VLSI Circuits and Architecture Techniques for Energy Efficient and Low V_{MIN} SRAMs

---

A Dissertation by

Arijit Banerjee

December 2017

Presented to the Faculty of the

School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment of the requirements for the dissertation and the subsequent Degree

Doctor of Philosophy in Electrical Engineering

Approved by

Dr. Mircea R. Stan

Dr. Benton H. Calhoun

Dr. John Lach

Dr. N. Scott Barker

Dr. Scott T. Acton

Dr. John A. Stankovic

© 2017 Arijit Banerjee

Abstract

The rapid growth of portable mobile devices such as smartphones, smartwatches, wearable health monitors, etc. suggests that the total number of Internet of Things (IoT) devices may reach 50 billion by the year 2020. Depending on the application such as mobile health monitoring, artificial organs, vision for visually impaired people, etc., portability and form factors of such IoT devices restrict the use of energy sources to smaller batteries. Additionally, these mobile devices could run on harvested energy from ambient light, body heat, etc. energy sources, which makes them highly energy-constrained. As energy depends quadratically on the supply voltage, dynamic voltage and frequency scaling (DVFS) reduce energy consumption drastically and boost performance from time to time when needed. Using DVFS, battery-operated IoT devices could duty-cycle the on-chip resources and save energy, to result in longer battery-life. To satisfy this requirement both logic and memory in the IoT system on chips (SoC) must be flexible to operate at such reduced supply voltages. With technology scaling, the logic's minimum operating voltage (V_{MIN}) scales easily with supply voltage. However, process variation increases with technology scaling, which poses a threat to lowering the V_{MIN} of widely used static random access memory (SRAM). Thus, there is a bottleneck for DVFS in area-efficient low-cost SoCs where SRAM and the processor core share the same power rail. Moreover, SRAMs could consume 20%-60% power in IoT SoCs, which requires V_{MIN} lowering techniques to drastically reduce the SRAM power consumption, such as using alternative bitcell topologies, peripheral assist circuits, etc. Furthermore, these solutions could use methodologies, such as design-for-the-worst-case to add voltage and timing guard-bands

to cope with the process variation. On the other hand, these techniques have energy and area overheads and require careful design based on low-power specification. Moreover, with technology scaling the high-density (HD) 6T SRAM suffers from write and read issues in FinFET processes at the nominal supply voltages for the worst-case corner. Thus, HD 6T FinFET SRAMs require additional voltage guard-bands to the V_{MIN} for safer operations. SRAM V_{MIN} is also hard to track because it varies with process, frequency, temperature, etc. parameter fluctuations. Canary SRAMs have been shown to monitor SRAM data retention voltage (DRV) V_{MIN}, which could help track the SRAM dynamic V_{MIN} for energy savings. Therefore, exploring these design knobs related to the SRAM V_{MIN} lowering could reveal important tradeoffs from energy efficiency, delay, and area standpoints for IoT applications.

One of the goals of this dissertation is to investigate the circuit and architecture methods to design energy efficient low V_{MIN} SRAMs by tweaking these design knobs. These knobs such as bitcell typologies, peripheral assist techniques, and architectures could push the boundary of the SRAM design space. In the first chapter, we discuss the state of the art trends and challenges in energy efficient and low-V_{MIN} SRAMs for ultra-low-power (ULP) IoT applications and describe the major contributions of this thesis and its organization. The second chapter investigates the scope for improvements in the existing bitcell topologies, array architecture knobs, and a peripheral architecture using a read-modify-write scheme to improve the SRAM read energy efficiency by 5.7X for the ULP sub-threshold applications. In the third chapter, we study the single and dual combinations of peripheral write and read assist techniques for lowering SRAM V_{MIN} in 14nm 6T HD FinFET SRAMs. The fourth chapter introduces the concept of SRAM dynamic write V_{MIN} tracking using canary SRAM and investigates the relationship between the input design knobs of canary and core SRAM and their output metrics. It further documents the tradeoffs of a reverse assist (RA) for canaries to track SRAM V_{MIN}, which could result in a 50% energy savings in 28nm technology. In chapter five, we further show the first proof of concept of RA-based canaries in silicon and examine how sensitive it is to the voltage, frequency, and temperature variation for a V_{MIN}

tracking application. In the sixth chapter, we investigate the classification of reverse assists using pulse-shaping techniques for wordline and bitline type RAs in pursuit of V_{MIN} tracking. We further compare their sensitivity properties across canary design knobs and investigate the energy and area tradeoffs of RA circuits. The seventh chapter proposes an architecture, which leverages combined peripheral assists with an in-situ canary-based self-tuning scheme to give 0.38V-1.2V wide-range SRAM. The SRAM architecture achieves a maximum of 1444X active power and 12X leakage savings compared to the nominal supply voltage. Chapter eight proposes a mathematical framework and a set of algorithms to analyze and design RA-based canaries. It further automates and reduces the burden of the analysis and design of RAs across canary design knobs from months to days. Finally, we conclude this thesis in chapter nine, by documenting the summary of contributions, open questions, and discussing the impact of these works.

Approval Sheet

This dissertation is submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Electrical Engineering)

Arijit Banerjee

Arijit Banerjee

This dissertation has been read and approved by the Examining Committee:

Mircea R. Stan

Dr. Mircea R. Stan, Committee Chair

Benton H. Calhoun

Dr. Benton H. Calhoun, Advisor

John Lach

Dr. John Lach

N. Scott Barker

Dr. N. Scott Barker

Scott T. Acton

Dr. Scott T. Acton

John A. Stankovic

Dr. John A. Stankovic

Accepted for the School of Engineering and Applied Science:

Craig H. Benson

Craig H. Benson, Dean, School of Engineering and Applied Science

December 2017

Arise, awake, and stop not till the goal is reached.
- sloka from 1.3.14 chapter of Katha Upanishad
translated by Swami Vivekananda

Acknowledgements

True it is that I have garnered all my knowledge and wisdom from this world, and I am grateful to all of the people with whom I interacted during my learning. The journey to my PhD at the University of Virginia campus is not an exception, which is tinged with such learning from professors, peers, juniors and non-technical officials. Over the course of these five years, I have gathered indelible memories of amazing friends and colleagues from diverse cultures and truly enjoyed working with them, which I will keep in my heart forever and I am thankful to them for where I stand today. However, without the unconditional love, support, encouragement, and sacrifice of my family, I would not have made it this far.

I would first thank my wife, Natasha, and son, Arunabha for being the very best blessings of my life and bearing with me during my doctoral studies. Thank you for your love, support, and sacrifice. I am eternally indebted and grateful to my parents Mrs. Gopa Banerjee and Mr. Bibek Banerjee, and my brother, Indrajit, for their love, sacrifice, encouragement, and support in my life. This thesis is dedicated to all of my family members.

I am truly thankful to my adviser, and one of the best mentors in my life, Professor Benton Calhoun, who gave me the opportunity to come to the University of Virginia and do world-class research in semiconductor memories. I am deeply inspired by his way of teaching and asking research questions that motivated me to think about the narrow and broad level views of research problems and their solutions. Throughout the course of graduate school, Ben's teachings in thinking, planning, and execution of work and research helped me keep myself focused in my research and maintain a higher standard and made me a better

researcher. Thank you for inspiring, motivating, and making me a better person.

I would like to thank Professor Mircea Stan for his deep insight into digital design and teachings in VLSI that inspired me to delve into the bigger picture of the digital VLSI and also into specific sub-components. I truly enjoyed learning from the advanced courses he taught that greatly helped me to understand the background and theory of modern VLSI. Thank you for all light of knowledge you shared with me by your teaching.

I am thankful to Professor John Lach for sharing his knowledge of categorizing engineering problems to identify and understand the requirements for existing or novel solutions. His teaching inspired me to think on a higher level for labeling known and unknown problems and their solutions, before even trying to solve them. Thank you for your teaching of thinking broadly about engineering problems and their solutions.

I would like to thank my other committee members: Professor John Stankovic, Professor Scott Acton, and Professor Scott Barker, for their valuable time, advice, feedback, and thoughts to educate me in the process of thinking from a different angle in my PhD studies and putting up with me. Thank you for everything.

I am very thankful to have worked with Dr. Yousef Shakhsheer, Dr. James Boley, Peter Beshay, Dr. Farah Yahya, Dr. Harsh Patel, Ningxi Liu, Dr. Abhishek Roy, and Dr. Dilip Vasudevan of Professor Ben Calhoun's Robust Low-power VLSI group and Sergiu Mosanu, Tommy Tracy of Professor Mircea Stan's group. Our discussions emanated many ideas presented in this thesis and beyond, and I am grateful for their sharing the knowledge and wisdom to allow my learning of it.

I am truly thankful to my friends: Peter Beshay, Dr. Divya Akella Kamakshi, Dr. Dilip Vasudevan, Patricia Gonzalez, Dr. Harsh Patel, Ningxi Liu, Hang Zhang, Xinfei Guo, and Sergiu Mosanu, who inspired me to think differently and supported me in my hardships during my graduate study.

I would like to thank Nvidia and my manager, Dr. Tom Gray, for the opportunity to intern during the summers of 2013 and 2014 as a researcher. I am grateful to have had

Mahmut Sinangil as my mentor while doing the internship and I gained a great deal of knowledge and insight working with him. I am also thankful to John Poulton and Bill Dally for giving me insightful feedback on my research that helped me think differently and figure out the correct directions during my internships.

I am greatly thankful to Terry Tigner who helped me in processing orders, booking conference rooms, booking flights, processing and submitting reimbursement requests, and many other things that would have been impossible without her.

I would thank once again Ben, Dilip, Divya, Harsh, Ningxi for helping me edit many of my papers, and I would like to thank Nancy for helping me proofread the thesis in a short notice. Any of the remaining typos are solely my own.

Last but not least I would like to thank Subramanian Senthivinayagam and Prathiba Jayaguru and Patrick Flynn and his family for helping my family and me with many needs.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.0.1 Motivation for SRAMs

The Internet of Things (IoT) revolutionizes personal lifestyles by connecting us to a new world of smart appliances, home automation controllers, wearable health monitors, etc. The total number of these connected IoT devices may be expected to reach 50 billion by the year 2020 [1]. Due to the requirements of portability and smaller form factor, IoT applications such as mobile health monitoring, artificial organs, vision for blind people, augmented reality goggles, selfie drones, etc. restrict the use of energy sources to smaller batteries. Additionally, these devices could harvest energy from ambient light, body heat, etc. energy sources. Table 1.1 shows typical IoT applications, their ballpark power requirements, and corresponding projected battery-life. As batteries, both non-rechargeable and rechargeable have limited energy density and battery-life (Figure 1.2), battery-replacement of millions of IoT devices could incur millions of dollars in replacement and maintenance costs annually.

On the other hand, energy harvesting transducers could scavenge electrical energy for IoT applications from light, vibration, radio waves, etc. sources that overcome the battery replacement issues. However, the available harvested power for various energy sources are limited (Table 1.2) for IoT systems those have strict constraints of portability, light-weightiness, and smaller form factor. Moreover, availability of the energy sources could be

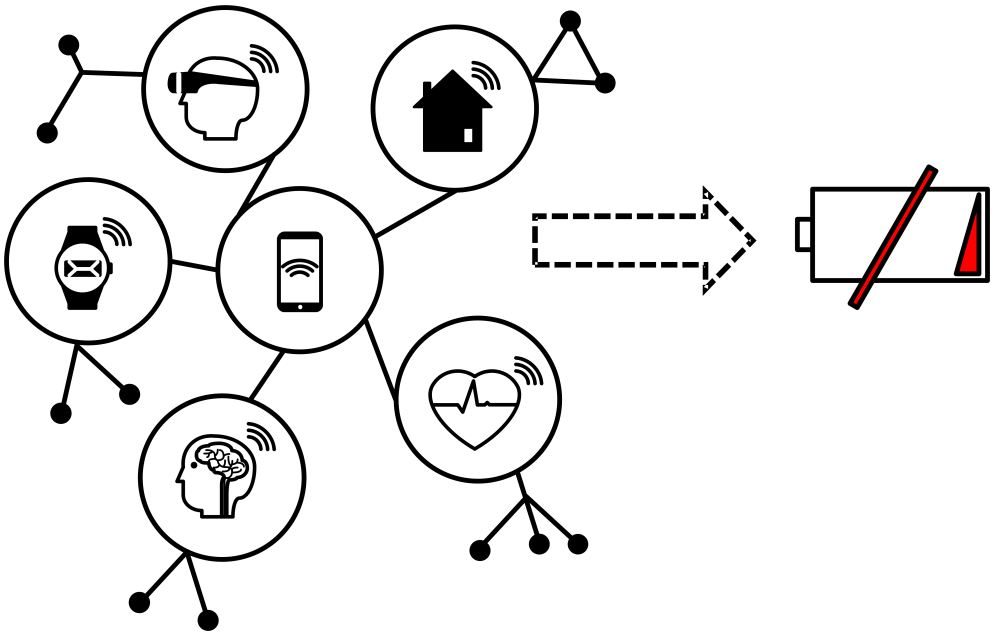


Figure 1.1: Battery-life issue in portable IoT applications.

Table 1.1: Typical IoT systems and their power consumption and battery-life [2].

Application device	**Power consumption**	**Battery-life**
Smartphone	$1W$	5h
MP3 player	$50mW$	15h
Hearing aid	$1mW$	5 days
Wireless sensor node	$100\mu W$	Lifetime
Cardiac pacemaker	$50\mu W$	7 years
Quartz watch	$5\mu W$	5 years

interrupted such as light, vibration, etc. energy cannot be guaranteed for a prolonged time. Thus, batteries having higher energy density are still the choice of IoT devices for running the vast majority of applications, compared to the energy harvesting transducers.

Portable batteries come in two types such as non-rechargeable and rechargeable coin cells. Non-rechargeable coin or button cell batteries are classified mainly into alkaline (1.5V), Silver Oxide (1.5V), Lithium (3V) [3], and Mercury (1.35V) cells. Among these portable batteries, Mercury-based cells are banned in most countries, as they are environmentally hazardous. On the other hand, rechargeable batteries are mainly classified into Nickel Cadmium (1.25V),

Table 1.2: Various energy harvesting sources, and their available power [2].

Energy source		**Available power** / cm^2	**Harvested power** / cm^2
Light	Indoor	$0.1mW$	$10uW$
	Outdoor	$100mW$	$10mW$
Thermoelectric	Human	$0.5m$ @ $1Hz$ $1m/s^2$ @ $50Hz$	$4\mu W$
	Industrial	$1m$ @ $5Hz$ $10m/s^2$ @ $1KHz$	$100\mu W$
Vibration/Motion	Human	$20mW$	$30\mu W$
	Industrial	$100mW$	$1 - 10mW$
RF	Cell phones	$0.3\mu W$	$0.1\mu W$

Nickel-Metal Hydride (1.2V), Lithium-ion (3.6V), and Lithium-ion Polymer (3.6V) [4] cells. These batteries have internal resistances starting from a few hundred $m\Omega$ to several hundred $m\Omega$. All of these batteries self-discharge and their terminal potential decreases with time. Due to internal chemistry of batteries, the maximum shelf-life or end-of-life of batteries is around 10 years. Table 1.3 shows the estimated self-discharge of various battery types. Due to smaller form-factor, the button or coin cell non-rechargeable batteries have several mAh to several hundred mAh capacities and require battery replacement based on load current or average power consumption. The rechargeable ones, on the other hand, can sustain 300-500 recharge cycles, usually, before the maximum battery capacity permanently drops to 80% of the initial capacity. Figure 1.2 shows the battery replacement time assuming self-discharge (Table 1.3) of various battery types, such as the non-rechargeable SR4165W (8mAh, 1.5V), rechargeable LIR2032 (40mAh, 3.6V), and the Apple iWatch equivalent battery A1578 (200mAh, 3.8V). Thus, battery replacement time or the cycle-life is a major design knob for the energy-limited IoT systems.

Such energy-constrained IoT devices may need to run for prolonged periods of time in various workload conditions and so demand longer battery-life. Dynamic voltage and frequency scaling (DVFS) could achieve such desired performance, as well as energy efficiency from time to time, which is a widely used technique in the modern system on chips (SoCs). However, operating at lower supply depends on the minimum operating voltage (V_{MIN}) of both logic and embedded memories.

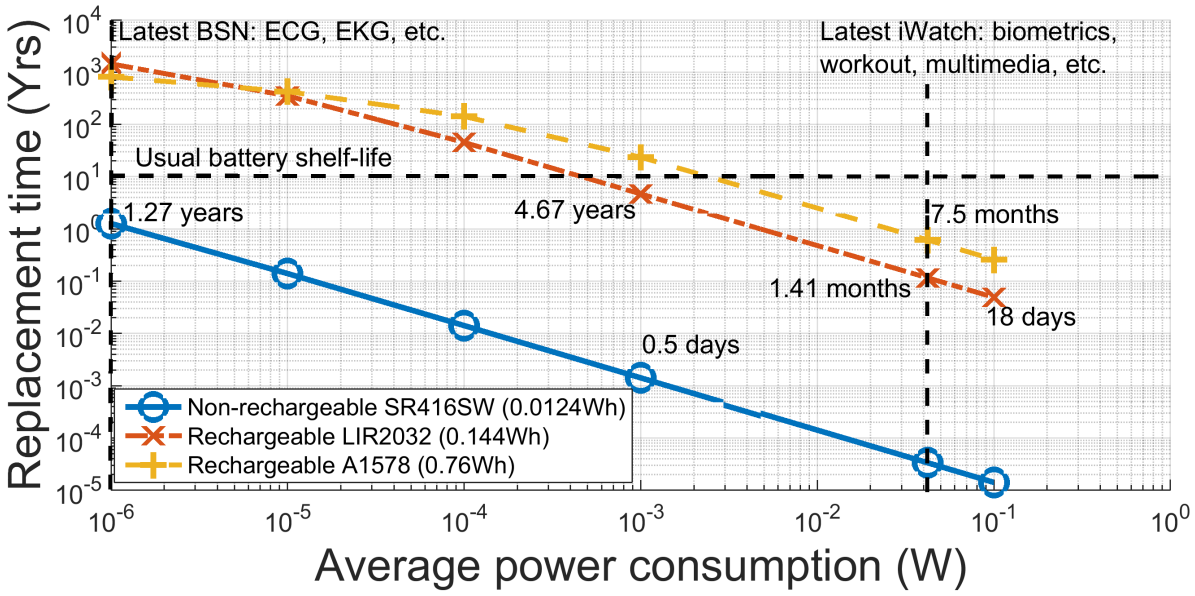


Figure 1.2: Estimated battery replacement time of non-rechargeable SR416SW (0.0124Wh), rechargeable LIR2032 (0.144Wh), and A1578 (0.76Wh) assuming worst-case 300 charge-discharge cycles across various power consumption numbers.

Table 1.3: Estimated self-discharge of various battery types [5], [6].

Battery type	**Estimated self-discharge**
Primary lithium-metal	10% in 5 years
Alkaline	2-3% per year (7-10 years shelf life)
Lead-acid	5% per month
Nickel-based	10-15% in 24h, then 10-15% per month
Lithium-ion	5% in 24h, then 1-2% per month (plus 3% for safety circuit)
Lithium-ion polymer	10% per month
Silver oxide	0.7% per month

Although logic usually has a lower V_{MIN}, not all types of memories are fit to work at lower supply voltages and frequencies, such as FLASH and dynamic random access memory (DRAM), respectively. Moreover, emerging memories such as spin torque transfer (STT) RAM, resistive RAM (RRAM), etc. are promising, but have severe limitations to operable at lower supply voltages. On the other hand, SRAMs are demonstrated to work in very low supply voltage with ultra-low energy [7] and in standard supply with higher performance [8], which makes them more flexible from the supply voltage, operating frequency, and energy standpoint, compared to other memories. SRAMs were predicted to outnumber the logic

counterpart in silicon area in the past [9]. This trend can be observed in recent SoC desktop and laptop processors [9] [10] (Figure 1.3 and 1.4) and IoT SoCs (Figure 1.5).

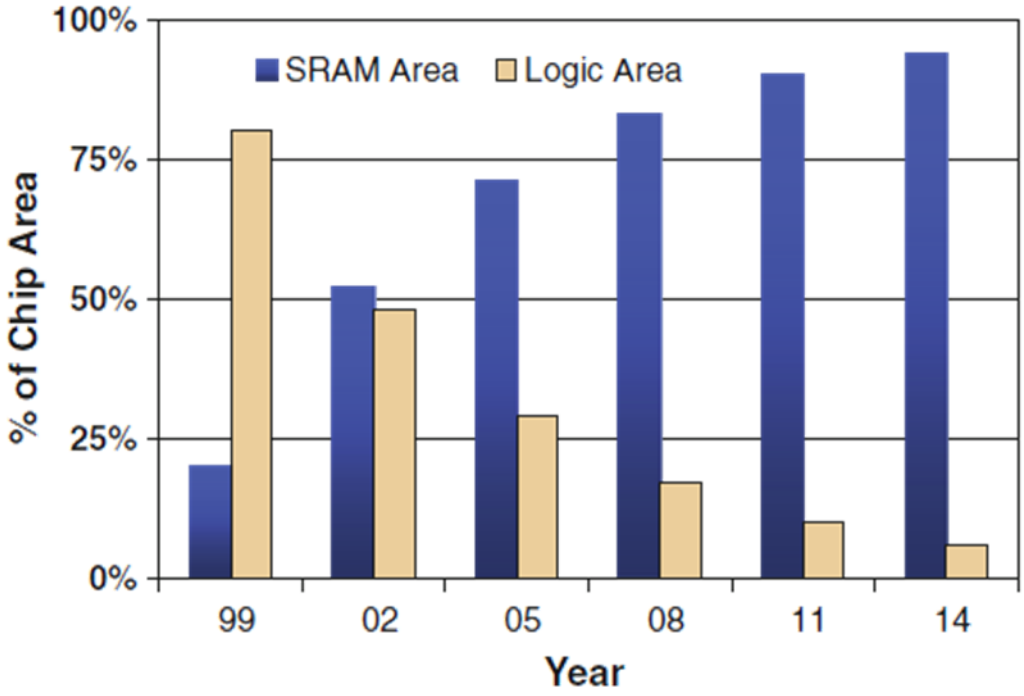


Figure 1.3: SRAM area trend in SoCs [9] (© 2013 Springer).

The flexibility to operate SRAMs in wider voltage range, frequency, and energy efficiency makes them a choice of general purpose embedded memory in cache memories, scratchpads, buffers, and register files for various applications. However, SRAMs can have significant amounts of power consumption in modern energy constraint SoCs [11] [12] (Figure 1.5 and 1.6). Applying DVFS could lower such energy consumption in SRAMs. However, with device scaling in deep sub-micron technologies, the process variation increases, which limits the SRAM minimum operating supply voltage (V_{MIN}), and creates a bottleneck for DVFS to lower supply voltages. This bottleneck makes the SRAM design for energy constrained IoT applications challenging. This chapter introduces the basic components and design metrics of SRAM and deals with the energy and leakage-limited low V_{MIN} SRAM design challenges. We also discuss the major contributions of this thesis and its organization in this chapter.

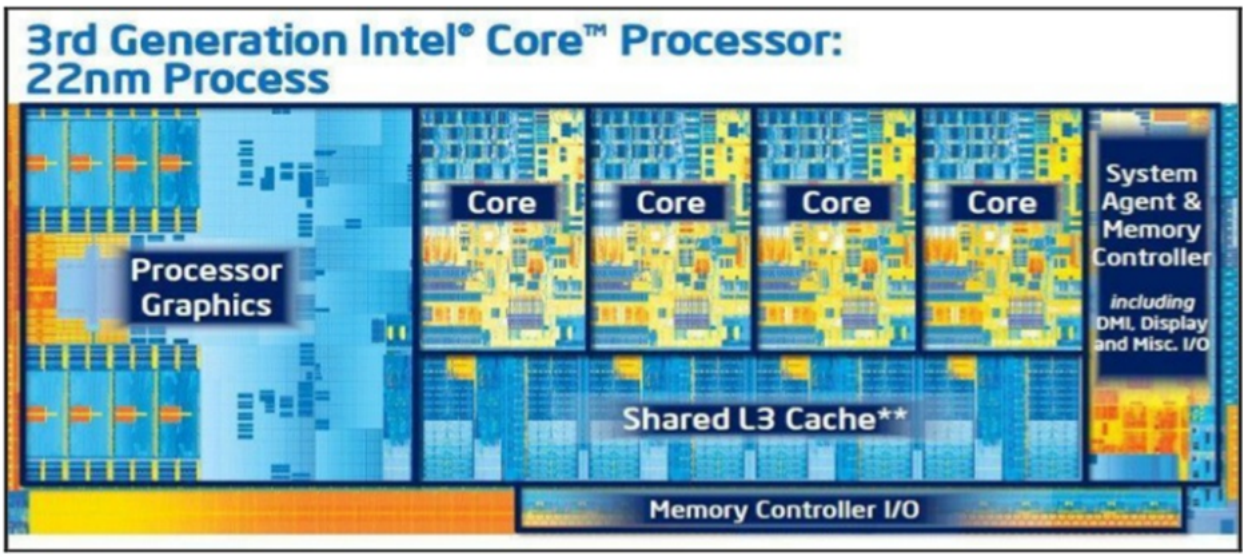


Figure 1.4: State-of-the-art Intel microprocessor with up to 50% of SRAM area [10] (image © Intel).

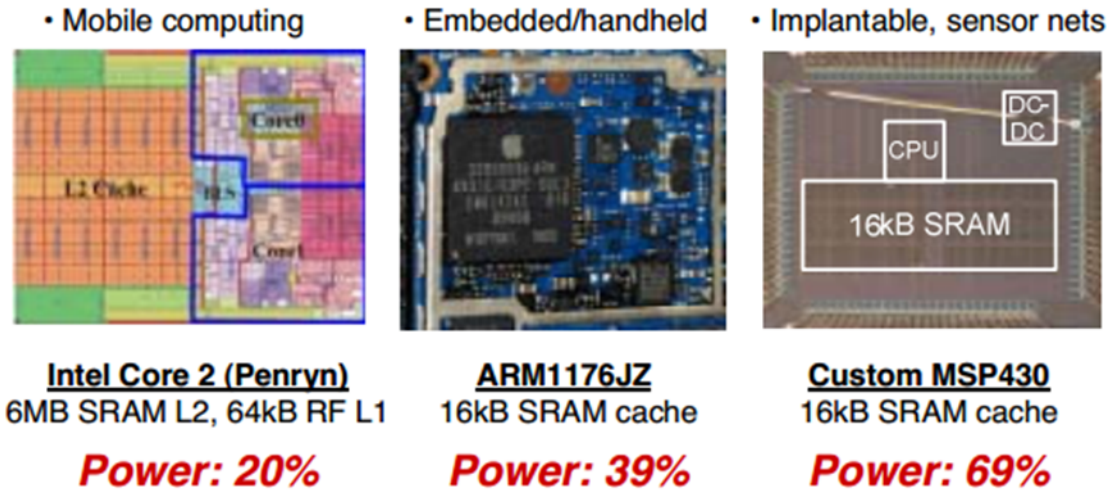


Figure 1.5: SRAM power consumption in state-of-the-art SoCs [11] (© 2009 Massachusetts Institute of Technology).

1.0.2 SRAM Architecture Overview

Figure 1.7 shows the block diagram of the basic SRAM architecture with its sub-components. SRAM, as a black box, can be thought as a block with input address (ADDR)

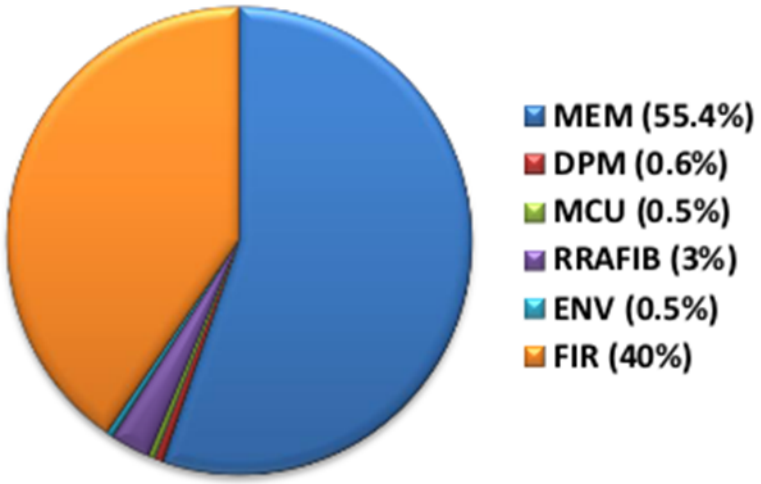


Figure 1.6: Power consumption of SRAMs in the BSN chip [12] (© 2013 Yanqing Zhang).

bus; data input (DIN) bus; clock, reset, enable, etc. (Control) signals as other inputs; and data output (DOUT) bus as output. SRAMs can have two logical operations such as write and read. In a write operation, one has to select an SRAM address through the ADDR bus and put data into the DIN bus to write the data into the corresponding address location. On the other hand, in a read operation, a particular address location can be selected to get the content of the address from the SRAM into the DOUT bus. Conventional SRAM is comprised of two logical sub-blocks such as core array and periphery. The core array consists of single or multiple core banks, and each of the banks contains row-column-based compactly tiled SRAM bitcells, which are the unit of storage in SRAMs. An SRAM bitcell can store a logical '1' or '0' data into it. Wordlines (WL) and bitlines (BL) are the only input and output control signals for the write and read operations to perform over an SRAM bitcell and a core bank. On the other hand, the SRAM peripheral circuits can be divided into row and column peripherals and control logic. The SRAM row periphery is comprised of a row address decoder. On the other hand, the column periphery usually consists of precharge logic, column muxes (optional), and read and write circuitry. The row address decoder decodes the row address bits and triggers appropriate WL in read and write operations, which selects bitcells in a row in an SRAM bank. The column decoder decodes the column multiplexers to select the corresponding word from the interleaved bitlines in a row. State-of-the-art circuit

design techniques [13] [14] [15] [16] [17] [18] are widely used in SRAM decoder design. The write and read circuitry in SRAM periphery are responsible for write and read operations correspondingly. And finally, the control logic generates all of the internal timing control signals to perform the SRAM write and read operations.

1.1 SRAM Design Metrics

The key SRAM design metrics are write-ability, readability, read-stability, hold-stability and data retention capability. Also, metrics such as performance, power, leakage, and area can be design constraints, which make the SRAM design even more challenging. An SRAM's minimum operating voltage depends on these design metrics mentioned above, which govern the SRAMs capability to do error-free operations with some confidence, such as write, read, and holding the data written until the supply is turned off. The detailed descriptions of the design metrics are given as follows in the context of the conventional 6T SRAM bitcell.

1.1.1 Write-ability of SRAMs

SRAM write-ability defines the ability to write data error-free into the SRAM. An SRAM write operation waveforms are shown in the context of conventional 6T SRAM bitcell 1.8 in Figure 1.9. In a write operation, the address (ADDR) and input data (DIN) are first applied with a setup-time before the clock comes. Within the clock cycle for a write operation, the row address decoder decodes the input address, and a finite wordline (WL) pulse triggers selecting a row of 6T bitcells, accordingly. Meanwhile, the write circuitry in SRAM periphery pulls down a set of bitline (BL) or bitline-bar (BLB) signals of a word selected in a bitcell-row using the column decoder. The unselected words in the same row and the corresponding BL or BLB signals are kept floating, those precharges to V_{DD} in the previous cycle. Pulling down one of the BL or BLB in an SRAM bank results in pulling down one of the internal nodes Q or Qb of the bitcell, while the WL is turned on. After some time, the internal nodes of the

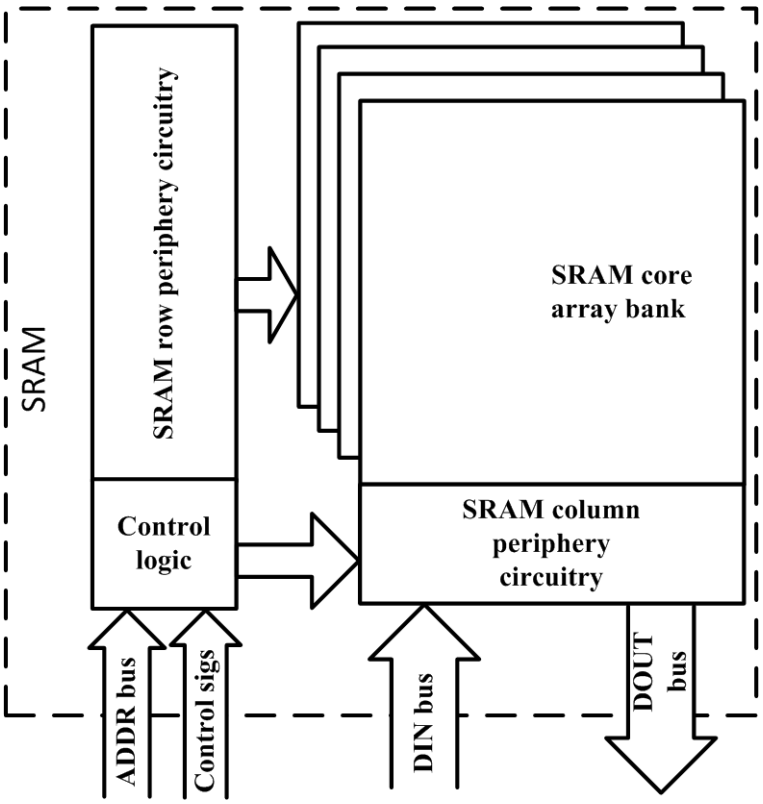


Figure 1.7: SRAM Architecture.

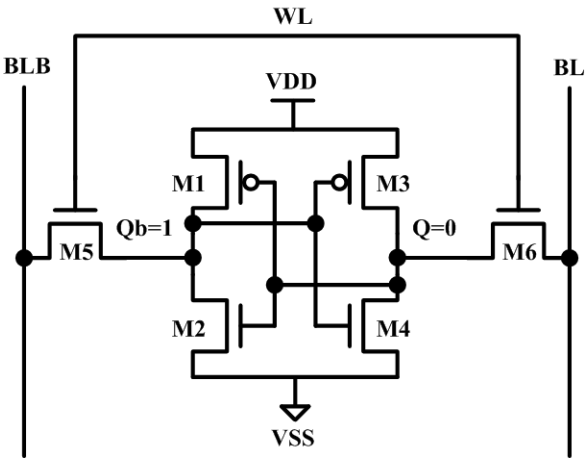


Figure 1.8: Schematic of the conventional 6T SRAM bitcell.

bitcell flip, if the previous data present in the bitcell were opposite compared to the data written. Once the write operations complete on bitcells, the BL and BLB signals precharge again to V_{DD} using a bitline conditioning circuit and prepare for a read or write operation for the next clock cycle. Thus, an SRAM writes the input data in the DIN bus into the

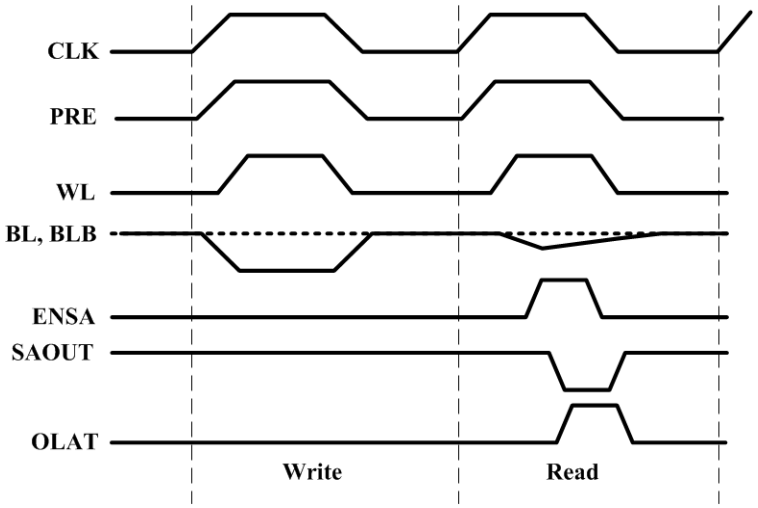


Figure 1.9: Write and read waveforms of a conventional SRAM with 6T SRAM bitcell.

corresponding address location. The precharge operation using 6T SRAM is essential for the read cycle. The SRAM write-ability is usually measured using a DC static metric called write margin (WM) which assumes an infinite wordline (WL) pulse. On the other hand, one of the dynamic metrics for write-ability is called dynamic write-failure probability for a given finite WL pulse-width. With voltage scaling, the SRAM write-ability reduces and becomes more prone to write failures. Moreover, device scaling in deep sub-micron technologies increases process variation, which worsens SRAM write-ability, too.

1.1.2 Readability of SRAMs

SRAM readability defines the ability of the SRAM to read out the data corresponding to the address provided. The conventional 6T SRAM read operation is shown in Figure 1.9. During the read operation, the address is applied with a setup-time using the address bus, before the clock comes. Within the clock cycle for a read operation, the address decoder decodes, and a finite wordline (WL) pulse triggers selecting a row of 6T bitcells, accordingly. Initially, the BL and BLB signals of the bitcell columns precharge to V_{DD}. After some time, a differential voltage (V_{Diff}) gets developed in between BL and BLB signals of the bitcell and a sense enable (ENSA) pulse signal triggers enabling a sense amplifier (SA). The SA takes

some time to evaluate the V_{Diff}, which is known as the SA reaction time (T_{SA}). After that, the SA reads output as logic '1' or '0.' Usually, the output of the sense amplifier connects to the output latch, which holds the read data after the read operation until the next read happens. The SA has a statistical V_{Diff} offset due to process variation below which it cannot correctly evaluate the input differential. Therefore, the ENSA signal awaits to develop enough differential between BL and BLB signals. Once the SA completes evaluating the BL-BLB V_{Diff}, the precharge operation triggers, which pulls up the BL and BLB to V_{DD} to get ready for the next write or read operation. The precharge operation is essential to the SA-based read in 6T bitcells, as it ensures that the V_{Diff} differential development in the current cycle does not affect the next read cycle. In a read operation, the access time (T_{Acc}) defines the time between rising edge of the SRAM input clock and when the data is available at the data output ports (DOUT). The V_{Diff} bitline differential development time ($T_{V_{Diff}}$) or the access time T_{Acc} measure readability in 6T SRAMs. With V_{DD} scaling the bitline differential development time $T_{V_{Diff}}$, the SA reaction time T_{SA}, and the recovery time of the output latch increase. These increases lead to the increase in overall T_{Acc}. Moreover, process variation increases the required $T_{V_{Diff}}$ development time limiting the readability, also.

1.1.3 Read-stability of SRAMs

The 6T bitcell is ratioed logic, and it shares the same electrical path for write and read operations, which makes it vulnerable to get written while reading. This phenomenon is called read-upset or read-disturb, which makes it challenging for SRAM designers to design the 6T bitcell with adequate margin for read-stability so that the contents of the bitcells are unharmed while reading. In the column multiplexing (mux) scenario, bitcells in multiple words interleave in the same row, so that the probability of single-word-multiple-bit-upset lowers [19] due to soft error rate (SER) emanating from high energy particle strikes. Here, the column mux ratio is the number of words interleaved in a row. In this scenario, selecting a word for write or read in the 6T SRAM will cause the row-wise selected other bitcells in

the same row to be half-selected. These half-selected bitcells undergo read-stress and become vulnerable to read-disturb. The read-stability has a static metric called read static noise margin (RSNM). The RSNM measures using the sides of the least fitted square in the SRAM butterfly curve using a DC read simulation [20]. However, RSNM assumes an infinite WL pulse-width, which is an overestimate of the read-stability. One of the dynamic metric for read-stability is the probability of read-disturb using a finite WL pulse, which is a more realistic measure of the SRAM read-stability in a transient read operation. V_{DD} scaling and process variation both reduce the RSNM, which makes it harder to design 6T SRAM in deep submicron technologies for low-V_{MIN} applications. Note that, the SER increases at higher altitudes due to high energy particle strikes, which degrades the read-stability of 6T SRAM, too.

1.1.4 Hold-stability and Data Retention of SRAMs

SRAM hold-stability defines the ability to retain the prior written data in the SRAM bitcells while the SRAM is not doing any transient operations such as a read or write. The 6T bitcell hold-stability measures using a static metric called hold static noise margin (HSNM). HSNM measures using the sides of the least fitted square in the SRAM butterfly curve with WL turned off. For a 6T bitcell, HSNM is always greater than the RSNM, which is the upper limit of the RSNM. V_{DD} scaling and process variation both reduce the HSNM. There is another design metric called data retention voltage (DRV), which corresponds to the V_{DD} at which the HSNM is zero. DRV defines the lower limit of SRAM supply below which the SRAM bitcell content becomes lost. Keeping the SRAM V_{DD} very close to DRV saves leakage power for SRAM core arrays. However, at DRV, the 6T bitcell is the most vulnerable to particle strikes due to SER.

1.1.5 Process, Voltage, and Temperature Variation affecting SRAM Metrics

Device scaling in deep sub-micron technologies increases process variation, which affects the write-ability, readability, read-stability, hold-stability, and data retention capability of SRAMs. Due to the transient nature of the SRAM write and read operations, there exists significant switching current and the SRAM supply rails experience voltage droops leading to voltage variation in the bitcell level. A small temperature change does not affect the SRAM design metrics; however, SRAM designers must consider the effects of extreme temperature variations, which has a degrading impact on the parameters. Moreover, due to the statistical nature of the yield of SRAMs depending on the number of SRAM bitcells used, operating frequency, desired energy per cycle, leakage and area constraints, and other design parameters mentioned above, SRAM designs must undergo statistical design verification. To cope with the process, voltage, and temperature variations for the SRAMs to be safely operable, designers use sigma-based statistical design methodology and put heavy guard-bands in supply voltage, timing, etc. margins. This methodology known as design-for-the-worst-case has some overheads in energy, area, design time, and time to market.

1.1.6 SRAM V_{MIN} and Key Design Challenges in Lowering SRAM V_{MIN} with Energy Efficiency

The minimum operating voltage, or V_{MIN}, defines as the SRAM V_{DD} at which the SRAM can operate safely with a required statistical yield corresponds to a required write-ability, readability, read-stability, hold stability, and data retention capability. With device scaling the process variation increases, which limits the SRAM write-ability, readability, and other metrics making the SRAM V_{MIN} higher. SRAM designers can further reduce the SRAM V_{MIN} using alternative bitcells, peripheral assist, etc. techniques. A lower V_{MIN} is usually expected to give better energy efficiency for SRAMs; however, this is not always true. As

SRAM dynamic energy is quadratically proportional to the supply voltage, lowering SRAM V_{MIN} in sub-threshold supplies reduces the SRAM dynamic energy drastically, which helps to enable energy-constrained designs. However, the SRAM delay increases exponentially with V_{DD} scaling in sub-threshold supplies, which increases the leakage energy per operation. Thus, the total SRAM energy per operation has a minima called minimum energy point (MEP), which means lowering the SRAM V_{MIN} below MEP V_{DD} increases the SRAM energy again. Thus, tradeoffs for the V_{MIN} lowering are an interesting topic to investigate.

1.1.7 Major Thesis Contributions

Increased use of SRAMs in SoCs and demanding energy-constrained applications for longer battery-life are pushing the energy envelope for SRAMs to its limits. This thesis will investigate the circuit and architectural methods and their tradeoffs for achieving such low V_{MIN} and energy efficient SRAMs. The major contributions of this thesis are listed below.

For ultra-low power battery operated biomedical applications, conventional sub-threshold SRAM bitcells such as 8T could be expensive regarding active and leakage energy. Moreover, 8T bitcell in sub-threshold supplies suffers from a row half-select issue in write operations, which limits its operational ability. There are other state-of-the-art low-V_{MIN} sub-threshold bitcells; however, they have higher dynamic and leakage energy consumption across SRAM design knobs. For energy constrained biomedical applications, we propose a 9T half-select-free sub-threshold bitcell, which has 2.05X lower read energy and 1.28X lower leakage current compared to the conventional 8T SRAM. We compare the state-of-the-art sub-threshold bitcells in a bitline interleaving scenario for minimum energy point across various SRAM design knobs such as the fraction of read and write, number of bitcell rows per bank, word-width, number of words per row or the column mux factor, and capacity. Besides, we propose a low-energy read peripheral architecture to lower sequential active read energy in a read-dominated sub-threshold cache or buffer employing a read-modify-write scheme for half-select avoidance.

With the device scaling in deep sub-20nm technology, the 6T high-density (HD) FinFET SRAM bitcell suffers from poor write-ability and read-stability issues and does not work at the nominal supply voltage at the worst-case corner. None of the single peripheral assists could lower the 14nm 6T HD FinFET bitcell's worst-case V_{MIN} for a bitline interleaving scenario. State-of-the-art works show some combinations of dual read-write assists with specific percentages that lower the SRAM V_{MIN}; however, neither of the prior arts compare all of the effective combinations, nor they have equal constraints in the overall assist percentages in the comparison. We propose a comparison of all of the possible combinations of read-write assists for without and with bitline interleaving scenarios using a total of 20% assist percentages. This work shows how the working dual assist combinations have minimas in lowering V_{MIN} across combinations and could reduce the static V_{MIN} from 0.8V to very close to 0.6V. A testchip in 130nm bulk technology shows 240mV of V_{MIN} reduction from 0.71V to 0.47V using two write and one read assists. The testchip saves over 300X active power at the worst-case 90^{th} percentile SRAM V_{MIN} of 0.47V compared to the nominal voltage of 1.2V, which bolster the usage of multiple read-write assist combinations for low-power operation.

The V_{MIN} of the conventional 6T SRAM is heavily guard-banded to ensure functionality across process, voltage, and temperature variations. Thus, across process and temperature changes there is a large V_{MIN} variation and SRAMs designed using the worst-case methodology have higher energy and area overhead for the best and typical case SRAM dies. Canary circuits have been demonstrated to track and lower this guard-banding for SRAM for the data retention voltage margin; however, before this work, no literature exists that addresses dynamic V_{MIN} tracking and lowering of active energy. We propose a theory for canary write V_{MIN} tracking that relates the SRAM design knobs to the canary sensor design knobs, which shows the capability of tracking SRAM V_{MIN}. We also propose a set of pulse-shaping reverse assist techniques that degrades a core SRAM bitcell to make canaries. We show the tradeoffs of different reverse assist techniques. We further demonstrate, in a 130nm testchip, the properties of reverse assists for V_{MIN} tracking across voltage, frequency, and temperature

variations for the first time. Finally, we demonstrate a 256kb self-tuning closed-loop SRAM that employs multiple combined read-write peripheral assist (CPA) and automatically tracks its V_{MIN} using embedded in-situ canary sensors. Using both CPA and canary schemes allow us to lower the SRAM V_{MIN} beyond the worst-case 0.47V V_{MIN} to 0.38V V_{MIN} and save a maximum of 1444X active power and 12X leakage power compared to the full-scale supply voltage.

Designing reverse assist for canary SRAMs could be a challenging and time-consuming task, as it requires tracking the V_{MIN} across process, voltage, temperature, and frequency variations. We propose a mathematical framework that enables the analysis and design of reverse assist-based canary SRAM for SRAM V_{MIN} tracking. To reduce manual design time from months to days, we propose a set of algorithms that use the mathematical framework converting the canary design to a computational problem to be solved. And finally, we implement the algorithms in a Perl-based tool-flow that automates the analysis and design of reverse assist-based canary SRAM for SRAM V_{MIN} tracking.

1.1.8 Thesis Organization

We organize this thesis into nine chapters. Chapter 2 investigates how bitcell topology, array, and peripheral architecture influence the metrics such as energy, delay, leakage, etc. for SRAM design, and how the minimum energy point (MEP) affects the V_{MIN} from an energy efficiency standpoint.

We investigate in chapter 3 the challenges in a 14nm 6T HD FinFET SRAM's static and dynamic write, read, and overall V_{MIN} across design knobs such as process, temperature, and operating frequency. Moreover, we investigate the effects of single and dual combinations of peripheral assists to improve the V_{MIN} across process variation. Furthermore, we demonstrate a 256kb SRAM testchip design with two write assists (wordline boosting and negative bitline) and a single read assist (V_{DD} boosting) achieving a V_{MIN} of 0.47V that beats all other assist combinations to improve the 90^{th} percentile V_{MIN}.

Chapter 4 proposes the theory of canary SRAM using reverse assists for tracking dynamic V_{MIN} and investigates the tradeoffs for input and output design metrics. We further investigate the power and area tradeoffs of bitline pulse-height-degradation type reverse assist circuit in the canary write driver across design knobs, such as SRAM capacity, the number of canary cells, reverse assist voltage, and failure threshold condition.

We show in chapter 5 our findings of the analysis of the canary testchip data that reveals the first proof of concept of reverse assisted canary to be sensitive to voltage, frequency, and temperature variation that allows a canary SRAM to track the V_{MIN} of another core SRAM. We further show that wordline pulse-height-degradation for 130nm bulk 6T SRAM is insensitive to frequency changes; however, it is sensitive to all other design knobs such as voltage and temperature variations.

Chapter 6 classifies the pulse-shaping techniques such as pulse height, slope, and width for wordline and bitline type reverse assists in the design of canary sensor SRAMs. We derive the sensitivity metrics of comparison for wordline and bitline pulse-shaping reverse assists across the design knobs such as the strength of the reverse assist, supply voltage, frequency, and temperature variation. We further propose pulse-shaping wordline type reverse assist circuits for canary sensor SRAM design and show their tradeoffs in metrics, such as area, energy, and figure-of-merit.

We describe in chapter 7 the design and development of a closed-loop self-tuning 256kb SRAM testchip with 0.38V-1.2V extended operating range using multiple combined peripheral assists and in-situ V_{MIN} tracking canary sensors. The SRAM has a maximum of 337X active power reduction using combined read-write peripheral assists, a 4.3X power reduction using V_{MIN} (overall 1444X active power reduction capability), and has a maximum of 12.4X leakage reduction capability.

Chapter 8 shows a mathematical framework to determine the optimal V_{MIN} tracking condition based on the available supply voltage granularity, which translates to the design consideration for reverse assist-based canary sensors to track SRAM V_{MIN} across process,

voltage, temperature, and frequency variations. We further propose a set of algorithms for analysis and design of reverse assist-based canaries to track SRAM V_{MIN} across process, voltage, temperature, and frequency variations. A Perl-based reverse assist design tool-flow introduces support for the analysis and design of pulse-shaping wordline and bitline reverse assist-based canaries, which track SRAM V_{MIN} across process, voltage, temperature, and frequency variations.

And finally, we conclude from this thesis in chapter 9 followed by derivations of battery discharge equations, a list of publications, a list of acronyms, and the bibliography.

Chapter 2

Bitcell Topology, Array, and Peripheral Architecture for Energy Efficient low V_{MIN} SRAMs

2.0.1 Motivation for Bitcell Topologies and Peripheral Circuits

[1]Energy-constrained IoT devices such as portable biomedical devices have stringent energy requirements, which require long-term data processing for electrocardiogram (ECG), electromyogram (EMG), etc. applications. These biomedical applications operate at lower clock frequencies starting from a few hundred kHz to a few MHz [21] [22] and impose energy restrictions on its system on chip (SoC) subcomponents, such as logic an SRAM. Due to the quadratic dependence of energy to the supply voltage (V_{DD}), energy in logic and SRAMs in IoT SoCs reduces using V_{DD} scaling. Lowering the V_{DD} below the threshold voltage (VT) of the MOSFET in a CMOS process, allows it to enter into the sub-threshold region of the device. Operating both logic and SRAM in scaled sub-threshold V_{DD}s reduces energy consumption, which optimizes energy per operation [23] [24]. Although scaling V_{DD} down

[1]This chapter is based on the published paper titled "An ultra low energy 9T half-select-free subthreshold SRAM bitcell" [AB1] and "An Ultra-Low Energy Subthreshold SRAM Bitcell for Energy Constrained Biomedical Applications" [AB2].

to sub-threshold supplies lower the SRAM dynamic energy, it increases the delay in an SRAM exponentially. Thus, at scaled sub-threshold V_{DD}s, the exponential delay results in the exponential increase in leakage energy per operation. Therefore, the total SRAM energy per operation has an optimum point named minimum energy point (MEP), below which pushing the SRAM minimum operating voltage (V_{MIN}) makes the SRAM energy-inefficient by increasing the total energy per operation. Notwithstanding voltage-scaling degrades the delay in logic and SRAMs, sub-threshold V_{DD}s allow adequate performance for meeting the throughput needs of energy constrained IoT applications, such as ECG, EMG, etc.

Process variation in sub-threshold V_{DD}s makes the conventional 6T bitcell to have poor read static noise margin (RSNM) [25], which is unreliable for sub-threshold operation. Alternative sub-threshold bitcells [26] [27] [28] [29] in the state-of-the-art improve in write-ability and read-stability design parameters by trading-off area, energy, etc. other metrics. Nevertheless, the sub-threshold compatible 8T [30] bitcell faces the half-select [28] issue in a bitline interleaving scenario in a write operation that uses a peripheral write assist, such as wordline boosting [31] [32]. The half-select issue increases the chances of read-disturb and unnecessary energy drainage during a write operation. Also, the degradation of read-stability in half-selected [32] bitcells imposes further limitations on the usage of peripheral assists in sub-threshold supplies, such as the boosted wordline. Although the state-of-the-art alternative bitcells promise half-select-free write operations, they could burn more energy in write and read cycles, which need an in-depth evaluation for comparison of the alternative bitcells.

Another way to avoid the half-select issue in 6T SRAMs is to employ read-before-write operation [27] [33] instead of usual write, or one can design a half-select-free SRAM bitcell [27] [28] [29] [34], which decouples the read and write operations. Implementing a read-before-write architecture in sub-threshold V_{DD}s can be a challenging and time-consuming task compared to a usual column-mux-based SRAM design. Moreover, the soft error disturbs (SED) are critical [35] in scaled sub-threshold supplies, and the bitline interleaving in the memory architecture [27] becomes a must, which improves on multi-bit single word SEDs. Thus, an

alternative half-select-free bitcell topology is preferable that supports bitline interleaving. Therefore, it would be interesting to investigate what alternative bitcell topologies are suitable for energy-constrained biomedical applications in a column mux scenario. Also, it is vital to compare the trends and trade-offs of the state-of-the-art alternative bitcells for ultra-low power IoT applications. Thus, in this chapter, our simulation-based work compares our proposed bitcell to the state-of-the-art sub-threshold bitcells across SRAM design parameters.

2.0.2 Prior Art in Sub-threshold Bitcell Topologies

The use of conventional 6T SRAM bitcell is widespread, which we show in Figure 2.1a. The 6T bitcell consists of two back-to-back inverters, and those form a latch that stores a logic "1" in one of the Q/Qb node, and a logic "0" in the other Qb/Q node. Thee two NMOS transistor M5 and M6 (Figure 2.1a) serve as two access transistors in a write or read operation.Although widely used, the 6T bitcell is not robust in sub-threshold V_{DD} because of a poor read static noise margin [25] (RSNM) and the half-select [32] issue in a column mux scenario. Alternative SRAM bitcells capable of operating at sub-threshold V_{DD}s, are based on the 6T bitcell. Figure 2.1b shows the conventional 8T SRAM bitcell [30], which is based on the 6T bitcell and operates at sub-threshold V_{DD}s. The 8T bitcell has a separate read path using M8 and M9 (2.1b) NMOS transistor forming a 2T read buffer. The read buffer of the 8T bitcell senses the information stored in the Qb node in a read operation, enabling decoupled read and write operations, which allow us to size the read and write paths independently. Thus, the 8T bitcell topology act as a bitcell design knob for energy efficient sub-threshold SRAM design. Figure 2.1c shows the Schmitt-trigger-based bitcell [26], which is a sub-threshold bitcell, which has lower minimum operating voltage (V_{MIN}). Having the hysteresis property of a Schmitt-trigger, the bitcell has higher RSNM values that bolster the read operation at lower V_{MIN}. Even though the sub-threshold bitcells such as 8T and Schmitt-trigger-based bitcells are robust in write and read operations, there is significant energy cost for using them in a column mux scenario.

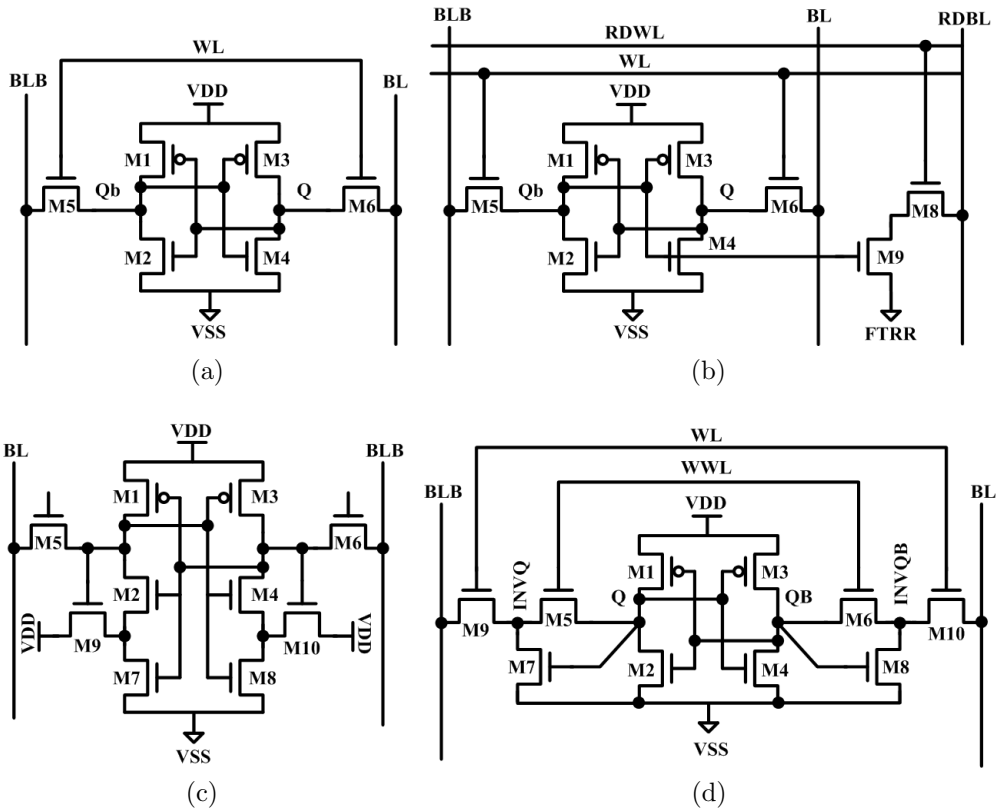
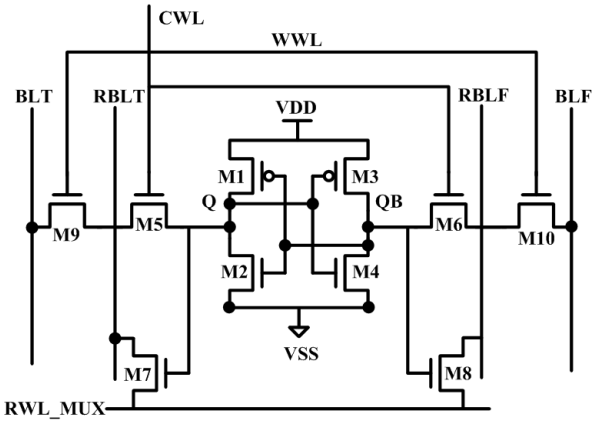
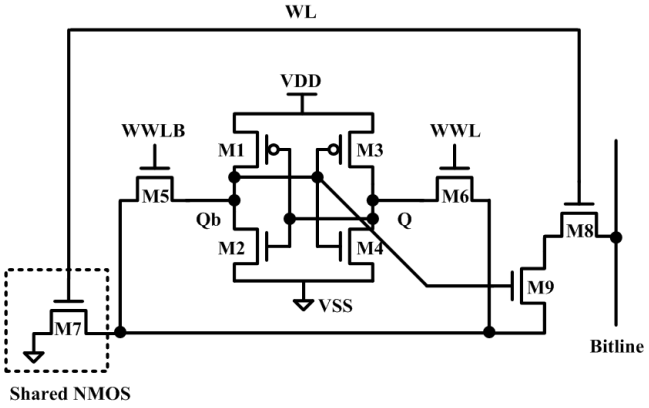


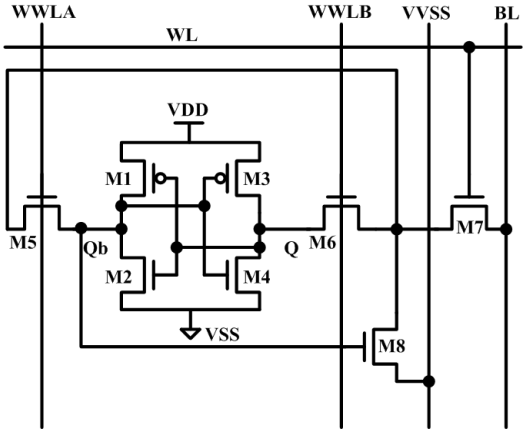
Figure 2.1: (a) 6T SRAM bitcell (© 1996 IEEE). (b) Conventional 8T SRAM sub-threshold bitcell (© 2008 IEEE). (c) Kulkarni's Schmitt-trigger-based sub-threshold SRAM bitcell (© 2007 IEEE). (d) Chang's 10T sub-threshold bitcell (© 2009 IEEE).



(a)



(b)



(c)

Figure 2.2: (a) Feki's 10T SRAM sub-threshold bitcell (© 2012 IEEE). (b) Chiu's 8T sub-threshold bitcell (© 2011 IEEE). (c) Yang's 8T SRAM bitcell (© 2011 IEEE).

The 8T and Schmitt-trigger-based bitcells in a write operation suffers from the half-select problem [32], too, which cause unnecessary energy consumption employing a column mux. There are alternative bitcell topologies available in the literature, those are half-select-free, such as Chang's 10T [27] (Figure 2.1d), Feki's 10T [28] (Figure 2.2a), and Chiu's 8T [29], as shown in Figure 2.2b. Even though Yang's 8T [34] (Figure 2.2c) is not claimed to operate at sub-threshold V_{DD}s, we include this bitcell for comparison in this work due to topological similarity with Chiu's sub-threshold bitcell. These half-select-free bitcells mostly have two separate wordlines for write and read operations, which allows us to size the write and read paths separately, such as Feki's bitcell, as shown in Figure 2.2a. However, some of the bitcells have common write or read nodes, those make the sizing the bitcells' write and read paths a challenging task, such as Chang's, Yang's, and Chiu's bitcell. All of these alternative bitcells improves some design metrics, such as read-stability, write-ability, V_{MIN}, leakage, etc.; however, they have limitations, too. The next section describes the disadvantages of the bitcells mentioned above.

2.0.3 Limitations of State-of-the-art Sub-threshold Bitcells

The sub-threshold bitcells mentioned in the previous section have drawbacks, too. Even though Kulkarni's bitcell presents the lowest reported V_{MIN}, its Schmitt-trigger-based feedback structure can consume more dynamic and leakage energy, which uses the additional transistors M9 and M10, as shown in Figure 2.1c. The M9 and M10 transistors strengthen the internal storage node causing higher dynamic energy dissipation and create a higher number of source or sink paths from V_{DD} to V_{SS}, which lead to more leakage current. Secondly, due to two additional transistors in the 10T structure of Kulkarni, Feki, and Chang's bitcells, they inherently should consume more dynamic energy compared to the 8T bitcells, such as the conventional 8T, Chiu, and Yang's bitcells. As we assume to size all the bitcells using a set of reference design metrics, bitcells with additional transistors will have more dynamic energy. Besides, Chang's bitcell having other leakage paths from bitline to ground due to transistors

M7 and M8, such as paths BLB-M9-M7-V_{SS}, BL-M10-M8-V_{SS}, it increases the bitcell leakage current. Note that we assume that the bitcells' back-to-back inverter sizes are the same and each of the control signals such as wordline has the same activity factor. Thus, bitcells having multiple wordlines or control signals, such as Chang's, Feki's, and Yang's bitcells, should have more dynamic energy compared to bitcells having fewer control signals. Thirdly, bitcells using shared path for write and read operations, such as Kulkarni's, Chiu's, and Chang's bitcells, should have higher dynamic energy overhead due to bitline precharge operation after the end of both write and read cycles. Also, in the bitline interleaving scenario, bitcells unselected in the same row becomes half-selected, and they experience unnecessary read stress in the write operation, which uses peripheral assists, such as boosted wordline. Thus, bitcells affected by the half-select problem should burn more dynamic and leakage energy, such as Kulkarni's, Chiu's, and Yang's bitcells. Therefore, for capturing these effects mentioned above for energy dissipation, we simulate all these bitcells in a column mux scenario. The next section discusses the half-select issue in details.

2.0.4 SRAM Half-select Issue in Write Operation

Figure 2.3 depicts the half-select problem in a column mux (CM) 4 scenario during an SRAM write operation. Here, we assume that the SRAM uses multiple banks. Each SRAM bank comprises of the same capacity core array of sub-threshold bitcells. The diagram illustrates that, in a column mux 4 scenario, every four-bitcell column shares a single I/O, which has the sub-components, such as a precharge logic, read and write column muxes, a write driver, and read logic. Choosing an SRAM address selects a word in an SRAM row by selecting one of the physical rows and multiple physical columns in the bank. For example, selecting the first word in the row chooses only the first physical bitcells from the columns, which consist of sets of four physical bitcells. Thus, bitcells unselected in this process, those located in the same row but different word are half-selected bitcells. In an SRAM write operation, these half-selected bitcells experience read stress similar to a read operation causing

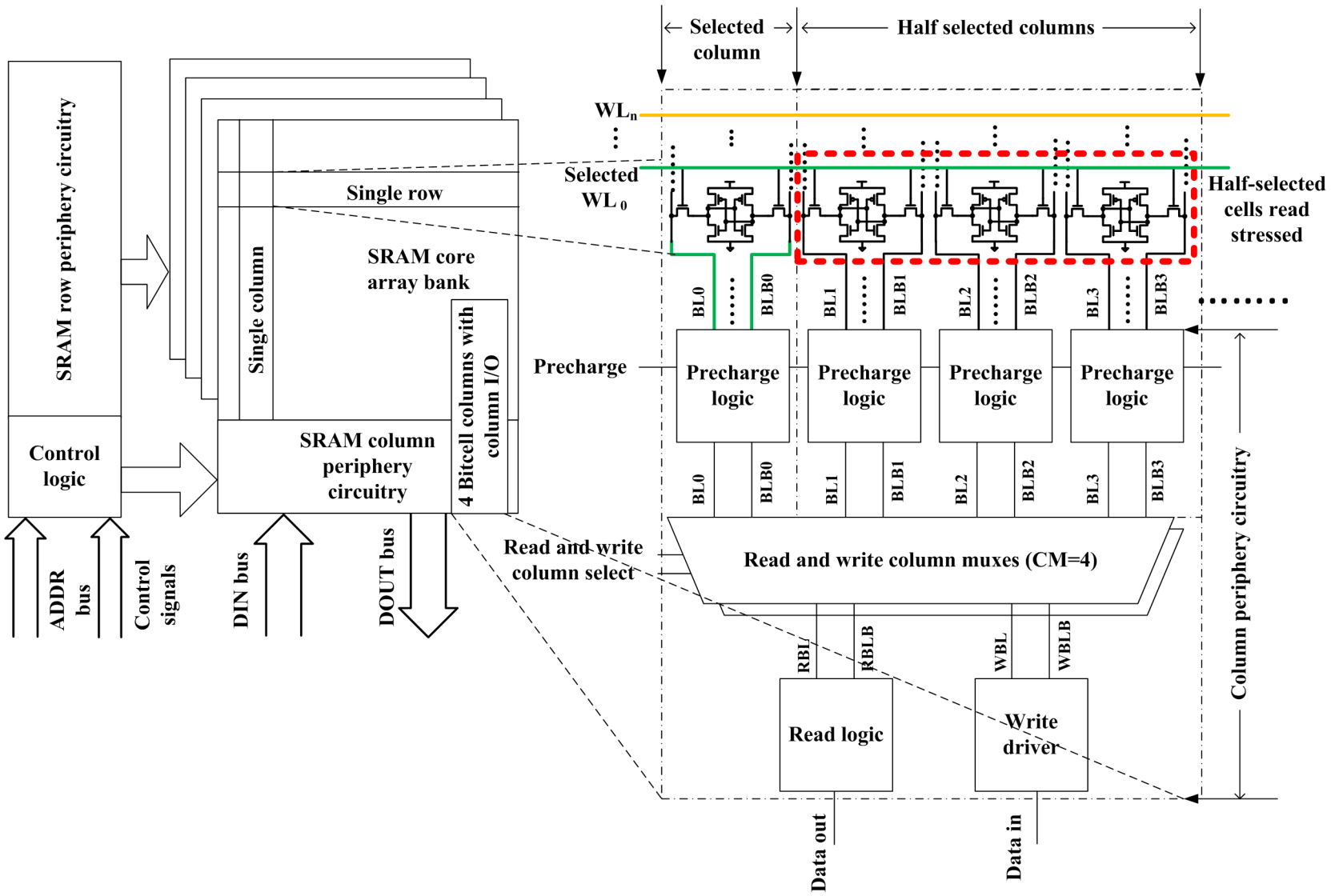


Figure 2.3: Illustration of SRAM half-select issue in write operation.

unnecessary energy drainage. However, the potential issue with the half-selected bitcells happens using a wordline boost type write-assist [31], [32] for improving the write-ability, which can result in a destructive read in the half-selected bitcells. In other words, using a boosted wordline write assist can affect the half-selected bitcells to flip, which is called the half-select issue. Although column multiplexing causes the half-select issue in write, its implementation is widespread in SRAM architectures due to the reason that the SRAMs based on column mux are easy to design compared to the read-before-write [27], [33] SRAM architectures.

Using a half-select-free [29] sub-threshold SRAM bitcell is another way of bypassing the half-select issue. Also, column mux-based designs are easy to implement using a half-select-free bitcell. However, half-select-free bitcells usually have a higher number of devices and control signals, which can cause unnecessary dynamic and leakage energy drainage in a core array. Moreover, some half-select-free sub-threshold bitcells have shared nodes in write and read paths that can affect sizing issues. Therefore, we present a couple of research questions in the next section to investigate, those relate to the metrics of sub-threshold bitcells across SRAM design knobs.

2.0.5 Research Questions

We plan to investigate the following research questions in this chapter:

- How to reduce dynamic energy and leakage current in bitcells topologically;
- How to avoid the half-select issue in a sub-threshold bitcell;
- How do the state-of-the-art sub-threshold bitcells' energy, delay, etc. metrics compare;
- What are the trade-offs among the different state-of-the-art sub-threshold bitcells in core array design using design knobs, such as bank size, rows per bank, column mux, etc.?

2.0.6 Proposed Approach

For comparison we assume that using suitable read and write peripheral assists [31], [32] mitigating the issues of read-stability [32] and write-ability [31] in sub-threshold bitcells incur a minimal penalty in energy per operation and area in SRAMs and SoCs. Also, we assume that we can allow trading off SRAM area for better energy efficiency, which is vital for the battery-life in biomedical applications. To answer the research questions, we demonstrate that a comparative lower energy bitcell requires the following set of properties.

- Completely decoupled half-select-free write and read paths to size independently for low-energy requirements in a column mux scenario;
- Lesser number of control signals to switch in write and read operations contributing to lower dynamic energy per operation;
- Lesser number of leakage paths to reduce leakage current.

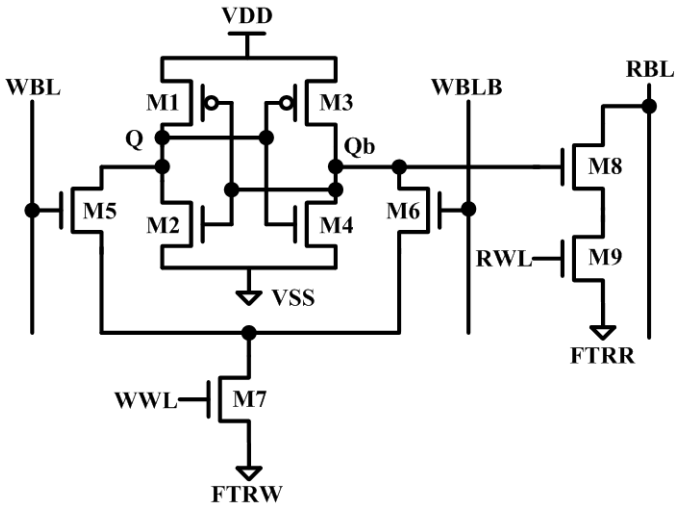


Figure 2.4: Schematic of the proposed half-select-free 9T SRAM bitcell (© 2013 IEEE).

Figure 2.4 shows the proposed bitcell [36] [37] ($W_{M1,\ M3} = 0.4u$, $L_{M1,\ M3,\ M5,\ M6,\ M7} = 0.22u$, $W_{M2,\ M4} = 0.28u$, $L_{M2,\ M4,\ M8,\ M9} = 0.15u$, $W_{M5,\ M6,\ M7} = 0.45u$, and $W_{M8,\ M9} = 0.36u$). Using the transistor sizes of a reference sub-threshold 8T bitcell used in our SRAMs for a Body

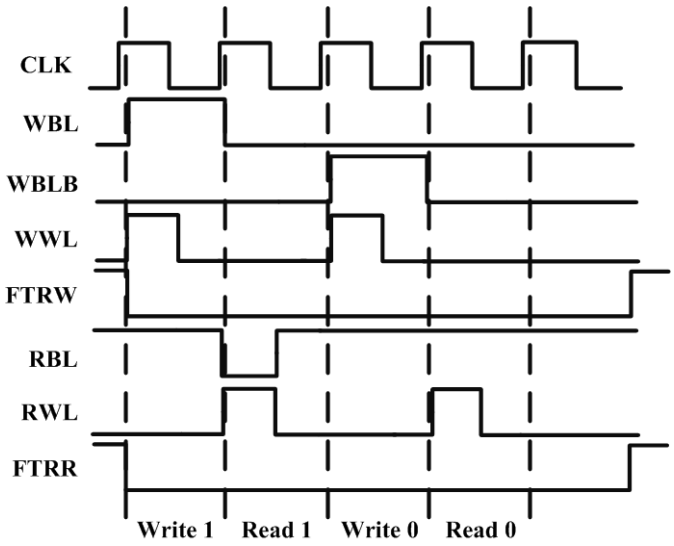


Figure 2.5: Write and read waveforms of the proposed 9T SRAM bitcell (© 2013 IEEE).

Area Sensor Node (BASN) chip [21], we size our 9T bitcell's back-to-back inverters (M1, M2, M3, and M4), and the read-buffer transistors (M8, M9). To cope with the sub-threshold process variation, we choose the transistor gate widths and lengths as non-minimum after running Monte Carlo simulations for write margin, HSNM, read time, etc. design metrics. Our 9T bitcell comprised of two back-to-back inverters similar to the 6T bitcell and an NMOS transistor network used for write accesses, which resembles a differential amplifier-like structure. We use a two transistor read buffer similar to the 8T read buffer [30] for read operations. Figure 2.5 depicts the waveforms for write and read operations of our 9T bitcell.

During a write cycle, only one of the write bitlines WBL or WBLB (Figure 2.4) remains at the logic high, and the other goes to logic low. After the write-wordline WWL turns on, the corresponding internal node storing a logic "1" discharges through the write path. In a write operation, we pull down the WBL and WBLB nets of the half-selected bitcells to the ground, which prevents the half-select issue. Our 9T bitcell operates without any precharge operation involving the write bitlines and consumes less dynamic energy. During a read cycle, we trigger the read wordline RWL to evaluate the read bitline RBL (Figure 2.4), which precharges to V_{DD} in the previous cycle. For our 9T bitcell, the node Qb serve as the reference node for the read operation. During a read, the RBL discharges if the Qb node

holds a logic "1" in the bitcell, which denotes a read "1" operation. On the other hand, the RBL stays at V_{DD} for Qb holding a logic "0" denoting a read "0" operation. For reducing the standby leakage of our 9T bitcell, the FTRR and FTRW signals remains at V_{DD} for the unselected rows not toggling in a normal read or write operation. For the selected row, the states of FTRR and FTRW are logic "0" s. We report the leakage savings by pulling the FTRR and FTRW signals to V_{DD} compared to setting it to V_{SS} is 34% at the TT_0.4V_27C corner. This scheme by pulling some of the signals to V_{DD} can improve leakage energy savings in lower technologies [30], also. We show the waveforms for read and write operations in Figure 2.5.

2.0.7 Minimum Energy per Operation (EPO) of Sub-threshold SRAMs

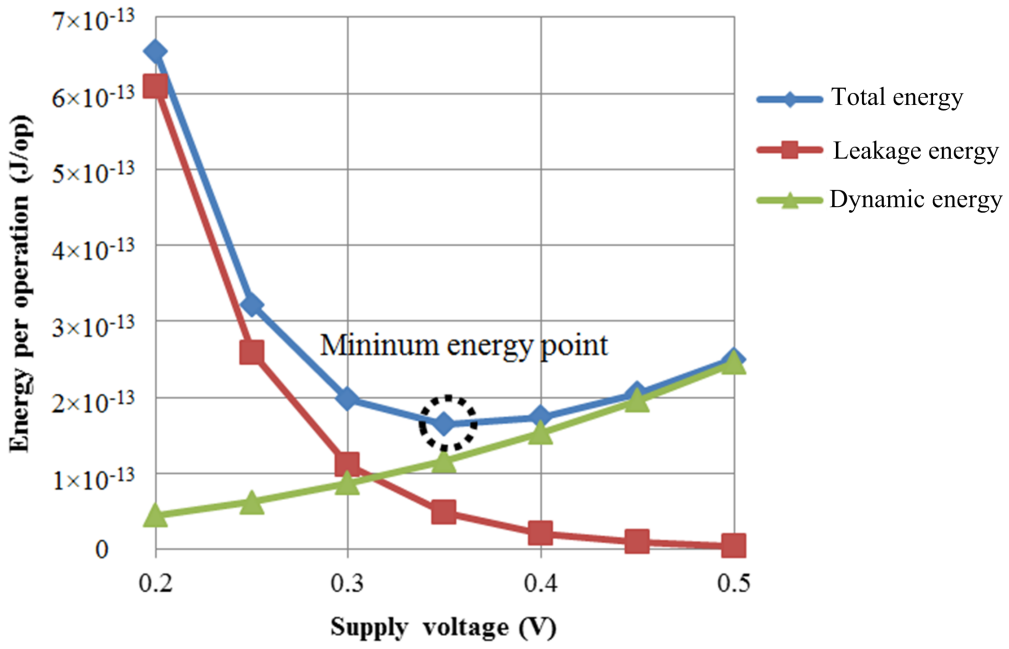


Figure 2.6: Minimum energy point (MEP) of SRAM.

The SRAM dynamic energy per operation (EPO) decreases with the V_{DD} scaling in SRAMs, as shown in Figure 2.6. However, the SRAM delay exponentially increases with

decreasing supply voltage at sub-threshold V_{DD}s, which leads to the exponential increase in leakage EPO with scaling down V_{DD} for sub-threshold SRAMs. Thus, the total EPO of sub-threshold SRAMs usually has a minimum energy point (MEP) (Figure 2.6), which stays within the sub-threshold supply voltage region. The core array is responsible for most of the leakage EPO in larger sub-threshold SRAMs. Moreover, the core array can consume a significant amount of dynamic EPO due to higher capacitance in the bitlines for the presence of multiple bitcells in a column. Increasing the dynamic EPO while keeping the leakage EPO fixed, lowers the SRAM V_{DD} at the MEP. On the other hand, lowering the leakage EPO while keeping the dynamic EPO fixed, reduces the MEP V_{DD}, too. In the first case, although the MEP V_{DD} shifts to a lower supply voltage, the total EPO increases. Nevertheless, reducing the leakage EPO gives a two-fold benefit of both lowering MEP and MEP V_{DD}. Furthermore, reducing the leakage and dynamic EPO at the same rate does not affect the MEP V_{DD}; however, it lowers the MEP itself.

2.0.8 Read-Write Weighted Energy per Operation (EPO) and Fraction of Read and Write

In SRAMs, we usually perform a higher number of read operations than write. Calculating an equivalent MEP requires weighing the read and write EPOs accordingly, which gives the read-write weighted EPO. Thus, the equation (2.1) gives the read-write weighted average EPO.

$$E_{avgop} = E_{wr} * (1 - F_{rdwr}) + E_{rd} * F_{rdwr} \tag{2.1}$$

Here, the parameter E_{avgop} denotes read-write weighted EPO; E_{wr} and E_{rd} are the write and read EPO, respectively. The parameter F_{rdwr} is the fraction of read and write, as shown in equation (2.1), which denotes the average number of read operations out of the total

number of read-write operations. Noticeably, if the E_{rd} is smaller than the E_{wr}, increasing the F_{rdwr} parameter decreases the weighted average EPO E_{avgop}.

2.0.9 Experimental Setup

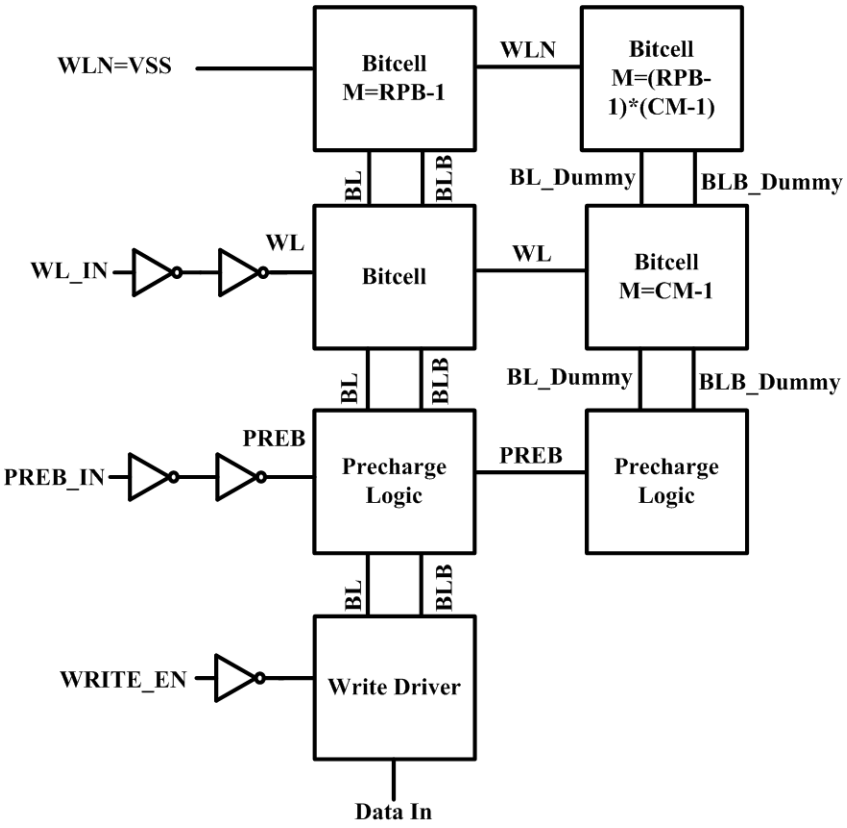


Figure 2.7: Experimental setup for dynamic write energy measurement for sub-threshold SRAM bitcells in a column mux scenario.

We perform all our simulation-based experiments in a commercial 130nm bulk technology at the TT_27C corner, using Cadence's Spectre simulator. We run 1000 Monte Carlo simulations for the mismatch analysis for comparison among bitcells at V_{DD} = 0.4V. We execute two sets of experiments: For comparisons of the energy and delay numbers, we perform the first experiment based on the setup shown in Figure 2.6 and Figure 2.7, where instead of the actual drivers we use voltage sources as input waveforms. However, in the second experiment, to compare the EPO metric and to obtain the MEP data, we use the experimental setup shown in Figure 2.6 and Figure 2.7 with the actual drivers. Here, the

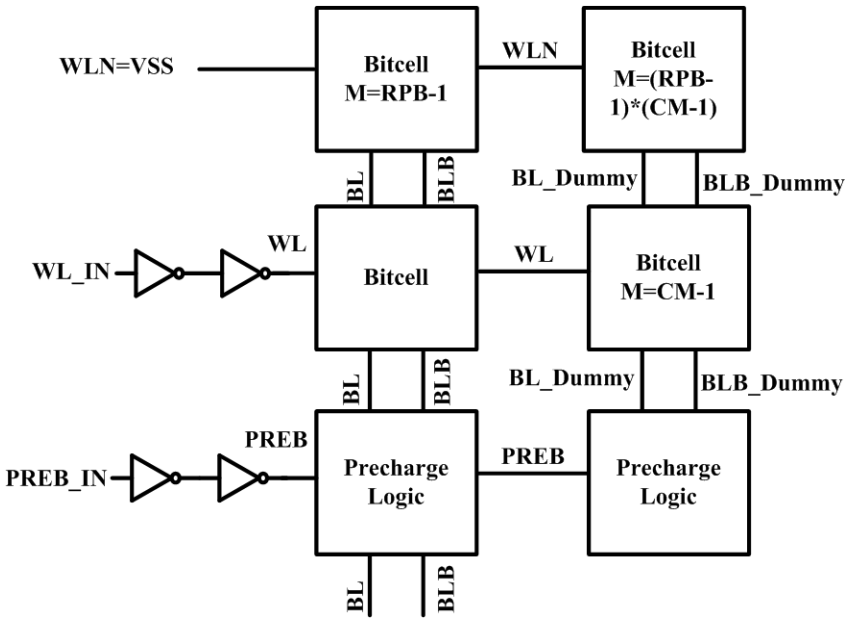


Figure 2.8: Experimental setup for dynamic read energy measurement for sub-threshold SRAM bitcells in a column mux scenario.

signals "WL," "PREB," and "WRITE_EN" stand for wordline, precharge bar, and write enable, respectively. For the drivers of wordline, bitline and write enable, etc. signals, we apply extracted inverter netlists from a standard cell library, which ensures that the rise or fall time of the inverter and buffer output signals are realistic in the sub-threshold V_{DD}s. Figure 2.6 and Figure 2.7 show the write and read setup for the experiments. Each of the write and read setups has two columns. We represent the leftmost column as the actual column for write or read operation, which gives the total load of the modeled bitline as the number of rows per bank (RPB) times a single bitcell's bitline load. We model the second column that represents the load of the wordline corresponding to an SRAM I/O column, which is one less than the column mux factor (CM) times the load of the wordline of a single bitcell. Overall, using the setup we model "RPB X CM" number of bitcells per physical bitcell-column in a single column mux I/O of an SRAM bank, which measures the dynamic energy. For example, to measure the dynamic energy using CM = 4 and RPB = 16 in an SRAM bank, our setup models four physical bitcell columns associated with each column mux 4 I/O, and 16 X 4 = 64 bitcells per bank in a column mux 4 I/O. To generate various

energy, delay, and MEP numbers, we run multiple simulations of different instances with RPB = 4, 8, 16, 32, and 64 values. For generating dynamic energy values across word-widths, we multiply the energy values of the set of columns consisting of multiple bitlines in a mux I/O, across SRAM word-widths of 2, 4, 8, 16, and 32. To extract the leakage values, we use the netlists of single bitcells, and those have individual voltage supply source for each circuit. We calculate the leakage numbers of the bitcells to compute the corresponding leakage values of core arrays. Finally, for each SRAM macro to calculate the total EPO values for the calculation of MEP, we add the modeled dynamic and leakage EPO values. Since prior works report a range of 5-46KB [21] [22] [38] [39] of SRAM usage in biomedical SoCs, we limit our simulation of memory sizes from 2-32KB range.

For comparing the bitcells for energy efficiency, we quantify the total EPO and the MEP metrics with some assumptions. Usually, an SRAM has not only bitcell arrays but also peripheral circuits, such as drivers for wordline and bitline, precharge logic, control logic, etc. circuits. Thus, to estimate of the MEPs of the state-of-the-art bitcells, we add some drivers and periphery circuits in the test bench that would be switching. We apply the same driver stages for wordlines across the bitcells. Also, we use the same driver stages for bitlines in most of the bitcells. Thus, all the bitcells that require a pull-down type write driver use the same circuit. Nevertheless, for a pull-up type write driver, we use buffers of comparable strengths. For bitcells requiring precharge cycles, we incorporate a precharge circuit, as shown in Figure 2.6 and Figure 2.7. Note that the bitcells that has multiple wordlines for write and read operations or use a precharge operation may have higher dynamic EPO overhead due to additional peripheral circuits. On the other hand, larger core arrays may have more leakage EPO than the periphery. Thus, we execute our experiments to obtain the dynamic EPO, leakage EPO, as well as the total EPO for each bitcell array with the corresponding peripheral circuits.

2.0.10 Experimental Assumptions

As sub-threshold SRAMs demand lower leakage numbers, we use a commercial 130nm low-leakage bulk technology for this work. To simulate and quantify the bitcell design parameters, we perform our experiments in the typical-typical (TT) corner. As we target biomedical applications for Body Area Sensor Node (BASN) [21] applications, we use 27°C room temperature condition for the simulations.

For comparisons, we design the 6T structures in all the bitcells using the same sizes, which make the statistical $\mu + 6\sigma$ data retention voltage (DRV) close to 150mV for all bitcells. Also, for each bitcell, we make the statistical $\mu - 6\sigma$ hold static noise margin (HSNM) at 0.4V roughly equal to 120mV across local and global variations. This work assumes that all the read operations are full swing. Thus, we exclude the use of a sense amplifier in a read for the experiments. Figure 2.6 and Figure 2.7) show the models for the simulations, which account for the energy dissipation in bitlines and wordlines in a column mux scenario for a set of bitcell columns present in a mux I/O. Also, we assume that the core array to be sufficiently larger, and its MEP contribute most to the MEP of the modeled SRAM macro. Moreover, in a real SRAM scenario, the MEP trends will change due to the inclusion of control logic, pre-decoder, and wordline drivers with the core array, accordingly, which depend on the periphery energy consumption. However, this work compares the core MEP trends of all the bitcells' modeled SRAM macros, which assumes an equal MEP for the periphery for all cases.

2.0.11 Evaluation Metrics

We choose the following metrics for evaluation and comparison of the state-of-the-art sub-threshold bitcells.

- Write energy and write delay;
- Read energy and read delay;
- Leakage current;

Table 2.1: Monte Carlo data comparison of bitcell design metrics at TT_0.4V_27C corner (energy in fJ, time in ns and current in pA units).

Metrics	**6T**	**8T [30]**	**10T [28]**	**10T [26]**	**This work**	**10T [27]**	**8T [29]**	**8T [34]**
Read time (μ)	0.3	0.73	0.28	0.48	0.45	0.69	0.65	0.45
Read energy (μ)	0.82	1.46	1.79	1.19	0.71	2.26	0.96	0.75
Write time (μ)	0.19	0.2	0.47	0.26	1.33	0.46	1.39	3.24
Write energy (μ)	1.35	1.36	1.24	1.98	1.21	421.71	1.69	180.67
Leakage current (μ)	187.8	188.2	136.1	468.4	146.1	211.8	161.9	245.3

- Data retention voltage (DRV);
- Hold static noise margin (HSNM).

To make array level comparison, we choose the total EPO and MEP [37] as the evaluation metrics across the design knobs for SRAM core array.

2.0.12 Results and Comparisons

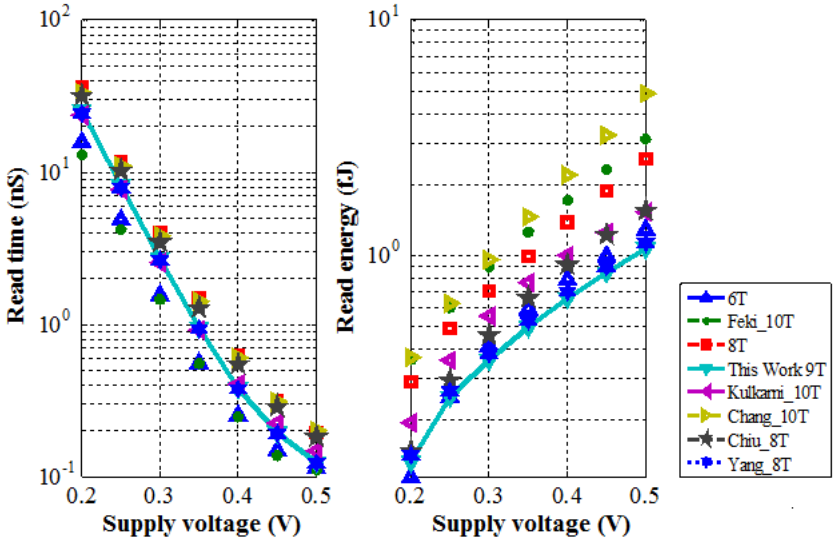


Figure 2.9: Read time vs. supply voltage and read energy vs. supply voltage (© 2013 IEEE).

We compare our proposed bitcell with the state-of-the-art sub-threshold SRAM bitcells across design parameters, such as energy, delay, leakage etc. metrics. To do a fair comparison, we size the back-to-back inverters: M1, M2, M3, M4 (Figure 2.4)) and two NMOS pass transistors: M5 and M6 (Figure 2.4) the same in all the state-of-the-art bitcells mentioned

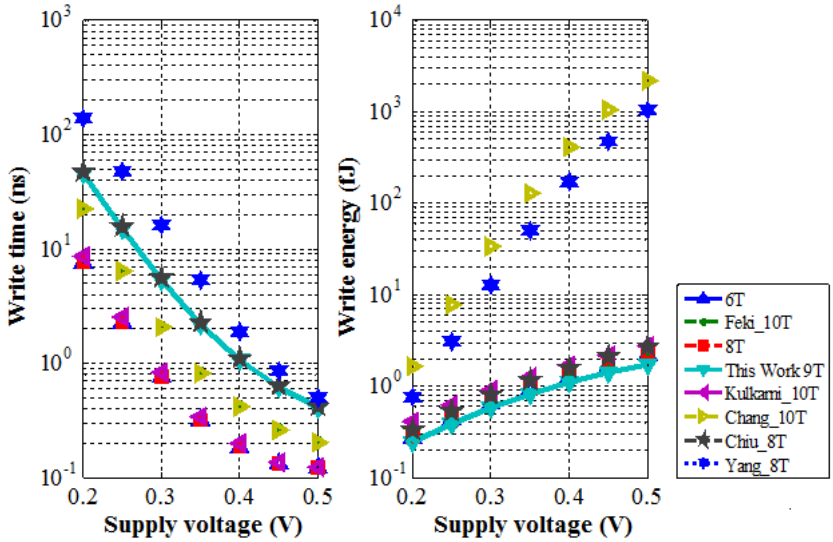


Figure 2.10: Write time vs. supply voltage and Write energy vs. supply voltage (© 2013 IEEE).

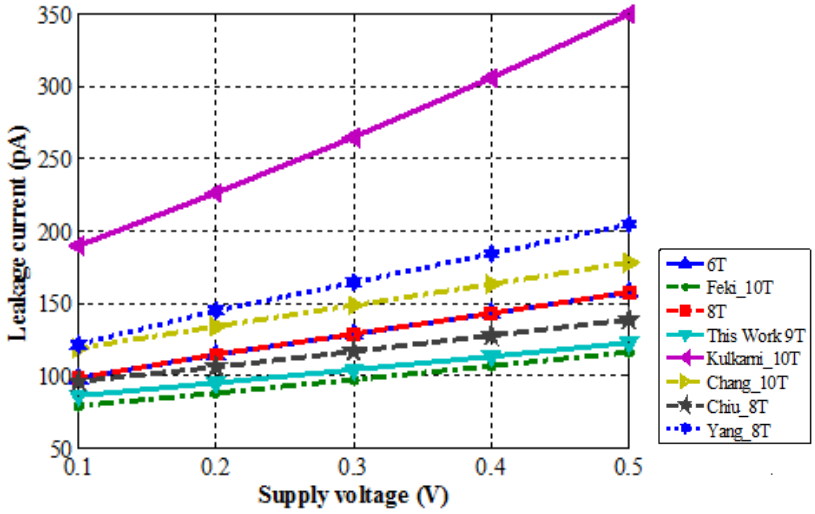


Figure 2.11: Leakage current vs. supply voltage (© 2013 IEEE).

in this chapter. The values of widths and lengths for the bitcells used are $W_{M1,\ M3} = 0.4u$, $L_{M1,\ M3,\ M5,\ M6,\ M7} = 0.22u$, $W_{M2,\ M4} = 0.28u$, $L_{M2,\ M4,\ M8,\ M9} = 0.15u$, and $W_{M5,\ M6,\ M7} = 0.45u$. Due to this reason, all the bitcells have the μ DRV close to 74 mV, and the μ HSNM roughly equal to 154 mV at the TT_0.4V_27C corner under local and global variations. The comparison with the half-select-free bitcells yields that the mean read energy of our work is 3.18X lower than Chang's [27], and 2.52X lower than Feki's [28]. Nevertheless, our bitcell has 50% larger read time and 7X larger write time compared to the conventional 6T bitcell at the same corner. Figure 2.9, Figure 2.10, Figure 2.11, and Table 2.1 give the comparison of the bitcells for voltages (0.2-0.5V) at the TT_0.4V_27C corner across different design metrics.

Comparison of Total Energy per Operation

We show the plot for total energy vs. supply voltage and MEP for the bitcells with column mux (CM) = 4 and RPB = 16 in Figure 2.12 and Figure 2.13, respectively. For the generated data we assume that on an average there are three reads and one write among four write-read operations, which sets the value of the fraction of read and write (F_{rdwr}) to 0.75. We observe that for most of the 8KB SRAMs, the MEP V_{DD} is close to 0.3V, and the MEP V_{DD} is around 0.35V for most of the 32KB SRAMs. However, Chang's bitcell becomes an exception to the bitcells' MEP trends, which does not have an MEP within 0.2-0.5 V range for the 8KB and 32KB scenarios. The reason for having a lower MEP V_{DD} below 0.2V is because Chang's bitcell has much higher dynamic EPO in the sub-threshold region, compared to its leakage EPO than other bitcells (Figure 2.9 and Figure 2.10). We take 0.2V as the MEP V_{DD} for larger SRAM macros using Chang's bitcell, as its MEP lies below the 0.2-0.5V region. On the contrary, even though, Yang's bitcell has a much higher MEP than other state-of-the-art bitcells, its MEP V_{DD} is close to 0.25V. Thus, the MEP V_{DD} of Yang's bitcell across 8KB and 32KB SRAM capacity is 16.66% and 28.57% lower than most of the bitcells' MEP V_{DD}, as shown in Figure 2.12 and Figure 2.13, respectively.

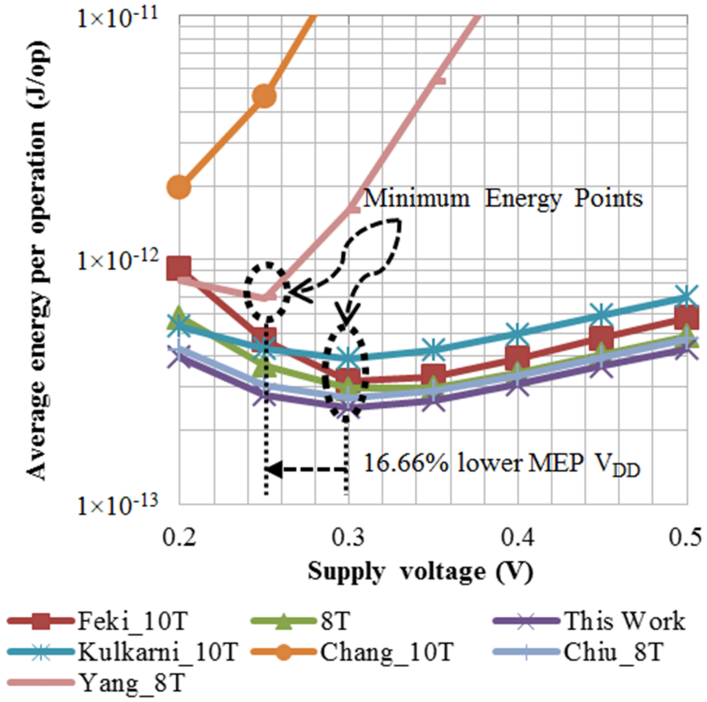


Figure 2.12: Total energy per operation vs. supply voltage of 8KB SRAMs (CM = 4, RPB = 16).

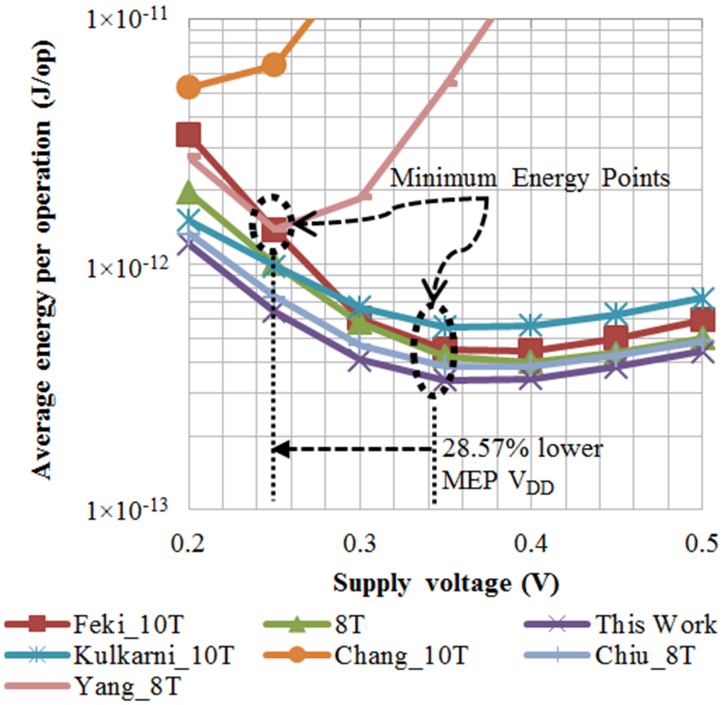


Figure 2.13: Total energy per operation vs. supply voltage of 32KB SRAMs (CM = 4, RPB = 16).

MEP vs. Fraction of Read and Write (F_{rdwr}) Comparison Results

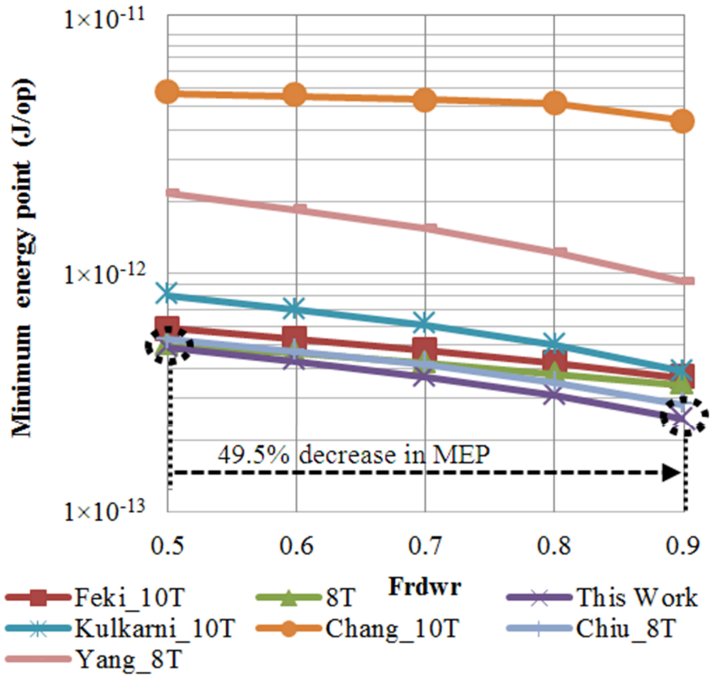


Figure 2.14: Minimum energy point vs. fraction of read and write (F_{rdwr}) for 32KB SRAM (CM = 4, RPB = 16).

To evaluate the effect of F_{rdwr} on MEP, we vary the F_{rdwr} in equation (2.1) to get the plots for MEP vs. Frdwr and MEP V_{DD} vs. F_{rdwr} for CM = 4, as shown in Figure 2.14 and Figure 2.15. Figure 2.14 shows that increasing the F_{rdwr} decreases the weighted MEPs in each bitcell for the SRAM capacity of 32KB that uses RPB = 16. Noticeably, increasing F_{rdwr} causes the change of slope of the MEP vs. F_{rdwr} curve to be almost same except for the Chang's bitcell, which has a much slower change in slope of the curve compared to the other bitcells. Figure 2.14 shows that our work has a 49.5% decrease in MEP, as the F_{rdwr} varies from 0.5-0.9. The reason for the decrease in MEP is because the read EPO of the modeled macro using our bitcell is much lower compared to its write EPO, and weighing more in read EPO reduces the weighted MEP. From Figure 2.15 we can observe that there is no vivid trend of the MEP supply voltage vs. F_{rdwr} curves among the bitcells. However, the MEP V_{DD} remains constant from F_{rdwr} = 0.6-0.8 at 0.45V for Chiu's and our bitcells. Similarly, Yang's and Chang's bitcells also have a constant MEP V_{DD} across F_{rdwr} = 0.6-0.9.

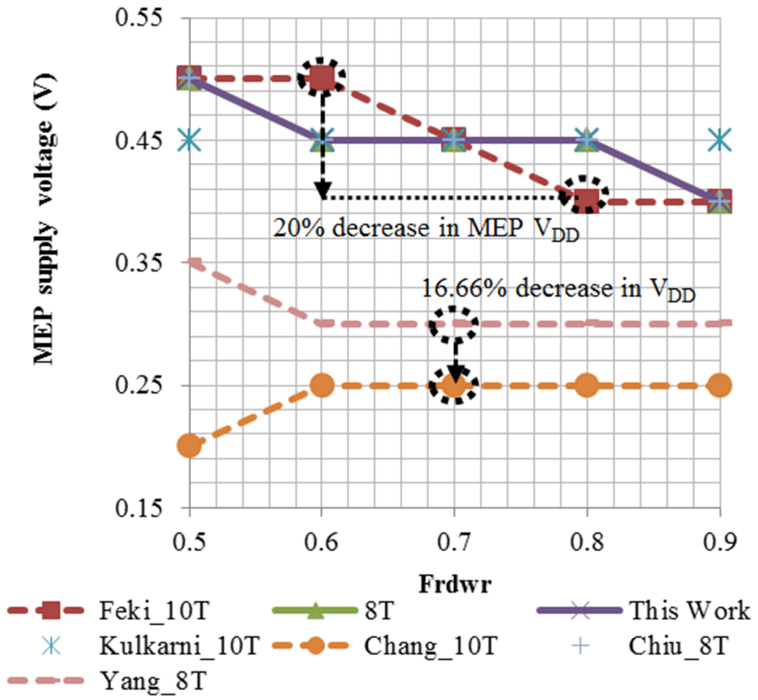


Figure 2.15: MEP supply voltage vs. fraction of read and write (F_{rdwr}) for 32KB SRAM (CM = 4, RPB = 16).

On the other hand, Feki's bitcell shows the MEP V_{DD} decreases linearly by 20% from F_{rdwr} = 0.6-0.8. Noticeably, the MEP V_{DD} of Chang's bitcell is 16.66% lower compared to the MEP V_{DD} of Yang's bitcell from F_{rdwr} = 0.6-0.9. Figure 2.14 and Figure 2.15 indicate that even though Chang's and Yang's bitcells have much higher MEP, due to their lower MEP V_{DD}, it may be suitable for bigger sub-threshold SoCs to have overall lower EPO. To make the overall EPO of the SoC be lower the EPO of logic cells used in the SoC needs to be equivalent or much higher compared to EPO of the sub-threshold SRAM used.

MEP vs. Number of Bitcell Rows per Bank (RPB) Comparison Results

Figure 2.16 depicts the plot for MEP vs. RPB of the bitcells for the SRAM capacity of 32KB using CM = 4. For this experiment, we use a fixed word-width = 32. However, for this experiment, keeping the SRAM macro size fixed at 32KB makes the bank size and number of banks vary with RPB. Thus, if the RPB increases, the bank size increases, too, and the number of banks decreases. We observe that for all the modeled bitcell macros the MEP

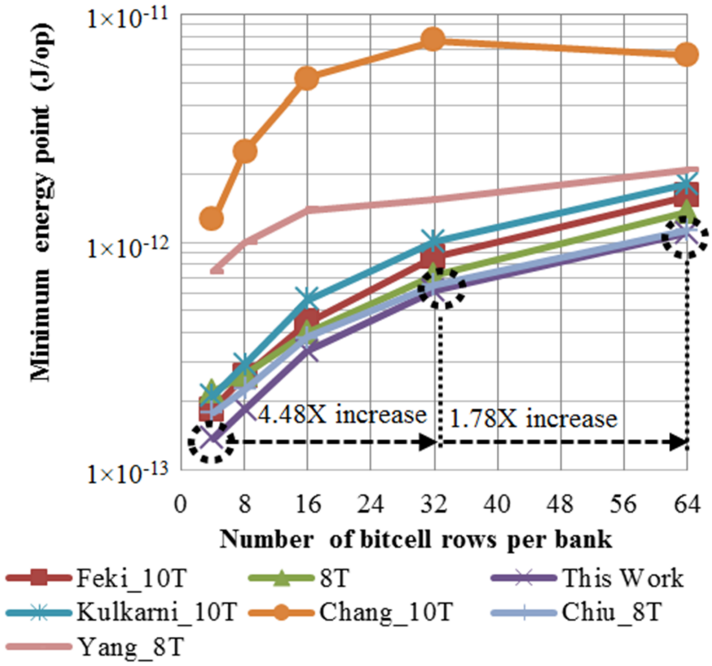


Figure 2.16: Minimum energy point (MEP) vs. number of bitcell rows per bank (RPB) for 32KB SRAMs (CM = 4).

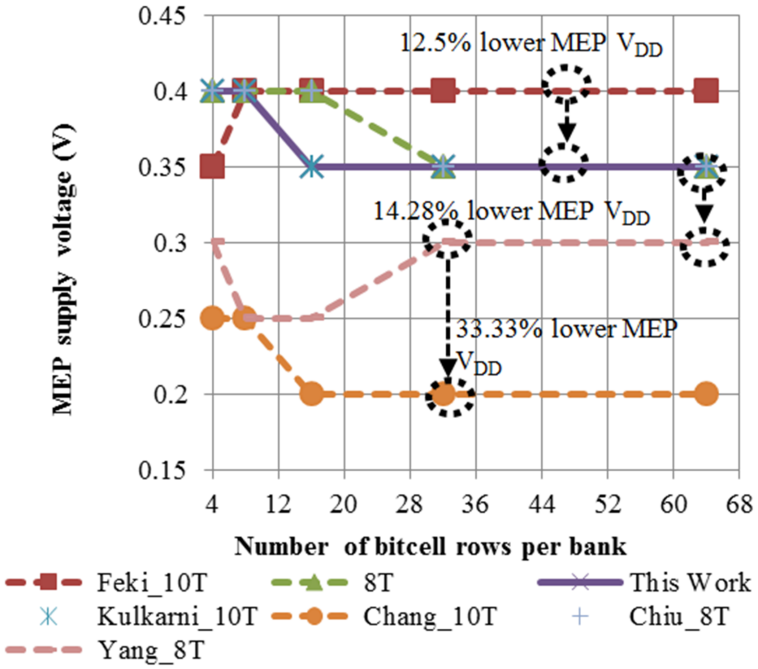


Figure 2.17: MEP Supply voltage vs. RPB of 32KB SRAMs (CM = 4).

increases nonlinearly with RPB. Our work has the minimum MEP among all the bitcells across RPB = 4 to RPB = 64. However, the variation of Chiu's bitcell's MEP becomes comparable to our work from RPB = 32 to RPB = 64. Moreover, the conventional 8T and Chiu's bitcell MEPs are comparable, too, for RPB = 16-32. We document that at RPB = 32 Feki's bitcell has 1.46X, 8T has 1.24X, Kulkarni's bitcell has 1.65X, Chang's bitcell has 6.05X, Chiu's has 2.8%, and Yang's bitcell has 1.9X higher MEP compared to our 9T work. Also, increasing the RPB 8X from RPB = 4-32, and 2X from RPB = 32-64, the MEP of the modeled macro using our bitcell increases 4.48X and 1.78X for, respectively. On the other hand, we observe a trend in the MEP V_{DD} vs. RPB plot, as shown in Figure 2.17. Here, from RPB = 32-64, the bitcells have a constant MEP V_{DD}. Also, across RPB = 16-64, Feki's, Kulkarni's, Chang's, and our bitcell have constant MEP V_{DD}. Thus, comparing the MEP V_{DD}s of state-of-the-art bitcells for RPB >= 32 yields that Chang's bitcell has 33.33% lower MEP V_{DD} compared to Yang's bitcell. Also, for the same criteria Yang's bitcell has 14.28% lower MEP V_{DD} than Kulkarni's, this work, and Chiu's bitcell. We report that our bitcell has 12.5% lower MEP V_{DD} than Feki's bitcell for RPB >= 16.

MEP vs. Word-width Comparison Results

For 32KB SRAMs with CM = 4, we show the plot for MEP vs. the number of SRAM bits in a word or the word-width in Figure 2.18. We keep the capacity of the banks fixed at 512 bits, and vary the word-width, and RPB simultaneously. The RPB decreases with the increase in word-width for keeping the bank size constant in this experiment. Thus, we observe a second order effect in the plot for MEP vs. word-width, as shown in Figure 2.19. Except Chang's and Yang's, for all of the other bitcells the MEP initially decreases reaching a minimum point at some word-width, then increases again. For the 8T and Chiu's bitcell, we observe these minimum MEP points at word-width = 8. On the other hand, at word-width = 16 for Kulkarni's, Feki's, and our bitcell have the minimum MEP point. Noticeably, the variation of MEP of our bitcell across the increase in word-width is much less compared to

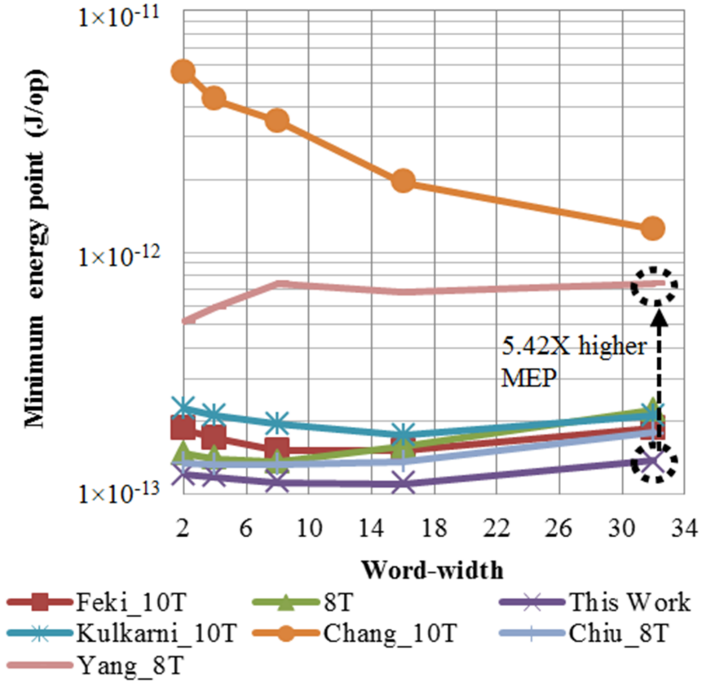


Figure 2.18: Minimum energy point (MEP) vs. word-width (bank size and number of banks kept fixed) for 32KB SRAMs.

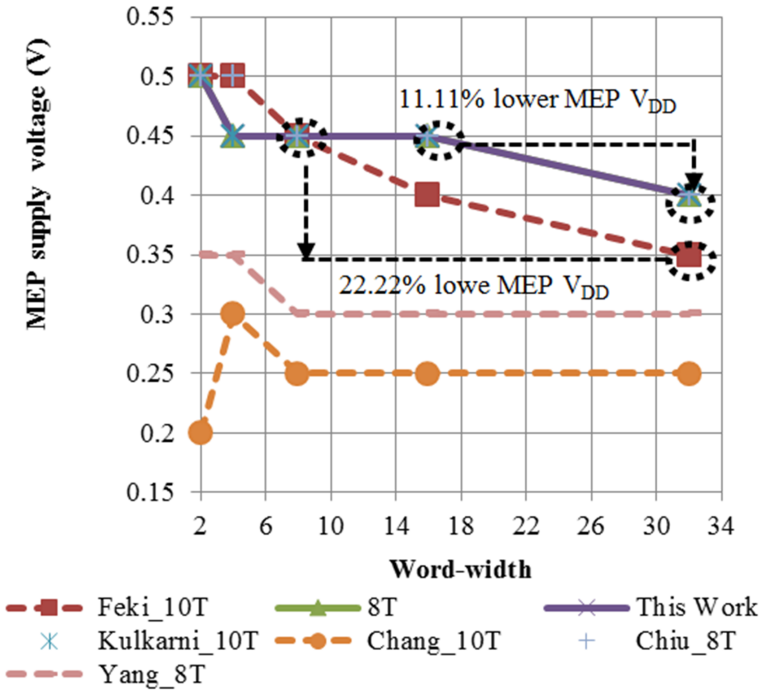


Figure 2.19: MEP supply voltage vs. word-width for 32KB SRAMs.

the Chiu's bitcell. For 32 KB SRAMs, with word-width = 32, we report that Feki's bitcell has 1.35X, the conventional 8T has 1.62X, and Kulkarni's bitcell has 1.55X higher MEP compared to our bitcell. We also report that Chang's bitcell has 9.14X, Chiu's bitcell has 1.3X, and Yang's bitcell has 5.42X (Figure 2.18) higher MEP compared to our bitcell using the same criteria for SRAM capacity and word-width. Thus, for larger sub-threshold memory macros such as 32KB, a higher word-width, and lower RPB is favorable using our 9T bitcell. Figure 2.19 depicts the plot for MEP V_{DD} vs. word-width for all the bitcells. Except Chang's and Yang's bitcell, we observe a trend of decreasing MEP V_{DD} among the bitcells. Feki's bitcell demonstrates a 22.22% reduction in MEP V_{DD} using a word-width increase of 4X from word-width = 8 to word-width = 32. Also, Chiu's and our bitcell yield an 11.11% reduced MEP V_{DD} using a 2X increase in word-width from word-width = 16 to word-width = 32.

MEP vs. Column Mux Comparison Results

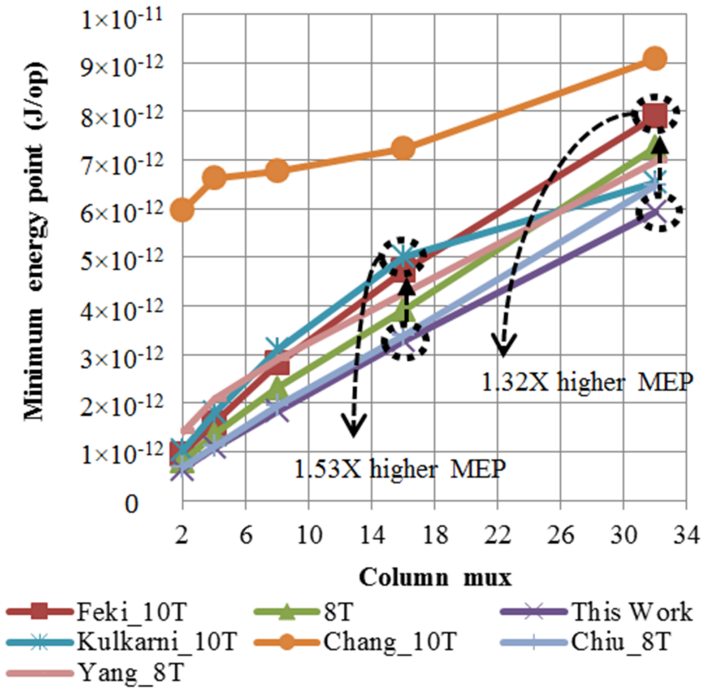


Figure 2.20: Minimum energy point (MEP) vs. column mux (words per row) for 32KB SRAM.

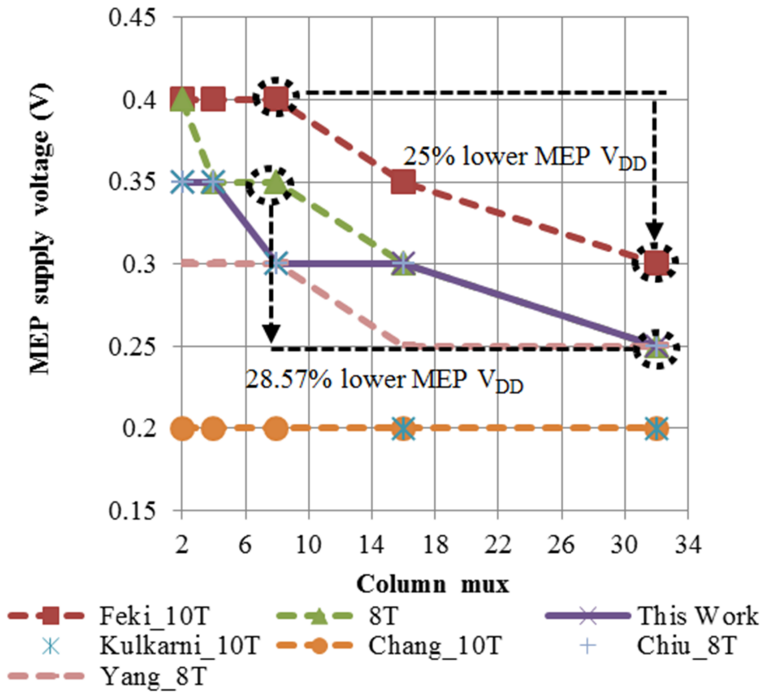


Figure 2.21: MEP Supply voltage vs. column mux of 32KB SRAMs.

We show how the MEP varies with increasing column mux in Figure 2.20. We use RPB = 64 and the word-width = 32, and 32KB SRAM size for this experiment. For keeping the SRAM size constant at 32KB, increasing the column mux (CM) increases the bank size, and decreases the number of banks. Thus, we observe a linear trend of increasing MEP with CM. Nevertheless, Kulkarni's and Chang's bitcells have different trends in different parts of this plot. On the other hand, even though the MEP of our bitcell is comparable to Chiu's bitcell, at CM = 32 our bitcell's MEP is 9.3% lower than Chiu's bitcell's MEP. Moreover, with CM = 32 and 32 KB macro size, we observe that Feki's bitcell has 1.32X, 8T has 1.22X, Kulkarni's bitcell has 9.8%, Chang's bitcell has 1.53X, and Yang's bitcell has 17.36% higher MEP compared to our bitcell. Also, our 9T bitcell gives the lowest MEP across all column mux configurations. Figure 2.20 shows that at CM = 16 Kulkarni's bitcell has 1.53X higher MEP compared to our bitcell. On the other hand, except for Chang's bitcell, with the increase in column mux factor, the MEP V_{DD} reduces among all the bitcells, as shown in Figure 2.21. Note that due to Chang's bitcell having lower MEP V_{DD} below 0.2V in this

memory configuration, we set 0.2V as its MEP V_{DD}. We also document that the MEP V_{DD} decreases by 25% for Feki's bitcell and 28.57% for conventional 8T for increasing the mux factor by 8X from CM = 4 to CM = 32 (Figure 2.21).

MEP vs. SRAM Size Comparison Results

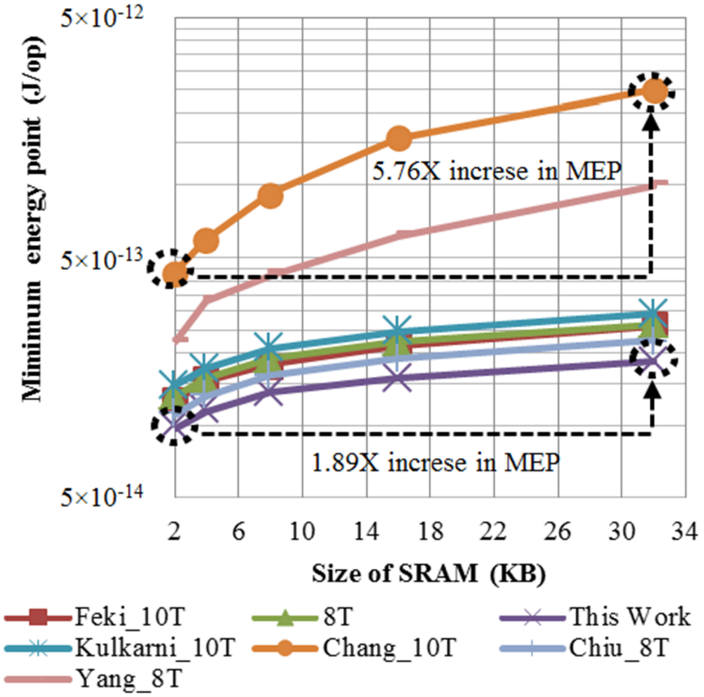


Figure 2.22: Minimum energy point (MEP) vs. SRAM memory size (KB).

We show the variation of MEP with increasing SRAM size for CM = 4 in Figure 2.22. We use a fixed bank size of 1024 bits per bank, RPB = 8, and word-width = 32 in a column mux 4 scenario for this experiment. The number of banks increases with the increase in SRAM capacity due to the reason that the size of the banks remains fixed. Figure 2.22 shows that the MEP of all bitcells increases with increasing SRAM memory size, which we expect, because using a fixed word-width, the leakage EPO increases with the SRAM capacity and the MEP increases. However, for RPB = 8 our 9T bitcell has the lowest MEP across 2-32KB SRAM capacity among all the bitcells, which is due to comparatively lower dynamic energy and leakage current of our bitcell among others that make the MEP for this work lower. For

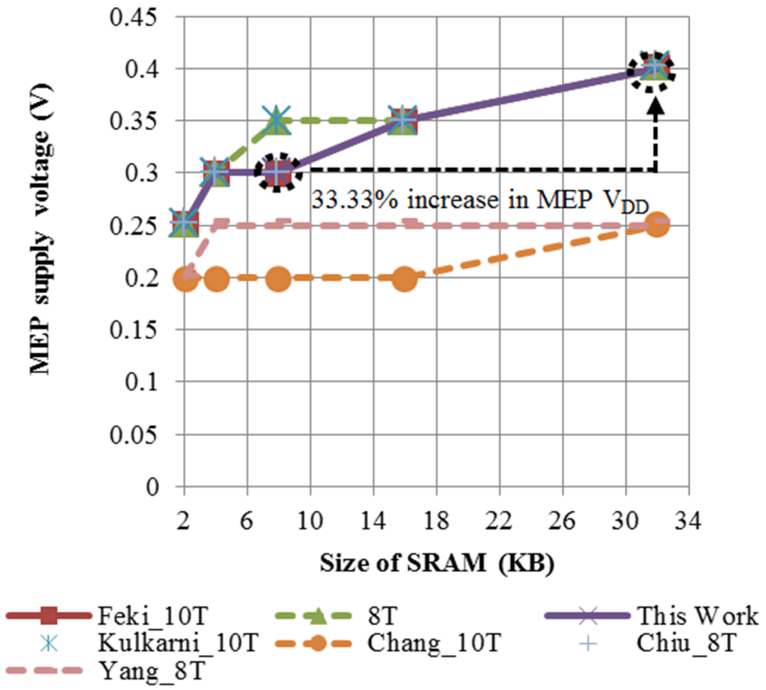


Figure 2.23: MEP supply voltage vs. SRAM memory size (KB).

an 8KB capacity of SRAM, we observe that Feki's bitcell has 1.31X, 8T has 1.39X, Kulkarni's bitcell has 1.51X, Chang's bitcell has 6.75X, Chiu's bitcell has 17.54%, and Yang's bitcell has 3.08X higher MEP compared to our 9T bitcell work. The MEP increases by 1.89X for our bitcell due to an increase in the capacity of SRAM by 16X from 2KB to 32KB. However, the other bitcells' MEP numbers increase to a much higher MEP with the same constraints, such as 2.04X for Feki's bitcell, 1.98X for both Kulkarni's and the conventional 8T bitcell, 5.77X for Chang's bitcell, 2.03X for Chiu's bitcell, and 4.43X for Yang's bitcell. We report the the variation of MEP V_{DD} vs. SRAM macro size, as shown in Figure 2.23. With the increase in the SRAM capacity, the MEP V_{DD} increases for almost all of the bitcells. Also, for Feki's, Chiu's, the conventional 8T, and our bitcell, we observe a 33.33% increase in MEP V_{DD}. On the other hand, for the capacity of 4-32 KB, Yang's bitcell gives a constant MEP supply voltage. Therefore, although the MEP of Yang's bitcell is much higher for various capacities of SRAMs, it can be suitable for larger sub-threshold SoCs, which has comparable logic EPO to SRAMs. Nevertheless, our SRAM bitcell gives lower MEP numbers among the

state-of-the-art bitcells.

2.0.13 System Level Projected Savings for 9T Bitcell

This section computes the system level battery-life savings using our 9T SRAM bitcell compared to the conventional 8T bitcell. The BSN revision 1 had $19\mu W$ of power consumption of which the instruction memory consumes 55.4% (36% dynamic and 64% leakage) and other digital components consume 44.6%. We assume that the instruction memory uses our 9T SRAM bitcell, which is active 100% of the time and the power source is an A1578 (0.76Wh) battery, we estimate the battery-life improvement of 4.40% from 1.10 yrs to 1.154 yrs (about half months of battery-life improvement). On the other hand, if we assume that the instruction memory is only leaking 100% of the time, the battery-life improvement becomes 13.59% from 1.10 yrs to 1.256 yrs, which is a bit more than one and half months of battery-life improvement. The self-discharge rate used in this calculation for the A1578 battery is 10% per month. Using the LIR2032 battery, the corresponding battery-life savings numbers are 9.36% and 30.70%. The self-discharge rate used for LIR2032 calculation is based on Table 1.3. On the other hand, using a non-rechargeable SR416SW battery the corresponding battery-life improvements are 11.05% and 37.40%, and the battery replacement time increases to 30 days and 37.13 days from 27.02 days, respectively. The equations used to calculate the battery-life using some self-discharge assumptions are derived in Appendix A for the SR416SW Silver Oxide, LIR2032 Lithium-ion, and A1578 Lithium-ion Polymer batteries.

2.0.14 Conclusions

Among the state-of-the-art ([26] [27] [28] [29] [36]) bitcells, including the conventional 6T bitcell, our bitcell [36] obtains the lowest read energy from 0.25-0.5V supply range. Also, this work has the lowest write energy among the state-of-the-art bitcells in the 0.35-0.5V supply range and the second lowest leakage current in the 0.1-0.5V range. This work improves the energy and leakage numbers at sub-threshold supplies trading off a penalty in SRAM timing.

Our 9T bitcell shows promises to have the lowest minimum energy point across F_{rdwr} = 0.5-0.9 for 32KB capacity. Also, this work gives the lowest MEP variation for 32KB SRAMs across different values of rows per bank (RPB), ranging from RPB = 4-64. However, Chiu's bitcell has comparable MEP values using 32KB capacity for RPB >= 32. This work also demonstrates that, by varying word-width and RPB, with fixed bank sizes and the number of banks, many state-of-the-art bitcells have a minimum in the MEP data close to word-width = 8 and 16. The minimum MEP happens because of a second order effect resulting in varying two of the design metrics simultaneously, such as word-width and RPB. Moreover, we report that our bitcell gives the lowest MEP values for word-width = 2-32. However, as we exclude the area comparison from this work, our 9T bitcell may have higher area penalty. We show that MEP vs. column mux trends are linear for most of the bitcells, and for the mux factor of 2-32, our work gives the lowest MEP values. Also, our bitcell provides the lowest MEP numbers across SRAM capacities using RPB = 8. Nevertheless, in a large sub-threshold SoC with comparable SRAM and logic energy per operation, Yang's and Chang's bitcells give reduced MEP supply voltages, and those may qualify as the best fit from the standpoint of minimum energy per operation metric. In the estimation of system level battery-life savings, our 9T bitcell improves battery life by 11.05% and 37.40% for 100% active and 100% leaking cases, respectively, using the SR416SW battery compared to the conventional 8T bitcell used in the BSN revision 1 SoC. Thus, for energy-constrained IoT SoCs, such as biomedical SoCs those have critical requirements for battery-life, our 9T half-select-free sub-threshold SRAM bitcell offers lower energy numbers and the lowest MEP values in read and write operations across SRAM design knobs.

2.0.15 A Low-Energy Peripheral Read Architecture for Body Area Sensor Node (BSN) SRAMs

The conventional 6T bitcell suffers from robustness issues due to poor write-ability, readability, and read-stability in sub-threshold supplies. Thus, the minimum operating

voltage (V_{MIN}) of 6T bitcell across process variation is higher, which leads to write or read failures if operated in the sub-threshold region. The conventional 8T SRAM bitcell fixes most of the issues of 6T bitcell for sub-threshold operations. However, in bitline interleaving scenario improving the write-ability using a peripheral assist, such as boosted wordline, creates a row half-select issue, which degrades the read-stability of 8T bitcells, also. Thus, a write assist may lead to the increase of V_{MIN} of 8T SRAMs, which increase the active energy dissipation, too. Therefore, for quadratic energy savings, voltage scaling in deep sub-threshold supplies faces a bottleneck due to the row half-select issue in 8T bitcells. Using a read-stability assist such as wordline under-drive or boosted V_{DD} can resolve the row half-select problem in 8T SRAMs causing degraded write-ability and increasing area penalty. Moreover, circuits for some read assists, such as V_{DD} boosting, may have higher energy and area penalty in smaller capacity SRAMs, such as 2-4KB. Furthermore, alternative sub-threshold bitcells incur much higher area penalty for core array and overall SRAMs. An architecture technique named the writeback scheme [33] allows avoiding the half-select problem in sub-threshold supplies. On the other hand, applications are prone to do localized read and write operations from memory due to the spatial and temporal locality of references. Thus, we employ a low energy read (LER) operation to mitigate read energy reduction in sub-threshold or near-threshold SRAMs for applications that require sequential read operations. We use the writeback scheme to implement a single cycle write-after-read (WAR) operations, which supports the LER operations. Thus, in this section, we investigate a d compare architecture level techniques to minimize energy consumption in sub-threshold SRAMs.

2.0.16 Issues in State-of-the-art Alternative Sub-threshold Bitcells

State-of-the-art Kulkarni et al. work [26] show that the Schmidt-Trigger-based bitcell can operate at 160mV in the deep sub-threshold region. However, the Monte Carlo (MC) data suggests that the $\mu - 3\sigma$ read static noise margin (RSNM) of the Schmidt-Trigger bitcell lies

between 50mV to 0mV and $\mu - 3\sigma$ hold static noise margin (HSNM) is lying closed to 100mV. The work also indicates that the $\mu + 3\sigma$ V_{MIN} of the bitcell lies between 350-400mV. Thus, across process variation using 400mV of supply voltage the 3σ worst-case V_{MIN} suggests memory failures if the Schmidt-Trigger SRAM operates below the V_{DD} of 350mV. Another sub-threshold bitcell work Chang et al. [27] has a poor 3σ worst-case RSNM and HSNM. For L=120nm, using a 300mV V_{DD}, the $\mu - 3\sigma$ write static noise margin (WSNM) is closed to 100mV, and $\mu - 3\sigma$ HSNM is 35mV. Using L=80nm the $\mu - 3\sigma$ RSNM becomes negative for Chang's bitcell, thus inoperable in sub-threshold supplies. On the other hand, the Reddy et al. work [40] documents the RSNM distributions of the proposed bitcell vs. the conventional 6T bitcell. The plots indicates that at 400mV $V_{DD}$the worst-case $\mu - 3\sigma$ RSNM is around 20mV. Thus, across process variation, there can be read failures occurring in Reddy's bitcell. Thus, we infer from the state-of-the-art bitcell works [26] [27] [33] [40] [41] [42] that lowering the V_{DD} below 400mV will cause write, and read failures in most of the published SRAM bitcells limited by their worst-case WSNM, RSNM, HSNM, and data retention V_{MIN}. Thus, V_{DD} scaling using alternative bitcell may not be a suitable solution to lower energy consumption in ultra-low power BSN SRAMs, and architectural techniques may help to reduce energy consumption in this regard. The next section discusses the state-of-the-art circuit and architectural energy reduction techniques for SRAMs.

2.0.17 State-of-the-art SRAM Energy Reduction Techniques

There exist state-of-the-art circuit and architectural techniques to lower SRAM energy, such as the floating bitline [43], bitline amplitude limiting [44], segmented virtual grounding [45], etc. We briefly describe some of the methods as follows.

Floating Bitline Scheme

Authors in [43] propose a disturb mitigation scheme, which supports low-power as well as low-voltage operations for SRAMs in a deep sub-micron technology. The authors show

that the proposed scheme involves a floating bitline technique, which employs a low-swing bitline driver that reduces the active leakage and power at the FF corner by 33% and 32%, respectively. Using the scheme the work also reports reducing active power by 47% and 60% at the CC and SS corners. Also, the authors show that the proposed scheme is 35% better in saving active energy compared to the conventional writeback scheme.

Bitline Amplitude Limiting Scheme

On the other hand, the work [44] proposes a bitline amplitude limiting scheme, which achieves a 26% total energy reduction at 0.5V V_{DD} trading off 7% of penalty in speed, and less than 2% of penalty in the area. This scheme uses a bitline amplitude limiter, which suppresses the excess bitline amplitude for lowering dynamic energy automatically. Using simulated results, the authors report having 20% and 29% reduction in dynamic and leakage energy, respectively. The authors implement the circuit in a 40nm technology and using the proposed scheme achieves a measured 19% energy reduction.

Segmented Virtual Grounding Scheme

The authors in [45] propose a novel architecture to lower SRAM's dynamic and static power consumption. The work uses a segmented virtual grounding scheme for the SRAM bitcells, which reduces the leakage current employing body bias by that increases the threshold voltage of the transistors. This work reduces the write and read energy by decreasing the swing in the bitline voltage. The authors report that the work reduces the read and write energy consumption by 44% and 84%, respectively, using a 130nm CMOS technology. The work also achieves a 15X leakage improvement relative to a conventional scheme.

Hierarchical Bitline Scheme

Moreover, the work [46] proposes circuit techniques, which reduces the SRAM energy consumption without V_{DD} scaling. The authors show an energy efficient hierarchical bitline

scheme that saves energy consumption by lowering the overall bitline precharge energy. The authors also show an energy efficient offset-canceling and a robust timing generation circuit to cope with the process variation. The work implements the proposed circuits in a 28nm 4Mb SRAM, which has a 7% area penalty. The authors report a 60% dynamic energy reduction and 10% leakage energy reduction using these proposed schemes.

Dynamic Voltage Management Scheme

Furthermore, the authors in [47] present a scheme for dynamic voltage and frequency control of a 256x64 SRAM macro, which reduces the active and standby energy. The scheme monitors the external clock frequency and adapts the supply voltage and the body bias, which lowers the energy consumption. The method achieves 83.4% and 86.7% reduction in energy in the active and standby modes, respectively. The work proposes an energy replica scheme that monitors the energy of the subsystem using its dynamic voltage management method, too.

We observe from prior works in sub-threshold SRAM bitcells that the poor robustness of bitcells below 400mV V_{DD} limits the reduction of dynamic energy consumption using V_{DD} scaling. Also, the existing schemes to mitigate the energy or power, fail to provide a 2X savings. Inspired by the DRAM timing in which for each Row Access Strobe (RAS) multiple Column Access Strobe (CAS) triggers, we investigate a novel architecture and compare our work with the state-of-the-art schemes to reduce energy consumption in sub-threshold SRAMs.

2.0.18 BSN SRAM Revision 1 and Scope of Improvement

The revision 1 of the body area sensor node (BSN) system on chip (SoC) requires a 1.5KB instruction memory or ROM and a 4KB data memory, those we implement using 8T sub-threshold SRAMs. The functionality of the instruction memory is to store 12-bit instructions for the execution of the digital power management block (DPM) and the peripheral interface

controller (PIC) processor. The instruction memory programs at the boot-up time using a scan chain, which we later use it for reading instructions only. On the other hand, the data memory (DMEM) operates as a first-in-first-out (FIFO) buffer. During the data acquisition of bio-medical signals, the data stream into the DMEM for buffering and, once the FIFO fills up, it resets the address to "0." As soon as the BSN SoC detects the atrial fibrillation (AFib) event, the content of the data memory transmits using the wireless transmission driven by the on-chip radio transmitter. To improve programmability, we plan a revision 2 of the BSN design that includes an openMSP430 architecture [48]. With this new architecture, the memory timing requires an update, and the FIFO buffers need a new architecture to become truly random access memories. Due to the usage of 8T SRAM bitcell, the SRAM read has no issues with the existing architecture; however, it becomes a problem for the write operation. In revision 1 of the BSN chip, the write data is stored in a write buffer of width eight and, once the buffer gets full, it writes into the SRAM. The revision 2 specification requires the word address to be incremented by one in a bitline interleaving scenario during each successive write operation, which leads to the row half-select issue using a boosted wordline write assist for improving the sub-threshold write-ability. Thus, three possible solutions could solve the revision 2 specification requirements, as follows. The possible solutions are 1) an alternative bitcell topology that has improved read-stability or RSNM, 2) a read-stability assist such as wordline underdrive (WLU) that improves the 8T RSNM, or 3) an architecture that supports the writeback or write-after-read (WAR) scheme. Although an alternative bitcell can have an improved read-stability, it usually has a large silicon area penalty, due to the usage of additional transistors compared to the 8T bitcell. On the other hand, simulation results [20] have shown that the leakage consumes 64.7% [20] of the read energy in BSN SoC. Thus, reducing the leakage energy consumed by the bitlines during the read operation reduces the overall SRAM energy. Therefore, applying a WLU read-stability assist could be a potential solution to reduce the bitline leakage. However, a WLU assist degrades the worst-case write margin to 100mV and scaling the SRAM supply voltage below 0.5V increases the probability

of inducing potential write failures. Nevertheless, the write-after-read (WAR) or the writeback scheme leverages the separate read-path of 8T bitcell using the two transistor read buffer, which prevents read-stability issues. During the WAR operation, the SRAM first reads the corresponding word, and it stores the word in an intermediate latch. Meanwhile, the data to be written bypass along with the other words stored in the intermediate latch to the write bitlines through a set of multiplexers, such that the order of the words stored in the memory row remains the same. Thus, the active write bitline columns write the new data, and the old data write back into the half-selected bitcells. This way the WAR scheme ensures that the data stored in the half-selected columns remain undisturbed. However, this scheme pays an additional timing penalty for the WAR operation and some hardware overhead.

2.0.19 BSN SRAM Revision 2 SRAM Architecture Using Low Energy Read Peripheral Architecture

To achieve the BSN revision 2 design we update the existing BSN revision 1 SRAM using the single cycle WAR control logic, low energy read (LER) peripheral logic, 16-bit output flip-flop, 128 to 16-bit bus-interface-logic (BIL), and input flip-flops. Figure 2.24 shows the architecture of the updated SRAM for BSN revision 2. The function of the 128-bit intermediate latch (Figure 2.24) in the SRAM is to latch all 8 words (16-bits each) in a normal read operation. If the user reads from the same row in two or more consecutive read operations, the LER logic signals the read wordline (RWL) automatically not to toggle, and the SRAM reads from the intermediate latches instead. Figure 2.25 portrays the BIL for the WAR operation. On the other hand, the burst-enable logic scans for any previous usual read operation and, if it follows by another read request, it issues the LER operation unless the row address has changed. We show the burst-enable-logic in Figure 2.26. With this scheme, for each normal read operation in SRAM, a user can have seven distinct LER operations without considering repetitive reads in the same address. We investigate the dynamic energy savings by not switching RWL, row and bank decoders in the LER operations.

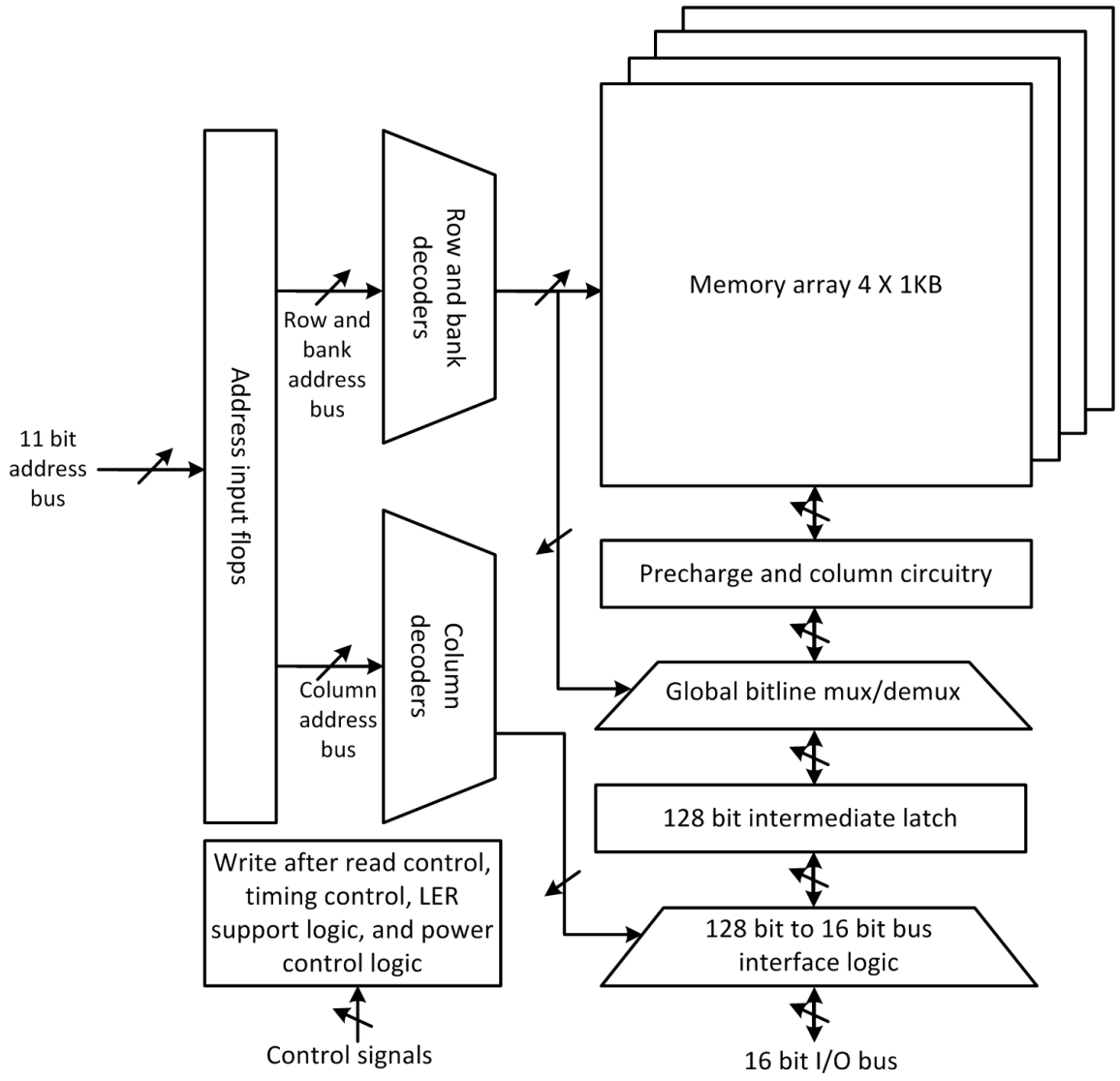


Figure 2.24: Architectural block diagram of 4KB sub-threshold BSN SRAM.

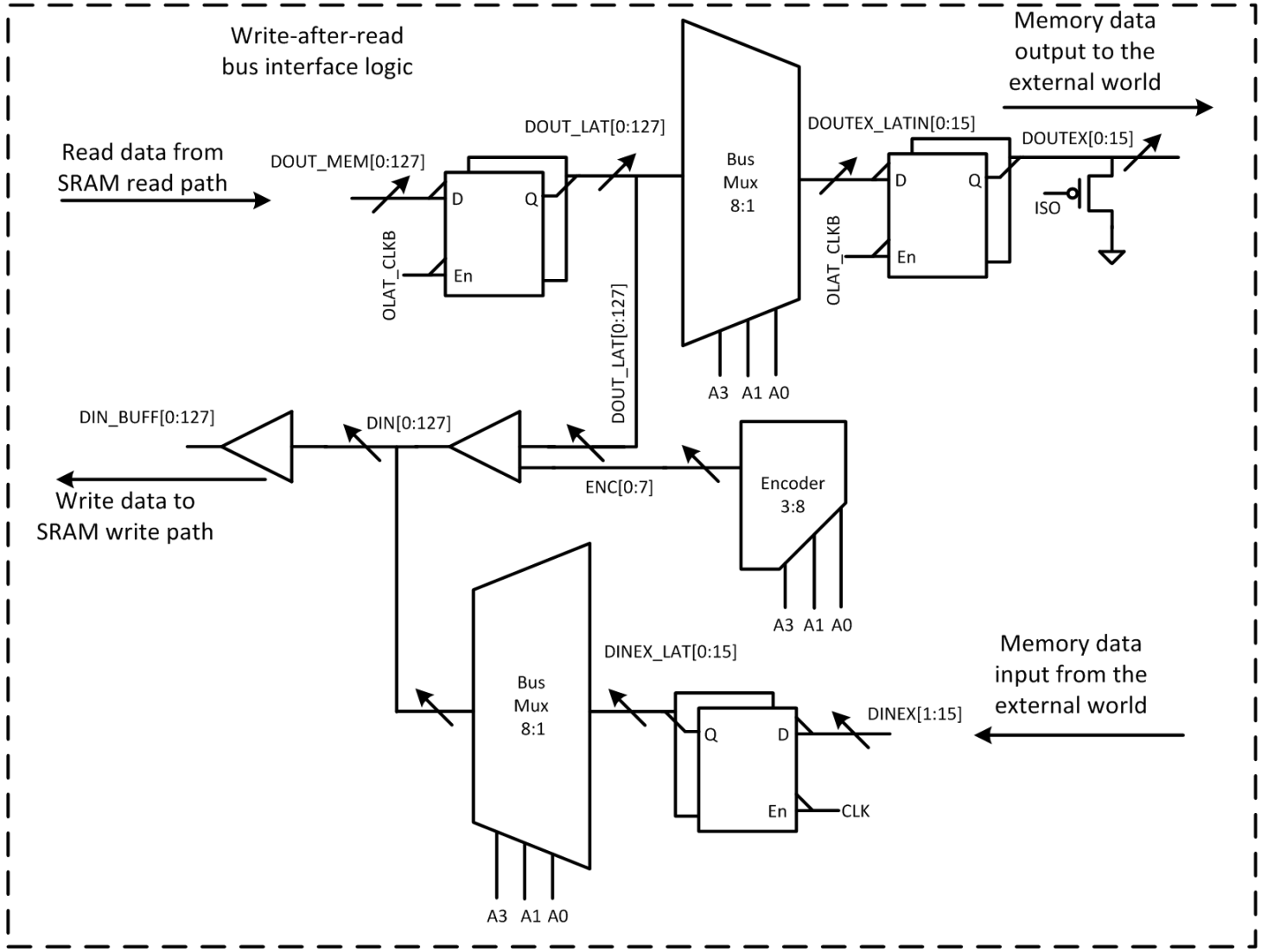


Figure 2.25: Circuit diagram of the write-after-read bus-interface-logic circuit of 4KB sub-threshold BSN SRAM.

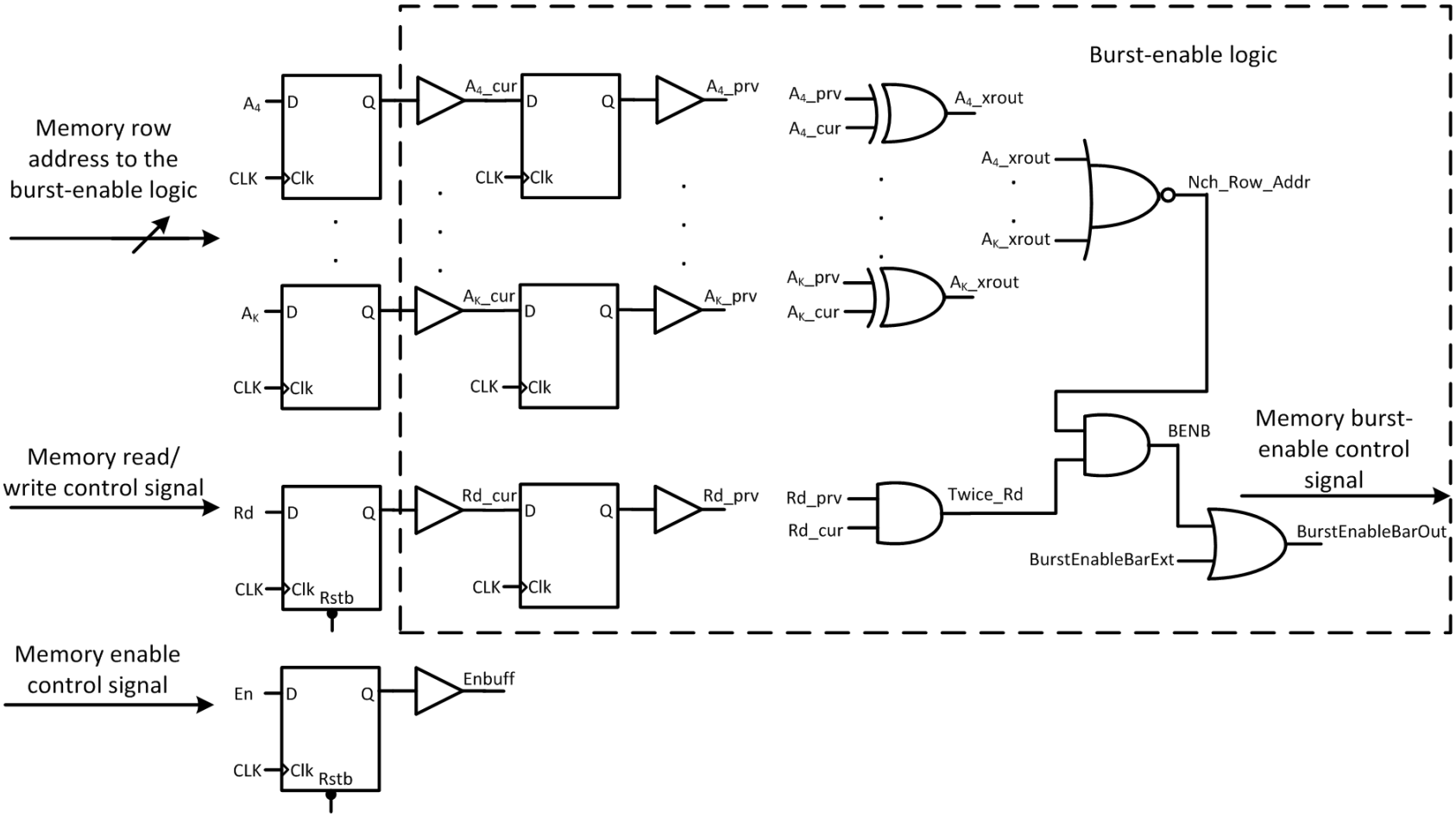


Figure 2.26: Circuit diagram of the burst-enable logic of 4KB sub-threshold BSN SRAM.

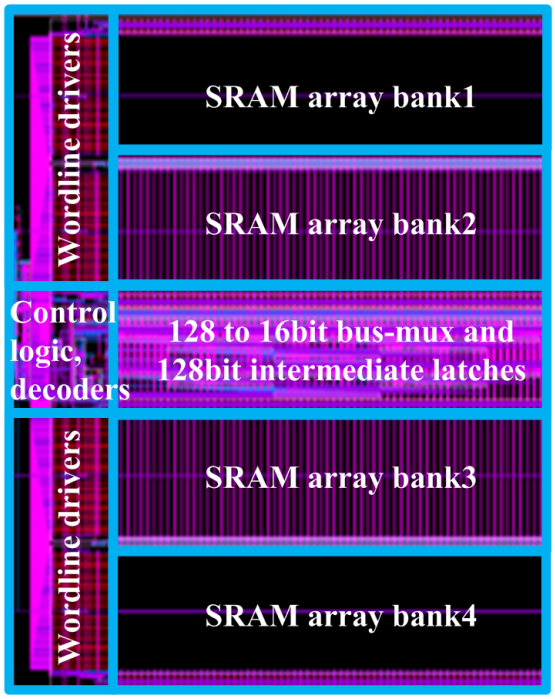


Figure 2.27: Annotated layout of the 4KB sub-threshold BSN SRAM.

Moreover, we implement the single cycle WAR operation by pulsing the RWL and write wordline (WWL) in the same cycle that uses pulse generator circuits. We also incorporated three-bit WAR margin control pins for sub-threshold margin variations to control RWL and WWL pulse widths, externally. Furthermore, with this scheme, we investigate the energy savings or penalty of implementing single cycle WAR operations. We report the worst-case maximum operating frequency of the revision 2 SRAM at SS_0.5V_27C is 1.03MHz. As the sub-threshold SRAM specification requires the SRAM to work in 200 kHz at 0.5V with 27C, we had more than sufficient margin to play with timing. We implement the SRAM macro in a commercial 130nm technology and simulate the modeled pre-layout netlist with HSIM using 100% SPICE accuracy. The block diagram of the 4KB sub-threshold SRAM is the same as the Figure 2.24. Figure 2.27 portrays the annotated layout of the 4KB sub-threshold SRAM macro.

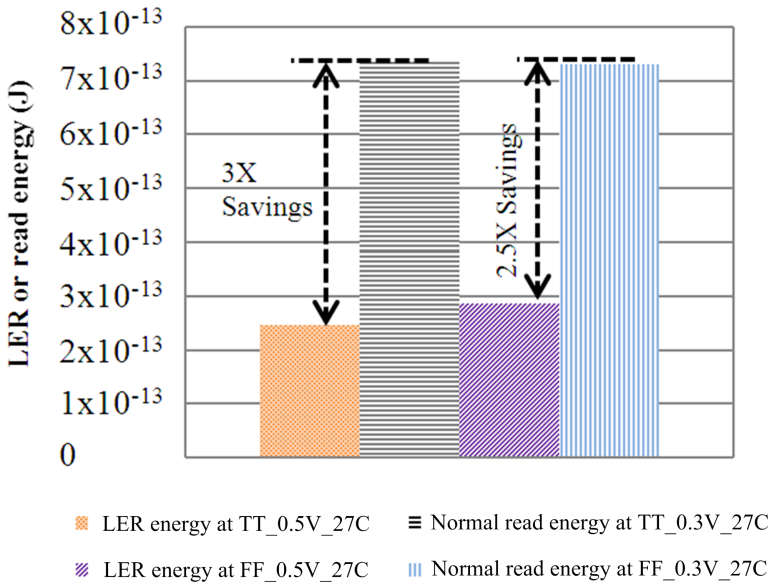


Figure 2.28: Comparison of normal read energy at 0.3V_27C with LER energy at 0.5V_27C in 4KB sub-threshold BSN SRAM.

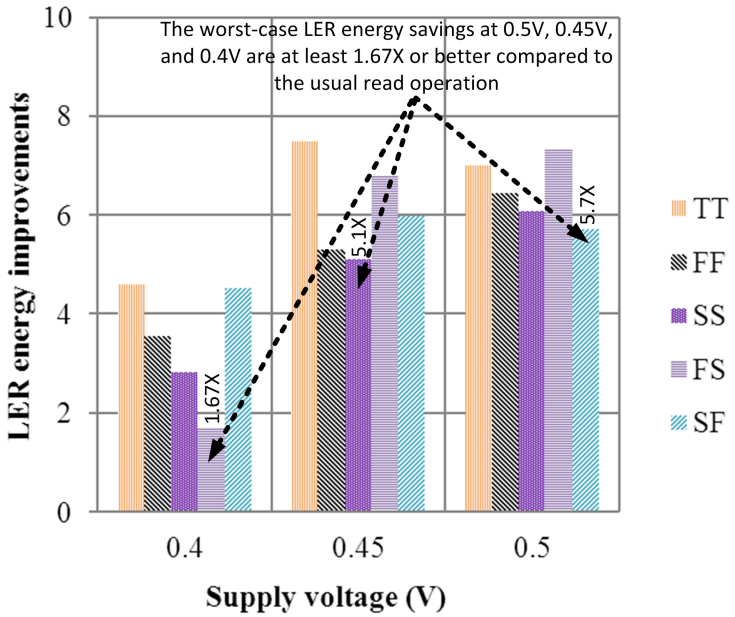


Figure 2.29: Comparison of the energy improvement ratio of LER scheme to normal read operation vs. supply voltage at 27C in 4KB sub-threshold BSN SRAM.

Table 2.2: Comparison of energy/power savings with prior arts.

Works	**Energy/power savings**
SRAM read-assist scheme [42]	21.3%
Low-energy disturb mitigation scheme [43]	32%
Bitline amplitude limiting (BAL) scheme [44]	26%
Segmented virtual grounding architecture [45]	44%
Hierarchical bitline without voltage reduction [46]	60%
This work LER energy savings	82.45% @ 0.5V SF 27C, 80.39% @ 0.45 SS 27C, 40.11% @ 0.4 FS 27C

2.0.20 Results

We portray the data of normal read energy and LER energy in two different supply voltages, as shown in Figure 2.28. We observe that at 0.5V 27C in the TT process, the LER energy is 3X lower compared to the usual read energy at 0.3V 27C in the same process. Moreover, at 0.5V 27C in the FF process, the LER energy is 2.5X lower than the normal read energy at 0.3V 27C in the same process. Thus, using LER scheme operating sub-threshold SRAMs in near sub-threshold V_{DD}s such as 0.5V is profitable from energy savings standpoint. Furthermore, the LER scheme avoids V_{DD} scaling, which may cause write and read issues in sub-threshold SRAMs. We portray the bar plot of the ratio of the usual read energy to the LER energy, as shown in Figure 2.29. Therefore, we document that the worst-case LER energy is 5.7X lower in SF_0.5V_27C PVT, 5.1X lower in SS_0.45V_27C PVT, and 1.67X lower in FS_0.4V_27C PVT, compared to the energy in a normal read operation.

We further notice that the revision 2 SRAM's worst-case LER energy improvement, compared to the revision 1 SRAM's read energy, is 6X at the SS_0.5V_27C process-voltage-temperature (PVT), and the best improvement is 7.4X at FS_0.5V_27C PVT. However, the worst-case normal read energy in revision 2 increases by 45% compared to the revision 1 read energy numbers at the SS_0.5V_27C PVT. Note that other than the TT and FS corners,

using the same V_{DD} and temperature, the revision 2 SRAM's usual read energy is always higher compared to the revision 1 SRAM's read energy. Nevertheless, the WAR energy improvements in revision 2 SRAM compared to the cumulative write and read energy in revision 1 SRAM design at 0.5V_27C are 2.5X, 2X, and 1.67X at FS, FF, and TT processes respectively. On the other hand, at the SS and SF processes using the same V_{DD} and temperature, the revision 2 SRAM's WAR energy increases 20% and 25% compared to the cumulative write and read energy in the revision 1 SRAM. The layout area overhead increases by 7% compared to revision 1 layout using our scheme in the revision 2 SRAM, which we can minimize by optimizing the floorplan and sub-component layouts. For our scheme, the worst-case standby leakage current penalty is 3% at the FF_0.5V_27C PVT, and the standby leakage current becomes 17% less compared to the revision 1 SRAM design for the best case scenario. We compare the LER scheme with state-of-the-art energy or power reduction methods in Table 2.2. We fabricate an LER derived design in [49], which achieves a 6.24pJ/access for battery-less IoT SoCs.

2.0.21 System Level Projected Savings for LER Scheme

This section computes the system level battery-life savings for LER Scheme compared to the conventional read scheme in BSN revision 1 instruction memory. The BSN revision 1 SRAM has $19\mu W$ of power consumption of which the instruction memory consumes 55.4% (36% dynamic and 64% leakage) and other digital components 44.6%. We assume that the instruction memory uses our LER scheme for sequential reads, which is active 100% of the time and the power source is an A1578 (0.76Wh) battery. Thus, we estimate the battery-life improvement of 6.37% from 1.10 yrs to 1.17 yrs (about a month of battery-life improvement). These projections assume a self-discharge of 10% per month for the A1578 Lithium-Ion Polymer battery. Using a rechargeable Lithium-Ion LIR2032 battery, the battery-life improves by 13.73% from 0.69 yrs to 0.78 yrs. The self-discharge rate used for LIR2032 calculation is based on Table 1.3. On the other hand, using a non-rechargeable

SR416SW battery, the battery-life improves by 16.30% and battery replacement time increases to 31.4 days from 27.02 days. We derive the equations used to calculate the battery-life using self-discharge assumptions in Appendix A for the SR416SW Silver Oxide, LIR2032 Lithium-ion, and A1578 Lithium-ion Polymer batteries.

2.0.22 Conclusions

The low energy read (LER) peripheral architecture allows sub-threshold SRAM operations without changing the core array, which requires minimal changes in the existing periphery and I/Os. The LER architecture is independent of the choice of SRAM bitcell that lowers the read energy for addresses spatially located or temporally accessed in the locality of the same SRAM row. The architecture reuses the sub-components of a single cycle write-after-read (WAR) operation, where we have controls to the WAR margins across PVT variation. This work has a 7% area, 3% worst-case standby leakage, and 25% worst-case WAR energy penalty compared to the existing design. We achieve a maximum of 5.7X LER energy improvements for the worst-case in kHz frequencies at 0.5V supply voltage for our 4KB sub-threshold instruction memory. Using this scheme one can have seven distinct LER operations per one normal read operation. Our LER method could improve IoT system level single charge battery-life by 13.73% from 0.69 yrs to 0.78 yrs, which uses a Lithium-ion LIR2032 battery.

2.0.23 Acknowledgements

These projects were supported in part by NVIDIA through the DARPA PERFECT Program and by the NSF NERC ASSIST Center (EEC-1160483).

Chapter 3

Read-Write Peripheral Assists for Improving the 6T SRAM V_{MIN}

3.0.1 Motivation

Energy-constrained IoT applications have stringent energy requirements that force the logic and the SRAM to operate at lower supply voltages. As energy has a quadratic relation with the supply voltage, scaling it down reduces energy quadratically for logic and SRAMs. However, across process variation, lowering supply voltage poses a risk for SRAMs from write-ability, readability, read-stability, and soft error rate (SER) [19] standpoints. Due to poor Write Margin (WM), Read Static Noise Margin (RSNM), and a degraded half-select issue in a bitline interleaving scenario, the 6T SRAM in a bulk technology is vulnerable to operate at near-threshold or sub-threshold voltages. Although alternative half-select free bitcells operate in near-threshold supplies, they come with significant area overheads for SRAM core arrays. In scaled FinFET technologies, these issues amplify and the smallest area high-density (HD) bitcell suffers from poor write-ability and read-stability issues at nominal supply voltages in the worst-case corner. Thus, improving the write and read operations for FinFET SRAMs require techniques, such as sizing the bitcell devices, choosing an alternative bitcell topology, applying peripheral assist, etc. Changing the topology or increasing the

sizing can improve the write and read operations of the 6T HD FinFET bitcells, but are not an option, as they affect the overall density of the bitcells by trading off the SRAM area and its silicon cost. Moreover, due to fixed design knobs in FinFET technology, such as the quantized width and length of FinFETs, making the worst-case 6T HD bitcell work at nominal voltage is quite challenging. Modern FinFET technologies allow the SRAM designers to use the number of fins as the only design knob to size the 6T bitcell, which increases the SRAM area, too. Peripheral assist remains the only design knob to improve the 6T HD SRAM minimum operating voltage (V_{MIN}) and silicon yield corresponding to write-ability, readability, and read-stability metrics trading off energy. Although SRAM metric such as data retention voltage (DRV) in FinFET technologies influences the DRV V_{MIN}, it is usually much lower compared to the write and read V_{MIN} and does not influence by the transient peripheral assists. Assists are widely employed in reducing the SRAM V_{MIN} in energy constrained designs. Recent works demonstrate single and multiple combined peripheral assists, which shows promises for lowering the HD 6T FinFET V_{MIN}. Nevertheless, no literature compares these solo and combined assists for write and read SRAM design metrics for HD 6T FinFET SRAMs. Thus, we investigate to reveal our findings in this chapter, whether applying a type of solo, or multiple combined peripheral assists can improve the V_{MIN} of the FinFET HD 6T bitcell below the nominal supply voltage across process and temperature variations. We further investigate whether applying combined peripheral assists could improve V_{MIN} and yield in 6T SRAMs in bulk technology to work near-threshold or sub-threshold supplies. Thus, a wide-scale supply voltage operability of 6T SRAM could be possible saving silicon cost for a multitude of IoT applications.

3.0.2 SRAM Write and Read Design Metrics Revisited

The SRAM write-ability defines as the ability to perform write operations at a supply voltage, frequency, and temperature with statistical confidence cross process variation. Similarly, the SRAM readability defines as the ability to execute read operations across design

parameters with statistical confidence. The read-stability defines as the ability to prevent any change in the existing content of the SRAM bitcells during a read operation with statistical confidence across design parameters. There are two types of quantifying metrics for SRAM operations such as static metric and dynamic metrics. In static metrics, we measure the DC characteristics of the SRAM bitcell and define DC metrics for write-ability, read-stability, data retention capability, etc. operations across process, voltage, and temperature (PVT) variations. DC metrics are widely popular for bitcell evaluation and comparison because they are easy to calculate. The DC metrics relate to statistical distributions due to the effects of PVT variations. On the other hand, the dynamic parameters, such as dynamic write and read V_{MIN}, write and read time, read differential, etc. are different measures of the dynamic write and read operations, which accurately represent the transient SRAM operations.

For the quantification of write-ability, we use write-margin (WM) [32] to represent the degree of ease to write into an SRAM bitcell. There exist two definitions of WM, such as wordline (WL) type and bitline (BL) type WMs. In a WL type WM, BL or bitline-bar (BLB) voltage is kept in a DC condition to V_{DD} or V_{SS} for writing data while the WL sweeps. The wordline voltage, at which the internal nodes of the bitcell flip, is negated from the V_{DD}, which gives the WL type WM. Similarly, if WL and BLB are kept at V_{DD}, and the BL sweeps, the BL voltage that corresponds to the internal nodes of the bitcell to flip is the BL type WM. Having a higher WM is desirable, which represents how easily one can write to an SRAM bitcell.

Another way of characterizing the write operation is through N-curve [50] [51] [52], which measures the SRAM write-ability using DC voltage and current. The N-curve has measurable metrics, such as write trip current (WTI) [50] and write trip voltage (WTV) [50]. Here, the WTI is the peak current in the negative direction between the third and second zero-crossings of the N-curve and the WTV is the difference of voltages between the third and second zero-crossings. However, authors in [50] show that both WTI and WTV correlate poorly with the write-ability of the 6T SRAM bitcell.

We can measure SRAM readability using DC metrics such as the DC cell current (I_{cell}) in a read operation. On the other hand, hold, and read-stability operations use a different parameter called static noise margin (SNM) to quantify the hold-stability and read-stability. For hold operation, the metric is called the hold SNM (HSNM), in which the WL remains off, and no write or read activity happens. In a read operation, the read-stability metric is called the read SNM (RSNM), in which we assume that the WL is turned on for an infinite time to stress the internal nodes of the bitcell. Thus, the RSNM will always be lesser than HSNM. HSNM and RSNM are both measured widely using the side of the minimum square fitted in the SRAM butterfly curves [25] [20]. Sizing and choices of the transistor threshold voltages (VTs) are essential in designing SRAM bitcell to make the HSNM and RSNM process variation tolerant. Similarly, another useful metric for evaluating the SRAM write operation is the write SNM (WSNM) metric, in which we assume that the WL has an infinite pulse width to write into the bitcell. The SNM metrics are very useful for characterizing both the write and read parameters that reduce the simulation burden to quantify write-ability and read-stability of 6T SRAMs.

3.0.3 Prior Art in Peripheral Assists

Peripheral assists are a class of circuits that help in performing write and read operations. The write assists improve the WM or WSNM of the bitcell and increase the write-ability. Existing works show that we can improve SRAM write-ability by applying peripheral assists such as wordline (WL (Figure 3.1a)) boosting (Figure 3.1b) [32] [53] [31], negative bitline (BL (Figure 3.1a)) (Figure 3.1b) [32] [53] [31], supply (V_{DD}) collapse [32] [53] [31], raising ground (V_{SS}) [32] [53] [31], etc. On the other hand, we can improve the SRAM readability using read assists such as V_{DD} boosting [32] [53], negative-V_{SS}(neg-V_{SS}) [32] [53], etc. One can bypass the read-stability issues due to poor RSNM of 6T SRAM bitcell using specific bitcell topologies [26] [27] [28] [29], or we can improve it by using read-stability assists such as WL under-drive [32] [53], V_{DD} boosting [32] [53], neg-V_{SS} [32] [53], etc., which have much lower

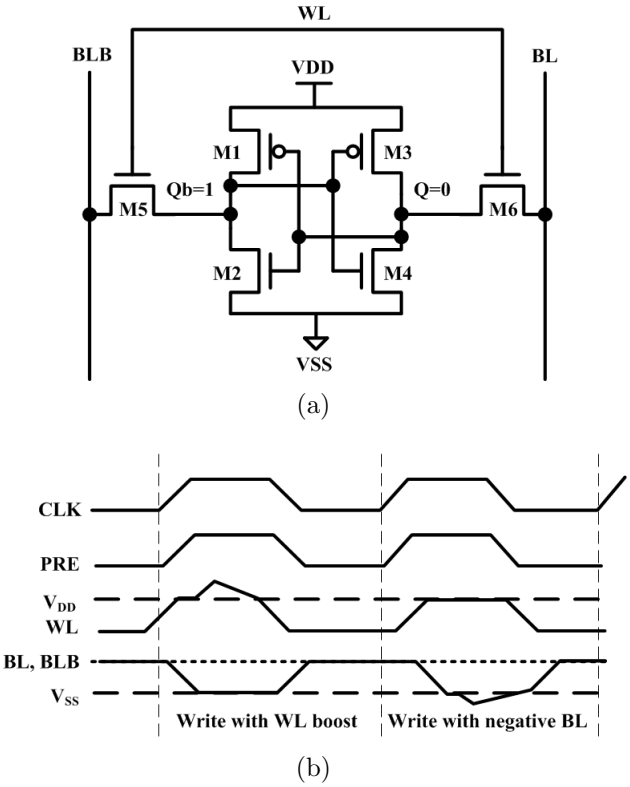


Figure 3.1: (a) Conventional 6T bitcell. (b) Wordline boost type write assist.

overhead than a bigger alternative bitcell topology in an array scenario. Prior works [54] [55] show the write and read static margins, V_{MIN}s, energy and delay trade-offs of a fixed capacity 6T SRAM across various peripheral assists. The work [54] propose a new combination of negative bitline with V_{DD} boosting for V_{MIN} improvement in 130nm bulk and sub-20nm FinFET technologies. However, these works lack findings on challenges of dynamic V_{MIN} for HD 6T FinFET bitcell and do not compare all possible combinations of write and read assists. Moreover, these works do not assume any constraints on the total voltage swing of the combined peripheral assists, without which the comparisons could have different total percentages of assists or voltage swings. On the other hand, authors in [56] show that for a fixed SRAM capacity and yield condition, the read and write dynamic V_{MIN}, and bit error rate vary across degrees of assists and operating frequencies in a 28nm bulk CMOS technology. However, this work lacks findings on how combining write and read peripheral assists would result in V_{MIN} improvement.

3.0.4 Selecting the Metrics for Comparison of Single and Combined Peripheral Assists

The static metrics are widely used for comparison among a set of bitcells for write-ability, readability, and read-stability. However, the SRAM static metrics and the corresponding V_{MIN}s do not represent the transient SRAM operation, which has a finite wordline pulse-width. Rather, the DC metrics assume that the wordline is turned on infinitely, which is an overestimate of the actual read-stability V_{MIN}, and an underestimate of write-ability V_{MIN} for a finite wordline pulse-width. On the other hand, the metrics such as dynamic V_{MIN}s corresponding to dynamic failure probabilities for write-ability, readability and read-stability criteria at an operating frequency, are direct measures of the chances that there will be a failure in the transient write and read operations. The required dynamic failure probabilities depend on the number of SRAM bits, SRAM yield, operating frequency, etc. design knobs. Hence, apart from investigating the 6T HD SRAM challenges using static metrics, we investigate the challenges and trade-offs of these assists and their combinations on the dynamic V_{MIN}s for dynamic write and read criteria across design knobs. Prior work shows that a dynamic metric defined as the critical wordline pulse-width (T_{crit}) [50] representing the transient SRAM operations, has weak or almost no correlation to N-curve metrics [50] and good correlation to WM and SNM type write-ability and read-stability static metrics [50]. Hence, we investigate the challenges and solutions of the single and combined assists for the write-ability, readability, and read-stability of 6T SRAMs using the following suitable design knobs and static and dynamic metrics.

3.0.5 Design Knobs

We vary the following design knobs to investigate the static and dynamic V_{MIN} and dynamic probability of failure (P_{fail}) for 6T HD FinFET bitcell.

- SRAM size (N)

- Process (P)
- Supply voltage(V_{DD})
- Temperature(T)
- frequency(f)

3.0.6 Evaluation Metrics

The static and dynamic evaluation metrics for this work are specified below.

- Write static noise margin (WSNM)
- Read static noise margin (RSNM)
- Static write-ability V_{MIN}
- Static read-stability V_{MIN}
- Dynamic write-ability V_{MIN}
- Dynamic readability V_{MIN}

3.0.7 Research Questions

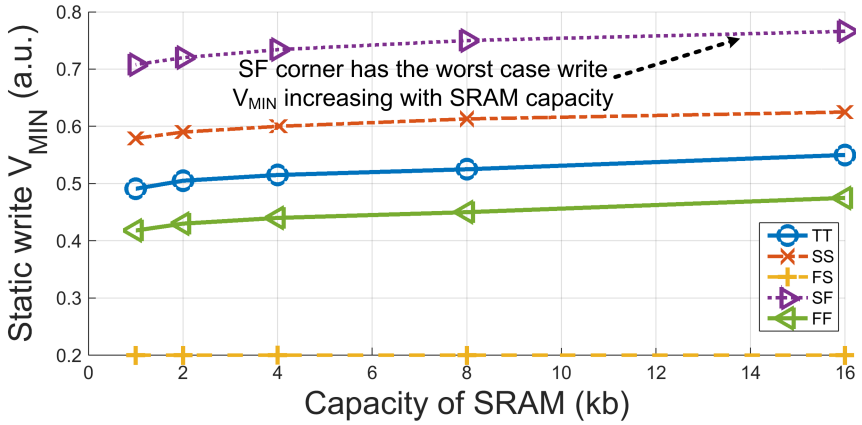
Following are those research questions we investigate in this chapter:

- What are the design challenges affecting SRAM static metrics, such as WSNM, RSNM, and write and read V_{MIN} for 6T HD FinFET SRAMs in 14nm technology across design parameters with single assists?
- What are the design challenges affecting SRAM dynamic metrics such as write and read V_{MIN} for 6T HD FinFET SRAMs in 14nm technology across design parameters using single assists?

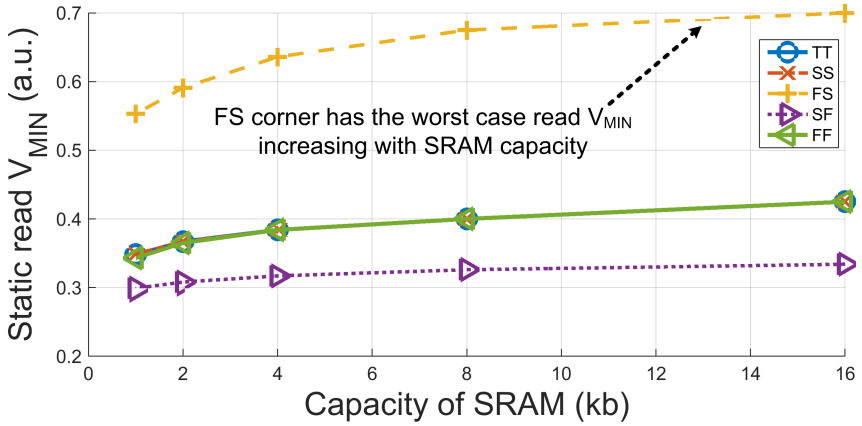
- How do the solo assists influence the SRAM static metrics, such as the WSNM, RSNM and write and read V_{MIN} at the worst-case corner in 14nm FinFET technology?
- What are the successful dual combinations of write-ability and read-stability assists with a fixed 20% of total voltage swing that improve the SRAM static metrics, such as the WSNM, RSNM and write and read V_{MIN} for the worst-case corner in 14nm FinFET technology?
- Which combined peripheral assist is the best using a 20% total voltage swing that reduces the write and read V_{MIN} for the worst-case corner in 14nm FinFET technology?

3.0.8 Experimental Assumption and Simulation Setup

We use the usual static metric measurement technique [25] for WSNM and RSNM margin, as well as static write and read V_{MIN} simulations. For the Monte Carlo (MC) simulations, we took 16K points to simulate static write and read V_{MIN}. Due to static DC simulations, we did not incorporate any parasitics in the bitcell. On the other hand, for the dynamic simulation, we use 10K MC points and use a parasitic switch in the FinFET models to add equivalent resistance and capacitance in each FinFETs in the 6T HD bitcell. For the combined peripheral assists we limit the total voltage swing of the assist percentages to 20% of the full-scale supply voltage. We assume the array size to be 128 rows and 128 columns and model the dynamic simulation deck using equivalent loading added to the wordline and bitline of the core array. We obtain the static and dynamic V_{MIN} data for 16kb and 10kb memory sizes and interpolate the failure rates using logarithmic interpolation to compute the V_{MIN} across 8kb, 4kb, 2kb and 1kb sizes. Note that we perform all the simulations in a commercial 14nm technology for the 6T HD FinFET bitcell.



(a)



(b)

Figure 3.2: (a) 6T HD SRAM static write V_{MIN} vs. capacity at 27C temperature across process variation. (b) 6T HD SRAM static read V_{MIN} vs. capacity at 27C temperature across process variation.

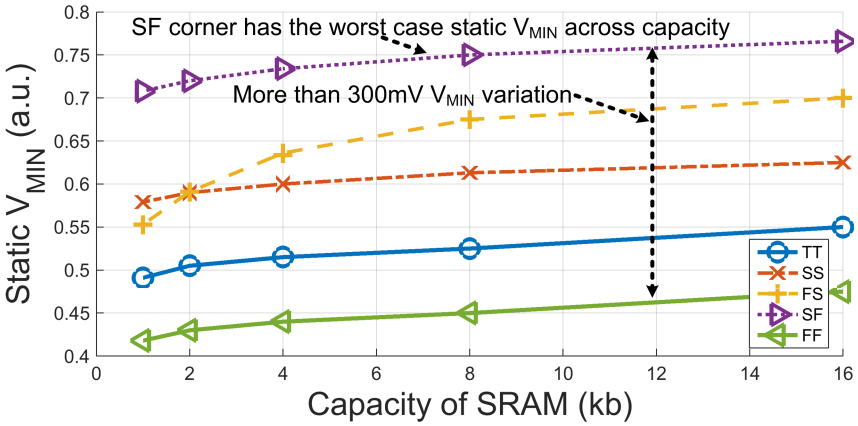
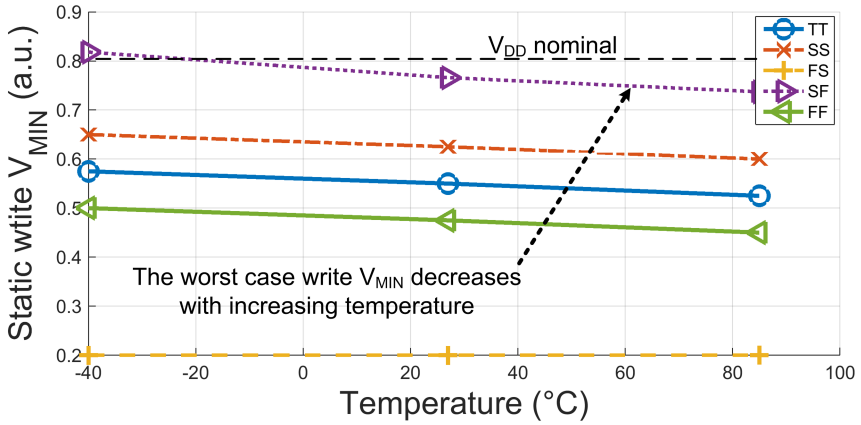
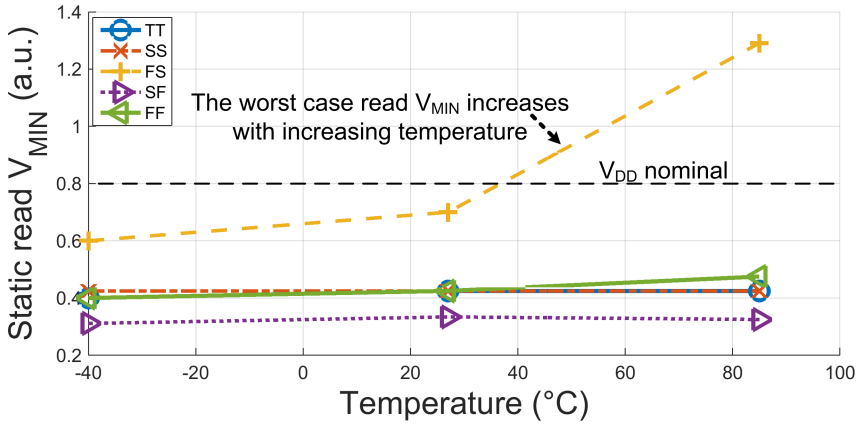


Figure 3.3: 6T HD SRAM static V_{MIN} vs. capacity at 27C temperature across process variation.



(a)



(b)

Figure 3.4: (a) 6T HD SRAM static write V_{MIN} vs. temperature for 16kb capacity across process variation. (b) 6T HD SRAM static read V_{MIN} vs. temperature for 16kb capacity across process variation.

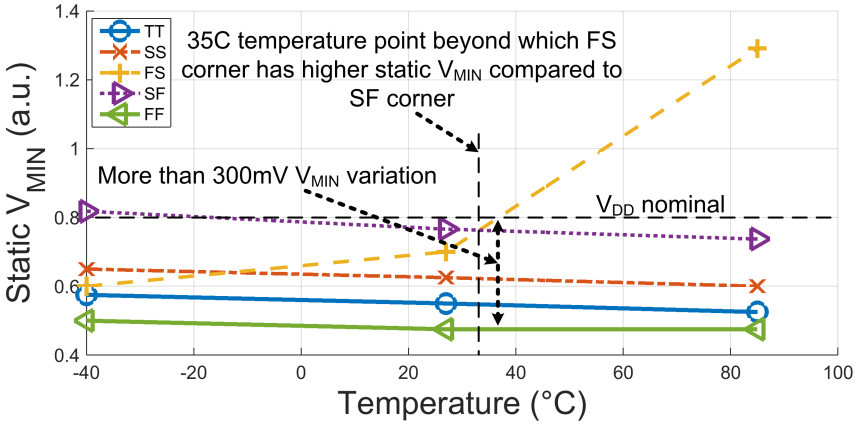
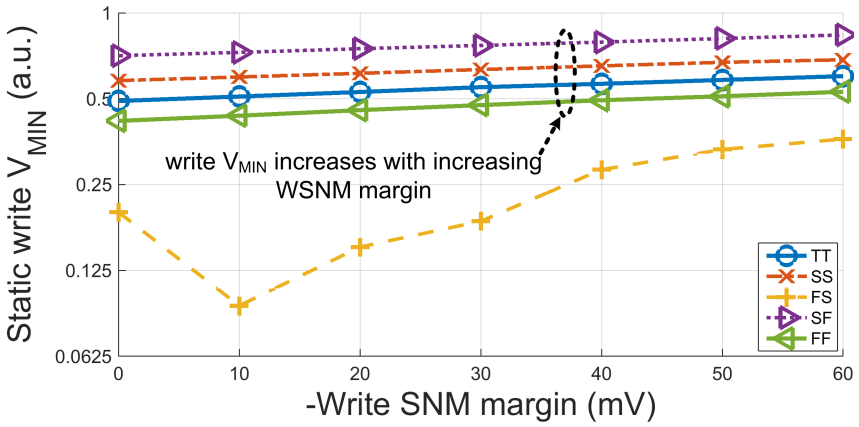
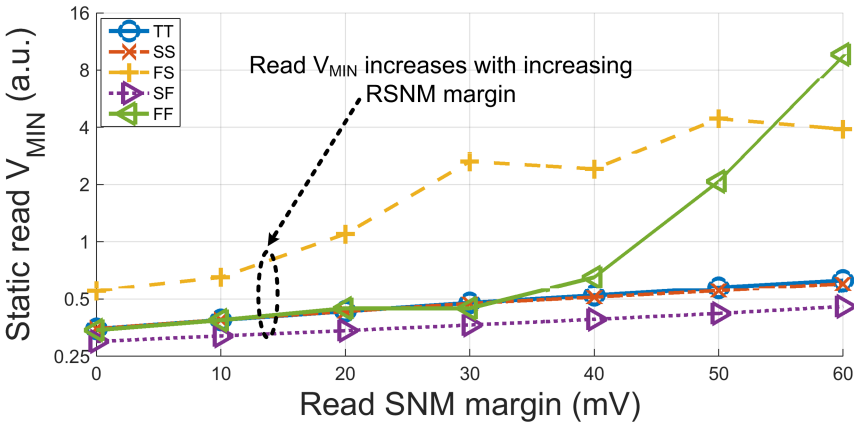


Figure 3.5: 6T HD SRAM static V_{MIN} vs. temperature for 16kb capacity across process variation.



(a)



(b)

Figure 3.6: (a) 6T HD SRAM static write V_{MIN} vs. negative write static noise margin (-WSNM) for 1kb capacity at 27C temperature across process variation. (b) 6T HD SRAM static read V_{MIN} vs. read static noise margin (RSNM) for 1kb capacity at 27C temperature across process variation.

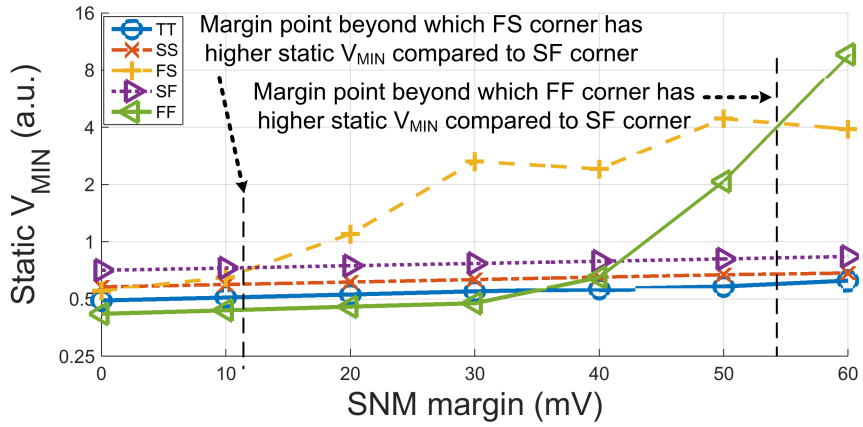
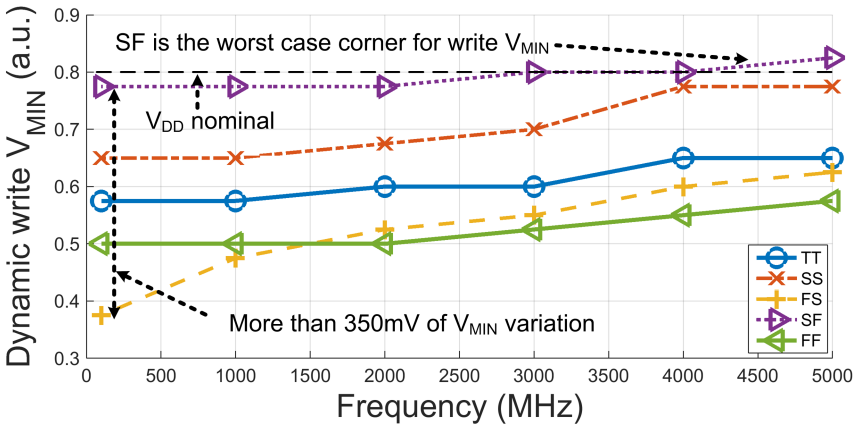


Figure 3.7: 6T HD SRAM static V_{MIN} vs. magnitude of the write or read static noise margin (SNM) for 1kb capacity at 27C temperature across process variation.

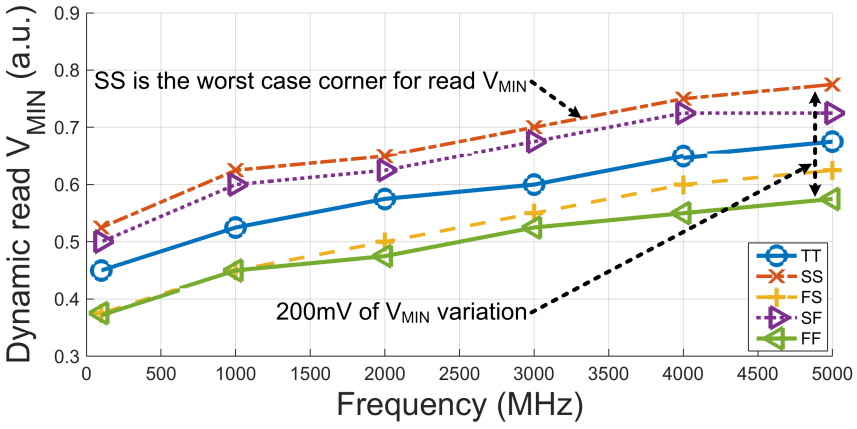
3.0.9 Challenges in Static V_{MIN} for 6T HD FinFET SRAMs

Figure 3.2a shows that, with the increase in capacity, the static write V_{MIN} of 6T HD FinFET SRAM increases and the SF corner is the worst-case. On the other hand, the static read V_{MIN} also increases with the capacity (Figure 3.2b) of the SRAM; however, the worst-case corner is FS. Overall, the write V_{MIN} is the worst-case and Figure 3.3 shows the worst-case corner for the static V_{MIN} as the SF corner. With a temperature increase, the static write V_{MIN} decreases (Figure 3.4a) and the write V_{MIN} at the SF_-40C corner is higher than that of the nominal supply voltage of 0.8V. On the other hand, Figure 3.4b shows that the static read V_{MIN} of the 6T HD FinFET SRAM slowly increases for most of the corners, except the FS corner, and the SF corner is the worst-case for read V_{MIN}. Beyond the 35C temperature, the SF corner has higher V_{MIN} compared to the nominal supply voltage of 0.8V, as shown in Figure 3.4b. The Figure 3.5 shows the overall static V_{MIN} across temperature variation. The plot shows that below 35C the SF corner is the worst-case; however, above 35C the FS corner becomes the worst-case for the static V_{MIN}. Figure 3.6a shows the static write V_{MIN} across the increasing magnitude of WSNM. As increasing WSNM magnitude is tantamount to higher write-ability yield, the plot captures the trends that the write V_{MIN} increases with increases in the magnitude of the WSNM metric. Similarly, the static read V_{MIN} also increases with the increase in RSNM magnitude (Figure 3.6b). Finally, Figure 3.7 shows an interesting trend across corners for the overall static V_{MIN} across the magnitude increase of the SNM margin. The figure shows that, below 10mV SNM margin, the SF corner has the worst-case V_{MIN}. From 10mV to about 55mV margin the FS corner has the worst-case V_{MIN}, and beyond 55mV the FF corner has the worst-case V_{MIN}. Thus, across capacity, temperature, and process variation we see more than 300mV of V_{MIN} variation, which makes the design of 6T HD SRAM a very challenging task that requires careful selection of assists for the reduction of 6T HD FinFET V_{MIN}.

3.0.10 Challenges in Dynamic Write and Read V_{MIN} for 6T HD FinFET SRAMs



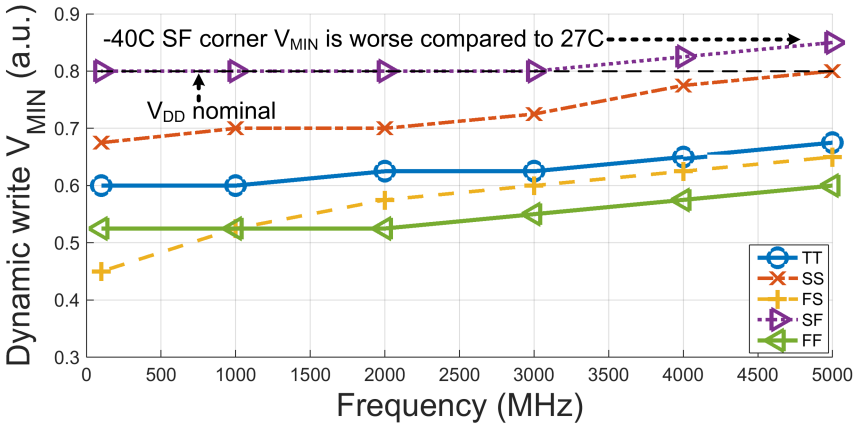
(a)



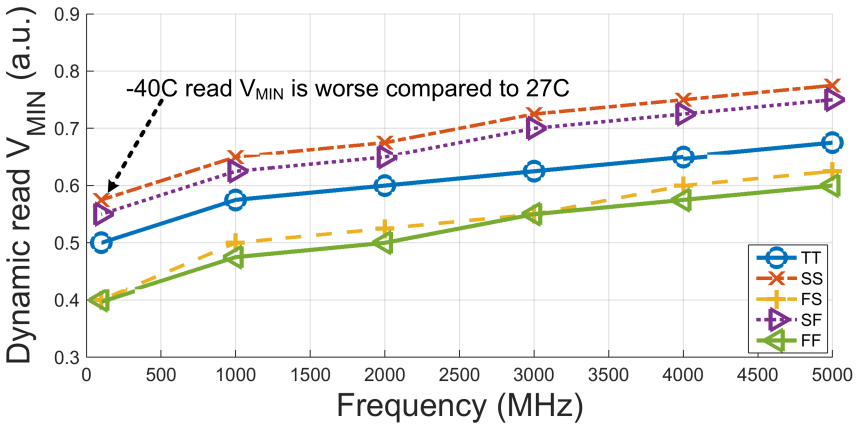
(b)

Figure 3.8: (a) 6T HD SRAM dynamic write V_{MIN} vs. clock frequency at 27C temperature for 10kb SRAM capacity across process variation. (b) 6T HD SRAM dynamic read V_{MIN} vs. clock frequency at 27C temperature for 10kb SRAM capacity across process variation.

For the dynamic V_{MIN} metric, we investigate the effects of 6T HD FinFET SRAM input design knobs, such as process, temperature, and frequency. The SRAM dynamic V_{MIN} is the worst of dynamic write-ability, readability, and read-stability V_{MIN}. Usually, at higher clock frequencies the half-select V_{MIN} excludes from the calculation of dynamic V_{MIN}. However, in tens of MHz of clock frequencies or lower, the row and column half-select V_{MIN}s have to be taken into account, those start to dominate the overall SRAM dynamic V_{MIN}. Figure 3.8a shows the plot for dynamic write V_{MIN} vs. frequency at 27C for 10kb SRAM capacity.

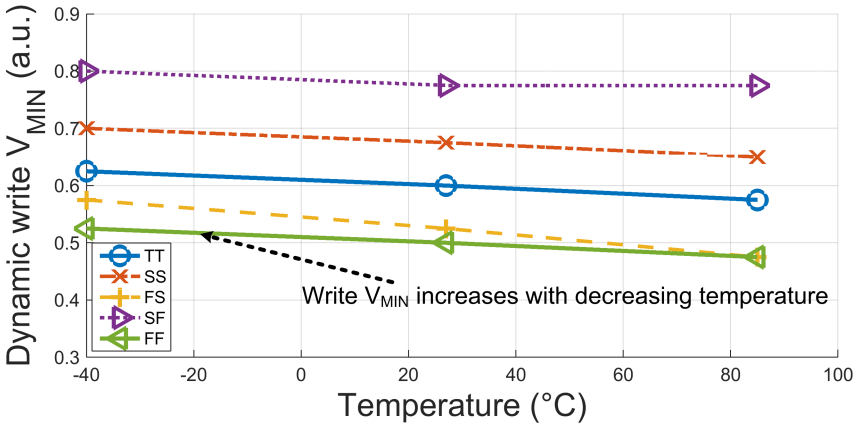


(a)

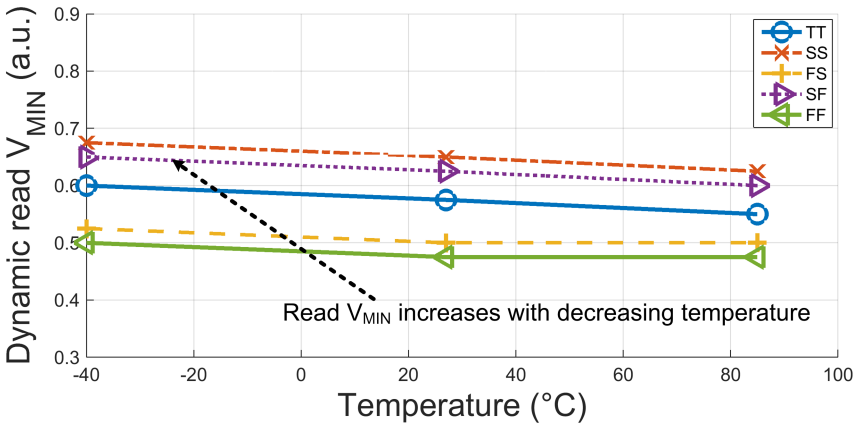


(b)

Figure 3.9: (a) 6T HD SRAM dynamic write V_{MIN} vs. clock frequency at -40C temperature for 10kb SRAM capacity across process variation. (b) 6T HD SRAM dynamic read V_{MIN} vs. clock frequency at -40C temperature for 10kb SRAM capacity across process variation.

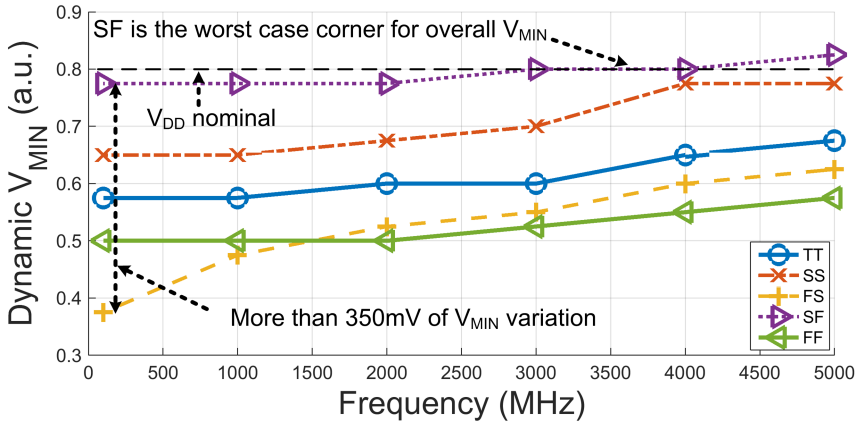


(a)

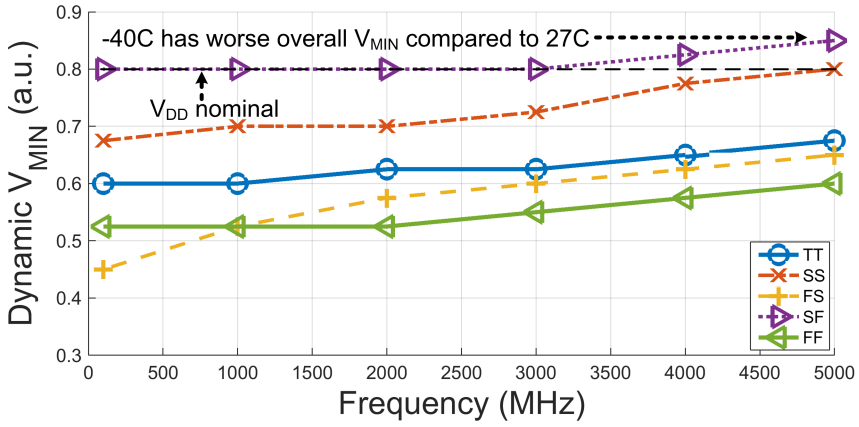


(b)

Figure 3.10: (a) 6T HD SRAM dynamic write V_{MIN} vs. temperature at 2GHz clock frequency for 10kb SRAM capacity across process variation. (b) 6T HD SRAM dynamic read V_{MIN} vs. temperature at 2GHz clock frequency for 10kb SRAM capacity across process variation.



(a)



(b)

Figure 3.11: (a)6T HD SRAM dynamic V_{MIN} vs. clock frequency at 27C temperature for 10kb SRAM capacity across process variation. (b) 6T HD SRAM dynamic V_{MIN} vs. clock frequency at -40C temperature for 10kb SRAM capacity across process variation.

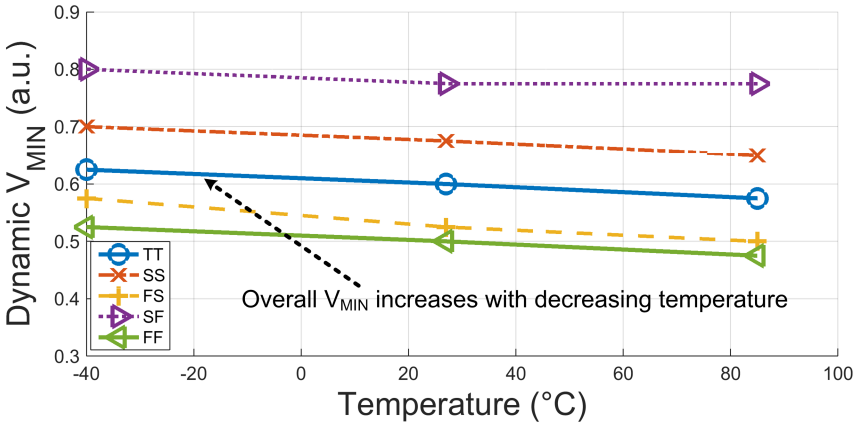
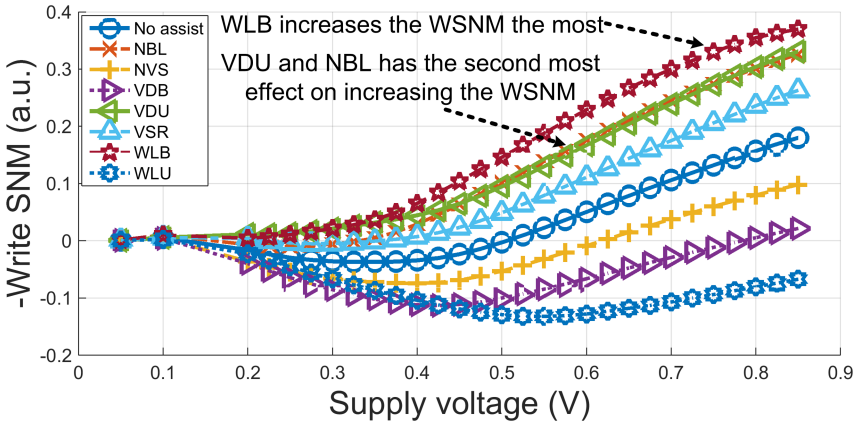


Figure 3.12: 6T HD SRAM dynamic V_{MIN} vs. temperature at 2GHz clock frequency for 10kb SRAM capacity across process variation.

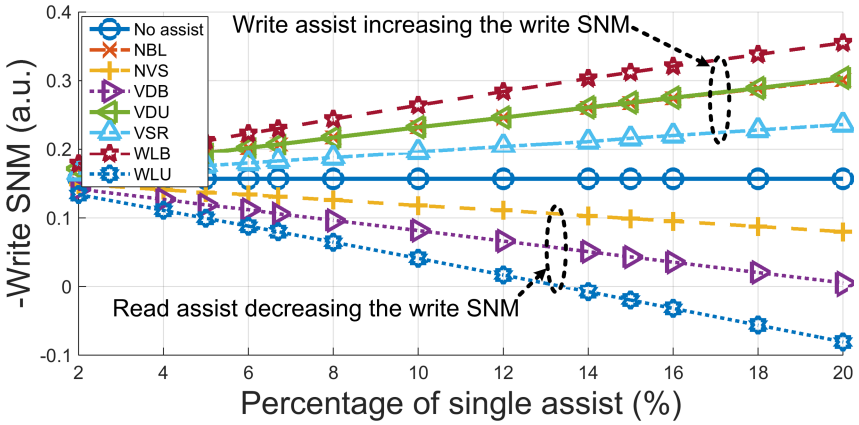
The figure depicts that the SF corner is the worst-case for dynamic write V_{MIN} and the overall write V_{MIN} variation is more than 350mV. Noticeably, at a higher frequency, the dynamic write V_{MIN} is higher than the nominal supply voltage. Figure 3.8b shows the plot for dynamic read V_{MIN} vs. frequencies at 27C for 10kb capacity. The read V_{MIN} is worst for the SS corner, and the total V_{MIN} variation is around 200mV. On the other hand, Figure 3.9a shows the plot for dynamic write V_{MIN} vs. frequency at -40C for 10kb capacity across process variation. It shows that at -40C, the dynamic write V_{MIN} increases compared to the 27C. Similarly, Figure 3.9b shows the plot for the dynamic read V_{MIN} vs. frequency at -40C for 10kb capacity, which shows the read V_{MIN} increases at -40C compared to the 27C temperature. Figure 3.10a and Figure 3.10b summarize the temperature trends for dynamic write and read V_{MIN}, which show that the write as well as read V_{MIN} increase with decreasing temperature at 2GHz for 10kb capacity. The overall V_{MIN} for the 6T HD FinFET SRAM is shown in Figure 3.11a for 27C and in Figure 3.11b for -40C temperatures. The Figure 3.12 depicts the temperature trend for SRAM dynamic V_{MIN} at 2GHz for 10kb capacity. Thus, addressing the large dynamic V_{MIN} variation across frequency and temperature variations for 6T HD FinFET SRAM design is a challenging task, which needs to be addressed.

3.0.11 Static Write-ability Margin Across Single Peripheral Assists of 6T HD FinFET SRAMs

In a conventional 6T SRAM bitcell, to perform a write operation, one of the pass-gate NMOS transistors pulls down the internal node of the bitcell. Thus, during a write '0' operation, one of the pass-gate transistors of the 6T bitcell based on ratioed logic fights with the pull-up PMOS transistor to overcome its drive strength. However, in a corner where the pass-gate transistor is weak, and the pull-up transistor is strong, such as the slow NMOS and fast PMOS (SF) corner, the 6T bitcell faces poor write-ability. Hence, the SF corner is the worst-case corner for the 6T SRAM bitcells. Using write peripheral assists one can improve the write-ability of 6T SRAMs. Figure 3.13a shows a plot of negative WSNM vs. supply



(a)



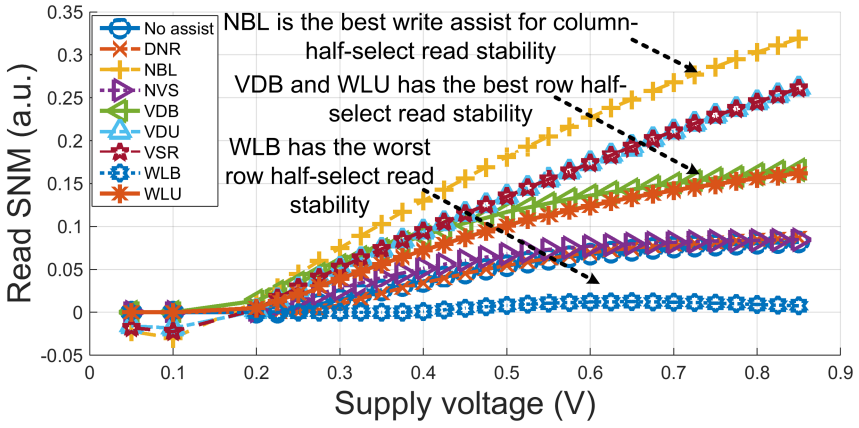
(b)

Figure 3.13: (a) Negative of WSNM vs. supply voltage at the worst-case write corner SF_-40C for a 20% of assist across single peripheral assists. (b) Negative of WSNM vs. percentage of peripheral assists at the worst-case corner SF_-40C at 0.8V supply voltage across solo assists.

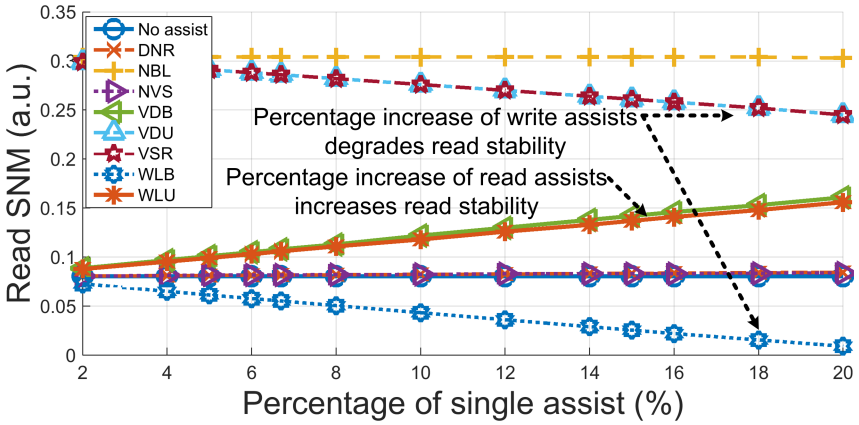
voltage across single peripheral assists at the worst-case write corner SF_-40C for a 20% of assist. The plot shows that the wordline boost (WLB) write assist improves the WSNM metric the most, and the V_{DD} collapse (VDU) and negative bitline (NBL) are the second most useful write assists for improving the write-ability of 6T HD FinFET bitcells. Although WLB may be the best write assist from WSNM metric, it hampers the read-stability of the row half-selected bitcells in a bitline interleaving scenario. Note that the bitline interleaving technique is widely used to improve on the Soft Error Disturbance (SED) from sub-atomic particle strike in 6T SRAM bitcells. Thus, it is critical to know what percentage of write assists improve the WSNM. Moreover, the read-stability assists are used to suppress the effect of row half-select issues, but hamper the write operation. Thus, Figure 3.13b shows the negative WSNM vs. peripheral assists to identify the trends and behaviors of write-ability and read-stability single assists at the FS_-40C corner V_{DD}=0.8V supply voltage. The plot shows that the WLB, NBL, VDU, and V_{SS} raising (VSR) are write assists and increasing the percentage linearly increases the magnitude of WSNM. On the other hand, wordline under-drive (WLU), V_{DD} boosting (VDB), and negative V_{SS} (NVS) are the read-stability assist, as they decrease the magnitude of WSNM with increased assist percentages. Choosing the appropriate assists for improving the SRAM write-ability is a challenging task. Thus, an SRAM designer should carefully choose the percentage of write assist based on the WSNM requirements to meet the write-ability and read-stability specifications.

3.0.12 Static Read-stability Margin Across Peripheral Assists of 6T HD FinFET SRAMs

The 6T bitcell read-stability metric RSNM depends on the beta-ratio of the pull-down NMOS transistor to the pass-gate NMOS transistor. For a better read-stability during the SRAM read, the drive strength of the pull-down NMOS transistor must be strong enough, compared to the pass-gate NMOS transistor. The higher drive strength of the pull-down NMOS ensures that the voltage drop in the internal node (junction of pass-gate NMOS and



(a)



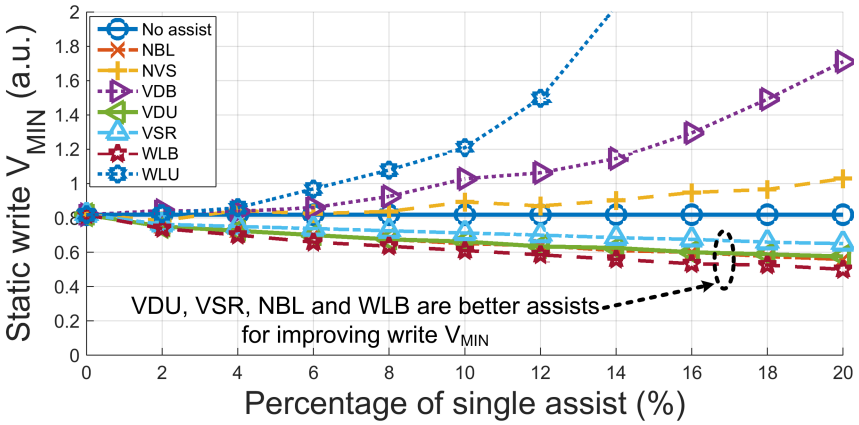
(b)

Figure 3.14: (a) RSNM vs. supply voltage at the worst-case read-stability corner FS_85C for a 20% of assist across single peripheral assists. (b) RSNM vs. percentage of peripheral assists at the worst-case read-stability corner FS_85C at 0.8V supply voltage across solo assists.

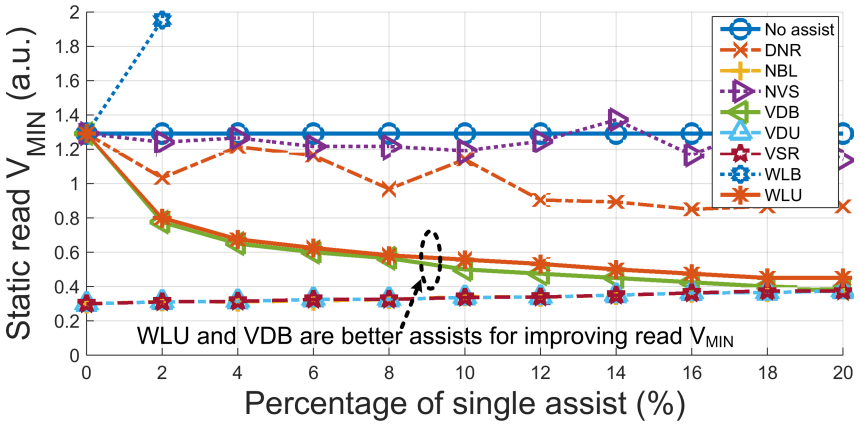
pull-down NMOS) keeps below the threshold voltages of the pull-up and pull-down transistors on the other side of the bitcell. Thus, the worst-case corner for the read-stability would be when the pass-gate NMOS transistor is robust, and the pull-down NMOS transistor is weak, or the pull-up PMOS on the other side is weak. In other words, the corresponding worst-case corner is the fast NMOS and slow PMOS (FS) corner. Using read assists one can improve the read-stability of the 6T SRAMs. Figure 3.14a shows the plot for RSNM vs. supply voltage at the FS_85C corner for 20% assist percentage across various solo assists. The plot shows that the VDB and WLU are the best row half-select read-stability assists. On the contrary, the WLB write assist has the worst row half-select read-stability. For the column half-select issue, the NBL write assist has the best read-stability compared to the others. It is crucial to ensure that an SRAM designer chooses the proper percentages of read-stability assist to meet the SRAM specifications. Thus, Figure 3.14b shows the trends and comparison of assists for RSNM metric for read-stability at the FS_85C corner at 0.8V supply voltage. The plot shows that the percentage increase of write assist degrades the read-stability; however, NBL has the best, and VDU and VSR have the second-best column half-select read-stability among all the write assists. On the other hand, WLB has the worst RSNM read-stability magnitude across assist percentages.

3.0.13 Worst-case V_{MIN} Improvement Using Single Peripheral Assists

Figure 3.15a shows the plot for static write V_{MIN} vs. single assist percentages at the SF_-40C corner. It shows that WLB, VDU, VSR, and NBL improve the write V_{MIN} at the worst-case corner. On the other hand, the VDB and WLU read-stability assists degrade the write V_{MIN} with the increase of assist percentages. The Figure 3.15b shows static read V_{MIN} vs. single assist percentage at the worst-case FS_85C corner. The plot shows that the VDB and WLU are the only assists that improve the read-stability V_{MIN} with increasing assist percentages. Besides, the NBL, VSR, and VDU write assists have the lowest column



(a)



(b)

Figure 3.15: (a)6T HD SRAM static write V_{MIN} vs. assist percentages at the SF_-40C corner across single assist for 16kb SRAM capacity. (b) 6T HD SRAM static read V_{MIN} vs. assist percentages at the FS_85C corner for 16kb SRAM capacity.

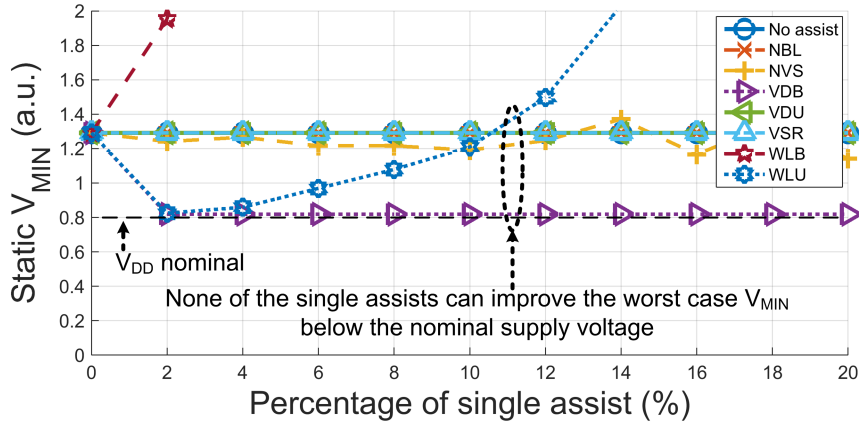
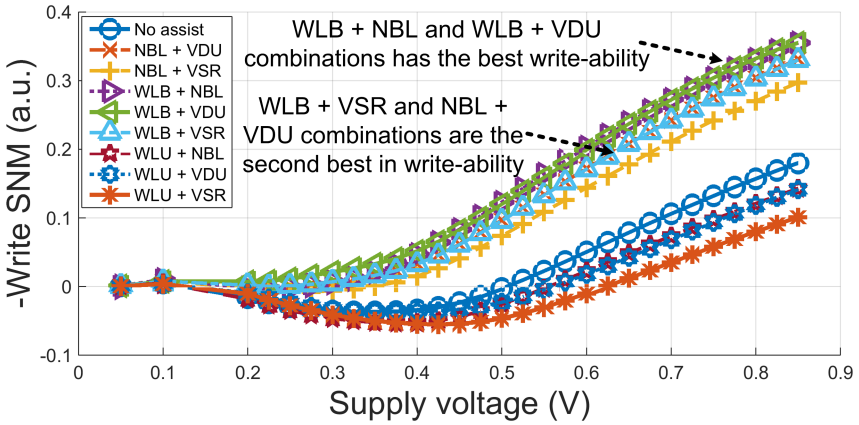


Figure 3.16: The worst-case static V_{MIN} of the 6T HD SRAM vs. assist percentages for 16kb SRAM capacity.

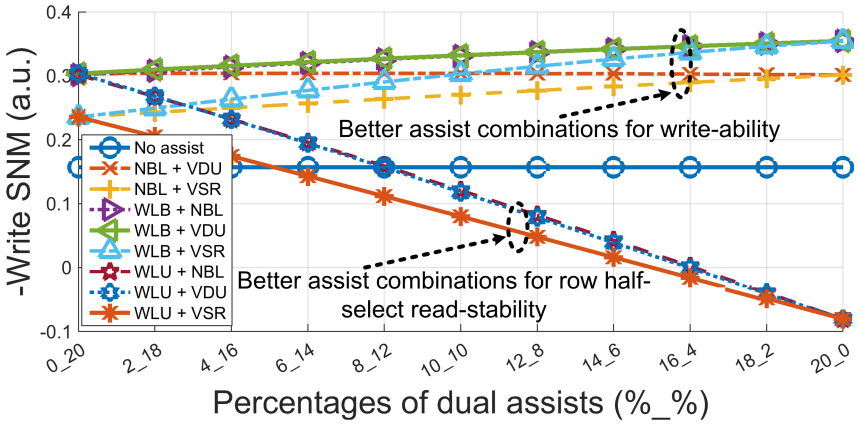
half-select V_{MIN} that increases slowly with assist percentages. Noticeably, without bitline interleaving single write and read assists can lower the 6T HD SRAM V_{MIN}. However, in widely-used bitline interleaving or column mux scenarios, none of the single assists can improve the worst-case V_{MIN} across assist percentages (Figure 3.16). Improving the worst-case V_{MIN} in column mux scenario requires combinations of write and read assist, those have shown to work in the state-of-the-art. However, it is not shown exhaustively what the potential write-read assist combinations are that would work, and which combination is the best, with the assumption that the total voltage swing of the assists is constant. We investigate the effect of write and read combined peripheral assists to improve the worst-case V_{MIN} in the next section.

3.0.14 Combined Peripheral Assists (CPA) for SRAM Margin and V_{MIN} Improvements

As single peripheral assists fail to lower the worst-case 6T HD FinFET V_{MIN} at 14nm technology, we investigate what the combinations of write and read assists are and their percentages that allow us to reduce the worst-case SRAM V_{MIN} across process and temperature variation. Figure 3.17a shows the plot for negative WSNM vs. supply voltage across dual assist combinations at the SF_-40C corner using equal 10% each assist percentages. The plot shows that the blends WLB + NBL and WLB + VDU are the best that improve the magnitude of WSNM. The second best combinations for strengthening the WSNM are WLB + VSR and NBL + VDU. Figure 3.17b shows the plot for negative WSNM vs. various dual assist percentage combinations at the SF_-40C corner at 0.8V supply voltage. The figure shows the better double assist combinations for improving the write-ability (NBL +VSR, NBL + VDU, WLB + NBL, WLB + VDU, and WLB + VSR) and row half-select read-stability (WLU + NBL, WLU + VSR, and WLU + VDU). On the other hand, the Figure 3.18a shows the plot for RSNM vs. supply voltage across various dual assist percentages at the FS_85C corner using (each assist with 10% strength) a total of 20% assist percentage per combination.

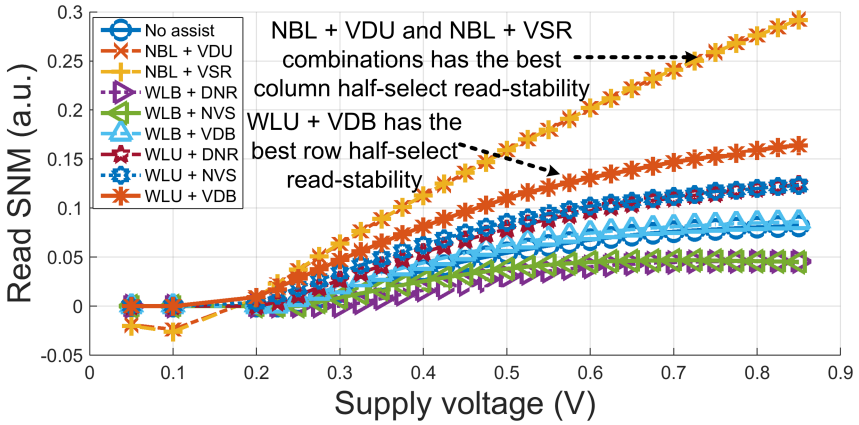


(a)

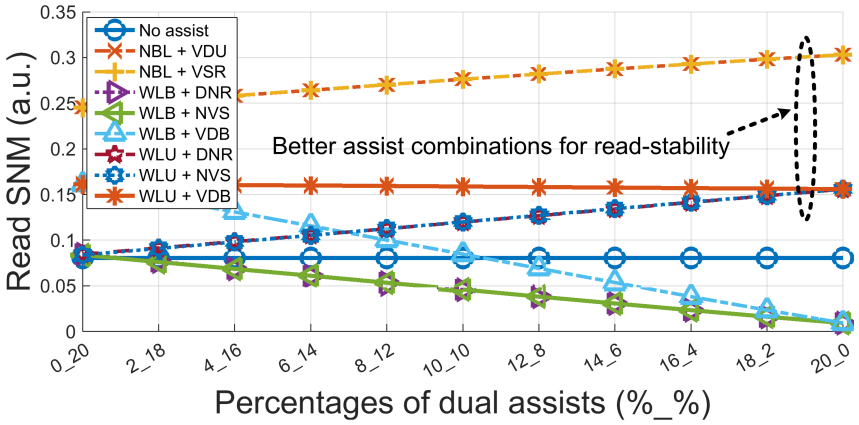


(b)

Figure 3.17: (a) Negative write static noise margin (-WSNM) vs. supply voltage at the SF_-40C corner across dual assists using a total of 20% (10% percentage each) assist percentage for 6T FinFET bitcell. (b) Negative write static noise margin (-WSNM) vs. dual assist percentage at the SF_-40C corner using 0.8V supply voltage across dual assist combinations.



(a)

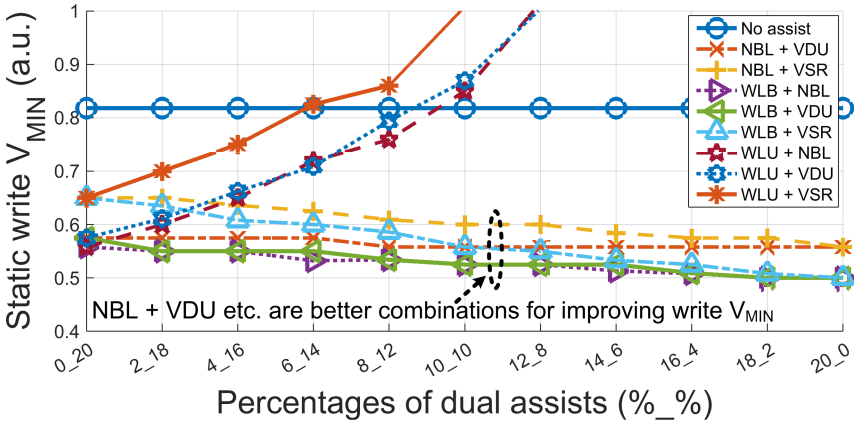


(b)

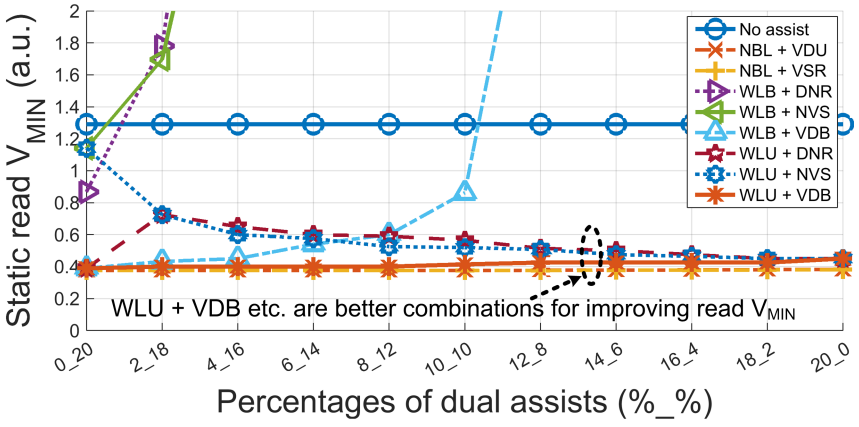
Figure 3.18: (a) Read static noise margin (RSNM) vs. supply voltage at the FS_85C corner across dual assists using a total of 20% (10% percentage each) assist percentage for 6T FinFET bitcell. (b) Read static noise margin (RSNM) vs. dual assist percentage at the FS_85C corner using 0.8V supply voltage across dual assist combinations.

The plot shows the assist combinations NBL + VDU, and NBL + VSR have the best RSNM for column half-select read-stability and WLU + VDB has the best row half-select stability. Similarly, Figure 3.18b shows the plot for RSNM vs. dual assist percentage with the total assist percentage of 20% at the FS_85C corner at 0.8V supply voltage and gives the better combinations for improving read-stability.

3.0.15 Improvement of SRAM Static V_{MIN} using CPA



(a)



(b)

Figure 3.19: (a) Static write V_{MIN} vs. dual assist percentages at the SF_-40C corner across dual assists using a total of 20% assist percentage for a 16kb capacity 6T HD FinFET SRAM. (b) Static read V_{MIN} vs. dual assist percentages at FS_85C corner across dual assists using a total of 20% assist percentage for a 16kb capacity 6T HD FinFET SRAM.

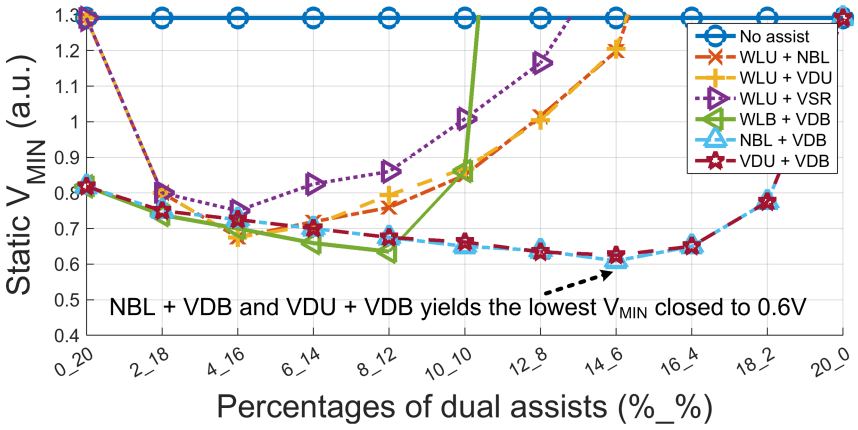
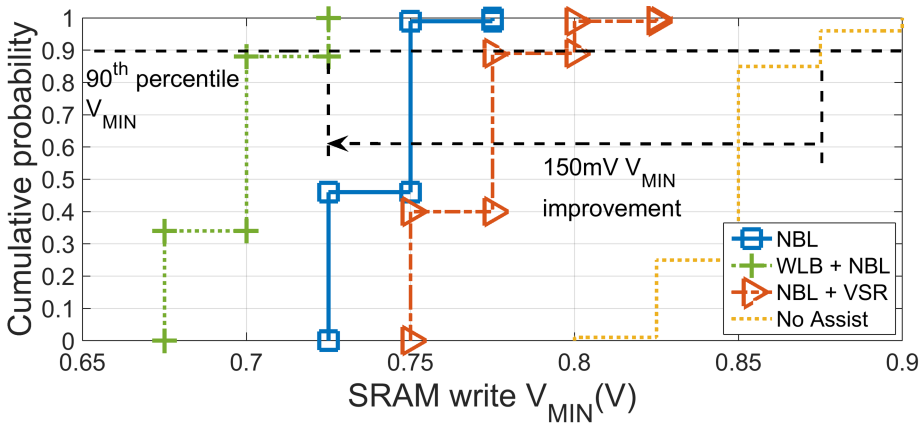


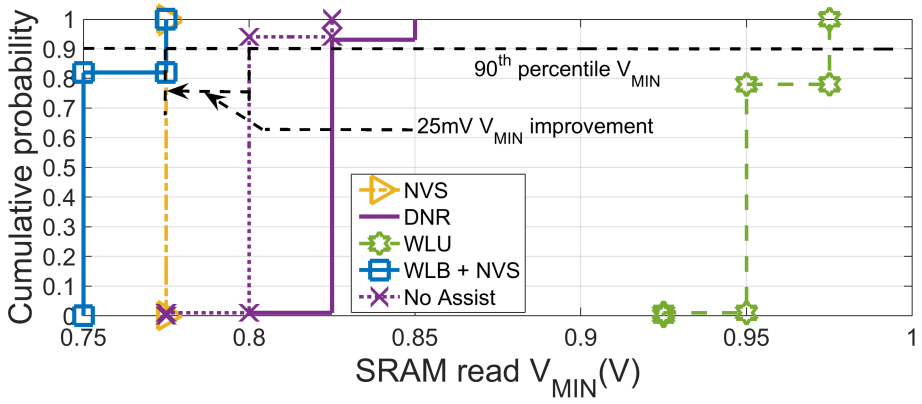
Figure 3.20: The worst-case static V_{MIN} vs. working dual assist percentages for 16kb SRAM capacity for 6T HD FinFET SRAM V_{MIN} lowering.

Figure 3.19a shows the plot for the static write V_{MIN} across dual assist combinations, with the total assist percentage being constant at 20% at the SF_-40C corner. The plot depicts the better combinations to improve the static write V_{MIN}, such as NBL + VDU, NBL + VSR, WLB + NBL, WLB + VSR, and WLB + VDU. On the other hand, Figure 3.19b shows the plot for static read V_{MIN} vs. dual assist combinations, with the total assist percentage being constant at 20% at the worst-case read V_{MIN} corner of FS_85C. The plot depicts the better combinations for improving the static read-stability V_{MIN}, such as WLU + VDB, etc. Finally, Figure 3.20 shows all valid combinations of write and read assist that allow us to lower the worst-case V_{MIN}. Noticeably, the blends NBL + VDB and VDU + VDB achieve the lowest V_{MIN} using 14%_6% assist combinations. The combinations WLU + NBL, WLU + VDU, and WLU + VSR have relatively higher minima in V_{MIN} across dual assist percentages compared to the NBL + VDB, etc. blends. However, using VDB has more energy and layout penalty compared to the circuit implementation of NBL or VDU or VSR, as the boosting cap for VDB is much higher due to higher V_{DD} wire capacitance. Thus, although the combinations NBL + VDB and VDU + VDB achieve a lower minima in V_{MIN}, they may have higher energy and layout penalties compared to the circuit implementation of the WLU + NBL, etc. combinations.

3.0.16 Dynamic V_{MIN} Improvement using CPA



(a)



(b)

Figure 3.21: (a) Cumulative distribution of 100 chip simulations of dynamic write V_{MIN} across single and dual assist percentages at the SF_27C corner using a total of 20% assist percentage for 10kb 6T HD FinFET SRAM capacity. (b) Cumulative distribution of 100 chip simulations of dynamic read V_{MIN} across single and dual assist percentages at the SS_-40C corner using a total of 20% assist percentage for 10kb 6T HD FinFET SRAM capacity.

We also perform 100 chip simulations using 10K Monte Carlo samples for dynamic write and read operations to capture if this notion of combined assist using a fixed total percentage of voltage swing is capable of lowering the V_{MIN} across assist blends. Figure 3.21a shows the plot for the cumulative distribution function of the dynamic write V_{MIN} across single and dual combinations of assists, with a total of 20% fixed assist percentage using 10% each for double assists. The plot shows that the blend of WLB + NBL beats all other combinations and achieves 150mV V_{MIN} improvement at the SF_27C corner. On the other hand, Figure

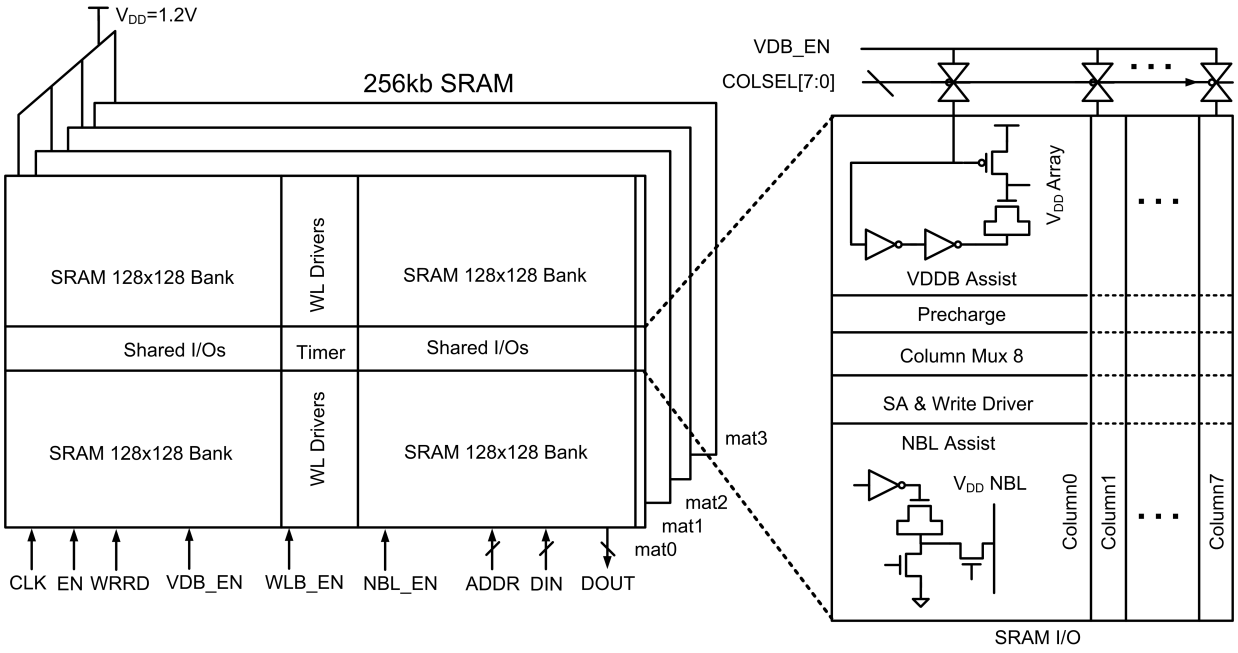
3.21b shows the plot for the cumulative distribution function of the dynamic readability V_{MIN} across single and dual combinations of assists, with a total of 20% fixed assist percentage using 10% each for double assists. It shows that WLB + NVS has a better cumulative distribution compared to other blends and it improves 25mV of read V_{MIN} across 100 chip simulations at the worst-case SS_-40C corner.

3.0.17 Testchip for 256Kb 6T SRAM Using CPA

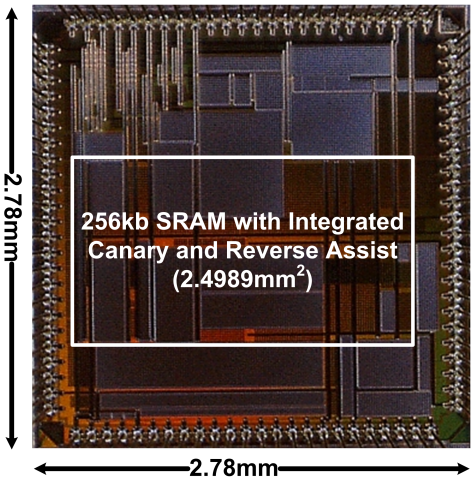
We design a testchip using three peripheral assists, such as WLB, VDB, and NBL. Among those peripheral assists, NBL and WLB are write assists, and the VDB is a read assist. The chip is fabricated using a commercial 130nm bulk technology. Figure 3.22a shows the internal architecture of the 256kb SRAM testchip for CPA, which has four SRAM subarrays named as mat0, mat1, mat2, and mat3. Each SRAM mat has four banks of 128x128 arrays of bitcells, shared input and output circuits (I/Os), one control logic for SRAM internal timing and power management, and two rows of SRAM wordline row drivers, which are part of the SRAM row decoder. The SRAM has external control pins such as CLK, EN, WRRD, assist control pins such as VDB_EN, WLB_EN, and NBL_EN, address bus ADDR, data input bus DIN, and data output bus DOUT. Figure 3.22b gives the die photograph of the testchip. Figure 3.23 shows that, among 30 chip measurements, using the blend WLB + VDB + NBL beats all other CPA combinations and achieves the V_{MIN} improvement of 240mV for 90^{th} percentile V_{MIN} compared to the no-assist case. The best V_{MIN} achieved using CPA is 0.38V at the 27C temperature among all the chips. The CPA area overhead is around 2.81%, which is very close to the overhead of 3% shown in the state-of-the-art. Table 3.1 compares this work with the state-of-the-art.

3.0.18 System Level Projected Savings for CPA Scheme

This section computes the system level savings using battery replacement time for the best CPA scheme that uses a dual rail architecture. Here, we assume that SRAM consumes 40%



(a)



(b)

Figure 3.22: (a) Architecture of the 256kb SRAM using combined peripheral assist (CPA) of wordline booting, negative bitline and V_{DD} boosting. (b) Die photograph of the 256kb SRAM testchip (© 2017 IEEE).

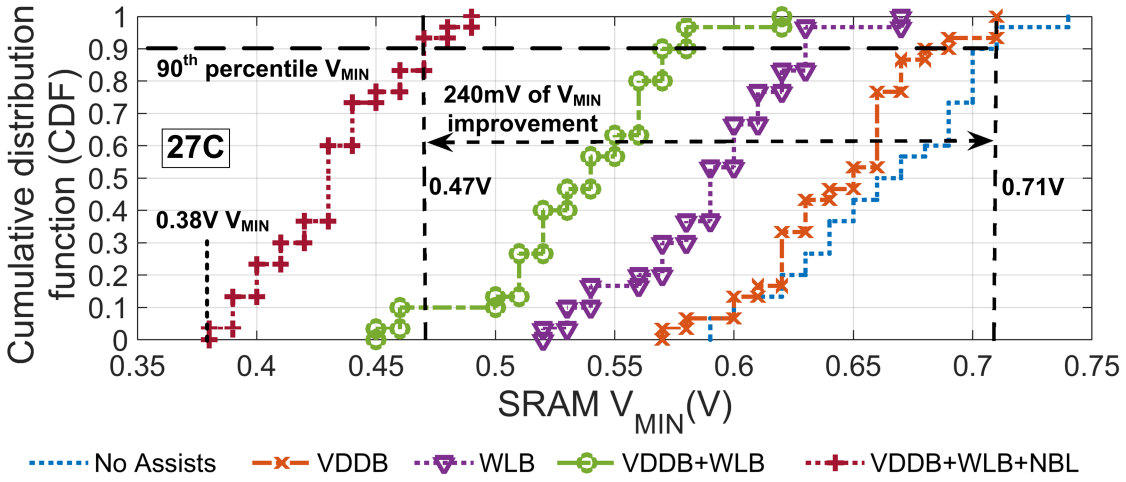


Figure 3.23: Cumulative distribution of measured SRAM V_{MIN} showing 240mV of V_{MIN} improvement using combined peripheral assist (CPA) of wordline booting, negative bitline and V_{DD} boosting for the 256kb SRAM (© 2017 IEEE).

energy while the logic core consumes 60% energy. We further assume that the IoT system uses an A1578 (0.76Wh) battery and the system average power consumption is the same as Apple iWatch average power consumption of 42.2mW. Thus, using CPA assuming 100% duty cycle, the battery-life improves by 31.26% and the corresponding battery replacement time increases from 1.025 yrs to 1.345 yrs. These projections assume a 10% self-discharge rate for the A1578 battery. Using a Lithium-ion LIR2032 (0.144Wh) coin cell, the corresponding battery-life improvement is 31.07% and the battery replacement time improves from 2.35 months to 3.08 months assuming a maximum of 500 recharge cycles. The battery-life improvement for a Silver Oxide SR416SW (0.0124Wh) battery is 31.37%. The equations used to calculate the battery-life using self-discharge assumptions are derived in Appendix A for the SR416SW Silver Oxide, LIR2032 Lithium-ion, and A1578 Lithium-ion Polymer batteries.

3.0.19 Conclusions

6T HD FinFET SRAM design sees challenges from the variations of process, temperature, frequency, and other design parameters, and has more than 300mV of V_{MIN} across these design knobs. Without the bitline interleaving scenario, the traditional write and read solo

assists can improve the V_{MIN} across design parameters. However, the worst-case V_{MIN} in 6T high-density FinFET SRAMs is above the nominal supply voltage, and none of the single peripheral assists can improve it in a bitline interleaving scenario. Only selected combinations such as V_{DD} boosting with wordline boosting, wordline underdrive with V_{DD} collapse, etc. are suitable. Using write and read combined peripheral assists, the overall static V_{MIN} reduces to have a minimum value across double assist percentages. On the other hand, we show that without bitline-interleaving scenarios the 10%_10% combination of WLB + NBL beats all other solo and dual combinations for write V_{MIN} improvement. Using the best CPA combination (NBL + VDB with 14%_6% proportions) projects the system level battery-life improvements to 31.37% using an SR416SW battery. Thus, selecting the right peripheral assist combinations with appropriate percentages could help improve energy lowering in SRAMs, as well as improving the SoC level battery-life in battery operated IoT applications.

3.0.20 Acknowledgments

We thank Farah and Harsh for laying out the ground-work for combined peripheral assist in 130nm bulk and sub-20nm FinFET technology. I am thankful to Ningxi for developing the sense amplifier, wordline boosting circuit, layout integration of the SRAM, and testchip measurements. I thank Harsh for designing the assist controller and frequency to digital converter blocks and the integration of the testchip. We are thankful to NVIDIA and DARPA for funding this project.

Table 3.1: Comparison table for 256kb SRAM testchip with the state-of-the-art (© 2017 IEEE).

	This work	**ISSCC '10** [57]	**ISSCC '12** [58]	**VLSI '14** [59]	**ISSCC '15** [60]
Technology	130nm	45nm	22nm	180nm	28nm
Cell type	6T	8T	6T	8T	6T
Capacity	256kb	512kb	576KB	16KB	256kb
DVS range	1.2-0.38V (850mV)	1.2V-0.57V (630mV)	1V-0.625V (375mV)	1.8V-0.6V (1200mV)	0.9V-0.58V (320mV)
V_{MIN}	0.38V	0.57V	0.7V	0.6V	0.58V
V_{MIN} improvement	240mV	330mV	175mV	-	280mV
Sub-threshold operation	Y	-	-	-	-
Active power	18mW @1.2Vand 12.6W @ 0.38V	169mW @1.2V	-	250mW @ 1.8V, 15mW @0.82V	-
Active power reduction using CPA	337X	-	-	16.4X	
Leakage power reduction to 0.38V V_{MIN}	12.4X	9.4X	-	-	-

Chapter 4

Theoretical Perspective of Reverse Assist-based Canary SRAM for SRAM Dynamic V_{MIN} Tracking

4.0.1 Motivation

[1]SRAM energy has a quadratic relationship with the supply voltage. Thus, scaling supply voltage reduces SRAM energy. State-of-the-art circuit and architectural methods reduce SRAM supply voltage to lower energy consumption, such as dynamic voltage and frequency scaling (DVFS), using dual rail design for SRAMs, etc. Widely used DVFS in the system on chips (SoCs) lowers the energy consumption [61] [62] [63] by altering the supply voltage and frequency from time to time, as required. The design cost for DVFS in the SoC level excludes from the SRAM design cost. On the other hand, the dual rail [64] design keeps SRAM cores at a higher supply while periphery runs at a lower supply for energy savings. Thus, the dual rail design and avoids readability issues in lower core supplies. Nevertheless, this technique complicates the SRAM design implementation and increases its design cost

[1]This chapter is based on the published paper titled "A reverse write assist circuit for SRAM dynamic write V_{MIN} tracking using canary SRAMs" [AB3].

area cost for SoCs. Notwithstanding the voltage lowering techniques mentioned above, the minimum operation voltage (V_{MIN}) of SRAM creates a bottleneck for voltage scaling of SRAMs and other digital blocks sharing the same power rail. The V_{MIN} of SRAM depends on the process, temperature, operating frequency, etc. variations, which is hard to predict in a fabricated chip in real time. Therefore, designers use voltage and timing guard-bands. Moreover, local and global variations affect the scaling of SRAM V_{MIN} more than the logic V_{MIN} [56] [65], and existing research work shows that SRAM write failures will increase with further technology scaling [65]. One of the solutions for improving the SRAM write and read V_{MIN} is to use bias-based peripheral assist circuits. Examples of write assists are wordline boosting [66] [31] [32], negative bitline [66] [31] [32] [67], V_{DD} lowering [31] [32], V_{SS} raising [31] [32], etc., for write improvement. On the other hand, examples of read assists are wordline under drive [32], partially suppressed wordline [67], V_{DD} boosting [32], negative V_{SS} [32], etc., for read improvement. Although assist methods require additional silicon area and have energy overheads, they allow us to lower the SRAM V_{MIN} significantly. Besides, SRAM circuits age [68] [69] [70] with time similar to all other solid state circuits, and the SRAM V_{MIN} gets higher and higher. Thus, the aging effect further adds to the crucial margin of SRAM V_{MIN} for the worst-case designs [71] [72] [58].

Hence, the prediction of V_{MIN} by measuring functional failures during DVFS can enable corrections to functionality issues in SRAMs. Using a closed-loop architecture solution for V_{MIN} prediction one can turn off or on assists or adjust the assist voltage dynamically when needed. Also, a closed-loop control can allow tracking the effects of voltage and temperature variations. So, there is a requirement detecting write or read failure dynamically. In this chapter, we demonstrate the use of canary cells, which detect failures and track SRAM dynamic V_{MIN}.

4.0.2 Prior Art in Canary Circuits

Prior arts show the canary circuit methods in different fields [73] [74] [75]. The SRAMs use the canary circuits by Wang and Calhoun [75] [76] [77] for tracking the data retention voltage (DRV) during standby. However, no prior art presents the canaries in depth for dynamic write or read V_{MIN} tracking. This chapter focuses on the study of canary SRAMs in tracking the dynamic write V_{MIN} of SRAMs, as device scaling degrades a successful write operation more than a read operation [65]. Nevertheless, the results obtained from this chapter is extendable to predict the read V_{MIN}, too.

4.0.3 Peripheral Assist Methods and Reverse Assists (RA)

An SRAM designer can create canary circuits in many ways. One solution is to alter the SRAM core bitcell so that it fails earlier in V_{DD} than the population of the core SRAM bitcells, during write or read operations. However, this type of canary bitcell may not track the same as core bitcells over parameter variations. Employing a built-in control in the canary bitcells for tuning the canaries to change the write-read failure point post fabrication is another option. Thus, one way to realize the tunable control in canaries is to use a circuit that modulates using a shorter wordline pulse-width, to make the write or read operation harder, which fail them earlier than the core SRAM bitcells start to fail. However, to control the wordline pulse-width precisely requires additional delay control circuits in the wordline driver, which may not be realizable across design parameter variations and may increase the area overhead and cause abutting problems in the SRAM layout. Thus, an easily controllable peripheral circuit with least overheads is desirable as a weakening knob for the canary write or read operations, which makes the canaries to fail earlier than SRAM bits.

As discussed earlier, a peripheral assist in the context of SRAMs is an auxiliary circuit, which improves write-ability [64] [66] [31] [32] [67], readability [31] [32], and read-stability [64]. A reverse assist (RA) defines as an auxiliary circuit that degrades the write-ability or

readability of an SRAM bitcell. Here, we use the same core SRAM bitcell in a canary SRAM, but canaries use a reverse assist to degrade the write-ability of its bitcells.

The benefits of using a reverse assist for a canary SRAM are two-fold: 1) we use the same SRAM core bitcells as canaries to track the core cells better and 2) a user can fine-tune the failure point of canaries dynamically after fabrication.

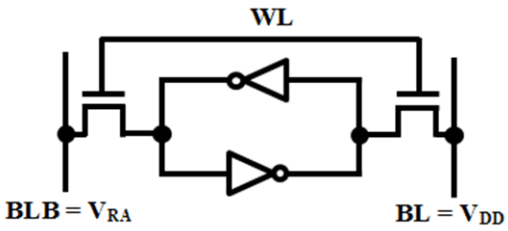


Figure 4.1: SRAM bitcell during write using bitline (BL) type reverse assist (© 2014 IEEE).

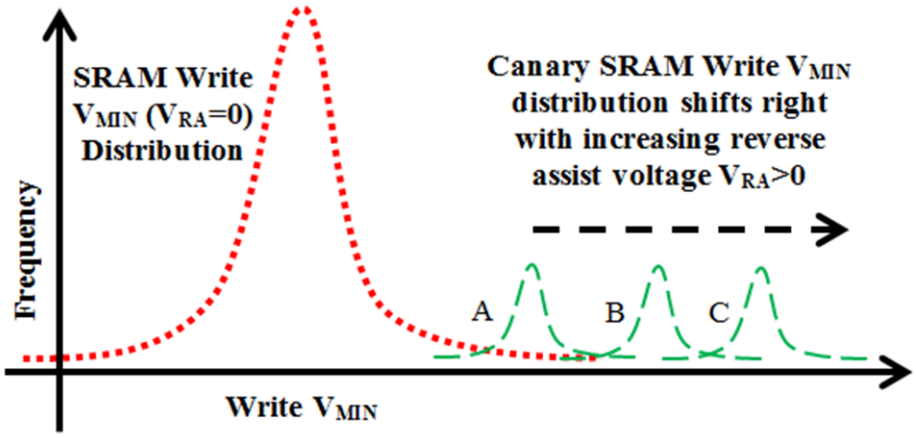


Figure 4.2: SRAM write operation using BL type reverse assist and write V_{MIN} distributions with reverse assist (A, B, C's are canary V_{MIN} distributions) (© 2014 IEEE).

In the context of this chapter, we refer peripheral assist or reverse assist to a bitline (BL) type assist or reverse assist, unless otherwise specified. Without peripheral assists, in core SRAMs during a write, either BL or BLB pulls down (Figure 4.1) to $V_{RA} = 0V$, while the other node (BLB or BL) floats at V_{DD}, and the wordline (WL) turns on. A peripheral write assist using BL signal [66] [31] [32] [67] improves the dynamic write-ability of the SRAM bitcells by pulling the bitline or bitline bar (BLB) node below the ground (V_{SS}) potential. On the contrary, in a canary SRAM, a reverse assist pulls the BL or BLB node to a positive

voltage, such as $V_{RA} = 0.1V$, while the other BLB/BL floats at V_{DD}, which we show in Figure 4.1. Therefore, a BL type reverse assist degrades the dynamic write-ability of the canary bitcells failing them earlier in V_{DD} than the SRAM bits start to fail. In other words, the distribution of canary failures shifts to the right, as shown in Figure 4.2, employing a reverse assist.

4.0.4 Effect of Reverse Assist on Canary SRAMs

The write-ability in an SRAM bitcell defines the ability to perform a write operation. The two widely used metrics for write-ability are 1) write static noise margin (WSNM), known as the metric for static write-ability; and the 2) critical wordline pulse width for write (T_{CRIT}), known as the metric for dynamic write-ability in SRAMs. WSNM assumes an infinite wordline pulse-width, which overestimates the static write-ability metric. However, the T_{CRIT} assumes a finite wordline pulse-width, due to the reason that the SRAM write operation is a finite transient process. An SRAM write assist improves the spread of the distribution of T_{CRIT} and to decreases the V_{MIN} to a lower value [31]. Thus, using a reverse assist in the canary bitcells relative to the core SRAM cells results in the canary write V_{MIN} distribution 'A' to shift to a higher V_{MIN} distribution 'B' or 'C'; as shown in Figure 4.2. Therefore, the V_{MIN} of the canary SRAM increases to make canary failures earlier in V_{DD} than the core SRAM bitcells during voltage scaling.

We target to make the canary SRAM start to fail before a single bit failure happens in an SRAM of a given capacity in bits (for example, a million bits). Figure 4.3 depicts the plots for the simulated probability of dynamic write failure (P_{fail}) vs. write V_{MIN} for the core SRAM bitcells and the canary bitcells using varying degrees of reverse assist. For the experimental setup, we use an extracted netlist of a 6T bitcell, as shown in Figure 4.1 and simulate the transient write operation using a 28nm commercial technology with HSPICE. We extracted the data for P_{fail}-V_{MIN} using an importance sampling algorithm [56] [78] [79] [80]. We use a FO4 delay table data across voltages, for the input slews and the timings of the

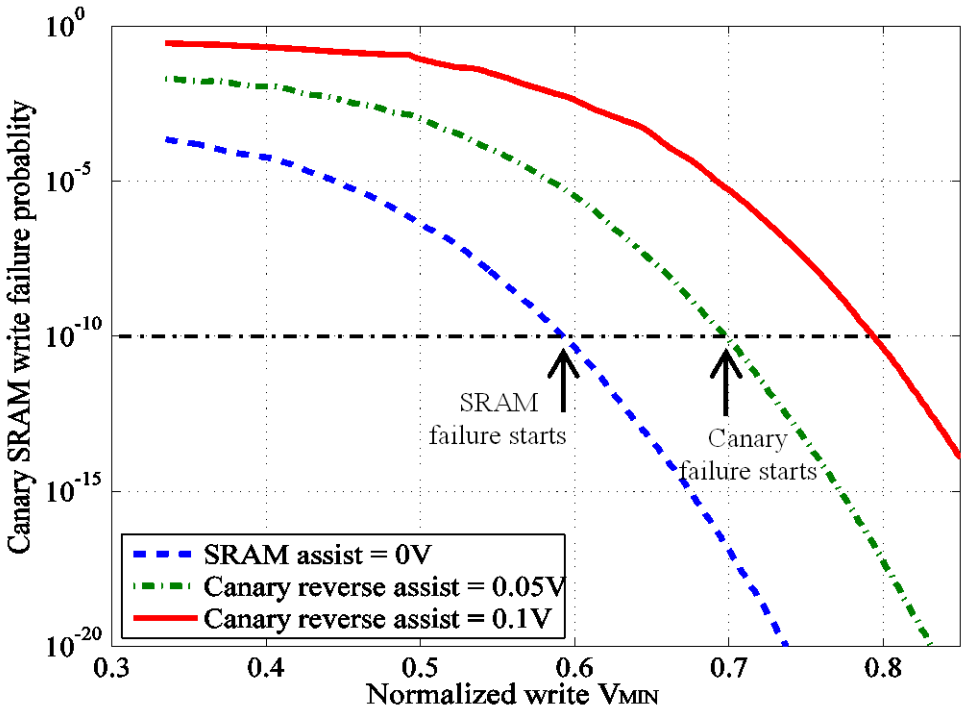


Figure 4.3: Canary SRAM dynamic write failure probability vs. normalized write V_{MIN} (© 2014 IEEE).

WL pulse-width. Noticeably, for $P_{fail} = 10^{-10}$, the canary SRAM employing reverse assist has a higher write V_{MIN} compared to the core SRAM without any assist.

4.0.5 Research Questions

We plan to investigate the following research questions in this chapter.

- What are the input and output design knobs for canary SRAM?
- How do the input and output design knobs for canary and core SRAM relate to each other?
- What are the trends for canary design knobs?
- What could be the possible architecture and operating principle of canary SRAMs using reverse assists?
- What are the power-area trade-offs for canary SRAM implementation?

4.0.6 Proposed Approach

We propose the following approach listed in bullets to investigate the research questions mentioned above.

- We define the core and canary SRAM input and output parameters, as shown in Table 4.1.
- We derived two mathematical equations, (4.1) and (4.2), to relate the core and canary SRAM input and output parameters.
- We plot the canary SRAM trends across design knobs, with some assumptions of a corresponding core SRAM design.
- Besides, in [81] we show a canary SRAM architecture and an algorithm to track SRAM V_{MIN} that turn on assists for SRAM based on the canary F_{th} value. Using the proposed algorithm, we turn on assists or increase the supply voltage if the canary failures are more than the F_{th} condition. A user can choose the F_{th} condition post-fabrication, also.
- We derive the area and power trade-offs for canary implementation using some assumptions.

4.0.7 Evaluation Metrics

To evaluate the canary SRAM, we propose the framework above to map the SRAM related matrices to the canary metrics, which one can use to tune the canary failure point and thus adjust the SRAM guard-bands for reducing the margins. Furthermore, the canary chip failure probability metric (P_{f_c}) is a core and canary SRAM combined metric, which shows that the canaries may not fail before a population of core SRAM bitcells starts to fail. Other evaluation metrics are listed as follows.

- Power overhead
- Area overhead
- Percentage energy improvement across corners

4.0.8 Canary SRAM Input and Output Design Metrics

Table 4.1: Input and output design metrics for the canary SRAM design (© 2014 IEEE).

Input metrics	
N	Number of SRAM bits on a chip
Y_{SRAM}	Core SRAM target yield
C	Number of canary SRAM bits
F_{th}	Canary failure threshold condition
V_{RA}	Canary BL type reverse assist voltage
Output metrics	
P_{f_c}	Canary SRAM chip failure probability

Table 4.1 gives the input and output design parameters for the canary SRAM design. The probability of write failure for the core SRAMs depends on the number of SRAM bits (N) on a chip with a target yield (Y_{SRAM}). Thus, using the canary failure probability to track dynamic write failure of core SRAM bits demands a certain number of canary bits (C). Other crucial input design parameters are the canary failure threshold condition (F_{th}) and the reverse assist voltage (V_{RA}) for tracking SRAM write failures. The F_{th} condition defines the number of canary bitcells allowed to fail before one in N SRAM core bits fails. As an example, if a user assumes F_{th}=8 for C=32 number of canaries in a chip, then one can take action if 8 canaries fail to write out of 32 canaries. As an action, the user can either turn on assists for the core SRAM or stop further voltage scaling in a DVFS scenario. The V_{RA} potential can control the degradation of the write-ability in canaries. A user has two input metric knobs available for fine-tuning canaries post-fabrication, such as V_{RA} and F_{th}. The SRAM designer sets all other input metrics at the time of canary design. The output metric defines P_{f_c} as the canary SRAM chip failure probability, which gives the probability of the canary SRAM to be unable to fail earlier than one in N SRAM core bitcells. As an example, if $P_{f_c} = 10^{-6}$ for a given $N = 10^7$ SRAM bits with $Y_{SRAM} = 99\%$, $C = 32$, and $F_{th} = 1$, then the canary chip failure probability denotes the following. In one in a million 10Mb SRAM

chips, the 32 canary cells will not experience a single bit failure before a single SRAM bit start to fail among the ten million SRAM bits on the chip.

We assume that the bit failure probability of the core SRAMs in a write operation is given by P_f. Thus, the success probability of the core SRAM bitcell is given by $P = (1 - P_f)$, and we can write the success probability of the SRAM chip as $P_{chip} = P^N$. Therefore, the chip failure probability of the SRAM is given by $P_{f_{chip}} = (1 - P_{chip})$. The equation (4.1) gives the SRAM chip yield for 'k' or fewer chip failures out of 'J' chips.

$$Y_{SRAM_{(J,k)}} = \sum_{i=0}^{i=k} P_{chip}^{(J-i)} * (1 - P_{chip})^i * \binom{J}{i} \tag{4.1}$$

$$P_{f_c} = \sum_{i=0}^{i=k} p_f^i * (1 - p_f)^{(C-i)} * \binom{C}{i} \tag{4.2}$$

Using (4.1), for a given Y_{SRAM} and N values, we calculate the corresponding bit failure probability P_f for the SRAM write failures. Similarly, we assume if p_f denotes the canary bit failure probability, then the equation (4.2) gives the probability of C number canary bits with $F_{th} = k$ condition being unable to fail earlier than a given number of core SRAM bits. Thus, the equations (4.1) and (4.2) relate the input metrics N, Y_{SRAM}, P_f, C, F_{th}, and p_f to our final output metric, which is canary chip failure probability P_{f_c}.

4.0.9 Calculation Methodology for Canary Chip Failure Probability

Figure 4.4 depicts the methodology for the calculation of the metric named canary bit failure probability p_f using SRAM bit failure probability P_f. Figure 4.4 shows the plots for P_{fail} vs. V_{MIN} for the core SRAM bitcells without any assist, and canary SRAM P_{fail} vs. V_{MIN} with reverse assist. For the given SRAM design parameters of Y_{SRAM} and N, the research questions we are addressing here are as follows. 1) What is the SRAM bit failure probability P_f for the parameters of Y_{SRAM} and N, and 2) what should be the corresponding

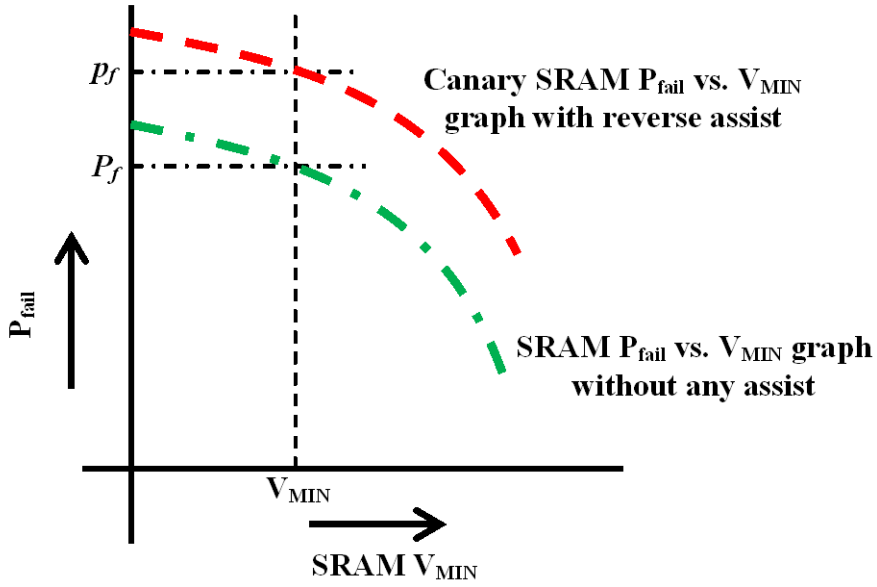


Figure 4.4: Methodology to calculate canary chip failure probability (© 2014 IEEE).

bit failure probability p_f for the canary SRAM bits? Additionally, we want to know how the input metric C influences the canary chip failure probability P_{f_c}. To relate the two equations, (4.1) and (4.2), we first extract the P_{fail} vs. V_{MIN} data for different voltage values of the reverse assist, which represent the data for the canary bitcells. Also, we generate the P_{fail} vs. V_{MIN} data for the core bitcells without any assist. At first, we calculate the corresponding failure probability (P_{fail}) P_f for the core bitcells using (4.1), and then we calculate the corresponding V_{MIN} of the core bitcells using the P_{fail} vs. V_{MIN} simulated data (Figure 4.4). Then, we calculate back the corresponding P_{fail} p_f for the canary bitcells with the same V_{MIN} value extracted from the canary SRAM P_{fail} vs. V_{MIN} simulated data, which is shown in Figure 4.4. We put the value of p_f into equation (4.2) to get the corresponding canary chip failure probability P_{f_c}, finally.

4.0.10 Results

To obtain the trends for the variation of the input vs. the output metric, we use the calculation method described in [81] and get the output metric (P_{f_c}) for the reverse assist

voltages of V_{RA}=0V, 0.05V, 0.1V, 0.15V, and 0.2V. Figure 4.3 depicts that one can achieve the same canary chip failure probability of $P_{f_c} = 10^{-5}$ by either increasing the number of canaries to $C = 512$ with a lower V_{RA}=70mV or decreasing it to $C = 8$ with a higher V_{RA}=170mV. To get the trends of C vs. N, C vs. Y_{SRAM}, and C vs. F_{th}, for a constant $P_{f_c} = 10^{-5}$ and for different values of V_{RA}, we interpolated the data obtained for V_{RA} in between known V_{RA} values.

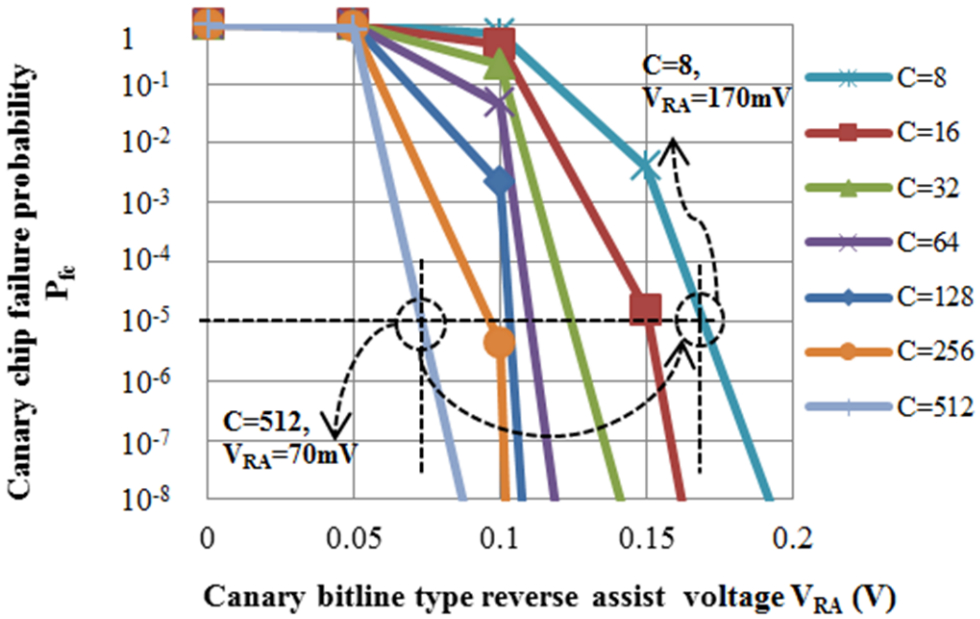


Figure 4.5: Canary chip failure probability vs. reverse assist voltage for 1 million SRAM bitcells with 95% yield at TT_85C (© 2014 IEEE).

We show the trend of the number of canary bits C vs. the number of SRAM bits N in Figure 4.4. We observe that increasing N two orders of magnitude from 1 million to 100 million bits, results in the number of canaries C required to maintain the same canary chip failure probability of $P_{f_c} = 10^{-5}$ (at different reverse assist voltages) to double. We also show the trend of the number of canary bits C vs. SRAM yield Y_{SRAM} in Figure 4.5. We observe that to keep the same canary chip failure probability of $P_{f_c} = 10^{-5}$, while increasing the SRAM yield from 99% to 99.99%, the number of canary bits increases by 8X from $C = 64$ to $C = 512$ for the V_{RA}=126mV BL type reverse assist.

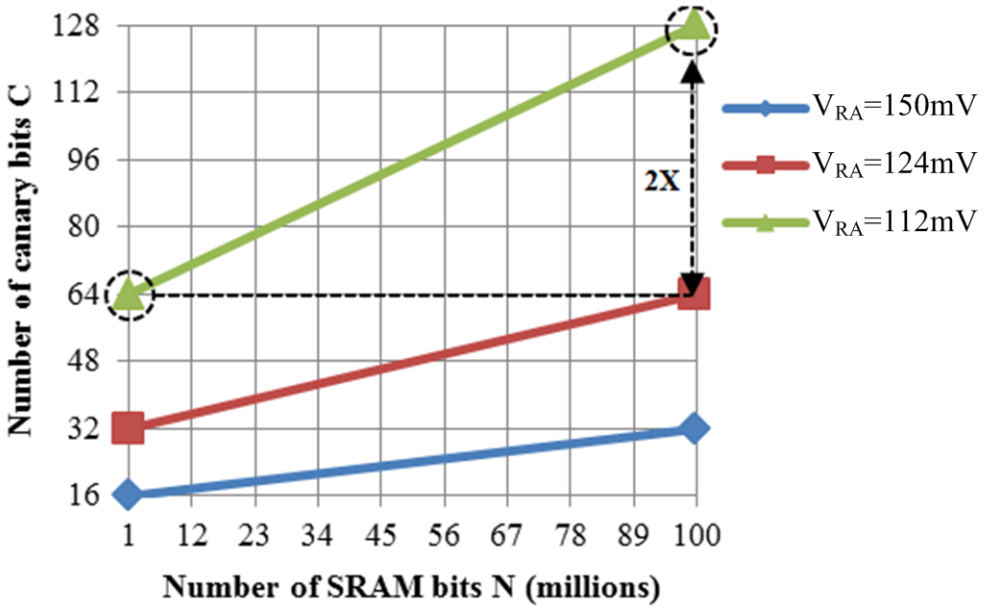


Figure 4.6: Trend for C vs. N with 95% SRAM yield at constant $P_{f_c} = 10^{-5}$ for different V_{RA} voltages at TT_85C (© 2014 IEEE).

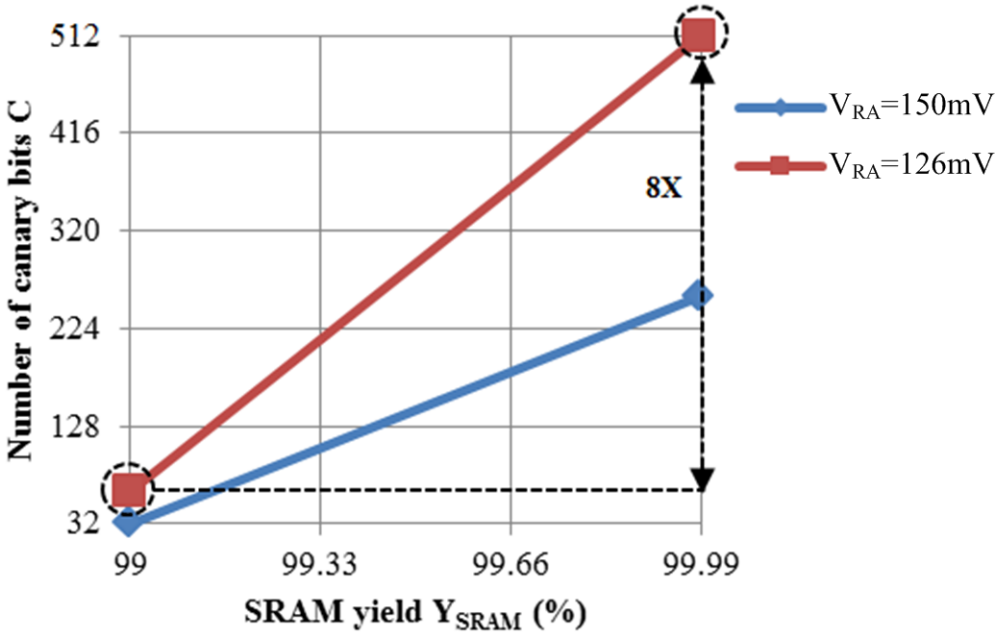


Figure 4.7: Trend of C vs. Y_{SRAM} with 100 million SRAM bitcell at constant $P_{f_c} = 10^{-5}$ for different V_{RA} voltages at TT_85C (© 2014 IEEE).

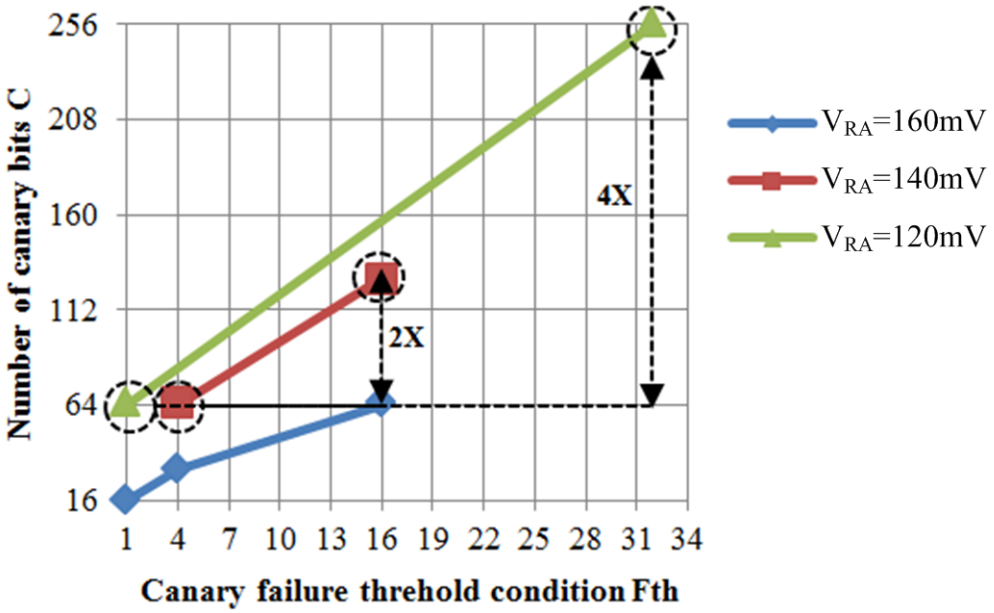
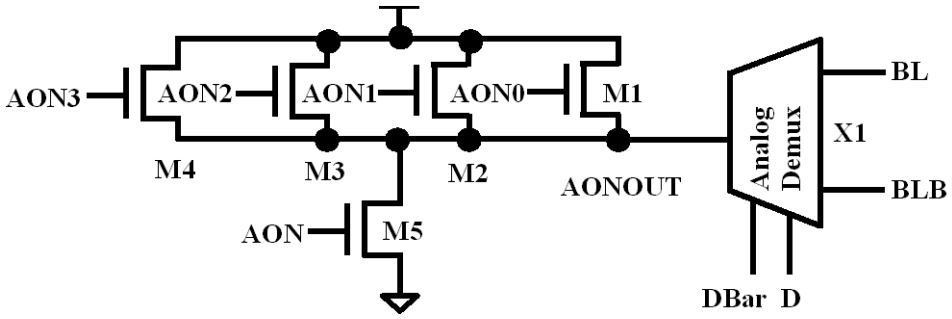


Figure 4.8: Trend of C vs. F_{th} with 100 million SRAM bitcell at constant $P_{f_c} = 10^{-5}$ for different V_{RA} voltages at TT_85C (© 2014 IEEE).

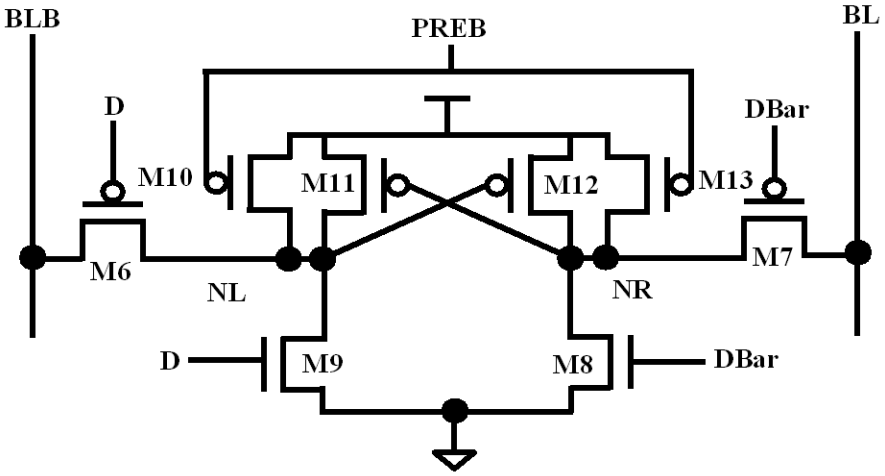
Similarly, Figure 4.8 plots the trend of the number of canary bits C vs. canary failure threshold condition F_{th}, while keeping other input metrics constant. The Figure 4.8 reveals that to maintain the same canary chip failure probability roughly at $P_{f_c} = 10^{-5}$ with V_{RA}=140mV, increasing the failure threshold from F_{th}=4 to F_{th}=16, requires 2X more canary cells compared to that of the C=64. For the reverse assist voltage of V_{RA}=120mV, a change of 32X in F_{th} condition requires a 4X increase in C from C=64 to C=256 maintaining the same $P_{f_c} = 10^{-5}$.

4.0.11 Circuit Implementation of BL type Reverse Assist

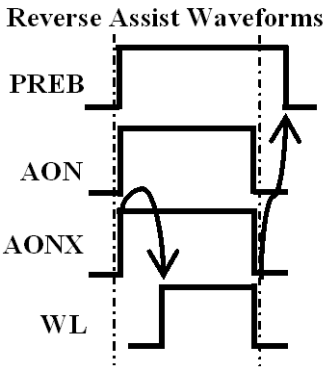
Here, we assume that a reverse assist is an integral component of the existing core SRAM I/O as canary I/O, which requires additional circuitry. One can realize a BL type reverse assist by using a positive charge pump, an analog closed loop voltage reference, a voltage divider circuit, etc., for the generation of the reverse assist voltage. An analog charge pump and a closed loop variable voltage reference could result in much higher design and area overhead per canary I/O. Besides, we simulate and observe that a PMOS-NMOS voltage



(a)



(b)



(c)

Figure 4.9: (a) Canary SRAM reverse assist circuit. (b) Canary write driver. (c) Reverse assist waveforms (© 2014 IEEE).

divider incurs much higher variation in output voltage compared to an NMOS-NMOS voltage divider. To implement a write operation in the canary SRAM, we propose a novel reverse assist (Figure 4.9a), which supplies the positive bias voltage (V_{RA}) for BL/BLB signals. The canary write driver (Figure 4.9b) pulls up the other node BLB/BL to V_{DD} during a write operation. Here, the name AONX cumulatively represents the signals AON0, AON1, AON2, and AON3, as shown in Figure 4.9c. Figure 4.9a depicts that the signals AON and AONX create the V_{RA} at node AONOUT by selecting M5 and M1-M4, accordingly. Here, the AONOUT node either connects to BL or BLB, which uses an analog de-multiplexer X1 controlled by D/DBar signals. During a canary write operation using reverse assist, D or DBar turns on M9/M8 to pull down one of the NL/NR nodes to the ground, as shown in Figure 4.9b. Thus, the nodes NR/NL pulls up accordingly, through cross-coupled M12 and M11 transistors. Nevertheless, only the pulled up node NR/NL connects to the desired BLB/BL node by M7 or M6. Therefore, M6 and M7 disconnect the internal pulled down node NL/NR from BL/BLB by turning off M6 or M7, accordingly. Thus, M6 or M7 enables to connect the reverse assist voltage node AONOUT to BL/BLB node, using the analog de-multiplexer X1. We size the analog demultiplexer, M1-M5, sufficiently to discharge the BL/BLB and to support the generation of a minimum of 50mV and a maximum of 200mV of BL type reverse assist in a write operation.

4.0.12 Block Diagram of Canary SRAM Architecture and an Algorithm to track SRAM V_{MIN}

To implement the circuit proposed in this chapter, we develop a canary SRAM architecture and an algorithm in this Section, which track the core SRAM V_{MIN}. Figure 4.10 shows the block diagram of the proposed canary architecture. The architecture of the canary SRAM has the canary I/Os, canary control, and a single row of canary bitcells distributed in the left and right banks. Figure 4.10 show that the canary SRAM block abuts to the core SRAM macro. The core SRAM consists of two SRAM core arrays, a decoder, I/Os, and an SRAM

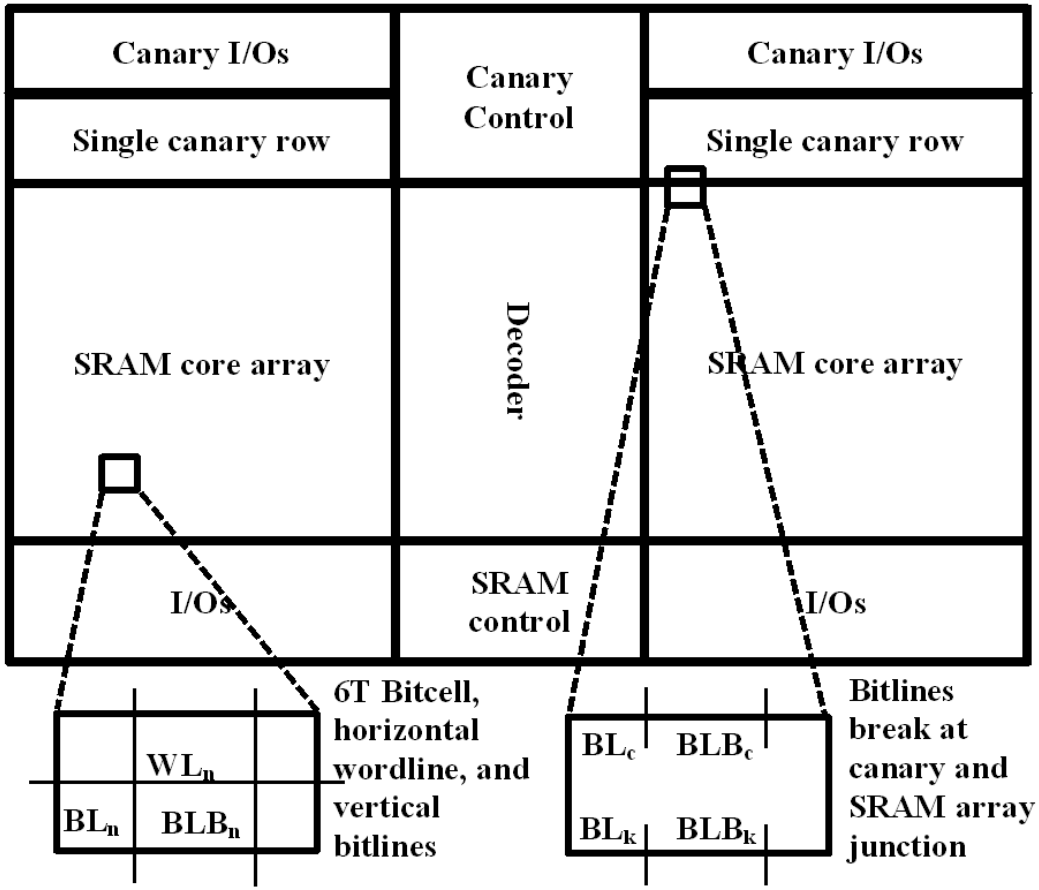


Figure 4.10: Block diagram of the canary SRAM inside SRAM macro (not in scale) (© 2014 IEEE).

control logic. As per the architecture, the canary control can communicate directly to the core SRAM control logic. The canary and core SRAM wordlines orient horizontally, and bitlines orient vertically. Operating the canary SRAM independently of the core SRAM requires the bitlines break at the junction of the core SRAM array and canary row. Besides, a designer can place the canary SRAM away from the core SRAM macros. In case of integrated canaries in all core SRAM macros, the canaries can track local and global fluctuations in voltage on the power grid, frequency, temperature, process variations, and aging effects in a large SoC. However, a standalone single canary SRAM macro can only track the impacts of global variation across process corners, aging, etc. in an SoC. The Figure 4.9a shows a reverse assist circuit, which appears inside each I/Os of the canary SRAM. To lower the effects of local variation, the AONOUT signal shares among the canary I/Os, as shown in Figure 4.9a.

We show the proposed algorithm in Figure 4.11, which tracks the SRAM V_{MIN}. At first, the Canary Control logic State Machine (CCSM) begins with an initial setting of V_{RA} on boot up. The initial V_{RA} setting depends on a couple of parameters, such as the SRAM V_{MIN} at a certain process corner, the size of SRAM (N bits), the number of integrated canaries C, a constant P_{f_c}, etc. as shown in Figure 4.5. The CCSM waits for a user signal 'S' post-applying the initial V_{RA} setting. If the user enables canary operation and turns on the signal 'S,' then the CCSM writes a user-defined word in the first cycle into the canary rows and reads it back in the second cycle from the canaries for comparing with the word written. A successful write operation in canaries results in less than or equal to F_{th} number of canary failures. If the canary failures exceed more than the F_{th} value, it denotes a canary write failure, which indicates the scaled voltage reaches the SRAM V_{MIN}. After canary failure, lowering the SRAM V_{DD} further by voltage scaling could result in an imminent SRAM failure. On a canary write failure, the CCSM can communicate to the DVFS control logic in the SoC to stop further voltage scaling, or take a user-defined action, such as halting the access to the SRAM for a couple of cycles or turning on a peripheral assist for SRAMs, etc. A successful canary write operation allows further voltage scaling or a user can turn off the

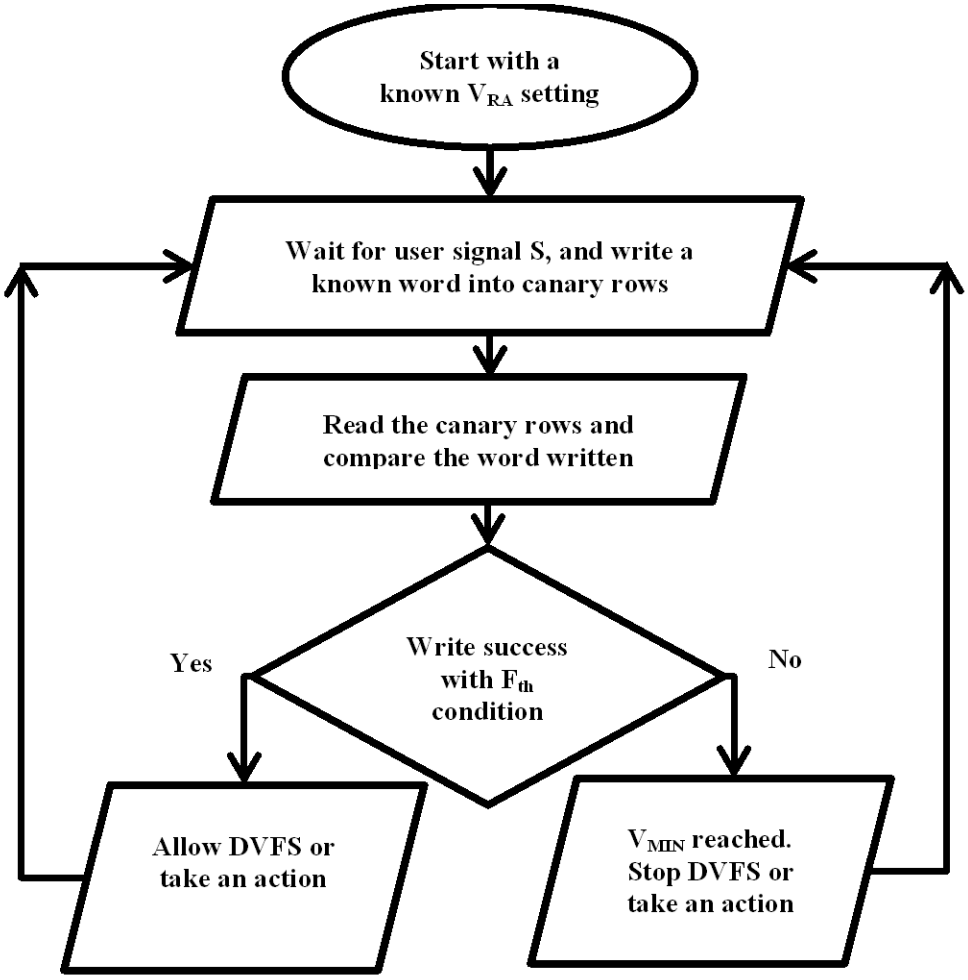


Figure 4.11: SRAM V_{MIN} tracking algorithm using canary SRAMs with reverse assist (© 2014 IEEE).

SRAM peripheral assists. Therefore, the algorithm can enable the tracking of the SRAM V_{MIN} for each core SRAM with in-built canary SRAM. Besides, the CCSM can quantify the number of bit failures in canaries to set the F_{th} value, accordingly. Furthermore, a user has an option to update the initial V_{RA} setting using an on-chip temperature or aging sensor or based on simulation data of P_{f_c}, which tracks the write V_{MIN} more precisely across years of operations.

4.0.13 Power and Area Tradeoff for the BL type Reverse Assist Circuit in a Canary Write Driver

We calculate the active power and area tradeoff numbers for P_{f_c} with the assumption that the total number of SRAM bits N is 100 million in an SoC, and the required SRAM yield Y_{SRAM} is 99%. For other metrics, we calculate tradeoff numbers using some assumptions of the layout width of the wordline driver, I/O height, average bitline energy, bitcell energy per bit, etc. parameters. Note that a canary SRAM independent of located adjacent or far away from a core SRAM macro, will not be able to track local variation of voltages in the power bus, frequency, and temperature in all 100 million core SRAM bits. This is due to the reason that the 100 million SRAM bits located all over the SoC, and the voltage, etc. fluctuations will vary from point to point across the SRAM macros in the SoC die. One can divide this total of N number of SRAM bits into an M number of equally or unequally sized SRAM macros. Thus we quantify the effect of area and energy penalty of canaries vs. the average size of SRAM macros.

For an integrated canary SRAM inside a core SRAM macro, assuming column mux (CM) 4 scenario, it requires the same number of canary I/Os as the number of SRAM I/Os to make a rectangular-shaped symmetric core-canary SRAM macro, as shown in Figure 4.10. Thus, the C depends on the number of I/Os in the SRAM. On choosing a logical macro size of 128 words, 64 bits with CM 4 (128x64x4), a designer has to use C=64x4=256 number of canary bits. Hence, the size of the SRAM block determines the number of canaries in an

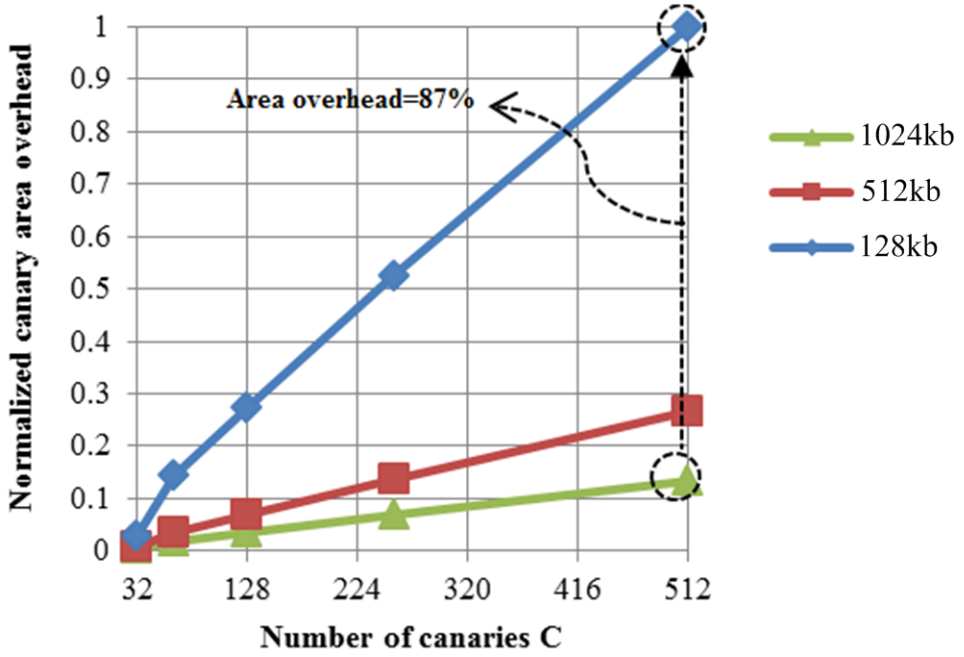


Figure 4.12: Normalized canary area overhead vs. number of canaries for different SRAM sizes (© 2014 IEEE).

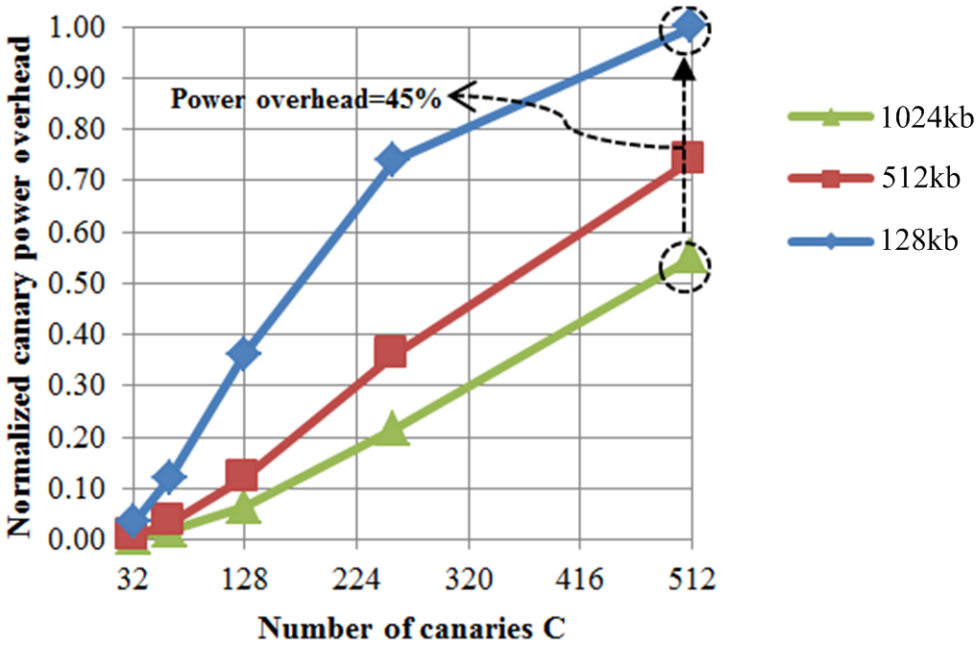


Figure 4.13: Normalized canary total power overhead vs. number of canaries C with constant V_{RA}=50mV for different SRAM sizes at 1GHz TT_85C corner (© 2014 IEEE).

integrated canary SRAM macro. Besides, a designer can use standalone canary SRAMs of user-defined size in between core SRAM macros. Nevertheless, canary I/Os have a much bigger area, relative to the canary bitcells, which increases the overall canary SRAM area penalty. We depict in Figure 4.12 that increasing the number of canaries increases the area overhead. Using the same C=512 canaries, a smaller macro of 128kb (128 I/Os with CM=4) size has 87% more area overhead compared to a bigger one of 1024Kb size. Thus, we can trade off SRAM area to track the V_{MIN} better of smaller SRAM macros across an SoC. Using the same C=512 number of canaries, Figure 4.13 plots that the 128kb SRAM macro has roughly 45% higher total active power overhead compared to a 1024Kb SRAM macro, which uses the same V_{RA}=50mV at the TT_85C corner at the 1GHz operating frequency. On the other hand, we show in Figure 4.14 that, if we change V_{RA}=50mV to V_{RA}=150mV, the power overhead for canaries in SRAMs increases by 30% for a 512Kb macro, which uses C=512 at the 1GHz operating frequency. Figure 4.15 and Figure 4.16 depict the P_{f_c} vs. power cost and area cost for different values of F_{th}. We observe that at TT_85C corner using C=128 and $P_{f_c} = 10^{-4}$ at the 1GHz frequency, if we increase the F_{th} condition from F_{th}=1 to F_{th}=32, it hikes up the power cost by 25%, and the canary I/O area cost by 1%.

Finally, Figure 4.17 show the simulated normalized SRAM V_{MIN} for 100 million SRAM bits with 99% SRAM yield at 85C temperature. Our simulation results suggest that one can use canary SRAMs to track the write V_{MIN} of core SRAM bits with a specified confidence. Figure 4.18 show the normalized write energy related to the core SRAM V_{MIN}. We observe that at the TT_85C corner one can operate SRAMs with 36% lower write energy cost compared to the worst-case V_{MIN} at the SF_85C corner, which sets the V_{MIN} guard-band. The lowest energy savings occur at the SS_85C corner, which is 30.7%. Figure 4.18 show that the maximum energy savings happen at the FS_85C corner, which is 51.5% better than the worst-case. Moreover, at the FF_85C corner, one can have up to 42.2% energy savings compared to the worst-case energy at the SF_85C corner.

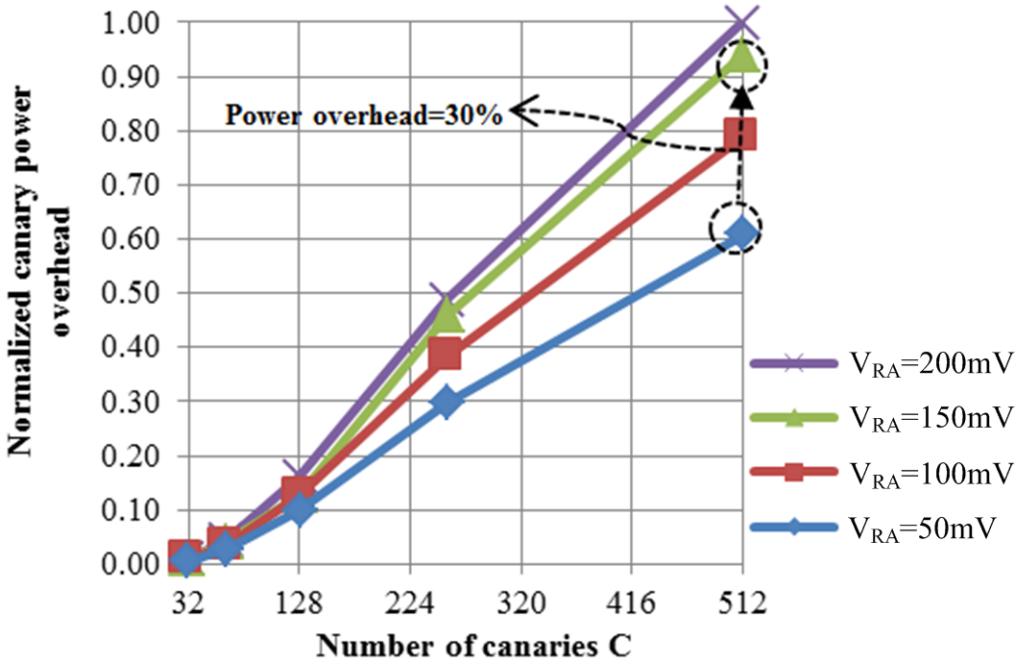


Figure 4.14: Normalized canary total power overhead vs. number of canaries C with N=512kb SRAM for different V_{RA} voltages at 1GHz TT_85C corner (© 2014 IEEE).

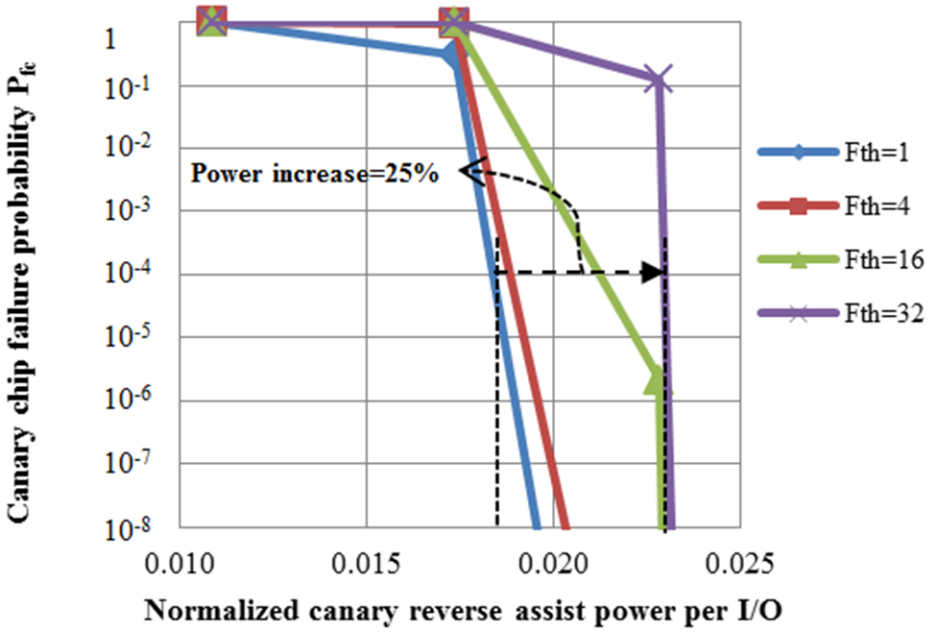


Figure 4.15: Canary chip failure probability vs. normalized reverse assist total power for increasing F_{th} conditions at 1GHz TT_85C corner (C=128) (© 2014 IEEE).

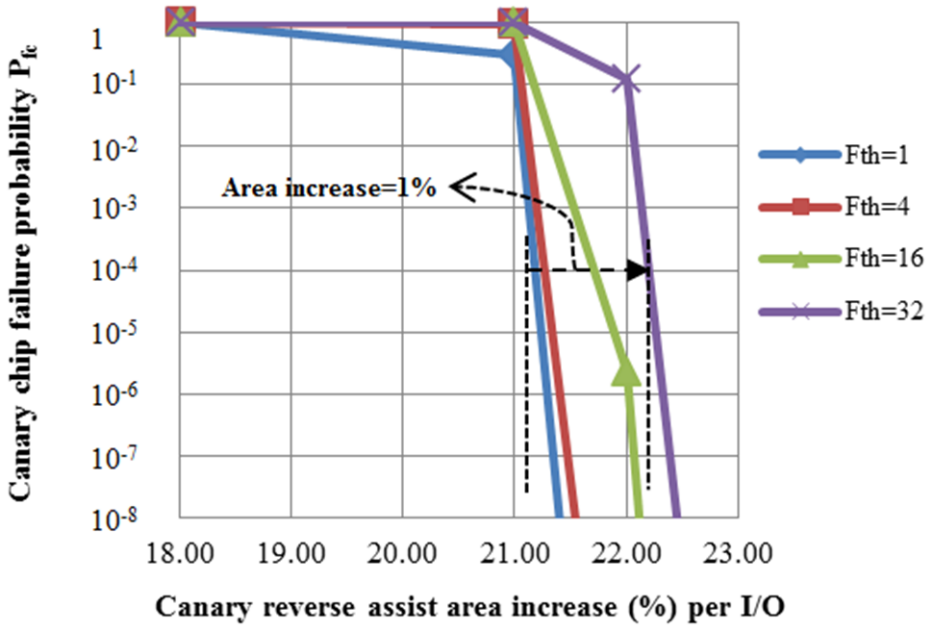


Figure 4.16: Canary chip failure probability vs. canary reverse assist area increase per I/O for increasing F_{th} conditions (C=128) (© 2014 IEEE).

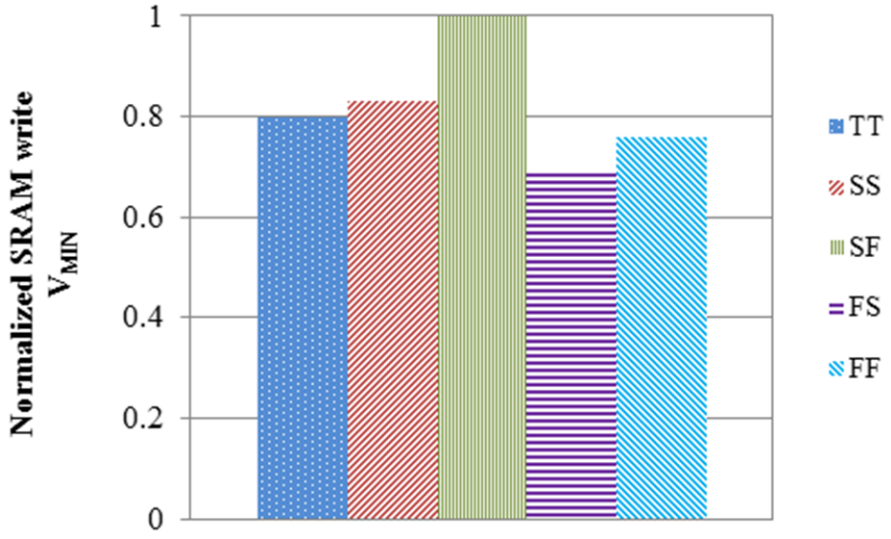


Figure 4.17: Normalized SRAM write V_{MIN} for 100 million SRAM bits with 99% yield constraints at 85C (© 2014 IEEE).

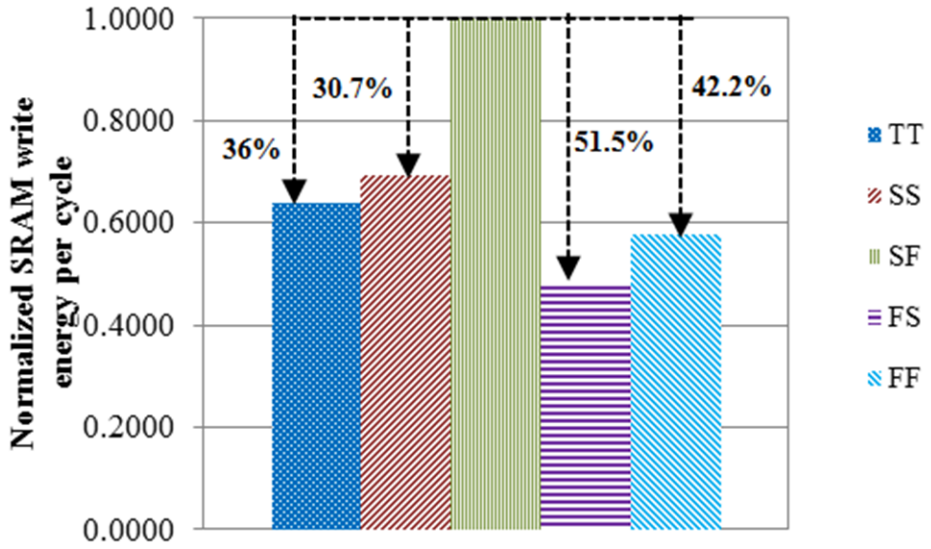


Figure 4.18: Normalized SRAM write energy per cycle at V_{MIN} for 100 million SRAM bits with 99% yield constraints at 85C (© 2014 IEEE).

4.0.14 System Level Projected Savings Using Canary Scheme

This section computes the system level battery-life and replacement time savings for the canary scheme at all the corners. Here, we assume that SRAM consumes 40% energy while the logic core consumes 60% energy. We further assume that the IoT system uses an A1578 (0.76Wh) battery. Thus, the estimated energy savings compared to the SF corner in TT, SS, SF, FS, and FF corners are 36%, 30.7%, 0%, 51.5%, and 42.2%, respectively, as shown earlier. Therefore, the relative energy consumption are 64%, 69.3%, 100%, 48.5%, and 57.8%, respectively. Assuming a 100pJ of energy consumption and 100% duty cycle in the worst-case SF corner, we compute the total system (SRAM and core) power consumption at the 100MHz frequency to be 0.016W, 0.017325W, 0.025W, 0.012125W, and 0.01445W at the corresponding corners. The corresponding battery-life improvements running the IoT SoC fabricated in different corners with respect the worst-case SF corner are 55.86%, 44%, 0%, 105.22%, and 72.45%. Assuming a 500 recharge cycles, the best-case battery replacement time using the A1578 battery will be for the SoC fabricated in the FS corner with 3.54 yrs, and the worst-case battery replacement time would be 1.72 yrs for the SF corner SoC for the SRAM. These projections assume a self-discharge of 10% per month for

the A1578 Lithium-ion Polymer battery. The equations used to calculate the battery-life using self-discharge assumptions are derived in Appendix A for the SR416SW Silver Oxide, LIR2032 Lithium-ion, and A1578 Lithium-ion Polymer batteries.

4.0.15 Conclusions

We conclude that the idea of canary SRAM employing a peripheral reverse assist is a promising solution to predict core SRAM failures due to write-ability issues. Canary SRAM can allow the tracking of SRAM dynamic write V_{MIN} using reverse assist across fluctuation of voltage, frequency, temperature, and aging effects. Also, canaries enable us to take necessary actions, such as turning on assists, stalling memory access, slowing down operating frequency, or boosting supply voltage. In this chapter, we do all of the calculation of bit failure probability using an importance sampling algorithm. Nevertheless, if a designer chooses an incorrect importance sampling distribution, it will mispredict the V_{MIN} resulting in higher energy dissipation or SRAM failures before canaries. The overhead for area and power of the integrated canaries are lower for the bigger SRAMs, which directly depends on the number of SRAM I/Os. One can qualitatively state that the parameter named canary failure threshold condition F_{th} allows the rejection of extreme canary outliers or fault in some bits. Furthermore, canaries help reducing the traditional design-for-the-worst-case SRAM write V_{MIN} in different corners, which saves energy. The battery-life improvement using canary scheme can achieve a maximum of 105.22% in FS corner dies to a minimum savings of 44% in the SS corner dies compared to the worst-case SF corner dies with 0% savings, assuming the SoCs operate at 100MHz. Finally, we conclude that the canary SRAM can track the dynamic write V_{MIN} of a bigger core SRAM in simulation and theory.

4.0.16 Acknowledgments

We thank DARPA and NVIDIA for supporting this research work.

Chapter 5

Characterization of Canary Sensor Properties for SRAM Dynamic Write V_{MIN} Tracking across Voltage, Frequency, and Temperature Variations

5.0.1 Introduction

[1]Scaling down bulk device technology induces an increase in the process variation, which makes the minimum operating voltage (V_{MIN}) of SRAM higher. Peripheral assist [32] techniques reduce the SRAM V_{MIN} for SRAM read and write operations, which has area and power overhead. Prior art [65] show that chances of SRAM write failures are higher than read failures in scaled deep sub-micron technologies. Thus, in a system on chip (SoC), where SRAM shares the same power rails with other digital blocks, the poor write-ability

[1]This chapter is based on the published paper titled "A 130nm canary SRAM for SRAM dynamic write V_{MIN} tracking across voltage, frequency, and temperature variations" [AB4].

of the SRAM can limit the write V_{MIN} (WV_{MIN}) and overall V_{MIN} of the SRAM and the SoC. The SRAM WV_{MIN} depends on process variation, dynamic voltage and frequency scaling (DVFS) [61], and unintended variations in voltage, frequency, and temperature during runtime. Thus, runtime tracking of the SRAM WV_{MIN} over the process, voltage, frequency, and temperature, and operating SoCs at or near the SRAM dynamic WV_{MIN} can reduce margin guard-bands and power consumption.

Authors in [75] [81] show that canary circuits can track SRAM design metrics using closed-loop solutions. Our prior work [81] provides a theory of SRAM WV_{MIN} tracking using canary SRAM bitcells, which uses reverse assists [81]. This chapter documents silicon results to confirm that reverse assist-based canary SRAMs can successfully change its failure rate across design knobs to track the behavior of a core SRAM array. We organize the remaining part of the chapter below. The chapter first describes the concept of peripheral assists and reverse assists briefly. After that, the chapter discusses the architecture of the canary testchip. Then the chapter documents the discussion of the test setup, simulation, and measurement results for canaries across voltage, frequency, and temperature variations, which uses wordline (WL) and bitline (BL) type reverse assists. The chapter further shows the strategy of tuning canaries to fail before the core SRAM bits start to fail and power results and concludes.

5.0.2 Peripheral Assists, Reverse Assists, and Canary SRAM

As discussed earlier, bias-based peripheral assist [32] circuits improve the SRAM WV_{MIN} and enable SRAM read and write operations even at lower supply voltages. On the other hand, a reverse assist [81] weakens an SRAM read or write operation and makes it vulnerable to read or write failures. Thus, using reverse assists, the canary SRAM bitcells [75] [81] become more prone to failure and act as sensors that track the WV_{MIN} of a bigger core SRAM. One of the key conditions of tracking the SRAM WV_{MIN} using canary SRAM is to tune the canaries with reverse assists to fail earlier in V_{DD} than the SRAM bits start to fail [81]. Besides, tracking the SRAM WV_{MIN} across supply voltage, frequency, and temperature

(VFT) variations require the canary SRAM to must have distinct failure trends and failure thresholds for reverse assists across design parameters. Thus, a shift in any of the values of VFT results in a measurable and definite change in the number of canary failures to allow necessary responses, such as turning on or off assists for the core SRAM, raising the V_{DD}, etc. [81]. This chapter shows the simulation and chip results of canary SRAM, which confirm the canaries to have vivid failure trends with different VFT parameters to track SRAM WV_{MIN}. Moreover, the canary failure point tunes with BL and WL type reverse assists, which occur before the core SRAM bits start to fail while scaling down the V_{DD}.

5.0.3 Block Diagram of the Canary SRAM Testchip

We design a testchip for the canary SRAM characterization in a commercial 130nm bulk technology. The chip uses six identical SRAM and canary memory blocks (BKs). Figure 5.1 depicts the annotated die photograph of one of the used BKs in the chip for testing. The GMUX block shown in Figure 5.1 serves as a global multiplexer for buses and control signals to and from all the BKs to the pins of the chip package.

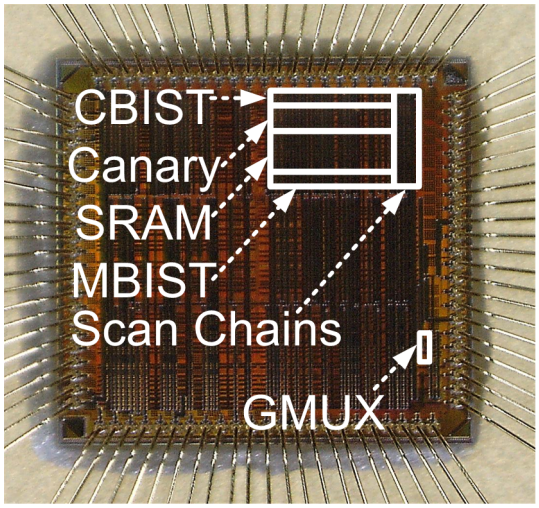


Figure 5.1: Annotated micrograph of the canary SRAM memory block (BK) in the testchip (© 2015 IEEE).

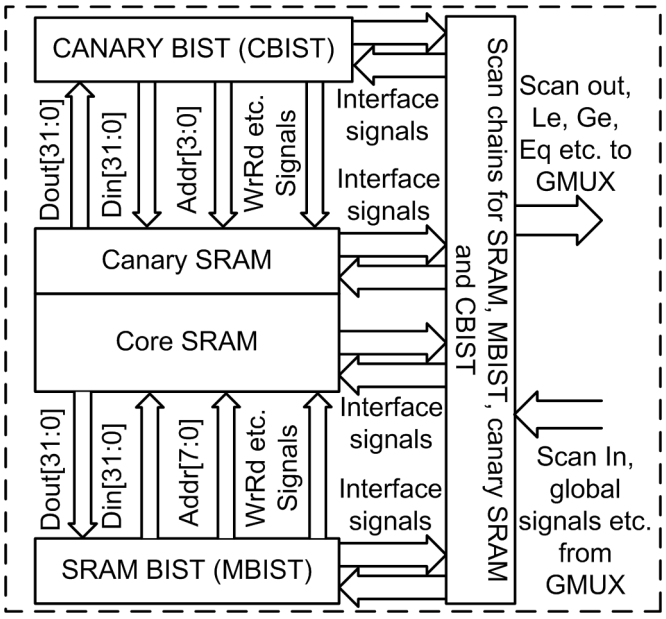


Figure 5.2: Block diagram (not in scale) of the memory block (© 2015 IEEE).

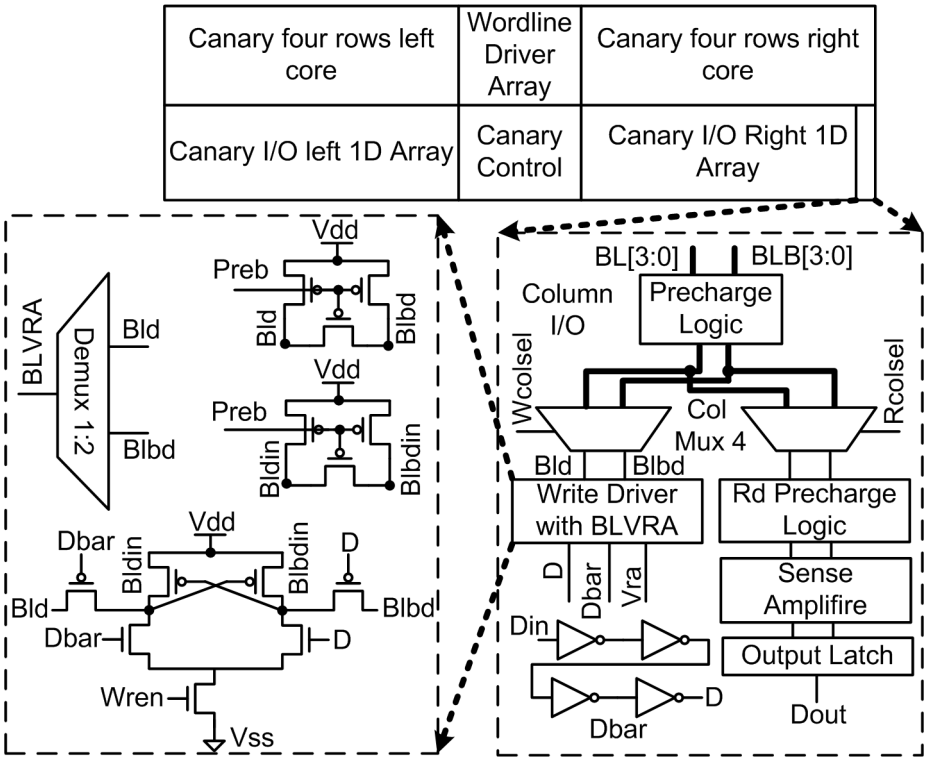


Figure 5.3: Block diagram (not in scale) of the canary SRAM column periphery (I/O) and BL type reverse assist (© 2015 IEEE).

5.0.4 Memory Block Diagram, Canary, and Core SRAM

Figure 5.2 shows the block diagram of the memory block. The memory block consists of an 8Kb core SRAM, a 512b canary SRAM, a memory built-in self-test (MBIST), a canary SRAM built-in self-test (CBIST), and a scan chain block. The scan chain block provides boundary scans for all of the four blocks, as depicted in Figure 5.2. It is not essential to use a 512b canary that tracks the dynamic write V_{MIN} (WV_{MIN}) of a smaller 8Kb SRAM. However, we incorporate a 512b canary SRAM to extract more data from the testchip. Note that we use the same 6T core SRAM bitcell for the canary SRAM, which has an identical read-write behavior of the core bitcells. Figure 5.3 shows the architecture of the core SRAM, which is almost identical to that of the canary SRAM. Both the core and canary use an identical leaf cell for the wordline drivers. Also, the read and write peripheral circuitry in the column periphery (I/O) is identical in both core and canary SRAM, except that the canary I/O has a bitline type reverse assist (BLVRA), as shown in Figure 5.3. The BLVRA pin uses an external positive supply voltage (BLVRA in Figure 5.3), which connects to the canary bitlines using the analog de-multiplexer (Demux) instead of using a pulling down circuit for BL or BLB to V_{SS} in the canary write driver. Both the core and the canary SRAM employ built-in power switches, which separate the supply voltage of the wordline from the core-array supply. Therefore in a write operation, reducing the WL supply voltage below the core-array supply voltage in either array results in a WL type reverse assist (WLVRA). We document the layout area of the core SRAM is $210,942.8\mu m^2$, and the canary SRAM is $151,236.9\mu m^2$ in the memory block.

5.0.5 Testing Circuitry

The SRAM and canary BISTs use similar architectures of a typical MBIST [82], as shown in Figure 5.4, which we design as semi-custom blocks for the characterization of write failures in the core and canary SRAMs. The BIST architecture comprises of a BIST computation pipeline (BCP) and a BIST finite state machine (BFSM). The BCP block (Figure 5.4)

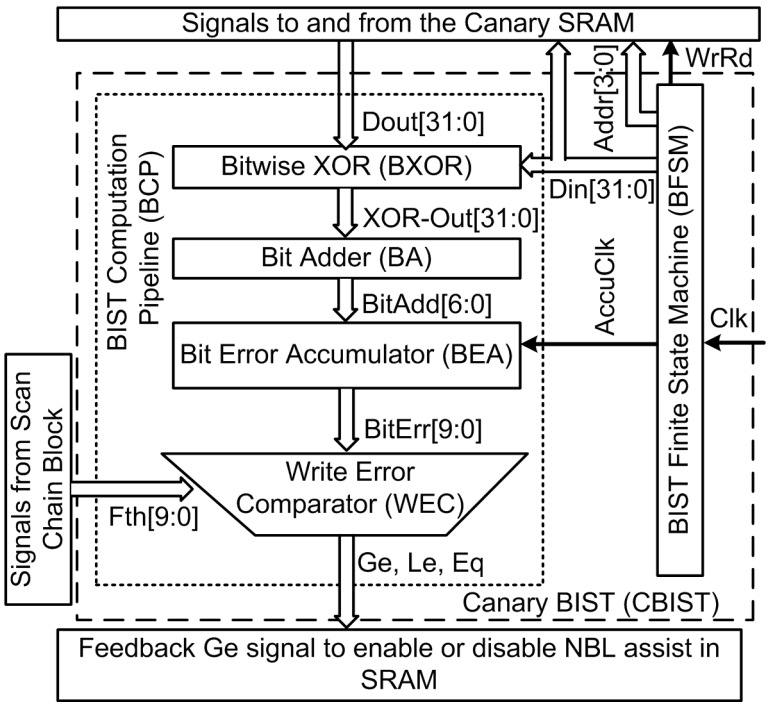


Figure 5.4: Block diagram of the canary BIST (© 2015 IEEE).

quantifies, accumulates, and compares the number of write failures in the memory array. On the other hand, the BFSM block is responsible for the generation the address, data, and read-write signals, as shown in Figure 5.4. The write error comparator (WEC) block compares the output of the BEA (Figure 5.4) and the failure threshold (F_{th}) [81] register and turns on appropriate status bits (Ge, Le, Eq) depending on the accumulated error. The status bits indicates the number of write failures to be greater, lesser, or equal to the F_{th} register value. Note that the CBIST uses the Ge signal as a feedback signal for the memory, which can control the negative bitline assist (NBLA) to turn on or off for improving the SRAM WV_{MIN}, accordingly. To track the SRAM WV_{MIN} using canaries requires identifying the equivalent canary failure point corresponding to a reverse assist setting for the SRAM WV_{MIN}, which defines as the initial reverse assist failure setting [81]. Using the initial reverse assist failure setting [81] for the canary SRAM corresponding to a specific SRAM WV_{MIN}, one can track the SRAM WV_{MIN} across frequency, temperature, etc. variations. Also, the user can take necessary actions, such as turning on or off assists for the SRAM, raising the

V_{DD}, etc. The characterization chip has scan chains for the SRAM, MBIST, canary SRAM, and CBIST, which configure the block for scan-in or scan-out operations. A multiplexer selects these scan chains to shift one at a time that uses a single scan-in and scan-out pin.

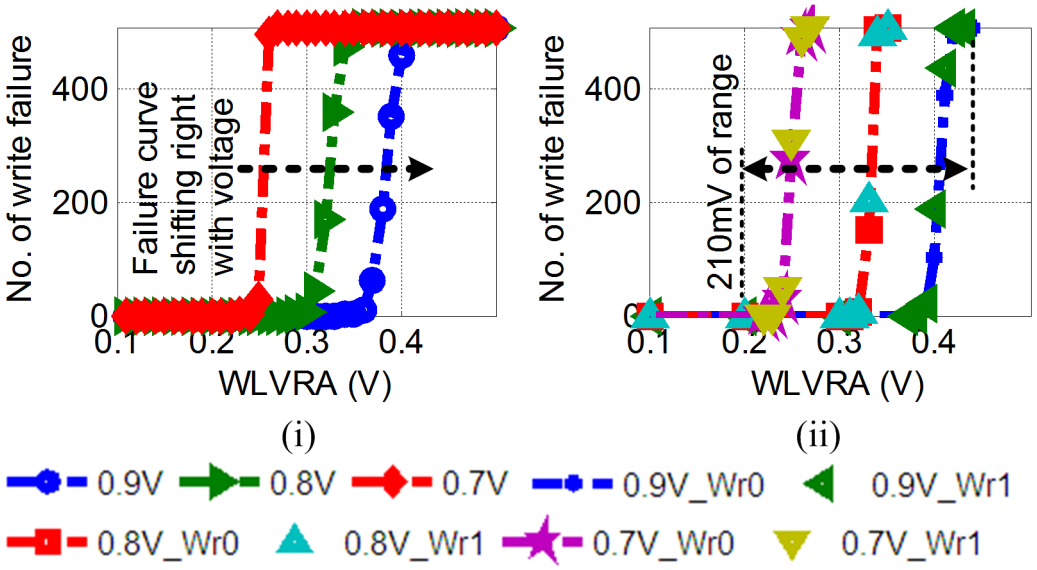


Figure 5.5: (i)Simulated (TT_24C_100MHz) canary write failures vs. WLVRA across 0.9V, 0.8V, and 0.7V supply voltages. (ii)Measured (24C_100MHz) canary write failures vs. WLVRA across 0.9V, 0.8V, and 0.7V supply voltages (© 2015 IEEE).

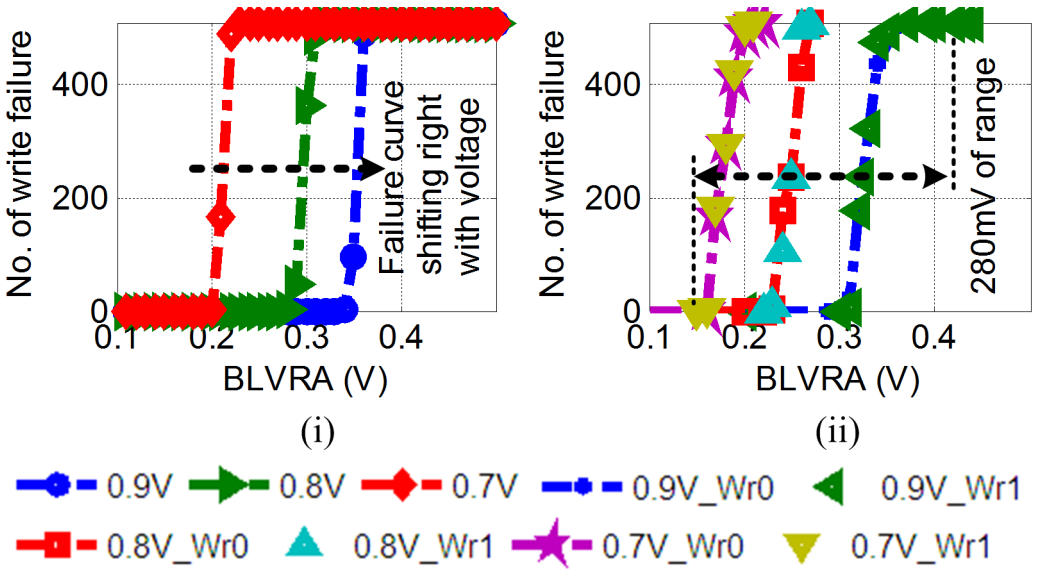


Figure 5.6: (i)Simulated (TT_24C_100MHz) canary write failures vs. BLVRA across 0.9V, 0.8V, and 0.7V supply voltages. (ii)Measured (24C_100MHz) canary write failures vs. BLVRA across 0.9V, 0.8V, and 0.7V supply voltages (© 2015 IEEE).

5.0.6 Test Setup And Chip Results

We use a Link Instruments IO3232B pattern generator and logic analyzer for testing. We perform testing related to the voltages and frequencies, which uses one chip at 24C. We use Tenny Jr. Oven for the requirements of temperature testing, with another chip for saving the first chip from accidental damage during the temperature testing. Besides, to simulate canary failure rates, we employ an extracted 6T bitcell netlist that uses modeled resistances and capacitances in the WL and BLs to run 10K Monte Carlo simulations per data point. The "range" defines as the bounds of a reverse assist in tuning canaries to fail within the acceptable bounds of DVFS or within the ranges of unintended voltage, frequency, and temperature variations.

5.0.7 Voltage Tracking

Figure 5.5 (i) and Figure 5.5(ii) depic the simulated (TT_24C corner) and measured plots of the canary write failure (CWF) vs. WL type reverse assist (WLVRA) at 100MHz for 0.9V, 0.8V, and 0.7V supply voltages. Across V_{DD} values, the write-failure vs. WLVRA plots are distinct. Therefore, scaling supply voltage will result in more canary failures for the same WLVRA value and the vice versa. We show in Figure 5.6 (i) and Figure 5.6 (ii) a similar plot for CWF vs. BL type reverse assist (BLVRA) at 100MHz, which has a similar trend. Thus, WLVRA and BLVRA can track voltage variations, as they induce distinct changes in the canary failures. The corresponding quantified range is 210mV for WLVRA and 280mV for BLVRA for 200mV of supply voltage variation, as shown in Figure 5.5 and Figure 5.6.

5.0.8 Frequency Tracking

Figure 5.7 (i) and Figure 5.7 (ii) depict simulated (TT_0.9V_24C corner) and measured plots of the CWF vs. WLVRA for 100MHz, 50MHz, and 25MHz frequencies. Note that there is no distinct shift in the canary failure threshold values or the curves themselves, which

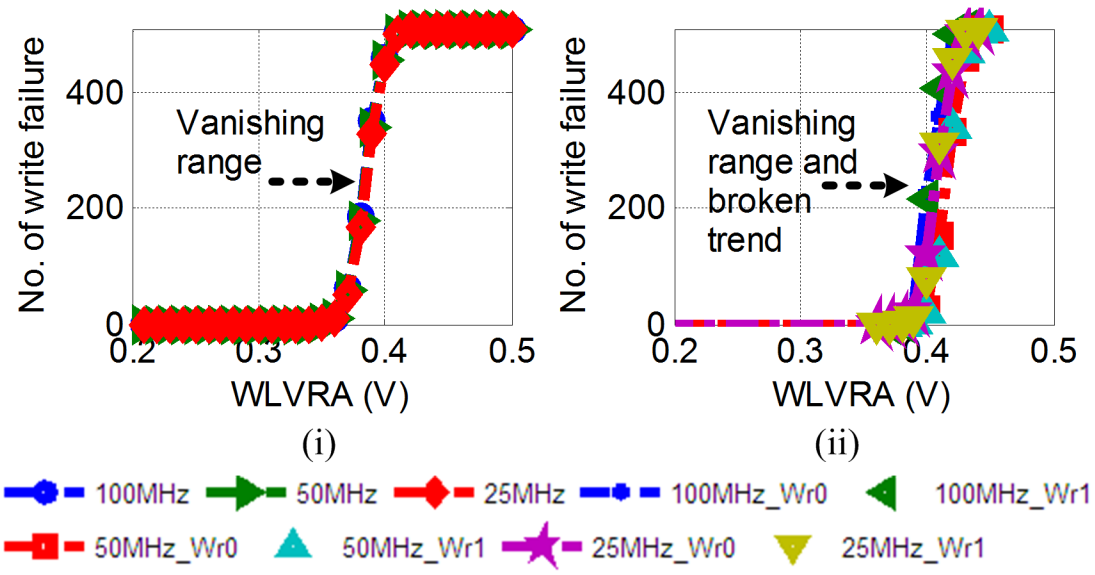


Figure 5.7: (i)Simulated (TT_0.9V_24C) canary write failures vs. WLVRA across 100MHz, 50MHz, and 25MHz clock frequencies. (ii)Measured (0.9V_24C) canary write failures vs. WLVRA across 100MHz, 50MHz, and 25MHz clock frequencies (© 2015 IEEE).

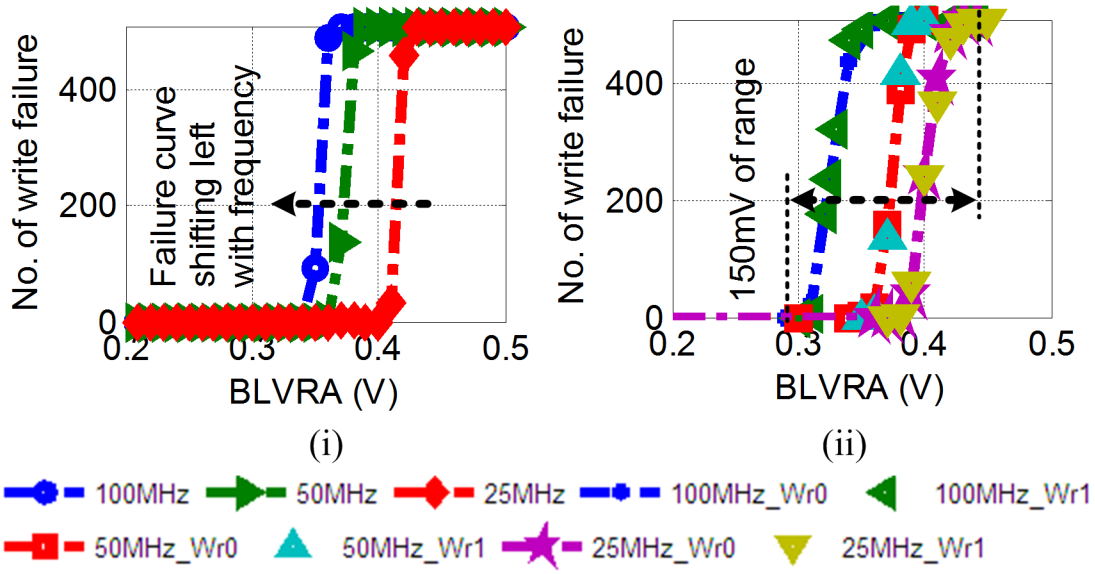


Figure 5.8: (i)Simulated (TT_0.9V_24C) canary write failures vs. BLVRA across 100MHz, 50MHz, and 25MHz clock frequencies. (ii)Measured (0.9V_24C) canary write failures vs. BLVRA across 100MHz, 50MHz, and 25MHz clock frequencies (© 2015 IEEE).

affects the ability to track using canaries across frequencies using WLVRA. This anomaly of using WLVRA for canaries may be due to the reason that the write margin becomes dominant over wordline pulse-width in write failure characteristics. However, using BLVRA across frequencies, we have distinct shifts in the canary failure thresholds and the curves themselves, as shown in Figure 5.8 (i) and Figure 5.8 (ii). Thus, we can use BLVRA for frequency tracking. We report the measured range for BLVRA is 150mV for 75MHz of frequency variation, as shown in Figure 5.8.

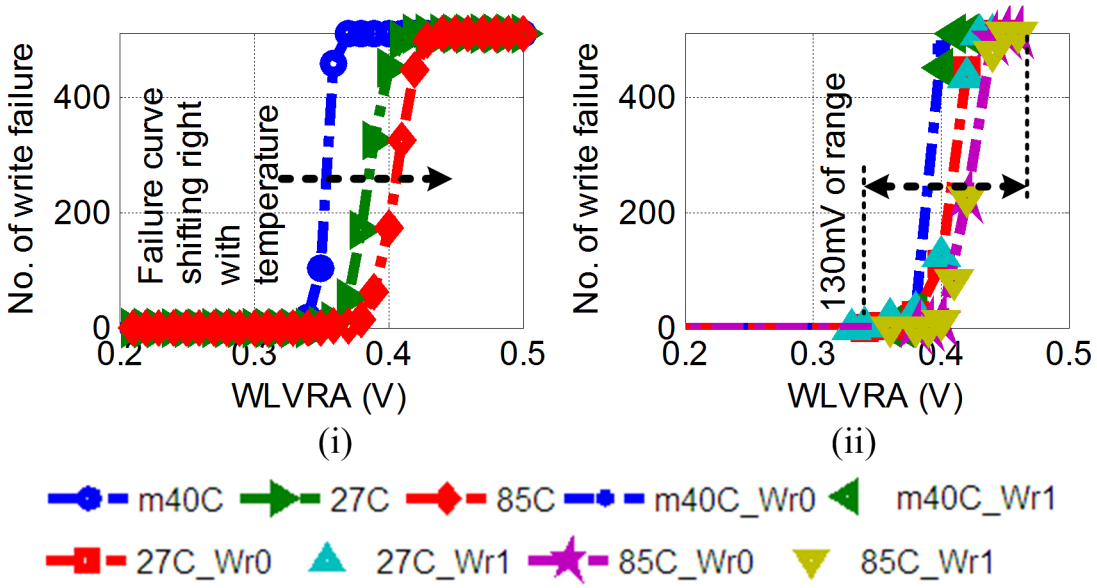


Figure 5.9: (i)Simulated (TT_0.9V_100MHz) canary write failures vs. WLVRA across -40C (m40C), 27C, and 85C temperatures. (ii)Measured (0.9V_100MHz) canary write failures vs. WLVRA across -40C (m40C), 27C, and 85C temperatures (© 2015 IEEE).

5.0.9 Temperature Tracking

We show in Figure 5.9 (i) and Figure 5.9 (ii) the simulated (TT_0.9V corner) and measured plots of the CWF vs. varying WLVRA for -40C (m40C), 27C, and 85C temperatures at 100MHz. Note that the canary failure threshold points shift, and the curves are distinct across temperatures. Figure 5.10 (i) and Figure 5.10 (ii) depict the distinct curves of the CWF vs. varying BLVRA for various temperatures. Thus, one can track temperature changes using WLVRA and BLVRA. We report the measured range as 130mV for WLVRA and

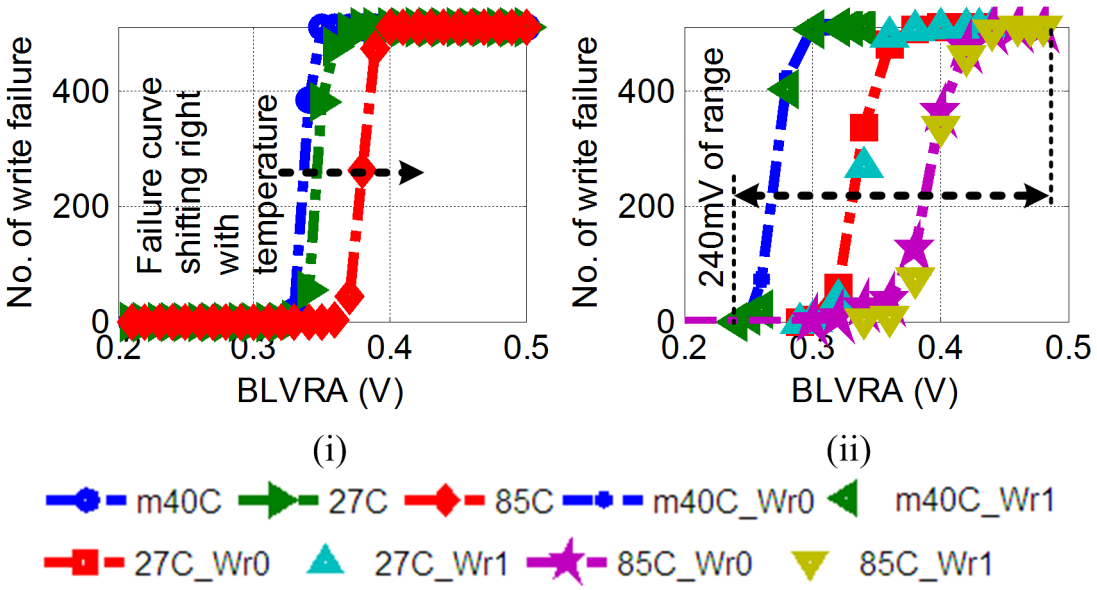


Figure 5.10: (i)Simulated (TT_0.9V_100MHz) canary write failures vs. BLVRA across -40C (m40C), 27C, and 85C temperatures. (ii)Measured (0.9V_100MHz) canary write failures vs. BLVRA across -40C (m40C), 27C, and 85C temperatures (© 2015 IEEE).

240mV for BLVRA for 125 degree centigrade of temperature variation (Figure 5.9 (ii) and Figure 5.10 (ii)). Our measured canary failure trends match the simulation results across the design parameters. Note that the failure trends are steep, due to the reason that at 130nm the process variation is not higher. Tuning the canaries in the BLVRA range requires the generation of the BLVRA voltages externally, using a voltage reference [83]. However, this approach results in a higher area penalty. Besides, we can generate the BLVRA voltages, using an NMOS-NMOS voltage divider inside the write driver of the canary SRAM [81], which requires additional circuitry.

5.0.10 Tuning Canaries before the SRAM Failure Point

We show in the Figure 5.11 the annotated plot at 100MHz for the measured data. The plot confirms that we can tune the canary using BLVRA to fail before the SRAM bits start to fail and turn on assist (NBLA), if necessary, to make the SRAM robust in write. Noticeably, using WLVRA the SRAM failure starts below WLVRA=0.35V. We compare the SRAM write failure probability at 100MHz (due to testing limitations) to the canary SRAM failure

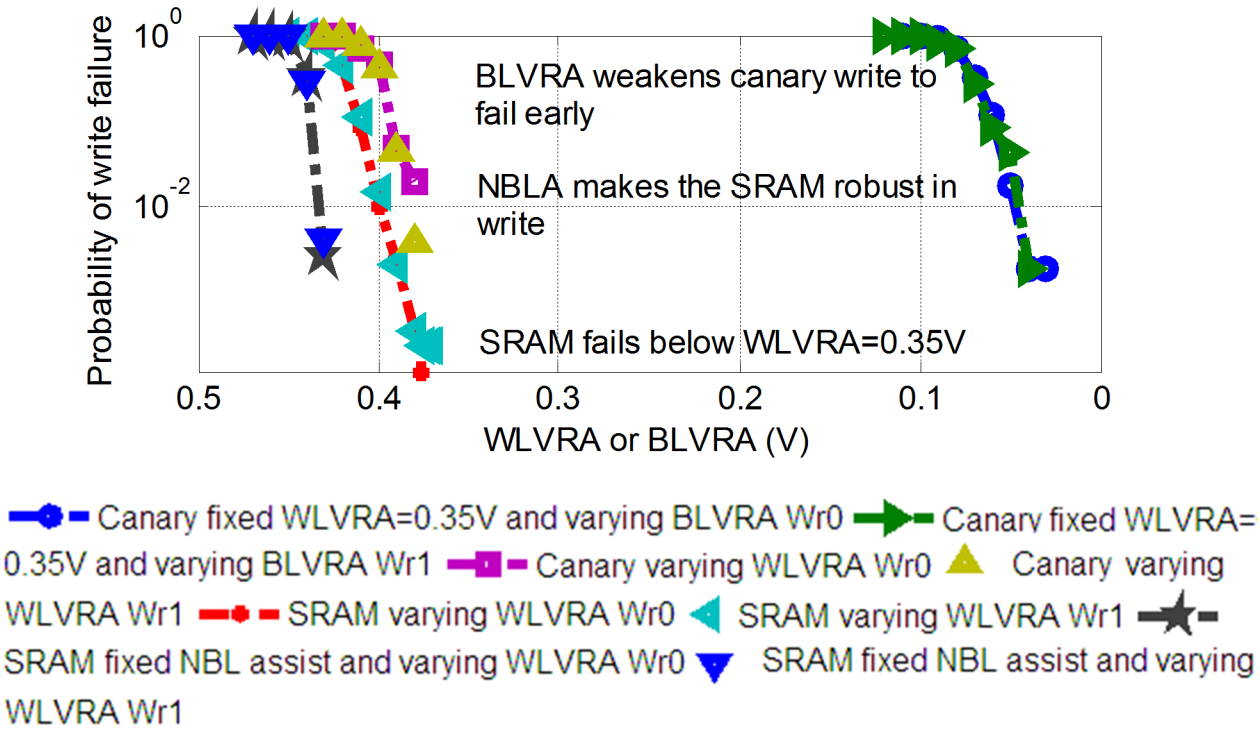


Figure 5.11: Measured canary SRAM failure point tuning before the SRAM bits fail at 0.9V_24C_100MHz (© 2015 IEEE).

probability using a fixed WLVRA=0.35V and varying BLVRA. The varying BLVRA makes the canaries more sensitive to write failures and the canary failure curve shifts to the right by failing earlier than the SRAM bits start to fail. We measure the power at 0.9V and 100MHz frequency to be 876.87μW for the canary SRAM with 400mV of BLVRA, and 848.43μW without BLVRA, and 1351.35μW for the SRAM.

5.0.11 Conclusions

This chapter shows the first silicon results, which confirms that the canary write failures distinctly change with voltage, frequency, and temperature variations for BLVRA. Hence, canary SRAM using BLVRA can track voltage, frequency, and temperature variations, and thus tracking of the SRAM WV_{MIN} is realizable. However, the failure trends for WLVRA are distinct to voltage and temperature changes only. Thus, WLVRA is useful for tracking WV_{MIN} across voltage and temperature variations, only. Finally, this chapter shows that one

can tune the canary SRAM failure point using WL and BL type reverse assists before the SRAM bits start to fail, which is a crucial condition for tracking the dynamic write V_{MIN} of SRAMs using canaries.

5.0.12 Acknowledgments

This project was supported in part by NVIDIA through the DARPA PERFECT program and by the NSF NERC ASSIST Center (EEC-1160483).

Chapter 6

Classification of Reverse Assists, Their Properties, and Tradeoffs

Canary SRAMs use reverse assists (RA) [81] to tune the canary failure rate to fail earlier than the SRAM bitcells start to fail. In other words, RAs make the canary V_{MIN} (CV_{MIN}) very closed to the SRAM V_{MIN} (SV_{MIN}), so that canaries fail earlier while scaling the voltage down compared to the SRAM bitcells. There are many ways to make RAs [87] [91] for write and read operations, and we investigate the design knobs for write-ability, readability, and read-stability RAs in the next section.

6.0.1 Write and Read RAs

As we know, the 6T SRAM write operation mainly involves the wordline and bitlines (Figure 1.8) as the control signal. During the usual write operation for writing a '0', the wordline switches from logic '0' to '1' and the precharged bitline is lowered to the ground potential, while bitline-bar is kept floating at V_{DD}. After some time the internal nodes flip, depending on the previous data, and a write operation completes. Hence, to make a canary SRAM that is weaker in writing data, we can do the following: a) degrade the wordline slope, or b) limit the height of the wordline pulse. Also, we can c) shrink the wordline pulse

Table 6.1: Summary of write reverse assist (RA) techniques using wordline and bitline controls.

Signal control knobs	**Write reverse assist mechanism**
Wordline	Pulse height degradation
	Pulse slope degradation
	Pulse width degradation
Bitline	Pulse height degradation
	Pulse slope degradation
	Pulse width degradation

width to make the writing of the data harder. Similarly, using the bitline we can do the following to make the writing of the data harder for canaries: a) degrade the slope of the bitline, or b) limit how low the bitline can reach down near to ground voltage level. Also, we can c) shrink the bitline pulse to a shorter time duration to make the writing of the data harder. Although there exist peripheral assists that degrade the write operation, including readability peripheral assists such as boosting the V_{DD}, V_{SS} lowering, etc., this dissertation only investigates wordline and bitline type RAs for the write operation. A complete table of possible write RAs is shown in Figure 6.1 for reference.

On the other hand, the 6T SRAM read operation involves wordline as the control signal and a sense amplifier (SA) measures the bitline differential voltage to read a successful '0' or '1' (Figure 1.9). During the read operation, the wordline switches from logic '0' to '1,' and bitline and bitline-bar signals float being precharged to V_{DD}. After some time one of the bitline and bitline-bar discharges more than the other, and a differential develops. To make a readability-RA, one can delay this differential development time or limit the amount of differential within a fixed time. Therefore, to make a canary SRAM that is weaker in reading data, we can do the following to the wordline: a) degrade the wordline slope, or b) limit the height of the wordline pulse. Also, we can c) shrink the wordline pulse width to make the writing of the data harder. Similarly, we can create a readability-RA by delaying the SA enable signal (ENSA) or increasing the slope of the ENSA that triggers the SA for reading out the differential voltages in bitlines. Moreover, peripheral write assists such as

V_{DD} collapse, V_{SS} raising, etc. would make a readability-RA, but we limit our discussions to wordline type RA for readability.

Moreover, there could be a third category of RAs for making a canary sensitive to read-stability. A read-stability RA degrades the stability of the 6T SRAM bitcells compared to the SRAM core bitcells and makes the canaries fail earlier in V_{DD}, which involves a bitline interleaving scenario using column multiplexers. The half-selected bitcells in the same row in a column multiplexing write scenario exhibit read stressing, and those are prone to read-stability issues. Usually, the read-stability of half-selected bitcells gets worse with a boosted wordline voltage or making the wordline pulse width wider. Thus, we give the design knobs of a read-stability canary RA as follows: a) boost the height of the wordline pulse, or b) widen the wordline pulse width to worsen the read-stability of canaries. Similarly, we can precharge the bitlines to a different voltage, lower or higher than V_{DD}, to make the read-stability worse for canaries. However, we limit our discussion to the RAs using wordline-based read-stability canaries, only. The next sections describe the pulse shaping methods to make RA-based canaries, focusing on wordline and bitlines as the control signals.

6.0.2 Pulse-height-degradation (PHD) type Read-Write RAs

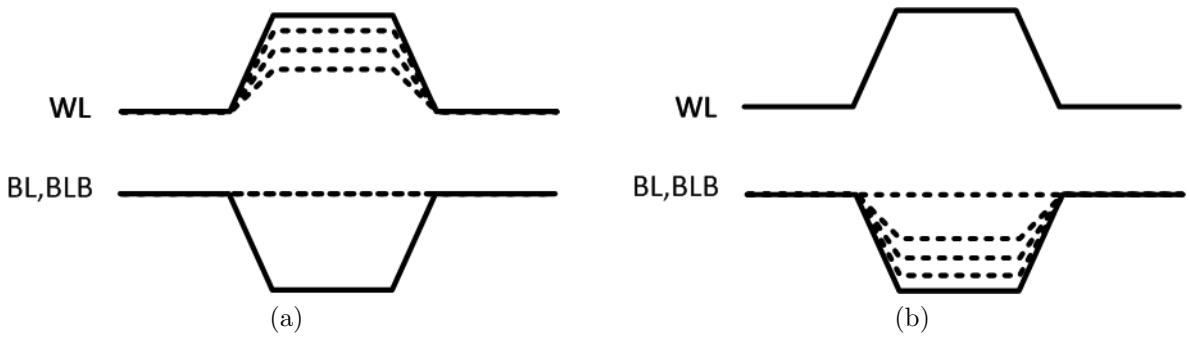


Figure 6.1: (a) Wordline pulse-height-degradation (WLPHD) and (b) bitline pulse-height-degradation (BLPHD) reverse assist waveforms for canary SRAMs.

Degrading the pulse-height of the 6T bitcell control signals such as wordline and bitlines results in write and read reverse assists (RA) to make canary cells. Here we assume that

limiting the pulse-height of the control signal does not affect the slope and the width of the pulse much. Therefore, the pulse height degradation (PHD) type RAs limit the upper bound of wordline pulse-height or lower bound of bitline pulse-height for write operations. Note that only wordline type PHD (WLPHD) creates the RA-based canaries for a read operation. Figure 6.1a and Figure 6.1b shows the typical waveforms of the wordline and bitline PHD RAs for canary write and read operations.

6.0.3 Pulse-width-degradation (PWD) type Read-Write RAs

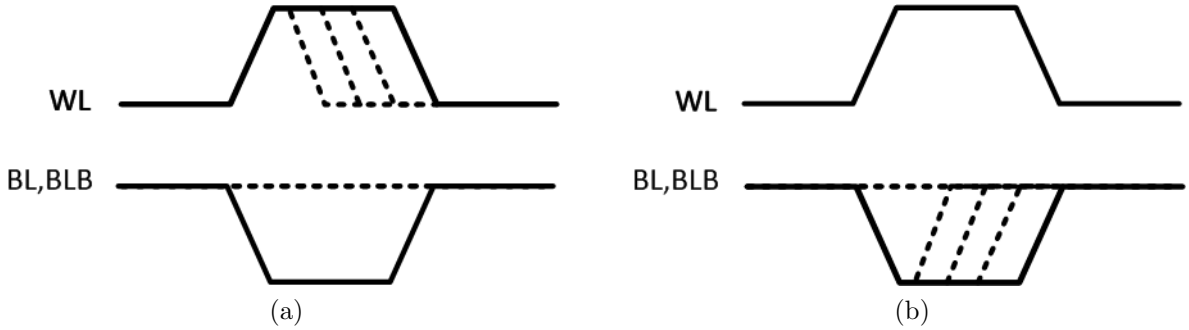


Figure 6.2: (a) Wordline pulse-width-degradation (WLPWD) and (b) bitline pulse-width-degradation (BLPWD) reverse assist waveforms for canary SRAMs.

Another way to make RA is by shortening the pulse width of the control signals. Thus, the write and read operations will undergo stress due to a shorter time of operation, and they will be degraded. Here we assume that shortening the pulse does not necessarily affect other features of the control signal, such as the pulse-height and pulse-slope, which should remain the same in an ideal case. In an extreme scenario, where the shortened pulse width is comparable to the rise or fall time of the control signal, it might change the signal pulse-height or pulse-width. Thus, the pulse-width-degradation (PWD) type RAs require shortening of the wordline or bitline pulse widths for 6T SRAM for the canary write operations. On the other hand, only wordline type PWD RA is applicable for a read operation. Figure 6.2a and Figure 6.2b show the typical waveforms of the wordline and bitline PWD RAs for canary write and read operations.

6.0.4 Pulse-slope-degradation (PSD) type Read-Write RAs

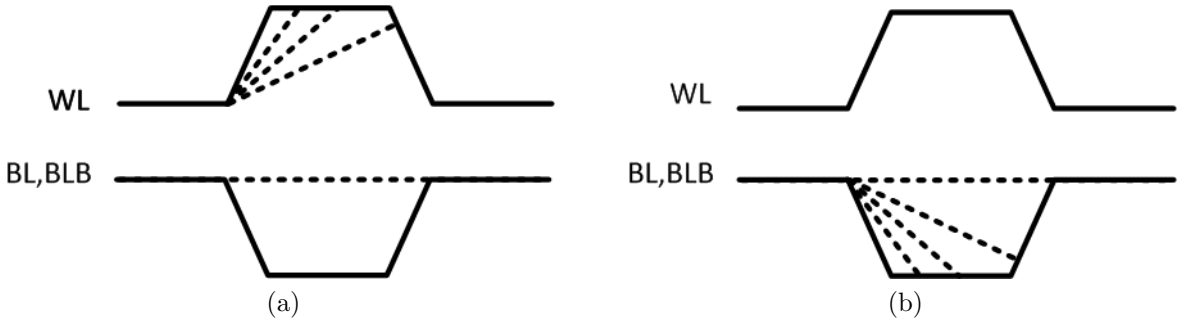
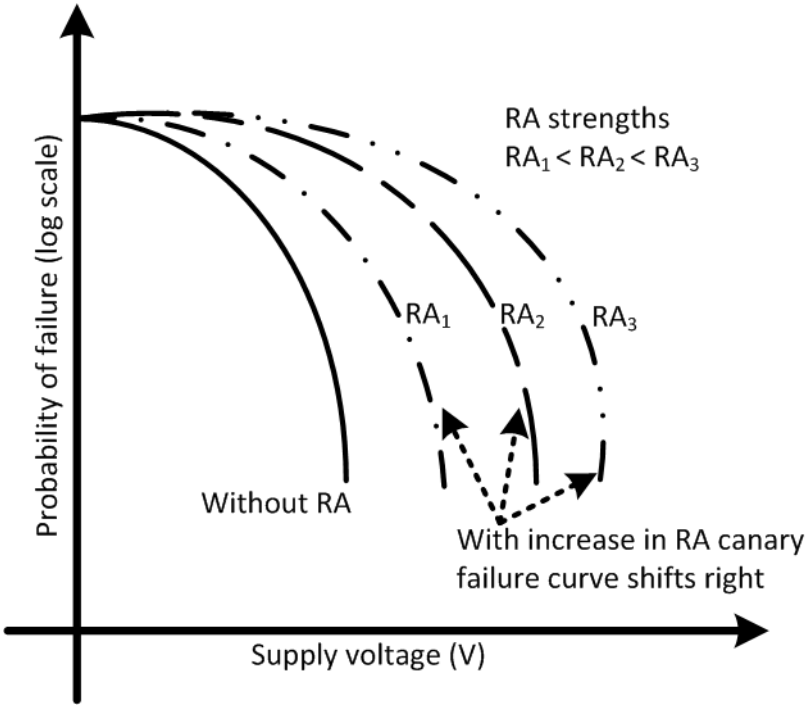


Figure 6.3: (a) Wordline pulse-slope-degradation (WLPSD) and (b) bitline pulse-slope-degradation (BLPSD) reverse assist waveform for Canary SRAMs.

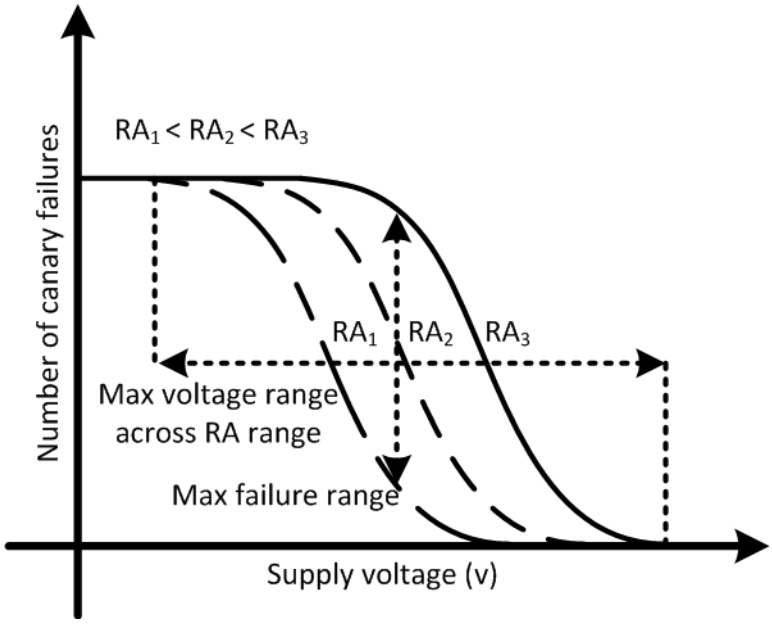
One of the fundamental ways to make a reverse assist (RA) for 6T SRAM is to degrade the slope of the pulse of write and read control signals that reduce the probability of a successful write or read operation. The effective pulse-width or pulse-height of the control signal may change due to slope degradation of the control signal. Thus, the pulse-slope-degradation (PSD) type RAs have combined benefits of PHD and PWD RAs. As the wordline and bitlines are key control signals for read and write operations in 6T SRAMs, degrading the slopes of wordline and bitline is tantamount to an RA for canaries. For a write operation, deteriorating any of the wordline or bitline slope would cause canaries to have weaker write-ability. On the other hand, only degrading the wordline slope for read operation incurs degradation of bitline differential for readability. Figure 6.3a and Figure 6.3b show the typical waveform of wordline and bitline PSD type RAs for canary write and read operations.

6.0.5 Metrics of Comparison of RAs

A typical RA plot for canary failure rate P_{fail} (log scale) vs. supply voltage looks like Figure 6.4a where, with increasing strength of RA, the canaries fail earlier in V_{DD} or canaries have a higher V_{MIN}. In other words, the canary failure rate curves shift right with increasing RA strengths. The same data can be visualized as the plot for the number of canary failures vs. supply voltage (V_{DD}) across RA settings (RAS), as shown in Figure 6.4b for a specific process



(a)



(b)

Figure 6.4: (a) Canary probability of failure vs. supply voltage across reverse assist strengths. (b) The number of canary failures vs. supply voltage across reverse assist strengths.

(P), temperature (T), and clock frequency (F). Here, we can define the max voltage range ($V_{max_{range}}$) across the RA range corresponding to the number of failures (N_f). Similarly, we can define the max failure range ($Fail_{MAX_{range}}$) corresponding to the RA range. Noticeably, for a fixed RA range having a larger $V_{MAX_{range}}$ indicates that the worst-case SRAM bitcell maps into canaries covering a more extensive V_{DD} range across process and temperature variations corresponding to the type of RA used. Similarly, a wider $Fail_{MAX_{range}}$ would result in distinct failure behavior of canaries across RAS values and, thus, a user can take a specific decision easily such as turning on assists or stopping voltage scaling, etc. On the other hand, a smaller or vanishing $V_{max_{range}}$ and $Fail_{MAX_{range}}$ would indicate an inferior property of the RA for mapping the worst-case SRAM bitcell into canaries for SRAM V_{MIN} tracking. Thus, equation 6.1 gives the number of canary failures. Hence, we define a maximum failure sensitivity metric across RA strength in equation 6.2, which relates to the ratio of maximum canary failure range to the RA range.

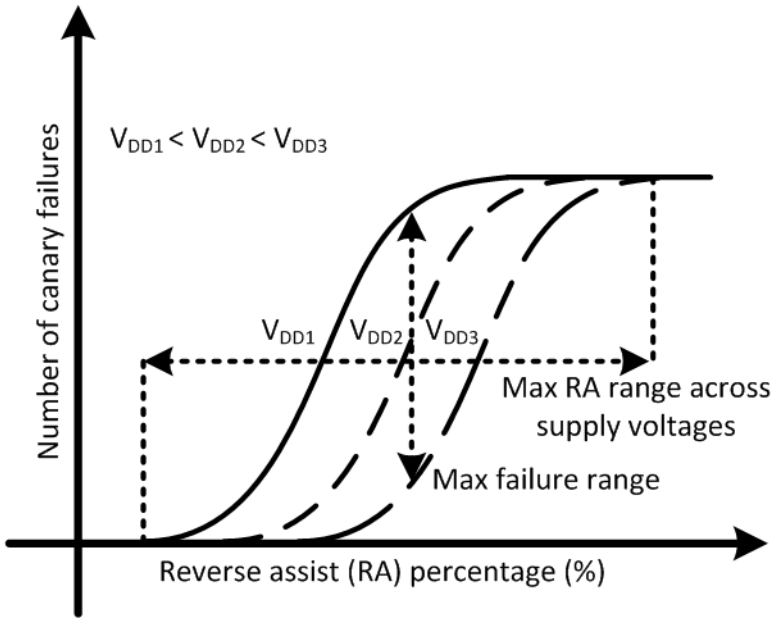
$$N_f = N_c * P_{fail} \tag{6.1}$$

$$\Lambda_{RA} = \left.\frac{\partial N_f}{\partial RA}\right|_{P,V_{DD},T,F} = N_c * \left.\frac{\partial P_{fail}}{\partial RA}\right|_{P,V_{DD},T,F} \simeq \frac{Fail_{MAX_{range}}}{RA_{range}} \tag{6.2}$$

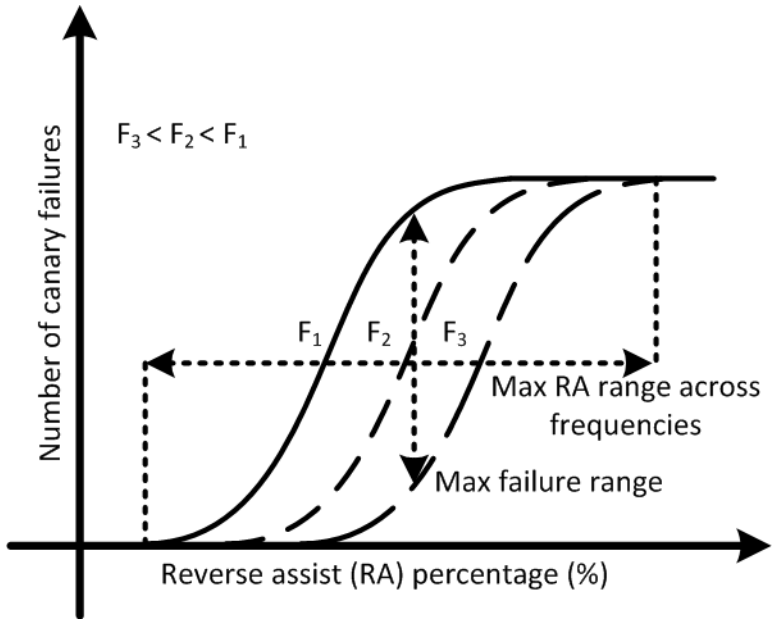
Similarly, we can plot the number of canary failures vs. RA percentage across supply voltages for a fixed process (P), frequency (F) and temperature (T), as shown in Figure 6.5a. Here, we define the maximum failure range $Fail_{MAX_{range}}$ corresponding to a fixed RA percentage value, where the supply voltage range is V_{range}. Hence, we express the corresponding maximum canary failure sensitivity related to V_{DD} change in equation 6.3.

$$\Lambda_{V_{DD}} = \left.\frac{\partial N_f}{\partial V_{DD}}\right|_{P,T,RA,F} = N_c * \left.\frac{\partial P_{fail}}{\partial V_{DD}}\right|_{P,T,RA,F} \simeq \frac{Fail_{MAX_{range}}}{V_{range}} \tag{6.3}$$

On the other hand, one can represent the number of canary failures vs. RA percentage across clock frequencies for a fixed process (P), supply voltage (V_{DD}), temperature (T), as



(a)



(b)

Figure 6.5: (a) The number of canary failures vs. reverse assist percentage across supply voltages. (b) The number of canary failures vs. reverse assist percentage across clock frequencies.

shown in Figure 6.5b. We define the maximum failure range $Fail_{MAX_{range}}$ corresponding to a fixed RA percentage value, where the frequency range is F_{range}. We express the corresponding maximum canary failure sensitivity related to F change in equation 6.4.

$$\Lambda_F = \frac{\partial N_f}{\partial F}\bigg|_{P,V_{DD},T,RA} = N_c * \frac{\partial P_{fail}}{\partial F}\bigg|_{P,V_{DD},T,RA} \simeq \frac{Fail_{MAX_{range}}}{F_{range}} \tag{6.4}$$

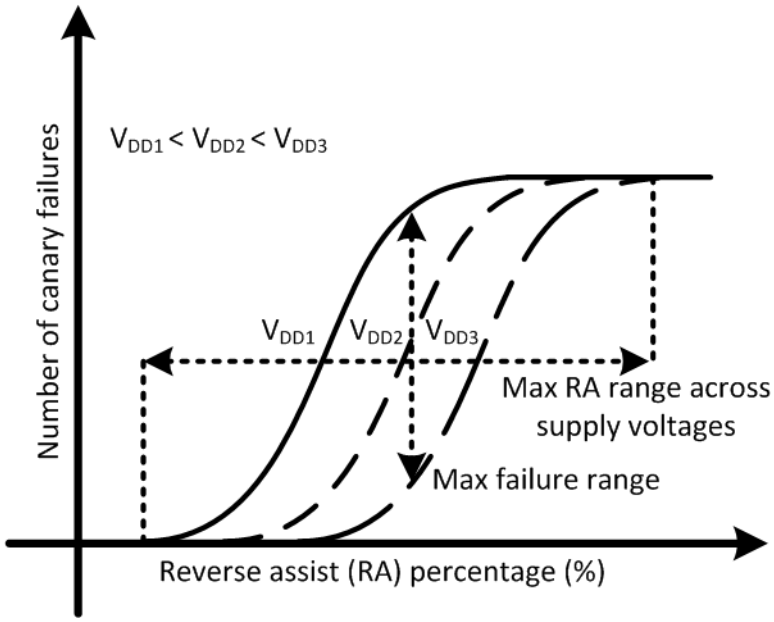


Figure 6.6: The number of canary failures vs. reverse assist percentage across temperatures.

Similarly, Figure 6.6 shows an illustration of the number of canary failures vs. RA percentage across temperatures for a fixed process (P), supply voltage (V_{DD}), and frequency (F). We define the maximum failure range $Fail_{MAX_{range}}$ corresponding to a fixed RA percentage value, where the frequency range is T_{range}. Equation 6.5 shows the corresponding maximum canary failure sensitivity related to T change.

$$\Lambda_T = \frac{\partial N_f}{\partial T}\bigg|_{P,V_{DD},RA,F} = N_c * \frac{\partial P_{fail}}{\partial T}\bigg|_{P,V_{DD},RA,F} \simeq \frac{Fail_{MAX_{range}}}{T_{range}} \tag{6.5}$$

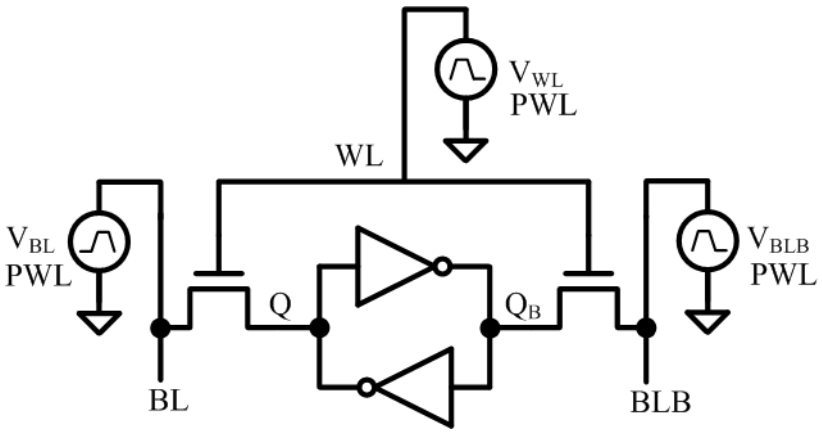


Figure 6.7: HSPICE simulation setup for canary probability of write and read failure extraction employing reverse assists.

6.0.6 Simulation Test Setup for Write and Read Wordline and Bitline type RA Comparison

To generate the required data for canary failure across design knobs such as supply voltage, temperature, reverse assist types, the strength of reverse assists, etc., we adopt the setup described in Figure 6.7. Here, a 6T SRAM circuit is under test, using wordline and bitline supply sources, modeled as Piece-wise Linear (PWL) sources. A waveform generator Perl script selects the suitable reverse assist PWL waveform, as directed by the user, for a write or read operation. For a canary write operation, we choose a wordline or bitline type RA such as wordline pulse-height-degradation (WLPHD) or bitline pulse-slope-degradation (BLPSD), etc. On the other hand, for a canary read operation only, we choose a wordline type RAs, such as wordline pulse slope degradation (WLPSD), wordline pulse width degradation (WLPWD), and WLPHD. For simulation-based comparison of RA metrics, we use a commercial 14nm technology and HSPICE simulator. Initially, we run HSPICE simulations across canary design knobs, as mentioned above and collect the probability of failure data of canary sensors. We use the data to compute the ranges of the canary design knobs and sensitivity metrics, which we discuss in the next sections.

6.0.7 Results and Discussion for Wordline and Bitline type RA for the Write-ability of Canary Sensor SRAM

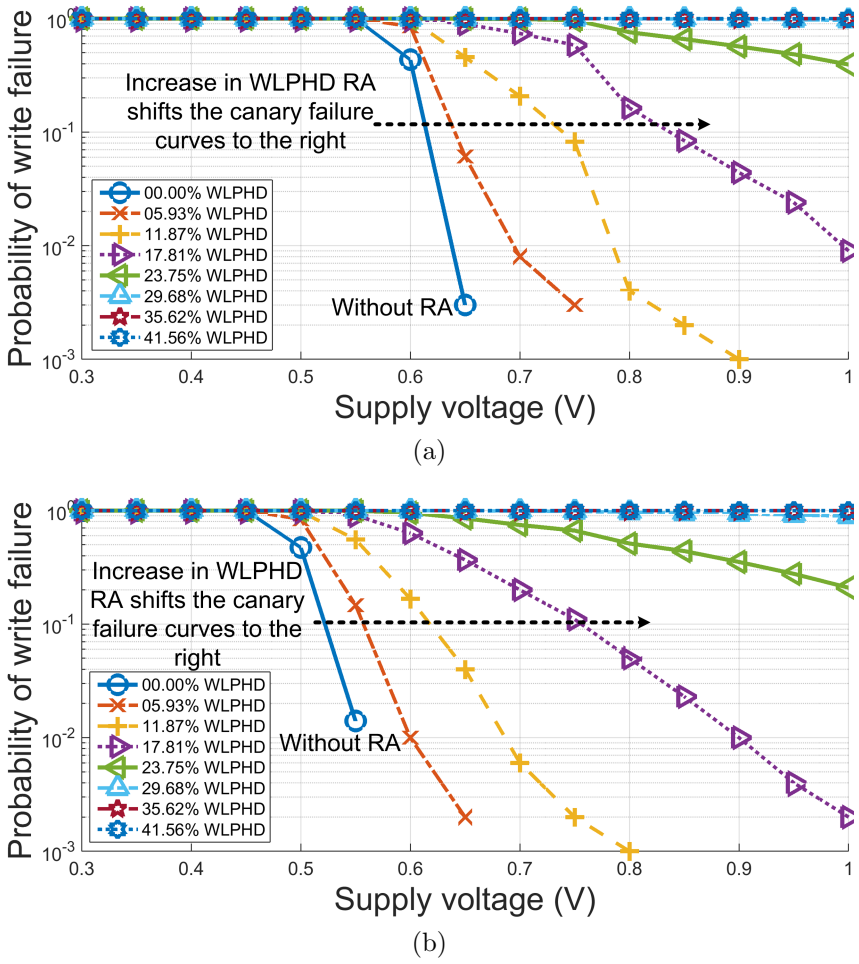
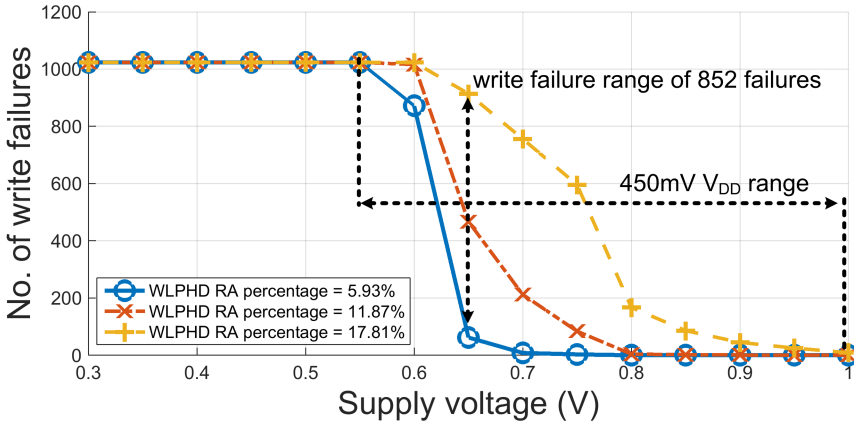


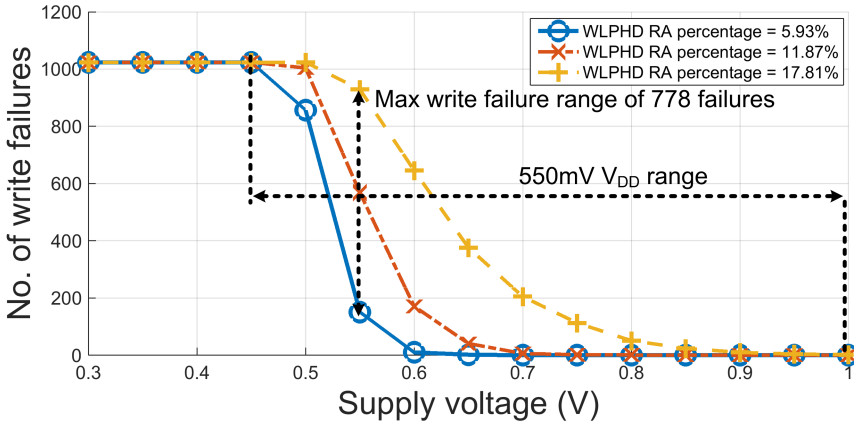
Figure 6.8: (a) Probability of write failure for canary sensor SRAM vs. supply voltage across WLPHD RA percentages at 5GHz 27C. (b) Probability of write failure for canary sensor SRAM vs. supply voltage across WLPHD RA percentages at 3GHz 27C.

As discussed earlier, pulse shaping RAs fall into pulse-height-degradation (PSD), pulse-slope-degradation (PSD), and pulse-width-degradation (PWD) categories. Thus, there can be three types of wordline-based RAs for canary write operation such as wordline PHD (WLPHD), wordline PSD (WLPSD), and wordline PWD (WLPWD) RAs. On the other hand, there are three possible bitline types of RAs such as bitline PHD (BLPHD), bitline PSD (BLPSD), and bitline PWD (BLPWD) RAs. Figure 6.8a and Figure 6.8b show write failure probability of canary sensors vs. supply voltage for 5GHz and 3GHz frequencies. We observe

that increasing the WLPHD RA percentage would shift the canary write failure probability curves to the right. In other words, increasing the WLPHD RA percentage increases the canary V_{MIN} for 5GHz and 3GHz frequencies. Figure 6.9a and Figure 6.9b show the plot for the number of write failures vs. supply voltage annotating the concept of maximum write failure range and supply voltage (V_{DD}) range. Noticeably, with the decrease in frequency from 5GHz to 3GHz, the V_{DD} range increases, but the range for the number of write failures decreases.



(a)



(b)

Figure 6.9: (a) The number of write failures for canary sensor SRAM vs. supply voltage across WLPHD RA percentages at 5GHz 27C. (b) Number of write failures for canary sensor SRAM vs. supply voltage across WLPHD RA percentages at 3GHz 27C.

Figure 6.10 shows the plot for the number of write failures for canaries vs. WLPHD RA percentages from 5GHz down to 3GHz clock frequencies. The corresponding RA range is

about 23.75%, and the maximum write failure range is 251 failures at 0.8V 27C. Similarly, Figure 6.11a and Figure 6.11b show the number of write failures for canaries vs. WLPHD RA percentages across temperatures. For the 5GHz clock frequency, the RA range is about 17.81%, and the range for the maximum number of write failures is 113 failures. On the other hand, at the 3GHz clock frequency, the RA range is 23.75%, and the range for the maximum number of write failures is 159 failures. Thus, with a decrease in clock frequency the RA range and range for the maximum number of write failures increase, which may help canary designers easily designing RA in lower frequencies.

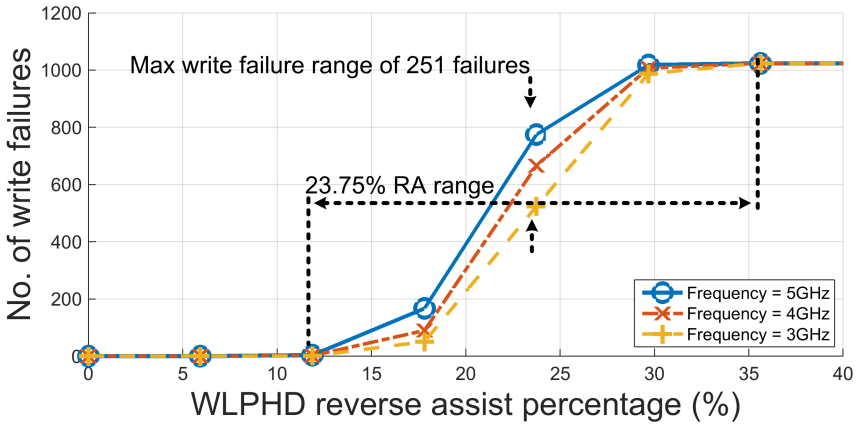
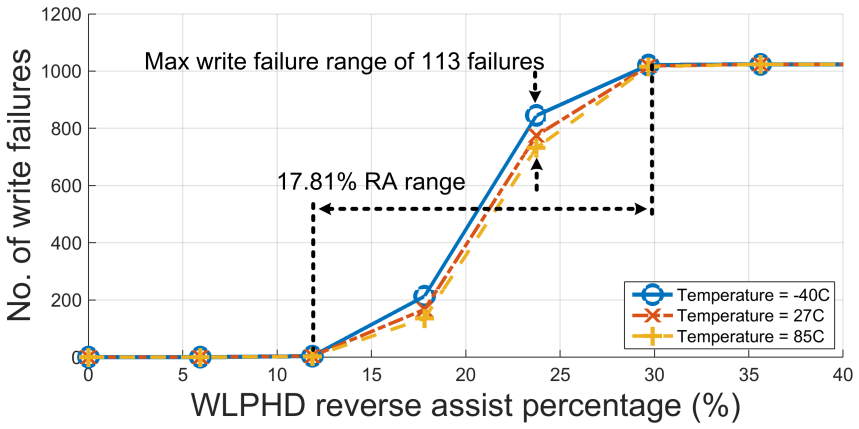
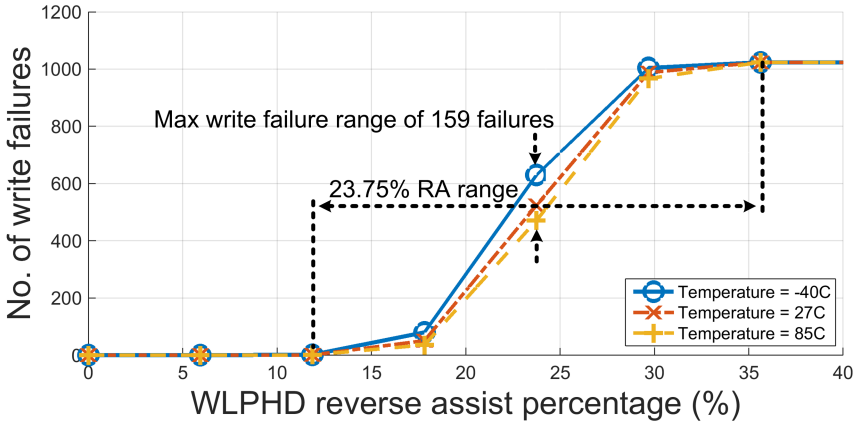


Figure 6.10: The number of write failures for canary sensor SRAM vs. reverse assist percentages across 5GHz, 4GHz, and 3GHz clock frequencies at 0.8V 27C.

Figure 6.12a shows the plot for the number of write failures for canaries vs. WLPHD RA percentages for 0.8V, 0.75V, and 0.7V supply voltages for 5GHz clock frequency. The corresponding WLPHD RA range is 23.75%, and the maximum write failure range is 589 failures. On the other hand, Figure 6.12b shows the plot for the number of write failures for canaries vs. WLPHD RA percentages for 0.8V, 0.75V, and 0.7V supply voltages for 3GHz clock frequency. The corresponding WLPHD RA range is 17.81%, and the maximum write failure range is 239 failures. Noticeably, with the decrease of clock frequency from 5GHz to 3GHz, both the WLPHD RA range and the maximum write failure range also shrink. Thus, WLPHD RA may be harder to design at lower frequencies, as the RA range decreases.

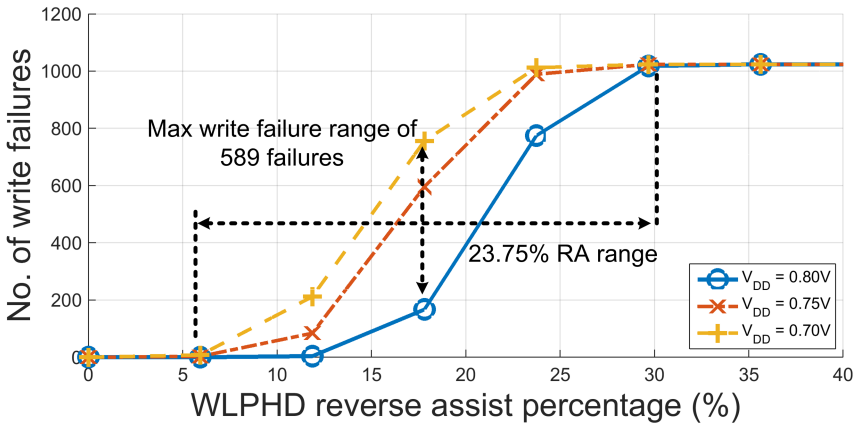


(a)

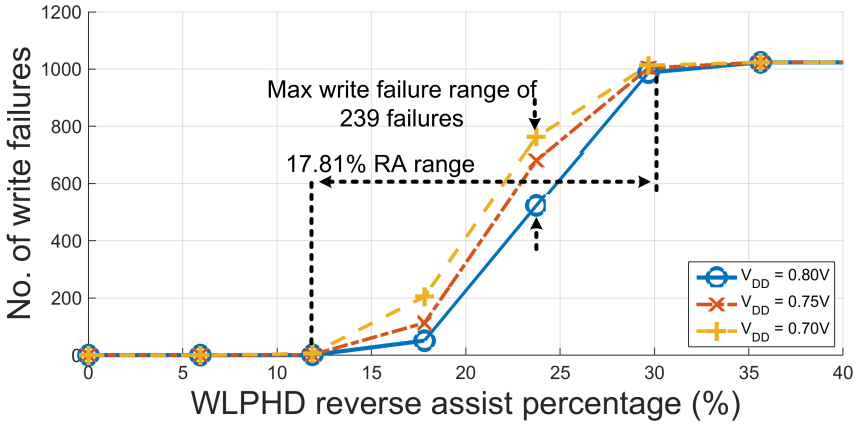


(b)

Figure 6.11: (a) The number of write failures for canary sensor SRAM vs. reverse assist percentages across -40C, 27C and 85C temperatures at 0.8V 5GHz. (b) The number of write failures for canary sensor SRAM vs. reverse assist percentages across -40C, 27C and 85C temperatures at 0.8V 3GHz.



(a)

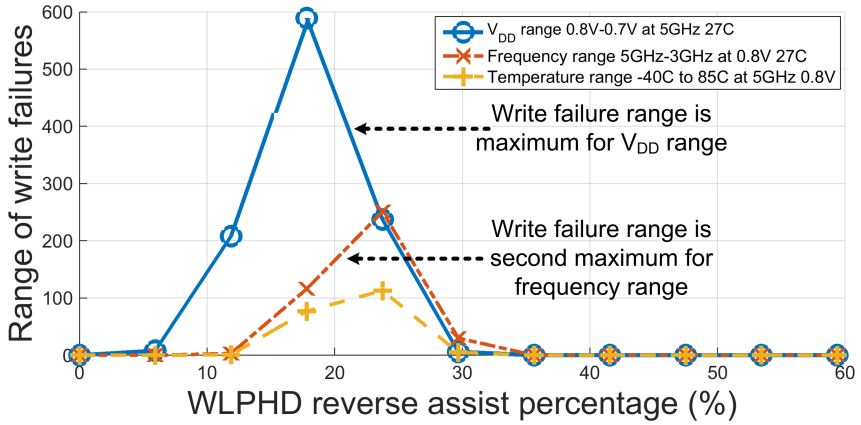


(b)

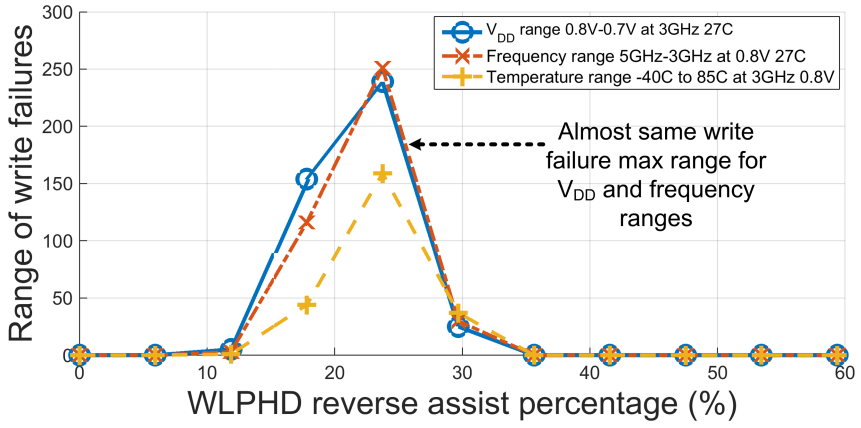
Figure 6.12: (a) The number of write failures for canary sensor SRAM vs. reverse assist percentages across 0.8V, 0.75V, and 0.7V at 5GHz 27C. (b) The number of write failures for canary sensor SRAM vs. reverse assist percentages across 0.8V, 0.75V, and 0.7V at 3GHz 27C.

To compute the sensitivity metrics of the WLPHD RA across other canary-RA design knobs such as the WLPHD RA range, V_{DD} range, frequency range, and temperature range, we calculate the corresponding write failure ranges. Hence, we compute the write failure ranges across the above canary WLPHD RA design knobs from the previously generated data and using Figure 6.9a to Figure 6.12b. Figure 6.13a shows the ranges of write failure for the V_{DD} range for 5GHz, frequency range, and temperature range for 5GHz across WLPHD RA percentages. The write failure range for the V_{DD} range at 5GHz has the maximum range for failures. The second maximum range for write failures is corresponding to the frequency range and the write failure range curve corresponding to the temperature range at 5GHz is the lowest. These trends indicate that the canary write failure has maximum sensitivity to the V_{DD} changes at the 5GHz frequency, frequency changes at nominal V_{DD} at 27C temperature, and temperature changes at 5GHz respectively. On the other hand, Figure 6.13b shows the ranges of write failure for the V_{DD} range at 3GHz, frequency range, and temperature range at 3GHz across WLPHD RA percentages. Noticeably, the write failure range curves for the V_{DD} range at 3GHz and frequency range are almost overlapping and have roughly the same maximum points. And the maximum write failure range for the temperature at 3GHz is better compared to the 5GHz case, which indicates that the temperature sensitivity of the canary write failures using WLPHD RA will increase at lower frequencies compared to higher ones. Thus, the worst-case frequency to consider designing a WLPHD RA is the highest one in a specification, as the RA will have the least temperature sensitivity.

Extending the RA experiments to BLPHD, WLPSD, BLPSD, WLPWD, and BLPWD RAs yield a comprehensive analysis of the RA properties across RA types for canary write-ability. Table 6.2 shows the RA ranges for the V_{DD} range of 0.8V-0.7V at temperature 27C across the canary RAs for write operation. Table 6.2 shows that the biggest RA ranges are for BLPSD RA and the smallest are for WLPHD RA across 5GHz and 3GHz clock frequencies. Note that a bigger RA range means a smoother transition of write failures and relatively easy to design canary tuning points in circuits compared to an RA with a smaller range.



(a)



(b)

Figure 6.13: (a) Range of write failures vs. reverse assist percentages across 0.8V-0.7V VDD range at 27C 5GHz, 5GHz-3GHz frequency range at 0.8V 27C, and -40C to 85C temperature range at 0.8V 5GHz. (b) Range of write failures vs. reverse assist percentages across 0.8V-0.7V VDD range at 27C 3GHz, 5GHz-3GHz frequency range at 0.8V 27C, and -40C to 85C temperature range at 0.8V 3GHz.

Table 6.2: RA range for V_{DD} range of 0.8V-0.7V at temperature 27C.

Freq (GHz)	**WLPHD (%)**	**BLPHD (%)**	**WLPSD (%)**	**BLPSD (%)**	**WLPWD (%)**	**BLPWD (%)**
5	23.75	35.63	250	375	49.5	43.31
3	17.81	29.69	125	625	24.75	18.56

Table 6.3: RA range for frequency range of 5GHz-3GHz at temperature 27C with 0.8V V_{DD}.

WLPHD (%)	**BLPHD (%)**	**WLPSD (%)**	**BLPSD (%)**	**WLPWD (%)**	**BLPWD (%)**
17.81	23.75	562.5	625	37.13	30.94

Table 6.3 shows RA ranges for the frequency range of 5GHz-3GHz at temperature 27C with 0.8V V_{DD} across the canary RAs in write operation. In this table, BLPSD has the highest RA range of 625% and WLPHD has the lowest RA range of 17.81%. On the other hand, the Table 6.4 shows the RA range for temperature range of -40C to 85C at 0.8V V_{DD}. At 5GHz the BLPSD has the highest RA range of 125%, and WLPSD has the second highest RA range of 62.5%. However, at 3GHz the WLPSD has the highest range of 712.5% and BLPHD has the second highest RA range of 250%. On the other hand, at 5GHz the lowest RA range occurs using WLPHD, but at 3GHz the WLPWD has the lowest range across temperature variations. Table 6.5 shows V_{DD} range for RA range of 17.81% to 5.94% at 27C temperature. In this case, the WLPHD RA has the highest supply voltage range of 0.4V at 5GHz and 0.3V at 3GHz and WLPSD has the lowest voltage range of 0.1V for both 5GHz and 3GHz. Finally, Table 6.6 and Table 6.7 document the sensitivity metrics across design knobs for 5GHz and 3GHz frequencies respectively. Across various types of RAs the maximum sensitivity occurs with V_{DD} change. The second maximum sensitive design knob is the frequency. The third maximum sensitivity occurs using RA strength change, and the forth and the lowest sensitive design knob is the temperature. For frequency changes from 5GHz to 3GHz, the Λ_{RA} decreases for WLPHD, BLPHD, WLPSD, and BLPSD, but increases for WLPWD and BLPWD. On the other hand, except for WLPSD and BLPSD (do not change), all the other RAs have a decreased sensitivity of $\Lambda_{V_{DD}}$ with lowering of the clock frequency. The sensitivity metric Λ_{V_T} increases with lowering clock frequency for WLPHD, BLPHD, WLPSD, and BLPSD; however, for WLPWD and BLPWD, it decreases.

Table 6.4: RA range for temperature range of -40C to 85C at 0.8V V_{DD}.

Freq (GHz)	WLPHD (%)	BLPHD (%)	WLPSD (%)	BLPSD (%)	WLPWD (%)	BLPWD (%)
5	17.81	29.69	62.5	125	18.56	18.56
3	17.81	29.69	712.5	250	6.19	12.38

Table 6.5: V_{DD} range for RA range of 17.81% to 5.94% at 27C Temperature.

Freq (GHz)	WLPHD (V)	BLPHD (V)	WLPSD (V)	BLPSD (V)	WLPWD (V)	BLPWD (V)
5	0.40	0.30	0.10	0.15	0.15	0.15
3	0.50	0.30	0.10	0.10	0.10	0.10

Table 6.6: Canary write-ability sensitivity metrics for WLPHD, BLPHD, WLPSD, BLPSD, WLPWD, and BLPWD reverse assists at 5GHz frequency.

Sensitivity metrics	WLPHD	BLPHD	WLPSD	BLPSD	WLPWD	BLPWD
Λ_{RA} at 5GHz 27C 0.8V	71.74	46.90	8.18	8.06	31.61	23.00
$\Lambda_{V_{DD}}$ at 5GHz 27C	5888	6287.36	10240	10240	10168.32	10209.28
Λ_F at 0.8V V_{DD} 27C	125.44	150.02	512	512	512	512
Λ_T at 5GHz 0.8V	0.90	0.97	0.09	1.26	1.32	1.93

Table 6.7: Canary write-ability sensitivity metrics for WLPHD, BLPHD, WLPSD, BLPSD, WLPWD, and BLPWD reverse assists at 3GHz frequency.

Sensitivity metrics	WLPHD	BLPHD	WLPSD	BLPSD	WLPWD	BLPWD
Λ_{RA} at 3GHz 0.8V V_{DD} 27C	65.54	51.14	7.42	7.750	32.35	27.97
$\Lambda_{V_{DD}}$ at 3GHz 27C	2385.92	3020.8	10240	10240	9871.36	10045.44
Λ_F at 0.8V V_{DD} 27C	125.44	150.016	512	512	512	512
Λ_T at 3GHz 0.8V V_{DD}	1.27	1.49	0.42	1.59	0.21	1.17

6.0.8 Results for Wordline type RA for the Readability of Canary Sensor SRAMs

We perform similar experiments using wordline type RAs such as WLPHD, WLPSD, and WLPWD for readability sensitivity exploration across canary design knobs. Note that the wordline type RAs are the only ways to make canaries between wordline and bitline type RAs, which degrades the bitline differential development in a canary SRAM. Moreover, one can use wordline type RAs as write-ability as well as readability RAs, which reduces the canary SRAM circuit design and layout area burden for write and read operations. Table 6.8 and Table 6.9 document the sensitivity metrics for canary readability for 5GHz and 3GHz clock frequencies, those have similar trends compared to the write-ability sensitivity metrics,

Table 6.8: Canary readability sensitivity metrics for WLPHD, WLPSD, and WLPWD reverse assists at 5GHz frequency.

Sensitivity metrics	**WLPHD**	**WLPSD**	**WLPWD**
Λ_{RA} at 5GHz 0.8V V_{DD} 27C	69.93	8.04	51.47
$\Lambda_{V_{DD}}$ at 5GHz 27C	9666.56	10240	10229.76
Λ_F at 0.8V V_{DD} 27C	446.46	512	512
Λ_T at 5GHz 0.8V V_{DD}	0.66	0.92	3.68

Table 6.9: Canary readability sensitivity metrics for WLPHD, WLPSD, and WLPWD reverse assists at 3GHz frequency.

Sensitivity metrics	**WLPHD**	**WLPSD**	**WLPWD**
Λ_{RA} at 3GHz 0.8V V_{DD} 27C	62.00	7.05	18.70
$\Lambda_{V_{DD}}$ at 3GHz 27C	8683.52	10240	10168.32
Λ_F at 0.8V V_{DD} 27C	446.46	512	512
Λ_T at 3GHz 0.8V V_{DD}	3.23	1.07	2.02

as shown in Table 6.6 and Table 6.7.

6.0.9 Pulse Shaping Write and Read RA circuits

This section delineates the pulse shaping practical RA circuits and compares them. We only discuss wordline type RA circuits for WLPHD, WLPSD, and WLPWD, as it can be used as write as well as read RAs. Figure 6.14a shows a typical wordline driver without any RA circuits. Figure 6.14b shows a wordline driver with a fixed WLPHD RA circuit. Here, an NMOS diode M_{RA0} connects in series with another NMOS switch M_{RA1}. The M_{RA1} can be turned on by asserting the WLPHDON signal to logic high. The wordline driver can retain its original output waveform if WLPHDON asserts to the logic low. We compare the circuit under the dashed area with the other RA circuit types. To achieve a range of RA settings, a designer can add similar WLPHD RA paths with more control knobs, such as the WLPHDON signal. Figure 6.14c shows a wordline driver using WLPSD RA circuit. The pulse-slope-degradation circuit uses a weak PMOS transistor M_{RA} in series with the final wordline driver inverter. Asserting the signal WLPSDONB to logic low activates the WLPSD RA, otherwise, for WLPSDONB being logic high the wordline driver acts as a usual

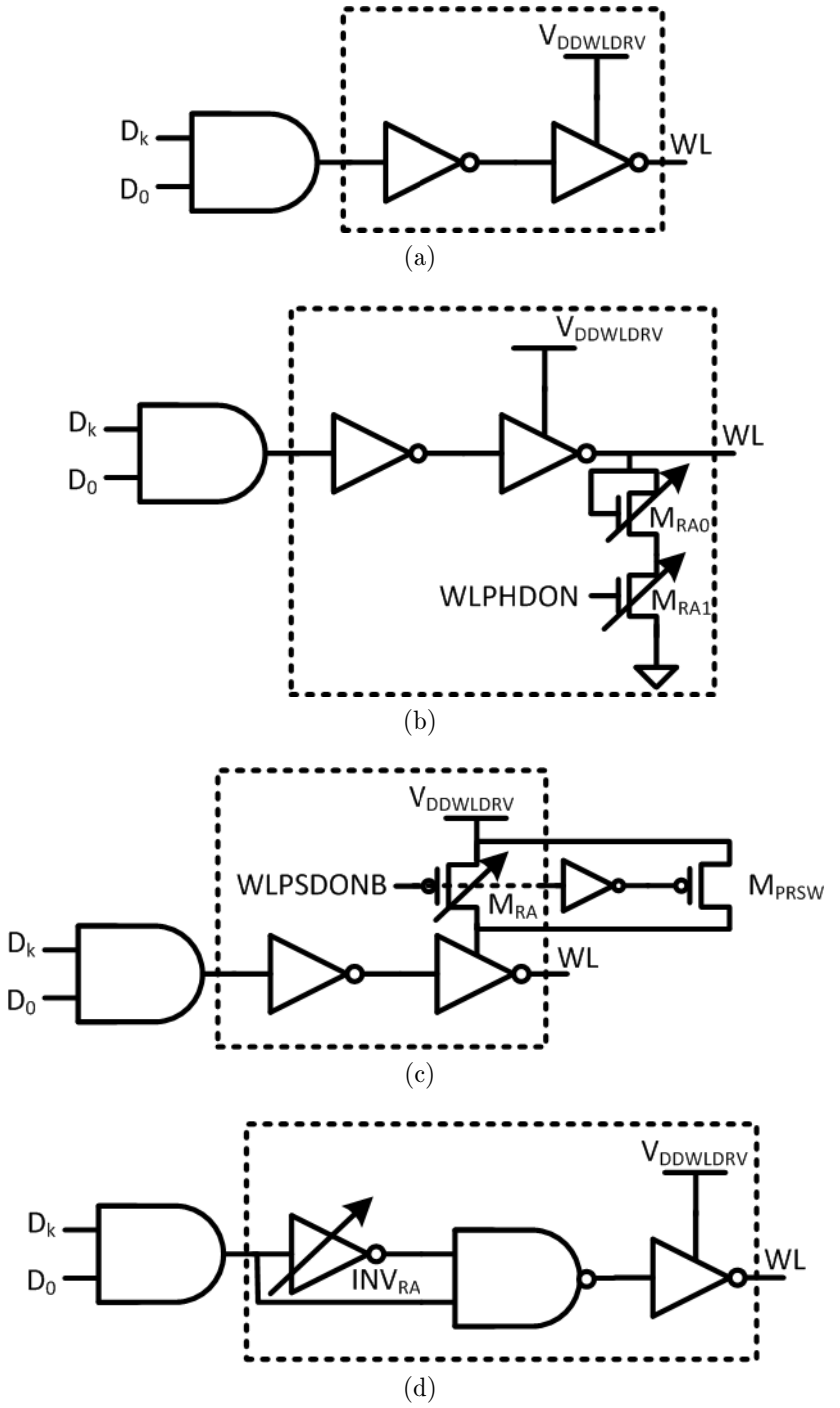


Figure 6.14: (a) Typical wordline driver without reverse assists. (b) Wordline driver using wordline pulse-height-degradation(WLPHD) reverse assist. (c) Wordline driver using pulse-slope-degradation (WLPSD) reverse assist. (d) Wordline driver using pulse-width-degradation (WLPWD) reverse assist.

Table 6.10: Comparison of WLPHD, WLPSD and WLPWD RA across normalized area, energy and FOM metrics.

RAType	**Without RA**	**WLPHD**	**WLPSD**	**WLPWD**
Normalized area	1	7.6	1.025	4.425
Normalized energy	1	2.988	0.357	1.977
FOM ($\frac{\mu}{\sigma}$)	-	16.49	22.48	7.68

wordline driver. Similarly, Figure 6.14d shows a wordline driver using WLPWD RA circuit. Here, a variable delay inverter INV_{RA} creates the appropriate delay required to generate the shorter pulse width of the WLPWD RA. A control can be added to further select various delay line using a multiplexer to adjust the WLPWD RA for various requirements for an RA range. We design the wordline RA circuits under the dashed area in a commercial 14nm FinFET technology for comparison across energy, area, and figure of merit (FOM). Here the FOM defines as the ratio of statistical mean to the statistical standard deviation ($\frac{\mu}{\sigma}$). We design the wordline RAs with the assumption that all of them reaches the max failure rate. Thus, the WLPHD is designed using 41.563% RA, WLPSD is designed using 437.5% RA, and WLPWD is designed as 61.875% RA for read and write operations. The FOM is the measure of variation of pulse height for WLPHD, pulse slope for WLPSD, and pulse width for WLPWD RAs. Table 6.10 shows the comparison of the three RAs for energy, area, and FOM metrics. Noticeably, the WLPSD RA has the lowest penalty in area and energy compared to the WLPHD and WLPWD RA circuit schemes. Moreover, the variation of pulse slope is the lowest. Thus, the WLPSD RA has the highest FOM number.

6.0.10 Conclusions

Selecting a reverse assist depends on the V_{MIN} tracking specifications. If a user wants to track only small hops of the supply voltage from 0.8V to 0.7V (100mV) across processes, frequency, and temperature variation, it makes sense to select a WLPSD or BLPSD reverse assist, as they have the highest sensitivity to detect changes in canary failures. On the other hand, for long hops of supply voltages such as 0.8V to 0.5V (300mV), selecting a less sensitive

WLPHD or BLPHD would be better to slowly change the number of canary failures to cover the entire 0.8V-0.5V supply range. Moreover, PSD type RAs have the biggest RA range that is tantamount to ease in designing an RA to pick some set of slopes compared to picking a pulse height or width for RA-based canary design. Lastly, WLPSD RA circuits are the best choice regarding lowest-cost energy, and area penalty and they have the best figure of merit in the variation of the RA slopes.

6.0.11 Acknowledgment

We thank NVIDIA and DARPA for supporting and funding this work.

Chapter 7

An Ultra Low-power Self-Tuning SRAM Architecture using In-situ Dynamic V_{MIN} Tracking Canary Sensors

[1]This chapter presents an adaptive, closed-loop SRAM that employs multiple combined peripheral assists (CPA) for both read and write and V_{MIN} tracking in-situ canary sensors that extends the operating range of a 256kb 6T SRAM by 67%. The SRAM system operates from 1.2V full-scale supply down to 0.38V deep-subthreshold voltages. The system uses reverse assists to tune canary bitcells for a closed loop control of the V_{DD}, which tracks the SRAM minimum operating voltage (V_{MIN}) at a specified operating frequency. The conventional 6T SRAM usually has higher V_{MIN} than logic circuits across process, voltage, and temperature (PVT) variations [57] [58] [59] [60], which adds large V_{MIN} guard-band. The use of peripheral assist improves the SRAM V_{MIN}; however, does not remove the V_{MIN} guard-band. Our design uses CPA and canary-based V_{MIN} tracking to minimize V_{MIN} guard-banding and maximize

[1]This chapter is based on the published paper titled "A 256kb 6T Self-Tuning SRAM with Extended 0.38V-1.2V Operating Range using Multiple Read/Write Assists and V_{MIN} Tracking Canary Sensors" [AB6].

the operating range for ultra-low power (ULP) applications. The SRAM system is compatible with the sub-threshold logic, too. Thus the system retains the density of the 6T bitcell and meets the ULP and varying frequency needs of a wide-range Internet of Everything (IoE) applications.

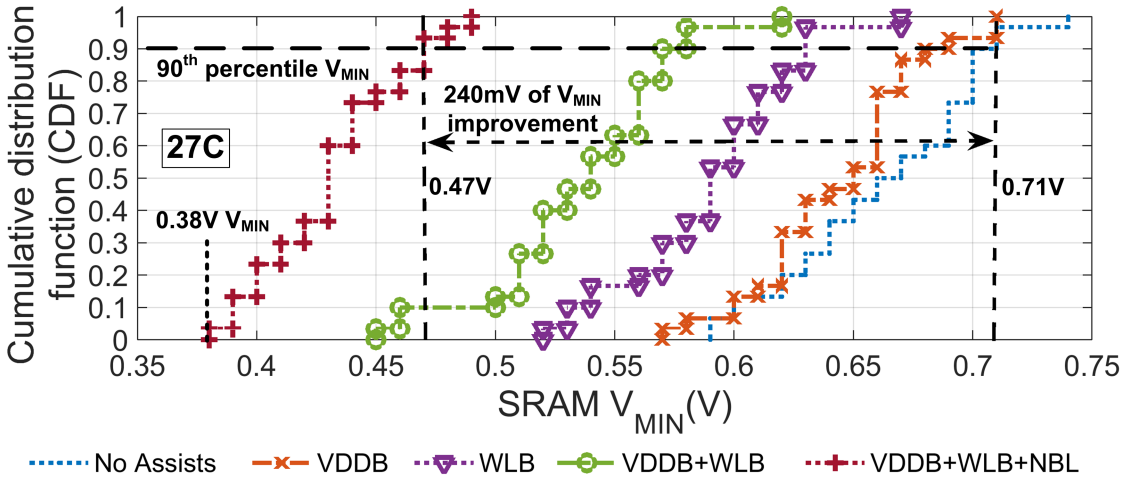


Figure 7.1: Measured CDF of 256kb SRAM V_{MIN} showing 90th percentile V_{MIN} improvement of 240mV using combined assists of V_{DD} boosting (VDB), WL boosting (WLB), negative bitline (NBL) (© 2017 IEEE).

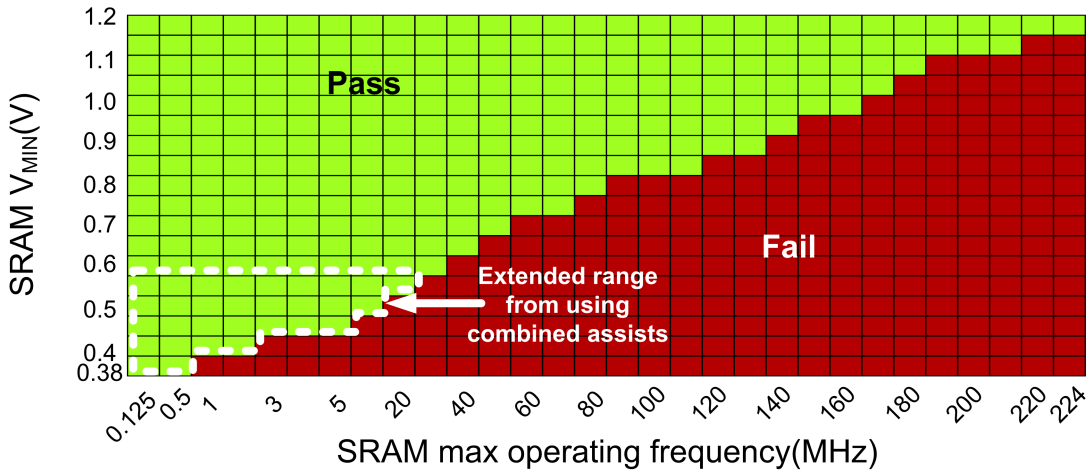


Figure 7.2: Measured V_{DD} Shmoo of the 256kb SRAM (© 2017 IEEE).

ULP Battery-operated or harvested energy IoE devices mostly operate at lower frequencies (10 kHz to 10 MHz) [84] [85]. Thus, there is a need to expand the 6T SRAM operating range in scaled sub-threshold or near-threshold supply voltages to achieve low power operation at

lower frequencies. Peripheral bias-based assist techniques lower SRAM V_{MIN} [57] [58] [60]; however, selecting the best CPA for lowering SRAM V_{MIN} depends on the supply voltage, which could affect trading off the power or performance. Figure 7.1 shows the measured cumulative distribution functions for the SRAM with three peripheral assists: (1) V_{DD} boosting (VDB) for the improvement of low-voltage readability and half-select [57] [60] read-stability; (2) wordline (WL) boosting (WLB); and (3) negative bitline (NBL) for improving write-ability. Using CPA of three assists achieves a 240mV of V_{MIN} improvement of 90th percentile V_{MIN}, which beats the V_{MIN} improvements of other single or combinations (Figure 7.1) of assist. However, fewer peripheral assists can save power overhead when the target V_{DD} is higher for a corresponding frequency.

Figure 7.2 shows the measured Shmoo plot highlighting the extended V_{MIN}-frequency range, which uses the CPA for the 256kb SRAM system. Assists alone requires V_{MIN} guard-banding to ensure functional operation of SRAMs across all chips across PVT, which reduces the potential power savings. Maximizing the benefits of CPA involves runtime determination of SRAM V_{MIN} [86], which decreases the guard-banding of SRAM V_{MIN} at a given frequency. However, this technique suffers from a substantial penalty in the number of clock cycles for writing and reading the whole SRAM. Moreover, there is an additional energy overhead using a built-in-self-test (BIST), which uses this scheme. On the other hand, a smaller sized in-situ canary sensor SRAM-based V_{MIN} tracking [87] allows each chip to operate at or near its V_{MIN} for much lesser clock cycles and energy.

7.0.1 Block Diagram of the System

Figure 7.3 depicts our full SRAM system, which consists of a 256kb SRAM in 4 sub-arrays (mats). Each mat has 4 banks of 128x128 6T bitcells, and each bank has 1 row of 128 canary bitcells. Thus there are 2kb canary bitcells in total in the SRAM system. The other components of the system are an assist controller (ASC), a frequency-to-digital converter (FDC), and a built-in self-test (BIST) block for the core SRAM and the canary bitcells

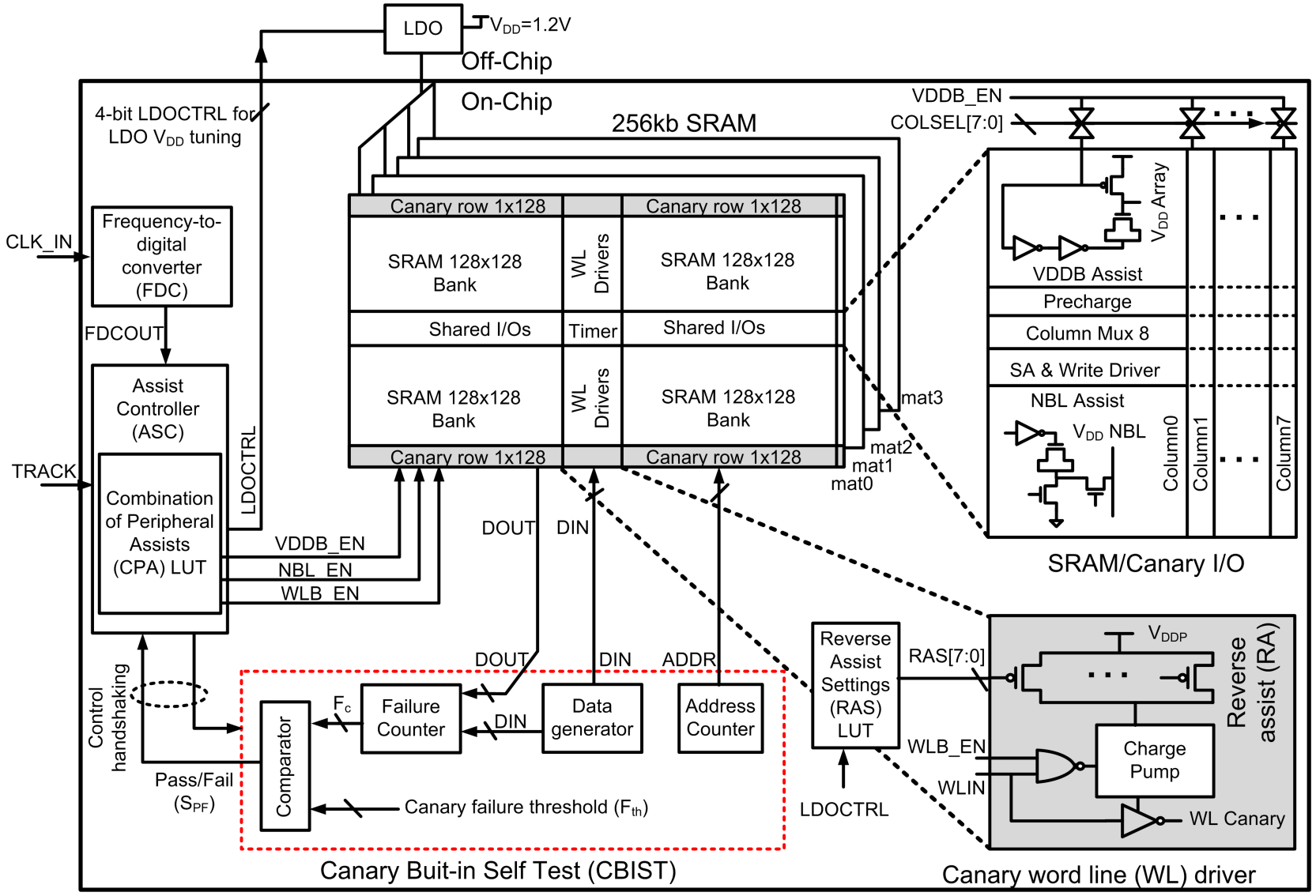


Figure 7.3: The system-level block diagram for the 256kb 6T self-tuning SRAM subsystem showing subcomponents (© 2017 IEEE).

(CBIST). The canary cells share the same peripheral circuits used for the core SRAM, such as write drivers, sense amplifiers, precharge circuits, etc. However, the canaries have dedicated reverse assist (RA) circuits [87], which controls and tune the write-ability and readability of the canaries by degrading the slope of the canary WL signal using eight programmable settings. The CBIST tests the canary SRAM whether the number of failures overshoots a given threshold value and provides the status of the failures to the ASC.

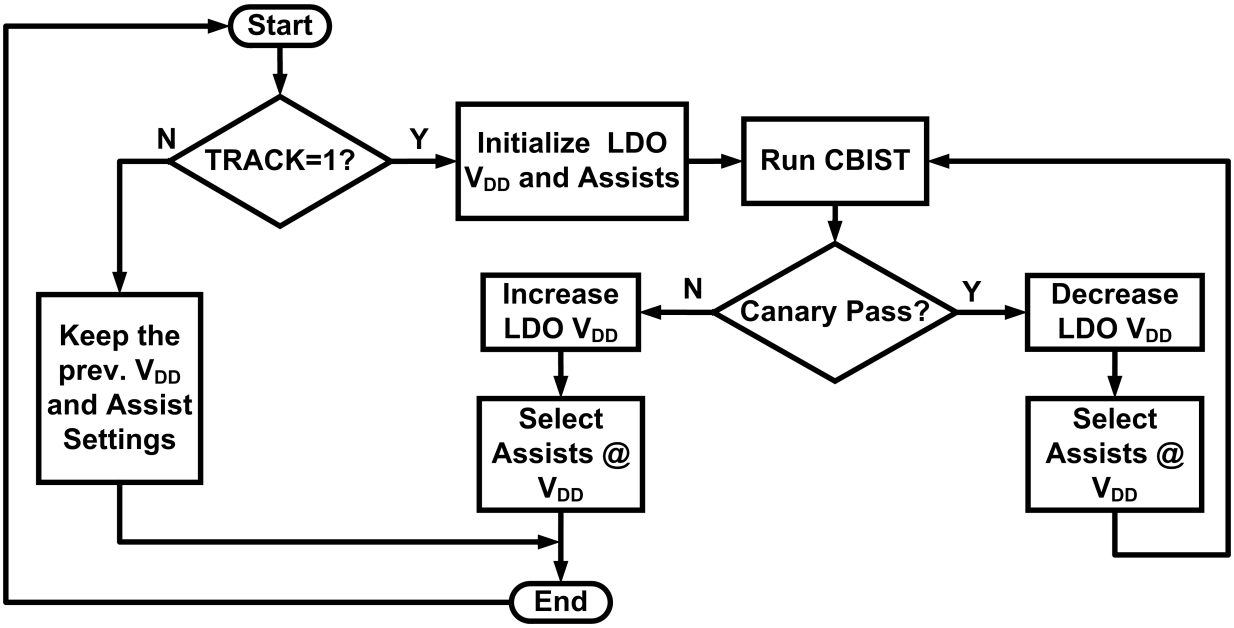


Figure 7.4: Flowchart for canary V_{MIN} tracking (© 2017 IEEE).

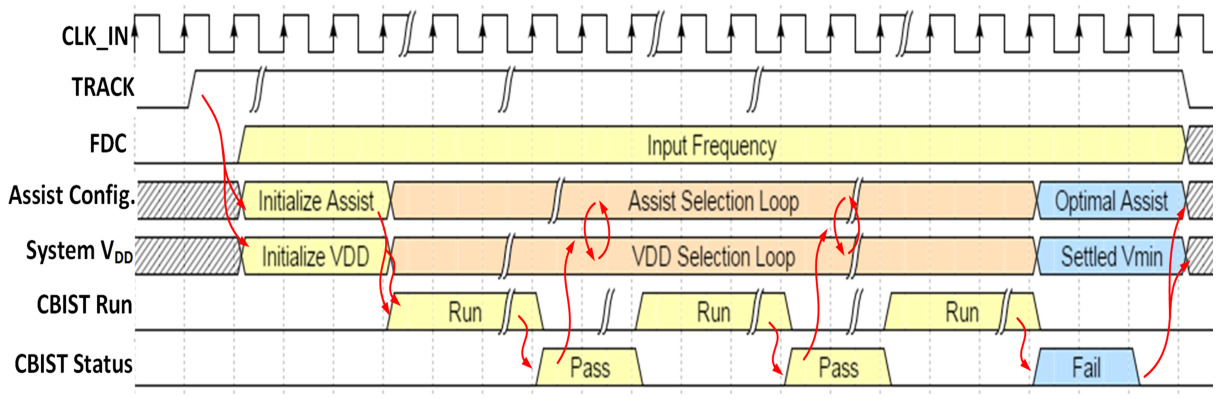


Figure 7.5: The system waveforms for the V_{DD} self-tuning strategy of the 256kb 6T self-tuning SRAM (© 2017 IEEE).

7.0.2 Self-tuning Strategy and Canary feedback Mechanism of the System

Figure 7.4 and Figure 7.5 show the self-tuning strategy for SRAM V_{MIN} tracking, dynamic control over combined peripheral assists, and SRAM V_{DD} selection employing in-situ canary sensor SRAMs. If a user enables tuning by asserting TRACK=1, the FDC converts the input clock (CLK_IN) frequency to a 16-bit digitized output (FDCOUT). The ASC initializes an (off-chip) Low-Dropout (LDO) regulator to an initial V_{DD} for the given digitized frequency FDCOUT. The ASC then selects CPA configuration for the current V_{DD} from a given look-up table (LUT). This LUT-based assist selection flexibly optimizes SRAM energy based on measured characterization across V_{DD}. The ASC then iterates to search the corresponding V_{MIN} for the given frequency depending on the canary outputs. The CBIST controls canary write and read operations across all canary addresses and calculates the number of canary failures (F_c). After that, the CBIST compares F_c with a given canary failure threshold value (F_{th}) to generate a pass/fail signal (SPF). If the CBIST passes, the ASC reduces V_{DD} by updating a 4-bit signal (LDOCTRL), which controls the off-chip LDO. The ASC reiterates this process until the CBIST fails, then it increases the V_{DD} to the last operational V_{DD}. Thus, the closed-loop V_{MIN} tracking work using canary sensor SRAM. The SRAM retains its data throughout the canary tuning process. The canary tuning can be re-run to reconfigure the SRAM V_{MIN} if the frequency or temperature changes.

As the CPA expands the operating range of our SRAM subsystem, the canary feedback is crucial, which ensures that V_{DD} scaling stops before the core SRAM bits start to fail. The RA [87] induces canary failures ahead of the core bits using eight programmable reverse assist settings (RAS). As the canaries are the same core SRAM cells with applied RAS, the canary failure distribution is a shifted version of the core cells, which tracks with frequency and temperature [87] variations. This enables us to set F_{th} based on measured CBIST results from a few dies for calibrating the canary failures relative to the core SRAM cells. Thus, all the SRAM chips track their V_{MIN} using the closed-loop canary sensors.

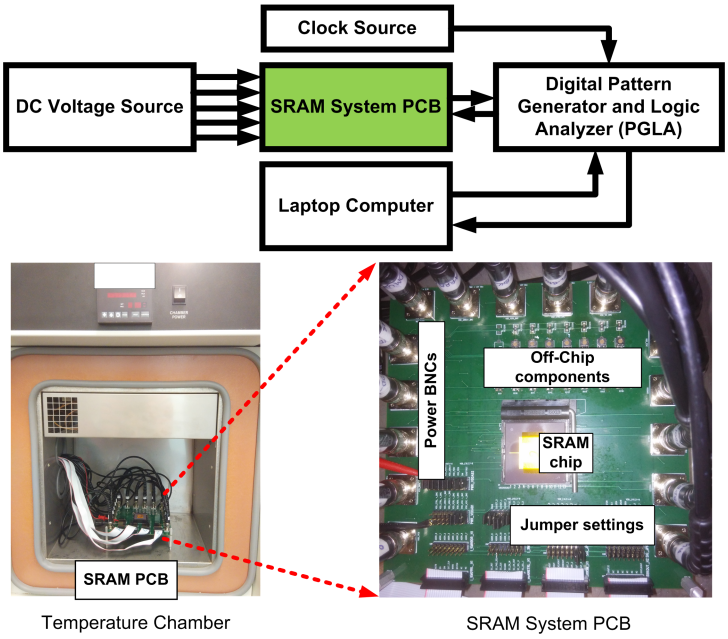


Figure 7.6: Experimental setup for the chip measurements (© 2017 IEEE).

7.0.3 Experimental Setup

We show the experimental setup for the measurement of data in Figure 7.6. Multiple DC voltage sources supply power to the SRAM printed circuit board (PCB). A digital pattern generator (PGLA) generates the control waveforms for the SRAM chip. An external clock source provides a stable clock to the PGLA, which generates a clock signal to the PCB and the testchip. For waveform generation and data collection, a laptop computer controls the PGLA.

7.0.4 Measurements and Results

Figure 7.7, Figure 7.8, and Figure 7.9 depict the measured tuning range of canaries and the SRAM V_{MIN} across temperatures and frequencies. We show the distribution of the V_{MIN} reduction (Figure 7.10), using CPA and V_{MIN} tracking across 30 dies. A user can tradeoff V_{MIN} guard-band margin with power savings, using the ASC that selects the F_{th} and uses a LUT to choose the *RAS* and sense amplifier delay depending on the current V_{DD}. Figure 7.7, Figure 7.8, and Figure 7.9 show settled system V_{MIN} values based on design knobs' settings

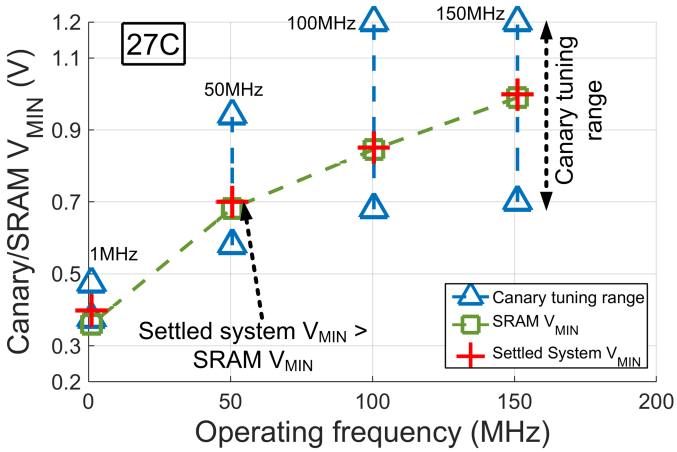


Figure 7.7: Measured canary V_{MIN} tracking across clock frequencies [1 or 10, 50, 100, and 150] MHz and 27C temperature showing V_{MIN} tuning range (© 2017 IEEE).

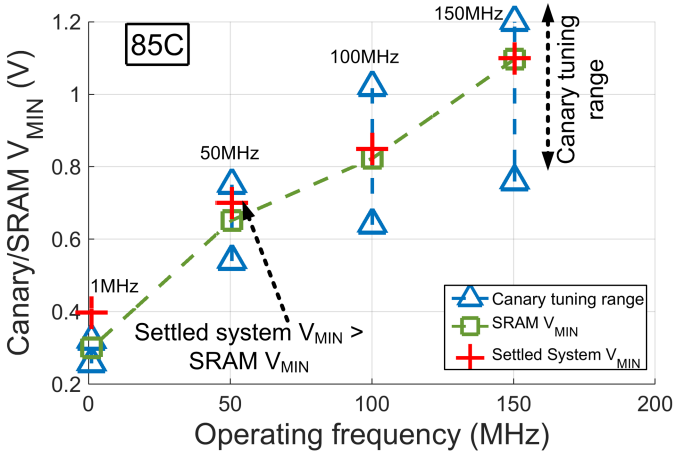


Figure 7.8: Measured canary V_{MIN} tracking across clock frequencies [1 or 10, 50, 100, and 150] MHz and 85C temperature showing V_{MIN} tuning range (© 2017 IEEE).

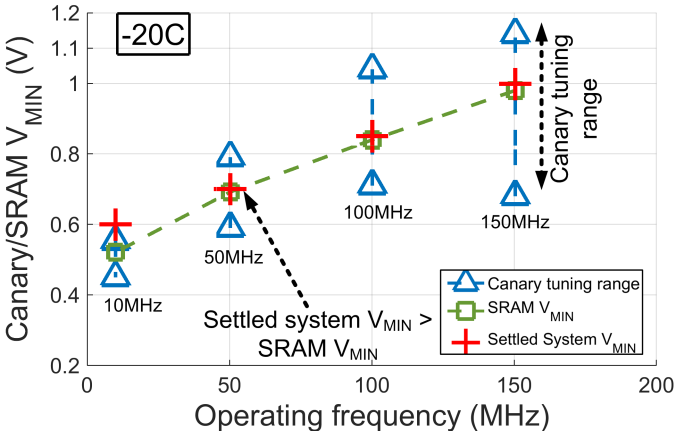


Figure 7.9: Measured canary V_{MIN} tracking across clock frequencies [1 or 10, 50, 100, and 150] MHz and -20C temperature showing V_{MIN} tuning range (© 2017 IEEE).

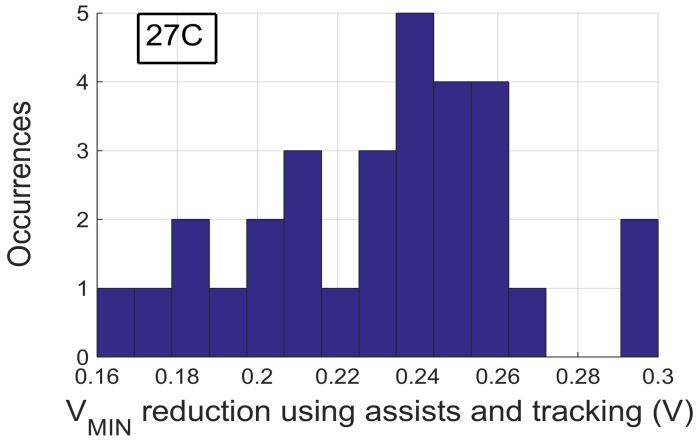


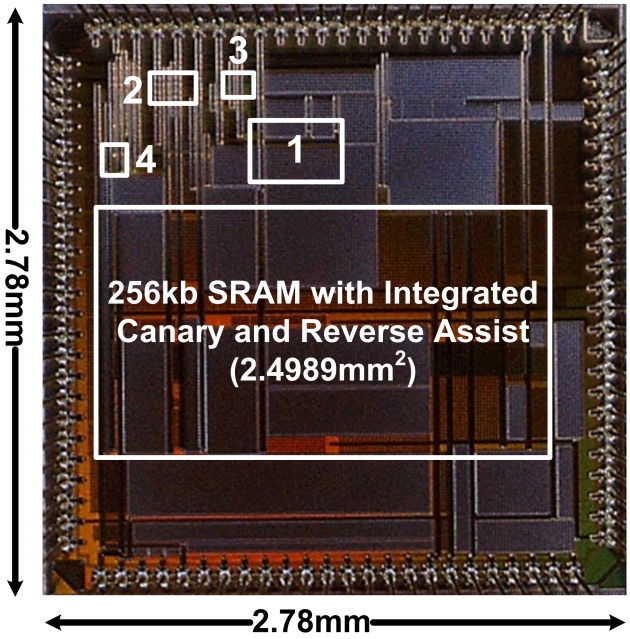
Figure 7.10: The distribution of overall V_{MIN} reduction using assist and canary-based V_{MIN} tracking (© 2017 IEEE).

Table 7.1: Power breakup numbers for the SRAM and the BIST (© 2017 IEEE).

Supply (V)	**SRAM and BIST power**	**SRAM power**	**BIST power**
1.2	$18.3mW$	$14.4mW$	$3.9mW$
0.47	$54.3\mu W$	$49.7\mu W$	$4.6\mu W$
0.38	$12.6\mu W$	$11.4\mu W$	$1.2\mu W$

that aggressively reduce the margin of V_{MIN} guard-band, which maximize the power savings. However, the flexible system allows including an arbitrary guard-band margin.

Using CPA and canary-based V_{MIN} tracking each chip self-tunes to its V_{MIN} for a given frequency, which expands operating range and power savings for low V_{DD} IoE applications. We show an annotated die photo of the SRAM chip in Figure 7.11. This work has an area overhead of 0.77% for the canary bits in each SRAM bank and has 1.8% overhead for the system components without BISTs. The overhead of the combined assist in the SRAM is less than 2.8%. Figure 7.12 shows the power savings for the combined approach, which improves the SRAM V_{MIN} to 0.38V, and allows a 12.4X lower (Figure 7.13) leakage power (9.5pW/bit) compared to 1.2V. Without the canary tracking, process variation would limit V_{DD} scaling to stop at 0.47V using the CPA for ensuring all chips work. With CPA only the system achieves 337X active power reduction for SRAM and BIST. However, V_{MIN} tracking allows an additional 4.3X power reduction by removing the V_{MIN} guard-band for those chips that can function at lower V_{DD}. Thus, using CPA and canary tracking the system saves up to



1- Canary and SRAM BIST (0.142mm^2)
2- Assist Controller (ASC) (0.0234mm^2)
3- Frequency to Digital Converter (FDC) (0.0191mm^2)
4- Level Converters (0.00384mm^2)

- Area overhead of canary in each bank is 0.77%
- Area overhead for the total system components without BISTs is 1.8%
- Area overhead for the total system components with BISTs is 7%

Figure 7.11: Annotated die micrograph of the SRAM chip (© 2017 IEEE).

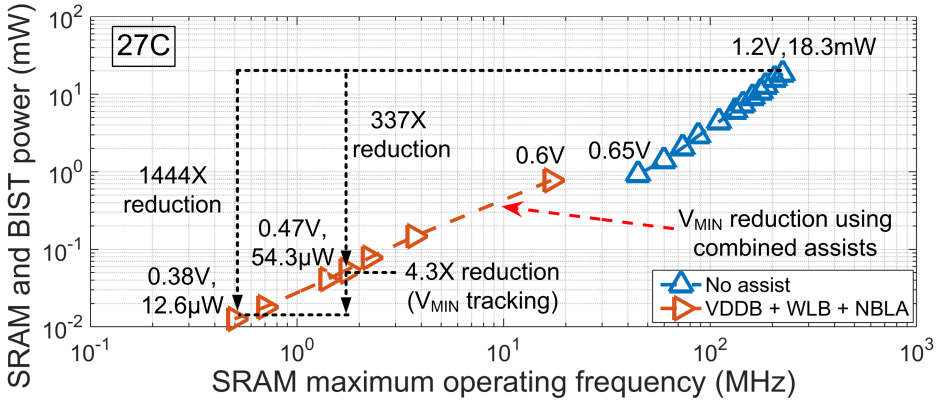


Figure 7.12: Measured active power reduction of SRAM and BIST with combined peripheral assists and V_{MIN} tracking (© 2017 IEEE).

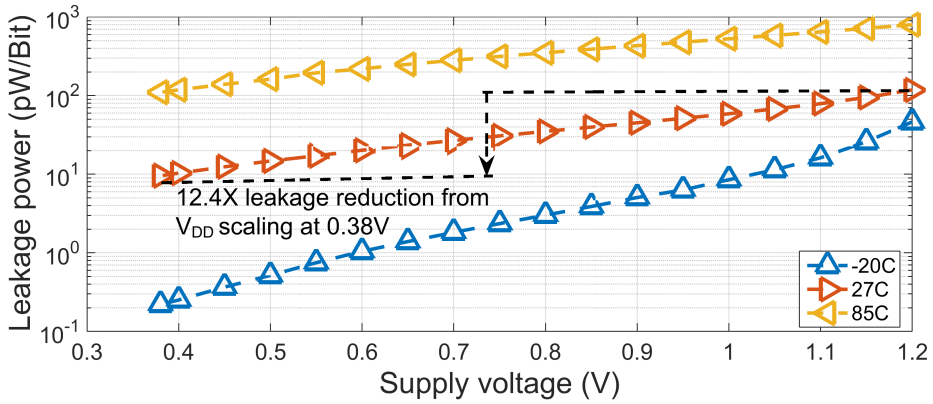


Figure 7.13: Measured leakage reduction from V_{DD} scaling (© 2017 IEEE).

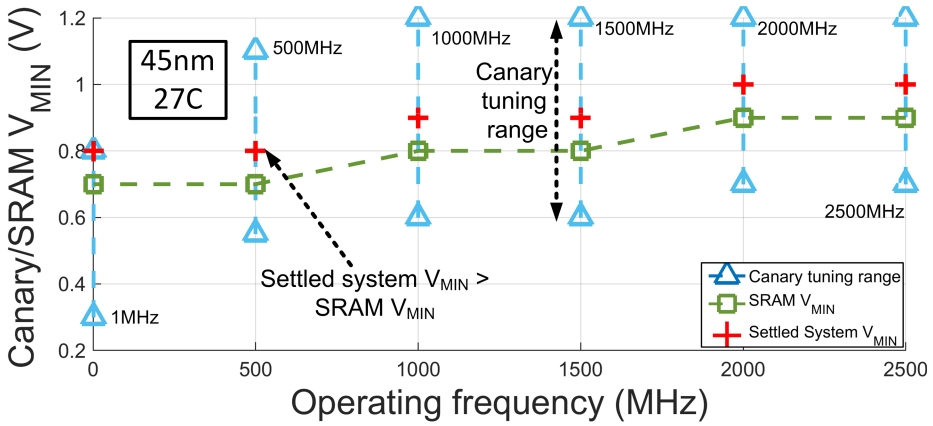


Figure 7.14: Simulation results of canary tuning at 45nm technology at TT_27C corner showing that canary-based system V_{MIN} can be tuned above the SRAM V_{MIN} (© 2017 IEEE).

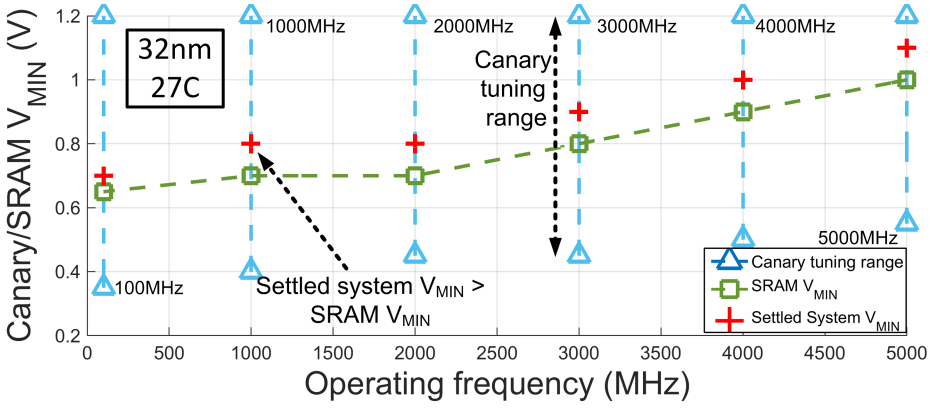


Figure 7.15: Simulation results of canary tuning at 32nm technology at TT_27C corner showing that canary-based system V_{MIN} can be tuned above the SRAM V_{MIN} (© 2017 IEEE).

1444X active power (Figure 7.12). Table 7.1 shows the power breakup of the SRAM and BIST in the testchip. It reveals that these techniques reduce the SRAM power from 14.4mW to 11.4μW, which achieves a 1263X power reduction. Table 7.2 compares this work to the state-of-the-art wide voltage range SRAMs for low power applications. We show in Figure 7.14 and Figure 7.15 that canary-based V_{MIN} tracking scales to 45nm and 32nm technologies for a wide range of voltages and frequencies.

7.0.5 System Level Projected Savings for the Testchip

This section computes the system level battery-life and replacement time savings at different SRAM V_{MIN}s with and without using CPA and in-situ canary sensor SRAM. Here we assume that both the SRAM and logic core shares a single supply rail and SRAM V_{MIN} limits the overall voltage scaling. Additionally, we assume that SRAM consumes 40% energy while the logic core consumes 60% energy from the supply rail or their power dissipation capacitance ratios are 2:3. We further assume that the IoT system uses an A1578 (0.76Wh) battery and the system average power consumption is the same as Apple iWatch average power consumption of 42.2mW at the full-scale supply voltage of 1.2V. Thus using voltage scaling without any peripheral assist applied to the SRAM, we can lower the system supply to 0.71V (Figure 7.1), which is the 90th percentile SRAM V_{MIN}. The corresponding battery-life for a single charge improves from 17.96 hrs to 7.82 days or about 946%. Applying CPA lowers the 90th percentile SRAM V_{MIN} to 0.47V, and the corresponding battery-life for a single charge improves to 6.84 months from 17.96 hrs, which is a 27329% improvement in battery-life. Turning on in-situ canary-based V_{MIN} tracking along with CPA further lowers the SRAM V_{MIN} to 0.38V removing the margin guard-banding. The corresponding battery-life saving for single charge improves to 1.25 yrs from 17.96 hrs, which is 60811% improvement in battery-life. We assume 300 cycles of worst-case maximum usage for the A1578 battery. Thus, the corresponding battery replacement time without CPA (at 0.71V), with CPA (at 0.47V), and with CPA along with canaries (at 0.38V) would be 6.43 yrs, 10+ yrs, and 10+

yrs improved from 0.61 yrs operating at the full-scale supply voltage of 1.2V. Due to the reason that overall battery shelf-life is around 10 years, the battery replacement time for using solo CPA and CPA combined with canaries will be limited by the shelf-life of A1578, which is about 10 yrs. Here we assume a 100% duty cycle for the calculations. The equations used to calculate the battery-life using self-discharge assumptions are derived in Appendix A for the SR416SW Silver Oxide, LIR2032 Lithium-ion, and A1578 Lithium-ion Polymer batteries.

7.0.6 Conclusions

This work expands the 6T SRAM operating range by over 67% (from 1.2V-0.71V=0.49V to 1.2V-0.38V=0.82V, in sub-threshold), which uses three combined read/write assists and in-situ canary sensor SRAM-based V_{MIN} tracking. The SRAM subsystem self-tunes close to the SRAM V_{MIN} across frequencies and temperatures. This adaptive solution allows us to enable a wide-range of IoE applications and achieves up to 1444X active power reduction. The system level IoE battery replacement time could improve to 10+ years operating at 0.38V from 6.43 yrs operating at 0.71V without peripheral assists, assuming that the SRAM and core logic shares the same supply rail. Our canary-based V_{MIN} tracking technique is scalable to 45nm and 32nm technologies.

7.0.7 Acknowledgements

I thank Ningxi for designing the sense amplifier, wordline boosting circuitry, layout integration of the SRAM, and testchip measurements. I am thankful to Harsh for developing the assist controller and frequency to digital converter blocks and the integration of the testchip. This work was funded in part by NVIDIA through the DARPA PERFECT program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

Table 7.2: Comparison table for SRAM subsystem with the state-of-the-art (© 2017 IEEE).

	This Work	**ISSCC '10 [57]**	**ISSCC '12 [58]**	**VLSI '14 [59]**	**ISSCC '15 [60]**
Technology	130nm	45nm	22nm	180nm	28nm
Cell type	6T	8T	6T	8T	6T
Capacity	256kb	512kb	576KB	16KB	256kb
DVS range	1.2-0.38V (850mV)	1.2V-0.57V (630mV)	1V-0.625V (375mV)	1.8V-0.6V (1200mV)	0.9V-0.58V (320mV)
SRAM V_{MIN} tracking	Y	N	N	N	N
V_{MIN}	0.38V	0.57V	0.7V	0.6V	0.58V
Sub-threshold operation	Y	-	-	-	-
Active power	18mW at 1.2Vand $12.6\mu W$ at 0.38V	169mW at 1.2V	-	250mW at 1.8V, 15mW at 0.82V	-
Power reduction using combined assists	337X	-	-	16.4X	
Power reduction using V_{MIN} tracking	4.3X	N	N	N	N
Max power reduction	1444X	-	-	16.4X	-
Min energy/op	25.3pJ/op at 0.38V	-	-	690pJ/op at 0.82V	-
Standby leakage power/bit	9.5pW/bit	644nW/bit	-	-	-
Standby leakage power reduction to V_{MIN}	12.4X	9.4X	-	-	-
Sensor	Canary SRAM	DRV Sensor	-	Energy monitor	-
Combined assist	Any combination of NBL, WLB, VDB	N	TVC-WA, WLUD	N	WLUD, NBL

Chapter 8

Analysis and Design of Reverse Assist-based V_{MIN} Tracking Canary Sensor SRAMs in the presence of Process, Voltage, Temperature, and Frequency Variations

8.0.1 Motivation

With the emerging market for ultra-low power (ULP) battery-operated or energy-harvested Internet of Everything (IoE), there is a higher demand for running high performance and ULP applications in a single device such as smartphones, smartwatches, etc. In those IoE devices, the worst-case high-workload applications drive the design of higher performance SoCs. Such SoC's power consumption must scale to a low-speed-workload mode for ULP application requirements, also. To optimize battery life for time-varying workloads, Dynamic Voltage Scaling (DVS) [88] is a widely used technique that quadratically lowers SoC power consumption. The digital logic in the SoC can easily scale to a lower V_{DD} to save power for the

low-speed ULP applications. However, due to process and temperature (PT) variation, the 6T SRAM minimum operating voltage (V_{MIN}) is heavily guard-banded to ensure functionality across PT conditions. Thus, the conventional 6T SRAM sharing the same supply rail with SoC's logic limits the overall V_{DD} scaling.

Peripheral assists [89] lower the SRAM V_{MIN} (SV_{MIN}) to improve this V_{DD} scaling-bottleneck trading off area and energy. However, this method does not eliminate the SV_{MIN} guard-bands across PT variations, which results in significant power overhead. On the other hand, splitting the power rail for SRAM core and periphery, known as the dual rail (DR) [90] technique, is another solution to overcome the DVS bottleneck for SoC core logic. However, this method has higher overhead in the routing of multiple power rails, and area and efficiency overheads of separate DC-DC converters.

In the previous chapter, our work [91] uses multiple combined peripheral assists, and canary SRAM sensors, which lower SV_{MIN} and reduce its guard-bands arbitrarily in conventional 6T SRAMs. Using both techniques save overall 1444X power and give a 67% wider 1.2V-0.38V DVS range at a low cost of about 3% area overhead. The canaries in [91] use reverse assist (RA) [81] settings (RAS) and canary failure threshold (F_{th}) knobs that tune the canary V_{MIN} arbitrarily close to SV_{MIN}, which reduces the SV_{MIN} guard-band. The worst-case SRAM bitcell maps into the canaries, after the canaries tune with a RAS and F_{th} setting at a particular frequency. If the frequency, temperature, etc. parameters change within a limit, the canaries will always fail earlier than the SRAM bits. Also, while scaling the V_{DD}, this scheme tracks SRAM V_{MIN}. However, the design time of the RA-based canary took more than a month during the design and tapeout of [91] testchip, which could add additional design cost for industrial designs. Moreover, our work [91] shows a specific canary design solution for SV_{MIN} tracking, which lacks information about how to design and analyze an RA to make canaries keeping process, temperature, etc. knobs' variations in mind. Thus, an important research question arises: how to choose RAS and F_{th} design knob values across process, voltage, temperature, and frequency variation or what is the necessary and satisfying

condition of the RA design knobs for canaries to track SV_{MIN} optimally? Although RA-based SV_{MIN} tracking canaries solve arbitrary lowering of SV_{MIN} guard-bands, none of the SRAM design exploration research or commercial tool-flows, such as ViPro [92] [93], currently support the design and analysis of RAs for canaries. In this chapter, we propose for the first time a mathematical formulation of the necessary condition for the design and analysis of canaries to track SV_{MIN} using RAS and F_{th} knobs. This work leads to the optimal tracking of SV_{MIN} across process, etc. design knobs' variations. The work further demonstrates for the first time a set of algorithms and a tool-flow named RADA: Reverse Assist Design and Analysis flow, that tests the necessary conditions for SV_{MIN} tracking across canary design knobs.

8.0.2 Contributions

The contributions of the work are summarized as follows:

- A mathematical formulation of the necessary satisfying condition for reverse assist-based canary design knobs to optimally track SV_{MIN} across process, etc. design knobs' variations.
- A set of algorithms that use the necessary condition stated above to analyze the RAS and F_{th} design knobs across process, voltage, temperature, and frequency variations.
- A Perl-based tool-flow that implements the algorithms for design and analysis of the RAs independent of process technology
- Results of the RADA tool to investigate optimal SV_{MIN} tracking in 45nm bulk, 32nm FDSOI, and 14nm FinFET technology.

8.0.3 SV_{MIN} Tracking Using Canary Sensors

The SRAM V_{MIN} (SV_{MIN}) is a function of intra-die and inter-die process variation, frequency, temperature, aging, etc. parameters, and it is hard to predict the actual SV_{MIN} of

an SRAM chip in real time. Thus, a 6T SRAM's SV_{MIN} is heavily guard-banded to guarantee functionality even at the worst-case corner. Removing this SV_{MIN} guard-banding can save more than 50% energy [81]. One way to remove this guard-band is to measure the SV_{MIN} for each chip and its SRAM macros. One can measure the SV_{MIN} using a built-in self-test (BIST) [94] engine. However, depending on the capacity of the SRAM, it may take a large number of cycles to determine the SV_{MIN} for a single change in frequency or temperature parameters. Moreover, the processor core must wait for these long BIST cycles, which can cause significant delays and increase the response time in the execution of time-critical programs. On the other hand, a smaller sized canary SRAM sensor can track SV_{MIN} that reduces its guard-band, which improves the timing overhead of SV_{MIN} measurement using a BIST, also.

8.0.4 SV_{MIN} Tracking Architecture of Reverse Assist-based Canary Sensor SRAMs

Canary SRAMs are first demonstrated to track [76] SRAM data retention voltage (DRV) metric. Authors in [76] show that canary bitcells with a voltage bias knob weaken canaries to fail earlier than the population of core SRAM bitcells. Tweaking the voltage bias knob in [76] degrades the canaries, and the DRV distribution of the canaries shift relative to the SRAM DRV distribution. Thus, aptly tuning the canary voltage bias can fail the canaries earlier than the core bitcells begin to fail. On the other hand, our work in [81] first proposes to track dynamic write SV_{MIN} using canaries. The work also proposes the canary design knobs for SV_{MIN} tracking and formulates their inter-relations. In this work, we use some core bitcells and a technique named reverse assist (RA) that degrades the core bitcells to behave like canary bitcells. The benefits of using RA-based canaries are to control and tune the canaries post-fabrication.

Peripheral assists improve the SV_{MIN} for write-ability, readability, and read-stability criteria. On the contrary, a reverse assist (RA) degrades the SRAM bitcell's write-ability,

readability, and read-stability, and makes it a canary sensor bitcell that fails earlier than the SRAM bitcell start to fail in V_{DD}. Using multiple RA-based canary sensors one can tune the canary V_{MIN} (CV_{MIN}) to be greater or equal to the SV_{MIN}. This tuning of CV_{MIN} using RA ensures that the worst-case SRAM bitcell maps into canaries. Thus, once tuned, the RA-based canaries can track the SV_{MIN} across changes in frequency and temperature. Tuning of canaries requires another design knob named failure threshold condition (F_{th}) [81], which defines how many canary bits fail to make it an actual canary failure point. Our work in [91] use a wordline slope degradation type RA (WLRA) and some measured F_{th} values to make canary sensors track the SV_{MIN} that reduces the SV_{MIN} guard-band. RA-based canary sensors require a closed-loop SRAM architecture [91] for SV_{MIN} tracking. In this architecture, a digitally controlled DC-DC converter or a low-dropout regulator (LDO) provides the necessary granularity for V_{DD}s for a wide-range DVS requirement based on the application. A frequency to digital converter (FDC) translates the incoming clock frequency to an assist controller (ASC) that controls the peripheral assists in the SRAM with integrated canary sensors. The ASC further controls the LDO and a canary BIST (CBIST) that writes and reads a known pattern to the canary sensors. The CBIST checks if the number of canary failures is greater than the F_{th} value specified, and passes the controls to the ASC to adjust the LDO V_{DD} by increasing or decreasing it, accordingly. Thus, the final LDO voltage, after canary-based V_{MIN} tracking, settles down to a CV_{MIN}, which is greater or equal to the SV_{MIN}. However, our works [91] [81] do not clarify the necessary and satisfying conditions for RA-based canaries to track SV_{MIN} across design knobs for the general and optimal case. The next section formulates the necessary satisfying condition for SV_{MIN} tracking for RAS and F_{th} knobs with other design parameters.

8.0.5 Derivation of Optimal SV_{MIN} Tracking Condition

To derive the necessary and satisfying condition for SV_{MIN} tracking using RA-based canary sensors across process, etc. design knobs' variations, let us assume the following:

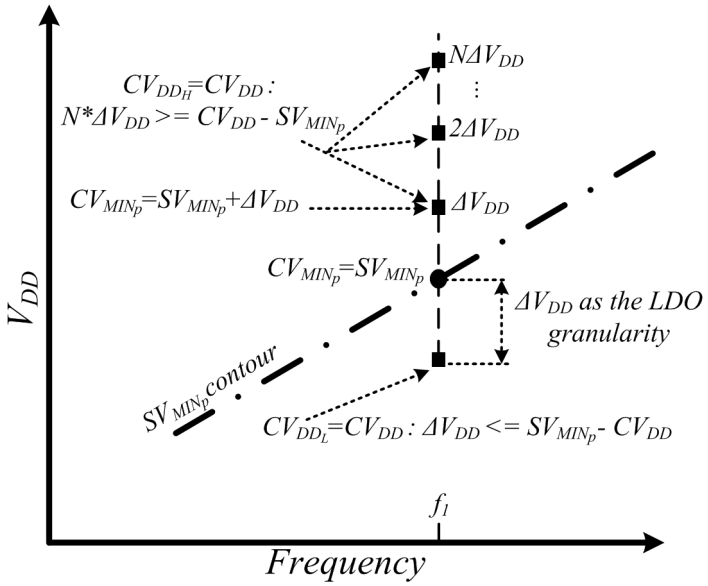


Figure 8.1: Possible canary supply voltages (CV_{DD}) for choosing an optimal canary V_{MIN} (CV_{MINp}) that tracks the SRAM V_{MIN} (SV_{MINp}).

$P = \{TT, SS, FF, SF, FS...\}$ is a finite set of process corners at a finite temperature $T_N \in T : \{T_0, T_1, ...T_N\}$; $F_{th_d} \subset N : N = \{1, 2, 3, ...\}$, is a finite set of failure threshold conditions (F_{th}); $RAS = \{RAS_0, RAS_1, ...RAS_N\}$, is a finite set of reverse assist setting labels. Here we denote SV_{MIN} as the SRAM V_{MIN}, CV_{MIN} as the canary V_{MIN}, and CV_{DD} as the canary supply respectively. The LDO or the DC/DC granularity ΔV_{DD} sets how close one can map the CV_{MIN} toward SV_{MIN} using the F_{th} and RAS design knobs. Thus, a non-optimal solution for system settled V_{DD} or the final CV_{MIN} at process p is given by the equation (8.1).

$$CV_{MIN_p} > SV_{MIN_p} \; \forall \; p\colon p \in P \tag{8.1}$$

Due to the ΔV_{DD} granularity set by the LDO according to the architecture [91], the SV_{MIN} can only be one of the V_{DD} set by the LDO or in-between some values. Thus, the tracking condition for the system settled V_{DD}, or the final CV_{MIN} can be given as follows.

$$N * \Delta V_{DD} \; \geq CV_{MIN_p} - SV_{MIN_p} \; \geq 0 \; \forall p\colon p \in P \tag{8.2}$$

Where N can be chosen arbitrarily (based on high sigma confidence level for SV_{MIN} simulation) or optimally ($N = 1$) to map the CV_{MIN} closest to the SV_{MIN}. To satisfy the equation (8.2), the canaries must fail at a CV_{DD} to settle to a $CV_{MIN} = CV_{DD} + \Delta V$ according to the canary tuning algorithm [91]. Hence, the corresponding CV_{DD} can take two values given by equations (8.3) and (8.4). Thus, the CV_{DD} can be just greater than SV_{MIN} or just below SV_{MIN} (Figure 8.1).

$$CV_{DD_H} = CV_{DD} \ : N * \Delta V_{DD} \ \geq CV_{DD} - SV_{MIN_p} \tag{8.3}$$

$$CV_{DD_L} = CV_{DD} \ : \Delta V_{DD} \ \leq \ SV_{MIN_p} - CV_{DD} \tag{8.4}$$

Hence, the corresponding failure rate of canaries at CV_{DD_H} and CV_{DD_L} using reverse assist RAS_N can be given by $P_{fc}(CV_{DD_L}, RAS_N)$ and $P_{fc}(CV_{DD_H}, RAS_N)$, respectively. Thus, the upper and lower bounds of canary failures, or the failure threshold conditions (F_{th}) are given as follows.

$$F_{th_H} = N_f(CV_{DD_L}, RAS_N) = P_{fc}(CV_{DD_L}, RAS_N) * C \tag{8.5}$$

$$F_{th_L} = N_f(CV_{DD_H}, RAS_N) = P_{fc}(CV_{DD_H}, RAS_N) * C \tag{8.6}$$

Where, C is the number of canary bits in the design and N_f is a table of canary failures across CV_{DD} and RAS values. The corresponding valid range of F_{th} values to achieve the system settled V_{DD} optimally closed to the SV_{MIN} such that CV_{MIN}=SV_{MIN} or CV_{MIN} =$SV_{MIN} + \Delta V$ (Figure 8.1) can be given as the follows.

$$F_{th_p} = F(p, RAS_N) = \{F_{th_L} + 1 \ to \ F_{th_H}\} \tag{8.7}$$

Here F is a table of sets of F_{th} values across process and RAS values. For a particular

process p, if F_{th_p} exist, then equation (8.1)-(8.7) will be valid, or the system settled V_{MIN} tuned by the LDO would be optimal that satisfy the equations if and only if the following is true.

$$\forall\ p : p \in P\ \exists\ a\ RAS_N \colon RAS_N \subset RAS, |RAS_N| \geq 1 \tag{8.8}$$

$$F_{th_p} \subset F_{th_d}, |F_{th_p}| \geq 1 \tag{8.9}$$

Across process variation we define a common F_{th} (F_{th_k}) set that can track SV_{MIN} using a single RAS_N setting, where the F_{th_k} is given by the following equation.

$$F_{th_k} = \cap F_{th_p}\ \forall\ p \colon p \in P, RAS_N \in RAS \tag{8.10}$$

The sets F_{th_k} and RAS_N will map the worst-case SRAM bitcells to canaries to track the SV_{MIN} optimally across process variation if and only if, the following are true: The equation (8.1-8.10) holds true, and the following is true:

$$\forall\ p : p \in P\ \exists\ a\ F_{th_k} \subset F_{th_d}, |F_{th_k}| \geq 1 \tag{8.11}$$

It would be useful to have a common F_{th_k} and RAS_N across process corners for RA designs with lower design cost and area overhead. Hence, the optimal SV_{MIN} tracking condition becomes a search problem for a specific type of reverse assist such that the sets F_{th_p}, F_{th_k} and RAS_N must not be a null set (Θ) for each process corners. Note that if the sets F_{th_p}, F_{th_k} and RAS_N are a Θ for any individual process corner, the SRAM designer must change the RAS values or use a different RA type to re-analyze the design space until the sets are non-null.

Algorithm1: Calculates the SV_{MIN} for a given threshold failure rate.
Input: SRAM bitcell netlist, size, read/write operations, peripheral assists, process, V_{DD}s, frequency and temperature.
Output: For each process, temperature, frequency, operations, and peripheral assist conditions
1) Gives the failure rate for write ability, readability or read stability operations.
2) Gives the SV_{MIN} values for SRAM write ability, readability or read stability.

p← All processes, *v*← all voltages, *t*← all temperatures, *f*← all frequencies, o*ps*← all write, read operations, *ast*← all assists.
Meas← intended measured parameter denoting *ops* being passed or failed.

Run High Sigma MC for Test_bench(*p,v,t,f,ops,ast*);
Parse *Meas* for Measurements(*p,v,t,f,ops,ast*);
Pfail_Meas(p,t,f,ops,ast,v) ← (# of Meas failed)/(# MC runs);
if P_{fail}*_Meas(p,t,f,ops,ast,v)* < P_{fail}*_Thresh_Meas*
    SV_{MIN}*_Array(v)* ← *v;*
end if
SV_{MIN}*(p,t,f,ops,ast)* ← *min*(SV_{MIN}*_Array*);

Figure 8.2: An algorithm to calculate the SV_{MIN} for a given set of SRAM specifications.

8.0.6 Algorithms for SV_{MIN} Tracking Condition

The RADA engine implements three algorithms to compute the analysis report for RA-based canary design. Figure 8.2 shows an algorithm that takes the input SRAM specification and calculates the 1) SRAM failure rates for write-ability, readability, or read-stability as per the user specification of operations. It also 2) computes the SV_{MIN} values for the corresponding write-ability, readability, and read-stability operations. Note that this algorithm could use the existing state of the algorithms such as Importance Sampling [79] [80] etc. for the V_{MIN} calculation for a specific set of design knobs. For each process, temperature, frequency, operation, peripheral assist, and supply voltage, the Algorithm1 runs high sigma Monte Carlo (MC) simulation using HSPICE or Solido Variation Designer. After simulation run, the Algorithm1 parses the measurement files for the desired 'Meas' parameter and computes the failure rate $Pfail\_Meas$, as shown in Figure 8.2. The SV_{MIN}_Array is calculated based on V_{DD}s for which $Pfail\_Meas < Pfail\_Thresh\_Meas$ is true, and the SV_{MIN} is calculated as the min(SV_{MIN}_Array).

The Algorithm2 shown in Figure 8.3 calculates the rate of the canary failures for a set of canary design specifications. The inputs to this algorithm are bitcell netlist, canary size, read/write operations, types of reverse assists, the number of reverse assist settings, etc. (Figure 8.3). With these inputs, Algorithm2 calculates the RA-based canary failure rate for write-ability, readability, and read-stability conditions, as specified by the user. It also computes the table for the number of canary failures for canary V_{DD}s and RAS values across process, temperature, frequency, operation, and RA types. Algorithm2 first runs a low sigma MC for each test bench and parses the $Meas\_Canary$ values to calculate the canary failure rate as $Pfail\_Meas\_Canary$. The $Pfail\_Meas\_Canary$ is normalized by multiplying by the number of canaries C [81] to give the V_{DD}-RAS canary failure table (N_f) across process, temperature, frequency, operation and RA types.

On the other hand, with the SV_{MIN} and N_f pre-calculated using Algorithm1 & Algorithm2, the Algorithm3 (Figure 8.4) first calculates the CV_{MIN}s that are just greater or equal

Algorithm2: Calculates the canary failure rates for a set of input RA-based canary design knobs.
Input: Canary bitcell netlist, size, read/write operations, reverse assists, processes, supply voltages, frequencies and temperatures.
Output: For each process, voltage, temperature, frequency, operations, and reverse assist conditions
1) Gives the RA-based canary failure rate for write ability, readability or read stability operations.
2) Gives the table for no. of canary failures for canary V_{DD}s and reverse assist settings across process, temperature, frequency, operations, and reverse assists.

p← All processes, *v*← all voltages, *t*← all temperatures, *f*← all frequencies, *ops*← all write, read operations, *ra*← all reverse assists, *ras*← number of reverse assist settings, C← number of canaries, *Meas*← intended measured parameter denoting *ops* being passed or failed.

Run Low Sigma MC for Test_bench(*p,t,f,ops,ra,ras,v*);
Parse *Meas_Canary* for Measurements(*p,t,f,ops,ra,ras,v*);
P_{fail}*_Meas_Canary(p,t,f,ops,ra,ras,v)* ← (# of Meas failed)/(# MC runs);
N_f*(v,ras)* ← *C*P_{fail}_Meas_Canary(p,t,f,ops,ra,ras,v);*
Canary_Failures(p,t,f,ops,ra) ← N_f*(v,ras);*

Figure 8.3: An algorithm to calculate the canary failures for a given set of SRAM and canary specifications.

Algorithm3: Calculates the canary failure threshold sets and the final set of reverse assist settings (*RAS*) across process corners for V_{MIN} tracking.
Input: Canary design specification and canary V_{DD}-*RAS* failure tables and SRAM V_{MIN}s across design knobs.
Output: For each temperature, frequency, operations, reverse assist, and process conditions
1) Gives the canary failure threshold sets per process corner and the common canary failure threshold set across processes.
2) Checks if the V_{MIN} tracking is possible and gives the final RAS_F settings for V_{MIN} tracking across process corners.

N←Positive integers 1, 2 … etc., p← all processes, v← all voltages, t← all temperatures, f← all frequencies, ops← all write, read operations, ra← all reverse assists, ras← all *RAS* values, C_{VDD} is canary V_{DD}, ΔV_{DD} is the LDO granularity. SV_{MIN} is the SRAM V_{MIN}, RAS_F← the set of final *RAS* values.

$N_f(C_{VDD},ras) \leftarrow Canary\_Failures(p,t,f,ops,ra)$; $RAS_F \leftarrow \{\}$;
if $N*\Delta V_{DD} >= C_{VDD} - SV_{MIN}(p,t,f,ops,ast)$
    $C_{VDD_H} \leftarrow C_{VDD}$;
elsif $\Delta V_{DD} <= SV_{MIN}(p,t,f,ops,ast) - C_{VDD}$
    $C_{VDD_L} \leftarrow C_{VDD}$;
end if
$\mathbf{F}(p,ras) \leftarrow \{ N_f(C_{VDD_H},ras)+1 \text{ to } N_f(C_{VDD_L},ras) \}$;
$F_{th_p} = \mathbf{F}(p,ras)$;
$F_{th_k} = \cap\mathbf{F}(p,ras)$;
if $(|F_{th_k}| >= 1 \text{ or } |F_{th_p}| >= 1)$
    $S_{VMIN}\_Track(p,t,f,ops,ra)=1$; $RAS_F \leftarrow RAS_F \cup \{ras\}$;
else
    $S_{VMIN}\_Track(p,t,f,ops,ra)=0$;
end if

Figure 8.4: An algorithm to calculate the SV_{MIN} tracking condition for a given set of SRAM and canary specification.

SV_{MIN} (CV_{DD_H}) and just lesser than the corresponding SV_{MIN} (CV_{DD_L}) across temperature, frequency, operation, RA type, and processes. Thereafter, it computes the $F_{th_p} = F(p, RAS_N)$ as the set of valid F_{th} conditions across processes to track SV_{MIN} shown in equation (8.7). The Algorithm3 finally computes the common F_{th} values across process variation as the F_{th_k}, which is the intersection of the $F(p, RAS)$ across given processes. Then the algorithm, checks if the cardinality of the F_{th_k} ($|F_{th_k}|$) is at least one or greater, and asserts $SV_{MIN}\_Track = 1$ for canary-based SV_{MIN} tracking as a possibility. Otherwise, $SV_{MIN}\_Track = 0$ is assigned. Based on SV_{MIN}_Track being one or zero, it writes the canary design report in a human-readable format. The user can fine tune or change the design decisions for the RA-based canary design by changing the scope of RAS values or re-assigning a different RA type for further analysis.

8.0.7 RADA Tool-Flow

To check the necessary conditions in equations (8.1)-(8.11), we develop a tool-flow that supports the RA-based canary design and analysis for the SV_{MIN} tracking for the guard-band lowering paradigm. Figure 8.5 shows the proposed tool-flow. There are five main parts in this flow, which include input specifications as i) SRAM design specification, ii) canary design specification, iii) technology specification, iv) simulation specification, v) the RADA engine and the output as the vi) canary design report. The inputs to the SRAM design specification are SRAM bitcell netlist, size of the SRAM, the read/write operations need to check, the peripheral assist options, the list of process corners, the list of temperatures, the list of operating supply voltages (V_{DD}s), and the list of operating clock frequencies. The inputs to the canary design specification are the canary size, reverse assist (RA) types, the upper and lower limit of the RA settings (RAS), and the number of RAS values. The specification of technology for the tool-flow includes the paths of the technology library files. Currently, this tool-flow only supports Synopsys HSPICE simulator, and the simulation specification requires inputs such as options and parameters for HSPICE to run. With all these input specifications,

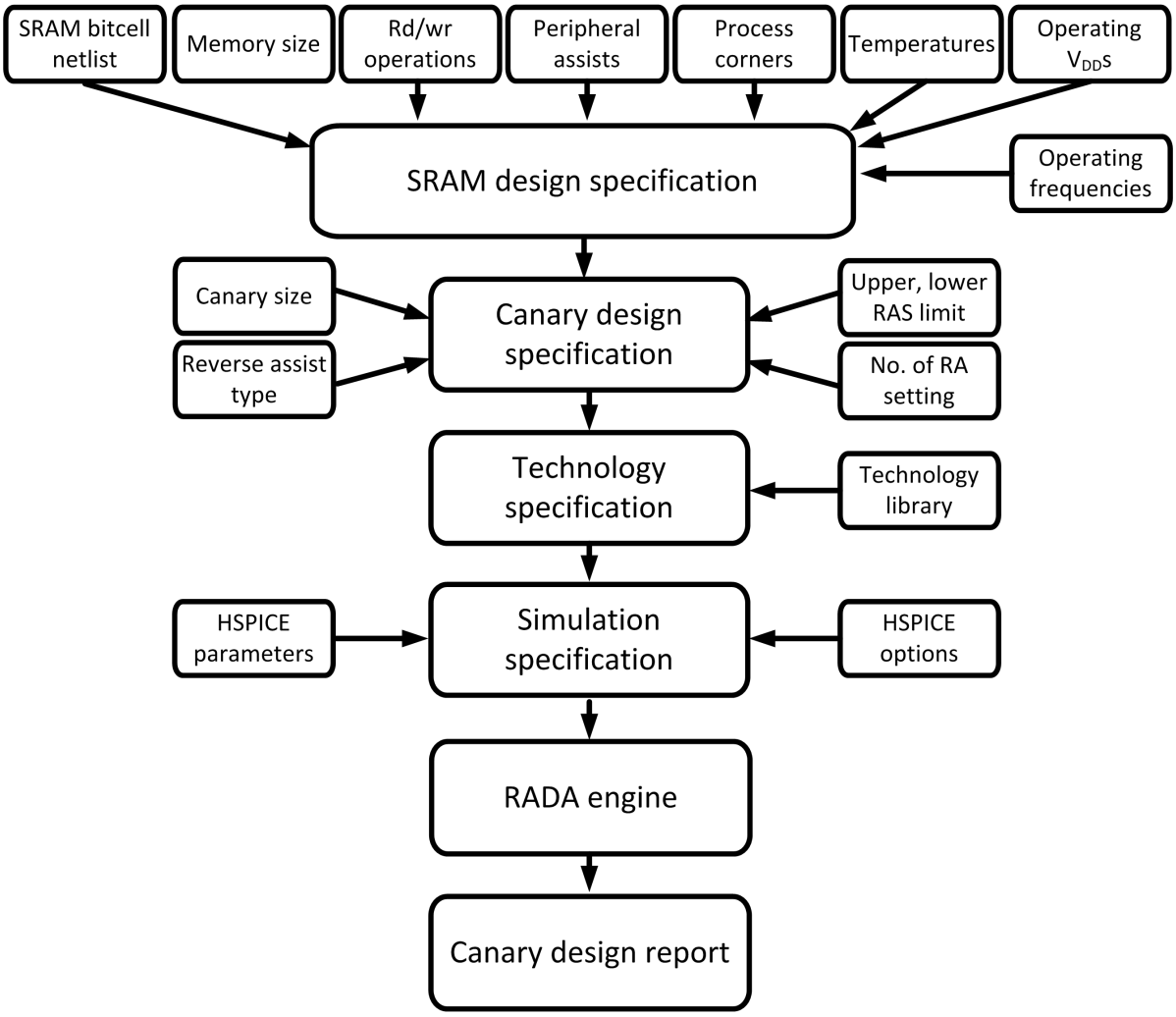


Figure 8.5: Block diagram of the tool-flow for RA-based canary design and analysis.

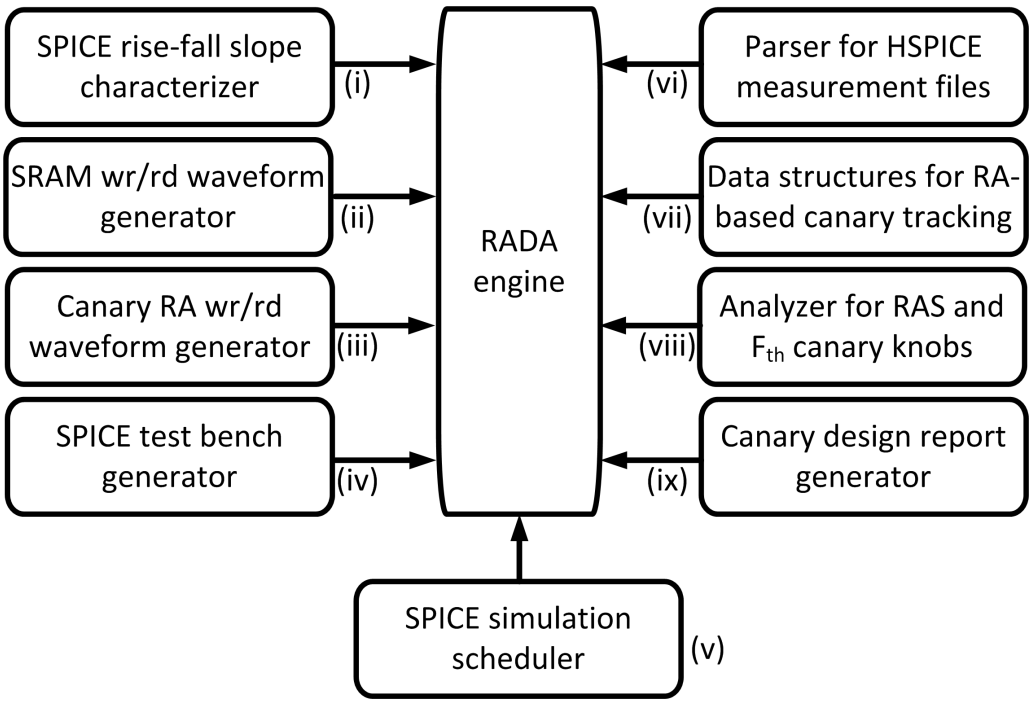


Figure 8.6: Block diagram of the nine internal components of the RADA engine.

the RADA engine runs HSPICE and analyzes the equation (8.1)-(8.11) over the measured data that shows if the RAS values for the corresponding RA-based canary support SV_{MIN} tracking for guard-band lowering. Figure 8.6 shows the nine components of the RADA engine. The first four components provide the inputs to run necessary simulations. The component v) manages the simulation runs as specified by the user. The simulation manager reruns simulations that do not complete within a specified duration. After the simulations complete, vi) a parser parses for HSPICE measurement files, and vii) a data structure for RA-based canary tracking is populated. The RADA identifies the user-specified measurement parameter and analyzes the RAS and F_{th} canary knobs using the component viii). After the completion of RA-based canary design and analysis, the component ix) generates the analysis report. This report is vital to the canary designer, as it shows the analysis of the RAS & F_{th} knobs for the corresponding RA type. The report provides for each process the list of F_{th} values for each RAS value that will be able to track the SV_{MIN} using the canary sensors. The canary designer can also generate a report on processes to check if there exist any F_{th} values that

may satisfy the SV_{MIN} tracking criteria specified in equation (8.11).

8.0.8 Implementation, Experiments, and Results

The RADA algorithms in the tool flow are written in Perl in several modules with functions distributed across different modules for each algorithm. In the Perl implementation of the algorithms, we code the portion of the SV_{MIN} calculation in Algorithm1 inside the Algorithm3 for the ease of implementation. This change hardly influences the overall runtime of the Algorithm1, as the runtime of the SV_{MIN} calculation is negligible compared to the HSPICE runtime in Algorithm1. As high-density 6T bitcell is mostly write limited in bulk, FDSOI, and FinFET technologies, we use 45nm bulk, 32nm FDSOI, and14nm FinFET technologies to analyze the canary design knobs for write SV_{MIN} tracking using RA-based canary sensors. To run the flow and HSPICE, we use an Intel(R) Xeon(R) CPU E5-2430 machine running at 2.20GHz with 24 cores. Using the RADA tool-flow, we initially compute the SV_{MIN} and N_f for all three technologies across design knobs and compute the F_{th_p} and F_{th_k} sets and finally the SV_{MIN}_Track for analysis. For SV_{MIN} simulations we use five process corners (TT, FF, SS, SF, and FS), twenty-nine voltages (1.0V-0.3V with 25mV steps), one temperature (27C), and six frequencies (5Ghz, 4GHz, 3GHz, 2GHz, 1GHz, and 100MHz). Thus, we run a total of 870 MC simulations with 10K MC samples for 32nm and 14nm technologies. For the 45nm experiments, we only use a different set of frequencies (2.5GHz, 2GHz, 1.5GHz, 1GHz, 500MHz, and 1MHz) with other inputs being the same. For the CV_{MIN} simulations, we use wordline slope degradation (WLSD) [91] type reverse assists with 8 RAS settings along with 1K MC samples per simulation and run a total of 3600 simulations. For the CV_{MIN} simulations, we use the process corners mentioned above, fifteen voltages (1.0V-0.3V with 50mV steps (which is the assumed LDO granularity)), the same temperature of 27C, and the same set of frequencies for 14nm and 32nm runs.

After analyzing the data from the RADA report, we constructed the similar plots shown in [91] for SV_{MIN} tracking across the technologies. Figure 8.7 shows the optimal SV_{MIN}

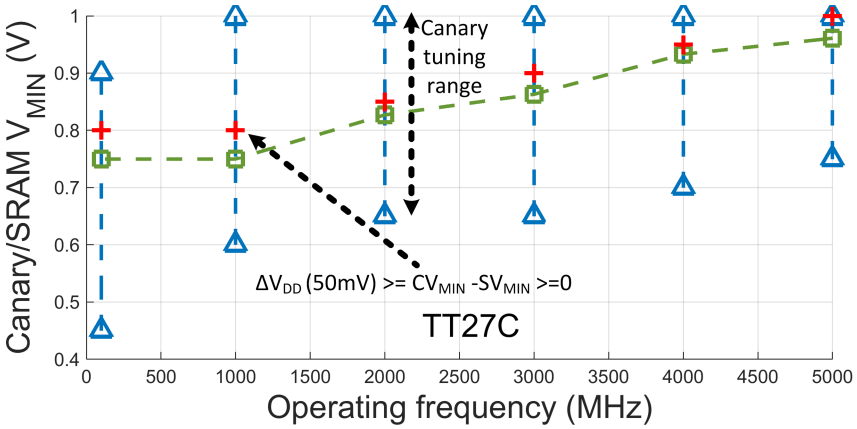


Figure 8.7: Simulated RA-based SRAM V_{MIN} (SV_{MIN}) tracking optimally using canary sensors and canary tuning range covering the SV_{MIN} at 45nm bulk technology at TT_27C corner.

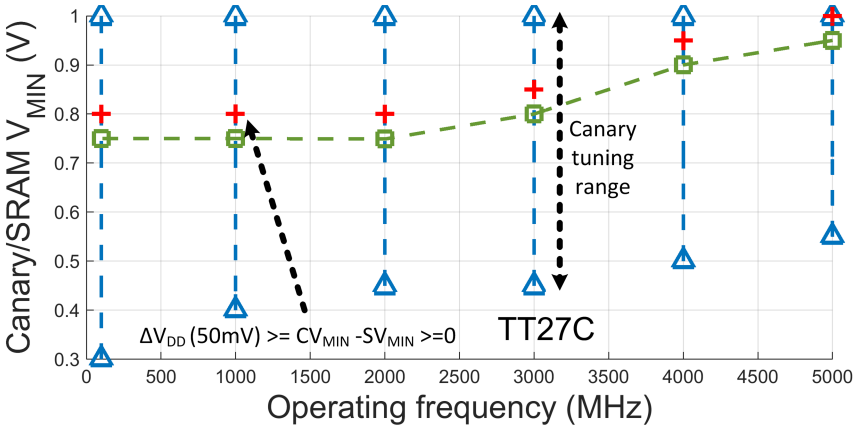


Figure 8.8: Simulated RA-based SRAM V_{MIN} (SV_{MIN}) tracking optimally using canary sensors and canary tuning range covering the SV_{MIN} at 32nm FDSOI technology at TT_27C corner.

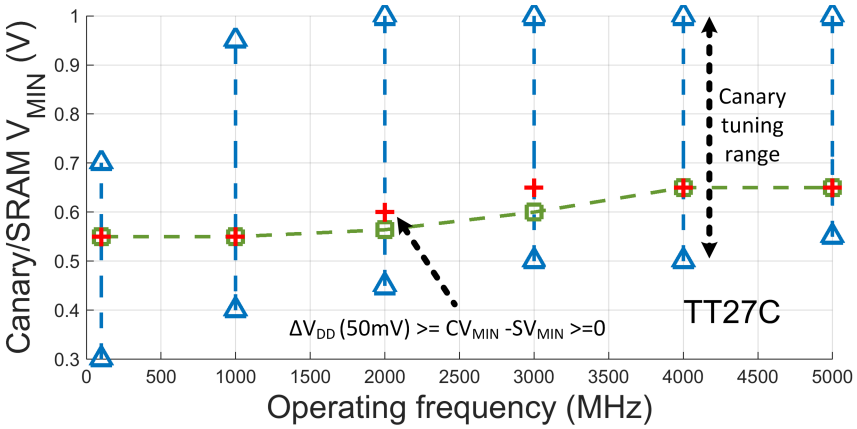


Figure 8.9: Simulated RA-based SV_{MIN} tracking optimally within ΔV_{DD} (50mV) using canary sensors and canary tuning range covering the SV_{MIN} at 14nm FinFET technology at the TT_27C corner.

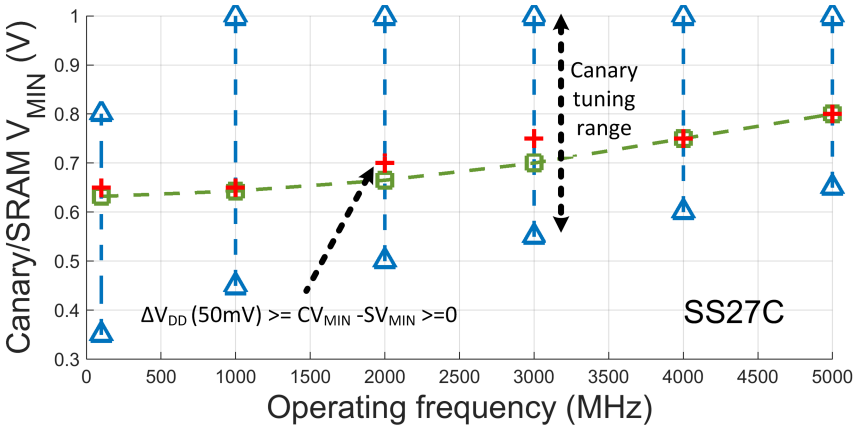


Figure 8.10: Simulated RA-based SV_{MIN} tracking optimally within ΔV_{DD} (50mV) using canary sensors and canary tuning range covering the SV_{MIN} at 14nm FinFET technology at the SS_27C corner.

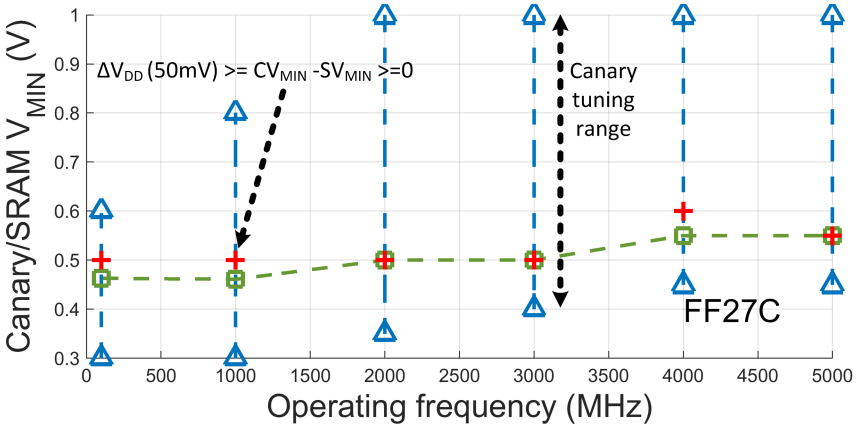


Figure 8.11: Simulated RA-based SV_{MIN} tracking optimally within ΔV_{DD} (50mV) using canary sensors and canary tuning range covering the SV_{MIN} at 14nm FinFET technology at the FF_27C corner.

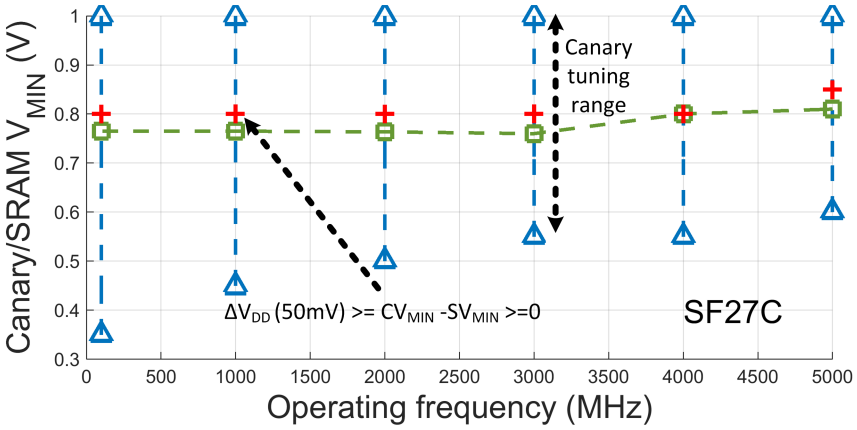


Figure 8.12: Simulated RA-based SV_{MIN} tracking optimally within ΔV_{DD} (50mV) using canary sensors and canary tuning range covering the SV_{MIN} at 14nm FinFET technology at the SF_27C corner.

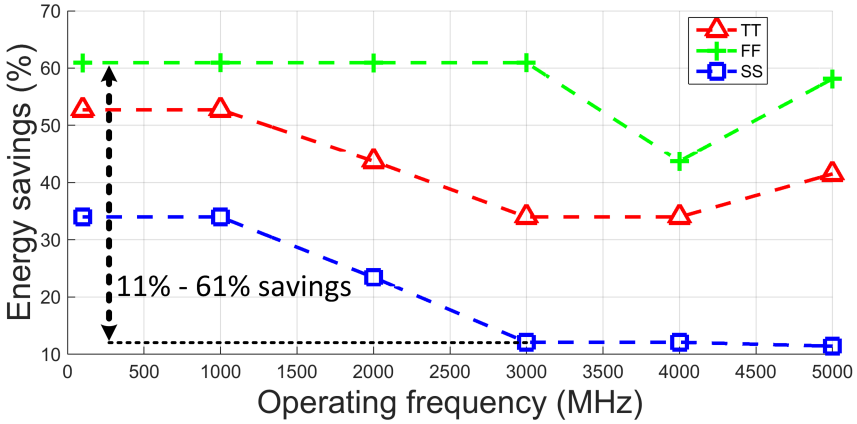


Figure 8.13: Simulated energy saving using SV_{MIN} tracking compared to the worst-case SF corners at 27C in 14nm FinFET technology.

tracking in 45nm bulk technology. Here the canary tuning range specifies the margin availability in RAS values so that across die-to-die process variation RADA enables canary-based SV_{MIN} tracking. Figure 8.7 shows that for all the frequencies the canaries can be tuned optimally at the TT_27C corner such that $\Delta V_{DD} >= CV_{MIN} - SV_{MIN} >= 0$, which satisfies the equation (8.2). Similarly, Figure 8.8 shows the write SV_{MIN} tracking in an FDSOI technology in the TT_27C condition. Figure 8.9, Figure 8.10, Figure 8.11 and Figure 8.12 show the optimal write SV_{MIN} tracking in a FinFET technology across TT, FF, SS, and SF corners at 27C temperature. The simulated energy savings compared to the worst-case write SF corner at 27C temperature using SV_{MIN} tracking saves 11.4% to 60.9% (Figure 8.13) across corners.

Table 8.1: Runtime (minutes) for Perl implementation of RADA algorithms.

Technology	**Algorithm 1 runtime (m)**	**Algorithm 2 runtime (m)**	**Algorithm 3 runtime (m)**	**RADA total runtime (m)**
14nm	534.1	288.61	0.016	822.73
32nm	665.63	901.66	0	1574.08
45nm	198.15	192.55	0	390.69

8.1 shows the runtime of each algorithm in RADA for the estimated process showing for 45nm bulk, 32nm FDSOI, and 14nm FinFET technology. It is impractical to design explore manually across the design knobs, run simulations and parse data to analyze this vast canary design space. This RADA tool-flow can improve the manual RA-based canary design exploration time from months or weeks to a few days. Thus, RADA reduces an SRAM designer's burden for design exploration of RA-based canary sensors across design knobs for the arbitrary lowering of SV_{MIN} guard-bands.

8.0.9 Conclusions

The reverse assist-based canary design is an attractive solution for reducing the SRAM V_{MIN} guard-band arbitrarily by tracking the SRAM V_{MIN} using canary sensors. This chapter derives the necessary conditions for the first time to track SRAM V_{MIN} across canary design

knobs. This work also proposes the RADA tool-flow and the corresponding algorithms that bridge the gap between the canary design knobs and analyzes solutions for a valid set of *RAS* values and F_{th} conditions to optimally track SRAM V_{MIN}. By supporting the automated design exploration and analysis of reverse assist-based canaries for guard-band minimization of SRAM V_{MIN}, RADA minimizes not only canary and overall SRAM design time but also enables wide-V_{DD} range DVS for ULP IoE applications. The RADA algorithm and tool-flow are independent of the process technology and support conventional bulk, FDSOI, and FinFET technologies.

8.0.10 Acknowledgements

This work was funded in part by NVIDIA through the DARPA PERFECT program.

Chapter 9

Conclusions

Portability and form factor of battery-operated IoT devices restrict the use of larger batteries, making them highly energy-constrained. On the other hand, harvested energy IoT devices would require an ultra-low power (ULP) system on chip (SoC) that has lower power consumption below the harvested energy budget. Moreover, there is a growing pressure of supporting a multitude of IoT applications in the existing ULP devices that requires wide-range voltage scaling to support variable work-loads from time to time. Microprocessors, digital accelerators, analog transceivers, radios, etc. used in ULP SoCs require low power SRAMs for the register file, cache memory, and FIFO buffer designs. These ULP SRAMs must be flexible to operate at lower supply voltages for energy savings. However, the smallest area low-cost 6T SRAM suffers from V_{MIN} scaling challenges in bulk, FDSOI, and FinFET technologies. As the most promising FinFET technology progresses towards the 7nm and eventually 5nm production node, the SRAM design requires novel combinations of peripheral assist for V_{MIN} improvement. Moreover, a designing-for-the-worst-case methodology for SRAMs adds additional energy and area penalty for the typical, and the best case dies. Thus, there is a scope to improve SRAM design beyond the traditional design-for-the-worst-case methodology for the future SRAMs, that would require energy saving canary sensor SRAMs to track V_{MIN} and apply peripheral assists, as measures. This thesis bestows novel

findings in the field of SRAM, such as bitcell topologies, array, peripheral circuits architecture, combinations of peripheral assists methods, and canary SRAM theory and architecture for ULP IoT applications. Thus, this thesis contributes additional knowledge and improves the state-of-the-art SRAM design techniques that would allow the design of low V_{MIN} and energy efficient SRAMs for a more extended lifetime of battery-operated or energy-harvested IoT systems, such as wearable health monitors, smart-watches, augmented reality goggles, etc.

9.1 Summary of Contributions

This section summarizes the contributions of this thesis in the domain of SRAM design as follows. The second chapter contributes in the field of alternative sub-threshold SRAM bitcell topologies, arrays, and peripheral circuits architecture. The third chapter adds new knowledge in the domain of combination of peripheral assists for 6T SRAM V_{MIN} improvement. The fourth chapter improves the state-of-the-art SRAM design with a novel theory of canary SRAM for SRAM write V_{MIN} tracking. The fifth chapter shows the proof of concept of canary-based V_{MIN} using the characterization of a canary SRAM in a commercial 130nm bulk technology. The sixth chapter classifies the reverse assists and their properties and shows tradeoffs. The seventh chapter demonstrates a closed loop canary architecture for SRAM V_{MIN} tracking, and the eighth chapter documents the algorithm of the automation of reverse assist design for canary SRAMs.

Alternative Sub-threshold SRAM Bitcell topologies, Arrays, and Peripheral Circuits Architecture

This thesis introduces a novel 9T half-select-free bitcell [36] for ULP battery operated, or energy harvested IoT applications and compares it with the state-of-the-art alternative bitcells. Across voltages of 0.25-0.5 V, our 9T bitcell work [36] has the lowest read energy among the state-of-the-art ([26] [27] [28] [29] [36]) bitcells, including the conventional 6T

bitcell. It has the lowest write energy among the bitcells across the voltages 0.35-0.5V and second lowest leakage current in the 0.1-0.5 V range. Though our bitcell has lower numbers in energy and leakage current in sub-threshold voltages, it suffers from a timing penalty. This work has demonstrated the lowest minimum energy point (MEP) across F_{rdwr} = 0.5-0.9 for 32 KB SRAMs. Our bitcell also provides the lowest MEP variation for 32 KB SRAMs across various rows per bank (RPB), ranging from RPB = 4-64;

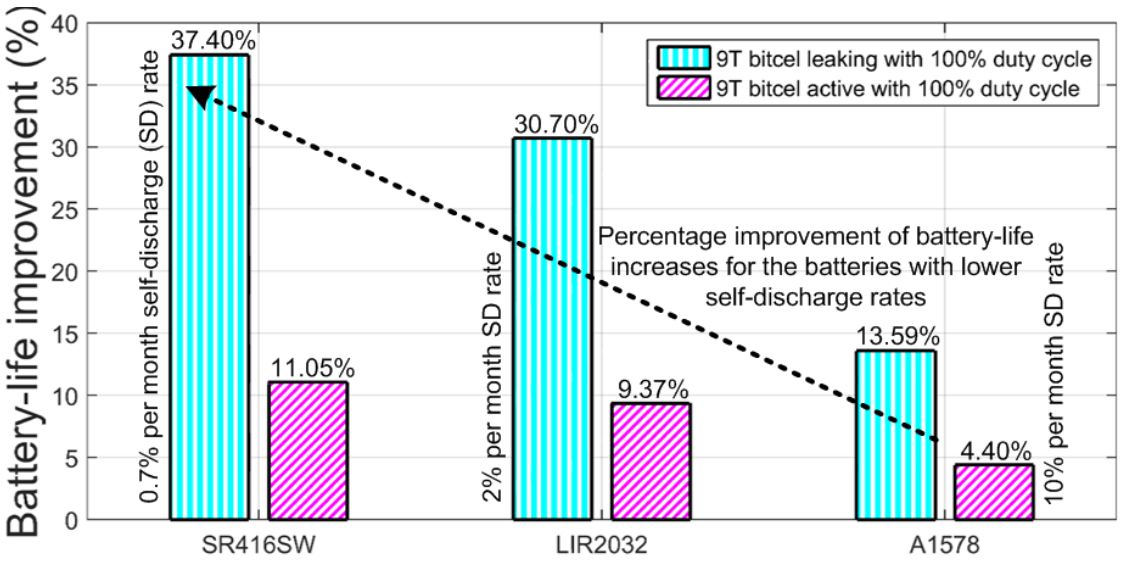


Figure 9.1: Estimated single charge battery-life improvement of SR416SW, LIR2032, and A1578 batteries using 9T half-select free bitcell in BSN SoC compared to the conventional 8T SRAM bitcell.

In the estimation of system level battery-life savings, our 9T bitcell improves battery-life by 11.05% and 37.40% for 100% active and 100% leaking cases (Figure 9.1), respectively, using the SR416SW battery compared to the conventional 8T bitcell used in the BSN revision 1 SoC. The BSN revision 1 had $19\mu W$ of power consumption of which the instruction memory consumed 55.4% (36% dynamic and 64% leakage) and 44.6% by other digital components. We assume that the instruction memory uses our 9T SRAM bitcell, which is active 100% of the time and the power source is an A1578 (0.76Wh) battery. Thus, we estimate the battery-life improvement of 4.40% from 1.10 years to 1.154 years (about half months of battery-life improvement), as shown in Figure 9.1 and Figure 9.2. On the other hand, if we

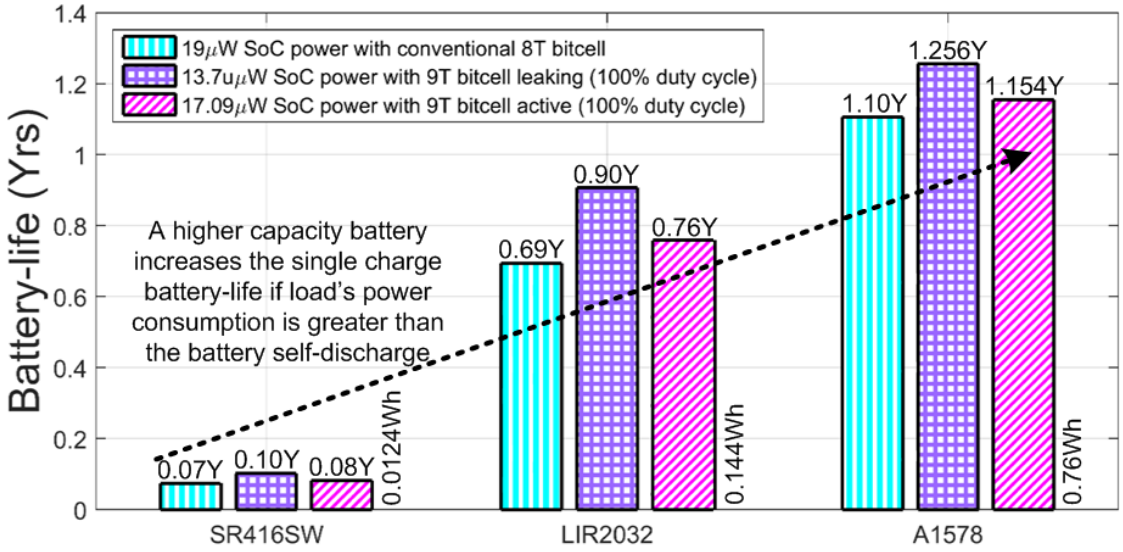


Figure 9.2: Estimated single charge battery-life time of SR416SW, LIR2032, and A1578 batteries using 9T half-select-free bitcell in BSN SoC compared to the conventional 8T SRAM bitcell.

assume that the instruction memory is leaking 100% of the time, the battery-life improvement becomes 13.59% from 1.10 years to 1.256 years (Figure 9.1 and Figure 9.2), which is a bit more than one and half months of battery-life improvement. The self-discharge rate used in this calculation for the A1578 battery is 10% per month. Using LIR2032 battery the corresponding battery-life savings numbers are 9.36% and 30.70%, as shown in Figure 9.1. The self-discharge rates used for LIR2032 calculations are based on Table 1.3. On the other hand, using a non-rechargeable SR416SW battery, the corresponding battery-life improvements are 11.05% and 37.40%, and the battery replacement time increases to 30 days and 37.13 days from 27.02 days, respectively.

Using the low energy read (LER) scheme for SRAM periphery, the read energy improvement of the BSN revision 2 SRAM, compared to the revision 1 read energy, is 6X at SS_0.5V_27C PVT corner, and the best improvement is 7.4X at FS_0.5V_27C PVT corner. The worst-case usual read energy in revision 2 is 45% more than the revision 1 read energy in SS_0.5V_27C PVT corner. Except for the TT and FS corners at the same supply voltage and temperature, the revision 2 usual read energy is always higher than the older design read energy. On the

other hand, the write-after-read (WAR) energy improvements compared to the cumulative write and read energy in BSN revision 1 SRAM design at 0.5V_27C are 2.5X, 2X, and 1.67X at FS, FF, and TT process respectively. However, for the SS and SF process, the revision 2 WAR energy is 20% and 25% more than that of the cumulative write and read energy of the revision 1 SRAM, at the same supply voltage and temperature. With our method, the revision 2 SRAM layout area is increased by 7%, compared to the revision 1 layout and one can minimize it by optimizing the floorplan and individual block layouts. The worst-case standby leakage current penalty is 3% at the FF_0.5V_27C PVT, and the best-case standby leakage current is 17% less than the revision 1 SRAM design.

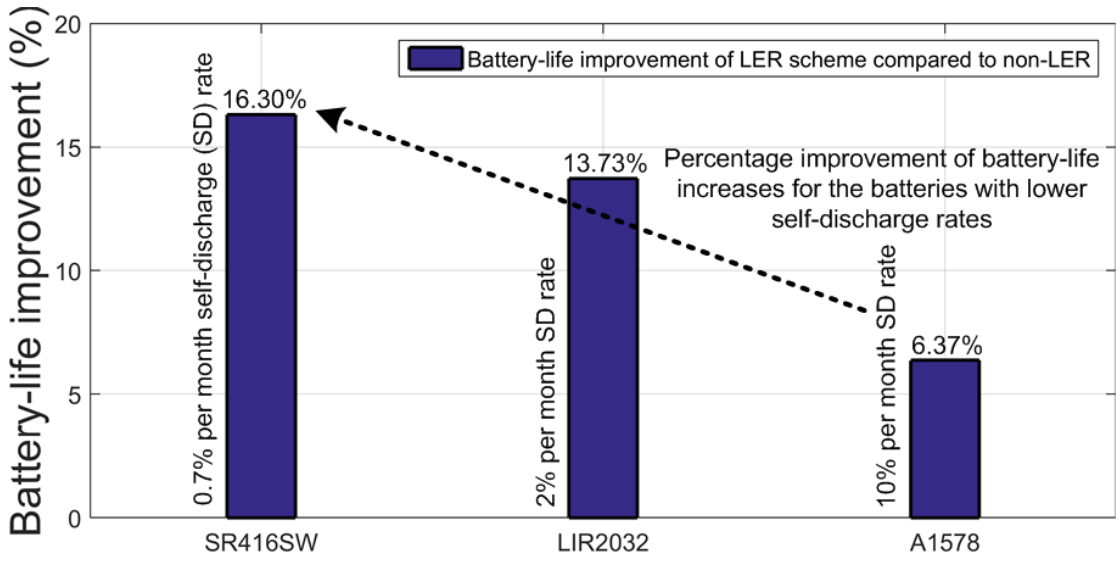


Figure 9.3: Estimated single charge battery-life improvement of SR416SW, LIR2032, and A1578 batteries using low energy read (LER) periphery scheme in BSN SoC compared to non-LER scheme.

The BSN revision 1 has $19\mu W$ of power consumption of which the instruction memory consumes 55.4% (36% dynamic and 64% leakage) and 44.6% by other digital components. We assume that the instruction memory uses our LER scheme for sequential reads, which is active 100% of the time and the power source is an A1578 (0.76Wh) battery. Thus, we estimate the battery-life improvement of 6.37% from 1.10 years to 1.17 years (about a month of battery-life improvement), as shown in Figure 9.3. These projections assume a self-discharge of 10%

per month for the A1578 Lithium-ion Polymer battery. Using a rechargeable Lithium-ion LIR2032 battery, the battery-life improves by 13.73% (Figure 9.3) from 0.69 years to 0.78 years. We use the self-discharge rates from Table 1.3 for LIR2032 battery-life calculations. On the other hand, using a non-rechargeable SR416SW battery, the battery-life improves by 16.30% (Figure 9.3) and battery replacement time increases to 31.4 days from 27.02 days. The sub-tasks and outcomes associated with this researches are listed below.

- An alternative sub-threshold 9T bitcell topology that has 2.05X lower read energy, 1.12X lower write energy, and 1.28X lower leakage current compared to the conventional 8T bitcell.

- Comparison of the state-of-the-art sub-threshold bitcells' write and read energy and leakage current in a bitline interleaving scenario.

- Comparison of the state-of-the-art sub-threshold bitcells in a bitline interleaving scenario for minimum energy point across various SRAM design knobs, such as the fraction of read and write, number of bitcell rows per bank, word-width, number of words per row or the column mux factor, and capacity.

- A low-energy read peripheral architecture improving energy consumption in a read-modify-write operation for half-select avoidance in sub-threshold SRAMs, which lowers active read energy in a read dominated cache.

Combination of Peripheral Assists for 6T SRAM V_{MIN} Improvement

The 14nm 6T high-density (HD) FinFET SRAM design sees challenges from the variations of the process, temperature, frequency, and other design knobs, and has more than 300mV of V_{MIN} across design parameters. Without the bitline interleaving scenario, the traditional write and read solo assists can improve the V_{MIN} across design knobs. However, the worst-case V_{MIN} in 6T HD FinFET SRAM is above the nominal supply voltage, and none of the single peripheral assists can improve it in a bitline interleaving scenario. Only selected combinations

such as V_{DD} boosting (VDB) with wordline boosting (WLB), wordline underdrive (WLU) with V_{DD} collapse (VDU), etc. are suitable. Using write and read combined peripheral assists, the effecting static V_{MIN} have minima across dual assist percentages. We show that for non-bitline-interleaving scenarios the 10%_10% combination of WLB + negative bitline (NBL) beats all other solo and dual combinations for write V_{MIN} improvement.

Figure 3.19a shows the plot for the static write V_{MIN} across dual assist combinations, with the total assist percentage being constant at 20% at the SF_-40C corner. The plot shows the better combinations to improve the static write V_{MIN} (NBL + VDU, NBL + V_{SS} raising (VSR), WLB + NBL, WLB + VSR, and WLB + VDU). On the other hand, Figure 3.19b shows the plot for static read V_{MIN} vs. dual assist combinations, with the total assist percentage being constant at 20% at the worst-case read V_{MIN} corner of FS_85C. The plot shows the better combinations for improving the static read-stability V_{MIN} such as WLU + VDB etc. Finally, Figure 3.20 shows all effective combinations of write and read assist that allow us to lower the worst-case V_{MIN}. Noticeably, the combinations NBL + VDB and VDU + VDB achieve the lowest V_{MIN} using 14%_6% assist combinations.

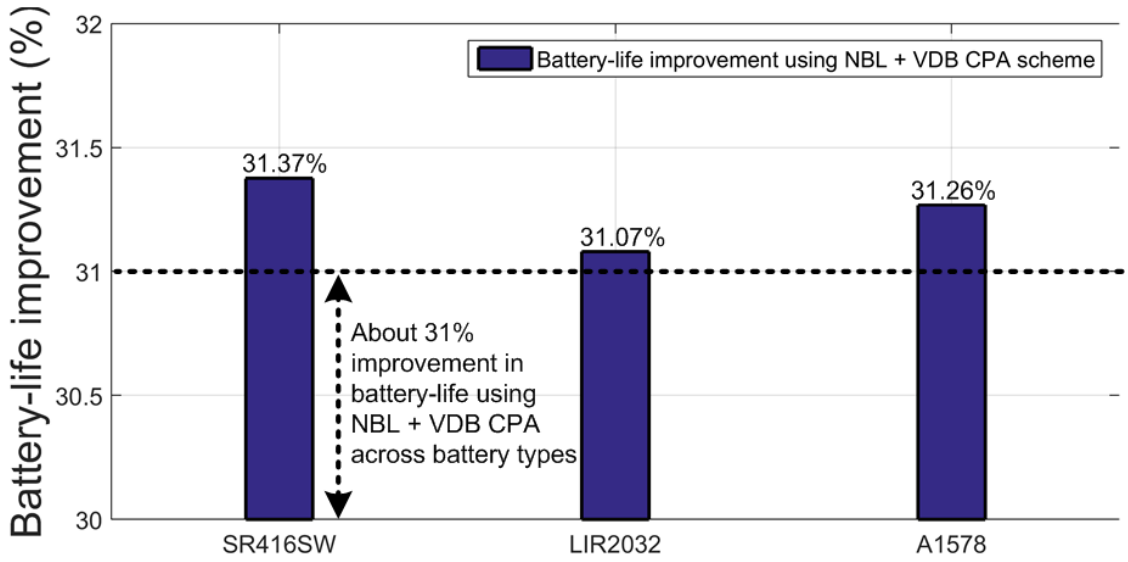


Figure 9.4: Simulated single charge battery-life improvement of SR416SW, LIR2032, and A1578 batteries using NBL + VDB combined peripheral assist (CPA) scheme in 16nm FinFET technology compared to the worst-case V_{MIN}.

For this chapter, we estimate battery-life with the assumption that SRAM consumes 40% energy while the logic core consumes 60% energy and the IoT system uses an A1578 (0.76Wh) battery. We also assume that the system average power consumption is the same as Apple iWatch average power consumption of 42.2mW. Using CPA assuming 100% duty cycle the battery-life improves by 31.26% (Figure 9.4) and the corresponding battery replacement time increases from 1.025 years to 1.345 years. These projections assume a 10% self-discharge rate for the A1578 battery. Using a Lithium-ion LIR2032 (0.144Wh) battery, the corresponding battery-life improvement is 31.07% (Figure 9.4) and the battery replacement time improves from 2.35 months to 3.08 months. Here we assume a maximum of 500 recharge cycles. The corresponding battery-life improvement for a Silver Oxide SR416SW (0.0124Wh) battery is 31.37%, as shown in Figure 9.4. The sub-tasks and outcomes associated with this research are listed below.

- Demonstrated challenges in a 14nm 6T HD FinFET SRAM's static write, read and overall V_{MIN} across design knobs, such as the process and temperature using the metrics of write static noise margin and read static noise margin.
- Exploration of challenges in a 14nm 6T HD FinFET SRAM's dynamic write, read and overall V_{MIN} across design knobs such as process, temperature, and operating frequency.
- Design exploration of the static write-ability and read-stability noise margin metrics of a 14nm 6T HD FinFET bitcell across single peripheral assists for a fixed 20% assist percentage.
- Design exploration of the static write-ability and read-stability noise margin metrics of a 14nm 6T HD FinFET bitcell across single peripheral assists for a variable assist percentage up to 20%.
- Exploration of static write-ability and read-stability noise margin metrics of a 14nm 6T HD FinFET bitcell across dual combinations of peripheral assists for a fixed 20% assist percentage with 10% each assist.

- Exploration of static write-ability and read-stability noise margin metrics of a 14nm 6T HD FinFET bitcell across dual combinations of peripheral assist for variable dual assist percentages, but overall a total assist percentage of 20%.

- Design exploration of the static write-ability, read-stability, and overall static V_{MIN} metrics of a commercial 14nm 6T HD FinFET bitcell across single peripheral assists for an assist percentage up to 20%.

- Design exploration of the static write-ability, read-stability, and overall static V_{MIN} metrics of a commercial 14nm 6T HD FinFET bitcell across dual peripheral assists for variable dual assist percentages. We use a total assist percentage of 20%, and show a successful combination such as 14% NBL + 6% VDB, etc. for improving the worst-case static V_{MIN}.

- Findings on the V_{MIN} improvement of dynamic write-ability and readability V_{MIN} across 100 chip simulations with 10kb SRAM capacity using dual combinations of read-write assists beating single and other combinations.

- Design and testing of a 256kb SRAM testchip with two write assists (wordline boosting and negative bitline) and a single read assist (V_{DD} boosting) achieving a V_{MIN} improvement of 240mV. The combination beats all other assist combinations to have a 90^{th} percentile V_{MIN} of 0.47V operating at sub-threshold supplies that save over 300X on active power.

Canary SRAM Theory for SRAM Write V_{MIN} Tracking

The theory of canary SRAM [81] in this thesis shows promise to track the core SRAM V_{MIN} and reduce energy consumption across process variation. As per our simulation results, one can use the canary SRAMs to track the write V_{MIN} of the SRAM bits with a specified statistical confidence. The normalized write energy corresponding to the core SRAM V_{MIN} is shown in Figure 4.18. At the TT_85C corner, we can operate SRAMs with 36% lower write energy cost than that of the worst-case V_{MIN} at the SF_85C corner, which would set

the guard-band. We achieve the least energy savings at the SS_85C corner of 30.7%. The maximum energy savings happen at the FS_85C corner (Figure 4.18), which is 51.5% lower than the worst-case. Furthermore, at the FF_85C corner, the energy savings can reach up to 42.2% compared to the worst-case energy at the SF_85C corner.

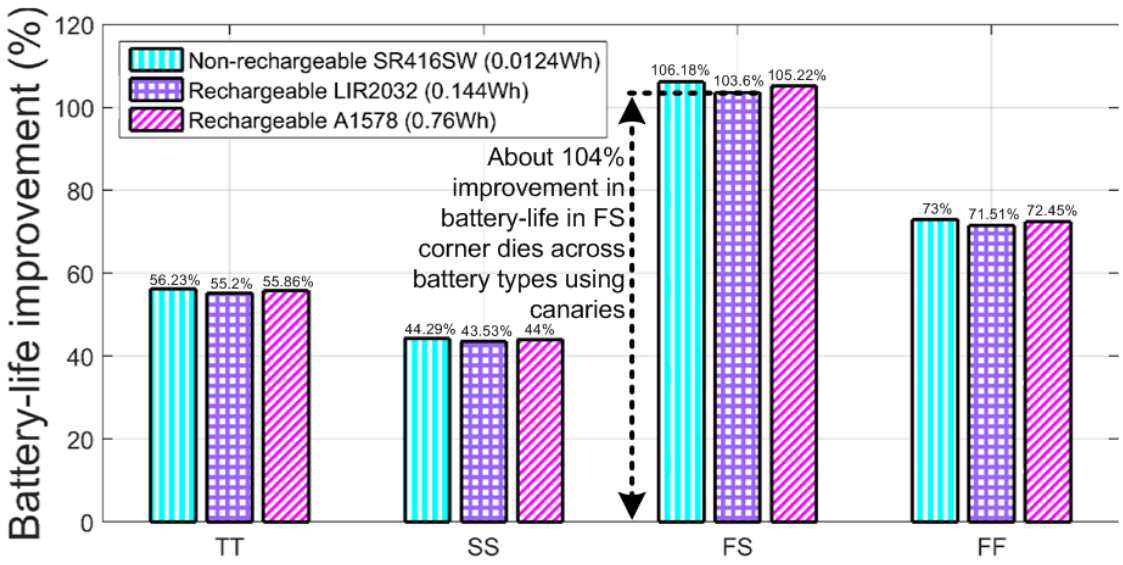


Figure 9.5: Simulated single charge battery-life improvement of SR416SW, LIR2032, and A1578 batteries using canary SRAM scheme across corners compared to the worst-case SF corner in a commercial 28nm bulk technology.

This work assumes that the SRAM consumes 40% energy while the logic core consumes 60% energy. We further assume that the IoT system uses an A1578 (0.76Wh) battery. The estimated energy savings compared to the SF corner in TT, SS, SF, FS, and FF corners are 36%, 30.7%, 0%, 51.5%, and 42.2%, respectively. Thus, the relative energy consumptions are 64%, 69.3%, 100%, 48.5%, and 57.8%, respectively. We assume a 100pJ of average SRAM energy consumption and 100% duty cycle for battery-life estimation. Thus, we compute the total system (SRAM and core) power consumption at 100MHz frequency to be 0.016W, 0.017325W, 0.025W, 0.012125W, and 0.01445W at the corresponding corners. The corresponding estimated battery-life improvements running the IoT SoC fabricated in different corners with respect the worst-case SF corner are 55.86%, 44%, 0%, 105.22%, and 72.45%, as shown in Figure 9.5 (SF corner data not shown). We further assume a 500

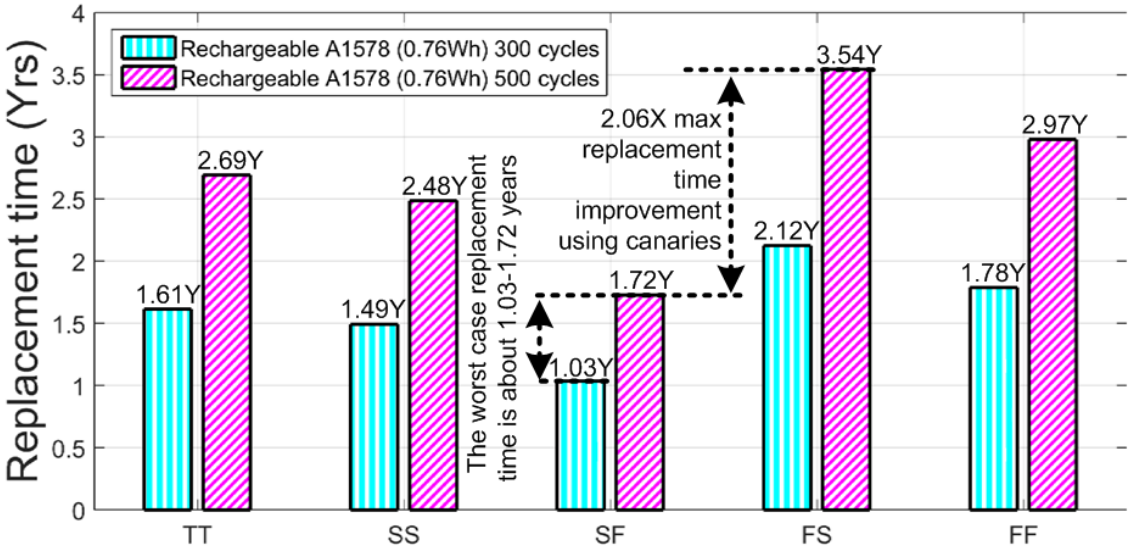


Figure 9.6: Estimated battery replacement time of A1578 batteries assuming 300 and 500 charge-discharge cycles using canary SRAM scheme across corners in a commercial 28nm bulk technology.

recharge cycles for an A1578 battery. Thus, the best case battery replacement time for the SoC fabricated in the FS corner will be 3.54 years. On the other hand, the worst-case battery replacement (assuming 300 charge-discharge cycles) time would be 1.72 years for the SF corner SoC with the SRAM, as shown in Figure 9.6. These projections assume a self-discharge of 10% per month for the A1578 Lithium-ion Polymer battery. The sub-tasks associated with this research are listed below.

- A theory of SRAM dynamic write V_{MIN} tracking using reverse assist-based canary sensor SRAMs that relates the input design knobs of canary and core SRAMs to the canary and core SRAM output metrics and allows us to save up to 51% of energy across process variation.

- Proposed a methodology for choosing a canary failure rate against an SRAM target failure rate based on specific yield requirements.

- Exploration of design tradeoffs of canary SRAM design for a target SRAM design specification.

- Proposed a novel reverse assist circuit implementation in a canary write driver based on bitline pulse-height-degradation scheme.

- Proposal of an embedded canary architecture for continuously tracking of SRAM V_{MIN} independent of the SRAM operation.

- Proposed a closed-loop algorithm to control the canary write and read operations to track SRAM V_{MIN} with user-defined controls.

- Power and area tradeoffs of bitline pulse-height-degradation type reverse assist circuit in the canary write driver across design knobs, such as SRAM capacity, the number of canary cells, the amount of reverse assist voltage, and failure threshold conditions.

Characterization of a Canary SRAM in a 130nm Bulk Technology

This chapter presents the first silicon results that canary write failures distinctly change with voltage, frequency, and temperature variations [87] for bitline reverse assist voltage (BLVRA). Hence, with canary SRAM and BLVRA, we can track voltage, frequency, and temperature variations, and thus track the SRAM write V_{MIN} (WV_{MIN}). However, the failure trends for wordline reverse assist voltage (WLVRA) are distinct to voltage and temperature changes, only. Thus, WLVRA is only useful for tracking WV_{MIN} across voltage and temperature variations. Finally, we show that we can tune the canary SRAM failure point using WL and BL type reverse assists before the SRAM bits start to fail, which is an essential condition for tracking the SRAM dynamic WV_{MIN}. The sub-tasks and outcomes associated with this research are listed below.

- Design and tapeout of a testchip for the characterization of canary SRAM properties for SRAM dynamic V_{MIN} tracking. The testchip includes components, such as an integrated 8kb core SRAM with an independent 512b canary SRAM, an SRAM built-in self-test, a canary built-in self-test, and scan-chains.

- Test setup creation, testing and data collection (in a team) of the testchip across various environmental design knobs. The design knobs used are supply voltage (0.7V, 0.8V, and 0.9V), frequency (25MHz, 50MHz, and 100MHz), and temperature (-40C, 27C, and 85C) points for wordline and bitline pulse-height-degradation type reverse assists.

- Analysis of the canary testchip data that reveals the first proof of concept of reverse assisted canary to be sensitive to voltage, frequency, and temperature variations that allows a canary SRAM to track the V_{MIN} of another core SRAM.

- Findings that show wordline pulse height degradation for a 130nm bulk 6T SRAM is insensitive to frequency changes from 25MHz to 100MHz; however, it is sensitive to all other design knobs such as voltage and temperature variations.

Classification of Reverse assists, Their Properties, and Tradeoffs

This chapter classifies and derives the tradeoffs of wordline and bitline pulse-shaping reverse assists (RAs) for canary-based SRAM V_{MIN} tracking. Selecting an RA depends on the V_{MIN} tracking specifications. If a user wants to track only small hops of the supply voltage from 0.8V to 0.7V (100mV) across processes, frequency, and temperature variations, it makes sense to select a wordline-pulse-slope-degradation (WLPSD) or bitline-pulse-slope-degradation (BLPSD) RA. As the WLPSD and BLPSD RAs have the highest sensitivity to detect changes in canary failures, tracking a smaller voltage variation would work the best. On the other hand, for large hops of supply voltages such as 0.8V to 0.5V (300mV), selecting a less sensitive wordline-pulse-height-degradation (WLPHD) or bitline-pulse-height-degradation (BLPHD) RA would be better to slowly change the number of canary failures to cover the entire 0.8V-0.5V supply range. Moreover, pulse-slope-degradation (PSD) type RAs have the largest RA range that is easy to design by choosing some set of slopes compared to selecting a pulse height or width in designing a pulse-width-degradation (PWD) or pulse-height-degradation (PHD) type RA. Lastly, WLPSD RA circuits are the best choice for lowest-cost energy and

area penalty and have the best figure-of-merit across RA slope variation. The sub-tasks associated with this research are listed below.

- Classification of pulse-shaping techniques such as pulse height, slope, and width for wordline and bitline type reverse assists (RAs) in canary sensor SRAM design.
- Definition and derivation of sensitivity metrics of comparison for wordline and bitline pulse-shaping RAs across the design knobs, such as the strength of the RA, supply voltage, frequency, and temperature variations.
- Comparison of wordline and bitline pulse-shaping RA-based canary design knob ranges in 6T HD FinFET SRAM write operation, such as RA ranges, supply voltage ranges, frequency ranges, and temperature ranges corresponding to the canary failure ranges.
- Comparison of wordline and bitline pulse-shaping RA-based canary sensitivity metrics in 6T HD FinFET SRAM write and read operations, across RA strengths, supply voltage, frequency, and temperature variations.
- Proposal of pulse-shaping wordline type RA circuits for canary sensor SRAM designs and their area, energy, and figure-of-merit tradeoffs.

A Closed-loop Canary Architecture for SRAM V_{MIN} Tracking

This chapter shows the proof of V_{MIN} lowering using combined peripheral assists (CPA) and in-situ canary-based SRAM V_{MIN} tracking in a commercial 130nm bulk technology [91]. Here, we compute the system level battery-life and replacement time savings at different SRAM V_{MIN}s with and without using CPA and in-situ canary sensor SRAM. We assume that both the SRAM and logic core shares a single supply rail and SRAM V_{MIN} limits the overall voltage scaling. Additionally, we assume that SRAM consumes 40% energy while the logic core consumes 60% energy from the supply rail or their power dissipation capacitance ratios are 2:3. We further assume that the IoT system uses an A1578 (0.76Wh) battery and

the average power consumption of the system is the same as Apple iWatch average power consumption of 42.2mW at the full-scale supply voltage of 1.2V. Thus, using voltage scaling without any peripheral assist applied to the SRAM, we can lower the system supply to 0.71V (Figure 7.1), which is the 90% SRAM V_{MIN}. The corresponding battery-life for a single charge improves from 17.96 hours to 7.82 days or about 946%, as shown in Figure 9.7 and Figure 9.8. Applying CPA lowers the 90% SRAM V_{MIN} to 0.47V and the corresponding battery-life for a single charge improves to 6.84 months from 17.96 hours, which is a 27329% improvement in battery-life (Figure 9.7 and Figure 9.8). Turning on in-situ canary-based V_{MIN} tracking along with CPA further lowers the SRAM V_{MIN} to 0.38V removing the margin guard-banding. The corresponding battery-life saving for single charge improves to 1.25 years from 17.96 hours, which is 60811% improvement in battery-life (Figure 9.7 and Figure 9.8). We assume 300 cycles of worst-case maximum usage for the A1578 battery. Thus, the corresponding battery replacement time without CPA (at 0.71V), with CPA only (at 0.47V), and with CPA and canaries (at 0.38V) would be 6.43 years, 10+ years, and 10+ years (Figure 9.9), respectively. Note that without CPA or canaries the battery replacement time is 0.61 years operating at the full-scale supply voltage of 1.2V, as shown in Figure 9.9. Due to the reason that overall maximum shelf-life or end-of-life of a battery is around 10 years, the battery replacement time for using solo CPA and CPA combined with canaries will be limited by the shelf-life or end-of-life of A1578, which could be a maximum of 10 years. Here we assume a 100% duty cycle for the calculations.

This work extends the 6T SRAM operating range by over 67% (from 1.2V-0.71V=0.49V to 1.2V-0.38V=0.82V, in sub-threshold) using three combined read/write assists and in-situ canary-based V_{MIN} tracking. The SRAM self-tunes to the V_{MIN} across frequency and temperature variations. This adaptive solution enables a range of Internet of Everything (IoE) applications and achieves up to 1444X active power reduction. The system level IoE battery replacement time could improve to 10+ years operating at 0.38V from 6.43 years operating at 0.71V without peripheral assists, assuming that the SRAM and core logic shares

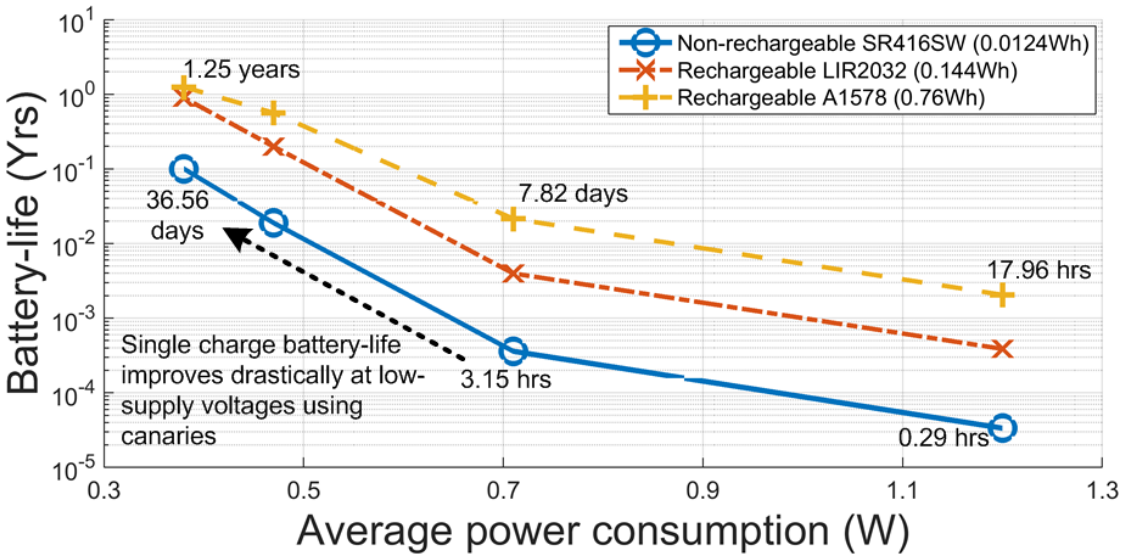


Figure 9.7: Estimated battery-life of SR416SW, LIR2032, and A1578 batteries using CPA and in-situ canary-based V_{MIN} tracking in a commercial 130nm bulk technology.

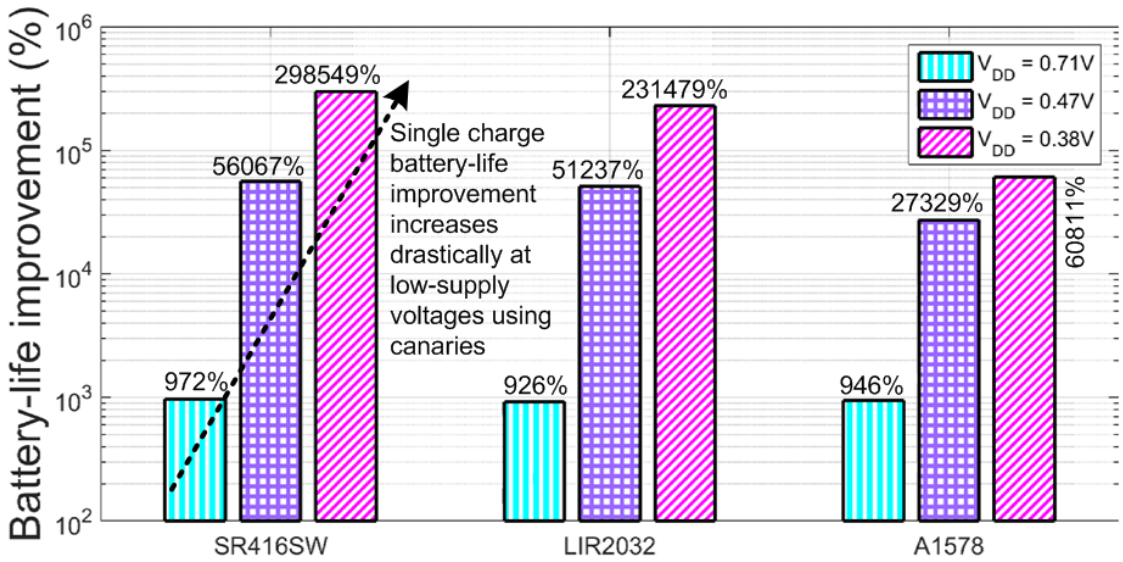


Figure 9.8: Estimated battery-life improvement of SR416SW, LIR2032, and A1578 batteries using CPA and in-situ canary-based V_{MIN} tracking in a commercial 130nm bulk technology.

the same supply rail. Simulations show that the canary-based V_{MIN} tracking technique is scalable to 45nm and 32nm technologies, too. The sub-tasks and outcomes associated with this research are listed below.

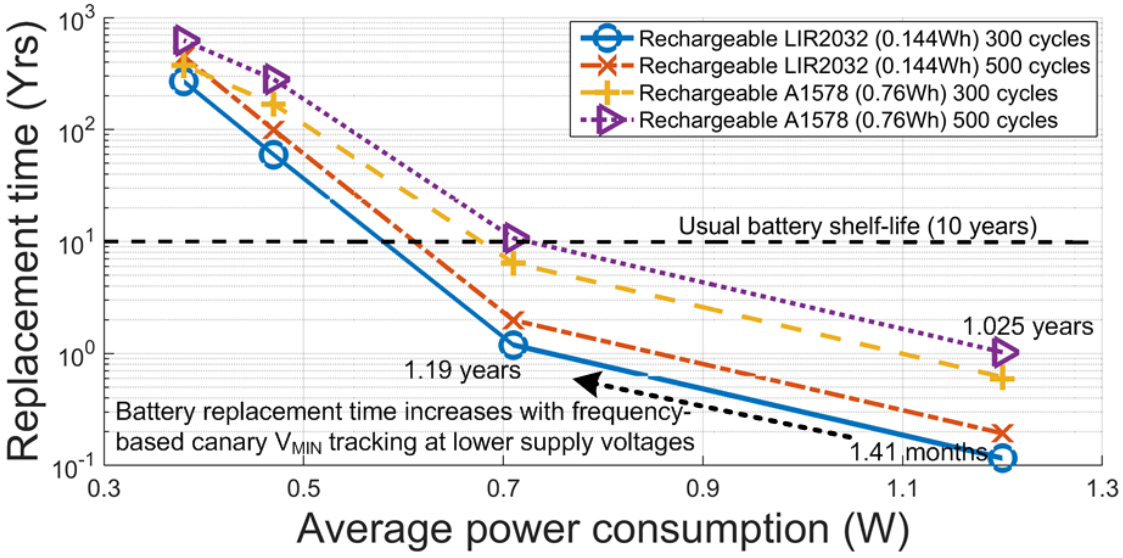


Figure 9.9: Estimated battery replacement time of LIR2032, and A1578 batteries assuming 300 and 500 charge-discharge cycles using CPA and in-situ canary-based V_{MIN} tracking in a commercial 130nm bulk technology.

- Planning and design of canary-based V_{MIN} tracking loop architecture in a team environment.
- Design and development of a closed-loop self-tuning 256kb SRAM testchip with 0.38V-1.2V extended operating range using combined peripheral assists and in-situ V_{MIN} tracking canary sensors in a team. We achieve a maximum of 337X active power reduction using combined read-write peripheral assists and 4.3X power reduction using V_{MIN} (overall 1444X active power reduction capability). The system has a maximum of 12.4X leakage reduction capability.
- Design of a wordline pulse-slope-degradation type pulse-shaping reverse assist circuit and embedded 2kb in-situ canary sensor in the SRAM sub-array.
- Design and development of a 6T SRAM and its sub-components such as 6T bitcell circuit and layout design. Design of 6T core-array with the corner, edge, and tap cell circuits

and layouts, row and column decoder circuit and layout. Design of periphery circuits, such as precharge and equalization logic, write driver, write and read column mux, negative-bitline peripheral assist circuits and layouts. V_{DD} boosting circuit and wordline boosting layout design for the wordline driver.

- Design of 6T SRAM built-in self-test and canary sensor SRAM built-in self-test circuits for supporting the testing of SRAM and enabling the closed-loop for SRAM V_{MIN} tracking.
- Printed circuit board (PCB) design and setup for testing, test vector automation setup, test bench with temperature chamber setup, and measurement of 30 chips in a team environment.
- Measurement of canary-based V_{MIN} tracking using measured results across frequencies such as 1MHz, 10MHz, 50MHz, 100MHz and 150MHz for -20C, 27C and 85C temperatures in 130nm bulk technology in a team setting.
- Findings for scaling of canary-based V_{MIN} tracking in lower technologies using simulated results in 45nm bulk technology across frequencies such as 1MHz, 500MHz, 1GHz, 1.5GHz, 2GHz, and 2.5GHz frequencies for 45nm. Simulation results confirm canary-based V_{MIN} tracking for 100MHz, 1GHz, 2GHz, 3GHz, 4GHz, and 5GHz frequencies in 32nm FDSOI technology at 27C temperature.

Automation of Reverse Assist Analysis and Design for Canary SRAM Design

Design of canary SRAM for SRAM V_{MIN} (SV_{MIN}) tracking across process, voltage, and temperature variation could be a tedious task. This chapter proposes a set of mathematical conditions and the corresponding algorithms (RADA) for canary design automation. Using these algorithms Figure 8.9, Figure 8.10, Figure 8.11 and Figure 8.12 show the optimal write SV_{MIN} tracking in a FinFET technology across TT, FF, SS, and SF corners at 27C temperature. The simulated energy savings compared to the worst-case SF corner at 27C temperature using SV_{MIN} tracking saves 11.4% to 60.9% (Figure 8.13) across corners.

Table 8.1 shows the runtime of each algorithm in RADA for 45nm bulk, 32nm FDSOI, and 14nm FinFET technology. It is impractical to design-explore manually across the canary design knobs, run simulations, and parse data to analyze the vast canary design space. The RADA tool-flow can improve the manual RA-based canary design exploration time from months or weeks to a few days. Thus, RADA reduces an SRAM designer's burden for design exploration of RA-based canary sensors across design knobs for the arbitrary lowering of SV_{MIN} guard-bands.

The RA-based canary design is an attractive solution for reducing the SRAM V_{MIN} guard-band arbitrarily by tracking the SRAM V_{MIN} using canary sensors. This chapter derives the necessary conditions for the first time to track SRAM V_{MIN} across canary design knobs. This work also proposes the Reverse Assist Design and Analysis (RADA) tool-flow in Perl and the corresponding algorithms that bridge the gap between the canary design knobs and analyzes solutions for a valid set of RAS values and F_{th} conditions to optimally track SRAM V_{MIN}. By supporting the automated design exploration and analysis of reverse assist-based canaries for arbitrary guard-band minimization of SRAM V_{MIN}, RADA minimizes canary and overall SRAM design time. RADA indirectly also enables wide-V_{DD} range dynamic voltage scaling (DVS) that requires canary SRAMs for ultra-low power (ULP) IoE applications. The RADA algorithm and tool-flow are independent of the process technology and support conventional bulk, FDSOI, and FinFET technologies. The sub-tasks and outcomes associated with this research are listed below.

- A mathematical framework to determine the optimal V_{MIN} tracking condition based on the available supply voltage granularity, which translates to the design consideration for reverse assist (RA)-based canary sensors to track SRAM V_{MIN} across process, voltage, temperature, and frequency (PVTF) variations.
- A set of algorithms for analysis and design of RA-based canaries to track SRAM V_{MIN} across PVTF variations.

- A Perl-based tool-flow that supports the analysis and design of pulse-shaping wordline and bitline RA-based canaries to track SRAM V_{MIN} across PVTF variations.
- Findings on 6T FinFET HD SRAM V_{MIN} tracking within the supply voltage granularity across process variation and possible energy savings compared to the worst-case corners.
- Benchmark results of analysis and design time for RAs across 45nm bulk, 32nm FDSOI, and 14nm FinFET technology.

9.2 Conclusions, Broader Impact, and Open Questions

With the technology scaling in FinFET transistors toward the molecular dimensions using traditional 193nm lithography, process variation is on the rise again. Even using the deep ultra-violet (UV) lithography, there will be a significant amount of geometry variation effects such as line edge roughness (LER) [95], gate edge roughness (GER) [95], and fin edge roughness (FER) [95] leading to high process variation due to patterning and chemical etching limitations. Moreover, transistor scaling for high volume manufacturing of devices may face a major challenge from gate controllability beyond 7nm and 5nm nodes, where short channel effects such as drain induced barrier lowering (DIBL) [96] and gate induced drain leakage (GIDL) [97] could hamper operating 6T SRAM in nominal and scaled supply voltages. SRAM design would require variable fin-heights or variable channel length FinFETs to support 6T high-density SRAM in future technologies, at the cost of additional mask layers and fabrication process changes. On the other hand, scaled interconnects in 16nm and 14nm are already facing challenges from electromigration (EM) [98] and IR drop (EMIR) issues and with further scaling down to 7nm and 5nm nodes, EMIR issues will be even higher. All of these challenges will influence the design decision of SRAMs for the next generation high performance, ultra-low power (ULP) battery-operated, and ULP energy harvested Internet of Things (IoT) devices. The knowledge contribution of this thesis would allow us to apply

and extend the results obtained to solve the challenges of future SRAM designs for IoT applications, as follows.

This thesis reveals a novel sub-threshold half-select-free 9T bitcell for battery operated ULP biomedical applications, which has lower active write and read energy, as well as lower leakage current per cell in a commercial 130nm bulk technology. The results of this 9T bitcell apply to scaled technologies such as FinFETs, as the solution improves the dynamic energy, leakage current, and row half-select issues using an alternative 9T bitcell topology. However, comparison and tradeoffs of these alternative bitcells across SRAM design knobs in future FinFET technologies is an open question for ULP biomedical applications. Also, this thesis shows a low energy read (LER) architecture for supporting ULP burst reads for sequential read-dominated buffers, which would benefit from energy savings. The results of this ULP architecture is beneficial to other applications too, such as a single line write-read cache memory inbuilt into an SRAM, which can be useful for active energy savings for both write and read operations in ULP buffer or cache applications. An interesting research question to investigate for future works would be: How would these results affect the overall energy savings across various IoT application algorithms running in an SoC?

As smallest size 6T HD FinFET SRAM bitcell loses design knobs to cope with process variation, such as the length and width of FinFETs are being quantized, it fails to write and read at the nominal supply voltage. This thesis shows that none of the single peripheral assists are capable of reducing the worst-case V_{MIN} below the nominal supply voltage for 14nm 6T HD SRAMs. Only a few combinations of dual read-write peripheral assists are capable of lowering the worst-case 6T HD V_{MIN}. These results apply toward the future scaled FinFET technologies to address the challenges of process variation to a certain extent. It would be an interesting research question to investigate how triple combinations of read-write peripheral assists would behave compared to the dual combinations. Moreover, the energy and delay tradeoffs for these dual and triple combinations of peripheral assists would be a crucial open question to be investigated.

This thesis also shows the theory of canary sensor SRAM-based dynamic write V_{MIN} tracking that relates to the SRAM and canary design knobs. It also reveals the tradeoffs of bitline pulse-height-degradation-based reverse assist circuit across canary design knobs. This theory can be extended to read V_{MIN} tracking, too, as it deals with the failure rates of SRAM and canaries without any assumption of write or read operation. This work shows that tracking V_{MIN} could potentially save more than 50% energy compared to the design-for-the-worst-case at 28nm bulk technology. As SRAMs could consume up to 60% of power consumption in modern high-performance as well as low-power SoC applications, applying an in-situ canary-based V_{MIN} sensor could save a lot of power and improve battery-life. However, it is an open question to investigate how the canary influences the design knobs of SRAMs across various power, performance, and area constraints in modern FinFET technologies.

One of the contributions of this thesis is the classification of the wordline and bitline pulse-shaping reverse assists, their properties, and their sensitivity to canary design knobs, such as reverse assist strength, voltage, frequency, and temperature. Findings of this work allow us to choose an apt reverse assist for designing the canary sensor SRAM for V_{MIN} tracking of 6T HD FinFET SRAMs. These results of reverse assist sensitivity, energy, area and circuit figure-of-merit tradeoffs in a 14nm FinFET technology could also apply to canary designs in scaled FinFET technologies of future. There could be other ways of creating a reverse assist for the sense amplifier in 6T SRAM macros, such as degrading the sense enable signal using pulse-shaping techniques. Hence, an open question remains: What are the design tradeoffs for other than bitcell-based canary reverse assist techniques and how do they compare with bitcell degrading reverse assist techniques?

Also, this thesis shows a self-tuning SRAM employing a combined peripheral assist and closed-loop V_{MIN} tracking canary SRAM architecture that automatically adjusts itself very close to the V_{MIN} of the SRAM. The combined assists save more than 300X active power and canary technique allows us to save 4.3X additional power by extending the supply voltage range more than 67% beyond the design-for-the-worst-case methodology. Using both

techniques save a maximum of 1444X power and 12X leakage. This architecture could be extended in FinFET and FDSOI technologies to employ peripheral assist selection based on canary responses to allow us to save energy consumption for the best and typical case SRAM dies. However, there are open questions regarding how and if at all these canary sensors could be used to detect bias temperature instability (BTI) [69] aging in SRAMs and EMIR issues. Moreover, it would be an interesting question to investigate what the design tradeoffs are for continuous IR drop monitoring to control the power management circuits to minimize voltage drop issues.

Besides, this thesis contributes to a mathematical formulation for the design of pulse-shaping-based canary reverse assist circuits for wordline and bitline type reverse assists. Moreover, this work shows an algorithm and a tool flow to analyze and design to expedite the process of canary design, which would be a massive burden if done manually. This work is technology independent and can be extended to future scaled technologies.

Appendix A

Derivation of Various Battery Discharge Equations for Battery-life and Replacement Time Estimation

In this chapter, we derive the battery discharge equations to estimate the single charge battery-life and battery replacement time assuming some practical self-discharge rates for SR416SW (0.0124Wh) Silver Oxide, LIR2032 (0.144Wh) Lithium-ion, and A1578 (0.76Wh) Lithium-ion Polymer batteries, as shown in Table 1.3. Here we assume there is a ξ percent exponential self-discharge [4] rate for initial t_1 hours and after that, every t_2 hours a ζ percent exponential self-discharge rate applies for rest of the battery-life for a single charge. Thus, we can write equation A.1 for the battery potential at time t such that $0 < t <= t_1$, where V_0 is the initial battery potential in Volts, t is the time in hours, and τ_1 is the corresponding time-constant.

$$V = V_0 * exp(-\frac{t}{\tau_1}) \tag{A.1}$$

At $t = t_1$ the battery potential will be self-discharged to $V_0\frac{(100-\xi)}{100}$, and we can write the following.

$$V_0 \frac{(100-\xi)}{100} = V_0 * exp(-\frac{t_1}{\tau_1}) \Rightarrow \tau_1 = -\frac{t_1}{lin(\frac{100-\xi}{100})} \tag{A.2}$$

Hence, after t_1 hours the available battery energy (E_{avail}) and the corresponding available potential (V_{avail}) for self-discharge only are given as follows in equation A.3 and A.4.

$$E_{avail} = E_{batt} * exp(-\frac{t}{\tau_1}) \Rightarrow E_{avail} = V_0 * I_0 * exp(-\frac{t}{\tau_1}) \tag{A.3}$$

$$V_{avail} = \frac{E_{avail}}{I_0} = V_0 * exp(-\frac{t}{\tau_1}) \tag{A.4}$$

Note that the rated average discharge current is denoted by I_0 and the unit of I_0 is Ah in equation A.3. Assuming the average circuit load power as P_{avg}, we can re-write the equation A.4 at time t_1 as follows.

$$V_{avail} = V_0 * exp(-\frac{t_1}{\tau_1}) - \frac{P_{avg} * t_1}{I_0} \tag{A.5}$$

From equation A.5 we can write in terms of E_{avail} as follows.

$$E_{avail} = E_{batt} * exp(-\frac{t_1}{\tau_1}) - P_{avg} * t_1 \tag{A.6}$$

Hence, for $t_1 < t < t_{EOL}$ assuming the battery self-discharge rate is ζ percent per t_2 hours (where t_{EOL} represents the battery's end-of-life time), we can write the following. Where τ_2 is the corresponding time-constant.

$$V = V_{avail} * exp(-\frac{t}{\tau_2}) \tag{A.7}$$

At $t = t_2$ the battery potential will be self-discharged to $V_0 * \frac{(100-\zeta)}{100}$, and we can write the following.

$$V_{avail} * \frac{(100-\zeta)}{100} = V_{avail} * exp(-\frac{t_2}{\tau_2}) \Rightarrow \tau_2 = -\frac{t_2}{lin(\frac{100-\zeta}{100})} \tag{A.8}$$

Hence, for batteries with two types of self-discharge rates ξ and ζ percent for the corresponding initial self-discharge time of t_1 hours and after that every t_2 hours, the single charge battery-life equation is given by the following.

$$(E_{batt} * exp(-\frac{t_1}{\tau_1}) - P_{avg} * t_1) * exp(-\frac{t}{\tau_2}) = P_{avg} * t \tag{A.9}$$

The solution of equation A.9 $t = t_2'$ gives a part of the battery-life, and the following equation gives the estimated total battery-life (t_{blt}).

$$t_{blt} = t_1 + t_2' \tag{A.10}$$

On the other hand, if a battery has a single self-discharge rate and a corresponding time constant, we can write the following equation for the single charge battery-life. Note that in this case, the solution of equation A.11 will give the total battery-life.

$$E_{batt} * exp(-\frac{t}{\tau_1}) = P_{avg} * t \tag{A.11}$$

For a Silver Oxide SR416SW (0.0124Wh) battery, using equation A.11 and self-discharge rate mentioned in Table 1.3 we write the following equation for the estimation of single charge battery-life.

$$0.0124 * exp(-\frac{t}{1.02497 * 10^5}) = P_{avg} * t \tag{A.12}$$

For a Lithium-ion LIR2032 (0.144Wh) battery, using equation A.9 and self-discharge rates mentioned in Table 1.3 we write the following equation for the estimation of single charge battery-life.

$$(0.144 * exp(-\frac{24}{467.8974}) - 24 * P_{avg}) * exp(-\frac{t}{35638.7878}) = P_{avg} * t \tag{A.13}$$

And lastly, for the Lithium-ion Polymer A1578 (0.76Wh) battery, using equation A.11 and self-discharge rate mentioned in Table 1.3 we write the following equation for the estimation of single charge battery-life.

$$0.76 * exp(-\frac{t}{6833.6795}) = P_{avg} * t \tag{A.14}$$

After every charge-discharge cycle in rechargeable batteries, the battery capacity decreases exponentially [4]. A Lithium-ion Polymer or a Lithium-ion battery capacity drops to 80% after 300-500 charge-discharge cycles or more that requires replacement. Assuming N_r number of recharge cycles the following equation gives the battery capacity, where τ_r is a time constant at which the $E_{batt}(N_r)$ reduces to its 80% capacity.

$$E_{batt}(N_r) = E_{batt} * exp(-\frac{N_r}{\tau_r}) \tag{A.15}$$

Hence, the following modified equations A.16 and A.17 for LIR2032 (0.144Wh) and A1578 (0.76Wh) rechargeable batteries give the estimation of battery-life with multiple charge-discharge cycles, respectively. The battery-life corresponding to N_rth recharge is t_r hours. Therefore, the equation A.18 gives the battery replacement time (T_{blt}), which requires all the roots of either of the equations A.16 or A.17 for each value of N_r. The equation A.18 can be simplified as equation A.19, which is used in this thesis for the estimation of replacement time of rechargeable batteries.

$$(0.144 * exp(-\frac{N_r}{\tau_r}) * exp(-\frac{24}{467.8974}) - 24 * P_{avg}) * exp(-\frac{t_r}{35638.7878}) = P_{avg} * t_r \tag{A.16}$$

$$0.76 * exp(-\frac{N_r}{\tau_r}) * exp(-\frac{t_r}{6833.6795}) = P_{avg} * t_r \tag{A.17}$$

$$T_{blt} = \sum_{r=1}^{r=N_r} t_r \tag{A.18}$$

$$T_{blt} <= \sum_{r=1}^{r=N_r} t_1 = N_r * t_1 \tag{A.19}$$

Appendix B

Publications

B.1 Completed

AB1 A. Banerjee and B. H. Calhoun, "An ultra low energy 9T half-select-free subthreshold SRAM bitcell," SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), 2013 IEEE, Monterey, CA, 2013, pp. 1-2.

AB2 A. Banerjee and B.H. Calhoun, "An Ultra-Low Energy Subthreshold SRAM Bitcell for Energy Constrained Biomedical Applications," Journal of Low Power Electronics and Applications. 2014, 4, 119-137.

AB3 A. Banerjee, M. E. Sinangil, J. Poulton, C. T. Gray and B. H. Calhoun, "A reverse write assist circuit for SRAM dynamic write V_{MIN} tracking using canary SRAMs," Quality Electronic Design (ISQED), 2014 15th International Symposium on, Santa Clara, CA, 2014, pp. 1-8.

AB4 A. Banerjee, J. Breiholz and B. H. Calhoun, "A 130nm canary SRAM for SRAM dynamic write V_{MIN} tracking across voltage, frequency, and temperature variations," Custom Integrated Circuits Conference (CICC), 2015 IEEE, San Jose, CA, 2015, pp. 1-4.

AB5 F. Yahya, H. Patel, J. Boley, A. Banerjee and B. H. Calhoun, "A Sub-threshold 8T SRAM Macro with 12.29nW/KB Standby Power and 6.24 pJ/access for Battery-Less IoT SoCs," Journal of Low Power Electronics and Applications (JLPEA), vol. 6, issue 2, 2016.

AB6 A. Banerjee et al., "A 256kb 6T Self-Tuning SRAM with Extended 0.38V-1.2V Operating Range using Multiple Read/Write Assists and VMIN Tracking Canary Sensors," Custom Integrated Circuits Conference (CICC), 2017.

B.1.1 Anticipated (Draft Ready)

AB7 A reverse assist design and analysis algorithm and tool-flow for in-situ canary SRAM design for SRAM dynamic V_{MIN} tracking (2018 TBD).

AB8 Feasibility of 6T FinFET SRAM write V_{MIN} tracking using canary SRAMs and reverse assists (TVLSI 2018)

B.1.2 Anticipated (Text and Figures Ready)

AB9 Classification of reverse assists and its tradeoffs for Canary SRAM (ISLPED 2018)

AB10 Combined peripheral assists combinations for improving read and write operations for 6T HD FinFET SRAMs (TVLSI 2018)

AB11 A self-tuning SRAM architecture enabling wide scale operation of 6T SRAMs using in-situ canary sensor SRAM and read-write peripheral assist combinations. (JSSC 2018)

AB12 A low energy write and read half-select-free peripheral architecture for sub-threshold applications (2018 TBD)

Patents

AB13 A. Banerjee et al., "Approach to Predictive Verification of Write Integrity in a Memory Driver," US Patent, NVIDIA Corporation. Jan 2014

AB14 A. Banerjee et al., "Method to Lower SRAM Dynamic V_{MIN} and Reduce the Dynamic V_{MIN} Guard-Band Arbitrarily," US Provisional Patent application, University of Virginia. April 2017

List of Acronyms

10T Ten transistor

2T Two transistor

6T Six transistor

8T Eight transistor

9T Nine transistor

ASC Assist controller

BA Bit adder

BASN Body area sensor node

BCP BIST computation pipeline

BEA Bit error accumulator

BFSM BIST finite state machine

BIL Bus interface logic

BIST Built-in self-test

BL Bitline

BLB Bitline-bar

BLPHD Bitline pulse height degradation

BLPSD Bitline pulse slope degradation

BLPWD Bitline pulse width degradation

BLVRA Bitline reverse assist voltage

BSN Body sensor node

BXOR Bitwise xor

CAS Column access strobe

CBIST Canary built-in self-test

CCSM Canary control logic state machine

CM Column mux

CMOS Complementary metal oxide semiconductor

CPA Combined peripheral assists

CPU Central processing unit

CV_{DD} Canary supply voltage

CV_{MIN} Canary minimum operating voltage

CWF Canary write failure

DARPA Defense advanced research projects agency

DIN Data input

DOUT Data output

DPM Digital power management

DRAM Dynamic random access memory

DRV Data retention voltage

DVFS Dynamic voltage and frequency scaling

DVS Dynamic voltage scaling

$\mathbf{E_{avgop}}$ Read-write weighted energy per operation

ECG Electrocardiogram

EMG Electromyogram

ENSA Enable sense amplifier

EPO Energy per operation

$\mathbf{E_{rd}}$ Read energy per operation

$\mathbf{E_{wr}}$ Write energy per operation

$\mathbf{F_c}$ Canary failure

FDC Frequency to digital converter

FDSOI Fully depleted silicon on insulator

FF Fast NMOS fast PMOS

FIFO First-in-first-out

FinFET Fin-shaped field effect transistor

FOM Figure of merit

$\mathbf{F_{rdwr}}$ Fraction of read and write

FS Fast NMOS slow PMOS

F_{th} Failure threshold condition

GHz Gigahertz

HD High density

HSNM Hold static noise margin

I/O Input and output

IEEE Institute of electrical and electronics engineers

IoE Internet of everything

IoT Internet of things

KB Kilobyte

kb Kilobit

kHz Kilohertz

LDO Low-dropout regulator

LER Low energy read

LUT Lookup table

MC Monte carlo

MEP Minimum energy point

MHz Megahertz

MOSFET Metal oxide semiconductor field effect transistor

mV Milli volt

NBL Negative bitline

NMOS N-type metal oxide semiconductor

NSF National science foundation

NVS Negative ground

PCB Printed circuit board

PERFECT Power efficiency revolution for embedded computing technologies

$\mathbf{P_{f_c}}$ Canary chip failure probability

PHD Pulse height degradation

PIC Peripheral interface controller

PMOS P-type metal oxide semiconductor

PSD Pulse slope degradation

PVT Process, voltage, and temperature

PWD Pulse width degradation

PWL Piece-wise linear

RA Reverse assist

RADA Reverse assist design and analysis

RAS Reverse assist setting

RBL Read bitline

RPB Rows per bank

RRAM Resistive random access memory

RSNM Read static noise margin

RWL Read wordline

SA Sense amplifier

SED Soft error disturb

SER Soft error rate

SF Slow NMOS fast PMOS

SNM Static noise margin

SoC System on chip

SRAM Static random access memory

SS Slow NMOS slow PMOS

STT Spin torque transfer

SVMIN Static random access memory minimum operating voltage

$\mathbf{T_{Acc}}$ Read access time

$\mathbf{T_{crit}}$ Critical wordline pulse-width

$\mathbf{T_{SA}}$ Sense amplifier reaction time

TT Typical NMOS typical PMOS

$\mathbf{T_{V_{Diff}}}$ Bitline differential development time

ULP Ultra-low power

VDB Supply voltage boosting

VDD Supply voltage

VDU Supply voltage collapse

VFT Voltage, frequency, and temperature

ViPro Virtual prototyper

VLSI Very large scale integration

VMIN Minimum operating voltage

VRA Reverse assist voltage

VSR Ground raising

VSS Ground voltage

VT Threshold voltage

WAR Write-after-read

WBL Write bitline

WBLB Write bitline-bar

WEC Write error comparator

WL Wordline

WLB Wordline boosting

WLPHD Wordline pulse height degradation

WLPSD Wordline pulse slope degradation

WLPWD Wordline pulse width degradation

WLU wordline under-drive

WLVRA Wordline reverse assist voltage

WM Write margin

WSNM Write static noise margin

WTI Write trip current

WTV Write trip voltage

WVMIN Write minimum operating voltage

Bibliography

[1] Dave Evans, "The Internet of Things, How the Next Evolution of the Internet Is Changing Everything," *CISCO, IBSG*, http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, April 2011.

[2] R.J.M. Vullers, R. van Schaijk, I. Doms, C. Van Hoof, R. Mertens, "Micropower energy harvesting," *Solid-State Electronics*, Volume 53, Issue 7, 2009, Pages 684-693, ISSN 0038-1101, https://doi.org/10.1016/j.sse.2008.12.011.

[3] https://www.batteries.com/pages/coin-cell-button-cell-battery-guide

[4] http://batteryuniversity.com/learn/archive/whats_the_best_battery

[5] http://batteryuniversity.com/learn/article/elevating_self_discharge

[6] https://www.eetimes.com/author.asp?section_id=36&doc_id=1322736

[7] T. H. Kim, J. Liu, J. Keane and C. H. Kim, "A 0.2 V, 480 kb Subthreshold SRAM With 1 k Cells Per Bitline for Ultra-Low-Voltage Computing," *in IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 518-529, Feb. 2008.

[8] Yen-Huei Chen et al., "A 16nm 128Mb SRAM in high-K metal-gate FinFET technology with write-assist circuitry for low-VMIN applications," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International, San Francisco, CA*, 2014, pp. 238-239.

[9] Mohamed Abu Rahma and Mohab Anis, "Nanometer Variation-tolerant SRAM: Circuits and Statistical Design for Yield," *Book, New York: Springer*, 2013. Print.

[10] SureCore, Technology Whitepaper, *Whitepaper, SureCore*, http://www.sure-core.com/downloads/sureCore-Technology-Whitepaper-May13.pdf, May, 2013.

[11] Naveen Verma, "*Ultra-low-power SRAM Design in High Variability Advanced CMOSCMOS*," PhD thesis, Massachusetts Institute of Technology, 2009.

[12] Yanqing Zhang, "*Synthesis Based Design Techniques for Robust, Energy Efficient Subthreshold Circuits*," PhD thesis, University of Virginia, December, 2013.

[13] S. Amrutur Bharadwaj, "*Design and Analysis of Fast Low Power SRAMs*," PhD thesis, Stanford University,1999.

[14] Arvind Kumar Mishra et al., "Novel design technique of address Decoder for SRAM," *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on, Ramanathapuram*, 2014, pp. 1032-1035.

[15] Y. Ren et al., "Low power 6T-SRAM with tree address decoder using a new equalizer precharge scheme," *SOC Conference (SOCC), 2012 IEEE International, Niagara Falls, NY*, 2012, pp. 224-229.

[16] T. Mori et al., "A 1 V 0.9 mW at 100 MHz 2 k/spl times/16 b SRAM utilizing a half-swing pulsed-decoder and write-bus architecture in 0.25 /spl mu/m dual-Vt CMOS," *Solid-State Circuits Conference, 1998. Digest of Technical Papers. 1998 IEEE International, San Francisco, CA, USA*, 1998, pp. 354-355.

[17] K. W. Mai et al., "Low-power SRAM design using half-swing pulse-mode techniques," *in IEEE Journal of Solid-State Circuits*, vol. 33, no. 11, pp. 1659-1671, Nov 1998.

[18] G. Samson et al., "Low-Power Dynamic Memory Word Line Decoding for Static Random Access Memories," *in IEEE Journal of Solid-State Circuits*, vol. 43, no. 11, pp. 2524-2532, Nov. 2008.

[19] Shah M. Jahinuzzaman, "*Modeling and Mitigation of Soft Errors in Nanoscale SRAMs*," PhD thesis, University of Waterloo, Electrical and Computer Engineering, Canada, 2008

[20] James Boley, "*Circuit and CAD Techniques for Expanding the SRAM Design Space*" PhD thesis, University of Virginia, Electrical and Computer Engineering, VA-22903, USA, 2014.

[21] Yanqing Zhang et al., "A batteryless 19 W MICS/ISM-band energy harvesting body sensor node SoC for ExG applications," *IEEE J. Solid-State Circuits*, 2013, 48, 199-213.

[22] G. Chen et al., "Millimeter-scale nearly perpetual sensor system with stacked battery and solar cells," *In Proceedings of the 2010 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, San Francisco, CA, USA, 7-11 February 2010; pp. 288-289.

[23] A. Wang et al., "Optimal supply and threshold scaling for subthreshold CMOS circuits," *In Proceedings of the IEEE Computer Society Annual Symposium on VLSI, Pittsburgh, PA, USA*, 25-26 April 2002; pp. 5-9.

[24] A. Wang et al., "A 180-mV subthreshold FFT processor using a minimum energy design methodology. " *IEEE J. Solid-State Circuits 2005* , 40, 310-319.

[25] E. Seevinck et al., "Static-noise margin analysis of MOS SRAM cells," *IEEE J. Solid-State Circuits*, 1987, 22, 748-754.

[26] J.P. Kulkarni et al., "A 160 mV robust schmitt trigger based subthreshold SRAM," *IEEE J. Solid-State Circuits*, 2007, 42, 2303-2313.

[27] I. J. Chang et al., "A 32 kb 10T sub-threshold SRAM array with bit-interleaving and differential read scheme in 90 nm CMOS," *IEEE J. Solid-State Circuits* 2009, 44, 650-658.

[28] A. Feki et al., "Proposal of a new ultra low leakage 10T sub threshold SRAM bitcell," *In Proceedings of the 2012 International SoC Design Conference (ISOCC), Jeju Island, Korea*, 4-7 November 2012; pp. 470-474.

[29] Y. W. Chiu et al., "8T Single-ended sub-threshold SRAM with cross-point data-aware write operation," *In Proceedings of the 2011 International Symposium on Low Power Electronics and Design (ISLPED), Fukuoka, Japan*, 1-3 August 2011; pp. 169-174.

[30] N. Verma and A. P. Chandrakasan, "A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy," *IEEE J. Solid-State Circuits*, 2008, 43, 141-149.

[31] V. Chandra et al., "On the efficacy of write-assist techniques in low voltage nanoscale SRAMs," *In Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE), Dresden, Germany*, 8-12 March 2010; pp. 345-350.

[32] R. W. Mann et al., "Limits of bias based assist methods in nano-scale 6T SRAM," *In Proceedings of the 2010 11th International Symposium on Quality Electronic Design (ISQED), San Jose, CA, USA*, 22-24 March 2010; pp. 1-8.

[33] T. H. Kim et al., "A high-density subthreshold SRAM with data-independent bitline leakage and virtual ground replica scheme," *IEEE J. Solid-State Circuits*, 2008, 43, 518-529.

[34] H. I. Yang et al., "A high-performance low VMIN 55 nm 512 Kb disturb-free 8T SRAM with adaptive VVSS control," *In Proceedings of the 2011 IEEE International SOC Conference (SOCC), Taipei, Taiwan*, 26-28 September 2011; pp. 197-200.

[35] C. Slayman, "Soft errors-Past history and recent discoveries," *In Proceedings of the 2010 IEEE International Integrated Reliability Workshop Final Report (IRW), Stanford Sierra, CA, USA*, 17-21 October 2010; pp. 25-30.

[36] A. Banerjee and B.H. Calhoun, "An ultra low energy 9T half-select-free subthreshold SRAM bitcell," *In Proceedings of the 2013 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), Monterey, CA, USA*, 7-10 October 2013; pp.1-2.

[37] Arijit Banerjee and Benton H. Calhoun, "An Ultra-Low Energy Subthreshold SRAM Bitcell for Energy Constrained Biomedical Applications," *J. Low Power Electron. Appl.* 4, no. 2: 119-137.

[38] H. Kim et al., "A configurable and low-power mixed signal SoC for portable ECG monitoring applications," *IEEE Trans. Biomed. Circuits Syst. 2013*, PP, 1.

[39] L. Yan et al., "A 3.9 mW 25-electrode reconfigured sensor for wearable cardiac monitoring system," *IEEE J. Solid-State Circuits 2011*, 46, 353-364.

[40] G. K. Reddy et al., "Process variation tolerant 9T SRAM bitcell design," *in Quality Electronic Design (ISQED), 2012 13th International Symposium on*, 2012, pp. 493-497.

[41] B. H. Calhoun and A. Chandrakasan, "A 256kb sub-threshold SRAM in 65nm CMOS," *in Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, 2006, pp. 25922601.

[42] Ali Valaee and Asim J. Al-Khalili, "SRAM Read-Assist Scheme for High Performance Low Power Applications," *in International SoC Design Conference (ISOCC) on*, 2011, pp. 179-182.

[43] S. Yoshimoto, M. Terada, S. Okumura, T. Suzuki, S. Miyano, H. Kawaguchi and M. Yoshimoto, "A 40-nm 0.5-V 20.1-W/MHz 8T SRAM with Low-Energy Disturb Mitigation Scheme," *in IEEE Symposium on VLSI Circuits Digest of Technical Papers on*, 2011, pp. 72-73.

[44] Atsushi Kawasumi, Toshikazu Suzuki, Shinich Moriwaki and Shinji Miyano, "Energy Efficiency Degradation Caused by Random Variation in Low-Voltage SRAM and 26% Energy Reduction by Bitline Amplitude Limiting (BAL) Scheme," *in IEEE Asian Solid-State Circuits Conference on*, 2011, pp. 165-168.

[45] Mohammad Sharifkhani, Manoj Sachdev, "A Low Power SRAM Architecture Based on Segmented Virtual Grounding," *in International symposium on Low Power Electronics and Design (ISLPED) on*, 2006, pp. 256-261.

[46] A. Kawasumi, Y. Takeyama, O. Hirabayashi, K. Kushida, F. Tachibana. Y. Niki, S. Sasaki and T. Yabe, "Energy Efficiency Deterioration by Variability in SRAM and Circuit Techniques for Energy Saving without Voltage Reduction," *in IC Design & Technology (ICICDT), 2012 IEEE International Conference on*, 2012.

[47] Mohammed Shareef I, Pradeep Nair, Bharadwaj Amrutur, "Energy Reduction in SRAM using Dynamic Voltage and Frequency Management," *in 2008 21st International Conference on VLSI Design on*, 2008, pp. 503-508.

[48] https:opencores.orgproject,openmsp430, *Created: Jun 30, 2009*

[49] Yahya, F., H. Patel, J. Boley, A, . Banerjee and B. H. Calhoun, "A Sub-threshold 8T SRAM Macro with 12.29nW/KB Standby Power and 6.24 pJ/access for Battery-Less IoT SoCs," *Journal of Low Power Electronics and Applications (JLPEA)*, vol. 6, issue 2, 2016.

[50] Jiajing Wang et al., "Analyzing static and dynamic write margin for nanometer SRAMs," *Low Power Electronics and Design (ISLPED), 2008 ACM/IEEE International Symposium on, Bangalore*, 2008, pp. 129-134.

[51] C. Wann et al., "SRAM cell design for stability methodology," *VLSI Technology, 2005. (VLSI-TSA-Tech). 2005 IEEE VLSI-TSA International Symposium on*, 2005, pp. 21-22.

[52] E. Grossar et al., "Read Stability and Write-Ability Analysis of SRAM Cells for Nanometer Technologies," *in IEEE Journal of Solid-State Circuits*, vol. 41, no. 11, pp. 2577-2588, Nov. 2006.

[53] Randy Mann, "*Interaction of Technology and Design in Nanoscale SRAM*," PhD thesis, University of Virginia, 2010.

[54] F. B. Yahya et al., "Combined SRAM read/write assist techniques for near/sub-threshold voltage operation," *Quality Electronic Design (ASQED), 2015 6th Asia Symposium on, Kula Lumpur*, 2015, pp. 1-6.

[55] H. N. Patel et al., "Improving Reliability and Energy Requirements of Memory in Body Sensor Networks," *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), Kolkata, India*, 2016, pp. 561-562.

[56] B. Zimmer et al., "SRAM Assist Techniques for Operation in a Wide Voltage Range in 28-nm CMOS," *in IEEE Trans. Circuits Syst. II*, vol. 59, no. 12, pp. 853-857, 2012.

[57] M. Qazi, et al., "A 512kb 8T SRAM macro operating down to 0.57V with an AC-coupled sense amplifier and embedded data-retention-voltage sensor in 45nm SOI CMOS," *ISSCC, Dig. Tech. Papers*, pp. 350-351, 2010.

[58] E. Karl et al., "A 4.6ghZ 162Mb SRAM design in 22nm trigate CMOS technology with integrated active Vmin-enhancing assist circuitry," *in IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC), San Francisco, CA*, pp. 230-232, 2012.

[59] Y. Sinangil, et al., "A self-aware processor SoC using energy monitors integrated into power converters for self-adaptation," *Symp. VLSI Circuits Dig. Tech. Papers*, pp. 1-2, 2014.

[60] M. F. Chang, et al., "A 28nm 256kb 6T-SRAM with 280mV improvement in VMIN using a dual-split-control assist scheme," *ISSCC Dig. Tech. Papers*, pp. 1-3, 2015.

[61] S. Herbert and D. Marculescu, "Analysis of dynamic voltage/frequency scaling in chip-multiprocessors," *in Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, pp. 38-43, 2007

[62] A. Genser et al., "Power emulation based DVFS efficiency investigations for embedded systems," *in Proc. Int. Symp. Syst. Chip (SoC)*, pp. 173-178, 2010

[63] R. Airoldi et al., "Improving Reconfigurable Hardware Energy Efficiency and Robustness via DVFS-Scaled Homogeneous MP-SoC," *in Proc. IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW)*, pp. 286-289, 2011.

[64] J. Pille et al., "Implementation of the CELL Broadband Engine in a 65nm SOI Technology Featuring Dual-Supply SRAM Arrays Supporting 6GHz at 1.3V," *in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, pp. 322-323, 606, 2007.

[65] A. Bhavnagarwala et al., "Fluctuation limits and scaling opportunities for CMOS SRAM cells," *in IEDM Tech. Dig.*, 2005, pp. 659-662, 2005.

[66] N. Shibata et al., "A 0.5-V 25-MHz 1-mW 256-kb MTCMOS/SOI SRAM for solar-power-operated portable personal digital equipment-Sure write operation by using step-down negatively overdriven bitline scheme," *in IEEE J. Solid-State Circuits*, pp.728 -742, 2006.

[67] Jonathan Chang et al., "A 20nm 112Mb SRAM in High-K Metal-Gate with Assist Circuitry for Low-Leakage and Low-VMIN Applications," *in Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, pp. 316 - 317, 2013.

[68] A. Shah and H. Mahmoodi, "Thermal estimation for accurate estimation of impact of BTI aging effects on nano-scale SRAM circuits," *in SOC Conference (SOCC), 2010 IEEE International*, pp. 230-235, 2010.

[69] A. Bansal et al., "Impact of NBTI and PBTI in SRAM bit-cells: Relative sensitivities and guidelines for application-specific target stability/performance," *in IEEE IRPS*, pp. 745-749, 2009.

[70] S. C. Yang et al., "Timing control degradation and NBTI/PBTI tolerant design for write-replica circuit in nanoscale CMOS SRAM," *in Proc. IEEE Int. Symp. VLSI Design, Autom., Test*, pp. 162-165, 2009.

[71] S. Nalam et al., "Dynamic write limited minimum operating voltage for nanoscale SRAMs," *in Proc. Des. Autom. Test Eur.*, pp. 1-6, 2011.

[72] J. Wang et al., "Two Fast Methods for Estimating the Minimum Standby Supply Voltage for Large SRAMs," *in Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 29, issue 12, pp. 1908-1920, 2010.

[73] B. H. Calhoun and A. P. Chandrakasan, "Standby power reduction using dynamic voltage scaling and canary flip-flop structures," *in IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1504-1511, 2004.

[74] Y. Otsuka et al., "Multicore energy reduction utilizing canary FF," *in Communications and Information Technologies (ISCIT)*, 2010 International Symposium, pp. 922-927, 2010.

[75] J. Wang and B. Calhoun, "Canary replica feedback for near-DRV standby VDD scaling in a 90 nm SRAM," *in Proc. Custom Integrated Circuit Conf. (CICC 07)*, pp. 29-32, 2007.

[76] J. Wang and B. H. Calhoun, "Techniques to Extend Canary-based Standby VDD Scaling for SRAMs to 45nm and Beyond," *in IEEE Journal of Solid-State Circuits*, vol. 43, pp. 2514-2523, 2008.

[77] J. Wang et al., "An Enhanced Canary-based System with BIST for SRAM Standby Power Reduction," *in Transactions on VLSI Systems (TVLSI)*, pp. 909-914, 2011.

[78] L. Dolecek et al., "Breaking the simulation barrier: SRAM evaluation through norm minimization," *in Proc.IEEE/ACM Int. Conf. Comput.-Aided Des.*, pp. 322-329, 2008.

[79] R. Kanj et al., "Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events," *in Proc. ACM/IEEE Des. Autom. Conf.*, pp. 69-72, 2006.

[80] A. Singhee et al., "Recursive Statistical Blockade: An Enhanced Technique for Rare Event Simulation with Application to SRAM Circuit Design," *in International Conference on VLSI Design, India*, pp. 131-136, 2008.

[81] A. Banerjee et al., "A reverse write assist circuit for SRAM dynamic write VMIN tracking using canary SRAMs," *Quality Electronic Design (ISQED), 2014 15th International Symposium on, Santa Clara, CA*, 2014, pp. 1-8.

[82] A. Kincel and M. Balaz, "MBIST for LEON3 processor core cache," *2013 IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, vol., no., pp. 287,288, 2013.

[83] A.M. Sodagar and K. Najafi, "A multi-output supply-independent voltage reference in standard CMOS process for telemetry-powering applications," *Signals, Circuits and Systems, 2003. SCS 2003. International Symposium on (SCS)*, vol.2, no., pp.493,496, 2003

[84] Joyce Kwong, et al., "An Energy-Efficient Biomedical Signal Processing Platform," *in IEEE Journal of Solid-State Circuits*, vol. 46, no. 7, pp. 1742-1753, July 2011.

[85] Abhishek Roy, et al., "A 1.3uW, 5pJ/cycle Sub-threshold MSP430 Processor in 90nm xLP FDSOI for Energy-efficient loT Applications," *17th International Symposium on Quality Electronic Design (ISQED)*, pp. 158-162, 2016.

[86] Y. C. Lai, S. Y. Huang and H. J. Hsu, "Resilient Self-VDD-Tuning Scheme With Speed-Margining for Low-Power SRAM," *in IEEE Journal of Solid-State Circuits*, vol. 44, no. 10, pp. 2817-2823, Oct. 2009.

[87] A. Banerjee et al., "A 130nm canary SRAM for SRAM dynamic write VMIN tracking across voltage, frequency, and temperature variations," *Custom Integrated Circuits Conference (CICC), 2015 IEEE, San Jose, CA*, 2015, pp. 1-4.

[88] J. Jiang and Y. Lu and W. H. Ki and S. P. U and R. P. Martins, "A dual-symmetrical-output switched-capacitor converter with dynamic power cells and minimized cross regulation for application processors in 28nm CMOS," *2017 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco*, CA, 2017, pp. 344-345.

[89] J. Chang et al., "A 7nm 256Mb SRAM in high-k metal-gate FinFET technology with write-assist circuitry for low-VMIN applications," *2017 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA*, 2017, pp. 206-207.

[90] M. Clinton et al., "A low-power and high-performance 10nm SRAM architecture for mobile applications," *2017 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA*, 2017, pp. 210-211.

[91] A. Banerjee, N. Liu, H. Patel, J. Poulton, C. Gray, and B. Calhoun, "A 256kb 6T Self-Tuning SRAM with Extended 0.38V-1.2V Operating Range using Multiple Read-/Write Assists and VMIN Tracking Canary Sensors," *2017 CICC, Austin, TX*, 2017.

[92] S. Nalam et al., "Virtual prototyper (ViPro): An early design space exploration and optimization tool for SRAM designers," *Design Automation Conference (DAC), 2010 47th ACM/IEEE, Anaheim, CA, USA*, 2010, pp. 138-143.

[93] J. Boley et al., "Virtual Prototyper (ViPro): An SRAM Design Tool for Yield Constrained Optimization," *in IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 12, pp. 3109-3113, Dec. 2015.

[94] V. R. Devanathan and S. Kale, "A reconfigurable built-in memory self-repair architecture for heterogeneous cores with embedded BIST datapath," *2016 IEEE International Test Conference (ITC), Fort Worth, TX*, 2016, pp. 1-6.

[95] X. Wang, B. Cheng, A. R. Brown, C. Millar and A. Asenov, "Statistical variability in 14-nm node SOI FinFETs and its impact on corresponding 6T-SRAM cell design," *2012 Proceedings of the European Solid-State Device Research Conference (ESSDERC), Bordeaux*, 2012, pp. 113-116.

[96] H. Mertens et al., "Gate-all-around MOSFETs based on vertically stacked horizontal Si nanowires in a replacement metal gate process on bulk Si substrates," *2016 IEEE Symposium on VLSI Technology, Honolulu, HI*, 2016, pp. 1-2.

[97] P. Kerber, Q. Zhang, S. Koswatta and A. Bryant, "GIDL in Doped and Undoped FinFET Devices for Low-Leakage Applications," *in IEEE Electron Device Letters*, vol. 34, no. 1, pp. 6-8, Jan. 2013.

[98] S. O. Koswatta et al., "Off-state self-heating, micro-hot-spots, and stress-induced device considerations in scaled technologies," *2015 IEEE International Electron Devices Meeting (IEDM), Washington, DC*, 2015, pp. 20.2.1-20.2.4.