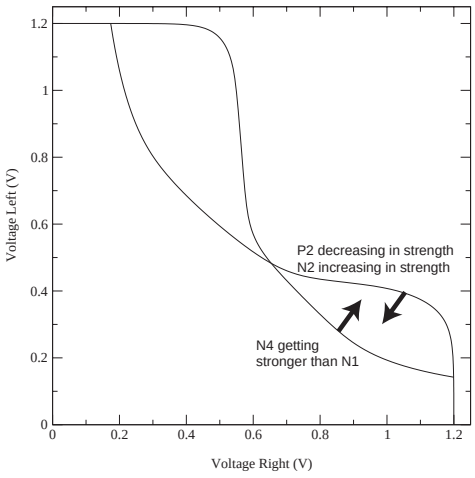


Figure 3.13: SNM of SE cell pinching off

Table 3.5: Worst-Case SNM

Cell	Worst-Case SNM(V)	% Increase over RV cell
RV	0.091	
LE	0.088	-3.73%
SE	0.065	-28.79%

Table 3.6: Mean and Standard Deviation during Monte-Carlo Analysis for SNM

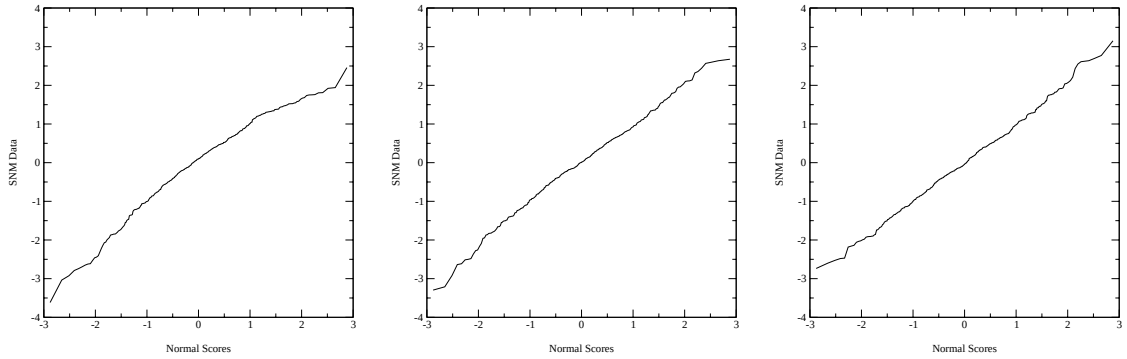
Cell	Mean(V)	Standard Deviation(V)
RV	0.231	0.0182
LE	0.247	0.0244
SE	0.218	0.0259

Monte-Carlo Analysis

The mean and standard deviation, calculated from Monte-Carlo analysis, are displayed in Table 3.6. The Normal Scores method also showed that the distributions for all cells were Gaussian because of the linear nature of the plots seen in Fig. 3.14. Due to their very small standard deviation of the SNM of all cells remains very close to their respective mean. Thus the mean of the SNM becomes a very important measure, and is a better reflection of the stability than the nominal or worst-case SNM. Using the mean as a measure of stability, the LE has a 7% increase in SNM and the SE has a 5.8% decrease. The mean SNM of the RV cell has decreased in comparison to its nominal value; this is because most process variations will cause one of the lobes of the SNM plot to become smaller, and thus become the limiting SNM value. However with the asymmetric cells, since one lobe of the SNM plot is already quite large, process variations can cause the SNM value of the limiting lobe to decrease or increase.

3.4.2 I_{trip}/I_{read}

When using the SNM as a measure of stability, the LE cell was comparable to the RV cell in stability while the SE cell showed a marginal decrease in stability. When



(a) RV cell (b) LE cell (c) SE cell

Figure 3.14: Normal Scores for SNM

$I_{\text{trip}}/I_{\text{read}}$ is computed by simulation, it will be seen that the SE outperforms the RV cell and the LE cell has lower stability. Table 3.7 summarizes the results that will be explained below.

The LE cell has a lower $I_{\text{trip}}/I_{\text{read}}$ value due to the V_t mismatch between the pass transistor and pull-down transistor on one side of the cell. The I_{trip} values from both sides of the cell show a drop compared to the I_{trip} value from the RV cell due to both pull-down transistors becoming high-V_t. However, with N3 remaining low-V_t, I_{read} on the fast side of the cell does not suffer the same drop, and $I_{\text{trip}}/I_{\text{read}}$ decreases compared to that of the RV cell.

The SE, which has the same strength pull-down and pass transistors on each side of the cell, does not experience the same problem as the LE cell. On the slow side of the cell, both I_{trip} and I_{read} decrease compared to the RV cell, but I_{read} decreases by a

Table 3.7: Nominal $I_{\text{trip}}/I_{\text{read}}$

Cell	Nominal(V)	% Increase
RV	2.26	
LE	2.10	-7.3%
SE	2.60	14.9%

larger amount, increasing the $I_{\text{trip}}/I_{\text{read}}$ value. On the fast side of the cell, I_{read} does not change compared to the RV cell, but I_{trip} increases slightly. I_{trip} is larger in the SE cell because the presence of high-V_t transistors on the '1' side of the cell does not allow for the reduction in voltage (due to leakage) at the stored '1' node which degrades the current sinking capacity of the pull-down NMOS in the RV cell.

Worst-Case Process Variations

After an exhaustive search of different corner cases of process variations, the worst-case $I_{\text{trip}}/I_{\text{read}}$ value was found for each cell, and is summarized in Table 3.8. The SE cell still performs better than the RV cell, while the LE cell has a lower $I_{\text{trip}}/I_{\text{read}}$ value compared to the RV cell.

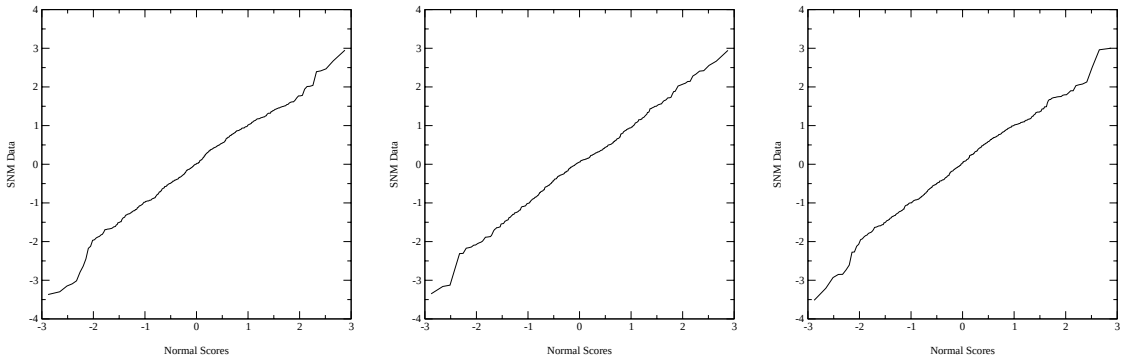
The LE cell and the RV cell record their worst-case $I_{\text{trip}}/I_{\text{read}}$ for the same process corner: when the difference in strength between N2 and N3 is amplified with N2 becoming weaker, and N3 becoming stronger. The SE cell however suffers its worst-case $I_{\text{trip}}/I_{\text{read}}$ when N4 becomes stronger than N1.

Monte-Carlo Analysis

The Monte-Carlo analysis illustrates that $I_{\text{trip}}/I_{\text{read}}$, just like the SNM stability measure, is Gaussian from the linear plots obtained from the Normal Scores Method shown in Fig. 3.15. Table 3.9 shows the mean and standard deviation of the three cells. Notice, once again the standard deviation is very small, and thus most cells will be very near the mean where the LE has a 4.35% decrease and the SE cell has a 14.84% increase in $I_{\text{trip}}/I_{\text{read}}$ compared to the RV cell.

Table 3.8: Worst-Case $I_{\text{trip}}/I_{\text{read}}$

Cell	Worst-Case(V)	% Increase
RV	1.71	
LE	1.58	-7.8%
SE	1.82	6.1%



(a) RV cell (b) LE cell (c) SE cell

Figure 3.15: Normal Scores for $I_{\text{trip}}/I_{\text{read}}$

Table 3.9: Mean and Standard Deviation during Monte-Carlo Analysis for $I_{\text{trip}}/I_{\text{read}}$

Cell	Mean (V)	Standard Deviation (V)
RV	2.20	0.077
LE	2.10	0.090
SE	2.52	0.110

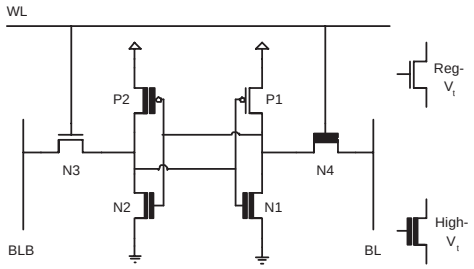


Figure 3.16: SLE cell

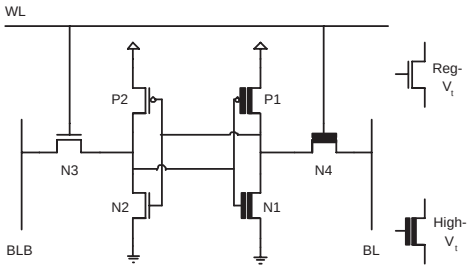


Figure 3.17: SSE cell

3.5 Stability Improved Cells through Threshold Voltage

The SE and LE cells have either a lower stability in the SNM test or the $I_{\text{trip}}/I_{\text{read}}$ test. In many cases, the stability of the cell is a critical factor to obtain a desired yield and to lower the cost of the chip. In that regard, two derivative cells, one from the LE cell and one from the SE, have been developed that improve upon their SNM, but do not decrease the leakage as much as the SE and LE cells[3]. The two new cells are named Stability-Leakage Enhanced (SLE) and Stability-Speed Enhanced (SSE).

One way to improve the SNM of the cells under process variations is to try to make the size of the lobes of the butterfly curve symmetric. For the LE cell the lobes can be made more symmetric by making N2 low-V_t, but this new cell would just be the SE cell. Another option is to make P1 low-V_t. This change, seen in Fig. 3.16, has the opposite effect of the lower arrow in Fig. 3.12, and makes the lobes of the butterfly curve more symmetric. The lobes are now $0.360V$ and $0.283V$ instead of $0.363V$ and $0.246V$; a 13% increase over the SNM of the RV cell. The change in SNM plot can be seen in Fig. 3.18.

To make the SE cell's SNM plot more symmetric, P2 can be made low-V_t to have the opposite effect as the top arrow in Fig. 3.13. By doing this, the SNM plot, shown in Fig. 3.19, has lobes of $0.256V$ and $0.362V$ instead of $0.222V$ and $0.366V$; a 2.4%

[3]The $I_{\text{trip}}/I_{\text{read}}$ of the cells is not improved since the only method of improving the LE cell's value considerably is to make the pull-down NMOS on the fast side of the cell low-V_t which would make it the same as the SE cell.

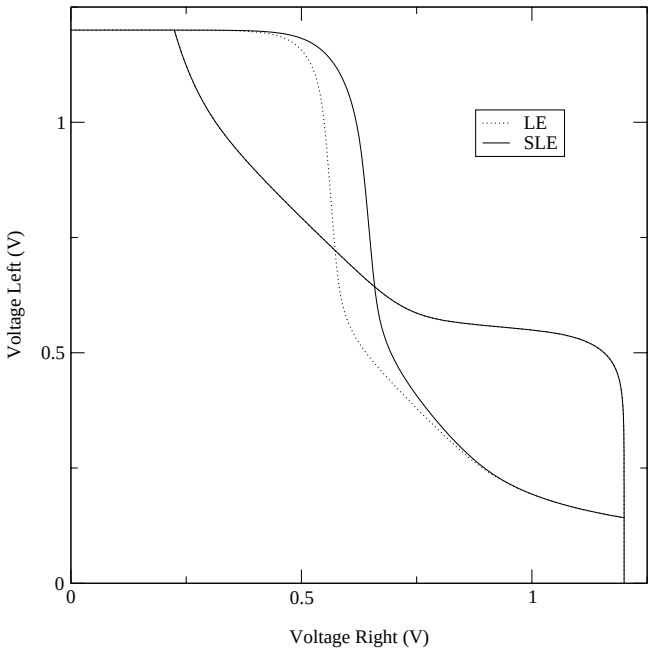


Figure 3.18: SNM for SLE cell

increase over the RV cell's SNM. The SSE cell is shown in Fig. 3.17. For both stability-improved cells, all the previous tests for leakage, performance, and stability can be performed to compare them to the cells they derived from as well from the RV cell.

3.5.1 Leakage

The leakage performance of the stability-improved cells falls off, because one transistor in the LE and SE has reverted back to a low-V_t transistor. For the SLE cell, the leakage reduction when holding a '1' remains unchanged at a 7X reduction, but the leakage reduction when holding a '0' changes from 70X to 2.5X. For the SSE cell, the leakage reduction stays at 2X when storing a '0', but when it is storing a '1' the leakage reduction changes from 7X to 1.9X

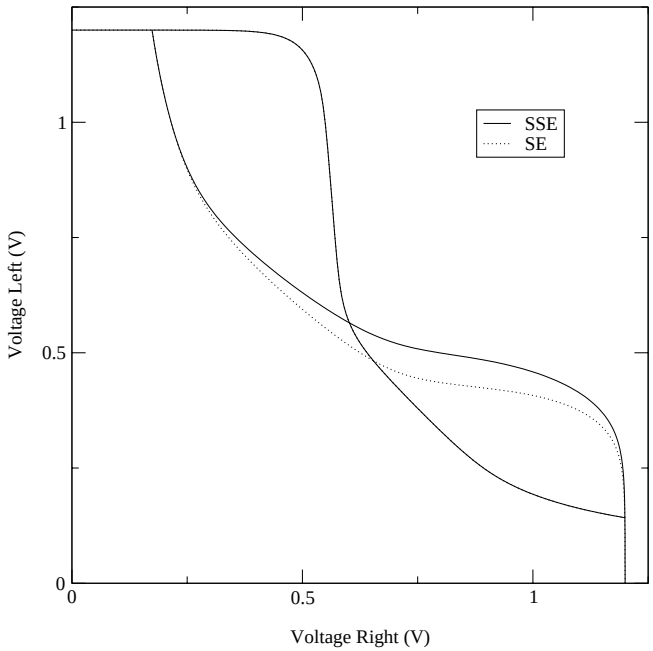


Figure 3.19: SNM for SSE cell

3.5.2 Performance

Since the PMOS transistors are not significant for discharging the bitlines, it would be expected that the discharge times for the stability-improved cells be very close to the cells they are derived from. Indeed, through simulation, it is seen that the discharge times along BL and BLB remain almost constant. As for the write times, the SLE's write time decreases to a 33.2% increase over the RV's write time from the LE's 36% increase. The SSE write time shows a 49.2% increase over the RV's write times.

3.5.3 Stability

The stability analysis was again performed on the derivative cells for both the SNM and $I_{\text{trip}}/I_{\text{read}}$. Both derivative cells perform better than the RV cell in both the worst case, and under Monte-Carlo analysis for the SNM. The results can be seen in Table 3.10.

Using the $I_{\text{trip}}/I_{\text{read}}$ method, there is very little change in the stability of the SLE and SSE cells compared to the LE and SE cells respectively. The lack of change is

Table 3.10: Percent Improvement over RV cell for stability-improved cells under SNM test

Cell	Worst-Case SNM % improvement	Mean SNM during Monte-Carlo analysis % improvement
SLE	35.4%	22.6%
SSE	7.8%	9.3%

Table 3.11: Percent Improvement over RV cell for stability-improved cells under $I_{\text{trip}}/I_{\text{read}}$ test

Cell	Worst-Case $I_{\text{trip}}/I_{\text{read}}$ % improvement	Mean $I_{\text{trip}}/I_{\text{read}}$ during Monte-Carlo analysis % improvement
SLE	-10.50%	-6.78%
SSE	4.06%	13.43%

due to the $I_{\text{trip}}/I_{\text{read}}$ method being strongly dependent on the NMOS transistors which have not been changed. The stability-improved cells, however, perform slightly worse than the cells from which they derived from. The results are summarized in Table 3.11.

3.6 Stability Improved Cells through Transistor Sizing

From the previous section, it can be seen that when stability is recovered through a change in threshold voltage of the PMOS transistors, a large portion of the leakage benefits of the asymmetric cells are lost. Furthermore, the low $I_{\text{trip}}/I_{\text{read}}$ of the LE cell could not be improved by threshold voltage assignment.

Another way of improving stability is to resize some of the transistors to reclaim the conductance lost due to the high-V_t assignment. This change does not have a large effect on the leakage characteristics because, as seen in Chapter 2, leakage increases exponentially with reduced threshold voltages, but increases only linearly with transistor

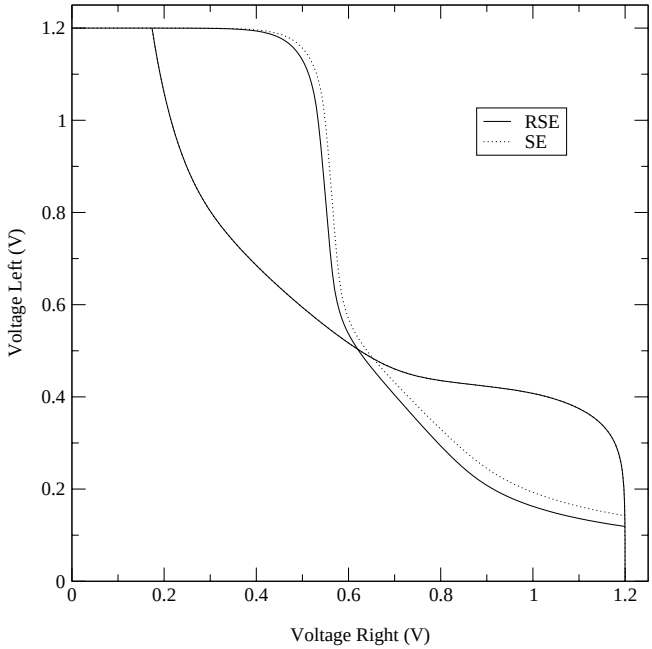


Figure 3.20: SNM for RSE cell

size.

The lobes of the SNM plot for the SE cell can be made more symmetric by making N1 wider. In this case, N1's width was increased by 26%, leading to a new cell which is referred to as Resized-Speed Enhanced (RSE) cell. The SNM of the RSE cell is comparable to that of the RV cell and the change in N1's size leads to an increase of only 2.9% in cell area. The change in the SNM plot can be seen in Fig. 3.20, where the SNM values are now $0.253V$ and $0.347V$ instead of $0.222V$ and $0.366V$. The SNM value is now within 1% of the RV's SNM value. The RSE cell's nominal value for $I_{\text{trip}}/I_{\text{read}}$ does not change much compared to the nominal value for the SE cell. On the slow side of the cell, which had the higher $I_{\text{trip}}/I_{\text{read}}$ value for the SE cell, the increase in N1's size allows I_{trip} to become larger and increases the $I_{\text{trip}}/I_{\text{read}}$ value. The fast side of the cell however, which has the limiting $I_{\text{trip}}/I_{\text{read}}$ value, has a reduced I_{trip} that reduces the final value of $I_{\text{trip}}/I_{\text{read}}$ to 2.53. The reduction in I_{trip} is due to the '1' storage node having a slightly lower voltage due to the increased leakage through N1. Nevertheless, the RSE cell's $I_{\text{trip}}/I_{\text{read}}$ value is still 11.8% better than that of the RV cell.

For the LE cell, increasing the width of N2 allows the conductance of N2 to approach that of N3, which leads to an increase in I_{trip}, thus increasing $I_{\text{trip}}/I_{\text{read}}$. By increasing N2's width by 22%,[4] the $I_{\text{trip}}/I_{\text{read}}$ value of the the new Resized-Leakage Enhanced (RLE) cell was made to be 2.28, which is comparable to the $I_{\text{trip}}/I_{\text{read}}$ value of 2.26 for the RV cell. The increase in N2's width also increases the SNM of the RLE cell where the margins are now $0.349V$ and $0.280V$ instead of $0.363V$ and $0.246V$; a 12% increase over the RV cell's SNM value. The SNM plot for the RLE cell can be seen in Fig. 3.21.

3.6.1 Leakage

As expected, the leakage performance of the resized cells is better than that of the SLE and SSE cells since transistor sizing instead of threshold voltage assignment was used to regain stability. For the RLE, cell the leakage reduction when holding a '1' remains unchanged at a 7X reduction, but the leakage reduction when holding a '0' is

[4]This change in N2's width only increases the cell area by 2.4%

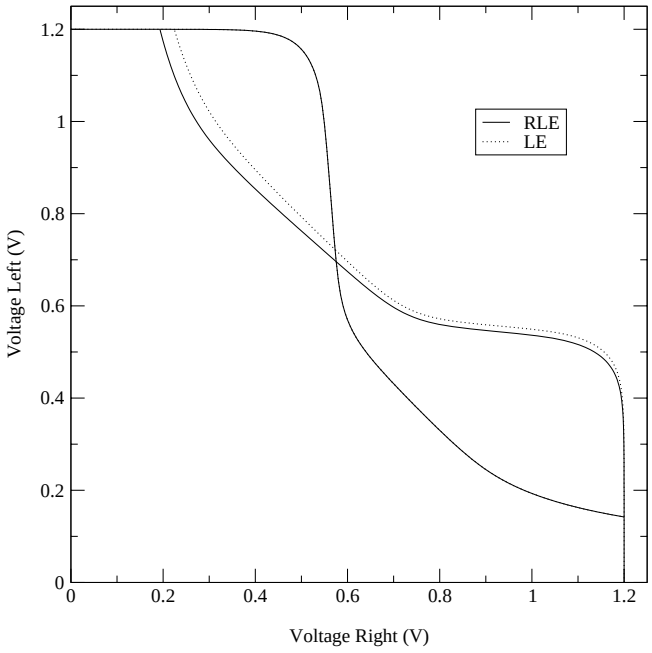


Figure 3.21: SNM for RLE cell

minimally reduced from 70X to 58X. (The SLE cell's leakage reduction when holding a '0' was only 2.5X). When the RSE cell is holding a '0,' the leakage reduction stayed at 2X relative to RV, and when it is holding a '1' the leakage reduction only changed from 7X to 6.8X. This difference is also minimal when compared to the SSE's leakage reduction of 1.9X when holding a '1.'

3.6.2 Performance

Due to the increased size of the pull-down NMOS transistors, the resized cells have the potential of improving the read-access time of the RV cell. For the RLE cell, the discharge time along BL remains at a 61.1% increase over the RV cell's BL discharge time, but the BLB discharge time is now only 3.7% longer than the RV cell's discharge time. As noted previously, only the BLB discharge time is important due to the timed read with the new sense amplifier. For the RSE cell, the discharge time along the fast side of the cell, BLB, does not change. The discharge time along BL, however, is reduced from SE's 61.7% increase to a 49.2% increase over the RV cell's discharge time. This increase in performance does not affect the cell's performance since it is along the slow side of the cell. As for the write times, the RLE's write time increases to a 39% increase over the RV's write time from the LE's 35.95% increase. The RSE write time increases to a 45% increase over the RV's write times.

3.6.3 Stability

The stability analysis has also been performed on the resized cells for both the SNM test and $I_{\text{trip}}/I_{\text{read}}$ test. Both resized cells perform better than the RV cell in the worst case, and under Monte-Carlo analysis for the SNM. The results can be seen in Table 3.12. Under the $I_{\text{trip}}/I_{\text{read}}$ method the RLE now outperforms the RV cell both in the worst-case and on average; the increase in N1's size accomplishes the higher $I_{\text{trip}}/I_{\text{read}}$. The RSE's $I_{\text{trip}}/I_{\text{read}}$ value also increases slightly under all tests, even surpassing the SE cell's $I_{\text{trip}}/I_{\text{read}}$ value in the worst-case. The increase in stability is due to the larger pull-down transistors, which are not affected as much by process variations. The results are summarized in Table 3.13.

Table 3.12: Percent Improvement over RV cell for resized cells under SNM test

Cell	Worst-Case SNM % improvement	Mean SNM during Monte-Carlo analysis % improvement
RLE	36.65%	21.4%
RSE	9.89%	7.9%

Table 3.13: Percent Improvement over RV cell for resized cells under $I_{\text{trip}}/I_{\text{read}}$ test

Cell	Worst-Case $I_{\text{trip}}/I_{\text{read}}$ % improvement	Mean $I_{\text{trip}}/I_{\text{read}}$ during Monte-Carlo analysis % improvement
RLE	2.51%	5.04%
RSE	11.6%	15.37%

3.7 Stability at Different Supply Voltages

Another figure of merit for the different cells is their stability at different supply voltages. For the technology being used, the nominal supply voltage is $1.2V$. Monte-Carlo analysis has been performed for the RV, LE, SLE, RLE, SE, SSE and RSE cells for supply voltages ranging from 0.75V to 1.6V, with the mean being plotted in Figures 3.22 and 3.23.

From the plot it can be seen that for voltages above 1.2V, the LE, SLE and RLE increase their advantage in stability over the RV cell. This is because at a higher V_{GS}, the difference in conductance between the pass-gate (N3) and pull-down (N2) transistors, which was the root cause of the unexpected low stability at 1.2V, diminishes. At higher voltages, the SNM of the SE and SSE cells starts to diminish similar to the SNM of the RV, albeit at a slower rate. The SNM of the RSE cell levels off at higher voltages.

With lower supply voltages, the SNM of the asymmetric cells starts to deteriorate. The SNM of the LE, SLE and RLE cells start decreasing rapidly, but the SLE's SNM

remains comparable to the RV's, while the RLE's SNM becomes comparable to that of the LE's. This decrease in stability is caused by the difference in conductance between RV and HV transistors at low values of V_{GS}, where the HV transistors are effectively off. Furthermore, at a low V_{GS} the extra conductance of the larger transistor in the RLE cell does not have a large effect since the transistor is not fully on. The SNM of SSE and RSE also decreases, but not as quickly as that of the LE. Again, this decrease in SNM is attributed to the difference in conductance of RV and HV transistors at low V_{GS}'s.

Based on [GAT02], the voltage regulator and power distribution network in microprocessors must maintain the supply voltage to within ±5% of nominal. Therefore, the reduced stability at low voltages for the asymmetrical cells may not be a big concern except perhaps during any chip testing that may need to be performed at low voltage.

The same tests were performed on the $I_{\text{trip}}/I_{\text{read}}$ method, with the curves for all cells being much more well-behaved. The $I_{\text{trip}}/I_{\text{read}}$ value for the SE and SSE cells are 24% larger than that of the RV cell at 0.75V and are 8% larger at 1.65V. The $I_{\text{trip}}/I_{\text{read}}$ value for the LE and SLE cells show a 16% decrease compared to the RV cell at 0.75V and are comparable at 1.65V. The resized cells behave slightly differently with the RSE cell having an 11.7% improvement at 1.65V and a 32.2% improvement at 0.75V. The RLE cells has a 9.6% improvement and 1.65V and a 4% decrease at 0.75V compared to the RV cell.

3.8 Supply Voltage Scaling

Since leakage power will become increasingly important in successive technologies, the different asymmetric cells have been tested with different supply voltages, and appropriately scaled threshold voltages, as in equations 3.3 and 3.4. Fig. 3.24 shows the leakage while holding a '0' and '1' for all cells under different supply voltages. The figure shows that the leakage savings incurred by using the asymmetrical cells continues for lower supply voltages, where it is more important as the leakage current rises exponentially with smaller supply voltages.

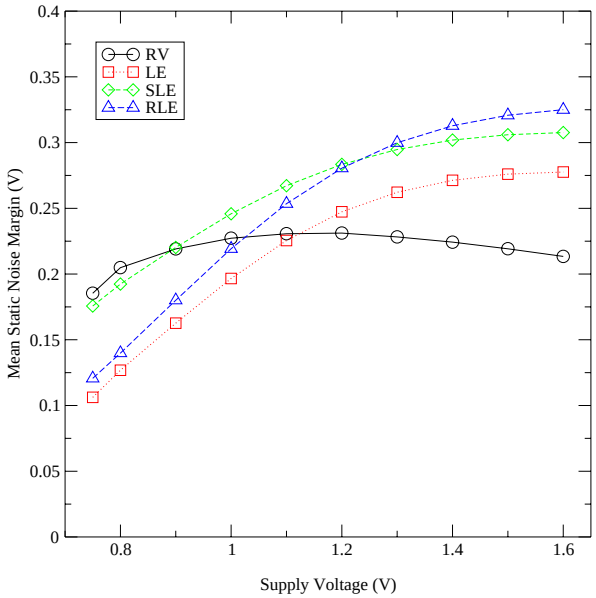


Figure 3.22: Mean SNM for different supply voltages for LE derived cells

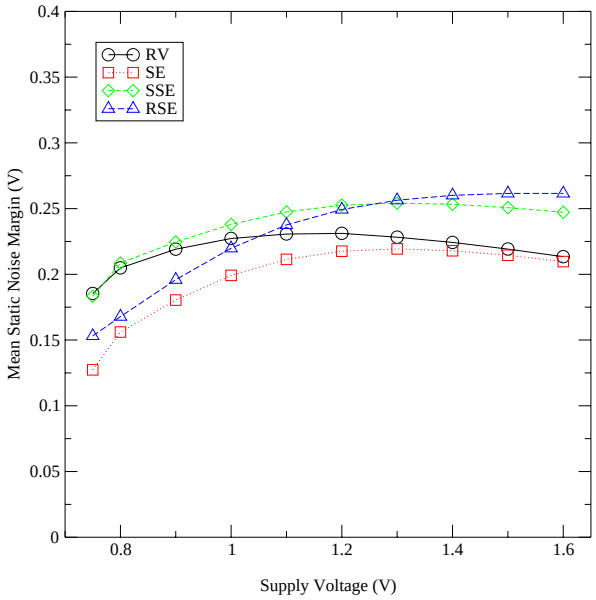
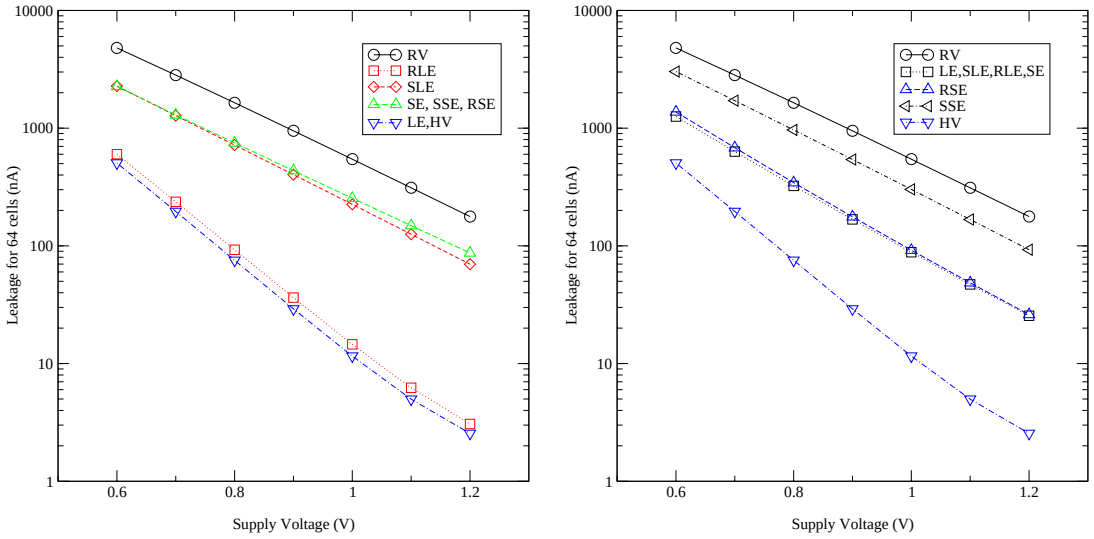


Figure 3.23: Mean SNM for different supply voltages for SE derived cells



(a) Holding a '0'

(b) Holding a '1

Figure 3.24: Leakage as supply voltage and threshold value are scaled down

$$V_{t_{low}} = V_t - V_t(1.2 - V_{supply}) \tag{3.3}$$

$$V_{t_{high}} = V_{t_{low}} + 0.2(\frac{V_{supply}}{1.2}) \tag{3.4}$$

The bitline discharge times on the fast side of the cell and write times for all cells are shown in Fig. 3.25 and Fig. 3.26 respectively. While the discharge time of the LE and SLE cells are slightly longer than that of the RV cell, they are much shorter than that of the HV cell for the whole range of supply voltages. The discharge time for the SE, SSE and RLE cells remain virtually unchanged from that of the RV cell for all voltages. The cell flip times of the asymmetric cell all lie in between the cell flip times of the RV and HV cells, but, as seen in Fig. 3.25 and Fig. 3.26, are just a fraction of the discharge times.

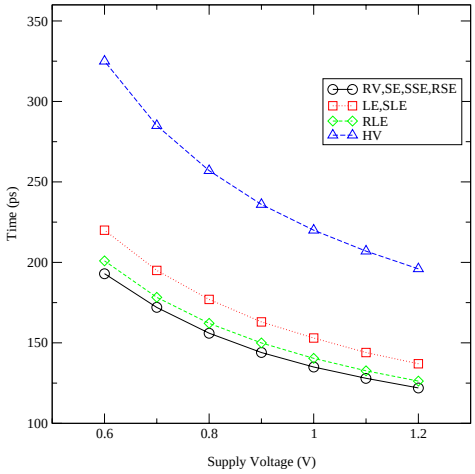


Figure 3.25: Read times at different Supply Voltages

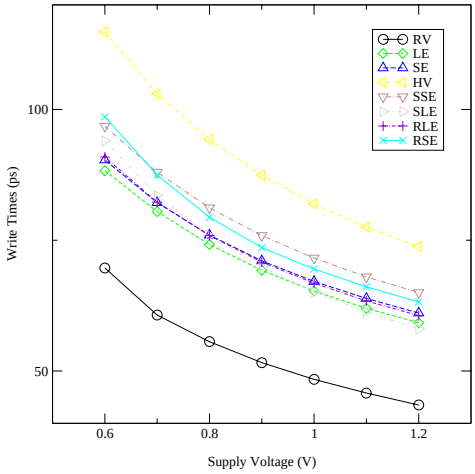


Figure 3.26: Write times at different Supply Voltages

3.9 Summary

A family of high-speed asymmetric dual-V_t SRAM cell designs that exploit a bit-level bias to reduce leakage power, while maintaining high performance, was introduced. The six best asymmetric cells offer different performance/leakage power/stability characteristics. A summary of the results pertaining to all the cells, relative to the RV cell, is shown in Table 3.14. Here, "Leakage (0)" and "Leakage (1)" refer to the leakage when the cell is storing a '0' and a '1', respectively, "Delay" refers to the bitline discharge delay, and "Area" refers to the total cell layout area.

Table 3.14: Summary results for all the cells.

Cell	Leakage (0)	Leakage (1)	Δ Delay	Δ Stability (SNM)	Δ Stability ($I_{\text{trip}}/I_{\text{read}}$)	Δ Area
RV	100%	100%	0%	0%	0%	0%
LE	1%	14%	12%	7%	-5%	0%
SE	14%	50%	0%	-6%	15%	0%
SLE	14%	43%	12%	23%	-7%	0%
SSE	50%	53%	0%	9%	13%	0%
RLE	2%	14%	4%	22%	5%	2%
RSE	15%	49%	0%	8%	15%	3%

4 Sense Amplifier/SRAM

This Universal Reality cannot be sensed, it cannot be seen

Abdu'l-Baha

4.1 Introduction

Throughout chapter 3 the cell design progressed with the assurance that bitline discharge degradation along BL was unimportant due to the ability to match the sensing time for a '0' to the sensing time of a '1.' A conventional sense amplifier, shown in Fig. 4.1(a), is not suitable for asymmetric cell designs due to the slow access time if the cell is storing a '0.' To obtain fast read times irrespective of the data, a new sense amplifier was designed and is shown in Fig. 4.1(b). The design of the sense amplifier is based loosely on MOS Current Mode Logic (MCML) ideas presented in [MR00]. Compared to the conventional sense amp, the new sense amplifier has four extra transistors and an area increase of roughly 0.229 μm^2 or 14.4%. In addition, the new circuit uses a set of *dummy bitlines*, D and DB, which are always fast[1] to trigger the reading of a logical '0' thus achieving fast access times when the slow bitline is discharging.

Sensing a '1' requires the same amount of time as a conventional sense amp since this is done by sensing a discharge of BLB due to the action of the fast side of the cell. Sensing a '0' is initiated at a later time than it would be in a conventional sense amp. This is done to allow sufficient time for the fast side to trigger the sense amp if it has to do so. Although initiating the sensing for a '0' is delayed, the combined effect of the

[1] as fast as the fast side of the asymmetric cell

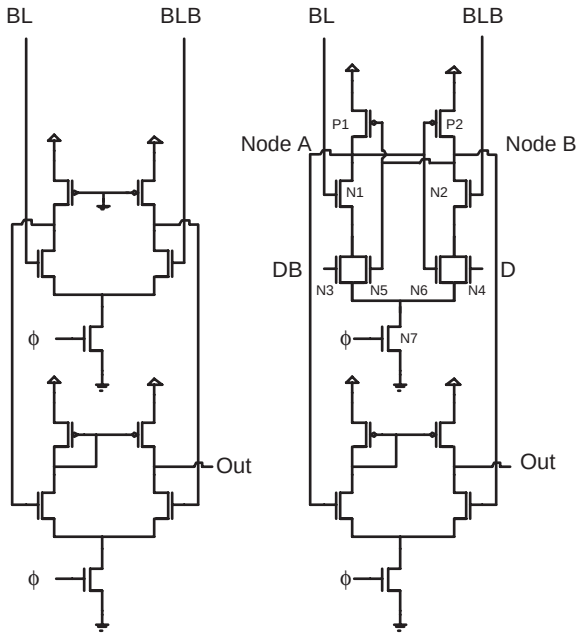


Figure 4.1: (a) Simple Sense Amplifier. (b) New Sense Amplifier

dummy cell and the slow side of the asymmetric cell makes the sensing process itself much faster once initiated. Therefore the end result becomes available at the same time as it would with a conventional sense amp.

This chapter will explain the architecture and operation of the new sensing scheme. Finally, the performance of the sense amplifier, within an SRAM, will be presented.

4.2 Sensing Architecture

To achieve quick sensing when cells are storing a '0', an additional column of cells are added to each bank of SRAM cells as seen in Fig. 4.2. The cells within this additional dummy column are all initialized to hold a logical '1', which allows BLB of the dummy column to discharge quickly. From this point on, the BL and BLB nets for the dummy column will be referred to as D and DB respectively.

The bitlines from the dummy column are connected to the D and DB terminals of all sense amplifiers within the SRAM block. During every read operation one of

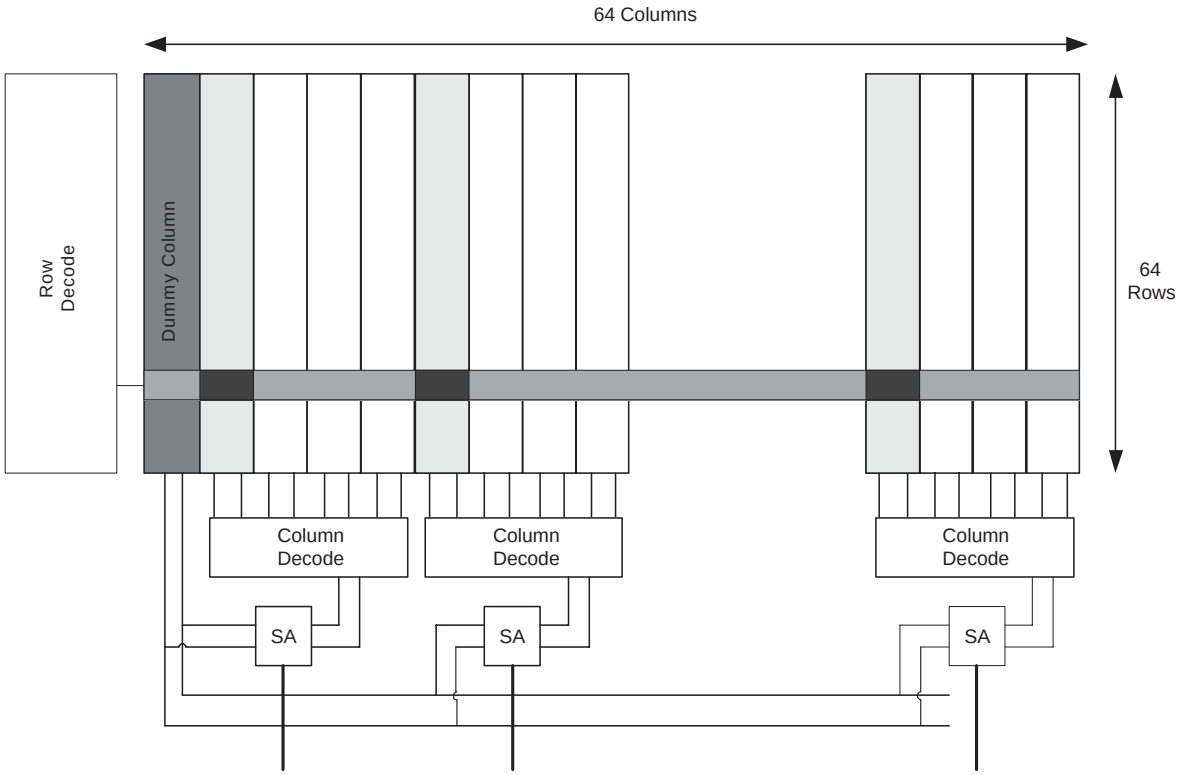


Figure 4.2: Dummy Column Architecture

the dummy cells will have its wordline asserted, and DB will be discharged. The D signal will remain near V_{DD} at all times, and is used as a reference for DB allowing for differential signaling, which has better noise immunity.

4.3 Circuit Operation

The new sense amplifier has two gain stages: the first stage amplifies the differential signal and the second stage, which is exactly the same as the second stage in the conventional sense amplifier, amplifies the signal even further and provides a single-ended output through a current mirror. Since the second stage of the amplifier is a well-known circuit, the rest of the discussion will center on the first stage of the sense amplifier.

The first stage of the sense amplifier operates as follows: initially, the bitlines are precharged and all four amplifier inputs rise to V_{DD}. During this phase the sense amplifier is being reset and nodes A and B on Fig. 4.1(b) are reset to an intermediate value. During a read operation, either BLB (cell's fast side) or BL (cell's slow side) will be discharged. Furthermore the signal DB, which is on the fast side of the dummy cell, will be discharged since the dummy cells permanently hold a logical '1.'

If BLB is being discharged (a logical '1' is being sensed), then the differential pair composed of N1 and N2 causes increased current to pass through the left branch, thus increasing the voltage at node B and decreasing the voltage at node A. Through the positive feedback loop of P1, P2, N5, and N6, the rate of change for nodes A and B are increased to achieve quick sensing.

When BL is being discharged (a logical '0' is being sensed), then it does so at a slower rate since it is being discharged from the slow side of the asymmetric cell. The dummy bitlines, which are connected to the differential pair of N3 and N4, initiate the sensing of a logical '0' by reducing the current in the left branch, and initiating the positive feedback. Through the combined effect of the DB bitline being discharged and BL being discharged, albeit at a slower rate, approximately symmetric sense times are achieved.

The operation of the sense amplifier can also be understood from the following two circuit simplifications. First, consider the situation when a '0' is being read and the slow side of the cell is discharging. For simplicity, assume that the fast signals from the dummy column are much faster than the slow signal from the read column. In this case, the gates of N1 and N2 are at V_{DD} due to precharging, and the transistors are on. Since the transistors are conducting they can be replaced as short-circuits, and the sense amplifier simplifies to the circuit shown in Fig. 4.3. When DB has not been discharged, nodes A and B are being pulled low by N3 and N4. Once DB starts to discharge, N3 turns off and there is nothing keeping node A low, and the back-to-back inverters latch a '0.'

When the sense amplifier is reading a '1,' the normal and dummy bitlines are equally fast, and N1 and N2 cannot be simplified away. However, the circuit can still be simplified to the circuit shown in Fig. 4.4. In Fig. 4.4, node B has its pull-down

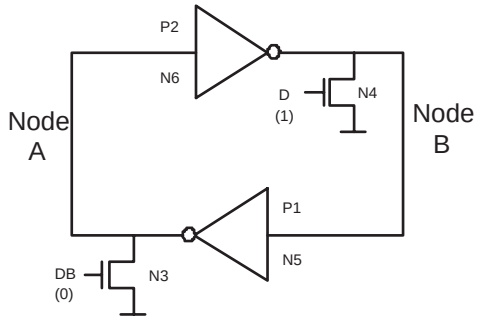


Figure 4.3: Approximate Sense Amplifier Circuit when reading a '0'

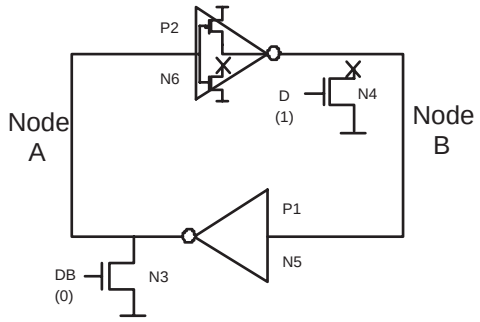


Figure 4.4: Approximate Sense Amplifier Circuit when reading a '1'

path blocked due to BLB discharging. Node B is then pulled high by P2 since N2 has cut off N4's and N6's ability to pull the node low. The circuit is then forced to latch the correct value.

For this sensing scheme to achieve reliable results, it must allow for adequate time for BLB to discharge before initiating a logical '0' read. This safety factor is achieved in two ways. First, the dummy bitlines are connected to all sense amps and, therefore, have a slightly higher capacitive load compared to real bitlines leading, to a slower discharge on DB compared to BLB. The extra capacitive loading does not slow the sense time when BL is discharging because of the combined effort between BL and DB to sense the same value. Second. the transistors connected to the bitlines are wider than the transistors connected to the dummy bitlines leading to a higher transconductance. This creates a higher gain from the bitlines to the output than from the dummy bitlines. Furthermore, since N3 and N4 are in parallel with N5 and N6, they cannot constrain

the current along the branch as much as N1 and N2 can.

4.4 Simulated SRAM

A 4-Kbyte SRAM was designed and simulated to measure the read times of the new sense amplifier. The SRAM was composed of 8 sub-arrays that each contained 64 rows and 64 columns. The layout of the SRAM is shown in Fig. 4.5, while the layout for the SRAM cell, sense amplifier and decoder are shown in Figures 4.6, through 4.8.

The SRAM was simulated at a temperature of 110°C with the RV, LE, SE, SLE, SSE, RLE, RSE and HV cells. Furthermore, the RV and HV cells were also simulated with a conventional sense amp, and these results were used as a reference for the new sense amplifier. The following sections detail the read and write times of the SRAM as well as some performance characteristics of the new sense amplifier.

4.4.1 Sense Times

The total SRAM read access time includes four components:

1. input register propagation delay.
2. the address decoding delay.
3. the delay for wordline, bitline and sensing (i.e. the time period from when precharging is complete to when the sense amplifier output has reached 90% of its swing.)
4. the output register setup time.

The simulation results showing these components for the various SRAMs composed of different asymmetric cells are shown in Fig. 4.9. Notice that only the sense times (the 3rd component of Fig. 4.9) is affected by the cell design.

Fig. 4.10 shows the sense times in more detail, but the SLE and SSE cells are not shown for clarity since their sense times mirror the LE and SE cells respectively. It can be seen that the worst-case sensing times of the asymmetrical cells with the new

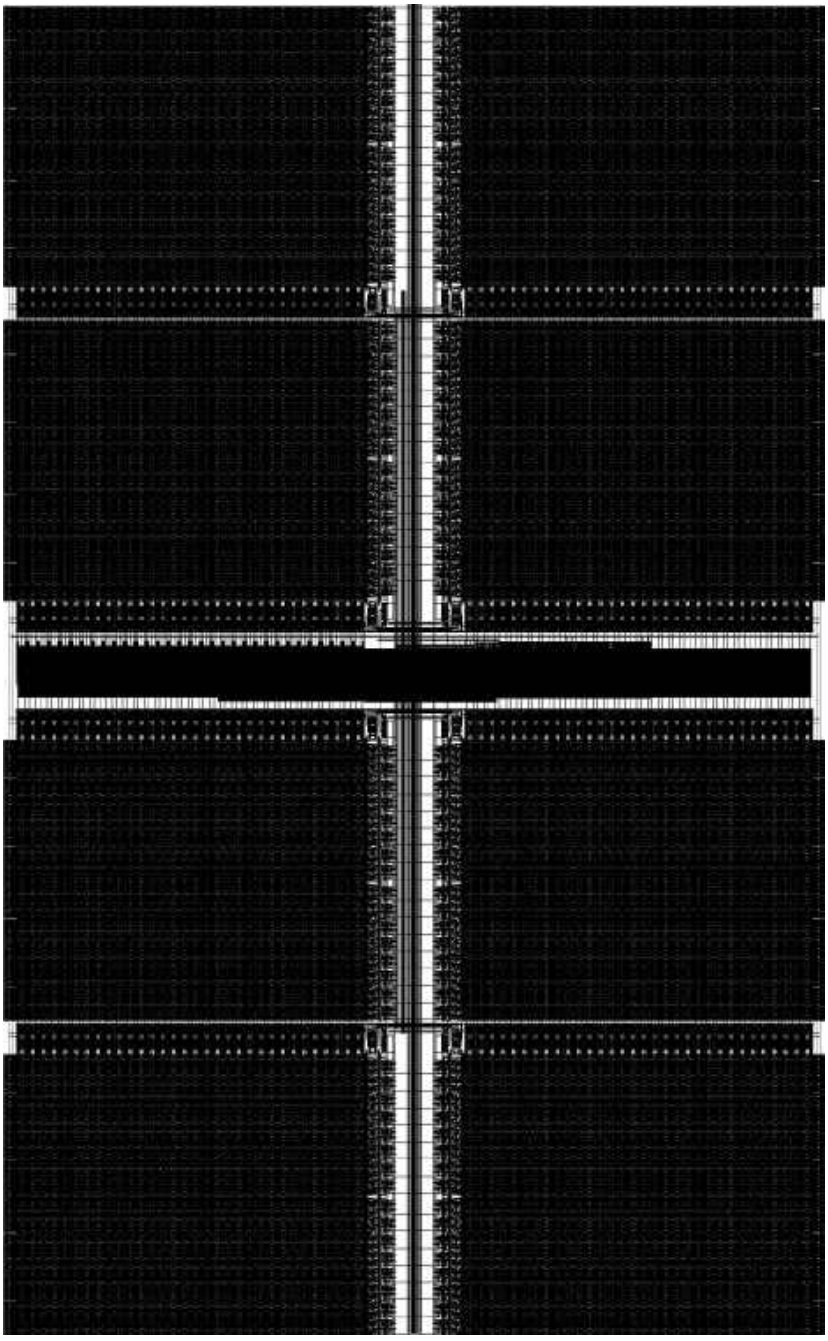


Figure 4.5: Layout of Simulated SRAM

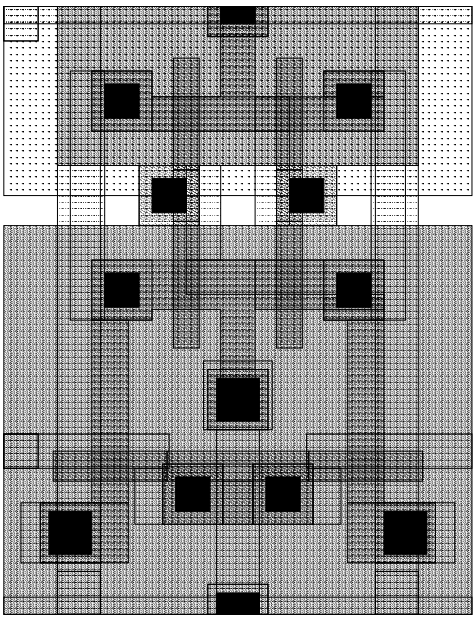


Figure 4.6: Layout of SRAM cell

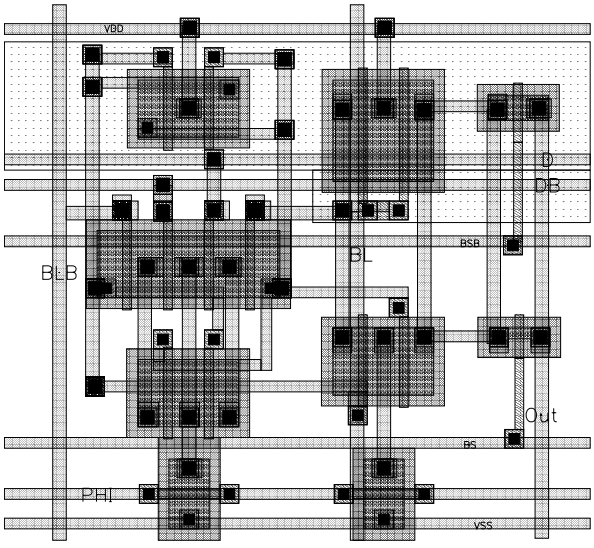


Figure 4.7: Layout of Sense Amplifier

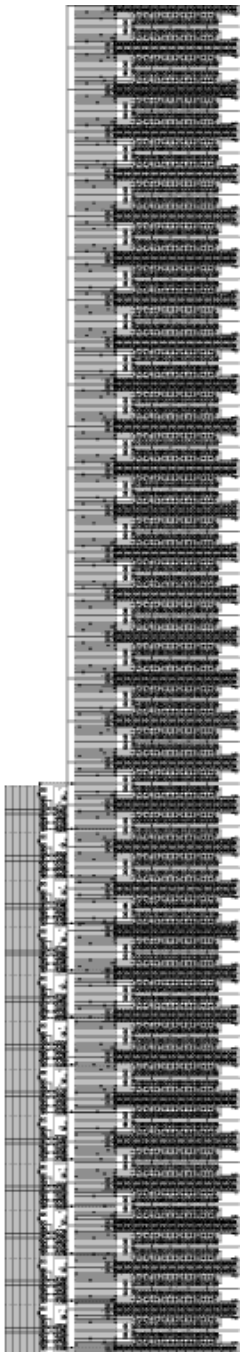


Figure 4.8: Layout of Decoder

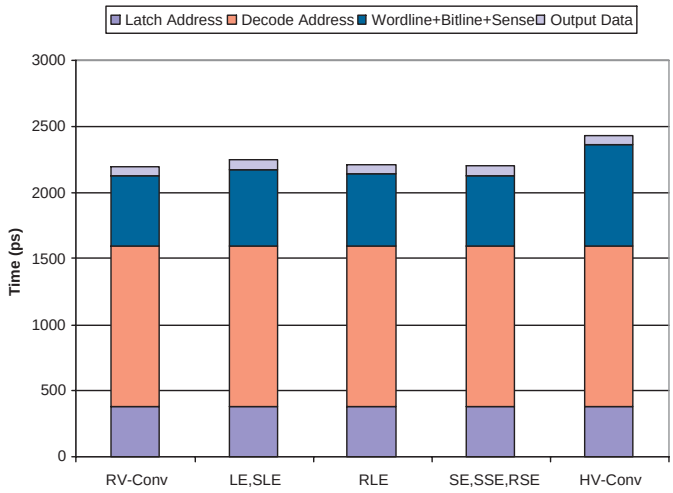


Figure 4.9: Breakdown of Memory Access Time

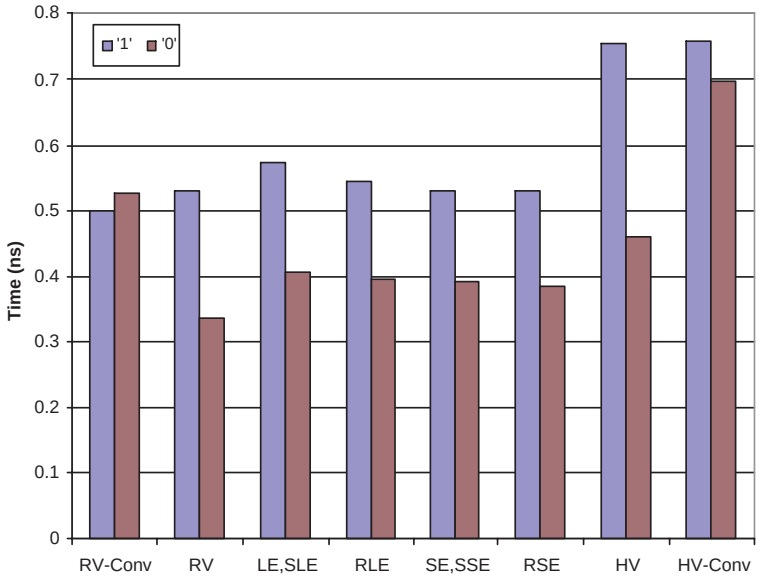


Figure 4.10: Sense times during a read cycle, i.e., the 3rd component of Fig. 4.9.

sense amplifier are now on-par with the RV cell with a conventional sense amplifier. In comparison with the RV cell using a conventional sense amp, the LE cell is 8.5% slower (although the effect on the total read time is an increase of just under 2%, as seen in Fig. 4.9). The SE cell, on the other hand, has sense times that are within 1% of the sense times of the RV cell. The RLE cell has a sense time which is only 3.4% longer than that of the RV cell, and the RSE cell has a worst-case sense time that is equal to that of the SE cell. It is interesting to note that the HV cell with a conventional sense amplifier is 43% slower. Furthermore, the new sense amplifier does not speed up the sensing for the RV and HV cells when compared to the sensing with the conventional sense amplifier; the RV and HV cells with the new sense amplifier have worst-case sense times which are 1% slower than the sense times with the conventional sense amplifier.

Read Margin

When a '1' should be read there is a race between BLB and DB to read a '1' and '0' respectively. If the signal along DB is faster than usual, while the signal along BLB is slower than usual there is a possibility that an incorrect value may be read by the sense amp. Therefore a margin has to be designed into the SRAM array; more specifically the discharge time along DB must be slower than BLB so that an incorrect value is not read. The built-in margin must also consider the variations in discharge times along BLB and DB.

Monte-Carlo analysis was performed on the different asymmetrical cells to find the mean and standard deviation of the read current from each side of the cell. The sense amplifier was then tested assuming the read current along bitlines could vary by $\pm 3\sigma$ from its mean value. The worst-case possibility when reading a '1' is when the signal along BLB is 3σ slower and the signal along DB is 3σ faster. The worst-case possibility when reading a '0' is when the signal along both BL and DB are 3σ slower. Under all conditions, the sense amplifier sensed the correct value.

To determine the margin that existed until the sense amplifier would sense the incorrect value, the signal along DB was incrementally made faster and BLB made slower until the sense amplifier would output a '0' when BLB was discharging. When DB was 3.74σ faster than it's nominal value and BLB was 3.74σ slower than it's nominal

value, the sense amplifier read the incorrect value. Noting that the capacitance along the dummy bitlines was 13.6% larger than the capacitance along the normal bitlines, the signal along the dummy bitline can be up to 17.4% faster than the signal along BLB before the sense amplifier reads the incorrect value in this design. Thus the sense amplifier will perform correctly as long as Equation 4.1 is satisfied:

$$\frac{I_{read_{dummy}}}{C_{dummy}} \leq 1.174 \frac{I_{read_{fastbitline}}}{C_{bitline}} \tag{4.1}$$

Furthermore, the capacitance along the dummy bitline, C_{dummy}, can become a design consideration. C_{dummy} can be controlled by connecting the dummy bitlines to different numbers of sense amplifiers, and not routing the bitlines through a column multiplexer. By lowering C_{dummy}, the '0' read time will become faster, and the '1' read time will become slower. At the same time the probability of a sense failure will increase. Conversely, by increasing C_{dummy}, the '0' read time will increase, and the '1' read time will decrease. The probability of a sense failure will also decrease. In this design C_{dummy} can be increased to make the read times between a '1' and a '0' more symmetric, and at the same time increase the margin before an incorrect value is read.

4.4.2 Write Times

The write times for the different cells are shown in Fig. 4.11. The write times range from a 13.4% increase over the RV cell's write time for the SSE cell to a 27.6% increase for the RSE cell. The increase in write times is of minor importance since the write times are shorter than the read times associated with the cells and therefore the speed of the SRAM is dependent on the read time.

4.4.3 Power Consumption

To limit the sense power, the sense amplifiers are clocked as in [HKM+90][ISN95][AH98]. The sense clock turns on the amplifiers and initializes them in their high gain region before the sensing occurs. To improve yield and ensure low-power operation, the clock path must be matched to the data path. This matching is achieved by using an extra

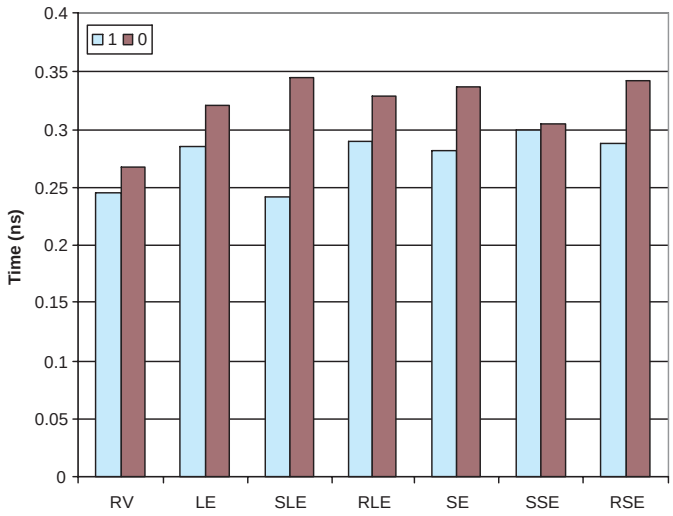


Figure 4.11: Write Times

set of dummy bitlines to match the bitline delay and clock the sense amplifiers at the appropriate time as in [AH98].

The energy consumed when the sense amplifier is in its high gain regions was measured while reading a '1' and a '0', and in the worst-case the LE and SLE cells dissipated 1.2% less energy than the RV cell with the regular sense amplifier. With the other asymmetric cells, the sense amplifier dissipates even less energy due to the quicker read time. The RV cell with the new sense amplifier has a 9.4% decrease in energy consumption, which follows from the observation that reading the RV cell with the new sense amplifier is slower.

When the sense amplifiers are off (when ϕ is low), the new sense amplifier has a leakage current that is 1.6 times the leakage of the conventional sense amplifier. This leakage current is, however, less than a hundred thousandth of the leakage within a single RV cell. The low leakage is attributed to the the body effect induced by the stacking of transistors, which effectively increases the V_t of the transistors.

Given that the leakage of the sense amplifiers is insignificant, and that the sense power of the new sense amplifier is comparable to the conventional sense amplifier, the leakage and power savings obtained with the asymmetrical cells are not affected, due

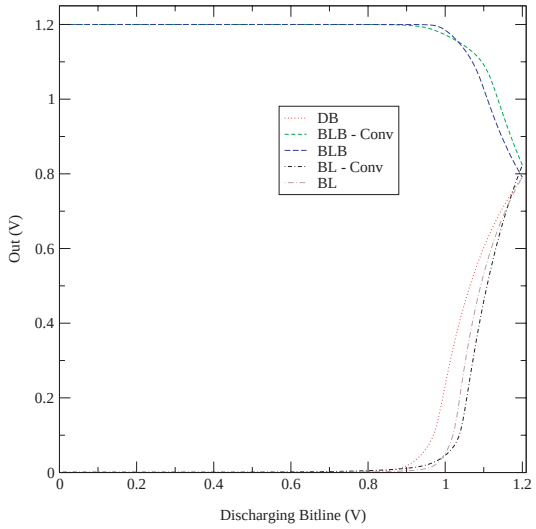


Figure 4.12: Sensitivity of New Sense Amplifier

to changes in the periphery of the SRAM.

4.4.4 Sensitivity

Sensitivity analysis has also been performed on the new sense amplifier, and it is not as sensitive to any individual bitline as the conventional sense amplifier, which makes it more noise insensitive compared to the conventional sense amplifier. Fig. 4.12 shows the effect on the output when either one of the bitlines (BL or BLB) for the conventional sense amplifier is discharging or when one of the of the bitlines (BL, BLB or DB) is discharging independently with the new sense amplifier. Note that the new sense amplifier is not as sensitive to any of the individual bitline signals, with it being least sensitive to the DB signal, as is the conventional sense amplifier to any of its bitlines.

4.5 Summary

A new sense amplifier that uses dummy bitlines to achieve quick sensing for the asymmetric cells regardless of data being stored was presented. The SRAM composed of SE cells is as fast as an SRAM composed of RV cells with conventional sense amplifiers.

The SRAM composed of LE and RLE cells have sense times which are 8% and 3.5% longer than that of the RV cell with a conventional sense amplifier respectively.

5 Architectural Enhancements

So powerful is the light of unity that
it can illumine the whole earth

Baha'u'llah

5.1 Introduction

In this chapter it is shown that the distribution of bit values (zeros or ones) in SRAM cells is highly skewed towards having a higher number of zeros, both in the data and the instruction streams. Memory bitstreams are more likely to have zeros than ones for several reasons. For example, in the data stream, zero and relatively small positive integers (such as loop iteration counts) are common in computation and are often stored as words. Also, heap objects are typically fragmented and may be zeroed at allocation time leading to a large fraction of zeros in cache blocks. Linked data structures in many systems (e.g., various flavors of Unix) often have zeros in the upper pointer address bits because the heap grows downwards. Similarly, the distribution of values in the instruction stream is skewed towards a higher number of zero bits. For example, immediate values, address displacement offsets, and branch displacements are often small positive integers [HP96]. In addition, modern instruction sets use a sparse encoding of opcodes to accelerate decode time contributing to the skew towards the zero bits.

Two Asymmetrical Cell Cache (ACC) designs are proposed: the first is statically biased towards the zero bit value. The second, uses dynamic selective inversion to increase the fraction of cache bits that hold a zero. In both caches, the cells of a conventional cache are replaced with asymmetric ones. The statically biased cache is

statically biased to dissipate low leakage power only when it stores the preferred bit value. These cache architectures are successful due to program behavior. As is shown in section 5.3.1, the SPEC2000 programs that were studied exhibit a strong bias towards zero. In principle, it is possible for a program to exhibit the reverse behavior, reducing the power benefits of this design; however, it would still dissipate much lower leakage power compared to the regular-V_t cell cache.

Furthermore, this chapter also documents the distribution of bit-values in other SRAM based structures within microprocessors such as register-files and branch prediction tables. Different structures and formats, such as coding and inversion, are analyzed for the SRAM based structures that can increase the skew in the bit-value distribution towards zero.

5.2 General Issues

5.2.1 Methodology

To evaluate the bit value distribution in the caches, Simplescalar v3.0 was used to simulate an aggressive wide-issue dynamically-scheduled superscalar processor. Table 5.1 depicts the processor configuration parameters that were assumed in this study. In all of the experiments, split Level-one Data (L1D), Level-one Instruction (L1I) caches and a unified Level-two (L2) cache were assumed. Results for systems with an L1I and an L1D of either 32 or 64 Kbytes each (32-byte blocks, 2-way set-associative) and an L2 of 1 Mbytes (64-byte blocks, 4-way set-associative) are presented. Characterization of the bit values in L2 was experimented with and the results were found to be quite similar to L1D results due to: (a) SPEC2000s tiny instruction footprints allowing the data streams to dominate occupancy in L2, and (b) SimpleScalar v3.0 only simulating a uniprogrammed environment with a single application footprint in L2. As such, only the characterization of bit values in level-one caches is presented. Similarly, bit value characterization in the cache tag arrays are not presented despite finding the values to be highly skewed, because SimpleScalar v3.0 does not simulate address translation of a realistic multiprogrammed environment.

Table 5.1: Base Processor Configuration

Branch Predictor	16K GShare + 16K bi-modal with 16K selector
Fetch Unit	Up to 8 instruction per cycle 64-entry Fetch Buffer 10 cycles branch misprediction penalty
Instruction Window Size	128 entries RUU-like
Load/Store Queue	64 entries, 4 loads or stores per cycle Perfect disambiguation
Issue/Decode/Commit Bandwidth	Any 8 instructions / cycle
Functional Unit Latencies	Same as MIPS R10000
L1/UL2 Access Latencies	3/16 cycles
Main Memory	Infinite, 100 cycles

Table 5.2: Applications used in the studies

App.	Ab.	Sim.	References		Skip	Miss Rates					
		Insts.	Total	Comm.		32k L1			64k L1		
						Instr.	Data	L2	Instr.	Data	L2
164.gzip	gzp	2B	968M	634M	1B	<0.1%	2.6%	2.2%	<0.1%	2.3%	2.7%
171.swim	swm	2B	659M	658M	1B	<0.1%	17.6%	30.8%	<0.1%	15.1%	34.4%
173.applu	apu	2B	763M	758M	1B	<0.1%	11.4%	27.7%	<0.1%	11.4%	11.3%
175.vpr	vpr	2B	1,008M	664M	4B	<0.1%	5.4%	10.3%	<0.1%	4.6%	12.1%
176.gcc	gcc	2B	1,106M	665M	0	0.6%	1.4%	2.0%	0.2%	0.8%	4.4%
177.mesa	mes	2B	763M	622M	1B	<0.1%	0.1%	27.5%	<0.1%	0.1%	0.28%
179.art	art	1.5B	679M	593M	0	<0.1%	34.6%	27.6%	<0.1%	34.6%	27.5%
181.mcf	mcf	2B	1,603M	736M	2B	<0.1%	26.0%	35.2%	<0.1%	24.6%	37.3%
183.equake	eqk	1.4B	550M	657M	0	<0.1%	4.4%	26.9%	<0.1%	3.8%	31.8%
188.ammp	amp	2B	866M	784M	2B	<0.1%	6.6%	11.2%	<0.1%	5.7%	12.7%
197.parser	prs	2B	1,118M	733M	3B	<0.1%	3.4%	5.1%	<0.1%	2.7%	5.9%
254.gap	gap	2B	1,209M	789M	4B	<0.1%	2.0%	24.2%	<0.1%	1.8%	27.3%
255.vortex	vor	2B	930M	862M	4B	0.6%	0.8%	4.5%	0.2%	0.5%	9.9%
256.bzip	bzp	2B	1,164M	1048M	3B	<0.1%	1.1%	33.4%	<0.1%	1.0%	34.7%
300.twolf	twf	2B	1,017M	632M	5B	<0.1%	9.1%	<0.1%	<0.1%	7.9%	<0.1%

5.2.2 Benchmarks Used

Table 5.2 depicts the SPEC2000 programs that are studied. The binaries are compiled for the Alpha 21264 architecture using Compaq's compiler for the Digital Unix V4.0F. For every application, Table 5.2 depicts the memory reference counts and the L1D and L1I miss ratios for the studied cache configurations. To account for variations in bit value distributions across compilations and instruction sets, results for SPEC2000 binaries compiled for PISA, the MIPS-like instruction set simulated by Simplescaler v3.0 are also presented. For brevity, for the rest of this document, applications will be referred by using the three letter abbreviations listed under the Ab. column.

5.3 Level 1 Caches

5.3.1 Bit Value Distribution of Ordinary Programs

Fig. 5.1 illustrates the bit value distribution in 32-Kbyte L1 caches. The graphs indicate that there is much opportunity to exploit the bias in the distribution of bits in the caches. In the applications that were studied on average, almost 80% of bit values are zero. Similarly, there is a considerable skew in the bit value distribution in the instruction cache. The instruction cache's results are, however, less dramatic and the fraction of zero bits in L1I is, on average, slightly over 60%. These results corroborate findings from recent research advocating zero-byte compression in caches especially due to the data stream [VZA00], zero-compression of operands in the pipeline datapath [CGS00] [BM99], and value prediction [Lip97].

Not surprisingly, the instruction cache does not exhibit much variation in the distribution of zeros across applications. It is expected that common opcodes, zero operand specifiers (i.e., immediate values, branch displacements, memory address offsets, and register zero) are more or less uniformly distributed across applications. Reduced Instruction Set Computer (RISC) instruction sets also tend to have sparse encoding of opcodes to facilitate and accelerate the decoding of instructions. In Alpha, the most common instructions such as the integer Arithmetic and Logic Unit (ALU) instructions have no more than two of the 14 opcode bits set to one.

Unlike instruction caches, data caches exhibit a large variation in the skew towards zero. Fig. 5.2 illustrates the breakdown of the zero distribution across memory addresses into data, heap, and stack segments. The results indicate that except for three applications, gzip, applu, and swim, the majority of zero-bit distributions are from the heap. Gzip maintains its main data structures (i.e., the input/output compression buffers) statically in the data segment. Applu and swim are FORTRAN77 applications and, therefore, do not use the heap. The rest of the applications allocate their data structure dynamically in the heap.

The data cache results can be divided into three groups:

1. The group of applications with predominantly dynamic data structures and a significant skew towards a high fraction of zero bits.

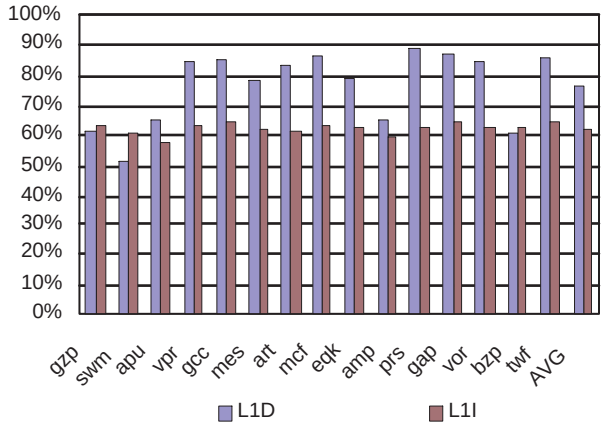


Figure 5.1: Fraction of cache bits that are zeroes for L1D and L1I for the ALPHA binaries for 32-Kbyte caches

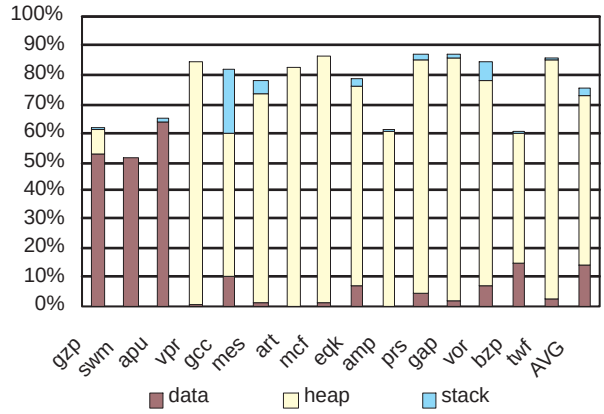


Figure 5.2: Memory space breakdown of zero bits for L1D

2. The compression applications.

3. The dense numerical (floating-point) applications.

The first group includes vpr, gcc, mesa, art, mcf, equake, parser, gap, vortex, and twolf, which all heavily use dynamic memory allocation. In these applications, the key contribution to the high skew towards zero bits is due to small positive values occurring frequently in computation. For instance, art and mesa compute over thermal and graphical images respectively, and benefit from a large distribution of zero image data bytes. However, there are secondary factors that may favor the skew towards zero. Dynamic memory allocation libraries typically allocate heap objects with a spectrum of pre-defined sizes (e.g., power-of-two size objects), irrespective of the data structure requirements in applications. Therefore, they are likely to exhibit fragmentation in heap objects, resulting in a high skew towards zero bits when these objects are loaded in the data cache. Moreover, compilers often align aggregates of structures to prevent misaligned accesses, thus padding aggregates with fields that remain zero. On a close inspection of the source code, it is found that some of these applications (e.g., art and mesa) request dynamic allocation for a small number ($\sim$16) of bytes, and inadvertently allocate a large number of small heap objects with a high probability of fragmentation rather than allocating one large object and dividing it up among a large number of small data structures.

The second group of applications, bzip and gzip, compress the input streams and as such reduce the longevity of large sequences of zero bits in the data cache during execution. As such, these applications exhibit less skew towards zero. The third group of applications, amp, applu, and swim are dense numerical computations, in which bit values are more uniformly distributed across zeros and ones. In the extreme case, swim bit values are almost equally distributed across zeros and ones.

Fig. 5.3 depicts the bit value distributions for a 64-Kbyte cache. It can be seen that the results are almost identical to a 32-Kbyte cache primarily because the application footprints do not change. This is corroborated by the small change in miss ratios, shown in Table 5.2, when increasing the sizes from 32 to 64 Kbytes. Fig. 5.4 compares the

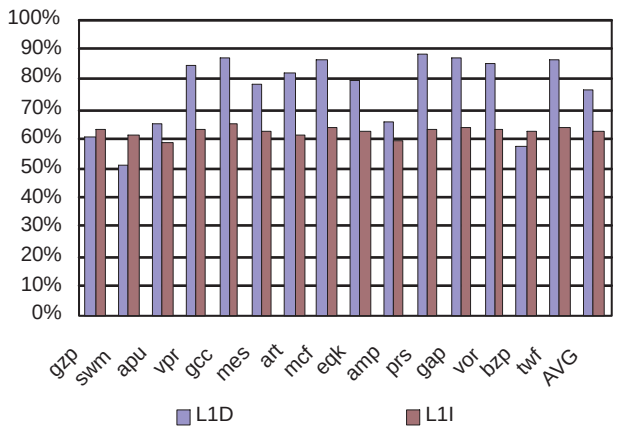


Figure 5.3: Fraction of cache bits that are zeroes for L1D and L1I for the ALPHA binaries for 64-Kbyte caches.

bit value distributions for L1D caches across Alpha and PISA binaries[1]. The results indicate a slight increase in the skew towards zero when using the Alpha binaries. Unlike PISA, Alpha uses 64-bit integer and floating-point values. There are many factors that may affect the skew across the binaries including, but not limited to, the compilers' use of 64-bit register values in Alpha,[2] which increases the skew of zero bits for small positive integers, and the allocation, alignment, and padding of dynamic data structures in memory.

5.3.2 Bit Values in Live Blocks

Recent architectural/circuit techniques to reduce cache leakage [KHM01] [YPF+01] have targeted identifying inactive cache block frames and using supply-gating [PYF+00] to reduce leakage in the inactive or dead frames. Asymmetric cells can be used in conjunction with these techniques to reduce leakage in both the live and dead cache block frames. In this section, the bit value distribution is evaluated in the live cache blocks.

[1]The comparison between bit value distributions in the instruction caches of Alpha and PISA binaries is not shown because PISA uses a 64-bit instruction format that is highly skewed towards zero.

[2]as compared to 32-bit values in PISA

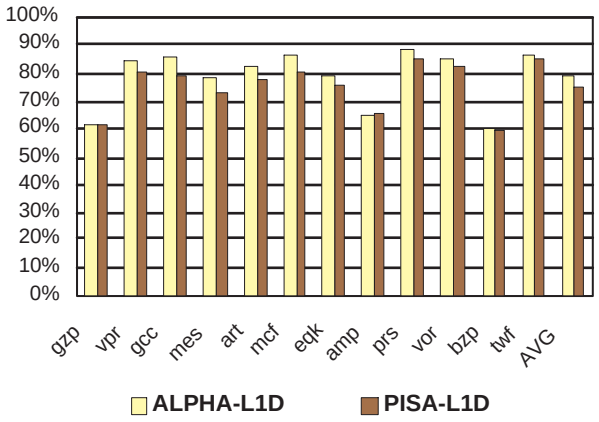


Figure 5.4: Comparing the bit value distribution across the ALPHA and PISA binaries for 32-Kbyte L1D caches.

Fig. 5.5 illustrates the fraction of zero bits in L1D and L1I for the live cache block frames; these are block frames to which a subsequent processor reference will result in a hit. The graphs indicate that, as expected, there is little change in distribution of zeros in the instruction stream. There are small variations towards a higher or lower skew depending on the application, affecting the average fraction of zero bits by 3%. Other applications either exhibit no change, or a slight reduction in the skew towards zero.

Fig. 5.6 illustrates the distribution of zero bits across the segments. Compared to Fig. 5.2, the breakdown indicates a significant shift from one segment of memory to another that contribute to the zero bit skew. In general the breakdown graphs indicate a higher shift in Fig. 5.6 towards the data segment and the stack due to three reasons:

1. The variables and data structures in the data segment are often used for lookup purposes (e.g., pointers to global structures that are actively referenced, constant scalars, tables holding constant values initialized statically by the compiler) remaining active and live.

2. Stack frames are always used for keeping temporary computation and as such always remain active.

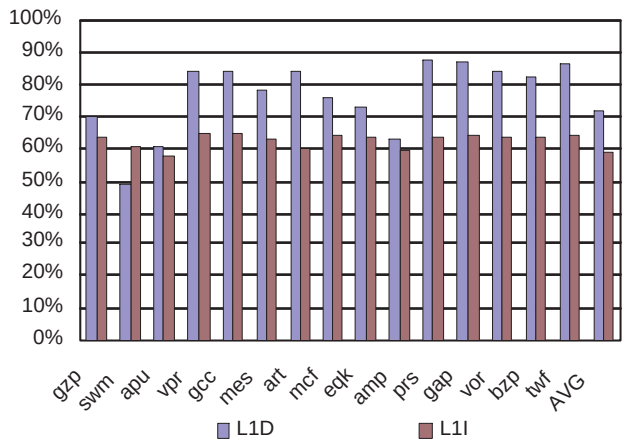


Figure 5.5: Zero bit fraction with 32K L1 caches and for only those blocks that hold live data.

3. The main data structures,which in most applications are allocated in the heap, are often swept once per application phase to compute new values and otherwise remain inactive or dead in the cache (e.g., updating of image pixels, or compressing data from an input buffer). Moreover, a reduction in live heap data may also result in a slight reduction in the skew towards zero due to the decreased contribution of zeros from fragmented heap objects.

Gzip, which makes heavy use of compile-time allocated data structures, where it stores the (uncompressed) input stream, is likely to have a large set of zero bits, while using the heap for scratch storage. Bzip, however, primarily uses statically-allocated arrays/buffers for scratch storage and, as such, exhibits a significant shift of zeros towards the data segment.

5.3.3 Selective Inversion

In selective inversion, the values stored within a block can be inverted to reduce leakage even further. For example, in one design, inversion is possible at the byte level; in this case, if a byte contains five or more ones it is inverted prior to storing it in the cache. For correct operation, it is necessary to also hold information on which bytes were

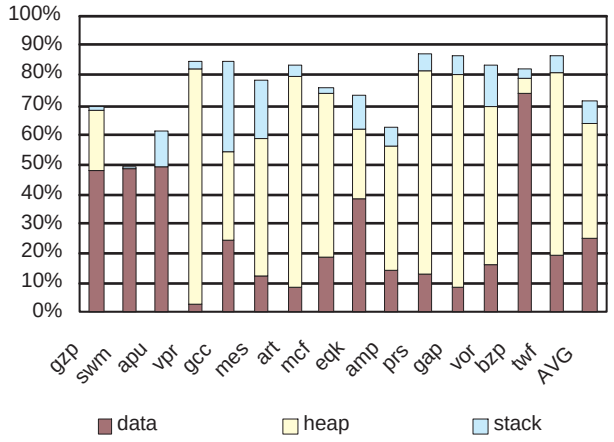


Figure 5.6: Memory space breakdown for live data.

inverted. To do so, an inversion flag, which is an extra SRAM cell, is introduced per byte. Selective inversion allows the fraction of cells that are in the preferred low-leakage power state to be increased. Another advantage of this method is that it reduces the maximum leakage power because it guarantees that at most 50% of the cells will be in the high-leakage power state. However, there are area and potential performance overheads associated with this technique. Inversion flags are required for each block of cells that may be inverted. These inversion flags add to the wordline length, and hence, may impact read and write access latencies. Moreover, upon reading a cache value, the value may have to be inverted. This inversion will, again, increase overall access latency. Circuits that perform the inversion are discussed in section 5.5.

Thus far, the granularity of inversion has been assumed to be a byte, however, other possibilities exist as well. For example, inversion could happen for every word (32 bits) or for a whole block. This reduces the number of inversion flags and hence, improves access time. However, if any other granularity other than a byte is used, partial writes will have to be converted to a read-modify-write access to maintain appropriate inversion. Moreover, using a larger than a byte granularity increases the complexity of the circuit that determines whether a value should be inverted or not.

In Fig. 5.7 the increase in zero bits with byte-level selective inversion is reported.

Fig. 5.7(a) shows this increase ignoring the inversion flag bits that are necessary. For example, in swim, the fraction of zero cells increases from 52% (Fig. 5.1) to 67%. Selective inversion is typically more effective for instructions than for data, since as it is shown in section 5.3.1, instructions have less zeros than data.

Selective inversion introduces additional inversion flags imposing overheads on power, area and potentially latency. In this design, one inversion flag is introduced per 8 bits of the original cache, introducing an additional 12.5% cells. Fig. 5.7(b) shows the fraction of those flag bits that hold a zero (not inverted). On average, 80% and 70% of the inversion bits hold a zero for the data and instruction caches, respectively.

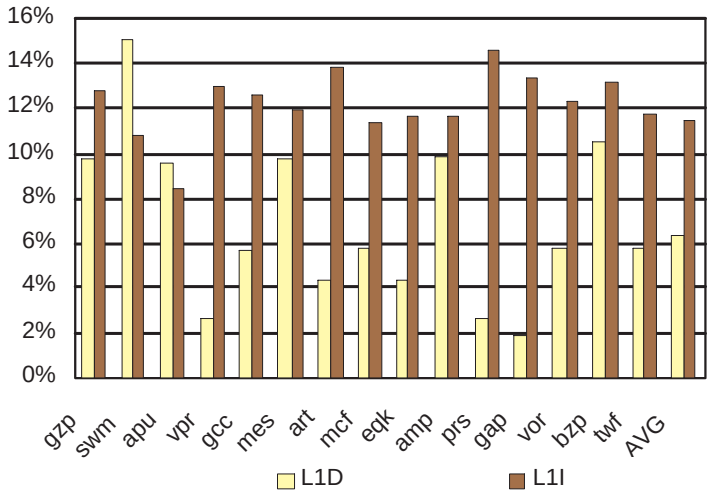
The reduction of leakage power by selective inversion also depends on the relative leakage power at the zero and one bit states. As we have seen for the given CMOS technology and applications, the overall benefit is small. Nevertheless, the results are presented since selective inversion bounds worst case leakage.

5.3.4 Leakage Power Reduction

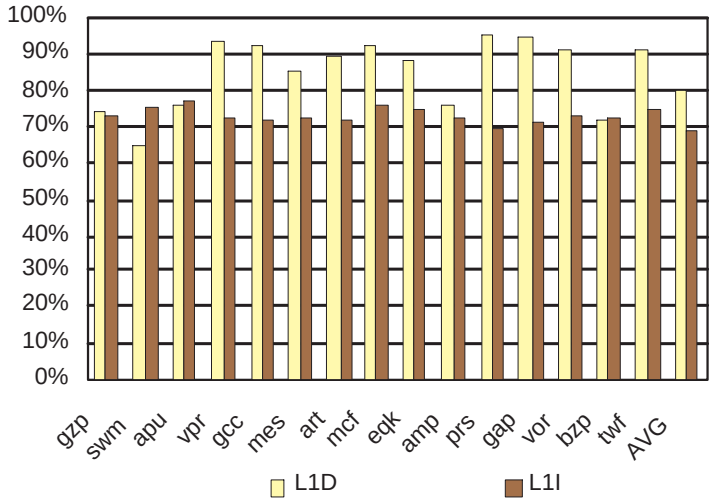
Fig. 5.8 shows the leakage power dissipation of various asymmetric cell caches. Leakage power is reported as a fraction of the leakage power dissipation of the RV cell cache. The chosen metric includes only the power dissipated by the cells of the data array. This is appropriate given that Simplescalar does not model a multiprogrammed environment and as a result tag bits are strongly skewed towards zero.

Fig. 5.8(a) shows the leakage power for the LE cell for the statically-biased cache and the cache using selective inversion. In all cases, leakage power is less than 8% of that of the RV cell. On the average, the statically-biased cache reduces leakage to 4% and 6% of the RV cell cache for the data and instruction caches respectively. By comparison, the HV cell cache reduces leakage to about 1.4%. Selective inversion, reduces leakage to 4% and 5% respectively. Compared to the leakage reduction that is achieved with the statically biased cache, this additional reduction is unimportant. However, in relative terms it is significant since it allows the asymmetric cell cache's leakage to be closer to the all high-V_t one.

Fig. 5.8(b) reports leakage power for the SE cell cache. Even for the data stream of swim that exhibited the lowest fraction of zeros, leakage power is reduced to 32%



(a) Increase in the fraction of zero bits ignoring inversion flag overhead



(b) Fraction of flag bits that hold a zero

Figure 5.7: Selective inversion at the byte level

of the RV cell cache (i.e., a reduction of about 3.1X). On average, the SE cell caches leakage power is only 23% and 26% of that of the RV cell data and instruction caches respectively. Selective inversion does not produce a discernible advantage. In some cases, it actually increases leakage power. The leakage power for the SE cell is only 3.2X higher when it stores a one as compared to when it stores a zero. The resulting reduction in overall leakage power with selective inversion does not compensate for the overhead introduced by the inversion flags. In future technologies, where the difference in leakage power between RV and HV cells is much larger (e.g., 104x [PYF+00]), selective inversion may be useful even with the SE cell.

The leakage power dissipated by the stability-improved cells is shown in Figures 5.8(c) and 5.8(d)[3]. On the average, the SLE cell cache's leakage power is only 20% and 24% of that of the RV cell data and instruction caches respectively, which is very close to the leakage power of the SE cell cache. Selective inversion does not produce a discernible advantage. For the SSE cell, the average leakage power is 49.8% and 50.2% of the RV cell data and instruction caches respectively. Since the leakage reduction in each state of the SSE cell is almost equal (2X and 1.9X), the overhead of selective inversion increases the leakage considerably, where the leakage of the SSE cell cache becomes 55.7% and 56.1% of the RV cell data and instruction caches' respectively.

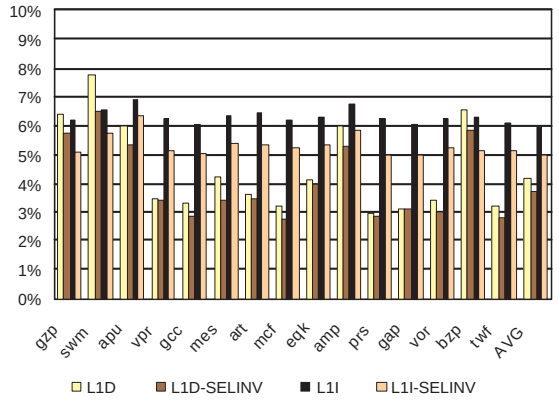
5.4 SRAM Based Structures

Since caches within microprocessors seem to have a large bit-level bias towards '0,' experiments were done to determine if a bias existed within other SRAM based structures in the microprocessor. More specifically, the distribution of bits in the General Purpose Register File (GPR) and the Floating-Point Register File (FPR) were documented, and the contents of bimodal and 2-level branch predictors where analyzed. Furthermore, the contents of the Branch Target Buffer (BTB) were also analyzed.

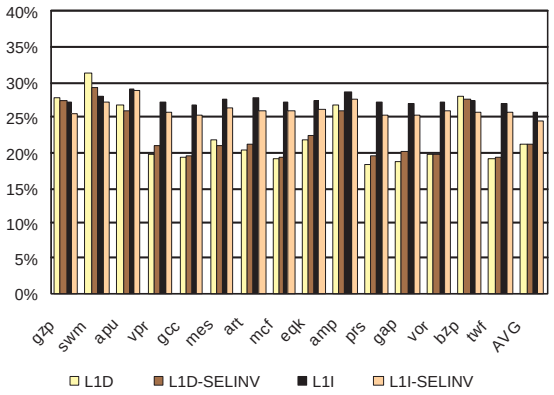
Selective inversion for the SRAM based structures was tested under many different policies. Table 5.3 describes the different policies[4]. While selective inversion incurs

[3]The leakage benefits for the RLE and RSE cells will not be presented in this chapter since they closely mirror the LE and SE cells leakage reduction

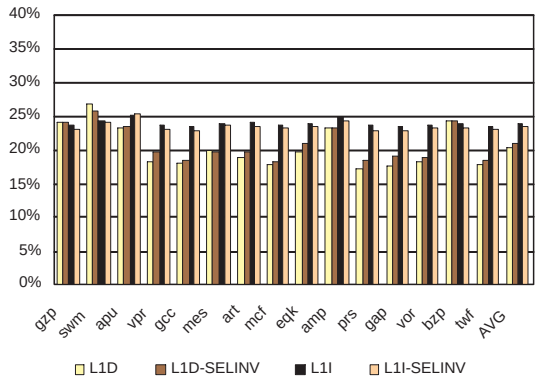
[4]For example, with the MajByte policy, bytes are inverted when the majority of bits within the byte



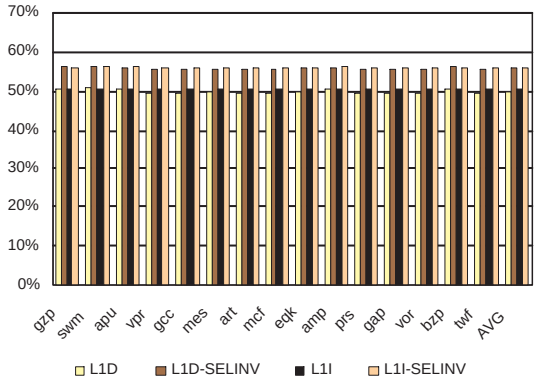
(a) LE cell



(b) SE cell



(c) SLE cell



(d) SSE cell

Figure 5.8: Data and instruction cache leakage power with the asymmetric cells relative to the RV conventional cache

Table 5.3: Inversion Policies evaluated for SRAM based structures

MajByte	Majority Count/Byte
1-0Byte	Invert bytes with zero or one '0' bits
Inv0Byte	Invert bytes with zero '0' bits
MajWord	Majority Count/Word
Inv0Word	Invert words with zero '0' words

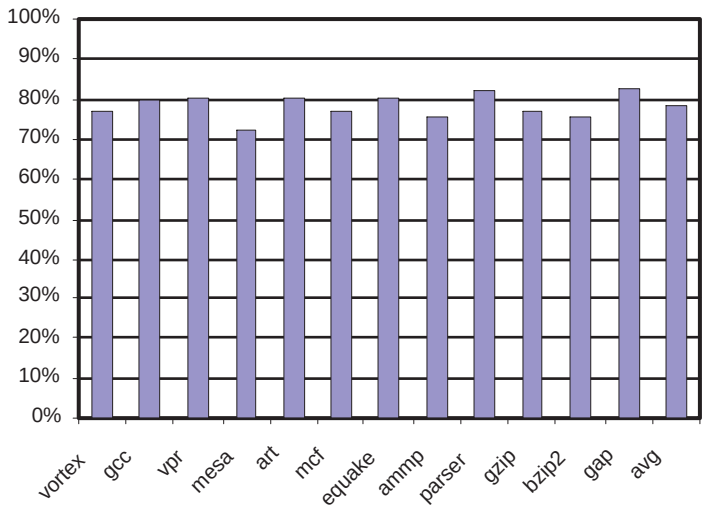


Figure 5.9: Fraction of GPR bits that are zeros

performance penalties that are too great for high performance register files and branch prediction tables, inversion will be analyzed since it bounds worst case leakage.

5.4.1 General Purpose Register Files

Fig. 5.9 illustrates the bit value distribution of the GPR. The data indicates that there are many opportunities to exploit the distribution of bits in the GPR. On average, 78% of bit values are zero in the applications.

are '1'.

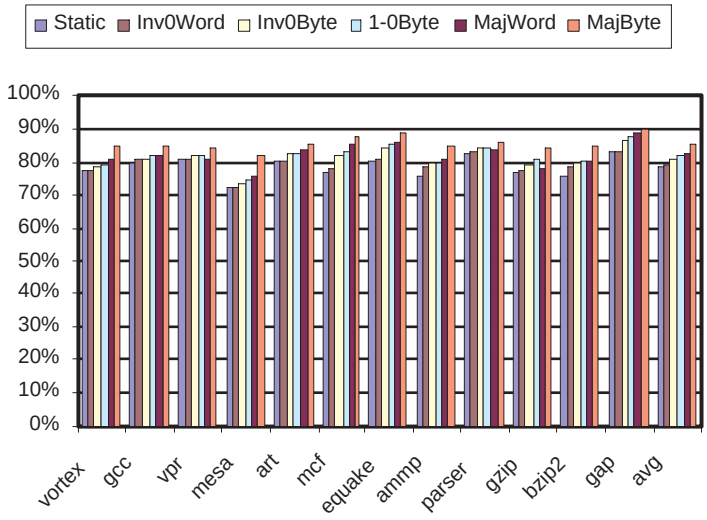


Figure 5.10: Fraction of GPR bits that are zeroes under different inversion policies

The static biasing towards zero within the GPR in programs allows for large leakage savings when the asymmetric cells are used. On average, the LE and SE register files reduce leakage to 4% and 22% of the leakage of the RV register file. The SLE and SSE register files reduce leakage to 20% and 49% of the leakage of the RV register file.

Fig. 5.10 shows the bit-level distribution under the different inversion policies. When data is inverted with the MajByte policy, the percentage of '0's within the register file becomes 85.4%. The simple 1-0Byte inversion policy increases the percentage of '0's within the register file to 82%. This simple inversion policy provides half of the additional leakage reduction of the most complex policy, MajByte.

Note the bit-level bias, seen in the GPR, follows the bias from the L1D very closely. Since the GPR can be viewed as a small cache for the L1D cache, the similarity in the bit-level bias is not surprising.

5.4.2 Floating-Point Register Files

The FPR shows a different behavior than that of the GPR. Since not all programs use floating point values, there is the possibility that the FPR will not be accessed and

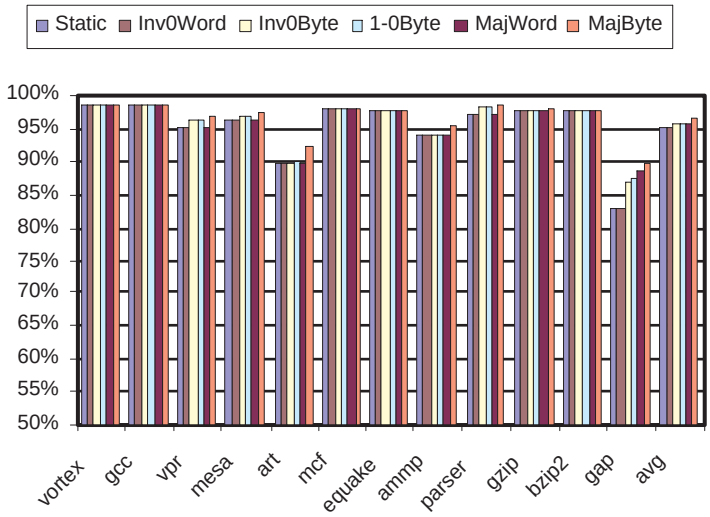


Figure 5.11: Fraction of FPR bits that are zeros

therefore contain a majority of '0's.

Fig. 5.11 shows the bit-level distribution among the benchmarks. Many programs show a zero-bit bias above 90%, with the minimum percentage of zeros occurring in gap with 82%. On average, the FPR has 95% of its contents filled with zeros. This large bias allows the LE, SE, SLE and SSE register files to reduce leakage to 2%, 16%, 16%, and 49% of the RV register file's leakage respectively.

With the MajByte and Inv0Byte selective inversion polices, the percentage of zeros becomes 96.6% and 95.8%. Selective inversion does not change the percentage of zeros much compared to the static case since the FPR bias towards zero is so large. Correspondingly the reduction in overall leakage power with selective inversion does not compensate for the overhead introduced by the inversion flags. For gap, however, which has the lowest percentage of zeros, selective inversion can increase the bias by an additional 7%.

5.4.3 Branch Predictors

Using SimpleScalar the contents of bimodal and 2-level branch prediction tables were analyzed. The predictors consist of 2 bits indicating if a branch would be taken or not. Fig. 5.12 shows the bit-level distribution of the bimodal predictor. There does not seem to be a consistent bias towards '0' or '1,' with an average of 50% zeros in the table. A lack of a bias, however, does not mean that the asymmetric cells will not decrease the leakage of the tables. Since the asymmetrical cells dissipate lower leakage in both states, the LE cell will reduce leakage to 8% of the original leakage, and the SE cell will reduce leakage to 31% of the original leakage. The SLE and SSE cells will reduce leakage to 27% and 50% of the RV table's leakage respectively.

The 2-level predictor exhibited much the same behavior as the bimodal predictors (for the second level tables). On average there was no clear bias for any bit level. Due to the lack of a bias for any bit value, leakage reduction for the asymmetrical cells was nearly the same as that obtained for the bimodal predictor. Fig. 5.12 shows the bit-level distribution for the 2-level predictor. The first level table for the 2-level predictor (which is quite small) showed a large bias towards '0's with nearly 85% of the bits being zero.

The lack of a bias in the bit levels in the branch prediction tables limits the leakage savings that are possible with the use of the asymmetrical cells. However, changing the way the values are stored in the branch prediction tables can recreate the asymmetry in the bit-level distributions. Branch prediction tables store one of four states, strongly not-taken, not-taken, taken, and strongly taken. If the values stored for each of these states are changed, then there is a possibility that an asymmetry in the bit levels may appear. (For example, say "00" originally indicated strongly not-taken, but instead the code is changed to "01"). Fig 5.13(a) shows the bit level distribution by changing the values stored for the four states to various other codings. With the new coding schemes there is a bias towards '1' or '0' for each program, but on average the best coding scheme (10,01,00,11) for (strongly not-taken, not-taken, taken, strongly taken) provides a distribution where 54% of the bits are zero. This extra degree of bias towards '0' does not decrease the leakage of the branch prediction table considerably.

The bimodal branch predictor, however, behaves quite differently. With the dif-

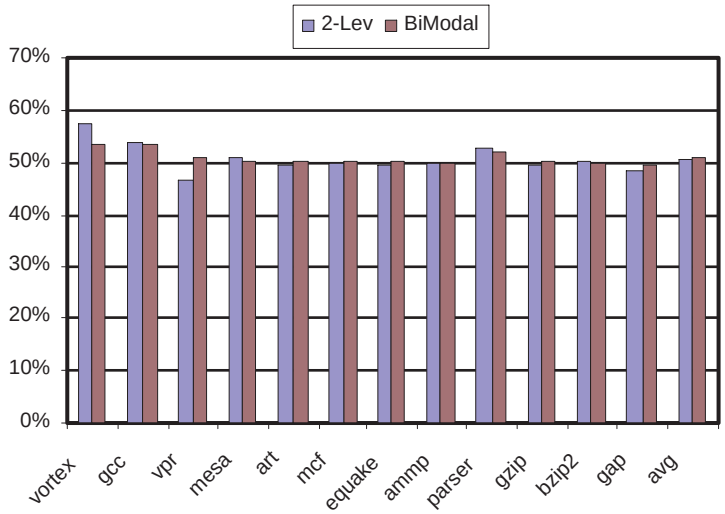
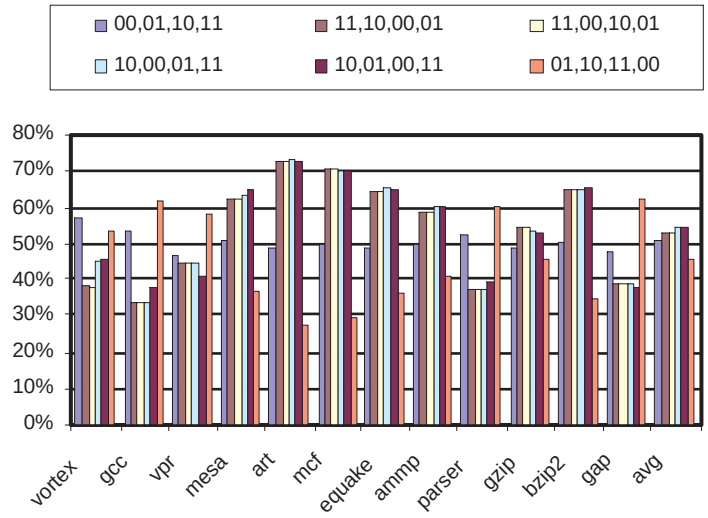


Figure 5.12: Fraction of bits in branch predictors that are zeros

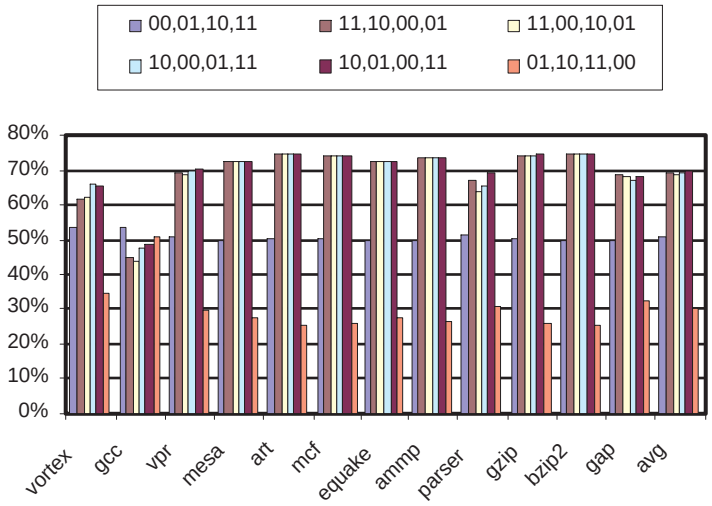
ferent coding schemes, shown in Fig. 5.13(b), a large bias is developed across most programs. On average the best coding scheme (10,01,00,11) develops a 70% bias towards '0.' With this coding scheme the LE, SE, SLE and SSE cells reduce leakage to 5%, 25%, 22% and 55% of the original tables leakage respectively. It is interesting to note, that while the original coding made it appear that the 2-level and bimodal branch predictors were quite similar, the use of different coding schemes shows that they are different.

5.4.4 Branch Address Predictors

The BTB target tables exhibited much the same behavior under all scenarios and thus only the results which were obtained with the bimodal predictor will be presented. Fig. 5.14 shows the percentage of zeros for all policies. A large proportion of programs show a very large bias towards '0', with 6 programs having biases toward zero above 95%. Other programs also showed a large bias of over 80% towards '0'; on average 92% of bit values were zero. Selective inversion does not change the proportion of zeros



(a) 2 Level



(b) Bimodal

Figure 5.13: Fraction of bits for Branch Predictors under different codings

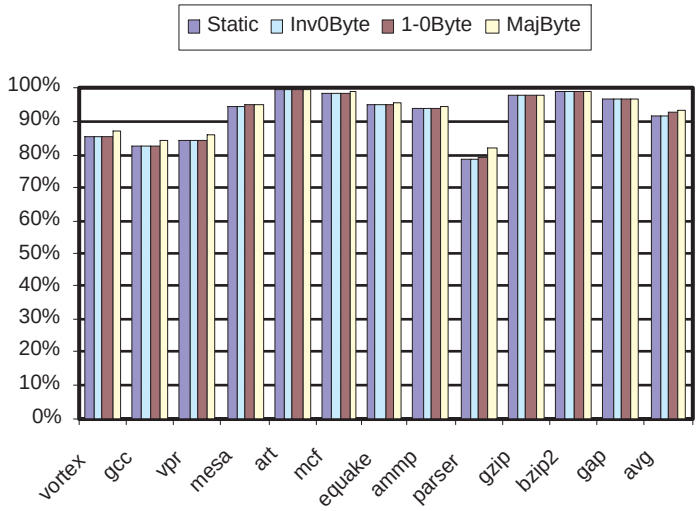


Figure 5.14: Fraction of zero bits in BTB Targets

much, due to the existing skew in bit-values. The leakage obtained with the use of the asymmetrical cells is 2.5%, 17%, 16% and 50% for the LE, SE, SLE, and SSE cells respectively.

The BTB does not only contain the target of the branch, but must also include the original branch address in a lookup table. Fig. 5.15 shows the results, which on average are quite similar to the BTB target tables. There is, however, more variation on a per program basis. On average, 90% of the bit values are zero allowing for a leakage reduction similar to that of the BTB target addresses. The large skew towards zero might be due to SimpleScalar v3.0 not simulating address translation of a realistic multiprogrammed environment.

5.5 Selective Inversion Circuits

With the addition of inversion flags, an additional latency is added to both the read access and write access times. During a read, the inversion flag has to be used to invert the data read from the array. During a write access, a calculation must be performed

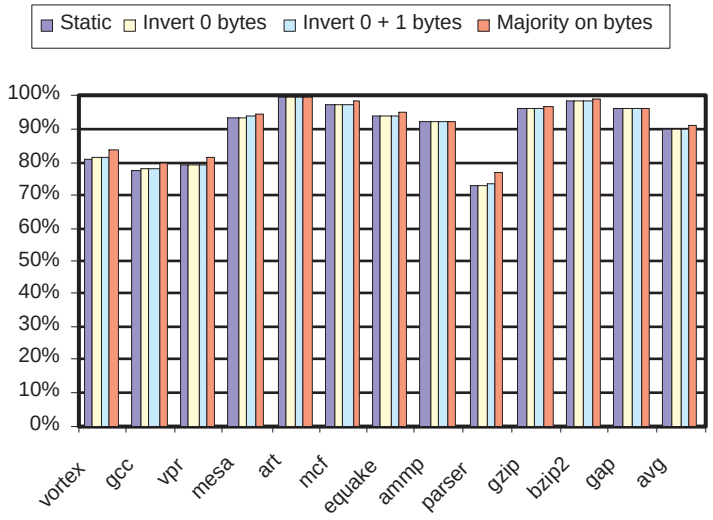


Figure 5.15: Fraction of zero bits in BTB Tags

to determine if the data is to be inverted before it is stored. The following subsections present circuits that can perform these functions.

5.5.1 Majority Counter

During a write, the tally circuit shown in Figure 5.16 can perform the calculation to determine if a byte should be inverted. The circuit is a pass-transistor network that routes a logical '1' to only one of eight outputs, indicating how many '1's were in the byte. For the MajByte inversion policy, the last four rows of the circuit must be used (8 '1's to 5 '1's) and a 4-input OR gate has to be connected to the four outputs to determine if an inversion must occur. For 1-0Byte policy, only the bottom two rows of the circuit are needed since an inversion only occurs if there are seven or eight '1's. In contrast, the Inv0Byte policy involves only the final row. For the MajWord and Inv0Word, a larger but similar circuit can be envisioned, but if anything other than a byte is used, partial writes will have to be converted to a read-modify-write access to maintain appropriate inversion and this adds complexity.

This circuit was simulated and the latencies of this circuit for the MajByte, 1-0Byte

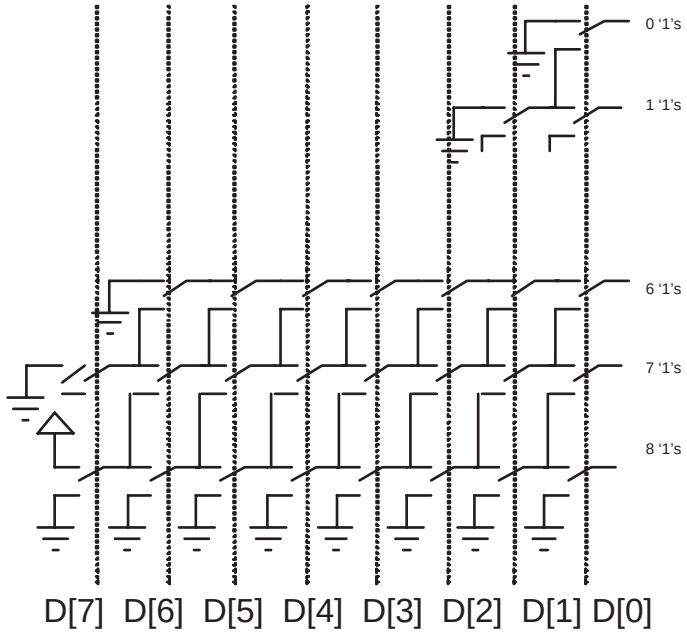


Figure 5.16: Tally Circuit

and Inv0Byte inversion policies are respectively 910ps, 450ps and 200ps. The decrease in latency is a result of reduced capacitance at every node due the pass-transistor network being cut-off.

5.5.2 Read Inversion

During a read, the data must be inverted based on the inversion flag. This inversion can happen at various points in the read, but to reduce the latency of the operation, it was found that the operation is best performed in the next cycle after the data has been latched. The circuit shown in Fig. 5.17 performs this process: the inversion flag is used to invert the Q and Qbar outputs of the registered data read from the SRAM. This circuit adds only 38ps to the cycle time for an 8-bit inversion and 180ps for a 32-bit inversion.

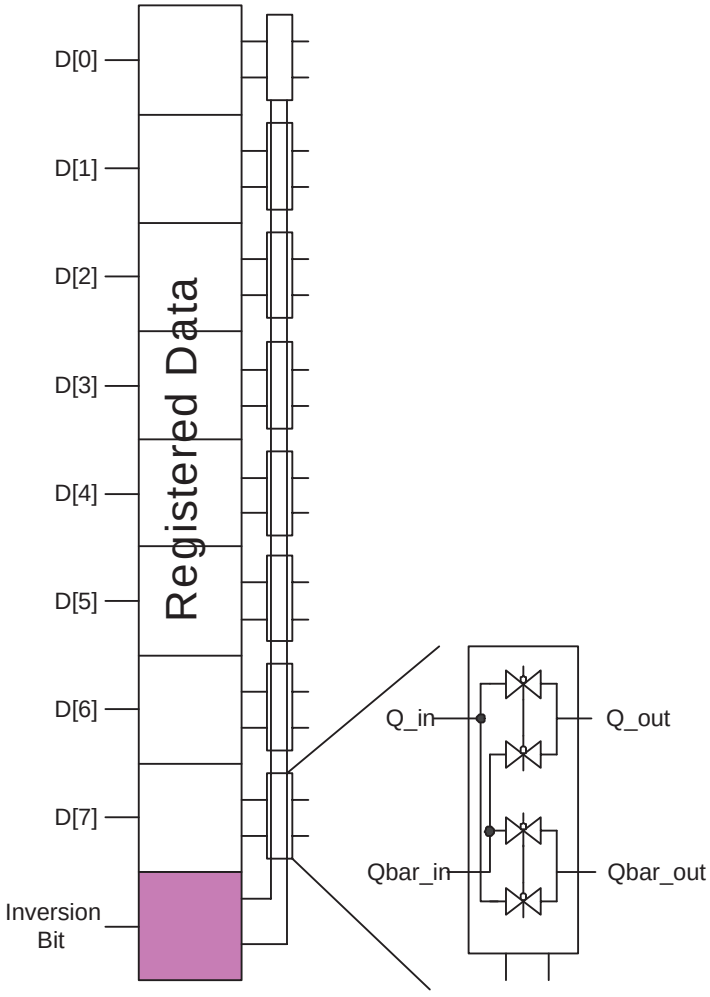


Figure 5.17: Circuit that performs inversion during read

6 Conclusion

> The fruits of the tree of man have ever been and are goodly deeds and a praiseworthy character . . . If they be accepted, your end is attained, and the purpose of life achieved
>
> Baha'u'llah

6.1 Summary and Conclusions

Current technology voltage scaling trends are leading to high levels of leakage power dissipation in microprocessors. Because cache memories account for a large fraction of on-chip transistors, much of the recent research has focused on reducing leakage in caches. Ideally, cache cells should be as fast as possible consuming as little leakage power as possible. Unfortunately, this requirement is increasingly at odds with a fundamental technology trade off: fast transistors tend to dissipate high leakage power. Accordingly, high-V_t cells reduce leakage power drastically, but are unacceptably slow for high-performance deep-submicron caches. Consequently, previous work at the architectural-level has focused on using regular-V_t cell caches and effectively turning off those parts of a cache that do not hold live data.

In this thesis, a novel approach that combines both circuit- and architectural-level techniques is proposed. The approach drastically reduces leakage power dissipation, even when most of the cache is holding live data most of the time. The key observations behind the approach are that:

Table 6.1: Summary results for all the cells.

Cell	Leakage (0)	Leakage (1)	Δ Delay	Δ Stability (SNM)	Δ Stability ($I_{\text{trip}}/I_{\text{read}}$)	Δ Area
RV	100%	100%	0%	0%	0%	0%
LE	1%	14%	2%	7%	-5%	0%
SE	14%	50%	0%	-6%	15%	0%
SLE	14%	43%	2%	23%	-7%	0%
SSE	50%	53%	0%	9%	13%	0%
RLE	2%	14%	1%	22%	5%	2%
RSE	15%	49%	0%	8%	15%	3%

1. Cache-resident memory values of ordinary programs exhibit a strong bias towards zero or one at the bit level.

2. Leakage power dissipation depends on the actual bit value stored in the cell.

A family of high-speed asymmetric dual-V_t SRAM cell designs that exploit this bit-level bias to reduce leakage power, while maintaining high performance was introduced[1].

The results for the best six cells are summarized in Tables 6.1 and 6.2. Here, "Leakage (0)" and "Leakage (1)" refer to the leakage when the cell is storing a '0' and a '1', respectively, "Delay" refers to the total read access time, and "Area" refers to the total cell layout area. The projected long-term leakage of a cache for with 70% '0's and 30% '1's is shown under "Expected Leakage" (accounting only for the cell leakage). In Table 6.2 the column for "Worst Δ Stability" gives the worst case between the two columns of Table 6.1 corresponding to the SNM test and the $I_{\text{trip}}/I_{\text{read}}$ test. If one had to single out *the best* cases, it is perhaps the case that SSE and RLE combine the best features. SSE has less than half the original leakage of the RV cell with no loss of either performance, stability or area. RLE has only 6% of the original leakage of the RV cell with a performance loss of only 1%, no loss of stability, and an area increase of only 2%.

Two cache organizations that used a static bias towards zero, or dynamic, selective

[1]Most technologies currently have dual-V_t or triple-V_t transistors, but the use of these transistors may incur extra cost due to the need for additional lithography masks for each type of transistor.

Table 6.2: Summary results for all the cells, showing expected leakage.

Cell	Expected Leakage	Δ Delay	Worst Δ Stability	Δ Area
RV	100%	0%	0%	0%
LE	5%	2%	-5%	0%
SE	25%	0%	-6%	0%
SLE	23%	2%	-7%	0%
SSE	51%	0%	9%	0%
RLE	6%	1%	5%	2%
RSE	25%	0%	8%	3%

inversion, to maximize the number of cache bits that are zero, are proposed. The latter bounds worst case leakage power, while the reduction possible with either technique depends on application behavior.

The cache resident bit value behavior of SPEC2000 programs was studied and found that there is bias towards zero. Except for dense FORTRAN applications, often more than 70% of data cache bits exhibit a strong bias towards zero. The instruction cache bits are less biased with about 60% being zero. The cache resident bit behavior for two different instruction set architectures and for different caches sizes was studied. This analysis identified why programs tend to exhibit a strong bias towards zero bits. Moreover, it was shown that the techniques can reduce leakage, even if it were possible to perfectly disable all dead cache blocks. Finally, it was shown that the asymmetric cells can drastically reduce leakage power compared to the conventional, regular-V_t caches. It was found that for the given process, using selective inversion does not offer a significant advantage. Nevertheless, selective inversion bounds leakage power under worst case assumptions for the bit values stored in the caches.

6.2 Future Work

While eleven asymmetric cells were presented in total, further possibilities exist. Also, asymmetric SRAMs could be useful in other SRAM-based structures in high-performance processors. Moreover, while threshold voltage was used to selectively weaken some of

the cells transistors, other alternatives may exist (e.g., resizing).

As technology continues to scale down, the gate leakage within SRAM cells will compose an increasingly larger percentage of total power dissipation. Asymmetrical techniques may be applied to the oxide thickness to limit gate leakage, and at the same time maintain performance.

References

[AH98] B. S. Amrutur and M. A. Horowitz. A replica technique for wordline and sense control in low-power SRAM's. *IEEE Journal of Solid-State Circuits*, 33:1208–1219, August 1998.

[ALR02] Amit Agarwal, Hai Li, and Kaushik Roy. DRG-Cache: A data retention gated-ground cache for low power. *Proceedings of the 39th Design Automation Conference*, June 2002.

[AN97] E. Ajith Amerasekera and Farid N. Najm. *Failure Mechanisms in Semiconductor Devices*. John Wiles and Sons, 2 edition, 1997.

[BB00] Thomas D. Burd and Robert W. Brodersen. Design issues for dynamic voltage scaling. *Proceedings of the 1998 International Symposium on Low Power Electronics and Design*, pages 9–14, 2000.

[BM99] D. Brooks and M. Martonosi. Dynamically exploiting narrow width operands to improve processor power and performance. *Proceedings of the 5th International Symposium on High-Performance Computer Architecture*, January 1999.

[Bor99] S. Borkar. Design challenges of technology scaling. *IEEE MICRO*, 19(4):23–29, July–August 1999.

[BTM01] Azeez J. Bhavnagarwala, Xinghai Tang, and James D. Meindl. The impact of intrinsic device fluctuations on CMOS SRAM cell stability. *IEEE Journal of Solid-State Circuits*, 36:658–665, April 2001.

[CGS00] R. Canal, A. Gonzalez, and J. E. Smith. Very low power pipelines using significance compression. *Proceedings of the 33rd Annual IEEE/ACM International Symposium on Microarchitecture*, December 2000.

[CJWR98] Zhanping Chen, Mark Johnson, Liqiong Wei, and Kaushik Roy. Estimation of standby leakage power in cmos circuits considering accurate modeling of transistor stacks. *Proceedings of the 1998 International Symposium on Low Power Electronics and Design*, pages 239–244, 1998.

[GAT02] Ed Grochowski, Dave Ayers, and Vivek Tiwari. Microarchitectural simulation and control of di/dt-induced power supply voltage variation. *Proceedings of the Eighth International Symposium on High-Performance Computer Architecture*, 2002.

[Har00] Tegze P. Haraszti. *CMOS Memory Circuits*. Kluwer Academic Publishers, 2000.

[HKM+90] Toshihiko Hirose, Hirotada Kuriyama, Shuji Murakami, Kojiro Yuzuriha, Takeao Mukai, Kazuhiro Tsutsumi, Yasumasa Nishimura, Yoshio Kohno, and Kenji Amani. A 20ns 4MB CMOS SRAM with hierarchical word decoding architecture. *IEEE International Solid-State Circuits Conferance., Digital Technical Papers*, pages 132–133, 1990.

[HP96] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufman, 2 edition, 1996.

[HYK+00] Fatih Hamzaoglu, Yibin Ye, Ali Keshavarzi, Kevin Zhang, Siva Narendra, Shekhar Borkar, Mircea Stan, and Vivek De. Dual V_t-SRAM cells with full-swing single-ended bit line sensing for high-performance on-chip cache in 0.13um technology generation. *Proceedings of the 2000 International Symposium on Low Power Electronics and Design*, July 2000.

[ISN95] K. Itoh, K. Sasaki, and Y. Nakagome. Trends in low-power RAM circuit technologies. *Proceedings of the IEEE*, 83(4), April 1995.

[KG97] Milind B. Kamble and Kanad Ghose. Energy-efficiency of VLSI caches: A comparative study. *10th International Conference on VLSI Design*, pages 261–267, 1997.

[KHM01] S. Kaxiras, Z. Hu, and M. Martonosi. Cache decay exploiting generational behavior to reduce leakage power. *Proceedings of the 27th International Symposium on Computer Architecture*, July 2001.

[KRK+00] Timothy Kam, Shishpal Rawat, Desmond Kirkpatrick, Rabindra Roy, Gregory S. Spirakis, Naveed Sherwani, and Craig Peterson. EDA challenges facing future microprocessor design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(12), December 2000.

[Lip97] M. Lipasti. *Value Prediction and Speculative Execution*. PhD thesis, Carnegie-Mellon University, April 1997.

[MR00] J. M. Musicer and J. Rabaey. MOS current mode logic for low power, low noise CORDIC computation in mixed-signal environments. *Proceedings of the 2000 International Symposium on Low Power Electronics and Design*, July 2000.

[PYF+00] M. D. Powell, S. H. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar. Gated-V_{DD}: A circuit technique to reduce leakage in cache memories. *Proceedings of the 2000 International Symposium on Low Power Electronics and Design*, July 2000.

[SLL87] E. Seevinck Sr., F. J. List, and J. Lohstroh. Static-noise margin analysis of MOS SRAM cells. *IEEE Journal of Solid-State Circuits*, 22:748–754, October 1987.

[VZA00] L. Villa, M. Zhang, and K. Asanovic. Dynamic zero compression for cache energy reduction. *Proceedings of the 33rd International Symposium on Microarchitecture*, December 2000.

[YPF+01] S. H. Yang, M. D. Powell, B. Falsafi, K. Roy, and T. N. Vijaykumar. An integrated circuit/architecture approach to reducing leakage in deep-submicron high-performance I-caches. *Proceedings of the 7th International Symposium on High-Performance Computer Architecture*, January 2001.

[ZTRC01] H. Zhou, M. C. Toburen, E. Rotenberg, and T. M. Conte. Adaptive mode control: A static-power-efficient cache design. *Proceedings of the 2001 International Conference on Parallel Architectures and Compilation Techniques*, September 2001.