Circuit Design of SRAM Physically Unclonable Functions

by

Anthony Hok Hin Ho

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Masters of Applied Science

in

Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2017

©Anthony Ho 2017

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A Physically Unclonable Function (PUF) is an entity that reliably provides a unique response to a given challenge and cannot be easily duplicated physically. PUFs are an alternative to using non-volatile memory (NVM) for secure key storage. NVMs are susceptible to reverse engineering and side channel attacks that can extract sensitive data. PUFs take advantage of random physical variations that are introduced during manufacturing. PUFs can be used to create digital fingerprints as secret keys for cryptographic algorithms or for device authentication. SRAM PUFs, in particular, are of great interest due to their omnipresence in electronics. One of the weaknesses of SRAM PUFs is their reliability as noise and other environmental effects reduce the reproducibility of the PUF.

This thesis provides an in depth analysis of the 6T SRAM PUF and 8T soft error robust SRAM PUF at the transistor level and provides a methodology to design a reliable PUF. We hypothesize that the V_{GS} of pull up and pull down transistors during the power up phase affects PUF reliability. Transistors with a larger V_{GS} have higher drive strength and more influence over the start-up value of the PUF. Changing the sizing ratio of PMOS to NMOS devices changes the V_{GS}. Nominal simulations recorded V_{GS} in relation to the V_{DD} ramp-up to predict which devices have a higher influence on start up values.

Two types of PUF schemes: V_{DD} manipulation and GND manipulation are simulated. Monte Carlo simulations are performed within the Cadence Virtuoso environment using TSMC general purpose CMOS kit. The reliability metric is called the assured response which is the number of Monte Carlo samples that show a consistent response over 100 power ups.

The results from V_{GS} dependency analysis and isolated mismatch show a clear trend between V_{GS} and the type of device that determines PUF reliability. Devices with higher V_{GS} during V_{GS} dependency analysis show a larger drop in assured response when their mismatch is disabled in the isolated mismatch simulation. Sizing sweeps show that skewed designs have higher assured response than less skewed designs. This is because smaller transistors have poor matching properties and relatively higher V_{GS} which contribute to improved reliability. V_{DD} manipulation and GND manipulation showed similar levels of reliability while 6T performed better than 8T. In an effort to improve the 8T PUF, a split V_{DD} scheme is proposed which introduces a delay between two V_{DD} signals in the cell. This shows a 3% improvement over a skewed 6T V_{DD} design which was previously the best performer.

Acknowledgements

I would like to thank several people who have helped me one way or another throughout my graduate studies.

First and foremost I would like to thank my thesis advisor Professor Sachdev. His insight and expertise have contributed to a very rewarding graduate school experience. He consistently allowed this research to be my own work but steered me in the right the direction whenever he thought I needed it. It has been a privilege to work under someone with as much experience as him.

Thank you to Professor Gebotys and Dr. Derek Wright for taking time out of your busy schedules to be my thesis reviewers.

I would like to thank my mentor Dr. Adam Neale for teaching me, sharing his knowledge and making my transition into graduate school so much smoother. I am extremely grateful to him for helping me shape my career.

I also would like to thank Dr. Muhammad Nummer for being my mentor during my internship at TSMC and easing my frustration with the tools.

Thanks to Qing for providing valuable lessons in layout and insightful talks. Thanks to Sunil for editing my thesis and providing Inkscape help. Thank you to fellow group members Kai, Sakib, Dhruv, Govind, Mahdi and Morteza for all the interesting conversations whether it was about circuits, career paths, travel or how hot it got in the office, one way or another it has truly enriched my experience as a graduate student.

Thanks to Phil Regier for maintaining our servers and helping with bugs and licensing issues so promptly.

I am indebted to my girlfriend Jacklyn who has always been there to provide unfailing support and continuous encouragement.

Lastly, I would like to thank my family for providing their support and always welcoming me home whenever I visit.

Table of Contents

Abstract iii

Acknowledgements iv

Table of Contents v

List of Figures viii

List of Tables xi

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 Wi-Fi Vulnerabilities 1

1.3 Current Security Methods 2

1.4 Attacks on NVM 2

1.4.1 Invasive 2

1.4.2 Semi invasive 2

1.4.3 Non-Invasive attacks 3

Chapter 2 Background 4

2.1 Use Cases 4

2.1.1 Supply chain management 4

2.1.2 FPGA code protection 6

2.1.3 Counterfeiting 7

2.2 PUF in Security Architecture 8

2.2.1 Device authentication 8

2.2.2 Secure key generation 10

2.2.3 Random seed generator 10

2.3 Cryptography 11

2.3.1 Symmetric key algorithms 11

2.3.2 Public key cryptography ... 11
2.4 Classification of PUFs ... 11
2.4.1 Strong PUF ... 11
2.4.2 Weak PUF ... 11
2.5 Types of PUFs ... 12
2.5.1 Ring oscillator (RO) PUF ... 12
2.5.2 Arbiter PUF ... 12
2.5.3 SRAM PUF ... 13
2.5.3.1 Metastability ... 13
2.5.4 Delay hardened PUF ... 15
2.5.4.1 Repetition code ... 15
2.5.4.2 Temporal majority voting ... 16
2.6 Mismatch Properties of Transistors ... 16
2.7 6T Cell Background ... 18
2.7.1 Read operation ... 18
2.7.2 Write operation ... 18
2.8 8T Cell Background ... 19
2.8.1 Read operation ... 20
2.8.2 Write operation ... 20
2.9 Noise ... 21
Chapter 3 Methodology ... 22
3.1 Transient Noise ... 22
3.2 V_{GS} Dependency ... 23
3.3 Isolated Mismatch Experiment ... 23

3.4 6T V_{DD} Manipulation 24

3.5 6T GND Manipulation 26

3.6 8T V_{DD} Manipulation 28

3.7 8T GND Manipulation 30

3.8 Split-V_{DD} Experiment 31

3.9 Simulation settings 34

Chapter 4 Simulation Results and Comparative analysis 35

4.1 V_{GS} Dependency Analysis 35

4.1.1 V_{DD} manipulation 35

4.1.2 GND manipulation 39

4.2 Isolated Mismatch Results 44

4.2.1 V_{DD} manipulation 44

4.2.2 GND manipulation 45

4.3 Sizing Sweep Results 46

4.3.1 Effect of supply noise 48

4.3.2 Effect of temperature 49

4.3.3 Uniqueness analysis 50

4.4 Split-V_{DD} Results 51

4.4.1 Best candidate comparison 52

4.4.1.1 Temperature sweep 53

4.4.1.2 Delay sweep 54

Chapter 5 Conclusion 55

Bibliography 56

List of Figures

Figure 2.1 RFID in supply chain [13] 5

Figure 2.2 FPGA with encrypted configuration system[13] 6

Figure 2.3 FPGA with PUF code protection: (a) Enrollment phase: helper data is generated; (b) Storing the encrypted configuration data; (c) FPGA configuration [13] 7

Figure 2.4 Typical device authentication with NVM 9

Figure 2.5 Device authentication with PUF 9

Figure 2.6 Secure key generation scheme 10

Figure 2.7 Arbiter PUF 13

Figure 2.8 (a) Unbiased metastable system (b) Strong logic 0 bias (c) Strong logic 1 bias 13

Figure 2.11 A 6T SRAM PUF butterfly curve is shown. The cell has a bias towards S1 since the V_Q=V_{QB} line is above M1 14

Figure 2.12 Repetition code generation 15

Figure 2.13 Repetition code usage 15

Figure 2.14 TMV scheme 16

Figure 2.17 6T cell schematic 18

Figure 2.18 6T cell read write operations 19

Figure 2.15 8T cell schematic 19

Figure 2.16 8T read/write operation 20

Figure 3.1 V_{gs} dependency analysis 23

Figure 3.2 6T VDD manipulation PUF testbench schematic 24

Figure 3.4 6T V_{DD} manipulation PUF cycle 25

Figure 3.5 6T GND manipulation PUF testbench schematic 26

Figure 3.6 6T GND manipulation PUF cycle 27

Figure 3.7 8T V_{DD} manipulation PUF testbench schematic ... 28

Figure 3.8 8T V_{DD} manipulation PUF cycle ... 29

Figure 3.9 8T GND manipulation PUF testbench schematic ... 30

Figure 3.10 8T GND manipulation PUF cycle ... 31

Figure 3.11 Split V_{DD} cell ... 32

Figure 3.12 Split V_{DD} PUF challenge V_{DD1} ramps up first ... 33

Figure 3.13 Split V_{DD} PUF challenge V_{DD2} ramps up first ... 33

Figure 4.1 6T 200nm NMOS 200nm PMOS ... 35

Figure 4.2 6T 200nm NMOS 600nm PMOS ... 36

Figure 4.3 6T 120nm NMOS 120nm PMOS ... 37

Figure 4.4 8T 200nm NMOS 200nm PMOS ... 37

Figure 4.5 6T 200nm NMOS 600nm PMOS ... 38

Figure 4.6 8T 120nm NMOS 120nm PMOS ... 39

Figure 4.7 6T 200nm NMOS 200nm PMOS ... 39

Figure 4.8 6T 200nm NMOS 600nm PMOS ... 40

Figure 4.9 6T 120nm NMOS 120nm PMOS ... 41

Figure 4.10 8T 200nm NMOS 200nm PMOS ... 41

Figure 4.11 8T 200nm NMOS 600nm PMOS ... 42

Figure 4.12 8T 120nm NMOS 120nm PMOS ... 43

Figure 4.13 Sizing effect on assured response at 27 °C ... 47

Figure 4.14 Sizing sweep with supply noise ... 48

Figure 4.15 Assured response with supply noise ... 49

Figure 4.16 Sizing effect on assured response at 125°C ... 49

Figure 4.17 Sizing effect on assured response at -40°C ... 50

Figure 4.18 Distribution delta 50

Figure 4.19 Split V_{DD} sizing effect when V_{DD2} rises first 51

Figure 4.20 Split V_{DD} sizing effect when V_{DD1} rises first 52

Figure 4.21 Temperature sweep comparison 53

Figure 4.22 Split V_{DD} delay sweep 54

List of Tables

Table 3.1 Simulation settings 34

Table 4.1 V_{GS} dependency analysis summary 43

Table 4.2 ΔV_{GS} dependence summary 44

Table 4.3 Isolated mismatch results: 200nm NMOS 200nm PMOS 44

Table 4.4 Isolated mismatch results: 200nm NMOS 600nm PMOS 45

Table 4.5 Isolated mismatch results: 120nm NMOS 120nm PMOS 45

Table 4.6 Isolated mismatch results: 200nm NMOS 200nm PMOS 45

Table 4.7 Isolated mismatch results: 200nm NMOS 600nm PMOS 46

Table 4.8 Isolated mismatch results: 120nm NMOS 120nm PMOS 46

Table 4.9 Isolated mismatch summary 46

Table 4.10 Input space 47

Table 4.11 Best candidate sizing 52

Chapter 1
Introduction

1.1 Motivation

The internet of things (IoT) is a growing network of internet-connected everyday devices that can send and receive information with minimal human interaction. In the present, many devices such as our mobile phones and vehicles are connected to the internet. The development of the internet of things has substantially increased the number of connected devices. Gartner Inc estimates 25 billion devices across the glove in 2020 whereas today it is estimated that there are around 6-9 billion devices connected [1][2]. With increased connectivity comes a greater need for security, as new avenues of attack become open and exploitable by criminals or cyber terrorists if not properly secured.

1.2 Wi-Fi Vulnerabilities

The development of wireless communications is one of the milestones of the information age; however, it is not without a cost to security. This section discusses several types of attacks that can hijack or disrupt communication through the Wi-Fi medium.

Denial-of-Service is a type of attack that disrupts the normal operation of a server. The attacker is able to overload a server by making superfluous requests for resources and prevents intended users from accessing the resources. The 802.11 framework also allows requests to de-authenticate clients or access points which the attacker can manipulate to deny service to individual clients or an entire channel [3]. These types of attacks are possible on 802.11 Wi-Fi protocol, which makes it possible for attackers to be physically located anywhere with a Wi-Fi connection.

Man-in-the-Middle attacks are used to eavesdrop and hijack communication in order to steal valuable information and distort messages between clients and servers [4]. An attacker will insert themselves between sender and receiver and impersonate both of them well enough that the other is satisfied. For example, this can be used to steal banking information by redirecting users to a fake online banking site that collecting login data.

Replay attacks involve eavesdropping that copying a message stream and resending it with the receiver perceiving it as legitimate. If not properly secured this technique can be used to intercept hashed keys and replay them to be granted access. The list of attacks continues but these vulnerabilities illustrate the need to advance security technologies.

1.3 Current Security Methods

Today, non-volatile memory (NVM) is a common way of hiding security related information. NVM can be discrete or embedded. Discrete NVMs are cheaper but have external pins that can be probed by an attacker [5]. Embedded NVMs do not have exposed pins but are expensive. Fuses are an example of NVM that can be used to store secrets but are vulnerable to reverse engineering attacks since blown fuses, can be physically observed and read out. Fabricating embedded NVM requires up to 15 additional masks and process steps [6] on top of standard CMOS process. Sometimes embedded NVM is not available in newer technology nodes so if it is needed, older technology must be used [5].

1.4 Attacks on NVM

Attacks can be classified into three different categories: invasive attacks, semi invasive attacks and non-invasive attacks. Invasive attacks require direct physical access to the device and expensive equipment to analyze the structure of the device. Non-invasive techniques are non-destructive and do not require initial preparation of the device under test. Semi invasive attacks require moderate access as de-packaging of the device is needed just like in invasive attacks but do not require expensive equipment to perform.

1.4.1 Invasive

Reverse engineering attacks involve analyzing the circuit structure through product teardown, system level/process analysis and circuit extraction. This can provide information, on how the encryption algorithm works, where keys are stored and critical areas to probe for further analysis [7]

Micro-probing is a functional analysis using signal generators, logic analyzers, and oscilloscopes to provide input and observe the behavior of the circuit. This can be used to extract keys from NVM by providing information on how the encryption algorithm works [7].

1.4.2 Semi invasive

Modification attacks on EEPROM are possible using microprobes by setting and resetting target bits in order to break the Data Encryption Standard (DES) algorithm [8]. It is possible to modify bits by shining UV light on small sections of memory. By setting specific bits and observing parity error messages, it is possible to determine the contents of the memory which is typically read protected. It is also possible to modify special write protect bits to prevent the codes from being erased [9]. This type of

attack is cheaper than invasive attacks since inexpensive equipment can be used to breach the passivation layer. Fault attacks rely on inducing faults by manipulating internal or external signals in order to obtain an information leak. There are many fault attacks such as glitch attacks that cause the CPU into a wrong execution path, manipulating write/read/erase operations [9].

1.4.3 Non-Invasive attacks

Non-invasive attacks do not damage the device but analyze the behavior of the device under irregular conditions to extract secured information. Non-invasive attacks are also known as side channel attacks and there are many different forms of these attacks. One popular form of side channel attack is the power analysis. A Power analysis analyzes the relationship between data and the power dissipated. By collecting many power measurements on the MSB for plaintext and calculating the difference between the means for a 1 and a 0, estimation can be made on which byte a power trace corresponds to. Knowing the encryption algorithm then allows a key to be deciphered [10].

Timing attacks are another form of side channel attacks that analyzes the amount of time a logic operation takes. This is a useful attack on algorithms with dependence on timing variation such as in [11].

It is also possible to analyze the radiation being emitted by the device and correlate the amount of radiation to the type logic operation being performed. This form attack is also a side channel attack since it is using auxiliary information [12].

Chapter 2
Background

PUFs are physical one-way functions that provide a response when presented a challenge. A PUF needs to perform reliably and provide unpredictable but unique responses. One analogy is that PUFs provide a fingerprint for a device much like a human fingerprint. A challenge in a PUF constitutes a physical stimulus and the response is a function of the physical arrangement of the PUF and environmental conditions. A PUF has the following properties

- **Evaluable**: A PUF needs to be easily evaluable in order to keep power, area, and cost down.
- **Reliable**: A reliable PUF entails that the same response is generated for a particular challenge. Typically, PUFs are arranged in words, each corresponding to a key. The metric used is the intra-hamming distance, which is the sum of different bits between the sample challenge and a reference (golden response). Reliability in PUFs does not have to be perfect but needs to be high enough that errors can be corrected with Error Correction Codes ECC or minimized through majority voting schemes.
- **Unclonable**: This means that knowledge of the response of a particular PUF provides no information on the function. The function is ideally impossible to model and in word based PUFs the metric is inter-hamming distance. This property prevents attackers from cloning the PUF. Half of the bits should match if each bit is truly independent which corresponds to an inter-hamming distance of 50%. On the other hand, the PUF response space needs to be large enough such that the collisions are practically impossible. This is because PUFs are unaware of the responses of other PUFs so a collision is mathematically possible. This issue is much like the birthday problem where there is a 50% chance that two people share the same birthday in a room with 23 people.

2.1 Use Cases

2.1.1 Supply chain management

Currently, barcodes are a popular form of product tracking in commerce. They are cheap and can be printed in the same processing step as the rest of the label. They are very limited in the sense that they only identify what type of product it is. RFIDs can bound a tag to a good but is not in use due to extra

processing steps compared to barcodes. This increases production costs to a point that RFIDs are considered economically unfeasible. PUFs are a way of bringing down the cost of the RFID. One reason is that PUFs would replace ROM required on a tag to provide an ID. Another reason is that the ID comes almost for free with the tag so no enrollment is needed as is the case with barcodes.

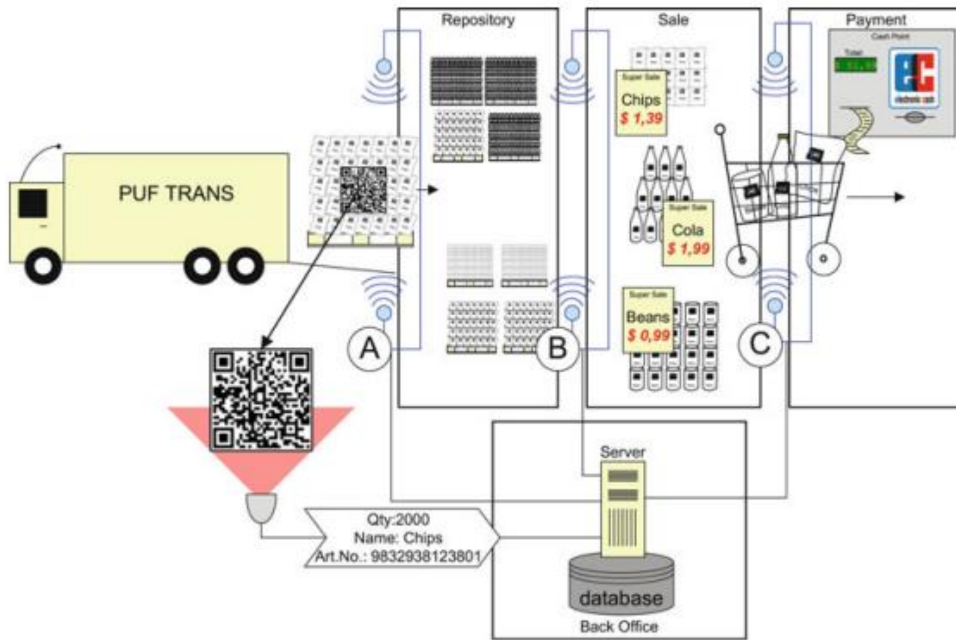

Figure 2.1 RFID in supply chain [13]

In Figure 2.1 the usage of a PUF-RFID tag utilized for item identification is shown: Near the start of the supply chain the merchandise must be registered and the enrollment information is stored in a database. The PUF-ID of each item must be assigned to an article-ID. The article-ID gets unique properties of the associated good. Moreover, properties such as expiry date and the date sold can be stored to aid consumers or producers and retailers. In point A in Figure 2.1, all delivered goods on a pallet are scanned registered at once. The pallet itself has a barcode that gives some general details. This barcode is then scanned and the data is assigned to every RFID on the pallet. The products are then shipped to the stores and scanned again to track their location in the supply chain. The customer then picks up the item like they normally would and can walk out the door and pay simultaneously through the RFID. The

advantages of this technology are numerous. Costs in staffing can be reduced, especially at checkout where billing can be performed automatically. Today, self-checkouts are implemented at retailers such as Walmart and Zehrs. Inventory management is simplified and human error is reduced across various roles in retail. The time-tested problem of queue management at checkouts is solved. There are challenges that arise as well, such as the need for RFID compatible equipment. Scanning needs to be able to cover long distances in order to be advantageous to barcode scanning. Also, many producers now have to modify their production lines to accommodate RFIDs, which can be costly. For the time being, the costs of implementing RFIDs outweigh the benefits which mean barcodes will be used until cheaper RFIDs can be developed.

2.1.2 FPGA code protection

Field Programmable Gate Array or FPGA is a versatile reconfigurable integrated circuit which is very popular due to its fast time to market. In times when the code on the FPGA is proprietary, attackers can clone the code which is stored in its NVM. There are methods of protecting this code [14] but nevertheless, there are vulnerabilities during data transfer to the FPGA. Encrypting the code with a key is one way of combating this. During the startup phase, the encrypted configuration files are loaded into the volatile memory of the FPGA.

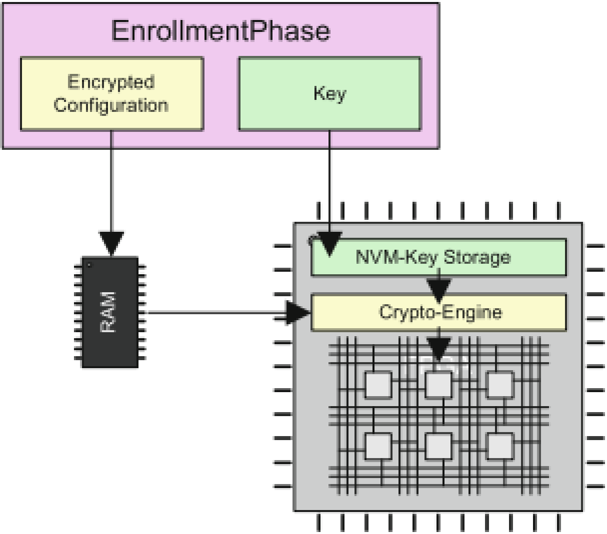

Figure 2.2 FPGA with encrypted configuration system[13]

Figure 2.2 shows the encryption process using traditional NVM key storage. First, an enrollment is done after the device is created and the key is stored inside the FPGA NVM. The encrypted configuration is stored outside the FPGA which attackers will have access to but do not have a key for. When reconfiguration is requested the key is read from the NVM and used to decrypt the configuration for use.

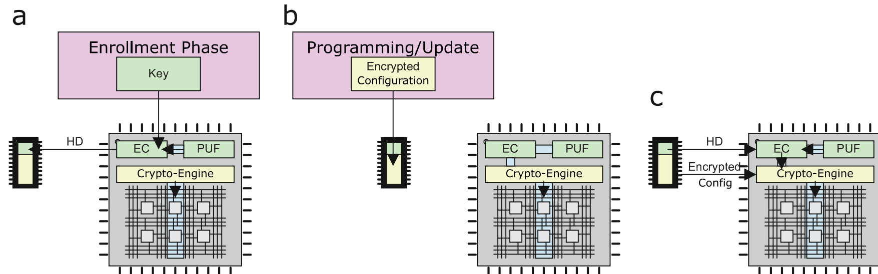

Figure 2.3 FPGA with PUF code protection: (a) Enrollment phase: helper data is generated; (b) Storing the encrypted configuration data; (c) FPGA configuration [13]

Error! Reference source not found. shows the PUF FPGA code protection scheme. Amid the enrollment stage, the framework is set up to unscramble the configuration file. This must be done in a trusted and secure location. The key is then sent to the FPGA. Inside, the PUF cells are read out. The result is utilized to mask the key and helper data (HD) is generated to assist in key generation. At this point, the key can be regenerated and used for decrypting.

2.1.3 Counterfeiting

Counterfeiting ICs has a huge impact on the economy of the semiconductor industry. It is estimated that companies lose about $100 billion of revenue each year due to counterfeiting [15]. It also poses a reliability and safety hazard especially in the automotive, military, and medical sectors where defective parts lead to equipment failure and possibly harm humans. Counterfeit ICs can exist as non-authorized copies of designs, non-authorized manufacturing of original designs or underperforming/defective ICs that are marketed as new and within specifications. Non-authorized manufacturing can occur when production is outsourced from a design company to a manufacturing and the manufacturer decides to create more products than requested and sells them illegally. This can be combated by identifying unique features in the product for tracking and identification. PUFs can be used to verify that the original equipment manufacturer indeed created the device to be used by the consumer. They can also be used as keys or key generators to encrypt messages in a public/private key protocol. Typically, secret keys are

stored in on-chip fuses, a form of NVM. Counterfeiters can clone devices by reading the key of a genuine device through invasive or non-invasive techniques and then reuse the key in order to get past authentication protocols.

2.2 PUF in Security Architecture

Because of the way that PUFs generate responses from the intrinsic properties of a device, random number generation and key storage does not have to be done outside of the chip. This reduces overhead and diminishes expenses substantially. This fact alone can make the difference between a competitive and a mediocre product.

In situations where a device is utilized for identification purposes, a unique ID must be accessible. Most systems today are storing IDs in NVMs. A chip without NVM is normally less expensive to deliver, on the grounds that additional processing steps are required in fabrication. This unique ID must be produced outside the chip and later transferred back which causes increased expenses. Using a PUF bypasses these issues since the ID is inherently available on chip.

2.2.1 Device authentication

Device authentication is a way of preventing counterfeiting by querying devices for a response. Typically when a device is created, a secret key is also created and stored on the device NVM and host database. This storing of this key is typically done in a trusted and secure environment to prevent keys from leaking out to potential attackers. When the device is shipped off to customers and customers, the key is read out of the NVM and checked against the database. If a match is found the device is considered authentic and access is granted. The problem with this operation is that when the device is shipped outside the secure environment, attackers can tamper with the device and copy keys and clone devices as described above.

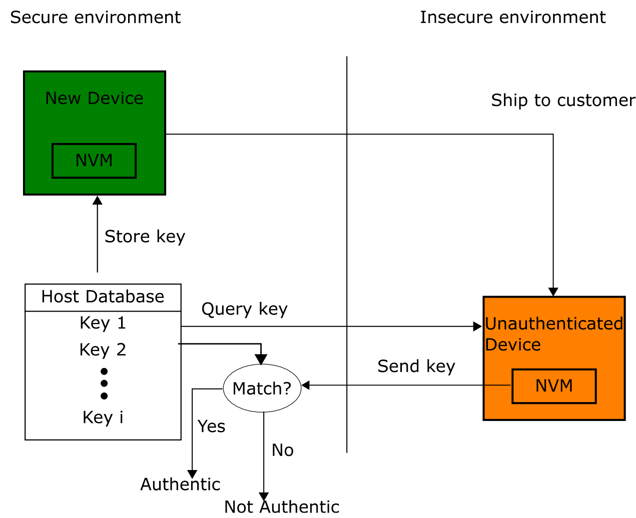

Figure 2.4 Typical device authentication with NVM

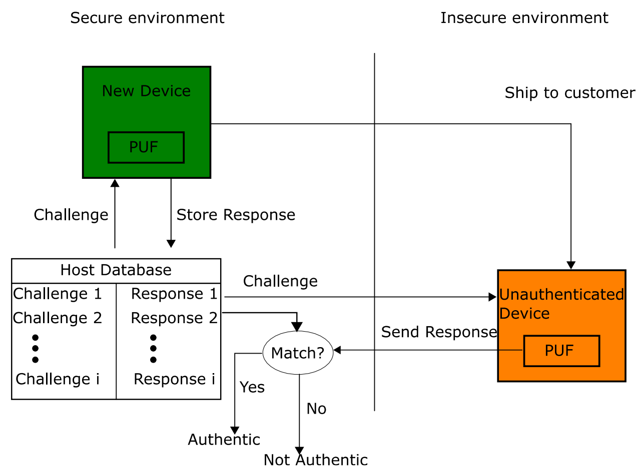

Figure 2.5 Device authentication with PUF

In Figure 2.5, instead of storing keys in memory, the keys are in the form of CRPs. When the device is created, it is issued a challenge and provides the response which is stored in a database. There is no need to store the response in an NVM as it is generated on the fly when the challenge is issued. The corresponding response to a challenge must be given for authentication to be successful. To prevent man in the middle attacks, the CRPs are used only once and discarded.

2.2.2 Secure key generation

PUF output can be used to generate volatile secret keys but require error code correction (ECC). This is because a perfect match is required which is difficult to obtain under noise and other environmental conditions.

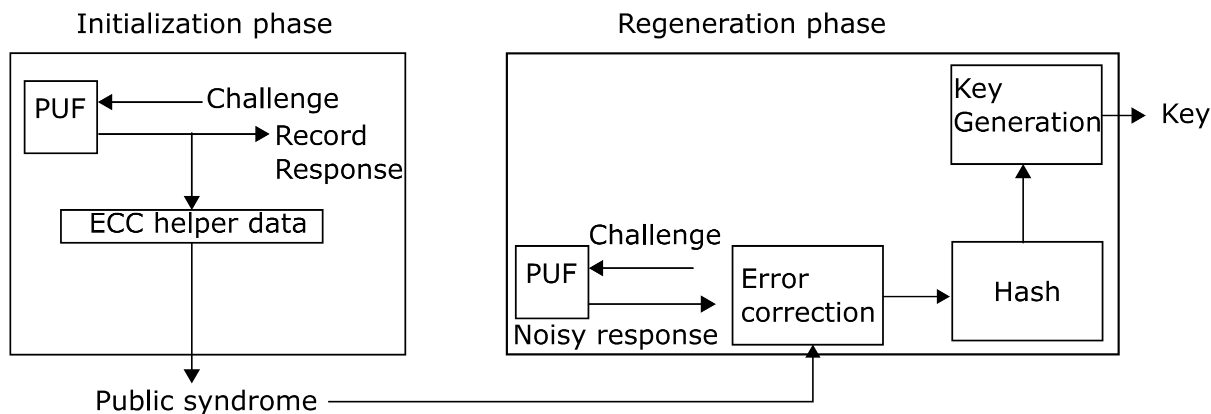

Figure 2.6 Secure key generation scheme

Figure 2.6 shows an overview of a secure key generation scheme with a PUF [16]. When the device is first created, its response is recorded, helper data and a public syndrome is generated for error correction purposes. The syndrome provides information on where bit errors are. This syndrome is public and could be figured out by attackers so it is important to have more secret bits than ECC bits. When a key is needed the PUF is challenged and the response is passed through error correction in case the response has changed. The response is then hashed according to the desired security algorithm in order to generate the key.

2.2.3 Random seed generator

Random number generation (RNG) is important for cryptography in order to generate keys that cannot be deduced by modeling how the key was generated. RNG generators are either truly random or pseudo random where a seed can perfectly reproduce a set of random numbers if the same set is needed (for reuse of keys or debugging).

2.3 Cryptography

2.3.1 Symmetric key algorithms

In this type of algorithm, the key is shared by sender and receiver with no restrictions on the keys, which relaxes specifications on the PUF. Examples of symmetric key algorithms include Advanced Encryption Standard AES and DES. AES is more popular due to longer key lengths. Longer key lengths hinder brute force attacks, which try every single combination of keys.

2.3.2 Public key cryptography

Also known as asymmetric key cryptography, this type of algorithm uses different keys for the sender and receiver. It is called public since one of these keys is visible to the public. The private key used by the sender to encrypt the message and the public key is used by the sender to decrypt the message. This allows the receiver to verify the origins of the message and is used in authentication. One disadvantage is that the keys need to satisfy special mathematical properties in order to be used. Rivest-Shamir-Adleman Encryption (RSA) and elliptic curve cryptography are examples of public key algorithms [17]. Another downside is the increased complexity and higher power consumption. PUFs can be used a key for these purposes, however, PUFs typically have bit errors at the output and require error correction. ECC is a is one major research area for PUFs since cryptographic keys require perfect reproducibility of the key.

2.4 Classification of PUFs

2.4.1 Strong PUF

Strong PUFs have many challenge and response pairs which are useful for having authentication protocols that only allow one use of each CRP before it must be discarded. This allows access to the challenge response system to be lax so attackers are able to issue challenges and read responses. However, due to the large number of CRPs, this does not provide useful information. As long as there is no correlation between the pairs it is impossible to model the system.[18]

2.4.2 Weak PUF

Weak PUFs have a small number of challenge and response pairs and this means that access to the challenge response system should have restricted access such that attackers cannot read the responses

even if they physically possess the PUF [18]. Static Random-Access Memory (SRAM) PUFs are the most popular form of weak PUFs which utilize threshold voltage (V_t) variation as the physical phenomenon that creates the digital signature. Manufacturing variation due to RDF and line edge roughness affects the V_t and creates an imbalance within the SRAM. The challenge response mechanism typically involves putting the SRAM PUF in a metastable state followed by the challenge which will enable the SRAM to self-evaluate to stable state which is primarily affected by the threshold variation. There are many methods of evaluation for SRAM PUF and a common technique is the power up. This is where the initial state is when the PUF does not have power and all storage nodes contain a logic 0 which is the metastable state. The V_{DD} is then powered on and the SRAM evaluates based on its physical parameters and environmental conditions. Environmental conditions can include temperature and noise which can cause the SRAM to evaluate to another state rather than its preferred state. Temperature can affect the ramp up time which has been observed to cause a change in reliability [19].

2.5 Types of PUFs

2.5.1 Ring oscillator (RO) PUF

A ring oscillator consists of an odd number of inverters joined in a ring fashion. An odd number allows the RO to oscillate with a frequency equal to twice the sum of the delays through the inverters. It is typically used in a phase lock loop but it has useful properties that make it a viable PUF. Due to process variation, the frequencies will be slightly different due to the variation in delay between gates. A single PUF bit can be created by comparing the frequencies between ROs. In the figure below, multiplexers are used to select the ROs to be compared and counters are used to measure the frequency. A comparator assigns a 1 or 0 based on which RO is faster.

2.5.2 Arbiter PUF

The Arbiter PUF is a strong PUF that consists of pairs of 2-1 multiplexers connected in series with an output D-latch as seen in Figure 2.7 [20]. The same input signal is connected to all inputs of the multiplexer in the first stage so there are initially four input signals but only two will proceed through the first multiplexer. The control signals to the multiplexers determine the path that will be taken by the input signals. This can be used as a PUF since the electrical paths are identical by design but variations create a slightly different delay between the paths. The response is determined which signal reaches the D latch first. If D is first then the response is a 1 and vice versa. The control signals provide the challenge and it is

clear that the number of challenges increases exponentially with the number of stages which is a benefit of the arbiter PUF. However, one downside to the Arbiter PUF is its susceptibility to modeling attacks[20].

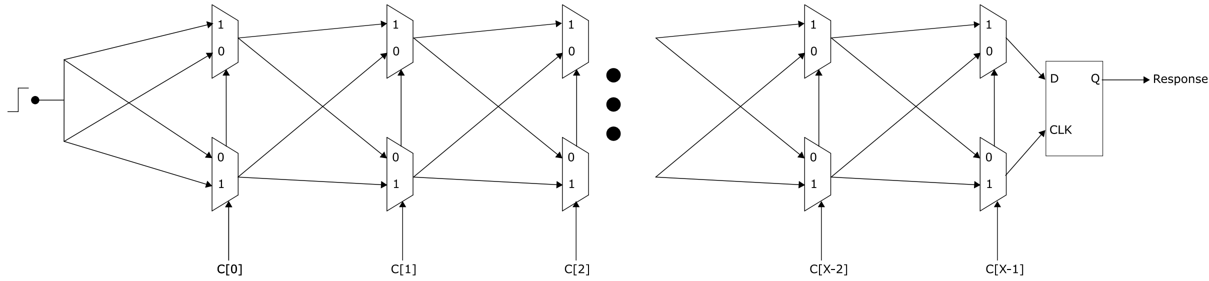

Figure 2.7 Arbiter PUF

2.5.3 SRAM PUF

The SRAM PUF is a weak PUF based on the positive feedback and exploiting metastability inherent to SRAMs. SRAMs are symmetrical by design but manufacturing variations cause an imbalance. SRAM PUFs are popular due to the ubiquitous nature of SRAMs in devices and their ease of use. This entails low overhead which is a desirable characteristic in terms of economics. A simple power up scheme is sufficient for PUF operation. SRAMs are volatile in nature which means attacks to obtain secrets from the SRAM are futile while power is off.

2.5.3.1 Metastability

To illustrate the concept of metastability in a SRAM, imagine a ball on top of a hill (Figure 2.8). The ideal case occurs when the ball stays on top of the hill and does not roll down. This case is analogous to a SRAM with no manufacturing variation and therefore no bias towards either a logic 0 or logic 1. With a real system, the ball will have a stronger bias to a particular state and will roll towards that state with a high probability. The small arrows represent a small chance that the system can resolve to that state, depending on environmental conditions.

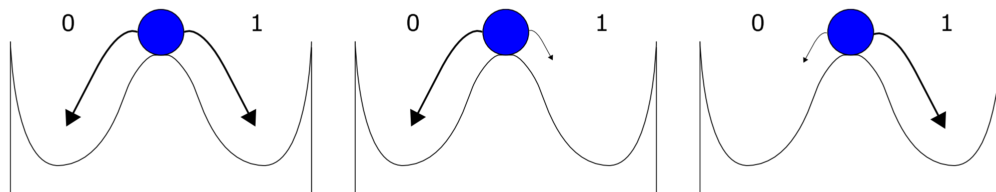

Figure 2.8 (a) Unbiased metastable system (b) Strong logic 0 bias (c) Strong logic 1 bias

The butterfly curve in Figure 2.9 illustrates the states. The red state is metastable. In reality, any noise at M1 will cause the system to shift to one of the stable states. The blue states are the stable states reinforced by positive feedback when power is applied. The 45-degree line from the origin is the line where V_Q=V_{QB}. Initially, the system is at the origin where no charge is present. Next, the power begins to ramp up and as this happens both voltages rise at the same rate. After some time, the difference in V_t between the left and right inverter starts to separate the nodes and the positive feedback mechanism takes over and swings the system to one of the blue stable states. This example shows a case where there is a bias towards S1. Under ideal conditions, M1 should be on the V_Q=V_{QB} line.

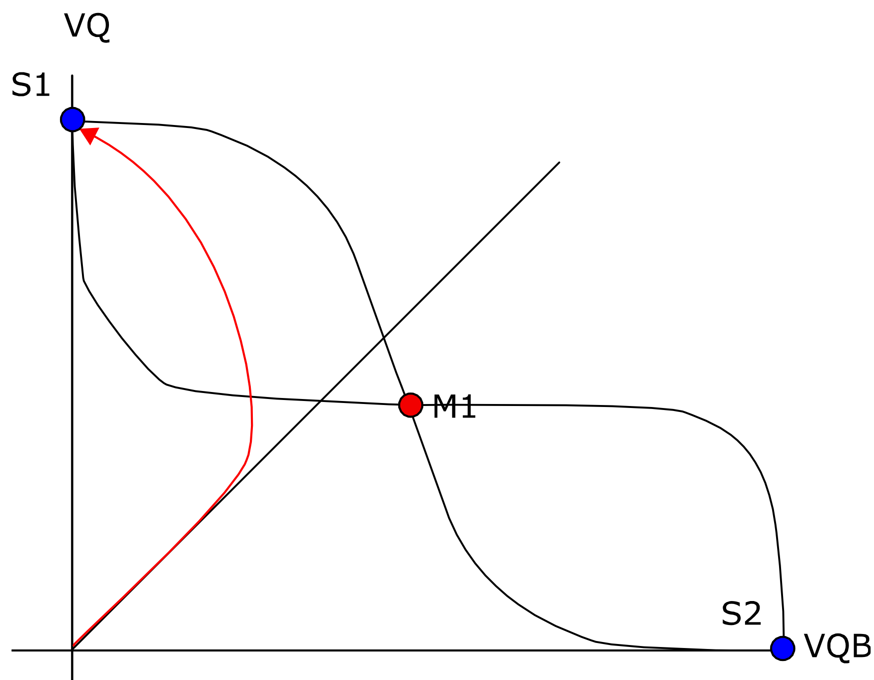

Figure 2.9 A 6T SRAM PUF butterfly curve is shown. The cell has a bias towards S1 since the V_Q=V_{QB} line is above M1.

High error rates are one weakness of SRAM-based PUFs. The error rates of SRAM PUFs are generally higher than those of pure PUF circuits. The designers of pure PUF circuits have the freedom to enhance local mismatches and to limit the impact of noise. In many PUF designs, ECC is utilized to decrease the error rate. Since SRAM PUFs demonstrate high error rates as much as 10%, ECC cannot correct all PUFs. Many ECC schemes are too complex, making it practically infeasible in some microcontrollers.

2.5.4 Delay hardened PUF

Researchers [21] recently described a hybrid PUF by Intel that utilizes metastability in a bistable element along with delay variation to improve reliability. The hybrid PUF also counteracts aging by using burn techniques to purposely bias the PUF towards the favored state. The burn in technique uses high temperature and high voltage to achieve this. This technique shows a 13% reduction in unstable bits and can be with ECC, Temporal Majority Voting (TMV) and unstable bit masking to further improve the reliability.

2.5.4.1 Repetition code

Repetition code is a simple but very effective method of lowering the bit error rate (BER) in SRAMs. It uses a form of spatial majority voting as redundancy. This means that the more bits that are used in the voting process, the lower the error rate. First, an odd number of bits are XORed with the first bit in the group and the result is the code. Next, when the SRAM is given a challenge, the raw output is XORed with the code and a majority vote is made. If there are more 1s than 0s than the output is a 1 and vice versa [22]. The output will be wrong if a majority of the bits flip so a tradeoff exists between reducing BER and the number of bits needed for a single PUF bit.

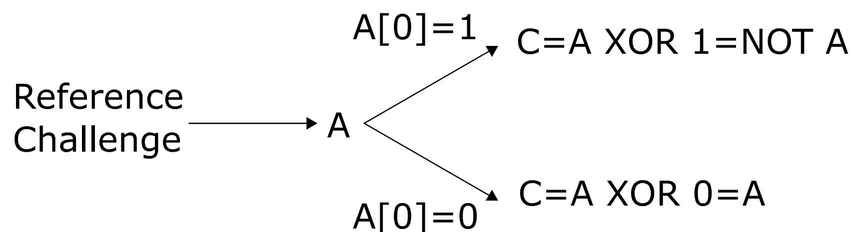

Figure 2.10 Repetition code generation

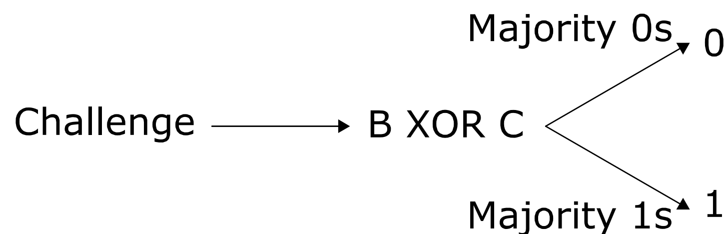

Figure 2.11 Repetition code usage

2.5.4.2 Temporal majority voting

Temporal majority voting [21] uses a counter to mitigate errors that occur over time due to noise. As shown in Figure 2.12 an N bit counter is used with a PUF cell and the value of the nth bit is the result of the TMV process after 2^{n+1}-1 cycles. If the nth bit is used this is called TMV(2^{n}+1) for n>0. For example, if the 2nd bit is used, there will be 3 cycles of voting and a decimal result of 3 or 2 have a 1 in the 2nd bit position meaning that the majority or cycles have chosen a 1.

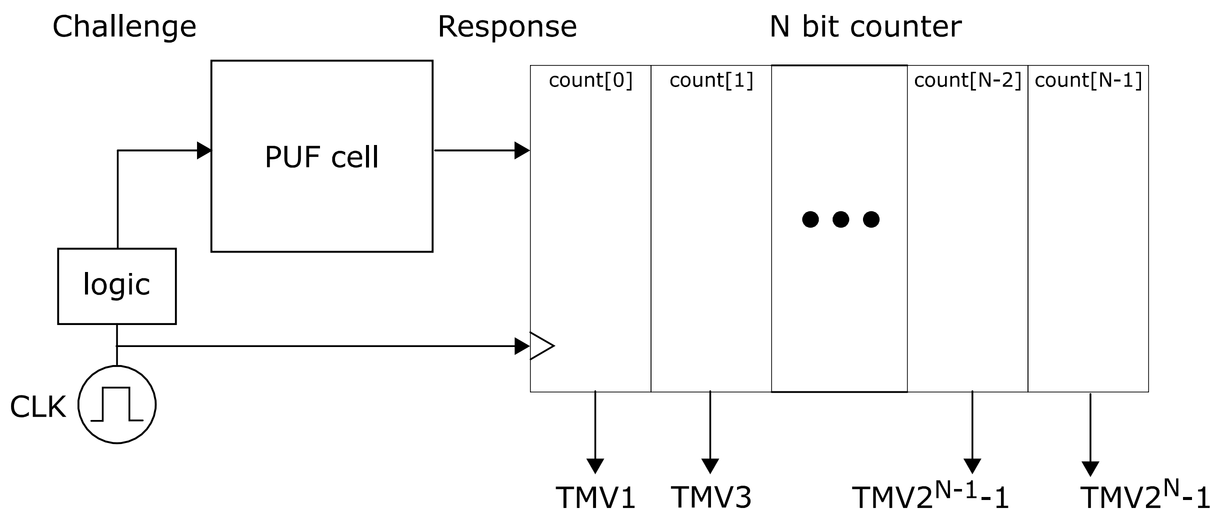

Figure 2.12 TMV scheme

2.6 Mismatch Properties of Transistors

In MOSFETs, there are two types of manufacturing variation, global and local. The local mismatch is derived from stochastic differences between nearby devices that cannot be controlled through production means.

Global mismatches originate from working procedures of a present day fabrication facility [23]. The global variation is brought on for instance by temperature gradients over the wafer amid annealing, by photoresist development, an etching process, or photolithographic process variations. There are layout techniques that aim to reduce this type of variation.

Local mismatch is important in creating the unique fingerprint of a PUF. In deep submicron technology, transistor gate oxide thickness is only a few nanometers thick and can vary up to 50% on a single device [24]. This affects carrier mobility and gate tunneling. In bulk CMOS, local mismatch is

influenced by Random dopant fluctuations (RDF), line edge roughness (LER), and polygate granularity [13].

There are also mismatches that occur over time and usage of devices. These are called temporal mismatches and can further be grouped in reversible and irreversible mismatches.

Local temperature shifts can occur when the distribution of current on a chip is uneven. This can cause areas with a high current to increase in temperature with respect to areas with less current. This can affect many characteristics such as drive current and leakage.

Negative bias temperature instability (NBTI) is a physical degeneration which brings about an increase in the V_t and the subthreshold slope and a lowering of the transconductance after some time of negatively biased transistors. Silicon-dioxide (SiO_2) is typically used as the oxide in MOSFETs. This contains dangling bonds Hydrogen is introduced to pacify these dangling bonds in SiO_2. Given a high enough temperature or negative bias, some of these hydrogen atoms will displace and some will form H_2 molecules. This causes gaps left in the SiO_2 that carriers must be filled, thus increasing the V_t. NBTI increases with growing temperature and with increasing negative gate to source voltage.

Hot Carrier Injection can occur when charge carriers are energetic enough to overcome energy barriers. A long mean free path gives carriers more acceleration time and also a high electric field will accelerate the carriers quicker. This can cause a leakage current to appear in the gate or channel and the high energy of the carriers can break Si-H bonds. This creates traps which will increase the V_t in a similar fashion to NBTI.[25] Pelgrom's model captures the effects of variation in a transistor into a simple relationship which says that the V_t variation is inversely proportional to the area of the transistor.

$$\sigma Vt \; \alpha \; \frac{\sqrt[4]{N}}{\sqrt{WL}}$$

Where N is the doping concentration, W and L are the length and width of the transistor. σVt is the standard deviation of the V_t. There is a weak dependence on doping concentration which means that the area is the major determinant in V_t variation.

Power supply variations can occur locally as well if the power grid is not designed evenly. This can be crucial to SRAM PUFs that use power on techniques for challenges.

2.7 6T Cell Background

The ubiquitous six transistor SRAM cell is composed of a back to back inverter with two NMOS access transistors. It is a high density cell and is volatile so power needs to be on in order to store data. P1 and N1 form the first transistor and P0 and N0 form the second transistor.

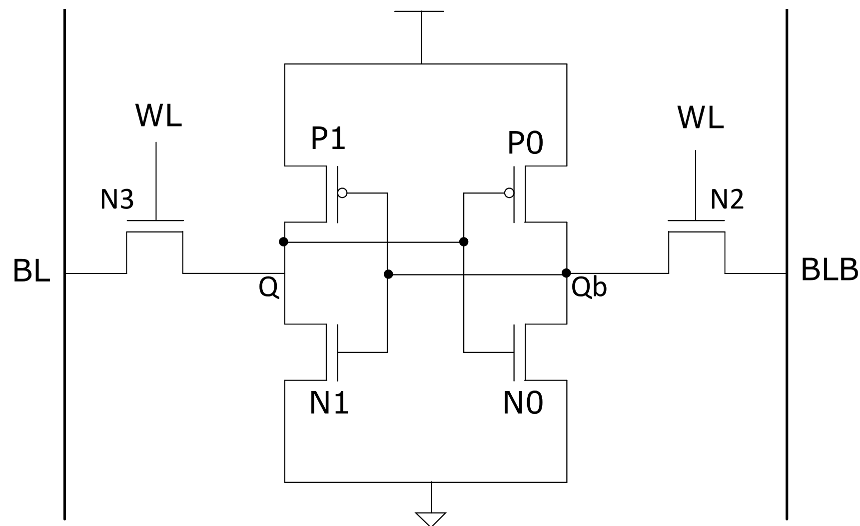

Figure 2.13 6T cell schematic

2.7.1 Read operation

To illustrate a read operation assume that Q is at V_{DD} while Qb is at 0V

1. BL and BLB are precharged to V_{DD}
2. The WL is activated and current flows though N2 and N0.
3. Qb sees a small rise in voltage as long as N0 is sized to be stronger than N2
4. A differential voltage is formed between the two bitlines since BL is still at V_{DD} and this differential is typically amplified by a sense amplifier.

2.7.2 Write operation

To illustrate a write operation assume that Q is at V_{DD} while Qb is at GND. We would like to write a 0 into the cell. Due to the read sizing constraint, the write cannot be accomplished through the pulldown NMOS transistor N1. Therefore it must be accomplished through P1.

BL is charged to 0 and BLB is charged to V_{DD}. When the WL is turned on N3 is sending charge into Q while P1 is trying to maintain the charge. The access transistor needs to be sized so that it has a

higher drive strength than the PMOS transistors. When Q has been pulled higher than the switching threshold of the right inverter, positive feedback latches the cell and the write is completed.

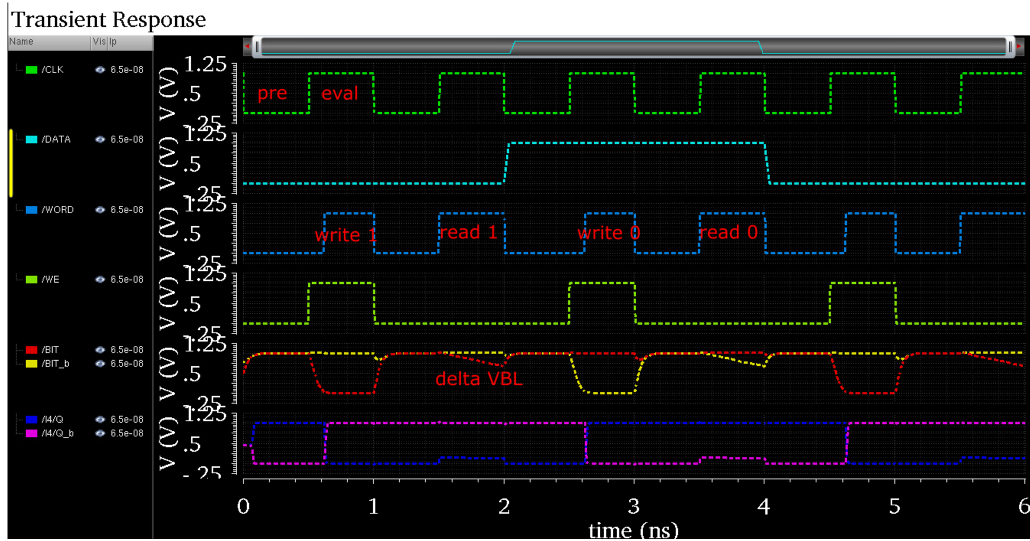

Figure 2.14 6T cell read write operations

2.8 8T Cell Background

The eight transistor cell was invented for low power and soft error robust SRAM applications. It is composed of four storage nodes as seen in Figure 2.15. [26]

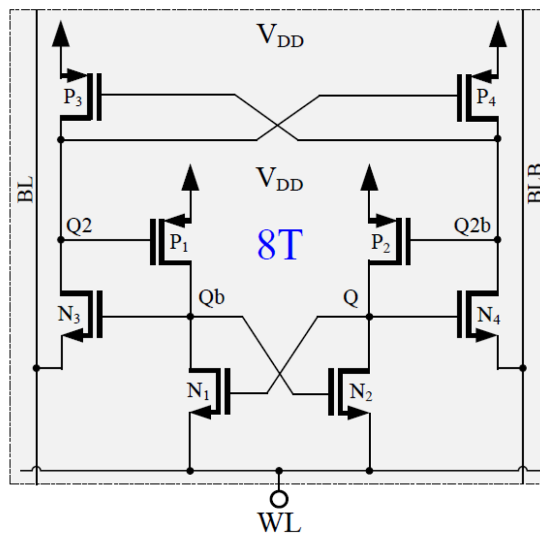

Figure 2.15 8T cell schematic

2.8.1 Read operation

To illustrate a read operation assume that Q and Q2 are at V_{DD} while Q2b and Qb are at 0V, BL and BLB are precharged to 0V.WL voltage needs to be higher than the threshold voltages of N3, but it cannot be too high or it will cause a destructive read. This creates a trade-off between read current and read stability. WL rises to 400mV which will cause Qb to rise to 400mV since Q is at V_{DD} and N1 is on. This turns on transistor N3 which will pulldown Q2 and cause a logic 0 degradation on BL. Marker V1 in Figure 2.16 illustrates this.

2.8.2 Write operation

To illustrate a write operation, assume that Q and Q2 are at V_{DD} while Q2b and Qb are at GND. We would like to write a 0 into the cell.

The WL is brought to 400mV to turn on N3. Since BLB is 1, N4 charges up Q2b until the V_{GS} of N4 is less than its threshold and turns off. While this is happening N3 is pulling down Q2 which occurs quickly once P3 turns off and the feedback mechanism takes over. Marker V2 in Figure 2.16 illustrates this.

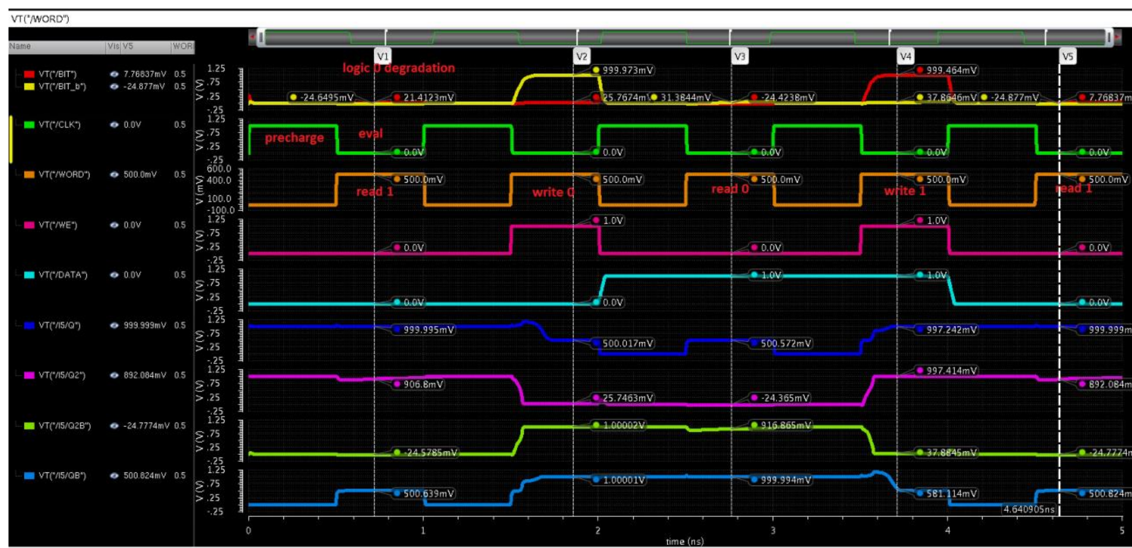

Figure 2.16 8T read/write operation

2.9 Noise

Noise can be categorized into white and colored noise. White noise is flat in the frequency domain meaning it has the same noise power across all frequencies. Colored noise has a frequency component in its noise power. Variation in V_{DD} across the chip and across time can also affect PUF reliability.

Thermal noise, I_{noise}^2, is a form of white noise that is generated within the channel of the MOSFETs. It is caused by the random motion of electrons which increases with temperature. It is modeled as a current source in parallel with the transistor. γ is a coefficient that changes between processes and is empirically found and is typically 2/3 for long channel devices.[27] T represents temperature in kelvin, g_m is the transconductance of the MOSFET and k is boltzman's constant.

$$I_{noise}^2 = 4kT\gamma g_m$$

Shot noise occurs when charge carriers need to surpass a potential barrier within transistors when moving from the source area into the channel. Since the intersection of the barrier is a stochastic process, the current flow is "noisy." The power spectral density of shot noise, I^2_{shot} ?, concerning the noise current is [28]:

$$I_{shot}^2 = 2qI$$

Where q is the charge on the carrier and I is the average current

Flicker noise is believed to be caused by multiple sources including the silicon oxide interface where dangling bonds exist. As carriers move close to the interface, some recombine and regenerate due to the extra energy states present. The following equation approximates flicker noise in MOSFETs but real flicker noise is more complex [27]. Where K is a coefficient dependent on V_{GS}, C_{ox} is the oxide capacitance, L and W are the length and width of the transistor directly and f is frequency.

$$V_{noise}^2 = \frac{K}{C_{ox}WLf}$$

Random Telegraph Noise (RTN) occurs when a device is so small that a single trap can have a significant impact on transistor behavior. A single trapped charge has been observed to impact the drain current by as much as 10% [28]. This type of noise occurs at a low frequency which can be a problem for off the shelf SRAM PUF that need very slow power up sequences to operate.

Chapter 3
Methodology

The two memory structures, 8T, and 6T memory cells were simulated. There are also two different challenges between analyzed called V_{DD} manipulation and GND manipulation. The main focus of the simulations is on the PUF reproducibility/reliability but some insight on the uniqueness is gained. The first two tests aim to create a new metric for PUF reliability called $\Delta V_{GS.}$ The tools being used for simulations is Cadence Virtuoso which enables schematic capture and editing, waveform viewing and simulations. The Spectre simulation engine is used as it allows simultaneous use of transient noise and Monte Carlo. Conservative settings were used due to its higher accuracy as the other preset settings showed issues with convergence The testbench consists of a single instance of a cell and ideal signals and switches to control the cell. The TSMC 65 nm general purpose models and macro model devices were used for Monte Carlo. The TT corner was used since Monte Carlo was already taking process into account.

Each Monte Carlo run contains 1000 unique cells/samples and is challenged 100 times. Cells that always respond with logic 1 or always respond with logic 0 are considered perfect cells. The number of perfect cells out of 1000 is the metric used to qualify the PUF. This chapter explains the testbenches as well as the PUF schemes being used

3.1 Transient Noise

Transient noise is added to the simulation or else the results would be perfect. The noise is generated from the MOSFETs based on their models. The important settings in Spectre are the NFmax and NF_{min}. The NF_{max} dictates the maximum bandwidth of the noise and the noise power above this frequency is 0. The NF_{min} is the lowest frequency that considers coloured noise so any frequencies below NF_{min} only contain white noise. A high enough NF_{max} is necessary and must contain the region of interest which is the rising/falling edge of the challenge. In the real world NF_{max} would be infinity but since simulations are really discrete systems, simulation time step is limited to 1/ (2*NF_{max}) or the shortest time step determined by Spectre. Previous simulations used an NF_{max} that was 10x the V_{DD} signal frequency but resulted in highly optimistic results that did not have noise.

3.2 V_{GS} Dependency

During ramp up, all devices are in subthreshold and mismatch between pullup PMOS and pulldown NMOS devices determines what start up value is favored. In the 6T SRAM cell, the ramp up initially causes nodes Q and QB to rise equally before splitting. Our theory suggests that during this time, the devices with larger V_{GS} determine the start up value. Sizing accordingly, it would be possible to isolate dependence of mismatch to either N1 and N0 or P1 and P0.

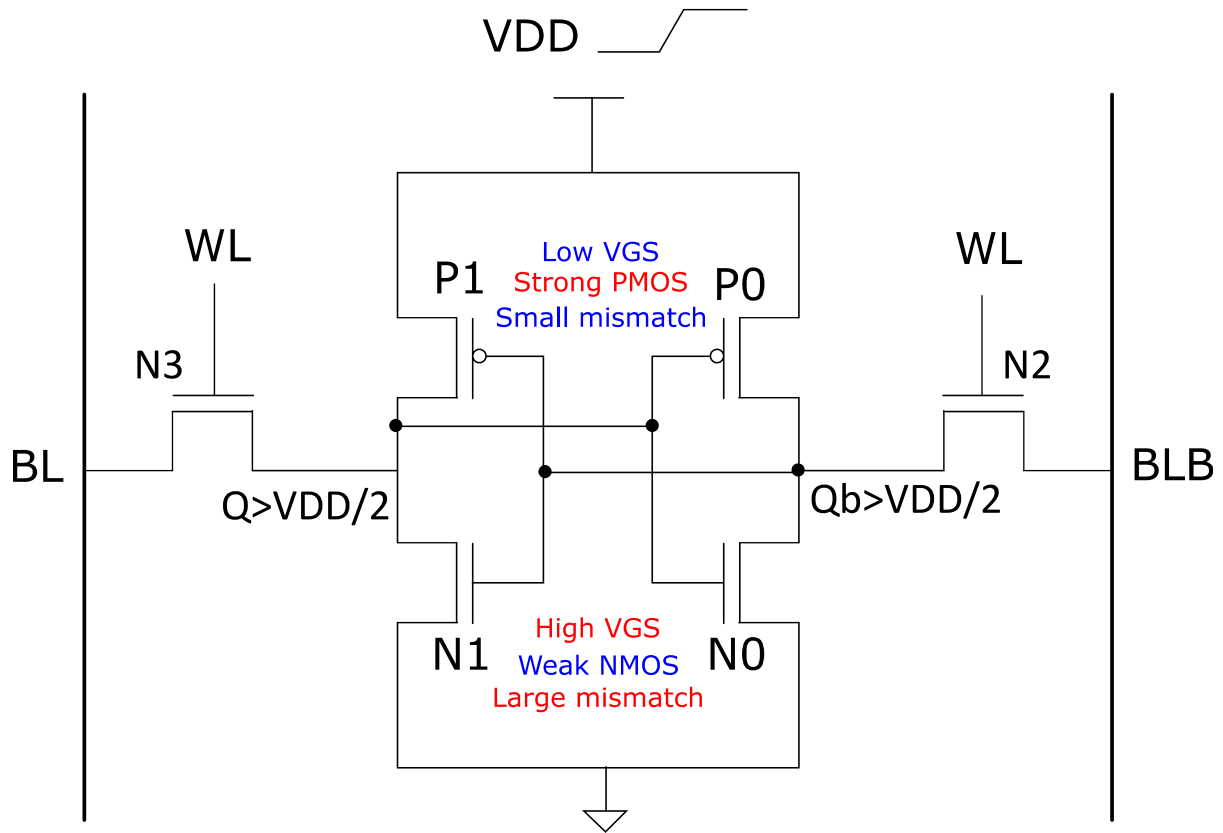

Figure 3.1 V_{gs} dependency analysis

Figure 3.1 shows a case where the PMOS is sized such that they are stronger than NMOS devices. As V_{DD} ramps up, V_Q and V_{Qb} will be closer to V_{DD} than ground. This implies that the V_{GS} of NMOS devices is larger than PMOS devices and that NMOS mismatch determines PUF reliability.

3.3 Isolated Mismatch Experiment

In this experiment, local mismatch from certain devices is disabled meaning that their threshold voltage is nominal. The results can indicate which transistors play a dominant role during PUF operation.

Mismatch is disabled for NMOS or PMOS transistors and these cases will be called ideal NMOS and ideal PMOS respectively.

As V_{DD} is rising up in the 6T SRAM cell, Q and QB initially rise together at approximately the same rate. If the voltage of Q and QB are lower than $V_{DD}/2$, this implies that PMOS devices have a larger V_{GS} and determine PUF reliability.

3.4 6T V_{DD} Manipulation

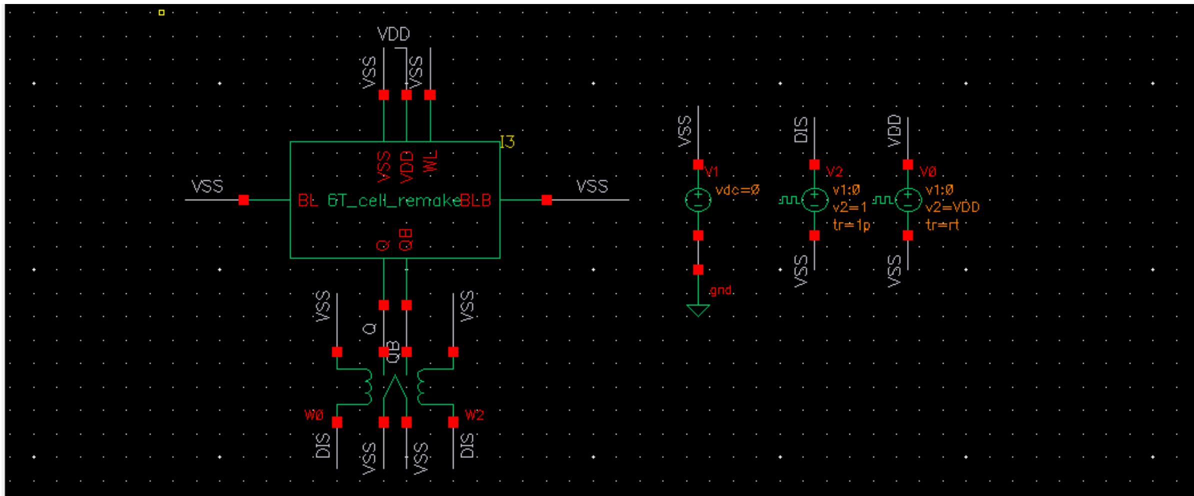

Figure 3.2 6T VDD manipulation PUF testbench schematic

In the 6T V_{DD} manipulation scheme, V_{DD} is ramped up to issue the challenge. The bit lines and word line of the 6T cell are connected to ground to isolate it. No read is performed so the measurements are done by probing the storage nodes of the SRAM cell. Switches are used to perform a purge of all charge within the storage nodes at the beginning of a PUF cycle. This is done to decrease simulation time since the time for a complete discharge is very long. The timing of the discharge is made so that it does not overlap the ramp up phase which prevents interference with the results. Also because we are concerned with the ramp up phase, V_{DD} fall time is sped up to decrease simulation time.

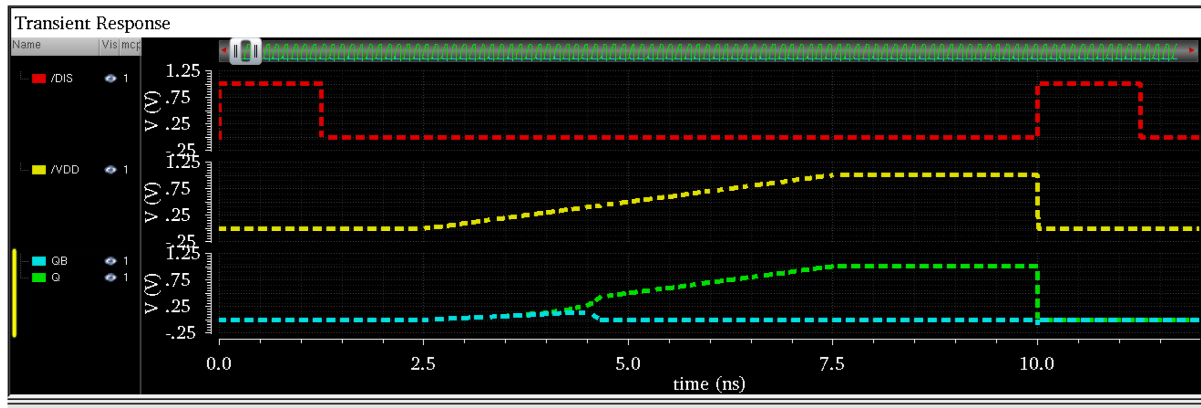

Figure 3.3 6T V_{DD} manipulation PUF cycle

The PUF cycle begins with a discharge of the internal nodes with DIS (red) going high and turning on the switches to discharge Q and QB (cyan and green). Here the nodes are already discharged since it is the start of the simulation but the second rising edge of DIS shows the discharging taking place. Next V_{DD} (yellow) starts to rise and so do Q and QB at an equal rate in relation to each other. Q and QB eventually diverge when they pass the threshold voltage and the cell latches to one of the stable states. In this example, it latches towards a logic 1 (Q=1).

3.5 6T GND Manipulation

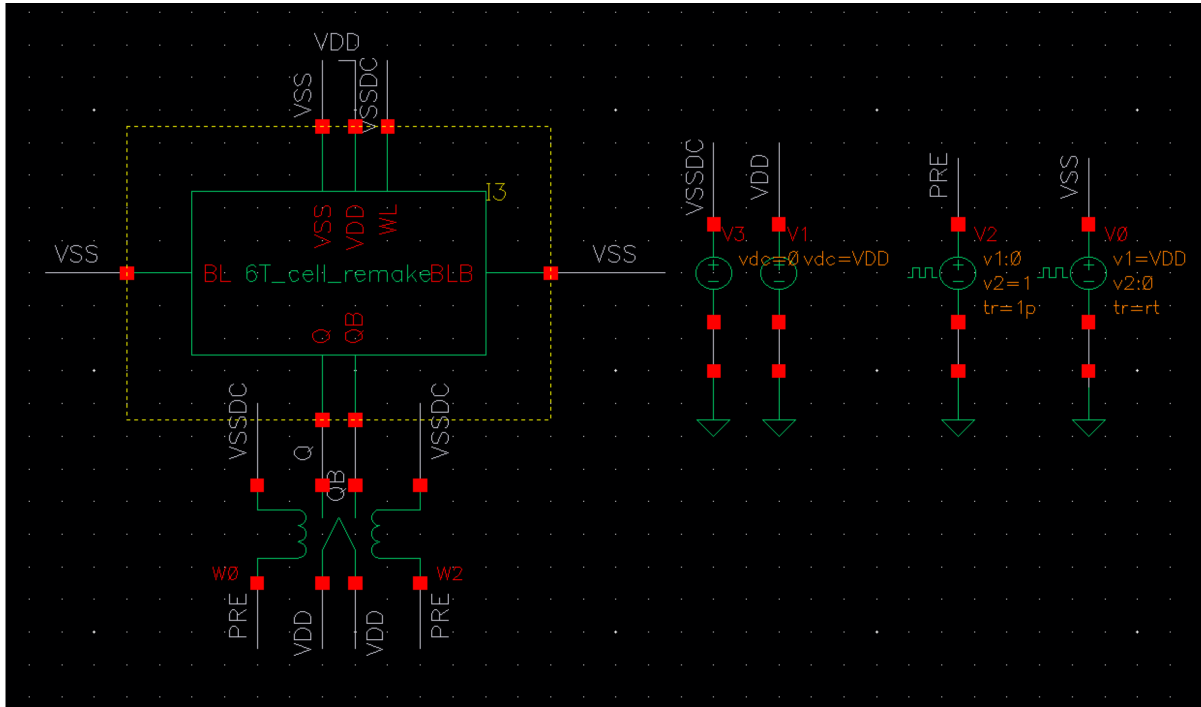

Figure 3.4 6T GND manipulation PUF testbench schematic

In the 6T GND manipulation scheme, V_{SS} is ramped down while V_{DD} is kept high. The bit lines are connected to V_{SS} while the word line of the cell is connected to ground to isolate it. This is to minimize leakage through the access transistor as well as isolating the cell. Switches are used to perform a precharge of the storage nodes at the beginning of a PUF cycle. This is done to ensure a strong 1 is present within the cell since passing V_{DD} through NMOS transistors results in a logic 1 degradation. The timing of the precharge is made so that it does not overlap the ramp up phase. Also because we are concerned with the ramp down phase, V_{SS} rise time is sped up to decrease simulation time.

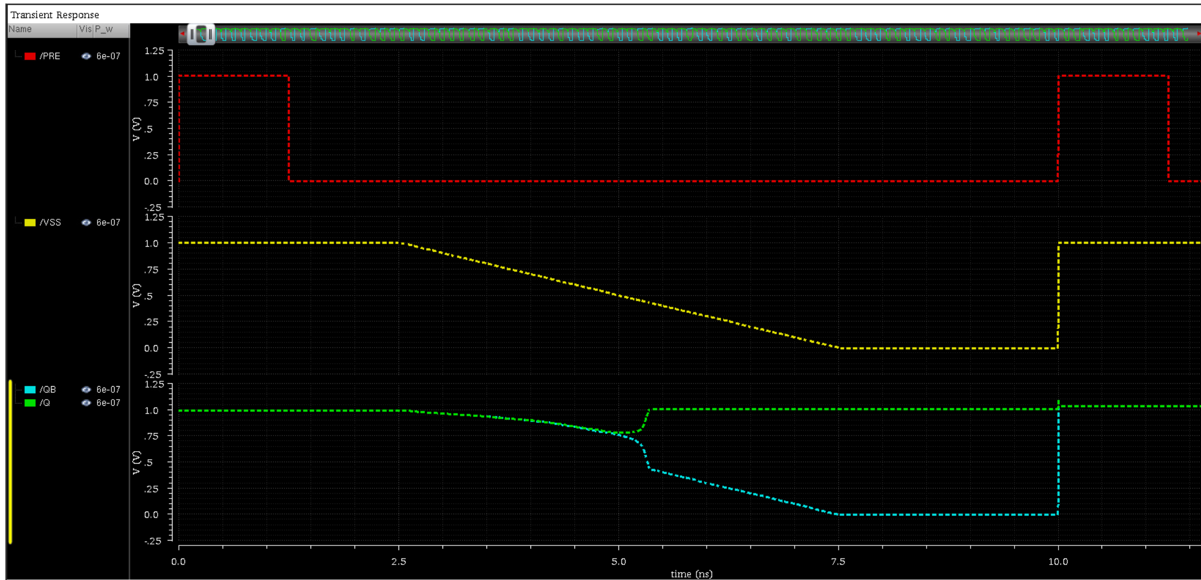

Figure 3.5 6T GND manipulation PUF cycle

The PUF cycle begins with a precharge of the internal nodes with PRE (red) going high and turning on the switches to precharge Q and QB (cyan and green) to V_{DD}. Next V_{SS} (yellow) starts to fall and so do Q and QB at an equal rate in relation to each other. Q and QB eventually diverge when they pass the threshold voltage and the cell latches to one of the stable states. In this example it latches towards a logic 1 (Q=1).

3.6 8T V_{DD} Manipulation

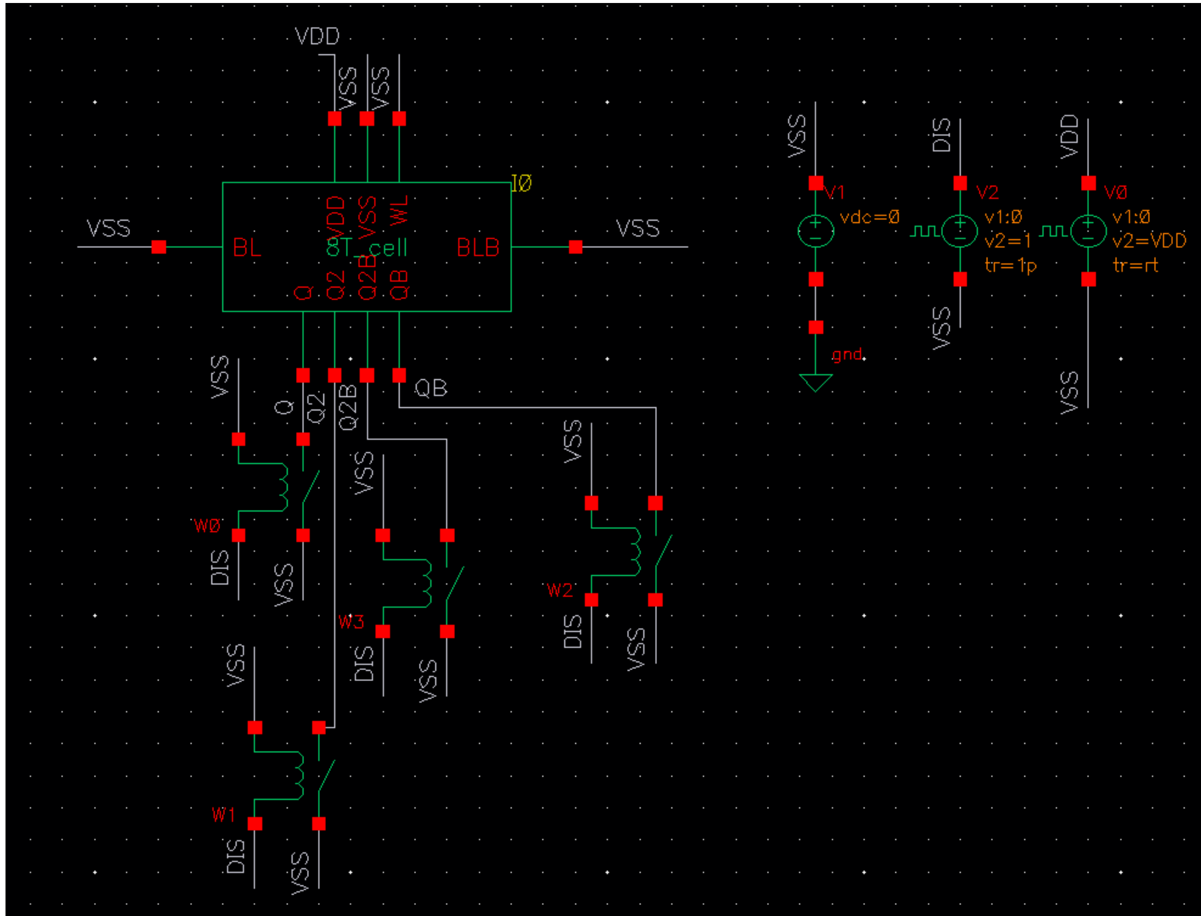

Figure 3.6 8T V_{DD} manipulation PUF testbench schematic

In the 8T V_{DD} manipulation scheme, V_{DD} is ramped up to issue the challenge. The bit lines and word line of the 8T cell are connected to ground to isolate it. No read is performed so the measurements are done by probing the storage nodes of the 8T SRAM cell. Switches are used to perform a purge of all charge within the storage nodes at the beginning of a PUF cycle. This is done to decrease simulation time since transient noise significantly increases the duration of the simulation. The timing of the discharge is made so that it does not overlap the ramp up phase. Also because we are concerned with the ramp up phase, V_{DD} fall time is sped up to decrease simulation time.

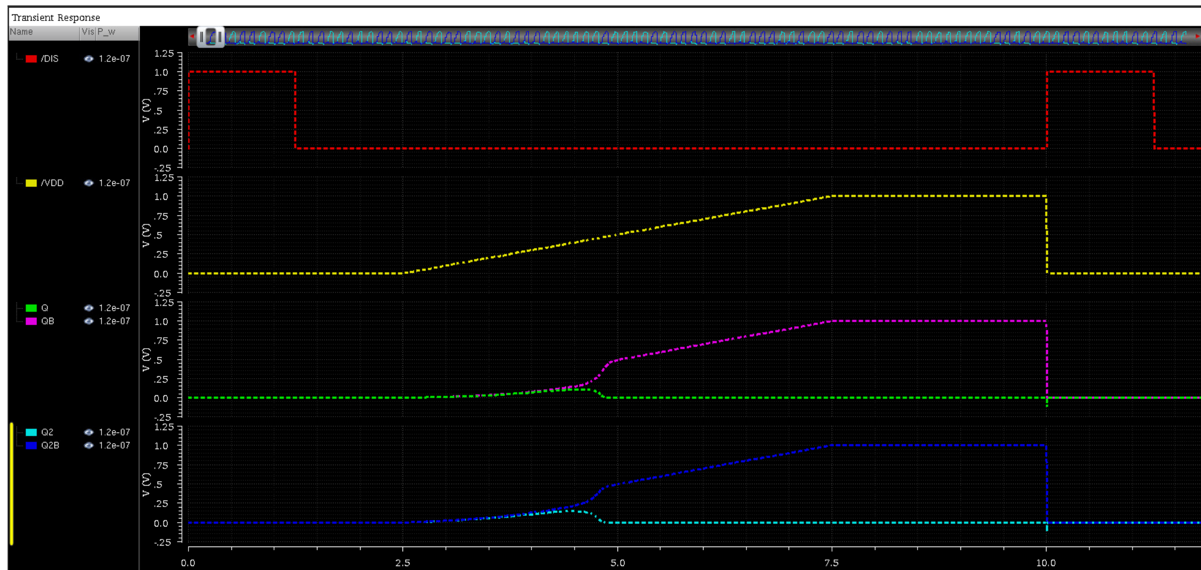

Figure 3.7 8T V_{DD} manipulation PUF cycle

The PUF cycle begins with a discharge of the internal nodes with DIS (red) going high and turning on the switches to discharge Q, QB, Q2 and Q2B (green , purple, cyan and blue respectively). Here the nodes are already discharged since it is the start of the simulation but the second rising edge of DIS shows the discharging taking place. Next V_{DD}(yellow) starts to rise and so do the storage nodes at an equal rate in relation to each other. The storage nodes eventually diverge when they pass the threshold voltage and the cell latches to one of the stable states. In this example, it latches towards a logic 0 (Q=Q2=0).

3.7 8T GND Manipulation

Figure 3.8 8T GND manipulation PUF testbench schematic

In the 8T GND manipulation scheme, V_{SS} is ramped up while V_{DD} is kept high. The bit lines are connected to V_{SS}. One main difference between this and the 6T GND manipulation testbench is that the wordline is now ramped as well since it is connected to the source instead of the gate. Switches are used to perform a precharge a strong 1 at the beginning of a PUF cycle. The timing of the precharge is made so that it does not overlap the ramp up phase.

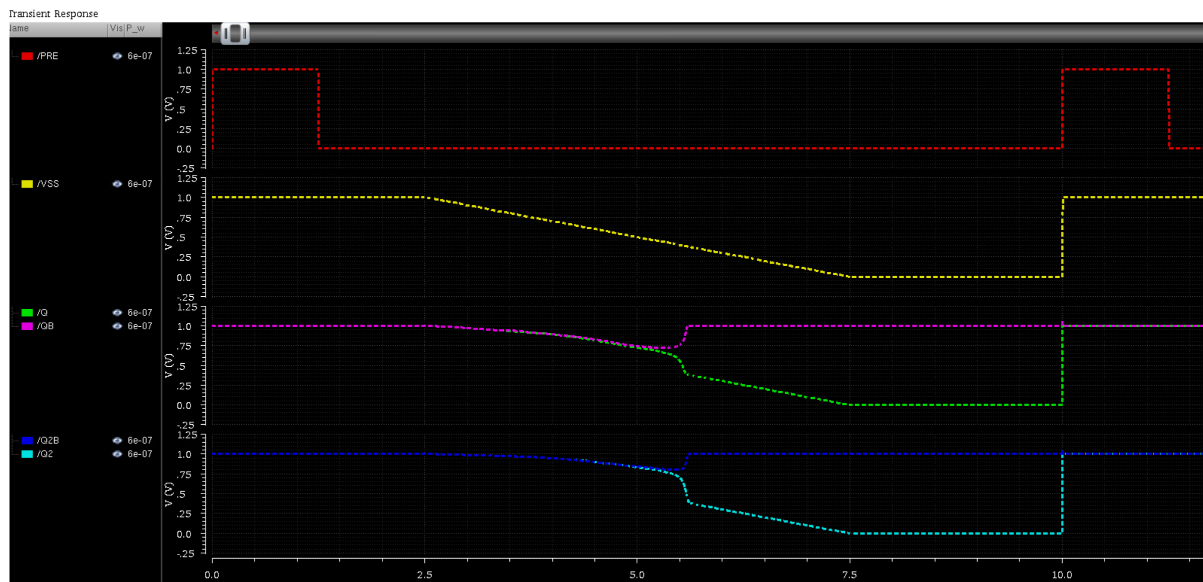

Figure 3.9 8T GND manipulation PUF cycle

The PUF cycle begins with a precharge of the internal nodes with PRE (red) going high and turning on the switches to discharge Q, QB, Q2 and Q2B (green , purple, cyan and blue respectively). Here the nodes are initially high since V_{SS} is also initially high. The simulator assumes this is the state of the system for an infinite amount of time in order to calculate the initial conditions. Next V_{DD}(yellow) starts to fall and so do the storage nodes at an equal rate in relation to each other. The storage nodes eventually diverge when they pass the threshold voltage and the cell latches to one of the stable states. In this example, it latches towards a logic 0 (Q=Q2=0).

3.8 Split-V_{DD} Experiment

With the 8T SRAM cell, it is possible to power up using two ramp signals with a delay between them. As seen in figure **Error! Reference source not found.**, outer PMOS devices P3 and P4 are controlled by V_{DD1} and inner PMOS devices P1 and P2 are controlled by V_{DD2}. By introducing a delay between V_{DD1} and V_{DD2}, the SUV values can be determined by inner or outer devices. Since SRAM PUFs are considered weak PUFs since they only have one challenge, this method can increase the number of challenges by modifying the delay between ramp ups and the order.

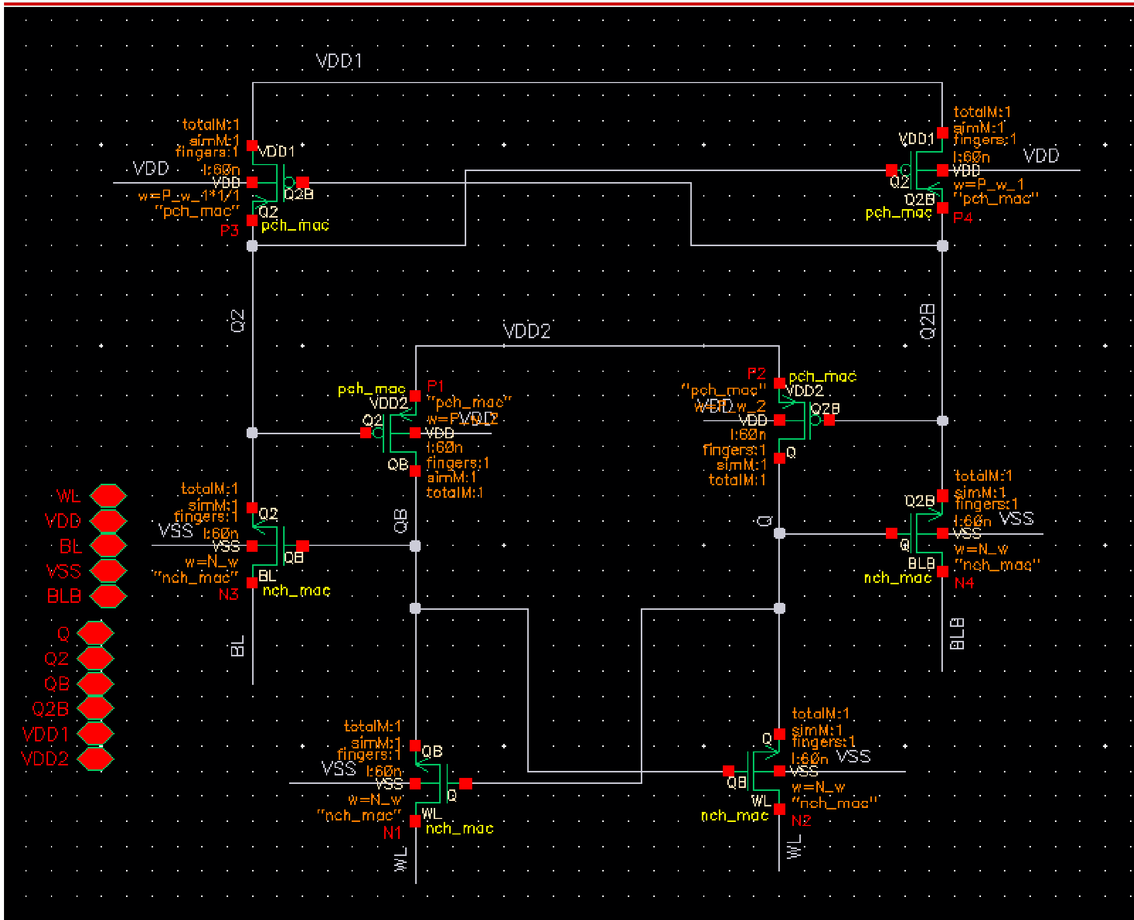

Figure 3.10 Split V_{DD} cell

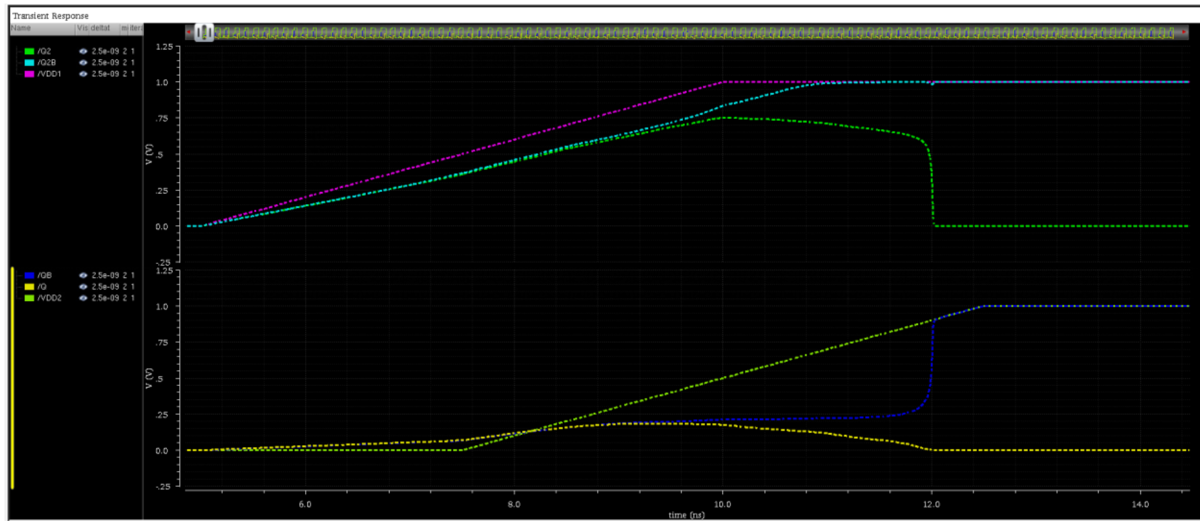

Figure 3.11 Split V_{DD} PUF challenge V_{DD1} ramps up first

Here, V_{DD1} (purple) for the outer PMOS devices ramp up first. V_{DD2} begins to rise 2.5ns after V_{DD1} begins to rise. What is interesting the splitting of Q2 and Q2B (green and cyan) is very gradual and does not latch until V_{DD2} is almost finished ramping up. This is because N3 and N4 do not form a latch and need to wait until N1 and N2 latches Q and QB (yellow and blue)

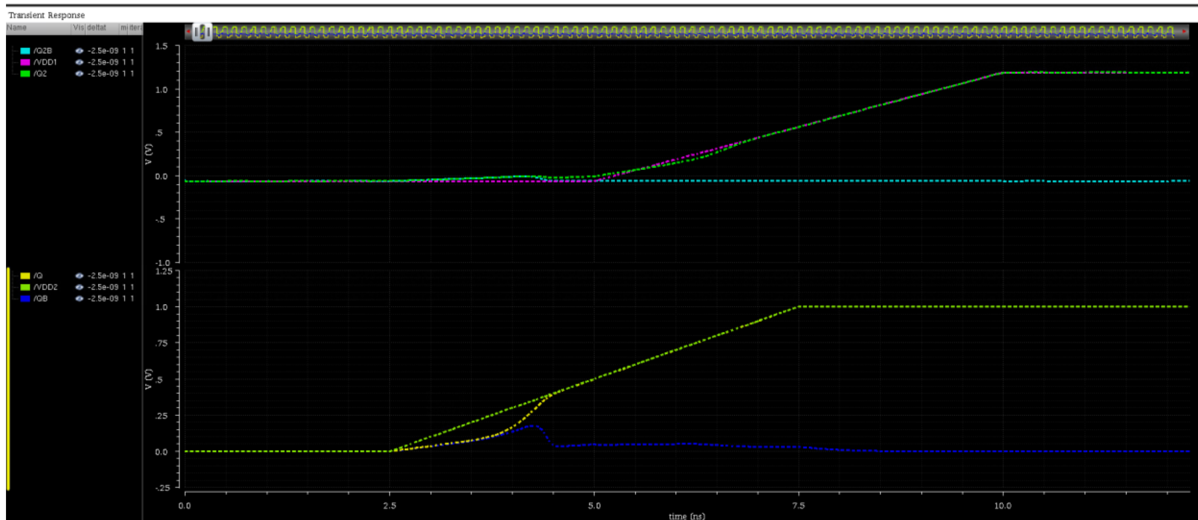

Figure 3.12 Split V_{DD} PUF challenge V_{DD2} ramps up first

Here, as V_{DD2} rises Q and QB rise together and split very quickly due to the presence of a half latch. Since V_{DD1} has not risen yet, Q2 and Q2B are close to V_{SS} which maximizes the V_{GS} of PMOS transistors P1 and P2.

3.9 Simulation settings

These settings are the baseline settings that are used unless otherwise specified:

Table 3.1 Simulation settings

Setting	Nominal Value
Period	20 ns
Rise time	5 ns
Cycles	100 cycles
V_{DD}	1 V
NF_{max}	2 GHz
NF_{min}	10 kHz
T	27C
Transistor width (all)	200 nm

Chapter 4

Simulation Results and Comparative analysis

4.1 V_{GS} Dependency Analysis

During the challenge phase, the storage node voltages will be equal for some time before latching. It is hypothesized that while the value of V_{GS} while the storage nodes are still equal can affect PUF reliability. Nominal simulations recorded V_{GS} in relation to the V_{DD} ramp-up to predict which devices have a higher influence on start up values. We will call the devices with a higher influence the dominant devices.

4.1.1 V_{DD} manipulation

For V_{DD} manipulation in 6T, the storage nodes are shown since V_{GS} of the pulldown devices is simply the voltage of the storage nodes themselves and the V_{GS} of the pullup devices are V_{DD} minus the storage node voltage. This means that if it is above $V_{DD}/2$ then NMOS devices are dominant and vice versa.

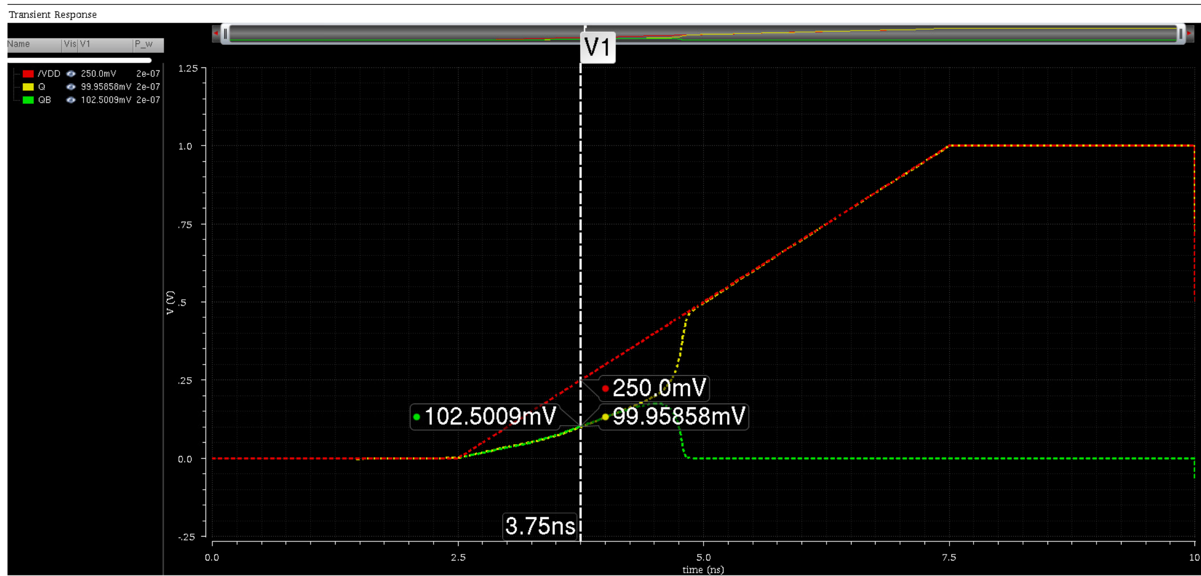

Figure 4.1 6T 200nm NMOS 200nm PMOS

In this simulation, observe that the voltages of Q and QB (yellow and green) are below $V_{DD}/2$ (125mV). This makes sense since NMOS and PMOS are sized equally but NMOS transistors have higher

mobility so the cell is better at keeping the node low during power up. Therefore this predicts that PMOS transistor mismatch is dominant.

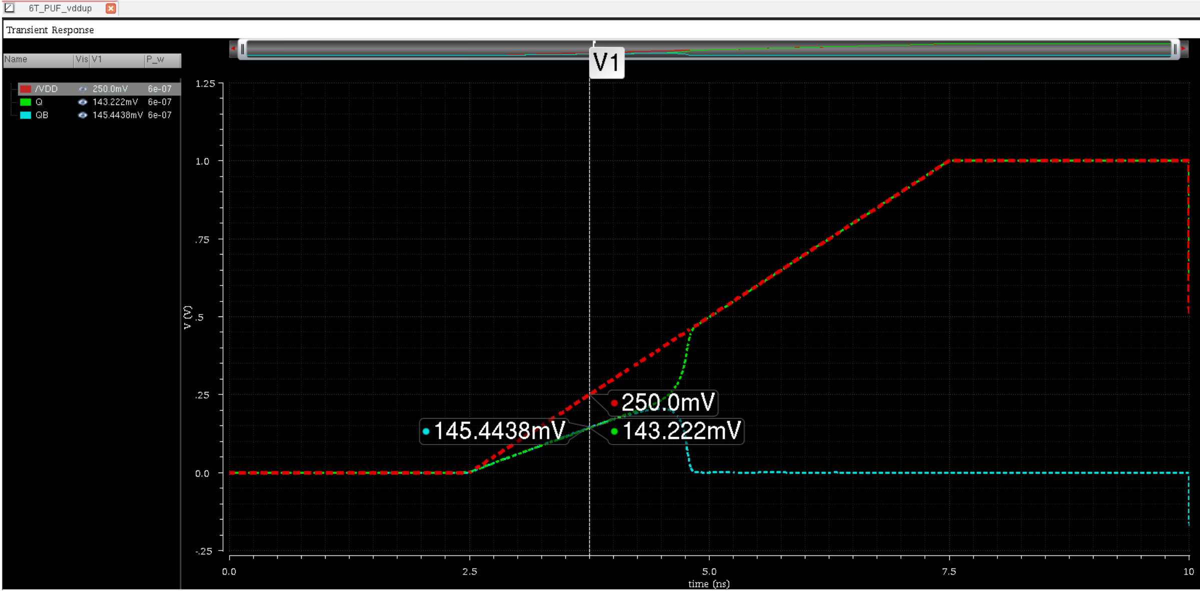

Figure 4.2 6T 200nm NMOS 600nm PMOS

In this simulation, observe that the voltages of Q and QB (green and cyan) are above $V_{DD}/2$ (125mV). PMOS devices here are 3x the size of NMOS devices. Therefore this predicts that NMOS transistor mismatch is dominant.

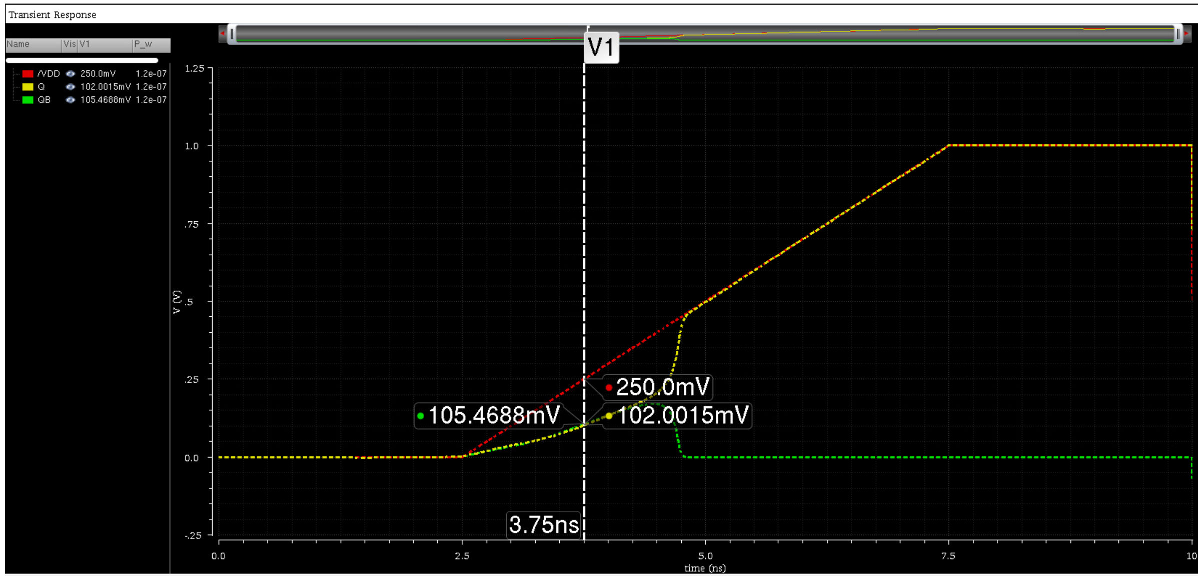

Figure 4.3 6T 120nm NMOS 120nm PMOS

With minimum sized devices, the voltage of the storage nodes is very similar to the case with 200nm devices. Again, PMOS devices are dominant due to the larger V_{GS}.

For 8T SRAM cell waveforms, the V_{GS} of each device is plotted and there are only V_{GS} plots since each device shares a gate with another device of the same type.

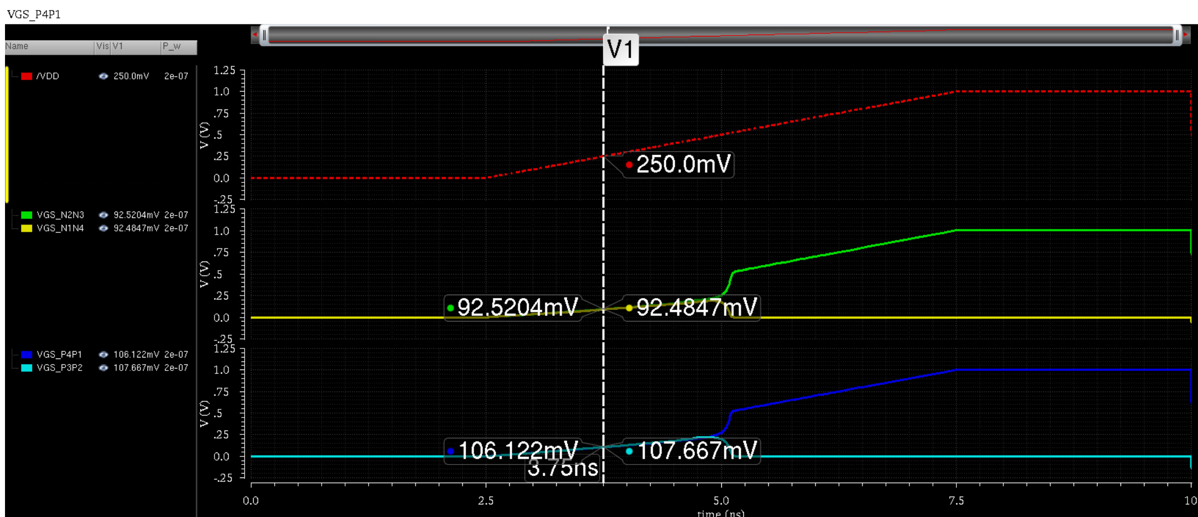

Figure 4.4 8T 200nm NMOS 200nm PMOS

During ramp up, the V_{GS} of PMOS transistors (cyan and blue) is larger than the V_{GS} of NMOS devices (green and yellow). This is due to a stronger NMOS device that sinks more current than the PMOS sources. This predicts that PMOS mismatch will be dominant

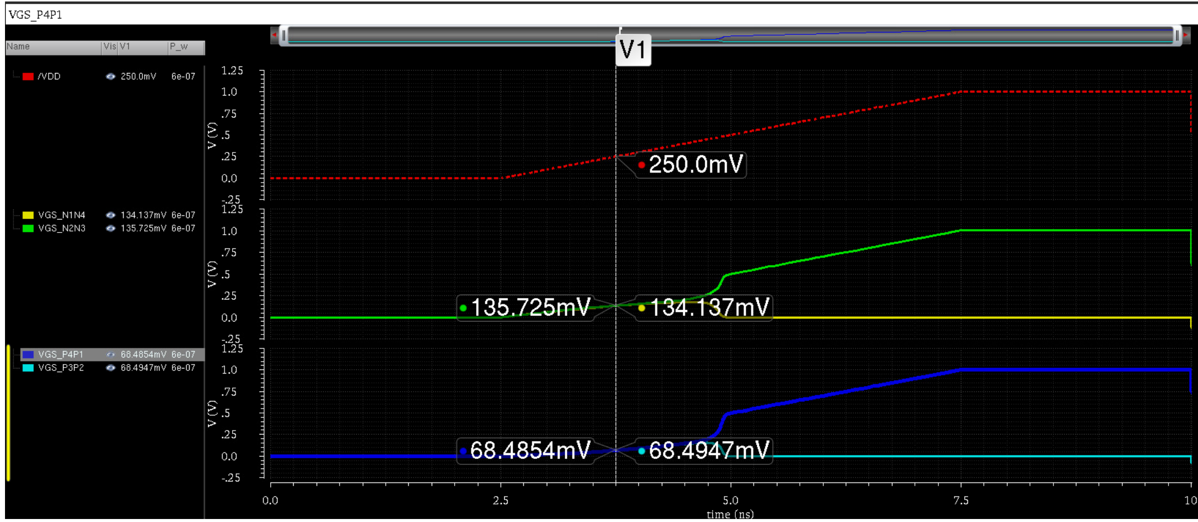

Figure 4.5 6T 200nm NMOS 600nm PMOS

During ramp up, the V_{GS} of NMOS transistors (green and yellow) is larger than the V_{GS} of PMOS devices (cyan and blue). This is due to a stronger PMOS device that sources more current than the NMOS sinks. This predicts that PMOS mismatch will be

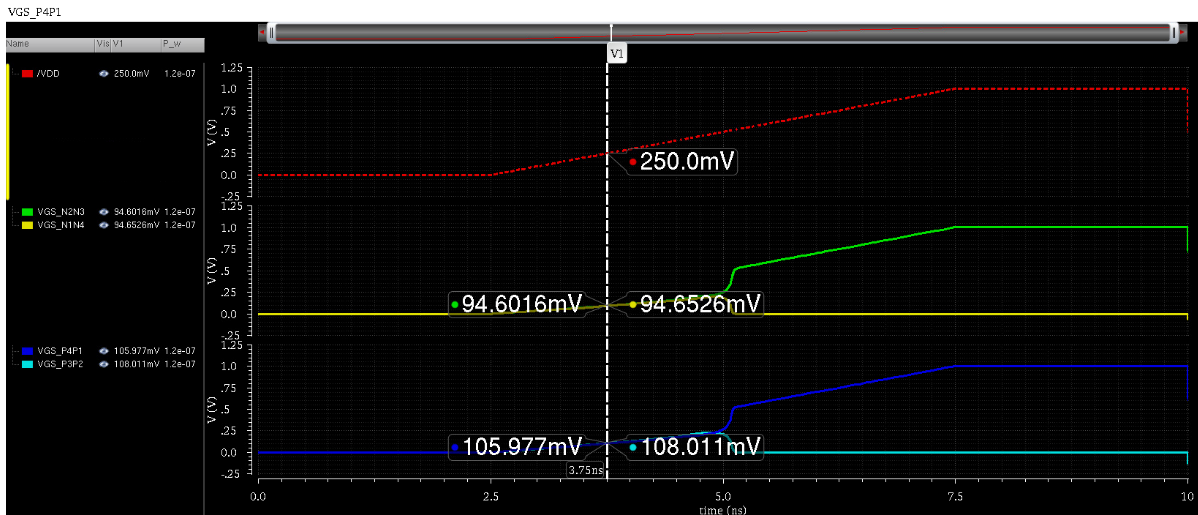

Figure 4.6 8T 120nm NMOS 120nm PMOS

During ramp up, the V_{GS} of PMOS transistors (cyan and blue) is larger than the V_{GS} of NMOS devices (green and yellow). This is due to a stronger NMOS device that sinks more current than the PMOS sources. This predicts that PMOS mismatch will be dominant

4.1.2 GND manipulation

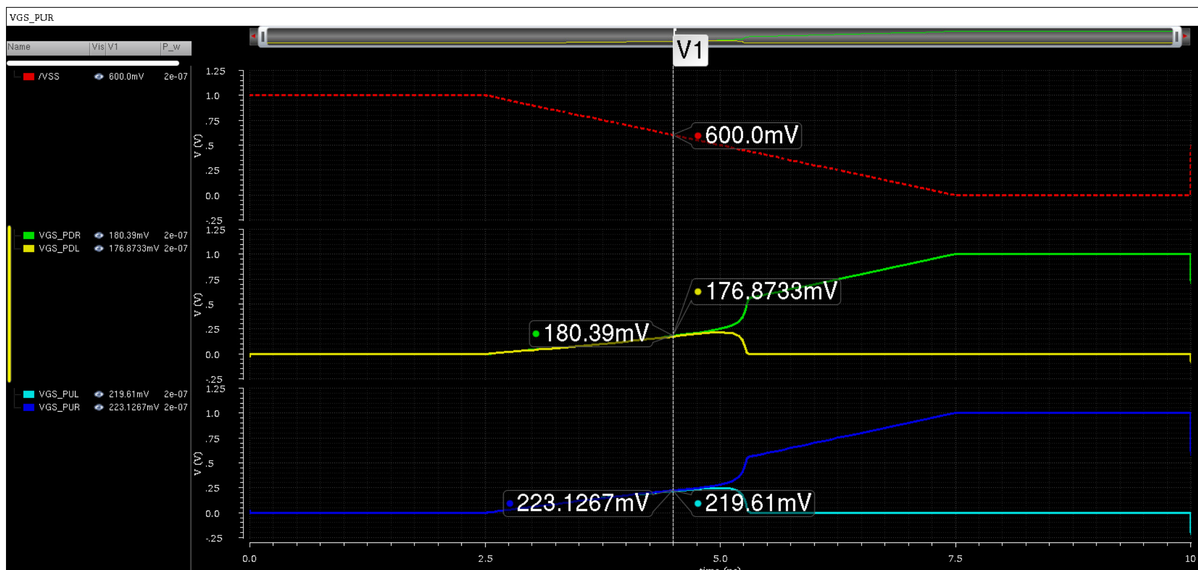

Figure 4.7 6T 200nm NMOS 200nm PMOS

During the ramp down of V_{SS}(red), the V_{GS} of PMOS transistors (cyan and blue) is larger than the V_{GS} of NMOS devices (green and yellow). This is due to a stronger NMOS device that sinks more current than the PMOS sources. This predicts that PMOS mismatch will be dominant

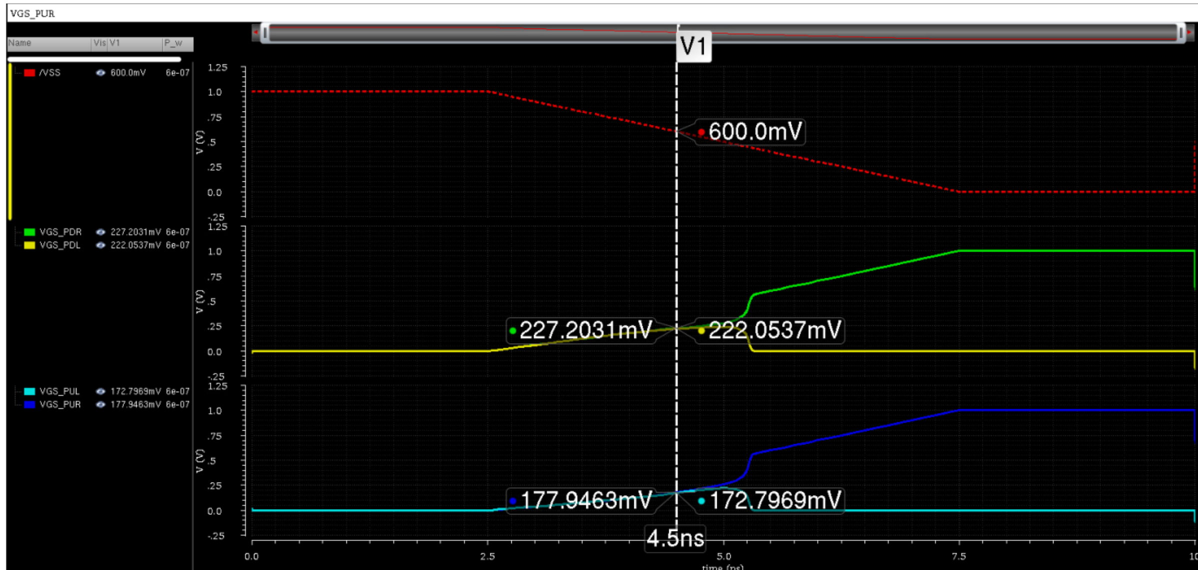

Figure 4.8 6T 200nm NMOS 600nm PMOS

During the ramp down of V_{SS}(red), the V_{GS} of NMOS transistors (green and yellow) is larger than the V_{GS} of PMOS devices (cyan and blue). This is due to a stronger PMOS device that sources more current than the NMOS sinks. This predicts that NMOS mismatch will be dominant

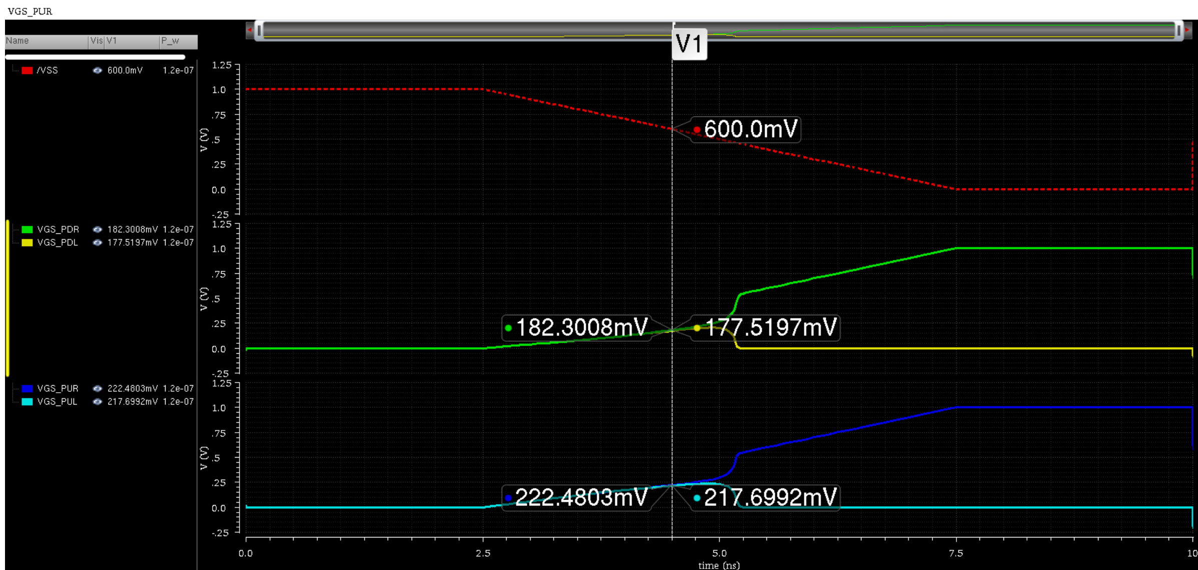

Figure 4.9 6T 120nm NMOS 120nm PMOS

This case is very similar to the first 6T GND manipulation case due to the equal sizing of NMOS and PMOS devices. Therefore it is predicted that PMOS mismatch will be dominant.

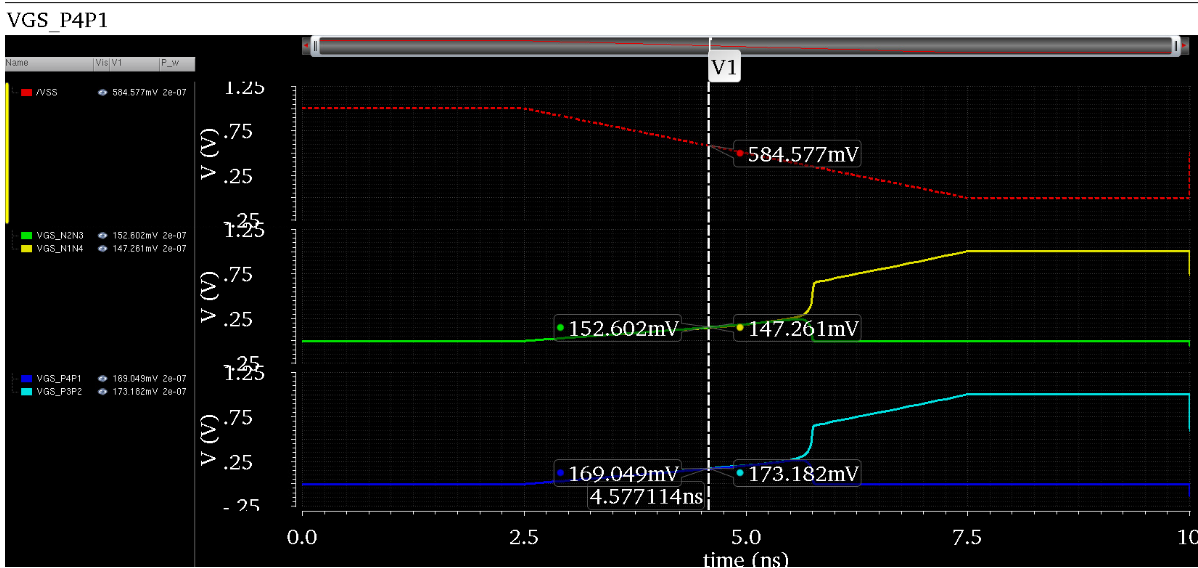

Figure 4.10 8T 200nm NMOS 200nm PMOS

During the ramp down of V_{SS} (red), the V_{GS} of PMOS transistors (cyan and blue) is larger than the V_{GS} of NMOS devices (green and yellow). This is due to a stronger NMOS device that sinks more current than the PMOS sources. This predicts that PMOS mismatch will be dominant

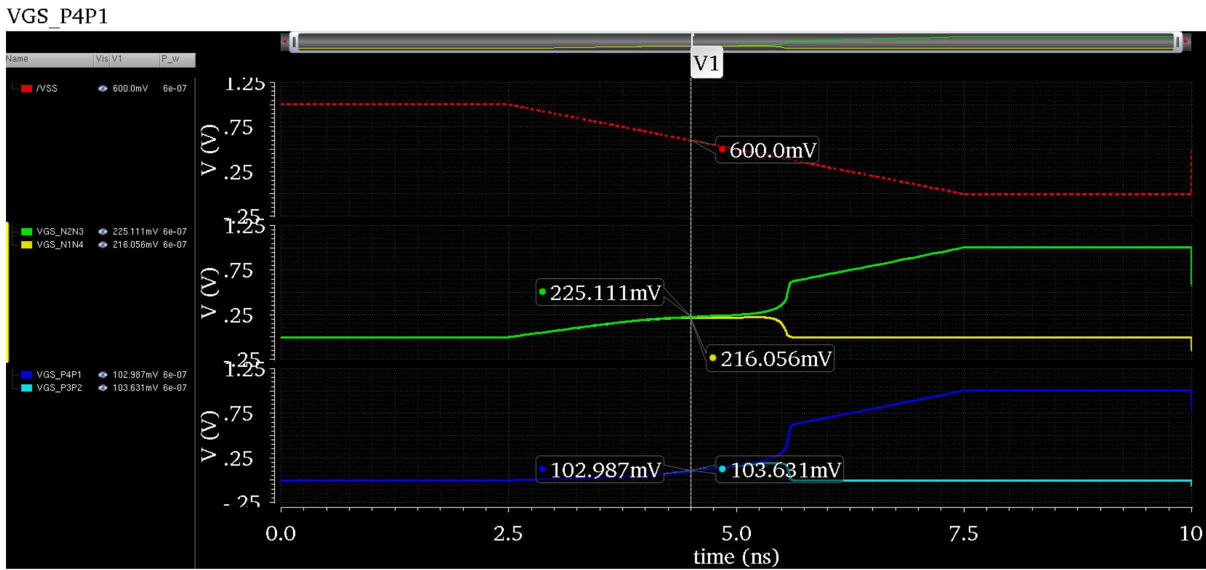

Figure 4.11 8T 200nm NMOS 600nm PMOS

During the ramp down of V_{SS} (red), the V_{GS} of NMOS transistors (green and yellow) is larger than the V_{GS} of PMOS devices (cyan and blue). This is due to a stronger PMOS device that sources more current than the NMOS sinks. This predicts that NMOS mismatch will be dominant

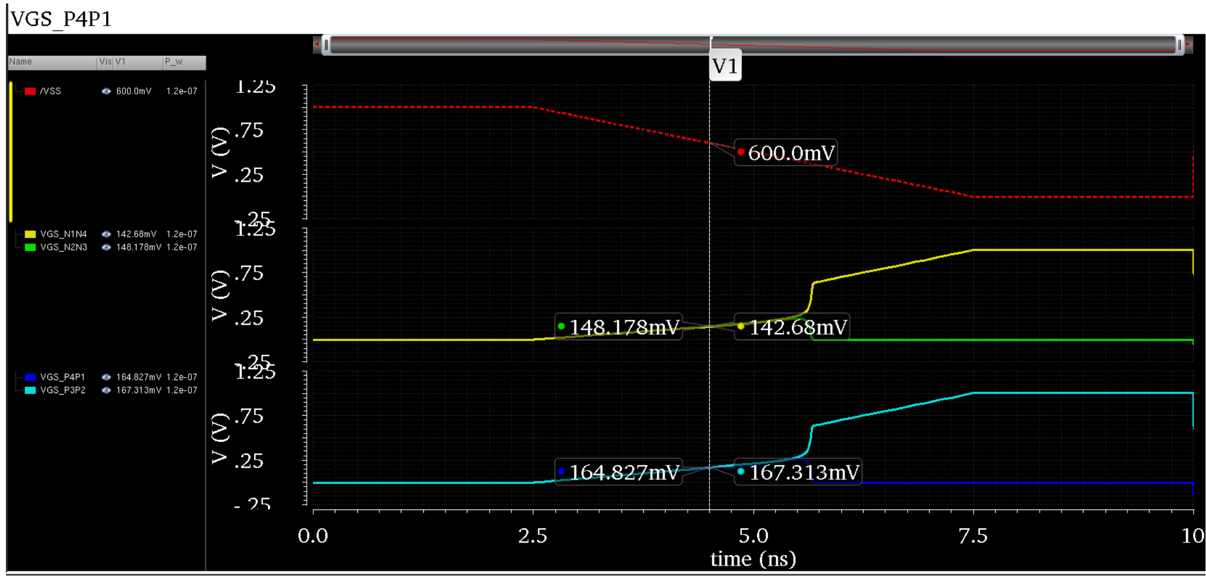

Figure 4.12 8T 120nm NMOS 120nm PMOS

This case is very similar to the first 8T GND manipulation case due to the equal sizing of NMOS and PMOS devices. Therefore it is predicted that PMOS mismatch will be dominant.

Table 4.1 V_{GS} dependency analysis summary

Width		Scheme			
NMOS	PMOS	6T V_{DD}	8T V_{DD}	6T GND	8T GND
200nm	200nm	PMOS	PMOS	PMOS	PMOS
200nm	600nm	NMOS	NMOS	NMOS	NMOS
120nm	120nm	PMOS	PMOS	PMOS	PMOS

The summary table outlines which device mismatch is dominant in each case. This analysis shows a consistent prediction across different cells and challenge schemes. The relative sizing of pull up PMOS devices to NMOS devices affects which type of device is the deciding factor in PUF quality. However, in

Table 4.2, the 8T cell shows noticeably smaller ΔV_{GS} values when PMOS and NMOS are equally sized. The next section aims to see if these predictions are correct.

Table 4.2 ΔV_{GS} dependence summary

Width		Scheme			
NMOS	PMOS	6T V_{DD}	8T V_{DD}	6T GND	8T GND
200nm	200nm	50mV	14mV	43mV	17mV
200nm	600nm	-36mV	-66mV	-50mV	-23mV
120nm	120nm	46mV	11mV	40mV	19mV

4.2 Isolated Mismatch Results

These results are meant to show whether NMOS or PMOS mismatch is the determinant of PUF reliability. It is expected that when a mismatch is disabled for certain devices, the assured response will drop drastically. When non-dominant devices are made ideal, they should slightly drop the assured response but not to the same degree as dominant devices. The range of results is 0-100% which represents the number of cells out of 1000 that respond with the same value over 100 cycles.

4.2.1 V_{DD} manipulation

Table 4.3 Isolated mismatch results: 200nm NMOS 200nm PMOS

Temp	8T	6T
Ideal NMOS	63.5	74.9
Baseline	75.4	76.8
Ideal PMOS	64.4	40.7

For the baseline result, 8T shows similar degradation in assured response while 6T shows a large degradation when using ideal PMOS devices. Looking at Figure **Error! Reference source not found.** the V_{GS} of PMOS is 50mV higher than that of NMOS. While in **Error! Reference source not found.** the V_{GS} of PMOS is about 15mV larger than that of NMOS. This smaller delta can be attributed to why 8T does not show a dependence on PMOS devices despite being sized equally.

Table 4.4 Isolated mismatch results: 200nm NMOS 600nm PMOS

Temp	8T	6T
Ideal NMOS	20.3	53.6
Baseline	82.8	81.8
Ideal PMOS	79.2	80.9

Table 4.5 Isolated mismatch results: 120nm NMOS 120nm PMOS

Temp	8T	6T
Ideal NMOS	62.7	77.0
Baseline	76.1	78.0
Ideal PMOS	67.5	47.0

4.2.2 GND manipulation

Table 4.6 Isolated mismatch results: 200nm NMOS 200nm PMOS

Temp	8T	6T
Ideal NMOS	60.2	69.7
Baseline	76.3	76.1
Ideal PMOS	65.0	62.4

Table 4.7 Isolated mismatch results: 200nm NMOS 600nm PMOS

Temp	8T	6T
Ideal NMOS	20.6	44.5
Baseline	83.4	82.9
Ideal PMOS	79.6	82.9

Table 4.8 Isolated mismatch results: 120nm NMOS 120nm PMOS

Temp	8T	6T
Ideal NMOS	63.9	69.6
Baseline	76.9	77.7
Ideal PMOS	68.2	61.8

Table 4.9 Isolated mismatch summary

Width		**Scheme**			
NMOS	PMOS	6T V_{DD}	8T V_{DD}	6T GND	8T GND
200nm	200nm	PMOS	even	PMOS	even
200nm	600nm	NMOS	NMOS	NMOS	NMOS
120nm	120nm	PMOS	even	PMOS	PMOS

From this experiment when the PMOS is 600nm, the predictions matched perfectly, but for other sizes, the 8T cell showed an even dependence on NMOS and PMOS. Recall that in

Table 4.2, the 8T cell shows a positive ΔV_{GS} but is much smaller than the rest when sized equally.

4.3 Sizing Sweep Results

The results from the isolated mismatch experiment showed strong results when the PMOS device was upsized. Sizing sweeps were done on each cell/scheme but a limited number of data points were simulated due to the high cost in computational resources.

Table 4.10 Input space

Width		
NMOS	PMOS	Wp/Wn ratio
120nm	600nm	5
200nm	600nm	3
200nm	200nm	1
600nm	200nm	1/3
600nm	120nm	1/5

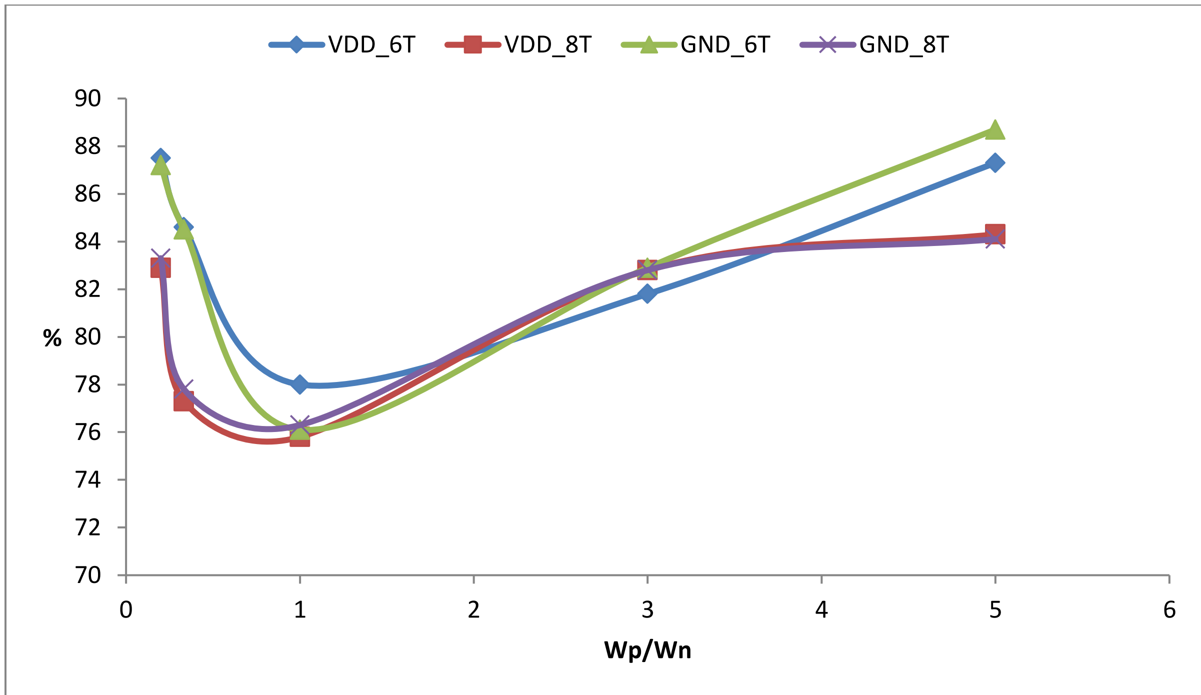

Figure 4.13 Sizing effect on assured response at 27 °C

From the figure above, it is clear that used non-equal sizing of PMOS and NMOS provides much better assured response. Compared to a Wp/Wn ratio of 1, a ratio of 5 improves 6T V_{DD} from 78% to 87.3%.

The 6T cell more often than not performs better than the 8T cell. Although 1000 samples were used for each data point, using a different seed can result in about 4% difference.

4.3.1 Effect of supply noise

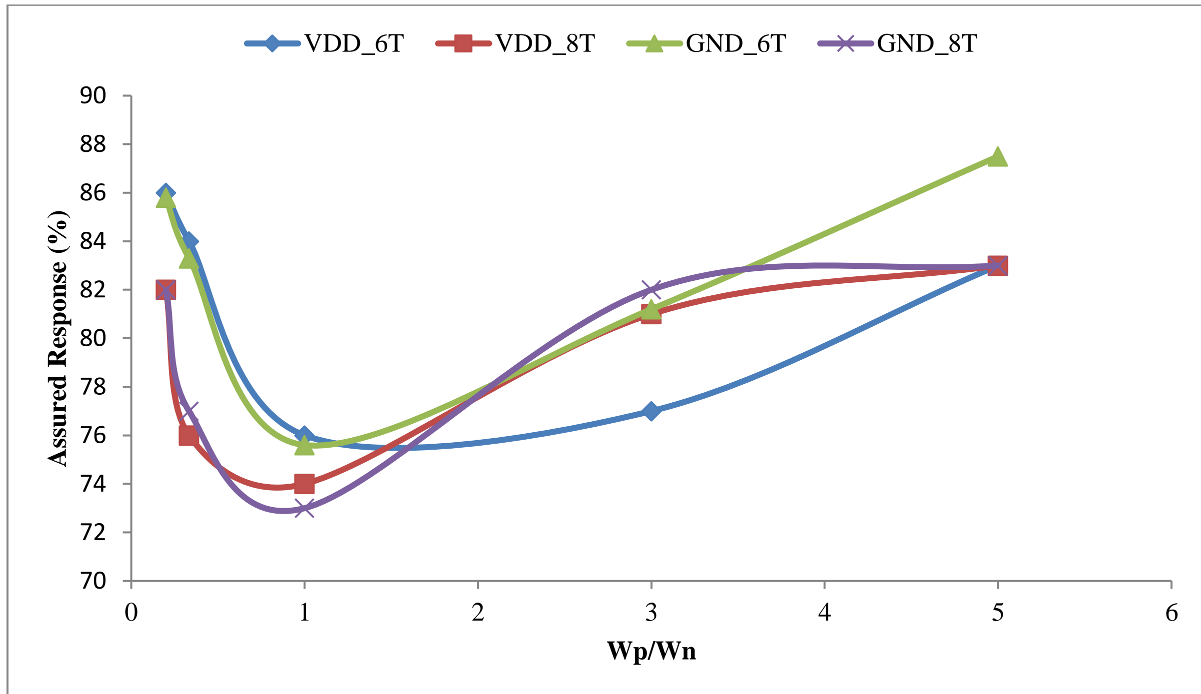

Figure 4.14 Sizing sweep with supply noise

The addition of supply noise degrades the assured response by an average of 1.6%. Overall supply noise degradation has less of an effect than relative sizing.

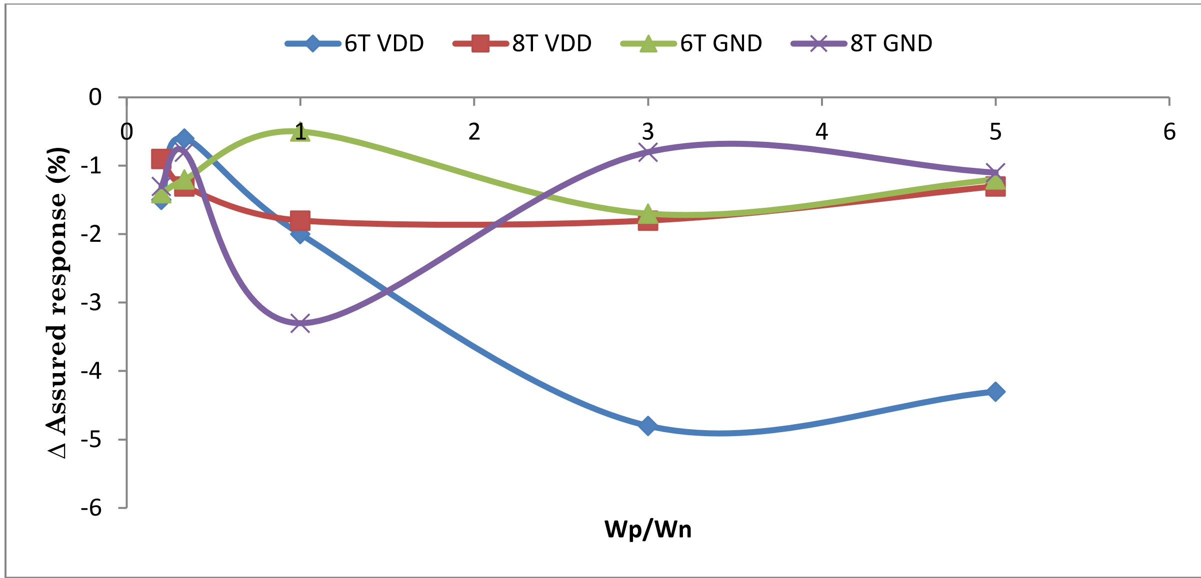

Figure 4.15 Assured response with supply noise

4.3.2 Effect of temperature

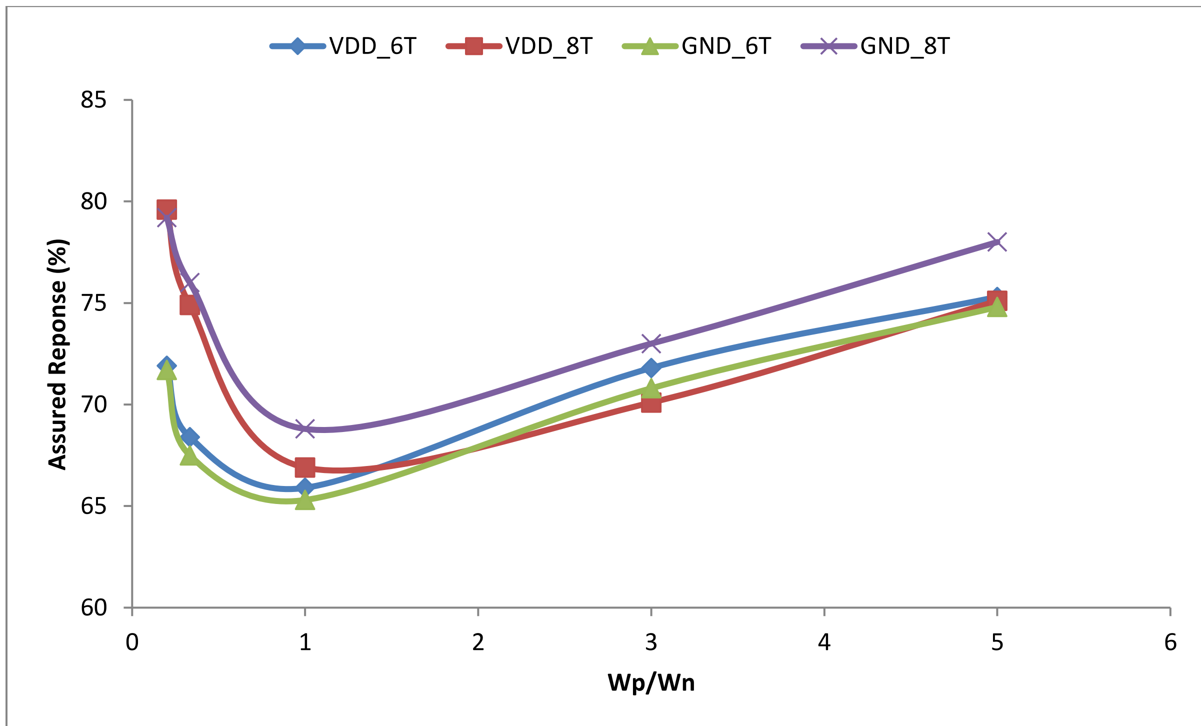

Figure 4.16 Sizing effect on assured response at 125°C

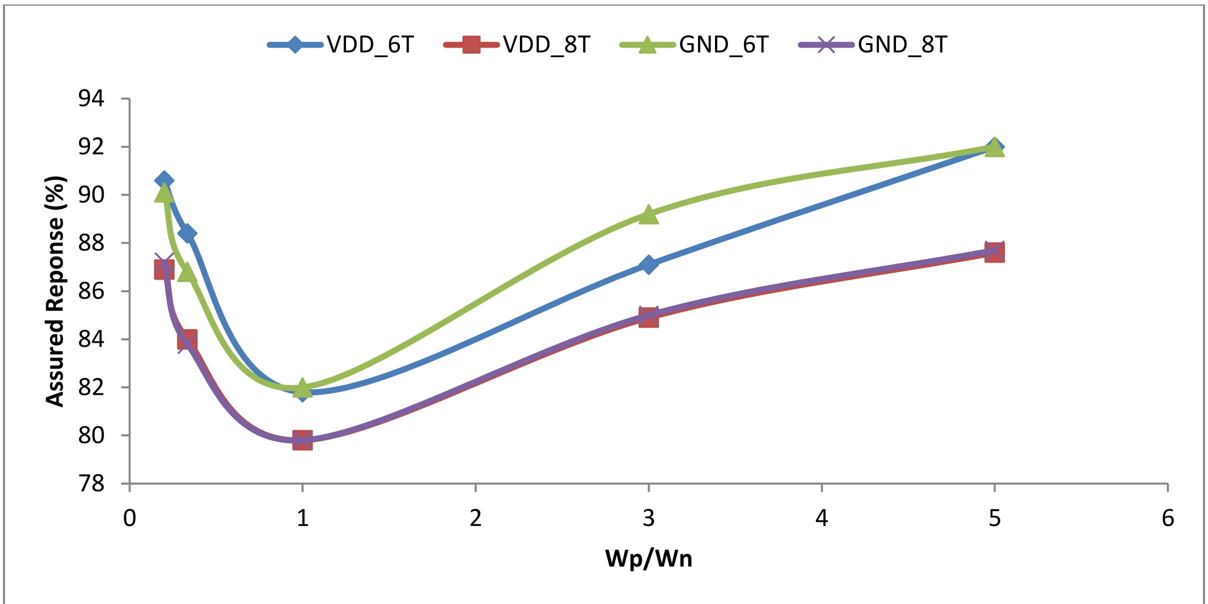

Figure 4.17 Sizing effect on assured response at -40°C

4.3.3 Uniqueness analysis

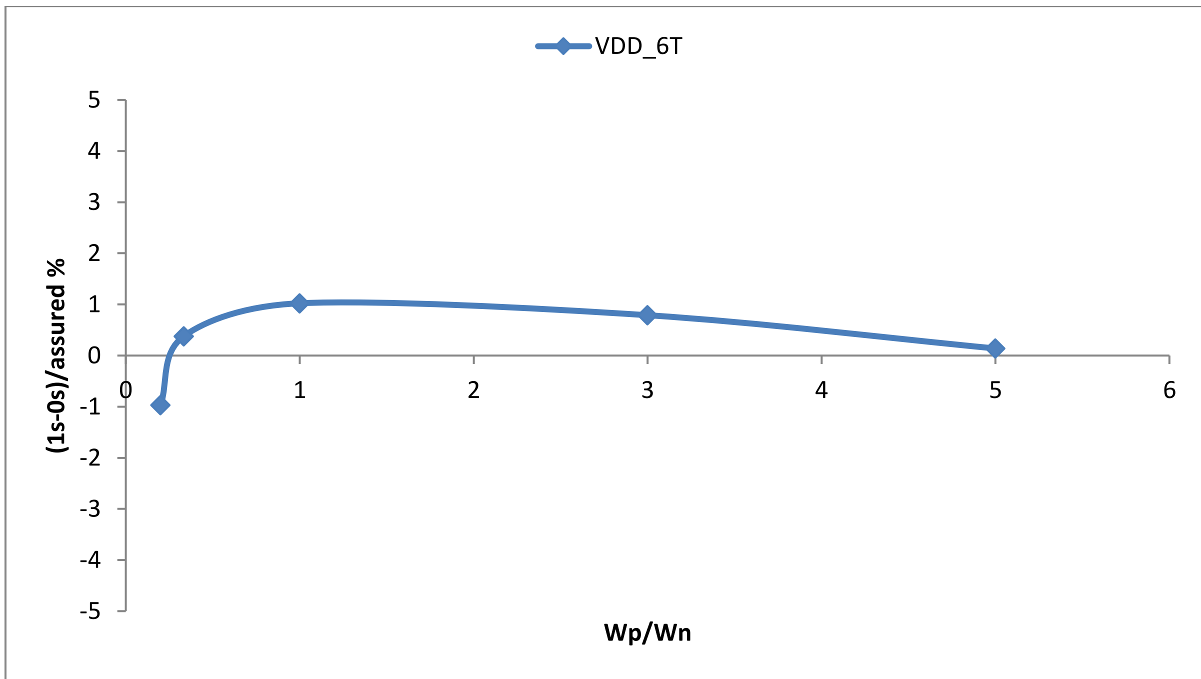

Figure 4.18 Distribution delta

The y axis shows the difference in logic 1s and logic 0s out of all samples with 100% reproducibility. This result shows an even distribution of 1s and 0s where the ideal result is 0. Any deviation is due to a limited number of samples. 5000 samples were simulated and show less than 1% difference in the distribution of 1s and 0s. It has been seen that as the number of samples increases, the delta approaches zero (i.e. equal number of 1s and 0s). This result should be obvious as no post layout simulations were done that would capture asymmetry.

4.4 Split-V_{DD} Results

Previously, only two design parameters were varied which were PMOS and NMOS width. With 8T Split V_{DD} there are now two PMOS widths that can be separately varied as well as the delay between the two V_{DD} signals. In total there are now four design parameters to be considered.

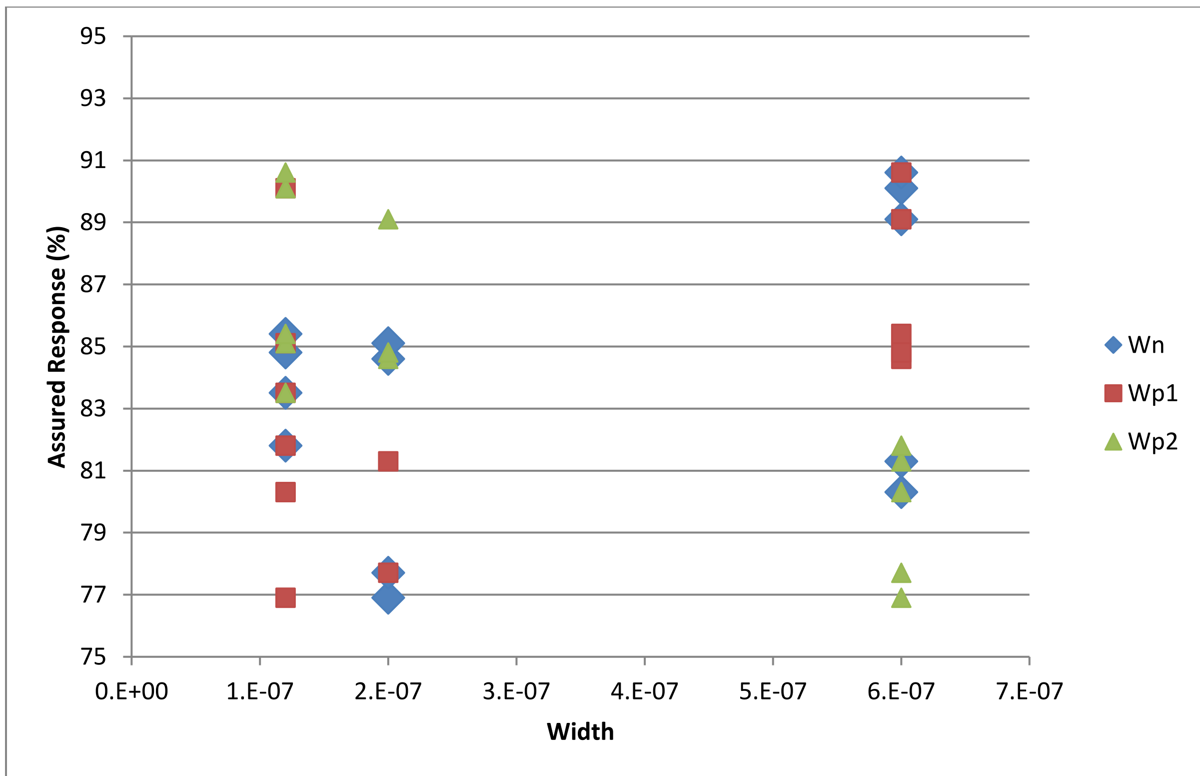

Figure 4.19 Split V_{DD} sizing effect when V_{DD2} rises first

The plot above shows the assured response for different sizing configurations. Smaller widths are shown on the left while larger widths are on the right. One set of sizes corresponds to drawing a horizontal line

and observing where each size intersects the line. Notice the highest assured responses occur with an upsized NMOS (blue square) and small inner PMOS (green triangle).

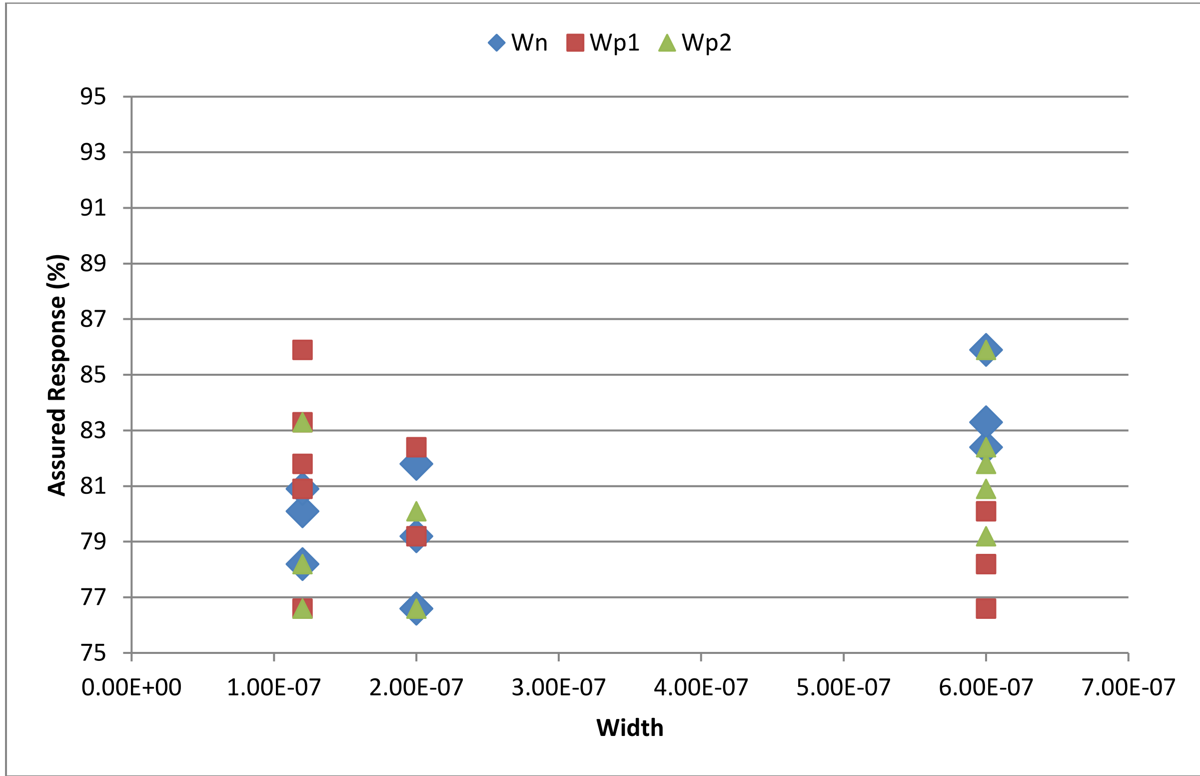

Figure 4.20 Split V_{DD} sizing effect when V_{DD1} rises first

Comparing **Error! Reference source not found.** to **Error! Reference source not found.**, it is much more beneficial to have V_{DD2} rise first. The best result is around 86.0% when V_{DD1} rises first but 90.5% when V_{DD2} rises first. This makes sense since the inner nodes have an NMOS half latch to provide positive feedback

4.4.1 Best candidate comparison

Table 4.11 Best candidate sizing

	Width			
Cell	PMOS	Inner PMOS	Outer PMOS	NMOS
Split V_{DD} 8T	n/a	120nm	600nm	600nm

8T	600nm	n/a	n/a	120nm
6T	120nm	n/a	n/a	600nm

4.4.1.1 Temperature sweep

A temperature sweep is done using the best performing cell for 8T with Split V_{DD}, 8T and 6T.

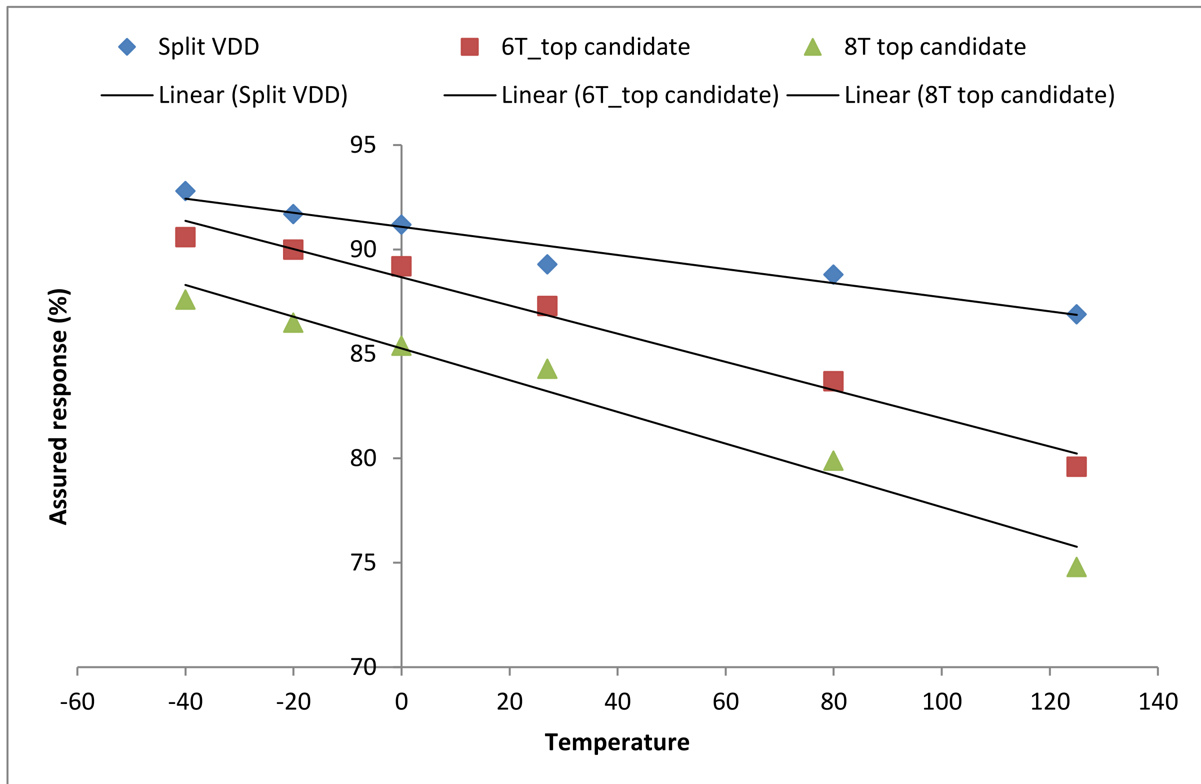

Figure 4.21 Temperature sweep comparison

As shown above, Split V_{DD} outperforms over cells from -40°C to 125°C. It also exhibits more temperature insensitivity compared with the other cells. From -40°C to 125°C there is a 6.0% drop for Split V_{DD} while 6T and 8T show 11.0% and 12.8% drop over the same range.

4.4.1.2 Delay sweep

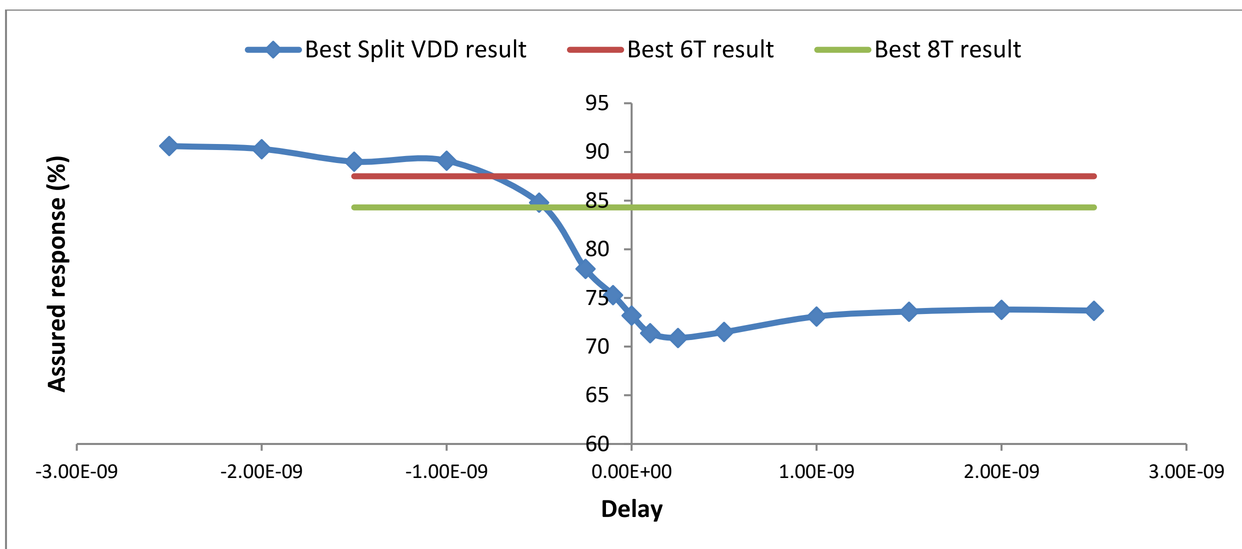

Figure 4.22 Split V_{DD} delay sweep

Here the effect of delay on the assured response is measured. Recall that negative delay implies that the V_{DD2}, or inner PMOS V_{DD}, rises first and vice versa. Since there is only one V_{DD} for 6T and 8T, the best result is shown as a horizontal line for comparison purposes. As the delay becomes more negative, the reproducibility improves greatly from 71.4% to 90.6%. With -1 ns of delay, Split V_{DD} performs better than the best 6T result. There is also a slight improvement by having a large positive delay of .5%. This is due to the fact that the outer PMOS devices are upsized and there is no NMOS half latch for the outer nodes QB and Q2B.

Chapter 5
Conclusion

Motivated by the IoT and the ubiquitous nature of SRAM in electronics this thesis' intent was to develop a design methodology to improve the PUF reliability of SRAM cells in the context of the 6T and 8T SRAM cell. Various applications of PUFs, their role in security systems and different types of PUFs are discussed. Analysis of the V_{GS} of during ramp up is performed to predict whether NMOS or PMOS devices are the main sources of bias in a SRAM PUF. To verify that the predictions via V_{GS} analysis are accurate, reliability simulations are performed while isolating the local mismatch to a particular type of transistor. The results indeed show that predictions can be made by analyzing the ΔV_{GS}. The simulations also showed that having larger PMOS devices improved reliability when NMOS devices were kept the same. A design sweep was done on relative sizing of NMOS to PMOS to further investigate reliability factors. The results show that equally sized devices performed the worse and highly skewed devices performed the best. A 9% improvement in reliability is shown in the 6T V_{DD} case. 6T overall performs better than 8T in this test but improvements can be made to the 8T to improve performance. The split V_{DD} technique takes advantage of multiple V_{DD} rails in the 8T cell and introduces a delay between two V_{DD} signals. It is much more beneficial to have the inner V_{DD} rise first. This is because the inner nodes have an NMOS half latch to provide positive feedback. Split V_{DD} performs 3% better than the best 6T V_{DD} design and is more robust over temperature. There are some limitations to this research. The scope of the simulations is at the cell level, meaning that no SRAM periphery circuitry was included. This was done to limit the number of variables that could influence the results and also speed up simulation run time. The run time of the simulations was also a limitation since transient noise greatly increases the number of computations needed. A balance had to be made between run time and accuracy. The layout of the cell was not considered in these results. This is because having layout could skew the uniqueness depending on how symmetrical the design is. Monte Carlo simulations were used to generate local mismatch between transistors but global variations were not considered in order to limit computation time.

Further work can include creating a test chip based on this methodology, practical implementation of split V_{DD} and GND manipulation and designing for memory and PUF applications simultaneously.

Bibliography

[1] Gartner Inc. (2014, November) [Online]. http://www.gartner.com/newsroom/id/2905717

[2] IEEE spectrum. (2016, Aug) [Online]. http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

[3] Bellardo, S,Savage J, "802.11 Denial-of-Service Attacks:Real Vulnerabilities and Practical Solutions," in *USENIX Security Symposium*, San Diego, 2003.

[4] Margaret Rouse. (2015, December) man-in-the-middle attack (MitM). [Online]. http://internetofthingsagenda.techtarget.com/definition/man-in-the-middle-attack-MitM

[5] Manik Advani, Vipin Tiwari, Laura Varisco,Narbeh Der Hacobian, Anurag Mittal, Michael Han, Al Shirdel,Alexander Shubat Jaroslav Raszka, "Embedded Flash Memory for Security," in *ISSCC*, Fremont, 2004.

[6] J.Rosenberg, "Embedded flash on a CMOS logic process enables secure hardware encryption for deep submicron designs," *Non-Volatile Memory Technology Symposium*, no. 10, pp. 3-21, 2005.

[7] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," in *48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, New York, 2011.

[8] Anderson, Markus Kuhn R, "Low Cost Attacks on Tamper Resistant Devices," in *Security protocols*, Paris, 1997.

[9] S. Skorobogatov, "Optical Fault Masking Attacks," in *Workshop on Fault Diagnosis and Tolerance in Cryptography*, Santa Barbara, 2010.

[10] P. Svasta, M. Dima, A. Marghescu and M. N. Costiuc M. Safta, "Design and setup of Power Analysis attacks," in *IEEE 22nd International Symposium for Design and Technology in Electronic Packaging (SIITME)*, Oradea, 2016.

[11] D.Bernstein, "Cache-timing attacks on AES," , Chicago, 2005.

[12] Sergei Skorobogatov, "Flash Memory ‘Bumping’ Attacks," International Conference on Cryptographic Hardware and Embedded Systems," , 2010.

[13] Maximilian Hofer Christoph Bohm, "Use Cases," in *Physical Unclonable Functions in Theory and in Practice*. New York: Springer, 2012, pp. 40-51.

[14] Schaumont P Simpson E, "Offline hardware/software authentication for reconfigurable platforms," in *Cryptographic hardware and embedded systems*, Berlin, 2006.

[15] M. Tehranipoor,K. Huang,D. DiMase,J. M. Carulli, Jr,Y. Makris U. Guin, "Counterfeit Integrated Circuits:A Rising Threat in the Global Semiconductor Supply Chain," *IEEE Transactions*, vol. 102, no. 8, p. 21, 2014.

[16] Suh, S,Devadas G, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in *Design Automation Conference*, San Diego, 2007.

[17] Patel A, Wander A, Eberle H, Shantz SC Gura N, "Comparing elliptic curve cryptography and rsa on 8-bit cpus," in *Cryptographic hardware and embedded systems*. Berlin: Springer, 2004, pp. 925–943.

[18] Ruhrmair U and Holcomb D, "PUFs at a Glance," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden , 2014.

[19] S. Hamdioui, V. van der Leest, R. Maes and G. J. Schrijen, M. Cortez, "Adapting voltage ramp-up time for temperature noise reduction on memory-based PUFs," in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, Austin, 2013.

[20] J.W Lee et al., "A technique to build a secret key in integrated circuits for identication and authentication application," *VLSI Circuits, Digest of Technical Papers*, pp. 176-179, 2004.

[21] F. Sehnke, J. S¨olter, G. Dror, S. Devadas, J. Schmidhuber. U. R¨uhrmair, "Modeling Attacks on Physical Unclonable Functions," in *ACM CCS*, 2010.

[22] S. K. Mathew et al, "16.2 A 0.19pJ/b PVT-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22nm CMOS," in *ISSCC*, Francisco, 2014, pp. 278-279.

[23] Maximilian Hofer Christoph Bohm, "Using the SRAM of a Microcontroller as a PUF," in *Physical Unclonable Functions in Theory and in Practice*. New York: Springer, 2012, pp. 249-259.

[24] Frank DJ, Gattiker AE, Haensch W, Ji BL, Nassif SR, Nowak EJ, Pearson DJ, Rohrer NJ 3. Bernstein K, "High-performance cmos variability in the 65-nm regime and beyond," , 2006.

[25] Brown A, Davies J, Kaya S, Slavcheva G Asenov A, "Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale mosfets.," *IEEE Trans Electron Dev*, pp. 1837–1852, 2003.

[26] Tam SC, Hsu FC, Ko PK, Chan TY, Terrill K Hu C, "Hot-electron-induced mosfet degradation – model, monitor, and improvement.," *JSSC*, pp. 295–305, 1985.

[27] D. Nairn and M. Sachdev J. S. Shah, "A 32 kb Macro with 8T Soft Error Robust, SRAM Cell in 65-nm CMOS," *IEEE Transactions on Nuclear Science*, vol. 62, no. 3, pp. 1367-1374, 2015.

[28] B.Razavi, *Design of Analog CMOS Integrated Circuits*. New York: McGraw Hill, 2001.

[29] McAndrew Tsividis Y, *Operation and modeling of the MOS transistor 3rd edition*. Oxford: Oxford University Press , 2011.

[30] S.Tam, F.Hsu, P.Ko, T.Chan, K.Terrill C.Hu, "Hot-electron-induced mosfet degradation – model, monitor, and improvement.," *JSSC*, pp. 295–305, 1985.

[31] P. Schaumont A. Maiti, "Improving the quality of a Physical Unclonable Function using configurable Ring Oscillators," in *Intrnational Conference on Field Programmable Logic and Applications*, Prague, 2009, pp. 703-707.