Data remanence vulnerabilities in embedded SRAM at low temperature

Farzana Hoque

Follow this and additional works at: https://louis.uah.edu/uah-theses

Recommended Citation
Hoque, Farzana, "Data remanence vulnerabilities in embedded SRAM at low temperature" (2024). Theses. 687.
https://louis.uah.edu/uah-theses/687

This Thesis is brought to you for free and open access by the UAH Electronic Theses and Dissertations at LOUIS. It has been accepted for inclusion in Theses by an authorized administrator of LOUIS.
DATA REMANENCE VULNERABILITIES IN EMBEDDED SRAM AT LOW TEMPERATURE

Farzana Hoque

A THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in The Department of Electrical and Computer Engineering to The Graduate School of The University of Alabama in Huntsville

August 2024

Approved by:
Dr. Biswajit Ray, Research Advisor
Dr. Aleksandar Milenkovic, Committee Chair
Dr. Aubrey Beal, Committee Member
Dr. Aleksandar Milenkovic, Department Chair
Dr. Shankar Mahalingham, College Dean
Dr. Jon Hakkila, Graduate Dean
Abstract

DATA REMANENCE VULNERABILITIES IN EMBEDDED SRAM AT LOW TEMPERATURE

Farzana Hoque

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering

Electrical and Computer Engineering
The University of Alabama in Huntsville
August 2024

Static Random Access Memory (SRAM) memory is prevalent as cache memory in computing platforms and embedded systems. SRAM frequently stores essential information such as cryptographic keys, passwords, and other confidential data. Consequently, a data remanence-based attack on SRAM can result in significant damage. When power is removed, SRAM gradually loses data rather than immediately, especially at lower temperatures. This phenomenon, known as data remanence, poses a significant security risk, as it can be exploited through cold boot attacks targeting encryption keys and other sensitive data. Understanding temperature-induced vulnerabilities in embedded SRAM is crucial for enhancing the security and reliability of embedded systems, especially in IoT devices. This thesis examines the impact of temperature on the data remanence characteristics of embedded SRAM in the 16-bit MSP430F5529 microcontroller developed by Texas Instruments. The research included conducting practical data remanence attack at low temperatures to assess their impact on modern embedded SRAM. The experimental method involved writing a specific image to the SRAM, observing its retention over time by toggling power, and comparing the retained data to the original to determine data loss percentages.
Our findings show that, at -25°C, data showed nearly perfect retention for up to 800 milliseconds for known images. These results underscore the potential security vulnerabilities of embedded SRAM memory when subjected to low-temperature data remanence attacks.
Acknowledgements

I would like to express my deepest gratitude to all those who have supported me throughout the completion of this Master’s thesis.

First and foremost, I am profoundly grateful to my research advisor, Dr. Biswajit Ray, for his invaluable guidance, continuous support, and encouragement throughout this research journey. His insightful feedback and unwavering patience were pivotal to the successful completion of this work. I extend my heartfelt thanks to Dr. Aleksandar Milenkovic, the Committee Chair, for his critical insights and advice, which greatly enhanced the quality of this thesis. I am also deeply grateful to Dr. Aubrey Beal, Committee Member, for his invaluable support and academic guidance.

I am sincerely thankful to my fellow lab mates, Misha, Umesh, Will, Anik, Sijay and Raquib from Dr.Ray’s research group whose collaboration and camaraderie created an enriching and enjoyable research environment. Your assistance and encouragement were instrumental throughout this journey.

I would also like to acknowledge the support of the entire faculty and staff of the Electrical and Computer Engineering department for providing the necessary resources and a conducive learning environment.

Lastly, I am deeply grateful to my family and friends for their unwavering support, understanding, and encouragement, which kept me motivated and focused during challenging times.

Thank you all for your support and encouragement.
Table of Contents

Abstract ................................................................. ii

Acknowledgements ...................................................... v

Table of Contents ....................................................... ix

List of Figures ........................................................... x

List of Tables ............................................................ xiii

Chapter 1. Introduction ............................................... 1
  1.1 Motivation ......................................................... 1
  1.2 Thesis Objective .................................................. 5
  1.3 Thesis Outline .................................................... 6

Chapter 2. Background and Related Works .......................... 9
  2.1 Memory Subsystem in Computing Devices ...................... 9
    2.1.1 Memory Hierarchy .......................................... 9
    2.1.2 Random Access Memory .................................... 11
  2.2 Static Random Access Memory .................................. 11
6.3 Contribution to the Field ... 68

References ... 70

Appendix A: Code for SRAM Data Remanence Analysis in MSP430F5529 MCU ... 75
List of Figures

2.1 Memory Hierarchy of computer storage. CPU registers, Cache, and Main memory are semiconductor-based. .................................................. 10

2.2 Six-transistor SRAM memory cell: (a) 6T SRAM bit cell where P1-N1 and P2-N2 form two cross-coupled inverters and A1 and A2 are the access transistors. (b) SRAM bitcell showing two cross-coupled inverter that stores one bit data. .................................................. 12

2.3 Architecture of an $N \times M$ SRAM memory array. ......................... 13

2.4 Two possible states of a pair of cross-coupled inverters, with the value stored in the cell defined by the value in $Q$. ......................... 14

2.5 SRAM cell bistability: (a) Voltage-transfer characteristic of two cross-coupled inverters (butterfly diagram), with the left diagrams showing the individual Voltage Transfer Characteristics (VTCs). (b) Metastable state C compared to stable operation points A and B. .................................................. 15

2.6 Illustration of the spatial process-variation hierarchy: (a) lot-to-lot, (b) wafer-to-wafer (c) chip-to-chip, (d) within-die variations, and (e) typical CMOS transistor device parameter space and performance space with corresponding worst-case corners [1]. ......................... 19

2.7 Schematic of an SRAM cell showing power up current distribution: (a) current distribution of phase-1 power-up transients, (b) current distribution of phase-2 power-up transients, (c) transient power-up characteristics of the output node [2]. ......................... 21

2.8 Example of a 64-bit fingerprint extracted from a larger SRAM fingerprint, shown in the background. The shading of each cell indicates the probability (p) of powering up to ‘1’, as measured over 100 trials [3]. ......................... 23

2.9 Basic principle of SRAM-PUFs for key generation [4]. ......................... 24
2.10 SRAM cell set to "1" state with P1 and N2 transistors active, P2 and N1 inactive. Access transistors remain off during data hold. If powered off, node Q discharges via N1, P1, and A1 due to transistor leakage.

2.11 Visualization of DRAM memory decay from [5]. A bitmap was loaded, and power was cut off for intervals of 5 seconds, 30 seconds, 60 seconds, and 5 minutes at room temperature. The degradation shows horizontal bars from alternating ground states and vertical bars from physical chip variations.

2.12 A cold-boot attack on a laptop, where the memory module is cooled down to $-50^\circ C$ by directly spraying an upside-down can of multipurpose gas duster onto the chip [5].

3.1 The Experimental Setup.

3.2 The development board and MCU utilized for our experiment: (a) MSP-EXP430F5529LP MSP430F5529 LaunchPad™ development board, and (b) the 16-bit MSP430F5529 microcontroller from Texas Instruments.

3.3 The Architecture for 16-bit MSP430F5529 microcontroller highlighting the SRAM memory subsystem that we utilized for our research.

3.4 Data collection procedure from the embedded SRAM during experiment.

4.1 The classification method for defining natural bits and unstable bits.

4.2 Bit value transition behaviour for SRAM power up state, at room temperature. (a) Bit transition curve for a stable natural 0. (b) Bit transition curve for a stable natural 1. (c) Bit transition curve for an unstable natural 0 bit. (d) Bit transition curve for an unstable natural 1 bit.

4.3 Data Remanence Time Distribution plot for stable natural bits, at room temperature. (a) DRT Distribution plot for stable natural 0’s. (b) DRT Distribution plot for stable natural 1’s.
4.4 Correlation of DRT between two sets of experiments, at room temperature, (a) Correlation for stable natural 0’s (b) Correlation for stable natural 0’s after 6 months. (c) Correlation for stable natural 1’s. (d) Correlation for stable natural 1’s after 6 months. 55

4.5 Bitmap representation of the 2K SRAM used for the experiment, at room temperature. (a) Colored bitmap where white represents stable natural 1’s, Black represents stable natural 0’s, Red Represents unstable natural cells. (b) Binary bitmap where Black represents all natural 0’s and White represents all natural 1’s (by majority voting). 56

5.1 Percentage of Data Loss in a known Image for 59 power-up cycle at room temperature. 60

5.2 Data loss in Einstein image over time, at room temperature. 62

5.3 Data loss in black image and white image over time, at room temperature. 62

5.4 Data Loss percentage for 800 power-up cycle at seven different temperatures. (a) Data-Loss in Einstein’s image. (b) Data-Loss in white image (all 1’s). (c) Data-Loss in black image (all 0’s). 64
List of Tables

3.1 RAMCTL Registers. ............................................. 40

4.1 Findings from data remanence characterization, at room temperature. ........................................... 54

5.1 Time required for variable data loss percentage at seven different temperatures ................................ 65
Chapter 1. Introduction

1.1 Motivation

In the current era, the proliferation of connected devices is reshaping modern life, integrating Internet of Things (IoT) technologies into everyday objects—from smart home appliances to entire urban infrastructures. However, this rapid digital integration also brings critical security and data privacy issues to the forefront. The fundamental challenge lies in ensuring robust security measures that are both effective and feasible within the constrained environments of IoT devices. Unlike traditional computing systems, IoT device system often operate with limited processing power, memory, and energy resources, making it impractical to implement conventional, resource-intensive security solutions. This necessitates the development of innovative security protocols that can protect these devices against a growing array of cyber threats without compromising their functionality or performance.

The urgency of addressing hardware security at the edge is underscored by the increasing number of attacks. In 2021, over 1 billion IoT attacks were recorded [6], highlighting the escalating security concerns in this rapidly growing sector. Vulnerabilities identified span a wide range of devices, reflecting both the diversity and the pervasive security gaps present in edge device technologies. For
instance, a lack of encryption was discovered in wireless keyboards [7], making them susceptible to interception and unauthorized access. Smartwatches were found to have insufficient authentication protocols [8], potentially allowing attackers to exploit these devices. Additionally, a critical vulnerability in Amazon’s Alexa allowed retrieval of personal information [9], underscoring the risks associated with voice-activated assistants. Furthermore, the security of smart cars has been compromised, as demonstrated by a team of hackers who remotely took control of a Tesla Model S from 12 miles away, exposing serious risks to electric automated vehicle safety and user privacy [10].

A promising avenue for addressing these security challenges is the exploration of lightweight security mechanisms, particularly through the use of embedded Static Random-Access Memory (SRAM) found in most commercially available microcontroller units (MCUs) often used in edge devices [11, 12, 13]. The power-up states of SRAM cells, exploiting the inherent process variation during the fabrication process, reveal a physical fingerprint called Physical Unclonable Functions (PUFs) [14, 15, 16, 17, 18, 19, 20]. These PUFs provide a hardware-level fingerprint of the device used for secure identification. The unique and inherent embedded SRAM PUFs within the MCU can generate device authentication keys, providing a robust security solution with minimal resource overhead.

SRAM PUFs hold great promise, but their dependability across different environmental conditions and their vulnerability to advanced attacks pose considerable challenges. Securing data goes beyond just the key-storage mechanism. During regular operations, data in use is unavoidably stored in RAM without
encryption, and encryption keys must be kept accessible in memory for programs to retrieve data from the disk. Likewise, personal information such as messages, media files, and login credentials are temporarily stored in RAM.

This brings us to the critical issue of data remanence. Data remanence refers to the phenomenon where residual data remains in memory cells even after power has been removed [21]. This lingering data can pose significant security risks, especially in applications that assume volatile memory is erased immediately upon power loss. Contrary to this assumption, SRAM gradually lose their contents over a period of time that can range from milliseconds to several seconds, depending on the temperature [5] and other environmental conditions [22]. At lower temperatures, data retention times can increase significantly, exacerbating the potential for data remanence. This residual data can be exploited through cold boot attacks, where an attacker reboots a system and accesses the remaining data in RAM, or even physically transfers the memory chip to another device to extract its contents.

Embedded SRAM, as found in microcontrollers, is generally believed to be more robust against data remanence attacks compared to DRAM, primarily because it is integrated into the MCU chip die. However, further studies [23, 24, 25] have revealed that data remanence is not limited to traditional computing systems but extends to embedded systems. For instance, modern microcontrollers often used in edge devices exhibit data remanence characteristics similar to those found in larger computing systems. The increasing tendency toward edge computing, where data processing and storage are brought closer to the source of data, fur-
ther amplifies the risk. Edge devices, typically left powered and unattended in unsecured environments, are particularly vulnerable to such attacks.

Various countermeasures have been proposed to mitigate the risks associated with data remanence. One approach is to implement a firmware feature that overwrites RAM contents during the boot process in case of an abnormal shutdown [26]. However, these protections can be circumvented if an attacker has physical access to the device and can manipulate the firmware settings to disable memory overwriting. Ensuring pre-boot authentication and forcing devices to shut down or hibernate before leaving them unattended are additional strategies to enhance security [27].

Despite these measures, the susceptibility of SRAM to data remanence remains a critical concern. The phenomenon is driven by the inherent electrical properties of memory cells, where residual charges can persist in the cell’s capacitances, allowing data to be reconstructed after power loss. Temperature plays a crucial role in this process; lower temperatures reduce leakage currents and prolong the time before data fades away, making it easier for attackers to recover the information.

Additionally, the implications of data remanence extend beyond simple data recovery. Attackers gaining access to residual data can compromise cryptographic keys, user credentials, and other sensitive information, leading to unauthorized access, data breaches, and significant privacy violations. In the context of edge devices, where devices control critical infrastructure and personal environments, the potential consequences of data remanence attacks are even more
severe. Therefore, a critical understanding of the data remanence and developing robust security mechanisms to counteract it is essential for safeguarding modern digital systems.

Therefore, a fine-grained analysis of individual bitcells of embedded SRAM under different thermal conditions can be used to develop robust security measures against SRAM data remanence-based security attacks. Such an analysis helps researchers understand how individual bits in SRAM cells behave under low temperatures, addressing stability and data retention across various applications.

Furthermore, developing methodologies that incorporate writing general user data, such as images, to SRAM and monitoring retention can help identify the vulnerability of user data in terms of the stable natural power-up state bits, beyond just cryptographic keys. It is important to explore the possibility and resilience of data remanence attacks on modern SRAM technologies to develop robust security for IoT systems, while carefully choosing MCUs for these applications. Providing insights and recommendations for mitigating these vulnerabilities is of paramount importance.

1.2 Thesis Objective

The objective of this thesis is to investigate the data remanence characteristics of embedded SRAM under varying temperatures, specifically utilizing the 16-bit MSP430F5529LP microcontroller. This study aims to:
• **Evaluate Power up State of the SRAM in the MSP430F5529 microcontroller:** Conduct experiments to assess the Power-up State characteristics of the embedded SRAM of the MSP430F5529 microcontroller.

• **Establish a Framework for Stability and Remanence Analysis:** Create a framework that identifies and characterizes the data remanence characteristics of individual SRAM bits. This will involve tracking the data loss of a known pattern (a pre-selected bitmap image) written to the SRAM and observing how it changes over time and under different thermal conditions.

• **Thermal Vulnerabilities:** Provide a comprehensive understanding of how temperature variations influence the fidelity of stored information in SRAM.

By achieving these objectives, this thesis aims to contribute to the development of secure and reliable IoT devices by addressing the critical issue of data remanence in embedded SRAM, particularly under varying thermal conditions. The findings will provide valuable insights for designing more effective security mechanisms that leverage the inherent properties of SRAM for device authentication and data protection.

1.3 Thesis Outline

This thesis is organized as follows:

• **Chapter 1: Introduction.** This chapter provides an overview of IoT and the significance of embedded SRAM in these devices. It introduces the
challenges related to data security and data remanence in SRAM, setting the stage for the research objectives and the structure of the thesis.

- **Chapter 2: Background and Related Works.** In this chapter, we delve into the technical background of SRAM, discussing its architecture, power-up states, and the phenomenon of data remanence. We explore the impact of temperature on data retention in SRAM and review existing literature on related studies. This includes a survey of previous research on SRAM vulnerabilities, cold boot attacks, and the application of SRAM PUFs in IoT security.

- **Chapter 3: Experimental Flow and Methodology.** This chapter details the experimental setup and methodology used to investigate the data retention characteristics of the MSP430F5529LP microcontroller’s embedded SRAM under various low-temperature conditions. It describes the process of writing and reading data to and from the SRAM, the selection of temperatures ranging from $24^\circ C$ to $-25^\circ C$, and the procedures for tracking data retention over time. The chapter also outlines the framework established for analyzing the stability and remanence characteristics of individual SRAM bit-cells.

- **Chapter 4: SRAM Power-Up State Characterization.** Here, we discuss the characterization of the data remanence in SRAM cells. This chapter explains how we identified stable bits and analyzed their behavior during multiple power-up cycles. The results of these characterizations provide in-
sights into the inherent stability of SRAM cells and their susceptibility to environmental variations.

• **Chapter 5: Analysis of Temperature Effect on SRAM Data Remanence.** This chapter provides a comprehensive characterization and analysis of data remanence within the selected embedded SRAM sector, focusing on how temperature variations affect data retention. The methodology involves writing a known image (a pre-selected bitmap image) to the SRAM and observing how long this data remains intact over time. The analysis evaluates the fidelity of stored information across different temperatures. Further, the chapter presents results from experiments that examine the impact of temperature on the data stored in the target SRAM. These findings provide a detailed insight into the thermal sensitivity of embedded SRAM and its implications for device security in varying environmental conditions.

• **Chapter 6: Conclusion and Future Work** The final chapter summarizes the key findings of the thesis and their significance for the field of IoT security. It discusses the limitations of the current study and proposes directions for future research. This includes recommendations for improving the resilience of SRAM-based security mechanisms and exploring advanced techniques to mitigate temperature-induced data remanence vulnerabilities.
Chapter 2. Background and Related Works

In this chapter, we will discuss the fundamental concepts necessary for our research on temperature-induced vulnerabilities in embedded SRAM. First, we will go over the memory subsystem in computing devices, emphasizing memory hierarchy, and various types of Random Access Memory (RAM). Second, we will narrow our focus on the characteristics and functions of SRAM. Next, we will discuss the concept of SRAM power-up states and their application in security mechanisms, including PUFs. This background will provide a comprehensive foundation for understanding the security challenges and solutions related to SRAM in embedded systems. Following this, we will examine the phenomenon of data remanence in SRAM and explore how temperature variations impact this property. Additionally, we will review prior works related to temperature-induced data remanence attacks on SRAM and other types of random access memory.

2.1 Memory Subsystem in Computing Devices

2.1.1 Memory Hierarchy

The memory hierarchy in computing systems is designed as a pyramid to optimize data access and storage efficiency, with the fastest, smallest storage units at the top and the largest, slowest at the bottom, as illustrated in Figure 2.1. At
Figure 2.1: Memory Hierarchy of computer storage. CPU registers, Cache, and Main memory are semiconductor-based.

the peak are registers, the quickest and smallest, used by the CPU for immediate data processing. Caches are located between the CPU and main memory, arranged in levels (L1 to L3) to store frequently accessed data and speed up processing. Main memory, or RAM, provides a larger workspace for active processes and is volatile, losing its contents when powered off. Flash disks, including SSDs, offer faster access than traditional disks and retain data without power, serving as permanent storage for applications and data. Traditional disks, such as HDDs and SSDs, provide substantial storage capacity and are used for long-term data storage. At the base, remote secondary storage like cloud services offers high capacity and redundancy but is limited by network speed and latency, making it suitable for backups and archival purposes.
2.1.2 Random Access Memory

Random Access Memory (RAM) is a type of volatile memory used in computing devices for temporary data storage while the system is running. It allows data to be read and written quickly in any order. There are two primary types of RAM:

i) **Static Random Access Memory (SRAM):** SRAM uses bistable latching circuitry to store each bit, which does not require periodic refreshing, making it faster than DRAM. However, SRAM is more expensive and consumes more power, limiting its use to smaller amounts of high-speed memory, such as CPU caches.

ii) **Dynamic Random Access Memory (DRAM):** DRAM stores each bit of data in a separate capacitor within an integrated circuit. Since capacitors leak charge, the information eventually fades unless the capacitor charge is refreshed periodically. DRAM is widely used as the main memory due to its high density and relatively low cost.

2.2 Static Random Access Memory

An SRAM bit cell typically consists of six Metal Oxide Semiconductor Field Effect Transistors (MOSFETs) forming two cross-coupled inverters with access circuitry, creating a stable feedback loop that ensures the bit cell remains in one of two stable states, representing a binary ‘0’ or ‘1’. This configuration, known as a 6T SRAM cell, leverages four transistors to form the inverters and
Figure 2.2: Six-transistor SRAM memory cell: (a) 6T SRAM bit cell where P1-N1 and P2-N2 form two cross-coupled inverters and A1 and A2 are the access transistors. (b) SRAM bitcell showing two cross-coupled inverter that stores one bit data.

two additional transistors for access. An SRAM cell is illustrated in Figure 2.2. The advantage of this structure lies in its speed and the fact that it does not require periodic refresh cycles to maintain its data, unlike DRAM. However, the inclusion of six transistors per cell makes SRAM cells significantly larger than DRAM cells, limiting the capacity of SRAM-based memory for the same silicon area.

2.2.1 SRAM Operation and Architecture

The operation of an SRAM cell involves accessing the stored bit by enabling the word line, which allows the bit and its complement to be sensed and amplified by the sense amplifier during a read operation. During a write operation, the bit lines are forced to the desired state, flipping the state of the cross-coupled inverters if necessary. Figure 2.2 illustrates a typical 6T SRAM bit cell. The cross-coupled inverters are formed by four transistors: $P_1$ and $N_1$ form one inverter, while $P_2$ and $N_2$ form the other inverter. The access transistors,
denoted as $A_1$ and $A_2$, connect the cell to the bit lines during read and write operations.

In the "1" state, the $P_1$ (PMOS) transistor of the first inverter and the $N_2$ (NMOS) transistor of the second inverter are turned on. This configuration stabilizes the "1" state at node $Q$, where the voltage is high (close to $V_{DD}$). Conversely, in the second inverter, the $P_2$ transistor and in the first inverter $N_1$ transistor are turned off, ensuring that the complement node, labeled as $\overline{Q}$, is at a low voltage (close to ground). The access transistors, $A_1$ and $A_2$, remain off during the data hold state, isolating the cell from the bit lines and thus preserving the stored data.

**Figure 2.3:** Architecture of an $N \times M$ SRAM memory array.
The SRAM array architecture, as shown in Figure 2.3, consists of memory cells organized in a grid of rows and columns, with each cell storing a single bit of data. The address decoder processes the row address input, selecting the appropriate row, while the column drivers control the specific columns to access memory cells within the chosen row. During a read operation, sense amplifiers detect and amplify voltage changes representing the stored data, making it readable. The write enable signal determines the operation mode; when active, data is written into the selected cells, and when inactive, a read operation is performed. This basic array configuration with essential peripheral circuits ensures efficient and reliable data storage and retrieval in the SRAM array.

### 2.2.2 SRAM Bistability

SRAM cells exhibit bistability due to their design using cross-coupled inverters, which can hold one of two stable states as shown Figure 2.4. In the left-side configuration, the cell stores a ‘1’, while in the right-side configuration, it stores a ‘0’. This bistability is fundamental to the operation of SRAM, as it ensures that each cell can reliably store a binary bit, either 0 or 1, without any intermediate states. The bistable nature is achieved through the positive feedback
loop created by the cross-coupled inverters, which reinforces the stored state and provides stability against disturbances.

Figure 2.5: SRAM cell bistability: (a) Voltage-transfer characteristic of two cross-coupled inverters (butterfly diagram), with the left diagrams showing the individual Voltage Transfer Characteristics (VTCs). (b) Metastable state C compared to stable operation points A and B.

The robustness of the bistable states against noise and variations in power supply is quantified by the Static Noise Margin (SNM). SNM is a measure of the maximum noise voltage that can be tolerated before the cell’s state is disturbed. To quantify SNM of SRAM, we use a butterfly curve. Figure 2.5(a) illustrates the butterfly diagram for an SRAM cell.

The butterfly diagram is created by plotting the VTC of one inverter against the inverse VTC of the other inverter. The points where the curves intersect represent the stable states of the cell. The 6T SRAM cell has three possible states: A, B and C. The A and B maintains logical ‘1’ or ‘0’ as long as power supply is maintained. The point C is called metastable point. Though, in theory, an SRAM cell should be stable in this point as well, however, due to any small variation, SRAM cell may switch unpredictably into A or B from C as shown in Figure 2.5(b).
The area of the largest square that fits within the lobes of the butterfly diagram represents the Static Noise Margin (SNM). Higher SNM values indicate greater resistance to noise-induced state changes, ensuring reliable data storage. Practically, a high SNM means that the cell can withstand larger disturbances without flipping its state, which is crucial for reliable operation. However, due to inherent variations in CMOS fabrication technology, power supply voltage fluctuations, and temperature changes—collectively known as Process, Voltage, and Temperature (PVT) variations—the SNM of an SRAM cell can be significantly affected. These variations can lead to skewed or disturbed SNM values, which is the primary factor influencing the power-up states of SRAM. Before moving into this further, it is important to understand how CMOS process variations work.

2.2.3 SRAM Leakage

Refer to the Figure 2.2 again for the typical structure of a 6T SRAM bitcell, which consists of six transistors: two pull-up transistors (P1, P2), two pull-down transistors (N1, N2), and two pass-gate transistors (A1, A2). Understanding the leakage mechanisms in these transistors is important for the SRAM data remanence and cell-level stability analysis.

In the context of sub-threshold leakage, the primary contributors are the pull-down and pass-gate transistors. During the hold state, where data is retained in the bitcell without any read or write operations, the voltage across these transistors can be such that they are in the sub-threshold region. In this state, the sub-threshold leakage current becomes significant. The sub-threshold
leakage current as described by the equation:

\[ I_{sub} = I_0 \cdot e^{\frac{V_{GS} - V_{th}}{nV_T}}, \quad (2.1) \]

where \( I_0 \) is a pre-exponential factor that depends on the device parameters, including the process technology and the geometry of the transistor, \( n \) is the sub-threshold swing coefficient, and \( V_T \) is the thermal voltage.

For instance, if the gate-to-source voltage \( V_{GS} \) is below the threshold voltage \( V_{th} \), yet not zero, a weak current flows through the channel of the pull-down transistors (N1 and N2) and the pass-gate transistors (A1 and A2). The sub-threshold slope (SS) indicates the efficiency of turning off the transistor. The sub-threshold slope (SS) defined as the equation:

\[ S = \frac{d \log I_D}{dV_G} \approx \frac{V_T}{n} \log 10, \quad (2.2) \]

where we see that the sub-threshold slope is influenced by the gate oxide capacitance \( C_{ox} \) and the depletion capacitance \( C_{dm} \). The \( C_{ox} \) and \( C_{dm} \) can be determined by:

\[ m = 1 + \frac{C_{dm}}{C_{ox}}, \quad (2.3) \]

where \( m \) is called the body factor.

In a typical SRAM cell, the pull-down transistors (N1, N2) are designed with careful consideration of their threshold voltage and body factor to minimize leakage. However, due to scaling and the need for higher density, achieving an
ideal low leakage current is challenging. The pass-gate transistors (A1, A2), which control access to the stored data during read and write operations, also contribute to leakage when in the sub-threshold region. The pull-up PMOS transistors supply current to the output node and restore its voltage lost by the leakage of NMOS transistors during hold operation. When power is turned off, the PMOS transistors no longer supply current and output node voltage gradually decays depending on the leakage current magnitude.

However, as we see in Equation 2.1 and Equation 2.2, the subthreshold leakage is fundamentally proportional to the temperature. Hence, lower temperature slows down the leakage which increases the chances of higher data remanence.

### 2.2.4 Effects of Process Variation

Process variations during semiconductor manufacturing can lead to discrepancies in the electrical characteristics of the transistors within an SRAM cell. These variations, inherent to the fabrication process, can significantly affect the performance and reliability of SRAM, often playing a crucial role in determining whether the SRAM cell leans towards point A or point B from the metastable point in the butterfly curve.

During semiconductor manufacturing, variations in parameters such as channel length, oxide thickness, doping concentrations, and other factors can result in non-uniformities across different transistors. These variations can be categorized into two main types: inter-die (die-to-die) and intra-die (within-die)
Figure 2.6: Illustration of the spatial process-variation hierarchy: (a) lot-to-lot, (b) wafer-to-wafer (c) chip-to-chip, (d) within-die variations, and (e) typical CMOS transistor device parameter space and performance space with corresponding worst-case corners [1].
variations. Inter-die variations refer to differences between different dies on a wafer, while intra-die variations refer to differences within a single die.

One of the primary effects of process variation is the alteration of the threshold voltages ($V_{th}$) of the transistors. Variations in $V_{th}$ can cause some transistors to turn on or off at different voltages than intended, leading to mismatches in the cross-coupled inverters of the SRAM cell. Another significant impact of process variations is on the current-driving capabilities of the transistors. Variations in channel length and doping concentrations can cause differences in the drive current, leading to asymmetric behavior in the cross-coupled inverters. This asymmetry can reduce the Static Noise Margin (SNM) of the cell, making it less robust against noise-induced state changes. Lower SNM values increase the likelihood of the cell flipping states under the influence of noise or power fluctuations, compromising data integrity.

Figure 2.6(a)-(d) illustrates the spatial process-variation hierarchy in semiconductor devices, including lot-to-lot, wafer-to-wafer, chip-to-chip, and within-die variations. These variations can be systematic (spatially correlated) or random (spatially uncorrelated). Systematic variations create predictable patterns across the die, while random variations lead to unpredictable discrepancies between transistors. Additionally, the Figure 2.6(e) shows the typical CMOS transistor device parameter space and performance space with corresponding process corners. Here, TT means both NMOS and PMOS are performing at their typical speed, FF means both of them are fast, and SS means both of them are slow. SF and FS are the corners where either of them is slower and other one is faster.
2.3 SRAM Power-up States

Upon powering up, the state of an SRAM cell can be somewhat indeterminate due to random fluctuations and variations in the threshold voltages of the transistors. However, once the power stabilizes, the feedback loop formed by the cross-coupled inverters forces the cell into one of its stable states (0 or 1), depending on the slight imbalances induced by the PVT variations.

After powering up, initially at Phase-1, as $V_{DD}$ is turned on, the difference in PMOS ($P_1$ and $P_2$) $V_{th}$ decides which capacitor at node $Q$ or $\bar{Q}$ get the charges faster. At Phase-2, the node that get charges faster pulls higher current, decides which NMOS ($N_1$ or $N_2$) turns ON. The current distribution of different power-up transient phases are depicted in Figure 2.7 [2]. The power-up state of an SRAM cell is thus determined by the slight asymmetries in the cell design and the inherent process variations. These factors cause each SRAM cell to power up into a state that is consistent across power cycles, providing a unique and...
reproducible pattern. This property is utilized in various security applications, such as generating physical fingerprints for device authentication.

When the device is powered on, the initial states of the SRAM cells are read, capturing the inherent randomness present in the initial states. This random pattern can be used as a source of entropy for various applications, including secure key generation and device authentication. The power-up state of an SRAM cell is therefore an essential feature that leverages the physical properties of the memory cells to enhance security and reliability in modern computing systems.

2.3.1 SRAM Power-up States as a Physical Fingerprint

The power-up state of an SRAM cell can serve as a unique physical fingerprint for a device due to inherent process variations during semiconductor manufacturing. These variations cause slight differences in the electrical properties of transistors, resulting in unique and consistent power-up states. When powered on, an SRAM cell randomly initializes to a stable state, either logic 0 or 1, influenced by variations in threshold voltages, channel lengths, and other manufacturing parameters. This randomness is reproducible, meaning each power cycle initializes the cells to the same states, creating a unique pattern.

This pattern can be captured and used as a “silicon fingerprint” or “chip fingerprint,” making it ideal for security applications. For instance, Figure 2.8 shows an example of a 64-bit fingerprint extracted from a larger SRAM fingerprint, with the background highlighting the overall pattern. The shading of each cell in the figure indicates the probability (p) of powering up to ‘1’, as measured
Figure 2.8: Example of a 64-bit fingerprint extracted from a larger SRAM fingerprint, shown in the background. The shading of each cell indicates the probability ($p$) of powering up to ‘1’, as measured over 100 trials [3].

over 100 trials. The identifier derived from the power-up state enhances security by being extremely difficult to clone or spoof. The readout process occurs immediately after powering on, capturing the initial state before any data is written. This unique identifier is then used in secure boot processes, device authentication, and anti-counterfeiting measures.

2.3.2 Physical Unclonable Functions (PUFs)

PUFs exploit the unique power-up states of SRAM cells to generate hardware-level fingerprints for secure device authentication [28, 29, 30, 31]. PUFs leverage the inherent manufacturing variations in SRAM cells to create a unique and reproducible response for each device, making it nearly impossible to replicate or clone.
A PUF works by taking advantage of the process variations that occur during the semiconductor fabrication process. These variations result in slight differences in the electrical properties of the transistors within each SRAM cell, causing each cell to initialize to a random but stable state when powered on. This randomness is due to variations in parameters such as threshold voltage, channel length, and doping concentration.

When an SRAM-based PUF is activated, the device is powered on, and the initial states of the SRAM cells are read. This power-up state forms a unique binary pattern of 0s and 1s, which serves as the device’s PUF response. This response is highly repeatable under the same conditions, ensuring that the same pattern is generated each time the device is powered on.

The uniqueness of the PUF response is critical for its use in secure device authentication. Each SRAM-based PUF response is unique to the individual device, thanks to the random variations introduced during manufacturing. This uniqueness means that even devices produced in the same manufacturing batch will have different PUF responses, providing a robust method for distinguishing
between different devices. Figure 2.9 illustrates the concept of an SRAM-based PUF. The figure shows how the power-up states of SRAM cells, influenced by manufacturing variations, generate a unique and stable response that can be used for authentication purposes.

PUFs are used in various security applications, including:

- **Secure Boot**: During the boot process, the device can generate its PUF response and compare it to a stored reference response. If the responses match, the device is authenticated, and the boot process continues. This ensures that only authorized devices can boot and operate.

- **Device Authentication**: PUFs provide a means for secure authentication without the need for storing sensitive cryptographic keys in non-volatile memory. When a device needs to authenticate itself to a network or another device, it generates its PUF response, which is then verified against a reference response stored during an initial enrollment phase.

- **Anti-Counterfeiting**: PUFs can be used to verify the authenticity of hardware components [32]. Manufacturers can generate and store the PUF responses of genuine components. At any point in the supply chain, the components can be authenticated by comparing their PUF responses to the stored references, ensuring that counterfeit components are detected and rejected.

The security of PUFs relies on the difficulty of replicating the manufacturing variations that produce the unique PUF responses. Unlike traditional
cryptographic keys stored in non-volatile memory, PUF responses are not susceptible to physical attacks aimed at reading stored data. Instead, the PUF response is generated on-the-fly, making it much harder for attackers to extract or clone the response. However, SRAM PUF can be affected by various factors which can be leveraged by attackers to exploit the cryptographic keys [2, 33].

2.4 Data Remanence in SRAM

Data remanence refers to the residual data that remains in a memory device even after the power supply has been removed or the data has been erased. In this section, we discuss how data remanence occurs in SRAM and the effect of temperature on SRAM data remanence.

2.4.1 Governing Equations of Data Remanence in SRAM

Data remanence in SRAM cells is characterized by the persistence of data after the power has been cut off. This phenomenon arises due to the non-instantaneous discharge of the parasitic capacitances within the SRAM’s transistor structures, enabling data to remain accessible briefly post power-off.

Figure 2.10 illustrates the data remanence mechanism in an SRAM cell. The node capacitance discharges through the $P_1$, $N_1$, and access transistor $A_1$. The leakage current ($I_{\text{leak}}$) plays a crucial role in this process, defined by the equation:

$$I_{\text{leak}} = \frac{\Delta Q}{\Delta t} = C \frac{\Delta V}{\Delta t},$$  \hspace{1cm} (2.4)
Figure 2.10: SRAM cell set to "1" state with P1 and N2 transistors active, P2 and N1 inactive. Access transistors remain off during data hold. If powered off, node Q discharges via N1, P1, and A1 due to transistor leakage.
where $C$ represents the capacitance, and $\Delta V/\Delta t$ indicates the rate of voltage change across the capacitor.

The leakage current is given by,

$$I_{\text{leak}} \propto e^{\frac{0-V_{\text{th}}}{\eta \left( \frac{k_{\text{B}} T}{q} \right)}},$$

(2.5)

where we see that it is exponentially dependent on the threshold voltage ($V_{\text{th}}$) relative to the thermal voltage ($V_T$), which is a function of the temperature ($T$) and Boltzmann’s constant ($k_B$).

The data remanence time ($t_{\text{rem}}$), or the period during which data remains recoverable, can be derived as:

$$t_{\text{rem}} = \frac{C \Delta V}{I_{\text{leak}}} \propto C \times (V_{dd}) \times e^{\frac{qV_{\text{th}}}{k_{B} T}},$$

(2.6)

which demonstrates that the remanence time is contingent upon the supply voltage ($V_{dd}$), the capacitance of the cell, and the thermal stability influenced by the temperature.

To capture and analyze the behavior of SRAM under power-down conditions, we utilized the 6T SRAM cell configuration, comprising cross-coupled inverters that maintain data stability as long as power is supplied. The associated parasitic capacitances and the leakage currents through the transistors determine the rate at which the data integrity degrades post-power-off.
An example of memory decay analogous to SRAM in DRAM is illustrated in Figure 2.11, where the parasitic capacitances discharge over time, leading to a loss of stored data once the voltage falls below a critical threshold.

### 2.4.2 Effect of Temperature on Data Remanence

Temperature plays a significant role in the data remanence characteristics of SRAM. The retention time of data stored in SRAM cells after power-off is heavily influenced by the temperature at which the memory is kept. This is primarily because temperature affects the leakage currents in the transistors that make up the SRAM cells.

Leakage currents in transistors, such as subthreshold leakage and gate oxide leakage, increase with temperature. At higher temperatures, these leakage currents become more significant, causing the residual charge in the SRAM cells’ parasitic capacitances to dissipate more quickly. Consequently, the data retention time at elevated temperatures is shorter, as the stored charge decays rapidly, leading to quicker data loss.

Conversely, at lower temperatures, the leakage currents are significantly reduced. The parasitic capacitances in the SRAM cells discharge more slowly, prolonging the retention time of the stored data. This extended remanence time at lower temperatures means that the data can remain for a longer period after power-off, increasing the window of vulnerability for potential data recovery.

In SRAM, elevated temperatures increase leakage currents and destabilize the bistable states of the memory cells, leading to a higher likelihood of data loss.
Conversely, lower temperatures reduce leakage currents, potentially extending data retention times. As we know from the relationship of temperature and thermal voltage,

\[ V_T = \frac{kT}{q}, \]

where \( k \) is the Boltzmann constant, \( T \) is the absolute temperature, and \( q \) is the charge of an electron. The exponential relationship between subthreshold leakage and thermal voltage discussed earlier indicates that as temperature increases, thermal voltage increases, hence, subthreshold leakage also increases, resulting in decreased data retention time. In contrast, as temperature decreases, subthreshold leakage decreases, leading to longer data retention times and prolonged data remanence.

2.5 Related Work

Low-temperature attacks on Random Access Memory were first widely discussed in [5], where the data remanence phenomenon in DRAM was explored. In DRAM, data remanence occurs due to the slow discharge of the charge stored in the capacitors. These capacitors lose their charge gradually, resulting in a decay of the stored data over time. The rate of decay depends on several factors, including temperature, the initial charge level, and the intrinsic properties of the capacitors.

Figure 2.11 illustrates how memory decay occurs over time in DRAM, where the voltage levels at the nodes decrease until they fall below the threshold needed to maintain the state. This study demonstrated that during a low tem-
Figure 2.11: Visualization of DRAM memory decay from [5]. A bitmap was loaded, and power was cut off for intervals of 5 seconds, 30 seconds, 60 seconds, and 5 minutes at room temperature. The degradation shows horizontal bars from alternating ground states and vertical bars from physical chip variations.

Temperature start-up (cold boot), DRAM retains data for several seconds, thereby limiting the operating system’s ability to protect cryptographic keys from attackers with physical access. In a cold boot attack, an attacker can rapidly cool the DRAM memory module, using materials like liquid nitrogen or dry ice, to significantly reduce its temperature. By doing so, the attacker can extend the retention time of the data, allowing them to extract the residual charge before it decays completely. This process is depicted in Figure 2.12, which shows the steps involved in a cold boot attack, including the cooling of the memory and the subsequent data extraction. While cold boot vulnerabilities in DRAM have been extensively studied and subsequent research has shown data remanence on off-the-shelf SRAM chips [34], on-chip SRAM has generally been considered more robust due to its inherent design, smaller memory size, and integration within SoCs or microcontrollers.
Recent studies, however, have highlighted vulnerabilities in on-chip SRAMs. For example, Volt Boot [24] exploits asymmetrical power states to force SRAM state retention across power cycles, bypassing traditional cold boot attack enablers such as low temperature or intrinsic data retention time. Further research has revealed significant data remanence in SRAM at low temperatures. A comprehensive study on SRAM PUFs indicated that data remanence could persist at -40°C [23], suggesting that cold temperatures could be used to retrieve cryptographic keys from SRAM PUFs, thus challenging the assumed security of these systems under extreme conditions. However, this research primarily focused on intrinsic SRAM PUFs on commercial off-the-shelf devices within a narrow temperature range of -40°C to -110°C. Additionally, the study emphasized security implications, neglecting general data retention and stability issues in SRAM cells under varying temperatures and focusing only on short-term data remanence with power-off times of 10ms and 20ms. Finally, [25] investigated SRAM data remanence on several older microcontrollers (MCUs), including the Freescale MC68HC908AZ60 and MC68HC908AZ60A, and the Texas Instruments
MSP430F112 and MSP430F427, fabricated with 0.9- to 0.35-µm processes. While valuable, this study’s methods are outdated, and it did not examine specific data types like uniform binary patterns or image data, limiting its applicability. Therefore, it provides remanence data only for these particular MCUs, highlighting the need for updated research with modern technologies and more comprehensive data pattern analysis.

In contrast, our paper extends the analysis to a broader temperature range and provides a detailed bitwise assessment of SRAM vulnerabilities, which is crucial for understanding the granularity of these vulnerabilities. This comprehensive approach reveals how individual bits in SRAM cells behave under different thermal conditions, addressing general stability and data retention across various applications. Furthermore, our methodology includes writing known images to SRAM and monitoring retention over extended periods, thereby capturing long-term stability and potential failure modes. Moreover, our study applies this analysis to modern SRAM technologies, providing insights and recommendations relevant to the latest generation of SRAM devices, thereby mitigating temperature-induced vulnerabilities.
Chapter 3. Experimental Setup and Methodology

This chapter discusses the experimental setup and methodology employed to explore the data remanence characteristics of embedded SRAM, particularly under varying thermal conditions. Initially, our experimental approach focuses on characterizing the power-up state of SRAM using the MSP430F5529 microcontroller. Following this, we examine how these states are affected by cold temperatures, utilizing a controlled environment provided by the SO-LOW FDC-4000 Freezer which enables us to systematically investigate the effects of low temperatures on the retention capabilities of SRAM.

Figure 3.1: The Experimental Setup.
3.1 Setup

The complete experimental setup is simply illustrated in Figure 3.1. The core of our experimental setup is the 16-bit MSP430F5529 microcontroller from Texas Instruments [35], integrated into MSP-EXP430F5529LP MSP430F5529 LaunchPad™ development board [36]. Figure 3.2(a) illustrates the development board, MSP430F5529 LaunchPad™, and Figure 3.2(b) shows the MSP430F5529 microcontroller. This board offers a supply voltage range of 1.8V to 3.3V through a USB-connected, high-efficiency DC/DC converter, ensuring stable power delivery to the microcontroller during testing. The development board facilitates computer interfacing via USB 2.0, and the MCU itself is furnished with 128KB of Flash storage and 8KB of SRAM. Our experiments specifically targeted the 2KB of general-purpose USB SRAM found in sector 7, as indicated in the microcontroller’s datasheet.

The LaunchPad™ board supports various low power modes, crucial for managing the SRAM’s power states effectively. By engaging the LPMx.5 low power mode, the board’s Low Dropout (LDO) circuit is disabled, disconnecting the MCU core from the supply voltage and thereby clearing the SRAM contents, a necessary step for accurate testing of data remanence. Software development and debugging were conducted using Code Composer Studio, which serves as our Integrated Development Environment (IDE) [37].
3.1.1 The MSP430F5529 Microcontroller (MCU)

The MSP430F5529 microcontroller, part of Texas Instruments’ MSP430 family, is designed for low-power and high-performance applications, making it an ideal choice for our experimental focus on SRAM data remanence. This microcontroller features a 16-bit MCU built on RISC architecture that offers a balance between low power consumption and efficient processing, capable of operating at speeds up to 25 MHz. Key features include 128 KB of Flash memory and 8 KB of SRAM, distributed over various sectors, providing substantial capacity for both program storage and volatile data handling. The general-purpose SRAM which is accessible via USB is one of the crucial features of this MCU for tests involving frequent read/write operations and power cycling. The device supports
Figure 3.3: The Architecture for 16-bit MSP430F5529 microcontroller highlighting the SRAM memory subsystem that we utilized for our research.

extensive low-power modes, particularly the LPMx.5 states, which are critical for our experiments as they facilitate full power-off conditions for the power cycling experiments conducted on the SRAM. Additionally, the MSP430F5529 is equipped with various digital and analog peripherals, including multiple timers, a 12-bit analog-to-digital converter, and several communication interfaces like UART, SPI, and I2C, allowing for versatile interfacing options. Its USB functionality, integrated into the MCU with a dedicated oscillator to ensure precise clocking, supports various USB modes, enhancing its utility in systems requiring reliable data transfer under stringent power constraints.
3.1.2 MSP430F5529 Memory Architecture

Figure 3.4 shows the detailed architecture of MSP430F5529 [36]. The MSP430F5529 MCU is engineered with a detailed and segmented memory configuration, specifically tailored to facilitate efficient power management and data integrity in embedded systems. This MCU contains 128KB of flash memory and 8KB of SRAM, the latter of which is segmented to support specific operational needs and power conservation strategies.

The 8KB of SRAM is partitioned into multiple sectors, each with designated roles, enhancing the microcontroller’s ability to perform various tasks simultaneously without compromising on performance. These sectors are typically used for storing temporary data, stack operations, and buffer storage for peripheral interfacing. Specifically, Sector 7, which we utilize in our experiment, consists of 2KB of general-purpose SRAM. This sector is crucial for our tests as it allows us to observe the behavior of SRAM under different power conditions and temperature settings, providing a focused area for detailed data remanence studies.

One of the significant features of the MSP430F5529 is its support for various low-power modes, which are critical for applications that require minimal power consumption without sacrificing functionality. These include multiple low-power modes (LPM0 to LPM4), with each level offering a deeper sleep state to conserve more power. Notably, the LPMx.5 modes are especially relevant to our studies because they allow for turning off specific SRAM sectors. Engaging these
modes can disconnect the power supply to the SRAM, thus effectively simulating a cold boot scenario.

### 3.1.3 MSP430F5529 LPMx.5 Mode

The MSP430F5529 microcontroller’s comprehensive range of low-power modes, including the particularly sophisticated LPMx.5 modes, plays a crucial role in managing power consumption and simulating various power-off scenarios that are central to studies of data remanence and system resilience. These modes are designed not only to minimize energy usage but also to control the operational state of the microcontroller’s memory and processing units effectively.

- **Core Voltage Supply Disconnection:** The most profound feature of LPM4.5 mode is the complete disconnection of the core voltage supply. This disconnection results in the clearing of all volatile memory contents, including the general-purpose SRAM, effectively simulating a total power loss scenario. This feature is crucial for experiments aimed at understanding the power-up states of SRAM when the microcontroller transitions back from such a deep sleep state.

- **Selective Data Retention and Power-Up States:** While LPM4.5 results in a complete data wipe from SRAM, LPM3.5 allows for the retention of data in specific memory sections, such as backup memory, which is not powered down. This selective retention is pivotal for applications and experiments that need certain data to persist through power cycles, thus enabling a controlled study of data remanence in embedded systems.
Table 3.1: RAMCTL Registers.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Acronym</th>
<th>Register Name</th>
<th>Type</th>
<th>Access</th>
<th>Reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>RCCTL0</td>
<td>RAM Controller Control 0</td>
<td>Read/write</td>
<td>Word</td>
<td>6900h</td>
</tr>
<tr>
<td>00h</td>
<td>RCCTL0_L</td>
<td>Read/write</td>
<td></td>
<td>Byte</td>
<td>00h</td>
</tr>
<tr>
<td>01h</td>
<td>RCCTL0_H</td>
<td>Read/write</td>
<td></td>
<td>Byte</td>
<td>69h</td>
</tr>
</tbody>
</table>

- **Transition to Power-Up States:** Importantly, both LPM3.5 and LPM4.5 support controlled transitions back to active power-up states. This transition capability is essential for mimicking real-world scenarios where devices may need to recover quickly and resume normal operation after being in a deep sleep or powered-off state. The ability to effectively manage these transitions is facilitated by the LPX mode, which allows the device to maintain low-power consumption while being ready to wake up swiftly and restore previous operational states.

The RAMCTL (RAM Controller) plays an integral role in managing the low-power modes of the SRAM. The RAMCTL provides access to different power modes, including a retention mode that reduces leakage current while the CPU is off. In this mode, the RAM content is preserved. Alternatively, the RAM can be switched off completely by software, leading to the loss of RAM content. The RAM is partitioned into sectors, typically of 2KB size, each controlled by the RAMCTL Control 0 register (RCCTL0). The RAMCTL register is protected with a key to ensure secure modification. Table 3.1 details the RAMCTL registers:
We utilize the RAMOFF mode, wherein each RAM sector can be turned off independently by setting the respective RCRSyOFF bit in the RCCTL0 register. This flexibility allows for detailed control over the RAM’s power states, essential for optimizing energy efficiency and studying the effects of power cycling on data retention.

### 3.1.4 MSP430F5529 UART Communication

The UART channel in the MSP430F5529 plays a critical role in the setup where it is used to interface with a workstation via a serial port to handle data during the microcontroller’s power-up sequences. The experimental protocol involves the following process:

- **Reading and Transmitting Data**: During the power-up sequences, the microcontroller reads data from SRAM sector S7, which is designated for experiments due to its typical non-utilization in standard program operations. The UART is actively used to transmit this data to a connected workstation.

- **Cycle Management**: The experiment involves toggling the power state of sector S7—turning it off for the desired amount of time and then turning it back on—to observe the effects on data remanence and recovery.

- **Data Handling and Analysis**: Data captured via UART is logged using the PuTTY application on the workstation and further analyzed through Python scripts. This handling is vital for evaluating the Physical Unclonable
Functions (PUFs) derived from the power-up states, which later extended to data

- **Data Handling and Analysis:** Data captured via UART is logged using the PuTTY application on the workstation and further analyzed through Python scripts. This handling is vital for evaluating the Physical Unclonable Functions (PUFs) derived from the power-up states, which later extended to data remanence experiments.

By carefully managing the power states of specific SRAM sectors and maintaining uninterrupted UART operations, the experimental design effectively captures the dynamics of SRAM behavior under simulated power cycling conditions. This methodology is not only robust but also replicable, providing a solid foundation for further research into memory stability and security applications of SRAM in embedded systems.

### 3.2 Methodology

In our experimental methodology, we first power up the SRAM and write data to the 2KB SRAM sector. Following this, we power down the SRAM, wait for time $t_d$ and power up the SRAM again. Finally we read the SRAM data through UART. The basic methodology has been shown in Figure 3.4

We have implemented our experimental procedures in embedded C, designed to investigate the SRAM behavior of the MSP430F5529 MCU during power cycling. These procedures are detailed in two algorithms.
Algorithm 1: SRAM Power Up State Analysis for MSP430F5529 MCU

1: **Initialization:**
2: Define MSP430F5529 SRAM Sector 7 Address
3: Set up UART on USCI_A1 for 115200 baud, 8N1 configuration
4: Define required total time, \( t_{\text{cycle}} = 59ms \)
5: **for** In Range of \( (t_{\text{cycle}}) \) **do**
6: Stop Watchdog Timer: \( \text{WDTCTL = WDTPW + WDTHOLD} \)
7: Power down SRAM: \( \text{RCCTL0 = RCKEY | RCRS7OFF} \)
8: Introduce a dynamic delay, increasing each cycle: \( \text{dynamicDelay(t)} \)
9: Power up SRAM: \( \text{RCCTL0 = RCKEY} \)
10: Transmit SRAM data over UART
11: \( t = t + 1ms \) \( \triangleright \) Increment delay for the next cycle
12: **end for**
13: **Procedure End:**
14: Continuously repeat the above steps for each temperature setting

Algorithm 2: SRAM Data Remanence Analysis for MSP430F5529 MCU

1: **Initialization:**
2: Define temperature range from 25\(^\circ\)C to −25\(^\circ\)C with steps of 5\(^\circ\)C
3: Define MSP430F5529 SRAM Sector 7 Address
4: Set up UART on USCI_A1 for 115200 baud, 8N1 configuration
5: **for** each temperature in the defined range **do**
6: Define total experiment cycle time, \( t_{\text{cycle}} \)
7: Define time delay, \( t_d \)
8: **for** In Range of \( (t_{\text{cycle}}) \) **do**
9: Stop Watchdog Timer: \( \text{WDTCTL = WDTPW + WDTHOLD} \)
10: Write 1/0/Hexadecimal bitwise data for image to SRAM sector 7
11: Power down SRAM: \( \text{RCCTL0 = RCKEY | RCRS7OFF} \)
12: Introduce a dynamic delay, increasing each cycle: \( \text{dynamicDelay(t)} \)
13: Power up SRAM: \( \text{RCCTL0 = RCKEY} \)
14: Transmit SRAM data over UART
15: \( t = t + t_d \) \( \triangleright \) Increment delay for the next cycle
16: **end for**
17: **end for**
18: **Procedure End:**
19: Continuously repeat the above steps for each temperature setting
Algorithm 1 delineates the method for analyzing the SRAM’s Power Up state. This involves systematic power-down and power-up cycles at increments of 1ms, totaling up to 100ms per cycle, to monitor how the random SRAM power-up state data evolves immediately after power restoration. The UART setup transmits the SRAM data at each cycle, allowing us to capture and analyze the transient states immediately following power restoration.

Algorithm 2 addresses the SRAM Data Remanence Analysis, wherein we conduct a more extensive study to understand how data persistence varies with temperature changes ranging from 25°C to -25°C, in decrements of 5°C. Here, each temperature setting involves repeated cycles of writing data which involves writing 0 or writing 1 or writing an image hexadecimal data to all the bitcells of sector 7, powering down, introducing variable delays, and then powering up to read back the data. This helps in quantifying the data remanence characteristics under different thermal conditions. The dynamic delay introduced in each
cycle is adjusted based on the total experiment cycle time, $t_{cycle}$, defined for each temperature sample for the extended period of that specific temperature.

Both algorithms utilize the MSP430F5529’s UART for data transmission, ensuring that all observed states are accurately logged for subsequent analysis. This dual approach, combining immediate power-up observations with longer-term remanence studies under varied temperatures, provides a comprehensive understanding of the SRAM’s behavior under different operational and environmental conditions.

3.2.1 Initial SRAM Value Acquisition and Data Transmission

The foundation of our experimental methodology involved the precise acquisition of SRAM contents at power-up. This was achieved through custom assembly language routines specifically programmed into the MSP430F5529 microcontroller. These routines were designed to capture the initial state of the SRAM immediately upon power-up, ensuring that no pre-existing software operations could alter the data before it was recorded. Following the capture, a custom bootloader was employed to transmit these values to a host system via a simulated UART over USB connection. This step was crucial for establishing a reliable baseline of SRAM contents for subsequent analysis.

3.2.2 SRAM Initialization and Power Cycling Simulation

To simulate real-world scenarios and study the SRAM’s data retention characteristics under various power cycling conditions, the embedded SRAM on
the Texas Instruments bootloader was populated with specific bit patterns. This methodical population of the SRAM allowed us to assess how different data patterns influenced the SRAM’s ability to retain information under thermal stress and power cycling. The test unit, comprising two identical MSP430F5529 microcontrollers, was pre-frozen according to established research protocols [38] to ensure consistent conditions across all experiments.

### 3.2.3 Bit Stability and Retention Analysis

Our methodology for identifying stable bits involved defining SRAM cells that consistently retained their logic state across multiple power cycles at room temperature. To achieve this, all SRAM cells were initially set to a logic ‘0’. This was followed by a controlled power-down sequence and a variable delay period that ranged from 1ms to 60ms before powering up again. It was observed that the majority of bits reached a stable logic state within this 60ms window. We meticulously cataloged the indices of bits that remained stable at ‘0’ and ‘1’ and then subjected them to further tests by inverting their states to evaluate their resilience and ability to revert to their original stable states.

### 3.2.4 Image Retention and Temperature Variation Tests

For a more detailed analysis of SRAM behavior under varied thermal conditions, a binary image was written to the SRAM, and its retention was monitored. At room temperature, approximately 77% of the image was retained till 19ms at room temperature (25°C), establishing a baseline for colder temperature
experiments. Subsequent tests were conducted at 0°C, −10°C, and −25°C, following a similar protocol of pre-freezing the MCU and meticulously controlling the freezer’s temperature settings. The experimental setup, including the placement of the board within the freezer and the stabilization of temperature before data collection, is illustrated in Figure 3.2.

3.2.5 Enhancements in Data Collection and Temperature Measurement

To refine the data collection process and enhance the accuracy of temperature measurements, an external temperature sensor was incorporated alongside the LaunchPad board’s integrated sensor. This dual-sensor approach allowed for more precise control and monitoring of the thermal environment, ensuring that the observed data remanence patterns were solely attributable to the experimental conditions and not influenced by external thermal fluctuations. This rigorous control was maintained throughout the testing sessions to validate the reliability of our findings.

Through this comprehensive methodology, we were able to systematically explore the SRAM’s behavior under varying conditions, leading to significant insights into the physical properties of embedded SRAM memory and its potential vulnerabilities in real-world applications.
Chapter 4. SRAM Data Remanence Characterization

This chapter explores the bit-by-bit data remanence characterization of embedded SRAM cells, focusing on their behavior at room temperature and over extended periods. Understanding the data remanence is critical for evaluating the security and reliability of SRAM-based systems.

4.1 Identification of Natural 0’s and 1’s

Upon powering up, once the power stabilizes, the feedback loop formed by the cross-coupled inverters in the SRAM cell will force the cell into one of its stable logic states ("0" or "1"). The group of cells that predominantly settle to a logic-1 state upon power-up is referred to as "natural one" cells. Conversely, the group of cells that predominantly settle to a logic-0 state upon power-up is referred to as "natural zero" cells.

We perform a bit-by-bit analysis to classify SRAM bits into three categories as follows: stable natural 1’s, stable natural 0’s and unstable bits, based on the logic state found in the read values upon powering up the SRAM. Figure 4.1 shows the classification method for these three categories. The unstable bits defined as those bits that flip their state during consecutive powering up state before settling down into one logic value. Whereas the stable natural 0’s
Figure 4.1: The classification method for defining natural bits and unstable bits.

and stable natural 1’s are the bits which remains stable throughout the entire 59 power-up cycles at the same logic value either 0 or 1 respectively. These strongly skewed bits are stable PUF bits because random thermal noise is not sufficient to change their states on consecutive power-up read operations.

In this data remanence characterization we checked the bit transtition behaviour among all of the bit-indices for the entire 2KB SRAM after writing the logic value 1 or 0 to each cell and powering up the SRAM for 59 cycles. As it takes some time to get stabilize the values in those cells due to the effect of data remanence, we checked the data remanence time and bit transition behaviour during the 59 power up cycles for each bit index. Figure 4.2 presents a comprehensive view of the stable and unstable natural bit transitions behaviour upto 59 power-up cycles. Figure 4.2(a) and (c) illustrate the transtition behavior of a sta-
Figure 4.2: Bit value transition behaviour for SRAM power up state, at room temperature. (a) Bit transition curve for a stable natural 0. (b) Bit transition curve for a stable natural 1. (c) Bit transition curve for a unstable natural 0 bit. (d) Bit transition curve for a unstable natural 1 bit.
ble natural 0 and an unstable natural 0 while initially written as 1. Figure 4.2(b) and (d) illustrate the transition behavior of a stable natural 1 and an unstable natural 1 while initially written as 0. The time it requires to get a bit cell stabilized on its natural logic state indicates the data remanence time (DRT) on that cell. In Figure 4.2(a) the DRT has been observed close to 24ms. For unstable natural bits DRT has been considered following the very first transition time and the logic state has been decided by the majority voting for the entire 59 cycles power up state values. As example, Figure 4.2(c) showing a DRT of 31 ms.

This initial bit-wise assessment is pivotal for understanding the inherent stability of the memory array and sets the stage for further analysis of data remanence and vulnerability to data remanence attacks. However it is found that 63% of bit-cells are stable natural 1’s and 24% of bit-cells are stable natural 0’s. 13% of the bit-cells are either unstable natural 1’s or unstable natural 0’s.

4.2 Remanence of Natural 0’s and 1’s

From the bit-transition behaviour analysis we have selected all stable natural bit-cells first in separate groups as stable natural 0’s and stable natural 1’s. We wrote logic ‘1’ on all stable natural 0’s bit-cells and conversely logic ‘0’ on all stable natural 1’s bit-cells. As minimum data remanence time (DRT) is required to have a logic transition in the bit cells from its initial logic value to its stable natural state, we listed that remanence time (DRT) for each bit index for 59 power up cycles.
Figure 4.3: Data Remanence Time Distribution plot for stable natural bits, at room temperature. (a) DRT Distribution plot for stable natural 0’s. (b) DRT Distribution plot for stable natural 1’s.
Figure 4.3 illustrates the DRT for all stable natural bit-cells which stabilize to either ‘0’ (Figure 4.3(a)) or ‘1’ (Figure 4.3(b)) among 2KB memory cells. The distribution is following probability distribution function. In the context of the provided plot, the probability density function (PDF) is used to describe the distribution of the DRT in millisecond time unit, for bit-cells in a SRAM memory array. The plot illustrates how frequently different DRT occur within the observed data set.

In Figure 4.3 X-Axis (DRT in ms) represents the time it takes for the data in the memory bits to flip or lose their retention state due to remanence effect in bit cell. In this case, it ranges from 15 to 35 milliseconds. Y-Axis (Density) represents the probability density, which indicates the relative likelihood of the DRT occurring within the specified range. The area under the curve represents the probability of the retention times falling within that range. Mean 24.11 ms (Figure 4.3(a)) is the average DRT for the stable natural ‘0’ bits in the SRAM memory array. It indicates that, on average, the written data remain in these bit-cells for about 24.11 milliseconds before loosing to it’s natural logic state. A standard deviation of 3.53 ms indicates that most of the DRT fall within a range of about 3.53 milliseconds around the mean value 24.11 ms (21 to 27 milliseconds approximately). For all cells which gets stabilize to logic state ‘1’ (Figure 4.3(b)) the PDF plot shows the mean value 23.09 and STD=3.67. The findings from the natural bits behaviour and the DRT distribution among stable cells has been summarized in the Table 4.1.
Table 4.1: Findings from data remanence characterization, at room temperature.

<table>
<thead>
<tr>
<th>Types of bits</th>
<th>Percentage of bit-cells amount</th>
<th>Mean DRT (ms)</th>
<th>Standard Deviation for DRT (ms)</th>
<th>Correlation Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable Natural 1’s</td>
<td>63%</td>
<td>23.09</td>
<td>3.67</td>
<td>97%</td>
</tr>
<tr>
<td>Stable Natural 0’s</td>
<td>24%</td>
<td>24.11</td>
<td>3.53</td>
<td>98%</td>
</tr>
</tbody>
</table>

This analysis highlights the typical retention behavior and variability within the memory array, crucial for assessing data remanence. This information is vital for understanding the bit cells behavior in embedded SRAM indicates their reliability and stability of data remanence in memory cells under given conditions such as room temperature.

4.3 Correlation Analysis

To assess the stability and reliability of SRAM data remanence times, we conducted a correlation analysis in all stable natural 1’s and 0’s by writing on logic 1 and 0 respectively and powering up for 59 cycles at room temperature. This experiment was performed twice, and the resulting DRT values were plotted under the same graph, creates a petal-shaped correlation plot.

In Figure 4.4 (a) and (b) We wrote '1’s to all stable natural '0’s and measured their remanence time. Figure 4.4 (c) and (d) showing the correlation while we wrote '0’s to all stable natural '1’s and measured their retention times. We found 98% correlation among all stable natural 0’s and 97% correlation among all stable natural 1’s between the two sets of data, indicates a very high degree of consistency in the remanence times across the experiments. In Figure 4.4 (b)
Figure 4.4: Correlation of DRT between two sets of experiments, at room temperature, (a) Correlation for stable natural 0’s (b) Correlation for stable natural 0’s after 6 months. (c) Correlation for stable natural 1’s. (d) Correlation for stable natural 1’s after 6 months.

and (d) the correlation between two sets of data taken after six months for the same device. This high correlation coefficient proves that the retention behavior of the SRAM cells is highly reproducible under the same conditions, validating the reliability of the device for stable data retention.

This analysis provides strong experimental validation for the stability and reliability of the SRAM device. The strong correlation observed in Figure 4.4
underscores the consistency and predictability of SRAM data remanence times, reinforcing confidence in its long-term performance.

4.4 Power-up State Bitmap in SRAM

Following the analysis of transition behaviour plots Figure 4.2, it is important to examine the bit-wise power-up state characterization of the SRAM cells in greater detail. The presence of unstable cells alongside stable natural cells highlights the importance of conducting a bit-wise assessment of the entire memory array to fully understand the power-up state behavior. This analysis includes having a bitmap of the 2KB embedded SRAM based on 59 power up cycle.

![Figure 4.5: Bitmap representation of the 2K SRAM used for the experiment, at room temperature. (a) Colored bitmap where white represents stable natural 1’s, Black represents stable natural 0’s, Red Represents unstable natural cells. (b) Binary bitmap where Black represents all natural 0’s and White represents all natural 1’s (by majority voting).](image-url)
Figure 4.5 provides a visual representation of the power-up states in a 2KB USB SRAM array, showcasing the indices of all stable natural ‘1’s, stable natural ‘0’s, and unstable natural cells. In Figure 4.5(a), stable natural 0’s are denoted in black and white for stable natural 1’s, with red indicating the unstable natural cells. Figure 4.5(b) highlights the all natural cells determined by majority voting upto 59 power cycles, where white represents all natural 1’s and black represents all natural 0’s. Although the majority of the cells consistently stabilize as either ‘1’ or ‘0’, the unstable natural cells indicate regions of instability towards achieving a natural logic state. This bit-wise assessment, obtained through repeated power cycles, demonstrates that while SRAM is generally volatile and should stabilize to a consistent state over time, certain cells may exhibit unpredictable behavior. Identifying and understanding these cells is essential for developing robust memory systems and ensuring the reliability of data remanence under various operational conditions.

In Chapter 4, the comprehensive characterization of data remanence in SRAM provided crucial insights into the stability and behavior of memory cells under the room temperature. Through a detailed analysis of stable natural 0’s, stable natural 1’s, and unstable natural states of the bit-cells, we identified the inherent stability of specific SRAM cells and their susceptibility. The correlation analysis between repeated experiments highlighted the reliability and consistency of SRAM cells over time, even when tested after a six-month interval. The power-up state bitmaps and subsequent bit-wise assessment emphasize the significance of understanding the initial logic states of SRAM while no data has been written
towards further assessment of data remanence vulnerability. These findings lay the groundwork for exploring temperature-induced vulnerabilities and developing robust security mechanisms to mitigate potential risks associated with data remanence in SRAM.
Chapter 5. Analysis of Temperature Effect on SRAM Data Remanence

This chapter examines the impact of temperature on SRAM data remanence by focusing on the retention of a known image and white and black images at varying temperatures. This analysis includes investigating the data loss percentage over time and comprehending the data loss behaviour based on the thermal effect on it.

5.1 SRAM Data Remanence in Room Temperature

Data remanence refers to the persistence of data in volatile memory even after power is removed. In the context of technology nodes, this persistence can pose significant security risks, especially for systems storing sensitive information. As a result, it is crucial for evaluating the data remanence of SRAM, particularly at room temperature. In this analysis data remanence evaluation starts by understanding the data loss percentage in SRAM at room temperature. The ability to predict and quantify data loss over time helps in identifying vulnerabilities and implementing measures to mitigate data remanence effects.

Data loss in SRAM refers to the gradual degradation of stored data once the power is removed. This degradation is influenced by factors such as tem-
Figure 5.1: Percentage of Data Loss in a known Image for 59 power-up cycle at room temperature.
perature, device variability, and the inherent characteristics of the SRAM cells leads to the effect of data remenance. In our study, the data loss percentage was calculated to understand the remanence effect in the SRAM cells over multiple power cycles. The equation used to calculate the percentage of data loss is:

\[
\text{Data Loss \%} = \frac{\# \text{ of set bits}(\text{Image}_{\text{original}} \oplus \text{PU}_{\text{read}})}{\# \text{ of set bits}(\text{Image}_{\text{original}} \oplus \text{PU}_{\text{ref}})} \times 100\%,
\]

where \(\text{Image}_{\text{original}}\) is the original binary image written to the SRAM, \(\text{PU}_{\text{read}}\) is the power-up state read from the SRAM after a specified time \(t_d\), \(\text{PU}_{\text{ref}}\) is the reference power-up state obtained from multiple power cycles.

Figure 5.1 demonstrates the percentage of image data loss observed in 59 power-up cycles at room temperature (24°C). As seen, the data remains largely intact for the first few milliseconds post power-up, after which there is a sharp increase in data loss, stabilizing at around 100% loss by 30 milliseconds. This behavior indicates that while SRAM cells exhibit a certain degree of data retention initially, the data integrity deteriorates rapidly beyond a critical point.

5.2 Image Retention

Writing a known image (Einstein’s image) to the SRAM and observing its retention over time at room temperature was another security analysis performed by measuring the data loss over time. For this analysis first we selected an Einstein’s image with the bit resolution of 128×128, contains 16384 bits of binary
Figure 5.2: Data loss in Einstein image over time, at room temperature.

Figure 5.3: Data loss in black image and white image over time, at room temperature.

information. After writing the image on the SRAM the data remanence effect has been investigated over 59 power-up cycle and calculated the data loss percentage over time. The experiment was also conducted by writing white and black images to compare the data remanence characteristics.

Figure 5.2 illustrate the progression of percentage of data loss over time for an Einstein’s image and Figure 5.3 for images consisting entirely of 1’s (white) and 0’s (black) at room temperature (24°C). Initially, the images retain 100% of their data, but as time progresses, data loss becomes evident. For the Einstein image, significant data degradation occurs between 14 ms and 17 ms, with complete data
loss by 45 ms. Similarly, for the all 0’s and all 1’s images, data loss begins around 18 ms and 21 ms, respectively, with total data loss observed by 48 ms and 43 ms.

These visual representations highlight the rapid decline in data retention and lower DRT. This little time making it difficult for attackers to exploit. At cold temperatures, reduced leakage currents extend data retention times, making the memory more vulnerable to cold boot attacks as attackers can retrieve residual data long after the power-off.

5.3 Low-Temperature Effect on SRAM Data Remanence

Inspecting the low-temperature effect on embedded SRAM data remanence is vital to understand the device vulnerability in modern technology. This investigation considered as a practical attack by lowering the environmental temperature to a certain extend which will increase the DRT in significant amount and create more flexibility to attackers to exploit the sensitive information stored in the SRAM.

The low temperature effect on SRAM data remanence were observed across a range of temperatures, from room temperature (24°C) to −25°C, in 5°C intervals, in the form of percentage of data loss. Figure 5.4(a)-(c) illustrate the data loss percentages for 800 power-up cycles at different temperatures for the Einstein’s image, white image, and black image, respectively. As the temperature gets lower the percentage of data loss curve gets less steep over time, indicating that lower temperatures effectively slow down the data degradation process, extending the remanence times considerably.
Figure 5.4: Data Loss percentage for 800 power-up cycle at seven different temperatures. (a) Data-Loss in Einstein’s image. (b) Data-Loss in white image (all 1’s). (c) Data-Loss in black image (all 0’s).

The remanence time, required for varying percentages of data loss for the Einstein’s image, white image, and black image were recorded for 800 power-up cycles. The remanence times for different percentages of data loss, recorded under seven different temperature conditions, are summarized in table 5.1.
Table 5.1: Time required for variable data loss percentage at seven different temperatures.

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Einstein’s Image</th>
<th>White Image</th>
<th>Black Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>T= 24°C</td>
<td>15ms 19ms 45ms</td>
<td>25ms 31ms 43ms</td>
<td>20ms 25ms 48ms</td>
</tr>
<tr>
<td>T= 0°C</td>
<td>84ms 111ms 278ms</td>
<td>120ms 151ms 207ms</td>
<td>148ms 190ms 548ms</td>
</tr>
<tr>
<td>T= -5°C</td>
<td>119ms 154ms 397ms</td>
<td>133ms 165ms 230ms</td>
<td>153ms 203ms 593ms</td>
</tr>
<tr>
<td>T= -10°C</td>
<td>164ms 219ms 451ms</td>
<td>235ms 295ms 394ms</td>
<td>283ms 355ms 668ms</td>
</tr>
<tr>
<td>T= -15°C</td>
<td>256ms 341ms 736ms</td>
<td>461ms 560ms 701ms</td>
<td>319ms 408ms 741ms</td>
</tr>
<tr>
<td>T= -20°C</td>
<td>406ms 516ms &gt;800ms</td>
<td>629ms 744ms &gt;800ms</td>
<td>615ms 744ms &gt;800ms</td>
</tr>
<tr>
<td>T= -25°C</td>
<td>649ms 793ms &gt;800ms</td>
<td>&gt;800ms &gt;800ms &gt;800ms</td>
<td>&gt;800ms &gt;800ms &gt;800ms</td>
</tr>
</tbody>
</table>

5.4 Temperature Vulnerabilities

The data from this study reveals significant vulnerabilities in SRAM data retention when subjected to varying temperatures. At lower temperatures, the leakage currents are reduced, resulting in enhanced data remanence time. This phenomenon is particularly evident at $-25^\circ C$, where the Einstein’s image retained almost 100% of its data for up to 800 milliseconds, demonstrating the pronounced effect of lower temperatures on data retention. Table 5.1 shows that at room temperature ($24^\circ C$), 20% data loss for the binary image occurred at 15 milliseconds, while 100% data loss occurred at 45 milliseconds. As the temperature decreased, data retention times increased significantly. For instance, at $-25^\circ C$, the binary
image retained nearly all of its data for 600 milliseconds, showcasing the impact of lower temperatures on data stability.

The extended data retention at low temperatures poses a critical security risk for embedded systems using SRAM to store sensitive data. Attackers can exploit these temperature-induced vulnerabilities to recover residual data post-power-off, including cryptographic keys and other sensitive information. This capability can undermine the overall security of IoT devices and other systems relying on SRAM for secure data storage.
Chapter 6. Conclusion and Future Work

6.1 Conclusion

This thesis has provided a thorough examination of two critical aspects of embedded SRAM security within the MSP430F5529 microcontroller developed by Texas Instruments. The initial phase of the research focused on investigating the data remanence characteristics of the embedded SRAM, utilizing the microcontroller’s built-in power cycling mechanism, which requires no additional hardware. This study has laid essential groundwork for developing PUFs, findings that are crucial for authenticating devices in edge computing applications where maintaining security and data integrity is paramount. Building on the methodologies developed for power-up state analysis, the research then explored the temperature sensitivity of data remanence in SRAM. The findings revealed that data remanence is significantly impacted by temperature, with notably extended remanence time periods at lower temperatures, underscoring the vulnerabilities to cold boot attacks and highlighting the need for robust security measures in environments where devices may be exposed to variable temperatures.
6.2 Future Work

The insights gained from this thesis suggest several pathways for future research to enhance security protocols and device reliability in embedded systems. Future studies should further explore the robustness of SRAM-based PUFs across diverse operational environments and microcontroller platforms to improve reliability and applicability in edge computing and IoT system. Additionally, expanding the investigation into the effects of temperature on SRAM data remanence over longer periods and under more varied environmental conditions will provide deeper insights. Developing security measures that can dynamically adjust based on detected environmental conditions will mitigate the risks posed by high data remanence at low temperatures. Applying the findings of this research to real-world scenarios in edge computing, assessing the performance and security of SRAM-based PUFs in field conditions, and examining the power-up behavior and temperature sensitivity of SRAM in other microcontroller units will help generalize the findings and enhance device security across different platforms.

6.3 Contribution to the Field

This research not only deepens the understanding of the behavior of embedded SRAM in microcontrollers but also sets the stage for the practical application of this knowledge in the design of secure, reliable embedded systems for edge computing and IoT systems. By sequentially addressing power-up states and temperature-induced vulnerabilities, this thesis enriches the field of cybersecurity
in embedded systems, promoting a more nuanced approach to hardware security in modern digital infrastructures. We believe that our experimental approach, methodology, and findings will be a valuable asset for further academic research and industrial reference on this topic.
References


[9] Yaara Shriki, Dikla Barda, and Roman Zaikin. Keeping the gate locked on your IoT devices: Vulnerabilities found on Amazon’s


Appendix A: Code for SRAM Data Remanence Analysis in MSP430F5529 MCU

/*-------------------------------------------------------------
 * Function: Reads unused parts for RAM memory and sends binary data
 * over UCA1 UART channel (visible through USB cable/JTAG)
 * Description: The program reads 4 RAM segments on MSP430F5529
 * The first segment starts at 0x1C00 (USB RAM)
 * The low bytes are sent first.
 * Clocks: ACLK = LFXT1 = 32768Hz, MCLK = SMCLK = default DCO
 * Instructions: Set the following parameters in putty:
 * Find the COM port number that corresponds to the name "MSP Application UART1" in Device Manager.
 * Enable logging (capture all characters)
 * in a file to capture power-on state of the SRAM memory.
 *
 * Port: COMx (x should be the number, e.g., COM4)
 * Baud rate: 115200
 * Data bits: 8
 * Parity: None
 * Stop bits: 1
 * Flow Control: None
 *
 * MSP430f5529
 * --------------------------------------------------

75
```c
#include <msp430.h>
#include <stdint.h>
#include <stdio.h>

#define SRAM_START 0x1C00 // Define start of SRAM
#define SECTOR_SIZE 2048 // Define size of each sector

void USCIA1_setup(void) {
    P4SEL |= BIT4 + BIT5; // Set USCI_A1 RXD/TXD to receive/transmit data
    UCA1CTL1 |= UCSWRST; // Set software reset during initialization
    UCA1CTL0 = 0; // Clear USCI_A1 control register
}
```
UCA1CTL1 |= UCSSEL_2; // Set clock source to SMCLK
UCA1BR0 = 0x09;    // Set baud rate (lower byte)
UCA1BR1 = 0x00;    // Set baud rate (upper byte)
UCA1MCTL = 0x02;   // Set modulation
UCA1CTL1 &= ~UCSRST; // Initialize USCI state machine

void UARTA1_putchar(char c) {
    while (!(UCA1IFG & UCTXIFG)); // Wait for transmission buffer to be ready
    UCA1TXBUF = c; // Transmit character
}

void UARTA1_putchar(char c) {
    char hv[5];
    sprintf(hv, "0x%hx,", c); // Convert byte to hex string
    for(int i = 0; i < 5; i++) {
        UARTA1_putchar(hv[i]); // Transmit each character
    }
}

void writeHexToSRAM(unsigned char* data, unsigned int length) {
    for (unsigned int i = 0; i < length; i++) {
        ((volatile unsigned char *) SRAM_START)[i] = data[i];
        // Write data to SRAM
    }
}

int main() {
register char s = 0;       // counts 2 KB SRAM sectors
register unsigned int i = 0; // counts bytes in an SRAM sector
register unsigned int t = 1;
register unsigned int p = 0;
register char c;              // byte read from SRAM
WDTCTL = WDTPW + WDTHOLD;   // Stop watchdog timer

for (p=0; p<800; p++) {
    char * mypc = (char *) SRAM_START;

    unsigned char imageData[] = "ImageDataPlaceholder";
    // Placeholder for image data
    writeHexToSRAM(imageData, sizeof(imageData));

    RCCTL0 = RKEY | RCRS7OFF; //this is for Sector 7

dynamicDelay(t);
    RCCTL0 = RKEY;

USCIA1_setup();            // Initialize USCI_A1 module in UART mode
__delay_cycles(50000);     // Wait a half second or so
for (s=0; s<1; s++) {
    // print next RAM sector of 2 KB
    for (i=0; i < SECTOR_SIZE; i++) {
        c = *mypc;
        UARTA1_putchar(c);
        mypc++;
    }
}
char buffer[5];
register int k = 0;
sprintf(buffer, "end%d", 1);
for(k=0; k<5; k++) UARTA1_putchar(buffer[k]);

t=t+1;

}