Design of a Low Power Latch Based SRAM Sense Amplifier

A Major Qualifying Project

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

in

Electrical and Computer Engineering

by

Sarah Brooks

Anthony Cicchetti

March 27, 2014

APPROVED:

Professor John McNeill, MQP Project Advisor

Abstract

There is a constant push to reduce power consumption and increase speed in transistor memory devices. The goal of this project is to improve Allegro MicroSystems current Static Random Access Memory technology by designing and implementing a new sense amplifier. The final design is latch based and uses positive feedback to quickly display data at the output, which stops static current flow and dramatically reduces power consumption.Additionally a three-dimensional LED structure was designed and built to display a series of letters, characters or visual effects.

Acknowledgments

Sarah would like to thank the following people for their guidance, support and contributions to this project.

Mu Sarwar (Allegro MicroSystems) for his countless hours of mentoring me and motivating me to learn as much as I have about SRAM, and also for taking the time to help with designs, teach me layout, and guide this project to completion.

Jim McClay (Allegro MicroSystems) for offering guidance on the direction of the project.

Frank Towey (Allegro MicroSystems) for sharing his knowledge about memory testing equipment and procedures.

Prof. John McNeill for creating this project, providing feedback,and guiding me through this process.

Prof. Steve Bitar for being a sounding board and providing encouragement throughout this project.

Ben Dwyer for his support, editing assistance, and for helping me to keep the big picture in mind.

I would especially like to thank everyone at Allegro MicroSystems and Worcester Polytechnic Institute, for without them this project would not have been possible.

Executive Summary

The original focus of this project was to improve Allegro Microsystems' SRAM technology by designing and implementing a new sense amplifier. The goal of the new design was to decrease the power consumption of the SRAM chip and decrease the access time during a read cycle. The design would then be fabricated and tested for errors and compared to the previous differential sense amplifier design. After the testing was complete, the SRAM chip could be interfaced with a micro-controller and a three-dimensional LED display.

SRAM sense amplifier options were researched after determining design constraints and requirements. A basic latch based design was chosen as no static current flows in the amplifier once a value is read from an SRAM cell. By decreasing the current flow during the read cycle, the power consumption will also decrease. After deciding upon the design, a sizing analysis was run to determine the transistor sizes that are used in the amplifier. The final sizes produced an amplifier with a high stability, low power consumption, and low access time.

Next, the amplifier was tested through three different testing phases. First, the amplifier was tested on a stand alone basis to verify its ability to read values correctly that are stored within an SRAM cell. Next, the amplifier was integrated into the top-level SRAM layout and tested with the full chip. These tests were used to verify the timing requirements and analyze the amplifiers behavior when parasitic capacitances were added to the system. The parasitic values were extracted from the SRAM layout and were used to mimic real world performance of the chip. Finally, the system was tested as a whole to verify that the chip was accurately writing, storing and reading values stored within the memory.

In parallel to the testing, the layout was created for the SRAM chip. The layout is the silicon level design for the chip which will be used when the chip is fabricated. The layout was compared

to the transistor level schematic to verify that the two designs were identical, and was also checked to make sure that it met all design constraints for the CMOS fabrication process. Although a test chip will still be made of the SRAM chip with the new sense amplifier design, the physical device was not able to return from fabrication within the time frame of this project. When the design does return in the future, the simulation results and collected data will be compared. After any necessary changes are made, the new sense amplifier design may be phased into production.

There were many things to think about when designing the orientation and structure of the LED display. There were three separate ways that the LEDs could be connected, all with their own pros and cons. Some of the things to look at were the addressing of the LEDs, the current needed to turn on a certain number of LEDs, and the way the LEDs would need to be bent to build the physical structure. After analyzing the advantages and disadvantages of each the final design was chosen. The next step was to start the actual construction of the display, after many different methods of bending the LEDs and soldering them together a proficient method was found. Once the 675 LEDs were all soldered together the beginning of coding would start. The way the coding works is described in the report.

Contents

List of Figures

List of Tables

Authorship

This major qualifying project is comprised of two main sub-projects: the design of an SRAM sense amplifier and the design and implementation of a three-dimensional LED display. The three project partners focused on different aspects of the project and report as is listed below.

Sarah Brooks: Sarah focused on the SRAM sense amplifier design and all SRAM-related aspects of the project and report. The chapters she wrote are listed below.

- Chapter 2: SRAM Background Research
- Chapter 3: SRAM Sense Amplifier Design
- Chapter 4: SRAM Results
- Chapter 5: SRAM Conclusion

Anthony Cicchetti: Anthony focused on designing and building the hardware to construct the LED display as well as the coding and addressing scheme. The chapters he wrote are listed below.

- Chapter 6: LED Display

Chapter 1

Introduction

This Major Qualifying Project includes two subprojects; the design and fabrication of a sense amplifier for an SRAM chip, and the design of a three dimensional LED display. This project is sponsored by Allegro MicroSystems LLC and NECAMSD Labs.

In many applications, Static Random Access Memory (SRAM) arrays make up a large area of high-performance integrated circuits [1]. There is a constant push to increase a chips speed and to reduce its power consumption. As memory sizes increase and transistor sizes decrease, a trade-off must be made between reducing the layout area and adding peripherals to increase the performance of the chip [1]. Designs should work as efficiently as possible while minimizing complexity.

Allegro MicroSystems SRAM designs come in various sizes and configurations, but all use a 6T cell and a differential voltage sense amplifier to read the value stored in the cell during a read cycle [2][3][4]. Although the differential amplifier is very reliable, constant current must be applied throughout the entire read cycle causing a large static power loss. The goal of this project is to design a new sense amplifier that uses less power during a read cycle. The scope of this project includes determine the most suitable sense amplifier design, sizing the amplifier transistors for minimum area, maximum stability, and minimum power consumption, testing the amplifier to confirm functionality and performance, and integrating the new sense amplifier with a current SRAM chip design.

After various sense amplifiers were evaluated, a latch-based design was chosen because it will

have no static current flow after the amplifier latches and reads the value stored in the memory [5][6]. There will be multiple steps in testing the sense amplifier. Initially, the stand-alone performance and characteristics of the design will be analyzed. Next, the amplifier will be tested with a simplified SRAM model. Finally, the amplifier will be integrated into Allegros top-level SRAM schematic, where the amplifier performance will be verified with as close to real world operating conditions as possible. The simulations and testing will be performed using Cadence Virtuoso Design Suite.

In addition to designing and testing the amplifier, the physical transistor layout must be created so that a test chip can be fabricated using CMOS technology. The layout determines the total area that the chip will use, and is an exact model of what will be fabricated. The different connections at the silicon level will impact the parasitics that will exist in the SRAM chip, which can negatively affect the device performance if not correctly placed [5][6]. Although the actual SRAM chip will not arrive back from fabrication in time for real world measurements to be made, parasitic values may be extracted from the layout, allowing the simulated results to provide a realistic estimate for the expected behavior.

The second set of goals for this Major Qualifying Project are to design and implement a scalable and power efficient three dimensional LED display. Gaining experience from design process is one of many things that will be accomplished through the completion of this MQP. There are many aspects of the design process that must be considered such as continuously adapting to variations within the project. The purpose of this MQP is to obtain a thorough and beneficial learning experience. For this MQP we are going to design and build an LED display. To do this we will need to not only design and build the physical structure, but also to develop an addressing scheme. This scheme will give us access to each of the 675 LEDs. This will allow for the structure to display three-dimensional letters or images. A Printed Circuit Board (PCB) layout will also be fabricated as a base for the LED display circuitry. An MSP430 microcontroller will be used to drive the LED display by providing the digital logic to control the state of all of the LEDs.

The overall project begins with research of current models of the LED display. Aspects such as the multiplexing scheme, microcontroller, and part selection were identified from different working LED cube models. Next, display features, driving technique, and microcontroller options are assessed to determine the necessities of the prototype. Many calculations went into determining

design parameters such as power consumption and frequency requirements. These parameters and the final prototype display size were chosen so that the design could be easily scaled. Once the design parameters had been set one of the final steps of the design process was to begin the selection of parts. The selection process consists of selecting a pool of appropriate parts for the prototype. A filtering process is done to eliminate less compatible components that do not meet the design qualifications. Finally, a value analysis will be done on the remaining pool to determine the optimal set of parts. Once all of the parts are chosen and a final design is completed, the final step is to design the PCB layout and assemble the working prototype.

1.1 Report Overview

This report is divided into two main sections that represent the two overarching goals of the project. Chapter 2 to Chapter 4 covers the design of a sense amplifier for a Allegro Microsystems' SRAM chip. Chapter 2 includes the background research for this project. This section includes an introduction to Static Random Access Memory (SRAM) and its components as well as memory testing options for both before and after fabrication. Chapter 3 explains the the project goals for the new sense amplifier design, multiple sense amplifier options, and a sizing analysis for the chosen design. Chapter 4 reviews the top-level SRAM design, explains the SRAM layout, and examines test results and compares the new amplifier to the preexisting design. The second part of the report contains the design process of the LED structure as well as the coding and addressing schemes. This is shown in Chapter 6.

1.2 Time Management

Over the course of this project, unforeseen issues and changing deadlines have altered the plans and goals. Initially, the goal of the project was to improve Allegro MicroSystems SRAM technology by designing a new sense amplifier. This new sense amplifier would be implemented into an existing SRAM design and the chip would be fabricated, tested, and interfaced with a microcontroller and a three-dimensional LED display. However, the fabrication of the SRAM chip will not be completed before the final project deadline. Due to this delay, testing the SRAM chip and interfacing it with the microcrontroller and LED display is no longer feasible. This unforeseen circumstance prevents the original plan from being completed and has altered the direction of the

project. Now, the SRAM portion of the project will focus on final simulation and layout results that are as close to the real world behavior as possible. The LED display is now driven by an alternate microcontroller that that does not require external memory.

Chapter 2

SRAM Background Research

2.1 Introduction to Memory

2.1.1 Terms and Sizing

A memory chip can be thought of as a matrix of locations that are each able to store one binary bit – either a 1 or a 0. Each location that is able to store a bit is known as a cell. To simplify the design and to save space, the memory matrix usually has a set width of bits that is usually one or more bytes long. This group has one complete unit of information and is known as the memories word length. The entire word of memory is stored at a specific location known as its address, and all actions performed on the memory such as reading and writing values will be done to the entire word at a specific address at the same time. The size of a memory describes how many different locations are available to store bits of information. The memory size can be calculated by multiplying the word length by the number of addresses [6][3]. Figure 2.1 shows an example SRAM memory array.

2.1.2 Address Decoders

Large memory arrays can have hundreds of different rows of cells. Since only one word of data is selected at a time, having every address assigned to only one pin is a waste of space. To decrease the number of pins needed in order to choose an address, most memory chips use address decoders [3][6]. Address decoders use combinations of logic 1's and 0's on the input to choose a

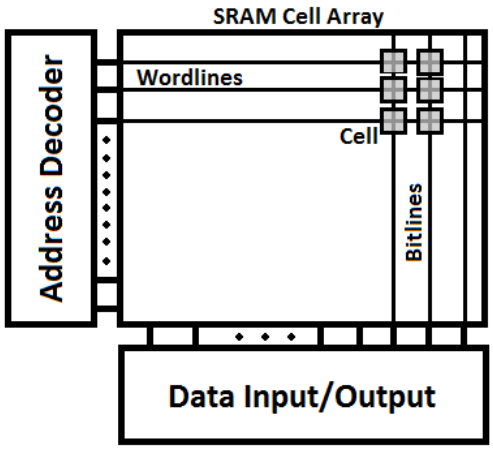


Figure 2.1: SRAM Overview

unique address. This allows n pins to access 2^n locations. In certain cases, pre decoders may also be added to further reduce the number of pins needed to access each address. Some synchronous chips will also integrate the clock signal in with the address decoders so that cells are only selected during a certain portion of the clock cycle.

2.1.3 Data Input and Output

Data busses are needed to carry input and output information to and from each memory cell. Memory chips can either have separate data input and output buses or they can share one bus. Sharing an input/output bus saves space on the chip, but using separate lines is simpler to design and implement. In both cases, the data available at the output is only important during a read cycle, and the data at the input is only important during a write cycle. If chips share one bus, tri state buffers are needed at the output so that data is only seen during a read cycle and a output enable pin is needed to enable the buffers [6]. Tri state buffers act like switches. If the buffer is enabled, the switch is closed, and the output follows the input. If the buffer is disabled, the switch is open and the output is in a high impedance or high z mode [2]. Figure 2.2 below shows a tri state buffer along with a representation of how it works.

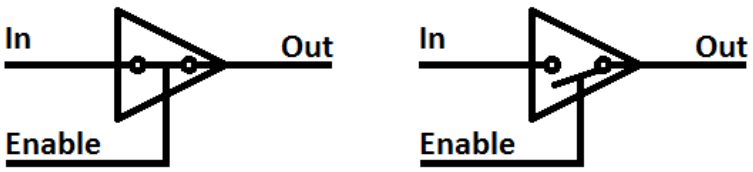


Figure 2.2: Tri State Buffer

2.1.4 Intro to the SRAM Cell

Static Random Access Memory, or SRAM, is a type of volatile memory with a nondestructive read cycle. Figure 2.3 shows an example of an SRAM cell in standard 6T configuration [2][3][6]. The SRAM cell contains two access transistors (N_{AX1}, N_{AX2}), and two cross coupled inverters (N_1, P_1 and N_2, P_2) comprised of a PMOS load transistor, and an NMOS driver transistor. The cross coupled inverters constantly reinforce which value is stored on each side of the cell. Node Q holds the cells value and node $\overline{Q}$ holds its complement.

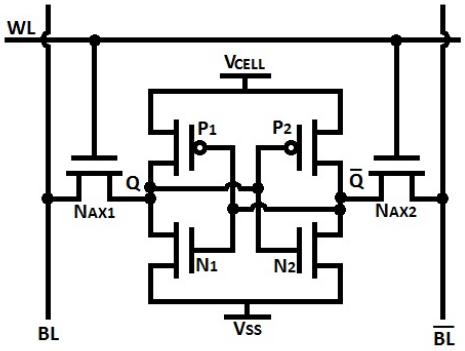


Figure 2.3: SRAM Cell

The cell is powered by the voltage applied at V_{CELL}. As long as power is supplied to V_{CELL}, the SRAM cell can hold its value indefinitely. The wordline (WL) selects the cell and allows values to be read from or written to the cell [6]. When the wordline is not selected, the cell is in standby mode. Selecting an address brings a wordline to a logic high, selecting all of the cells in that word. This brings the cells out of standby by enabling the access transistors, which allows the cells to be read from or written to. The cell also has two bitlines which control both the input and output of the data from the cell. The first bitline (BL), known as bitline, holds the same value that is stored in the cell. The second bitline $\overline{BL}$, known as bitline bar, or $\overline{bitline}$, or bitline not, holds the

inverse of the value that is stored in the cell.

2.1.5 Read and Write Peripherals

Since only one row of SRAM cells can be accessed at the same time, only one cell per column requires the use of the read and write peripherals. Because of this, all of SRAM cells in one column are able to share the same peripherals, allowing for space to be saved on the chip.

Write Circuitry

The SRAM write circuitry connects directly to the bitlines and is used to write a value to the SRAM cell. The value stored in the SRAM cell is stored at node Q, while the inverse of the value is stored at node $\overline{Q}$. The cell can be seen above in Figure 2.3 . Instead of writing either a logic 1 or a logic 0 to side Q, a 0 is written to either Q or $\overline{Q}$ depending on which value should be stored in the cell. To write a value to the cell, one bitline is held at V_{DD} while the other is held at V_{SS}. Holding the bitline at V_{SS} will cause the inverter to be pulled past its trip point, which will cause the cell to latch and hold the new value [6].

Precharge Circuitry

The precharge circuitry brings the bitline and $\overline{bitline}$ to be the same voltage before a read cycle. The simplest version of precharge circuitry consists of three transistors that can be either NMOS or PMOS devices. Two of the transistors are used to connect the bitlines to V_{CC}, and the third transistor is connected between the two bitlines to ensure that the lines end up being the same voltage [6]. Figure 2.4 shows an example of a precharge circuit that uses three PMOS transistors. To precharge the bitlines, the signal PRE is brought to a logic 0. This turns on all three transistors which then charge and equalize BL and $\overline{BL}$ to be V_{CC}.

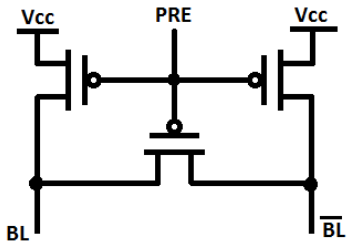


Figure 2.4: Example Precharge Circuitry

Sense Amplifier

The sense amplifier is in charge of detecting what value is stored in an SRAM cell during a read cycle and displaying that value at the output. Since only one row of data is accessed during each read cycle, each column of cells within the SRAM array requires only one sense amplifier [3].

A sense amplifier works by sensing a relatively small difference between the voltages of the two bitlines, then amplifying the difference at the output to show if a cell is storing either a logic 1 or 0 [3][6][4]. The bitlines are precharged before each read cycle to ensure that the difference between the bitline voltages are caused by the value that is stored in the cell. If the bitlines are not precharged, there is a chance that the sense amplifier could misread and present an incorrect value at the output. Often times sense amplifiers are followed by an output buffer to ensure that the full logic level is shown at the output. There are two main sense amplifier categories; voltage sense amplifiers and current sense amplifiers [7]. This report mainly focuses on voltage sense amplifiers. Figure 2.5 shows an example of an SRAM column setup with a voltage sense amplifier.

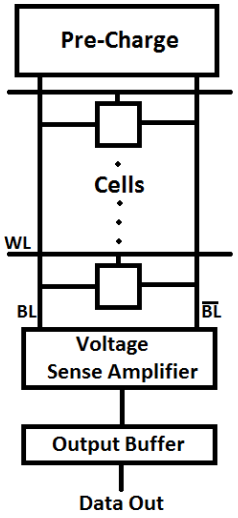


Figure 2.5: SRAM Column with Sense Amplifier

In order to understand how a sense amplifier works, it is important to first understand how an SRAM cell behaves during a read cycle. An example of the behavior of one cell during a read cycle may be seen in Figure 2.6 along with an example timing chart. The read cycle begins after the bitlines are precharged to the supply voltage but before the cell is activated. This may be seen in Figure 2.6 on the left at time t1. When the precharge cycle is complete, the cells bitlines are both the same value.

Next a wordline is selected which activates a certain cell or row of cells. This can be seen in Figure 2.6 in the middle which corresponds to time t2. For this example, the SRAM cell is storing a logic 0, which can be seen at node Q. This means that a logic 1 is stored at $\overline{Q}$. Due to the nature of the cell, only two of the cell transistors remain activated once a value is stored. Transistors

N_2 and P_1 are both off because of the values that appear at their gates. When the wordline is selected, the access transistors, N_{AX1} and N_{AX2} connect the SRAM cell to the precharged bitlines. The value stored in node $\overline{Q}$ is approximately the same value as the voltage that the bitlines were precharged to, transistor P_2 helps to maintain this value and bitline $\overline{BL}$ does not discharge. The value stored in node Q is much less than the value that the bitlines were precharged to. Transistor N_1 creates a path to ground which causes the bitline voltage to drop. The bitline and the selected cell have paracitic resistances and capacitances which form an RC time constant that causes the voltage to discharge exponentially.

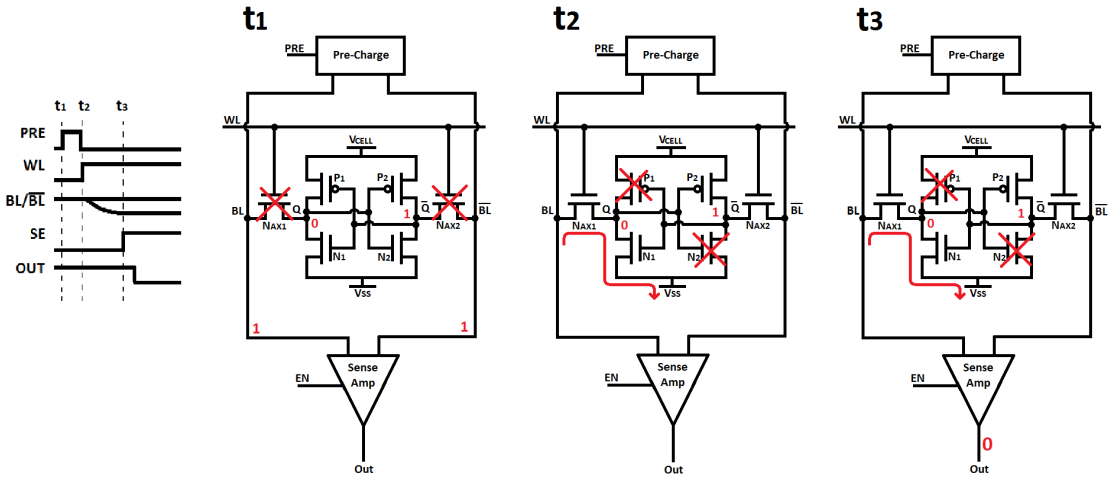


Figure 2.6: SRAM Read Cycle

Instead of waiting for the cell to completely discharge the bitline, the sense amplifier is used to amplify the small difference between the two lines. Depending on the design that is used, the sense amplifier may be activated as soon as the cell is selected, or it may be selected after a certain amount of time has passed to allow the voltage difference between the bitlines to be large enough to not cause a misread of the cell. The final step where the sense amplifier is activated can be seen in Figure 2.6 on the right which corresponds to t3 on the timing chart. Later in this report in section 3.2 specific sense amplifier designs will be explained.

2.1.6 SRAM Timing

The timing parameters of an SRAM chip must be known in order to have the device perform successfully. The timing defines the maximum frequency that can be used for a synchronous device,

and lets the user know how fast they should expect to see values at the output.

Figure 2.7 shows an example timing diagram for a synchronous SRAM chip. The timing diagram shows six different parameters that affect the timing performance of an SRAM chip. The first line shows the clock input which is usually an input from an external source. All of the other timing information relates back to the clock speed. The period of the clock can be seen in the figure as t_{CYCLE}, and is the minimum amount of time that the memory needs to perform consecutive read or write operations. The minimum amount of time needed for a read cycle is not necessarily the same as the amount of time needed for a read cycle. Despite this, a system will usually use the same unit of time to execute either operation in order to simplify the design [3]. The maximum clock frequency that an SRAM chip can handle can be seen in Equation 2.1.

$$f = \frac{1}{t_{cycle}} \tag{2.1}$$

The setup timing, shown as t_{SETUP}, is the amount of time needed to set up the system before executing a read or write operation. The set up time accounts for things such as pre-charging the bitlines before reading, and readying the data to be written to the cells.

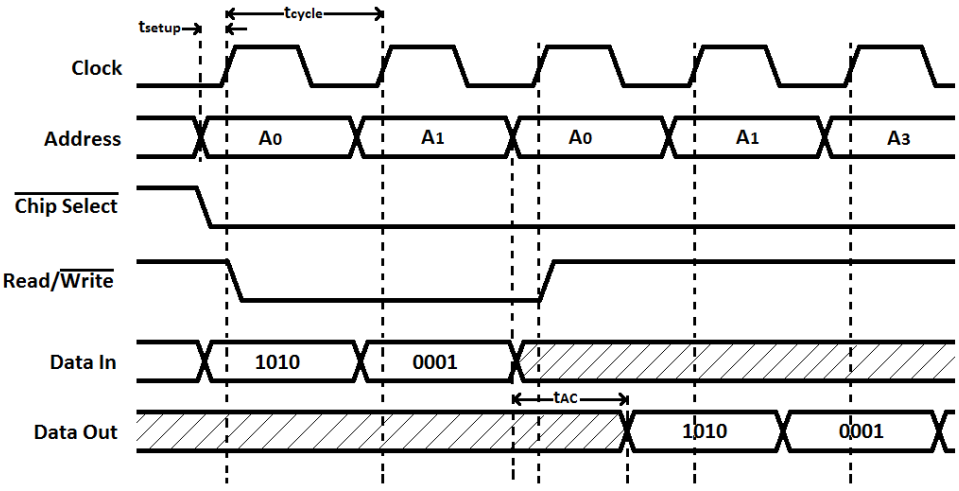


Figure 2.7: SRAM Timing Diagram

The second row in the timing diagram shows the addresses for the memory locations. Some of the other timing parameters, such as the read access time are measured with respect to the time when a valid address first appears.

The third line shows the chip select. The bar represents that chip select is enabled when the voltage is brought low. A chip must be selected to operate. If chip select is left high, the SRAM is in standby mode.

The fourth line shows the read and write control for the chip. When the signal is low, data will be taken from the input, which can be seen on line five, and will be written to certain cells that are selected by the address. When the cells are not being written to, it does not matter what data is seen on the input.

On the other hand, when the read and write signal is high, data will be read from the cells that correspond to the selected address. The amount of time that it takes to display the data from the selected cells once a valid address is accessed is known as the read access time. This can be seen as t_{AC} in the timing diagram. The read access time is usually between 20% to 80% of the cycle time [6]. If the read access time were to exceed the cycle time, incorrect data would b seen at the output and the SRAM chip would not be able to successfully prepare for another read or write procedure. While data is not being read from the SRAM chip, it does not matter what data is seen at the output.

2.2 Memory Testing

Testing is used to ensure both the reliability and quality of a physical product. The test chip that is fabricated by Allegro Microsystems must be tested before it can be implemented with the LED cube. For large scale production, testing must be done to separate out defective devices and to verify that those that pass testing behave as expected within the specifications they were designed for [8]. Due to the decrease in transistor sizing and the condensing of layouts, it is extremely likely that a defect will occur in an SRAM chip. Defects can be caused by design flaws, manufacturing and packaging errors, in-the-field errors, and process variations. A seemingly small variance in device parameters can be the difference between a functioning and failing chip. Since each SRAM chip is made up of millions of transistors, it is not physically or financially possible to exhaustively test the memory. To conserve both time and money, testing should ideally have maximum fault coverage with minimum complexity [1].

2.2.1 Types of Tests

Memory chips can be tested using both digital and analog techniques. Digital testing also known as logical testing is done after fabrication and makes sure that the memory functions like it should. Analog testing is usually done through simulation during the design phase and looks at how the chip behaves under certain operating conditions.

2.2.2 What Digital Tests Test For

When a defect is present in an SRAM chip, incorrect outputs will be produced under certain conditions. For digital testing, the inputs and outputs of a memory chip are explored on a functional level. Fault models have been created so that a representation of possible physical defects can be understood at a functional level. Each fault model only explains a functional fault, but many different defects at the transistor level can lead to the same functional fault [1]. All memory faults are caused by either hard (permanent) or soft (transient) defects [1] [9].

2.2.3 Hard Errors

Hard or permanent errors are caused by a physical defect and will occur under all SRAM operating conditions. These errors are caused during manufacturing, or later during use. Errors from manufacturing can be caused by extra or missing material, oxide breakdown, imperfections or impurities in the wafer, missing contacts, and packaging mistakes [10].

2.2.4 Fault Models

Below common fault models that are caused by hard errors are explained [1][11][12][13].

Functional SRAM Cell

Before explaining how a faulty SRAM cell behaves, it is important to understand the behavior of fully functional cells. Figure 2.8 shows the state diagram for a single fully functional cell. The cell has two states; it can either be holding a logic high or 1 or a logic low or 0. Reading is non-destructive to the cell so it is not included in the diagram. Writing to the cell is denoted with a w followed by the state that is being written to the cell –either a 1 or 0. Writing the same value to a cell will cause the state to stay the same, writing the opposite value will cause the state to

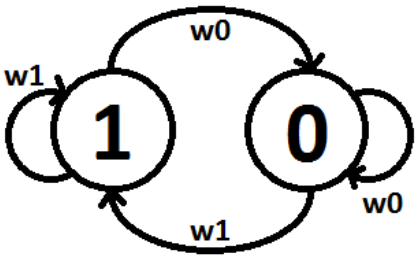


Figure 2.8: Fully Functional SRAM cell

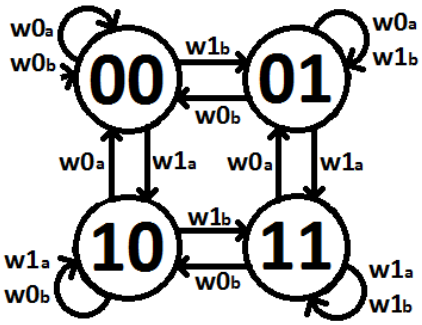


Figure 2.9: Two Fully Functioning SRAM Cells

switch. Figure 2.9shows the state diagram for the interactions between two fully functional cells, a and b.

Stuck-At Fault

Stuck-at faults are models for cells whose values are unable to change from either the logic 0 or 1 state. This fault only affects one cell. The cell value is permanently stuck and will stay the same no matter what is done to the rest of the system. Figure 2.10 shows the state diagrams for both a stuck-at 1 cell (on the left) and a stuck-at 0 cell (on the right). As the diagram shows, the cell will not be able to change out of its original value. Stuck-At faults are one of the most common faults to appear in memory testing [1].

Figure 2.10: Stuck-At Faults

During testing, stuck-at faults can be found by writing all of the cells to a specific value (either logic 0 or logic 1), reading the cell, and then repeating with the inverse value. If a stuck at fault exists, that cell will not change its value when a new one is written to it. Both logic values must be checked as cells can be stuck at either logic state.

Stuck Open Fault

A stuck open fault only affects one cell. This fault simply means that a cell that cannot be accessed. It is said to be stuck open because no actions can reach the cell. This fault is usually caused by an issue with the bitline or wordline. Stuck Open faults can be found in the same way as stuck at faults as can be seen above.

Transition Fault

Cells with transition faults are unable to transition from 0→1 or from 1→0. This fault only affects one cell. Failure to transition from 0→1 is known as a rising transition fault and failure to transition from 1→0 is known as a rising transition fault. Transition faults differ from stuck at faults because it is possible for the cell to be in either state, it is just not possible for it to switch into its complementary state from a certain position [1]. Transition faults do not have to exist in both directions. Figure 2.11 shows a cell with a falling transition fault.

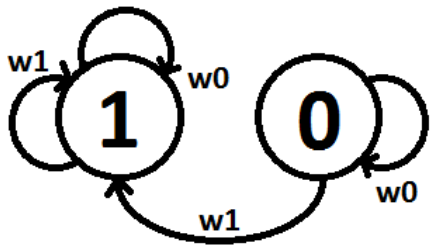


Figure 2.11: Transition Fault

Transition faults can be found by having each cell undergo both transitions and reading the results. This is done by writing the cell to one state, writing the cell to the opposite state and then reading the value out of the cell [14]. Since SRAM is volatile, the value of a cell is random when the system first powers on. If a cell with a transition fault initializes to the state where it cannot transition from, it may look like a stuck-at fault [1].

Coupling Fault

In addition to single cell faults, some faults can be caused by coupling between two or more cells. These faults are modeled as an interaction between an aggressor cell and a victim cell. When the aggressor cell switches state, it will cause a change in the victim cell. An Inversion coupling

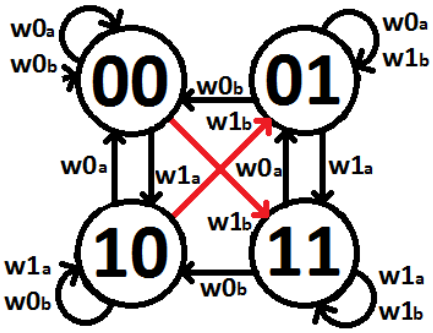


Figure 2.12: Inversion Coupling Fault

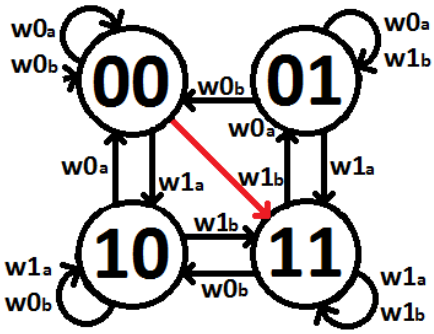


Figure 2.13: State Coupling Fault

fault causes the victim cell to invert values when the aggressor cell goes through a transition [10]. An idempotent coupling fault forces the victim cell to be either a 1 or a 0 when the aggressor cell goes through a transition. A state coupling fault forces the victim cell to become a certain state depending on what the aggressor cell is. Coupling faults can only happen between cells that are physically near each other in the memory. Figure 2.12 shows an example of an inversion coupling fault. When cell b, the aggressor, goes through a transition from 0 to 1, cell a, the victim, switches state. Figure 2.13 shows an example of a state coupling fault. When cell b, the aggressor, is a 1, cell a, the victim is also forced to be a 1. Please refer to Figure 2.9 for the interactions between two fully functional SRAM cells.

Coupling faults can be found by writing one value to the victim cell, writing first one value, then the inverse to the aggressor cell, then reading back the value from the victim cell. Because the victim and aggressor cells are not known initially, this test will take a longer time to test out.

Neighborhood Pattern Sensitive Fault

Neighborhood pattern sensitive faults (NPSF) are faults between many cells. The victim cell in this case is known as the base cell and will have its value changed when the other cells surrounding it are a certain pattern. There are three different kinds of NPSFs. an active fault will cause the cell to change value, a passive fault will cause the cell to stay at a certain value even if a different value is written to the cell, and a static fault will cause the cell to change to one specific state [14]. NPSF can only happen between cells that are physically near each other in the memory.

The base cell is read after changing the neighboring cells to different sets of patterns that could

possibly cause the fault to appear. Because each cell can act as a base cell, testing for a NPSF can be very complicated and will take a large amount of time. Due to the large amount of time and the fact that NPSF are not very common, they are not usually tested for [14].

Address Decoder Fault

Address Decoder Faults (AF), are faults within the address decoder that lead to issues when the memory cells are selected. Address Decoder faults include cells that are never accessed, addresses that access no cells, multiple cells accessed by one address, and certain cells accessed by multiple addresses.

2.2.5 Test Equipment

In order to logically test an SRAM chip, set input values are written to the cells, stored in the memory, read from the cells, and checked on the output. The logical tests are made of different sequences that can be used to check for issues in the memory. There are many different varieties of test equipment that can be used to test for errors in memory. In this section pattern generators and logic analyzers will be covered.

Pattern Generator

Pattern generators are used to create and run a sequence of logic 1s and 0s known as a test vector. The pattern generator can control what voltages correspond to a logic 1 and 0, when a logic 1 or 0 will be written to a cell, when the read and write cycles will take place, and which row of cells are accessed. They can also be used as an external clock for the chip during testing.

Logic Analyzer

Logic Analyzers are used for viewing digital waveforms. They display the voltages that they observe in the time domain which produces something very similar to a timing diagram. Logic Analyzers usually have between 34 and 136 channels that are each able to record and display information from a different digital input at the same time [15]. Logic analyzers only display two logic levels logic low and logic high. These values are set by the user before collecting data along with a threshold voltage. Any value below the threshold will be interpreted as logic low and any value above the threshold will be interpreted as logic high. A logic analyzer interfaces with

the device under test (DUT) through acquisition probes. Each connection on the probe has an internally compares the threshold voltage to an incoming voltage to decide if it is logic high or low [15]. Logic analyzers have many different trigger options including the ability to trigger off of a pattern in the data that the user defines, or an edge which looks for a threshold crossing [16]. In addition, the triggering position can be set to determine how much post and pre data is kept with respect to the trigger point. Logic analyzers use either a timing mode or a state mode when they collect data. Timing mode is used for asynchronous data collection. It uses an internal clock and will sample the incoming data at a preset sampling rate. State mode is used for synchronous data collection and will use one of the inputs as a clock to decide when to sample the data [16]. Logic analyzers can be used with pattern generators to test memory by displaying input and output logic information. The inputs and outputs can then be compared to find data errors which can be cell faults.

2.2.6 Digital Testing Patterns

This section covers the digital testing patterns used during the logical tests in order to find certain memory faults.

March Tests

A march test is a technique for sequentially reading and writing data to every SRAM cell in a certain address order. There are many different varieties of march tests that cover a wide range of fault models. A single march test is made up of many different march elements that consist of either a read or write operation [Pavlov]. Each operation is performed on every call and can be carried out in either increasing address order (starting at row 0 and ending on row n-1), or decreasing address order (starting at row n-1 and ending on row 0). Performing an operation in increasing order is denoted with an up arrow ($\Uparrow$), decreasing order with a down arrow ($\Downarrow$), and either order with a double arrow ($\Updownarrow$). A write operation is indicated with a w followed by either a 1 or 0 to show which logic level is being written to the cell. A read operation is notated the same way but with an r followed by a 1 or 0. The read operation tells you what value you should expect to see at the output. Some examples of this notation are w0, which means write a 0 to the cells and r1 which means read a 1 from the cells. One of the simplest march tests is known as the Modified Algorithmic Test Sequence, or MATS. Below is the MATS testing sequence [1][11].

MATS: ⇳(w0); ⇳(r0, w1); ⇳(r1);

The test starts by writing a value of 0 to all of the SRAM cells in either increasing or decreasing address order. Next, the first line of cells are read in either address order and the value 0 is expected at the output. After reading, a value of 1 is written to all of the SRAM cells in that location. After this the address is either increased or decreased depending on the chosen order, and the same read and write cycle is applied to the next set of cells. Finally, the cells are read in either address order and the value 1 is expected at the output. Each march test has different fault coverage and test length. The test length is usually measured in terms of the memory size, where N represents the number of cells. For example, the MATS test has a test length of 4N, because it goes through the memory a total for four times. Table 2.1 describes some of the more common march tests, their testing sequence, their test length, and some of the faults that they are able to cover [1]. In the table, AF stands for Address Decoder Fault, SAF stands for Stuck At Faults, TF stands for Transition Faults, and CF stands for Coupling Faults.

Table 2.1: Common March Tests

Name	Sequence	Length	Coverage
MATS	⇳(w0); ⇳(r0,w1); ⇳(r1);	4N	Some AFs, SAFs
MATS+	⇳(w0); ⇑(r0,w1); ⇓(r1,w0);	5N	AFs, SAFs
MATS++	⇳(w0); ⇑(r0,w1); ⇓(r1,w0,r0);	6N	AFs, SAFs, TFs
March Y	⇳(w0); ⇑(r0,w1,r1); ⇓(r1,w0,r0); ⇳(r0);	8N	AFs, SAFs, TFs, Some CFs
March C-	⇳(w0); ⇑(r0,w1); ⇑(r1,w0); ⇓(r0,w1); ⇓(r1,w0); ⇳(r0);	10N	AFs, SAFs, TFs, Some CFs
March C	⇳(w0); ⇑(r0,w1); ⇑(r1,w0); ⇳(r0); ⇓(r0,w1); ⇓(r1,w0); ⇳(r0);	11N	AFs, SAFs, TFs, Some CFs
March SS [12]	⇳(w0); ⇑(r0,r0,w0,r0,w1); ⇑(r1, r1, w1, r1, w0); ⇓(r0,r0,w0,r0,w1); ⇓(r1, r1, w1, r1, w0); ⇳(r0);	22N	AFs, SAFs, TFs, CFs

Many other march tests exist and more are constantly being created. March tests are one of

the shortest but most efficient tests for finding memory faults in SRAM.

Galloping Pattern Test

Another common test pattern is known as the Galloping Pattern Test, or GALPAT. The GALPAT test has a longer test length than many of the march tests, but is able to cover more faults. There are two different variations of the GALPAT test; gallop a 1 through a background of 0s, or gallop a 0 through a background of 1s. Below is an explanation of the GALPAT test [11].

GALPAT

1. Write a background 0 (1) to all of the memory cells.
2. Invert one cell to be a 1 (0)
3. Repeat the following sequence for all the memory cells, incrementing the non-inverting cells address with each iteration:
   (a) Read the inverted cell
   (b) Read a non-inverted cell
4. Switch back the inverted cell to be a 0 (1)
5. Invert the next cell to be a 1 (0)
6. Repeat sequence from Step 3 for all memory cells

Figure 2.14 shows an example of the GALPAT test for a four cell memory block. As the figure shows, the 1 appears to gallop across the cell. GALPAT covers all Address Decoder Faults, Stuck At Faults, Transition Faults, and Coupling Faults, with a test length of $4N^2+2N$.

0	0	1	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	1

Figure 2.14: GALPAT Pattern for a Four Cell Memory Block

2.2.7 Analog Testing

In order to understand the analog testing procedures, soft errors, and the remaining SRAM cell faults, the stability of the SRAM cell must first be explored.

Cell Stability

Stability is one of the most important factors in an SRAM cell design [17]. An unstable cell is unreliable and can cause incorrect data to be read from, written to, or stored within a memory cell. Modeling the noise margins of a cell can help to determine the stability and define the limits to which the cell will behave normally before it is fabricated. A total of five simulations can be run in each case, one for each process corner in order to show the extreme process parameter variations that could occur within the cell. The combinations were TT, FF, SS, FS, and SF, where the first letter stands for the NMOS device and the second letter stands for the PMOS device, with possible options being T-Typical, F-Fast and S-Slow process parameters. In addition to varying the process parameters, parametric analysis can be run to vary the cell temperature and supply voltage under each testing condition.

This section will explain how to determine the read, write and static noise margin for an SRAM memory cell in the standard 6T configuration, which may be seen below in Figure 2.15. As was explained in Section 2.1.4 The cell contains two access transistors (N_{AX1}, N_{AX2}), and two cross coupled inverters (N_1, P_1 and N_2, P_2) comprised of a PMOS load transistor, and an NMOS driver transistor.

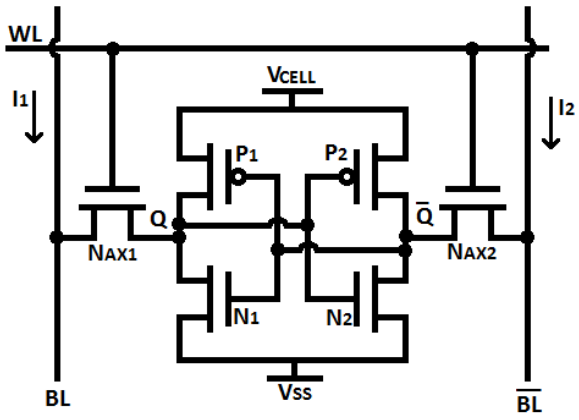


Figure 2.15: 6T SRAM Cell Configuration

The wordline (WL) of the cell is connected to the gate of both access transistors which are used to enable the cell when the wordline is connected to V_{DD}, so that values can be written to or read from the inverters via the bit line and its complement (BL and $\overline{BL}$). The read cycle should be non-destructive so that the cell will maintain its value unless it is overwritten. When the WL is brought low, the access transistors will turn off and the cell will hold a written value as long as V_{CELL} has a sufficient voltage to power the inverters. The inverters are bi-stable and their configuration causes a value to be stored at Q while its complement is stored at $\overline{Q}$.

Static Noise Margin

The static noise margin (SNM) of an SRAM cell is determined by how well the cell inverters will maintain their given state. The SNM is defined as the maximum voltage noise that can be tolerated by the inverter outputs without causing the cell to change states [17] [18]. The SNM can be found by looking at the Voltage Transfer Characteristic (VTC) of the cells inverter and its mirror, known as the butterfly curve, and drawing the largest square that can fit between the two curves [17] [18] [19]. Figure 2.16 shows an example butterfly curve with the boxes used to find the SNM.

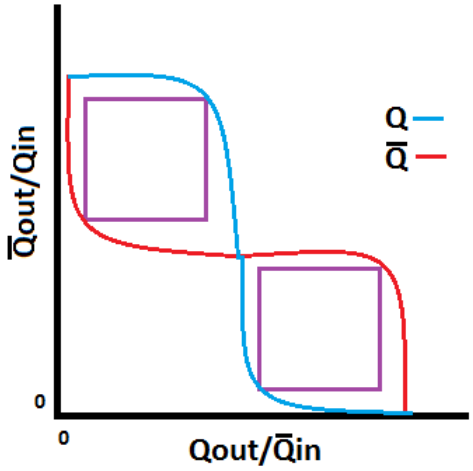
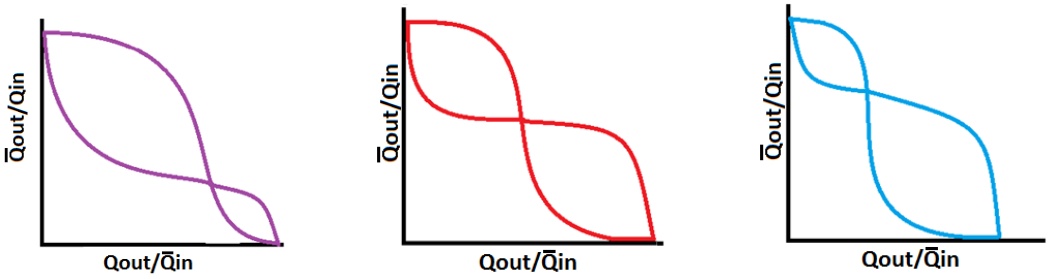


Figure 2.16: Static Noise Margin Butterfly Curve

Figure 2.17 shows some example Static Noise Margin Curves. Figure 2.17a has a poor noise margin. The upper half of the butterfly curve is significantly larger than the lower half which will cause the latch to tend towards node Q storing a logic high and node $\overline{Q}$ storing a logic low. Figure



(a) Bad SNM (b) Good SNM (c) Bad SNM

Figure 2.17: Example Static Noise Margin Curves

2.17c also has a poor noise margin. For this case, the inverters will have the opposite problem, causing the latch to tend towards node Q storing a logic low and node $\overline{Q}$ storing a logic high. Figure 2.17b is an example of an even inverter with a high static noise margin. This case will cause the latch to not be biased towards either logic level. If the SRAM cells do not have a high static noise margin, there is a higher chance of storing incorrect values within the memory array.

Read Stability

Determining an SRAM cells read stability can be done in a number of ways. One technique frequently used determines the Read Static Noise Margin (RSNM) by looking at the VTC of the inverters during a read cycle [19] [20]. Because the RSNM depends on the voltage supplied to the SRAM cell, the RSNM can also be measured by setting initial values so that a 0 stored at node Q, and a 1 at node $\overline{Q}$, then sweeping the cell supply voltage downwards while monitoring the current, I_2 [20]. When the cell supply voltage is no longer sufficient to power the inverters, the SRAM cell cannot store a state in its memory, which will cause the current, I_2, to steeply drop off. The RSNM is then measured as the difference between this voltage and V_{DD}. An example curve for measuring the RSNM may be seen below in Figure 2.18

Writability

This section explains the two different tests that can be run to determine the SRAM cells writability. The two tests find the bitline write margin and the wordline write margin.

Bitline Write Margin During a write cycle, the SRAM cell is enabled by bringing the WL high, while the bit line on the side of the inverter where the logic 1 was previously stored is pulled

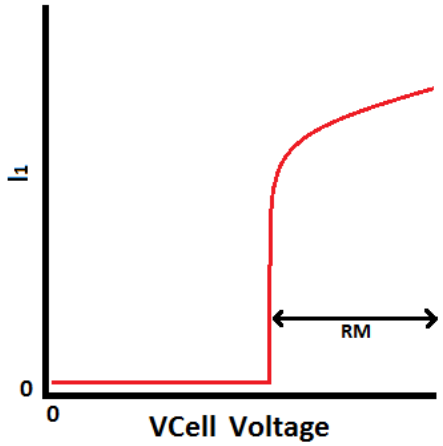


Figure 2.18: Read Margin

below the trip point, causing the cell to change state and write a 0. This method can be applied to either bitline to write a 0 or a 1 to Q. The first stability measurement for the write cycle is the Bit Line Write Margin (BLWM) and is defined as the highest bitline voltage that will still cause the inverter to change state [19]. To find the BLWM, the cell is set to initial values so that a 0 stored at node Q, and a 1 at node $\overline{Q}$, then the current, I_1, is observed while the voltage at BL was swept from V_{DD} to V_{SS} . Both the WL and BL are kept at V_{DD} during the sweep. The voltage at $\overline{BL}$ that causes the current at I_1 to drop significantly is the BLWM [20]. Figure 2.19 shows an example of the curve that the BLWM is measured from.

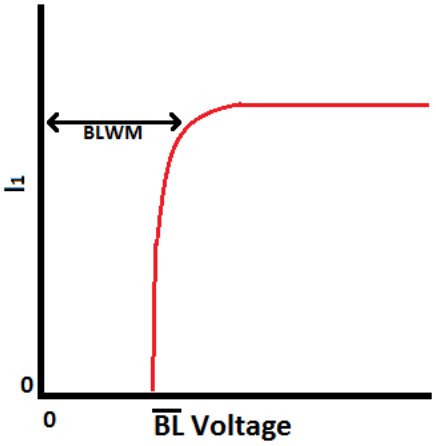


Figure 2.19: Bitline Write Margin

Wordline Write Margin The second stability measurement for the write cycle is the Wordline Write Margin (WLWM) and is defined as the maximum difference between V_{DD} and the WL voltage that occurs when the current curve returns to zero [20]. This point on the current curve corresponds to the inverters changing state. To find the WLWM, the cell is initialized so that a 0 stored at node Q, and a 1 at node $\overline{Q}$, then I_2 is observed during a WL sweep from V_{SS} to V_{DD}, while BL is held at V_{DD} and $\overline{BL}$ was held at V_{SS}. Figure 2.20 shows an example curve that the WLWM is measured from.

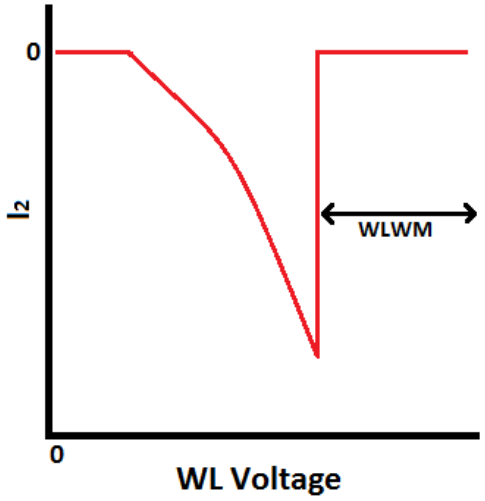


Figure 2.20: Wordline Write Margin

2.2.8 Soft Errors

Soft or transient errors are non-permanent errors that will cause a cell to act in a defective way under certain operating conditions. Soft errors will disrupt the stability of the cell, causing it to fail to maintain its proper state. They can be caused by alpha radiation and fabrication mistakes [1].

Stabilty Fault

Stability faults are caused by weak SRAM cells. Weak cells have issues with one of their noise margins. If the cell is not operating under ideal conditions, it will cause the cell to fail to retain its value. There are different degrees of stability faults, some cells may become unstable under almost all conditions, and other cells may only have issues under very specific circumstances. Some factors

that will affect cell stability include the voltage of the cell, temperature, and noise [1]. When the static noise margin is small for one value it will cause the inverter to flip states even when it should not [1]. Data retention faults are one of the most sever stability faults so they are considered in a category of their own. Stability faults are difficult to find because parametric analysis must be run with controlled testing conditions that can precisely vary the cell operating conditions.

Data Retention Fault

An SRAM cell should hold the value written to it as long as it has sufficient power. Data retention faults happen when a cell fails to hold its value even though there is still sufficient power to the cell. A cell that has a data retention fault will not be able to keep its correct value stored after a certain amount of time passes [1]. Data retention faults can happen for only one value, or for both values that a cell can store. Data retention faults are difficult to find because most tests will read the value saved to the cell shortly after writing it.

IDDQ Test

Another technique that is used to test semiconductor devices is known as the IDDQ test. The IDDQ test is able to find soft errors that do not affect the logical output, such as certain types of coupling and shorts in the system [1]. These errors often waste the power of the system and can lead to more severe issues such as data retention faults. The IDDQ test works by monitoring the quiescent current of the circuit, which is the current drawn by the system when it is fully powered but not active. This current is usually extremely small. If there is an error, such as a break, in the system, the current will rise by a few orders of magnitudes [1]. By monitoring the current for this change, certain defects that are otherwise undetectable can be found.

The IDDQ test is not commonly used in SRAM testing for a number of reasons including sizing and scaling. Due to the decreasing transistor size, the leakage current and parameter variation make it very difficult to accurately detect errors [1]. The IDDQ test has only been successfully used for devices that have a channel length of 0.25m and above, which is much larger than the current technology [1]. In addition to this, changes must be made to either the SRAM cell or the row decoders in order to access the whole memory cell which increases the footprint of the chip [21]. Despite this fact, several methods have been proposed for adapting the IDDQ test for current technology, but as sizing continues to decrease, the IDDQ test is becoming less reliable for accurate

fault detection [21][1].

Although it is not always a useful method for determining faults, IDDQ testing is still used to evaluate current flow and power consumption within memory devices.

Chapter 3

SRAM Sense Amplifier Design

One of the main goals for this Major Qualifying Project is to improve Allegro MicroSystems SRAM technology by designing and testing a new sense amplifier design. Before determining what changes can be made to improve the current sense amplifier design, the overall system level requirements must be defined outlined as is shown in section 3.1. Next, the operating characteristics and design limitations of existing sense amplifiers will be researched in section 3.2. After this, a new SRAM sense amplifier design will be chosen, sized, and tested as may be found in section 3.3.

3.1 Sense Amplifier Project Goals

Allegro Microsystems most significant goals are to decrease the total power consumption and access time of their SRAM chips. The focus of this project is to design a new sense amplifier that will be both faster and more power efficient than the current design. To meet this goal, the new sense amplifier design must consume less power than the current differential sense amplifier, The sense amplifier must be at least as fast as the current design. If the amplifier is too slow, it will increase the access time (as was discussed in Section 2.1.6), and decrease the maximum clock frequency that may be used with the chip.

Although these are the main objectives, the sense amplifier must meet certain other criteria as well. These requirements are listed below.

The sense amplifier must be modular and able to fit into the current SRAM design without

significantly affecting the other peripherals. Although slight modification may be necessary, the new design should make as small of an impact on the rest of the design as possible.

The layout area must be reasonable. Since each column must have a sense amplifier, any change in the footprint will have a dramatic impact on the overall size of the chip. A larger overall layout area also increases the parasitic capacitance. If the new design consumes too much physical space, the chip will become impractically large and it will not be worth the improvements it introduces

The sense amplifier must perform reliably during the read cycle and be resistant to noise within the system. In order for the sense amplifier to function properly, it must accurately read the values stored in the SRAM cells. If values are misread the majority of the time, the sense amplifier is useless.

Although the main goals of the new design are to reduce the power consumption and increase the performance of the amplifier, the modularity, layout footprint, and accuracy must be examined and considered during the design process. By taking into account all of these requirements, a robust and reliable sense amplifier will be created.

3.2 Sense Amplifier Designs

There are two main categories of sense amplifiers Differential sense amplifiers, also known as voltage mode sense amplifiers, and non-differential amplifiers, also known as current mode sense amplifiers [7]. Within the category of voltage sense amplifiers there are both static designs, which are latch based and read the difference between the bitlines once and hold that output, and dynamic designs, which constantly watch the difference between the bitlines and adjust their output accordingly [5].

Voltage mode sense amplifiers are connected directly to the bitlines. At the beginning of a read cycle, the bitlines are precharged. As one bitline begins to discharge, the difference between the bitline voltages determines the output logic value that is seen at the output. The performance of voltage mode sense amplifiers depends on the bitline capacitances. The larger the bitline capacitance, the larger the RC time constant of the system, and the larger the delay between enabling the cells and having a large enough difference between the bitlines for a voltage mode sense amplifier

to detect the value and show it at the output [5].

If the bitline capacitance is too large and the bitlines cannot be discharged in a reasonable amount of time, current mode amplifiers are sometimes cascaded and used as a buffering stage to disconnect the excessively capacitive bitlines and voltage mode sense amplifier inputs [5]. Because of this, current mode sense amplifiers are often found in large memory arrays that have significantly larger bitline capacitances. Current mode sense amplifiers may also be used independently. The amplifier inputs are connected to the bitlines, and after precharge, the discharging bitline causes a decrease in current flow which is used to determine the output voltage level.

Below are examples of some amplifier types, along with their pros, cons and characteristics.

3.2.1 Differential Amplifier

The differential sense amplifier is a dynamic voltage mode amplifier. Allegro Microsystems also currently uses this design in their SRAM chips. Figure 3.1 shows an example circuit diagram for the differential sense amplifier. The amplifier is composed of a differential pair (Transistors N_1 and N_2) with an active current mirror load (P_1 and P_2) and a biasing current source (N_3). The bitlines are connected to the gates of the differential pair transistors and the output is taken from the same side as $\overline{BL}$.

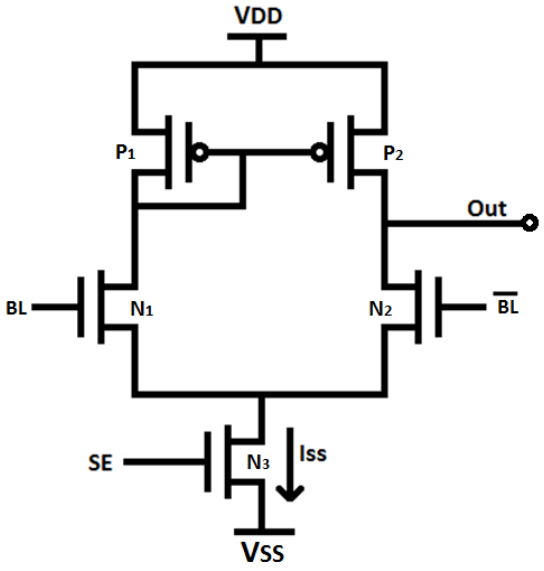


Figure 3.1: Differential Voltage Sense Amplifier

The differential sense amplifier is commonly used because it is simple and reliable [2][5]. The sense amplifier may be enabled at the same time as the wordline during a read cycle, making timing requirements for this amplifier very easy. The differential amplifier is resistant to supply noise variations due to its ability to reject common mode voltages; only differences between the inputs are amplified. Any noise that appears on both inputs of the amplifier will not affect the output.

In addition to this, the differential sense amplifier will almost never misread the cell because the output is constantly adjusted depending on the input voltage. A read cycle begins when the signal SE is brought high. This causes a constant biasing current, I_{SS}, to flow through the biasing current source (N_3). If both input voltages are exactly equal, this will cause a current equal to $\frac{I_{SS}}{2}$ to flow through both halves of the amplifier. To ensure that the current is divided equally, the PMOS devices P_1 and P_2 must be sized identically. The NMOS devices N_1 and N_2 must also be sized identically.

When the amplifier is first enabled, the input voltages are approximately equal because the bitlines are precharged. After the SRAM cells are enabled, one bitline will begin to discharge as was explained in Section 2.1.5. As the bitline discharges, the voltage supplied to the gate of the transistor on that side of the amplifier will decrease, causing the drain current to decrease as the transistor begins to turn off. Because the current source is supplying a total current to the amplifier equal to I_{SS}, the decrease in drain current on one side will cause the current in the opposite side of the amplifier to increase. As the transistor continues to turn off, the output voltage will rise to equal the supply voltage, V_{DD}. A relatively small difference between the bitlines will produce a large difference in output voltages.

In order for the sense amplifier to work properly, the biasing current source must remain on throughout the entire read cycle. If the biasing current stops flowing, a valid value will no longer appear at the output and the amplifier will not function[2]. This constant current flow causes the device to consume a significant amount of power throughout the entire read cycle.

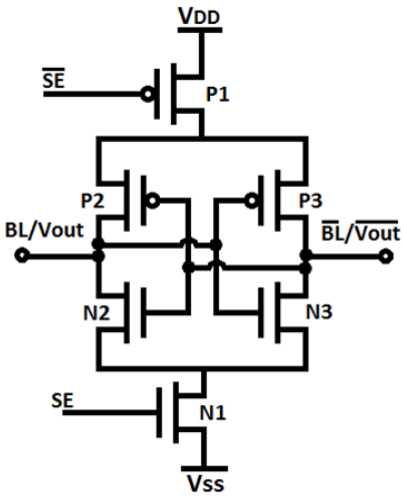


Figure 3.2: Basic Latched Voltage Sense Amplifier

3.2.2 Basic Latched Sense Amplifier

The basic latch based sense amplifier is a static voltage mode sense amplifier. An example schematic for the basic latch based sense amplifier may be seen in Figure 3.2. There are many variations of the basic latched sense amplifier, but this original design is one of the simplest and reliable choices [7][6]. The latch mechanism for the sense amplifier is composed of two cross coupled inverters (N_1, P_1 and N_2, P_2) which use positive feedback to cause latching behavior. The latched sense amplifiers voltage transfer characteristic curve may be seen in Figure 3.3. This plot shows that the amplifier has two stable points where V_{OUT} and $\overline{V_{OUT}}$ are at opposite rails and one metastable state in the middle of the transition region where V_{OUT} and $\overline{V_{OUT}}$ are at the same voltage.

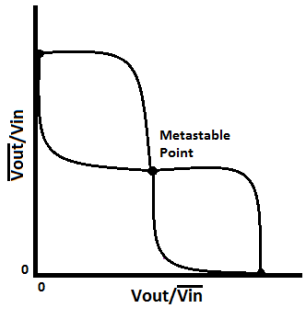


Figure 3.3: Latch Based Amplifier VTC

Before a read cycle, the bitlines which are connected to the amplifier inputs are precharged to

the supply voltage. If the amplifier were to be turned on right after the precharge stage, there would not be a sufficient voltage difference between the bitlines, which could cause the amplifier to stay at its metastable point or latch incorrectly. To solve this issue, the bitlines are allowed to discharge until they have created a large enough voltage difference for the sense amplifier to correctly identify the value stored in the memory cell. The discharging bitlines are connected to the inverter inputs, allowing a small bias to develop on the MOSFET gates before the sense amplifier is enabled and the inverters are turned on. To turn on the amplifier, the signal SE is brought high, which turns on the NMOS current source (N_3) and the PMOS access transistor (P_3). After being enabled, the bias allows the sense amplifier to latch the correct output value.

3.2.3 Basic Latched Sense Amplifier with Pass Transistors

One common variation of the basic latch based sense amplifier includes the addition of pass transistors [5]. Figure 3.4 This addition effectively decouples the amplifier inputs and outputs from the bitlines. Because the inputs and outputs of the sense amplifier are the same physical node, the sense amplifier will attempt to further discharge the bitlines while its reading the value stored in the SRAM cell. The pass transistors connect the input and output of the sense amplifier to the bitlines and are active when the sense amplifier is disabled. This allows the bias to still develop on the gates while the sense amplifier is off and the bitlines are discharging. Once the sense amplifier is enabled, the pass transistors turn off and the bitlines are decoupled from the sense amplifier inputs. Once the amplifier is enabled, the bitlines can no longer discharge. Unlike the amplifier without the pass transistors, this amplifier must account for the voltage drop over the pass transistors. This may lead to issues if one pass transistor is fabricated with a slight sizing error, as it may potentially bias the amplifier towards one logic value.

3.2.4 Current Controlled Latched Sense Amplifier

Another common voltage sense amplifier is the Current Controlled Latched Sense Amplifier. This design combines asspects from the latch based voltage mode sense amplifier and the differential pair amplifier. An example schematic may be seen below in Figure3.5. This design was introduced in 1993 by T. Kobayashi, and is a popular choice in SRAM memory [22][23][5]. The amplifier uses the same latch based design as the basic latch based amplifier that was described above in 3.2.2, but has the bitlines attached to the gates of transistors N_3 and N_4. This causes an extremely high

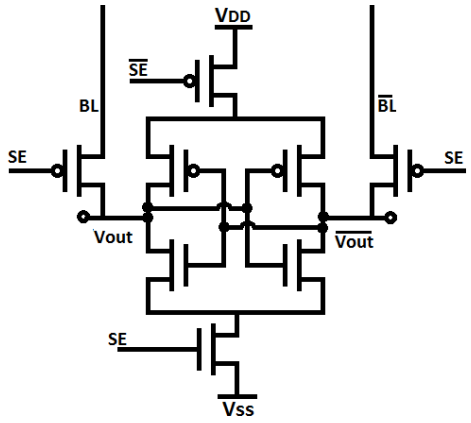


Figure 3.4: Latch Based Amplifier with Pass Transistors

impedance to be seen at the input, and the bitlines to be decoupled from the output . The two inverters are made from the MOSFET pairs N_1,P_1 and N_2,P_2. When the amplifier is off, the signal SE is at a logic low. This turns on the reset transistors P_3 and P_4, clearing the previous latched value and holding the latch at its metastable point. The bitline inputs are connected to the gates of MOSFETS N_3 and N_4. When the sense amplifier is enabled, the reset transistors turn off, and the current source N5 turns on, causing an even current to flow through each half of the amplifier. The bitline that is discharging causes the gate voltage to drop, which decreases the current flowing through that side of the amplifier and causes the voltage to rise. This will cause the amplifier to latch, which will stop the static flow of current and display a valid value at the output.

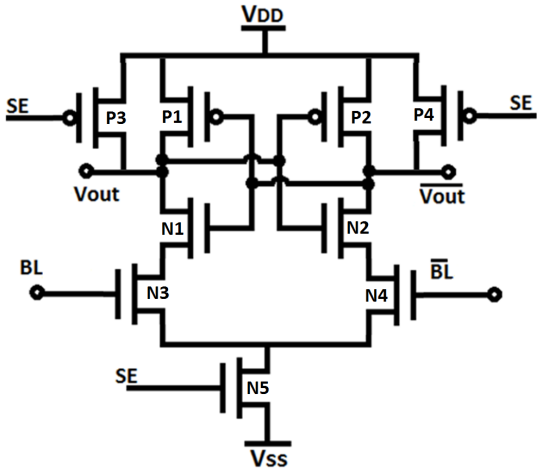


Figure 3.5: Current Controlled Latched Sense Amplifier

3.2.5 Amplifier Characteristics

Before determining which amplifier should be used, it is important to understand its overall characteristics and how it compares to the other amplifier options. Below, different properties of the various amplifiers are outlined.

Output Voltage Levels

The output of the amplifier should be capable of reaching both rail voltages. This will insure that the output is not miss interpreted by any following logic, and will also allow the amplifier to perform better with variations in the supply voltage. To meet this specification, and also to provide a high enough output current for driving any following logic, the amplifier output is connected to a buffer or an inverter, and the actual output is taken after this logic gate. All of the amplifiers meet this requirement.

Input Voltage Levels

The maximum possible voltage should be available at the amplifier inputs. This will allow a smaller difference between the bitlines to be detected by the sense amplifier which will decrease the access time during a read cycle. The maximum input voltage to the differential amplifier, basic latched sense amplifier, and current controlled latched amplifier is the same as the rail voltages. The maximum input voltage to the latched amplifier with the pass transistors is the rail voltage minus the drop over the PMOS pass transistors. This voltage difference is one of the disadvantages of the amplifier with the pass transistors. The impact of this decrease in voltage range may seem small, but it effectively decreases the noise margin of the sense amplifier and increases the chance of a misread, since a slight decrease in supply voltage can lead to the inverter transistors not having a sufficient voltage to keep them in their proper operating region during the read cycle. Also if the PMOS devices are fabricated with slightly different sizes, the difference in the voltage drop may bias the amplifier to latch to a certain value.

Input Impedance

Ideally the input impedance should be as high as possible for the sense amplifier. All of the amplifiers have their inputs connected directly to the gates of MOSFETs. The gate of a MOSFET

is a nearly perfect insulator and does not draw any DC input current, therefore causing the input impedance of the amplifier to appear virtually infinite [2].

Static Current Flow

An amplifier that has no static current flow will consume significantly less power during the read cycle. The differential sense amplifier requires a static current flow throughout the entire read process. If the current were to be disconnected, the amplifier would no longer produce a valid output. The three latch based designs will have no static current flow once the amplifiers have latched, because they have no direct path between the supply voltage and ground. This decrease in power consumption in one of the most significant benefits of the latch based sense amplifier designs.

Propagation Delay

The propagation delay determines how much time it takes to produce a value at the amplifier output once the chip is enabled. The propagation delay is measured from the 50% transition time to the full logic level at the output. The delay is caused by an RC time constant that is inherent to the physical design of the integrated circuit. The three factors that contribute to the time constant are the capacitance and resistance in the bitlines, the input impedance of the sense amplifier, and the resistance of the SRAM cell. Since all of the amplifiers will have either an inverter or a buffer driving the output, the propagation delay will be directly impacted by the choice of the output logic gate.

Slew Rate

The slew rate is the rate of change of voltage divided by the time it takes for the output of the sense amplifier to reach its final logic level. The amplifiers slew rate will be directly related to the transistor sizes that are chosen. This slew rate determines how quickly the output of the amplifier can change. The slew rate that is visible at the output is also affected by the parasitic RC time constant that exists within the integrated circuit similar to the propagation delay. The slew rate will also be directly impacted by the choice of the output logic gate.

3.2.6 Sense Amplifier Choice

After comparing the different sense amplifier design choices, the basic latch based amplifier that was described in Section 3.2.2 was chosen as the final design. One of the latch based designs was selected because there is no static current flow after the amplifier latches, which will cause a reduction in power. This design is simple, straightforward and documented well, making it an ideal choice within the available project time-frame. Ideally the maximum voltage possible should be available at the amplifier inputs, so that in future designs the supply voltage could be decreased. During the layout phase, the amplifier may be positioned near the write circuitry to produce a high impedance at the amplifier inputs, negating the need to decouple the inputs from the bitlines by using pass transistors. Thus, no pass transistors were included. The basic sense amplifier design is also the lowest transistor count out of any of the latch-based sense amplifier designs that were considered, which allows less space to be used for the overall layout area.

3.3 Sizing Analysis

As was discussed in Section 3.1 the two main goals are to choose transistor sizes that will lead to a low power and low delay amplifier. The sizing of the transistors in the sense amplifier impacts nearly every characteristic of the amplifier performance. Sizes will be chosen to minimize power consumption while maintaining amplifier stability. The sense amplifier will be tested with various sizes using both hand analysis and Ocean simulation within Cadence Virtuoso Design Suite. Using both these simulation techniques, optimal sizing for each transistor may be found.

3.3.1 Sizing Overview

One of the most important steps in designing the SRAM sense amplifier is determining the transistor sizes. Different transistor sizes will affect the region that the MOSFETs operate in, the timing of the system, the power consumed during the read cycle, the performance of the amplifier, and the overall layout area. The sizes should be chosen to minimize the area and footprint while still meeting power and performance specifications.

Figure 3.6 shows the basic design of an N-channel MOSFET. The P-channel MOSFET is similar, but uses the opposite dopings for the substrate, body, drain, and source areas. As the

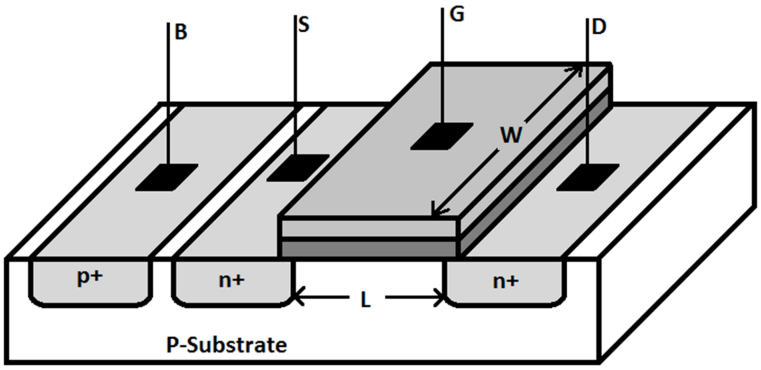


Figure 3.6: MOSFET Cross-Section View

figure shows, the two variables that may be changed during the sizing process are the channel width (W), and channel length (L). The oxide thickness and doping concentrations are directly related to the CMOS fabrication process that is used and will remain constant for the entire wafer.

The MOSFET drain current equations are often referenced while determining MOSFET sizes. The transistor equation for the drain current when the MOSFET is in triode region may be seen in Equation 3.1 and Equation 3.2 for the NMOS and PMOS devices, respectively. The MOSFET equation for the drain current when the MOSFET is in saturation region may be seen in Equation 3.3 and Equation 3.4. In these equations, μ is the carrier mobility, C_{OX} is the gate oxide capacitance, W is the MOSFET width, L is the MOSFET length, V_{GS} is the Gate to Source voltage, V_{th} is the threshold voltage, and V_{DS} is the Drain to Source voltage. Often times the parameters that can not be changed after fabrication, ($\mu C_{OX} \frac{W}{L}$) are grouped into the transconductance variable k.

$$I_{Dn} = \mu_n C_{OX} \frac{W}{L} \left((V_{GS} - V_{th}) V_{DS} - \frac{V_{DS}^2}{2} \right) \tag{3.1}$$

$$I_{Dp} = -\mu_p C_{OX} \frac{W}{L} \left((V_{GS} - V_{th}) V_{DS} - \frac{V_{DS}^2}{2} \right) \tag{3.2}$$

$$I_{Dn} = \frac{\mu_n C_{OX}}{2} \frac{W}{L} (V_{GS} - V_{th})^2 \tag{3.3}$$

$$I_{Dp} = -\frac{\mu_p C_{OX}}{2} \frac{W}{L} (V_{GS} - V_{th})^2 \tag{3.4}$$

Amplifier Sizing Plan

The length to width ratio of the MOSFET will determine the way that the MOSFET will behave in different situations. The basic latch based sense amplifier as shown in Figure 3.7 consists of six total transistorsfour of which must be sized. The latch within the sense amplifier consists of a pair of cross coupled inverters which will be identically sized that is P_2 will be the same size as P_3, and N_2 will be the same size as N_3. In addition to the latch, the sense amplifier contains the PMOS transistor P_1 which is used to enable and disable power to the latch, and the NMOS transistor N_1 which is used as a current source for the sense amplifier. The different components of the sense amplifier will each have different effects on the overall design. Because of this, the amplifier will be designed in three separate parts. The sense amplifier will have the inverters for the latch sized first, followed by the current source transistor and the enable transistor.

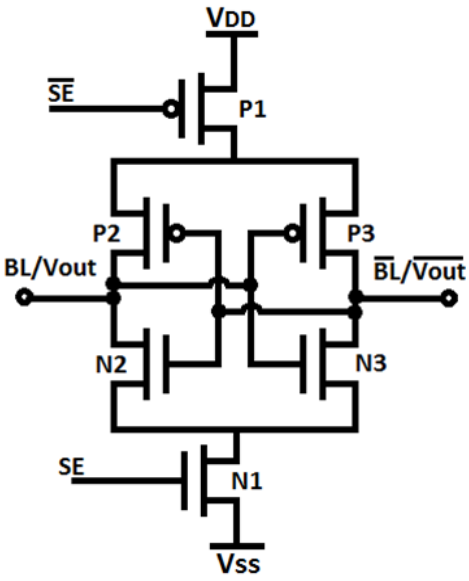


Figure 3.7: Latch Based Sense Amplifier

3.3.2 The Inverter

The sense amplifier with the inverters highlighted may be seen in Figure 3.8 . In this section, the inverter sizes will be chosen.

The speed of the latch depends on the sizes chosen for the two inverter transistor pairs P_2-N_2,

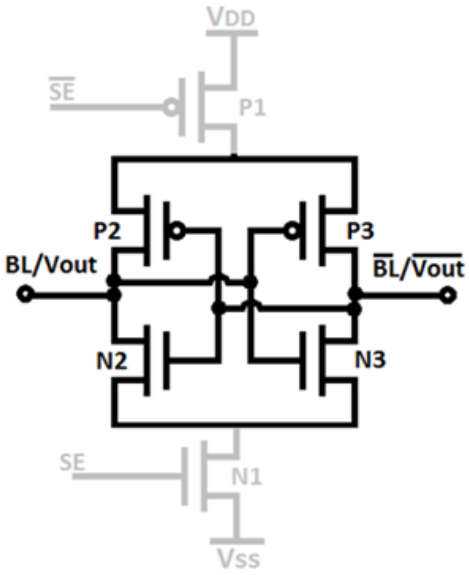
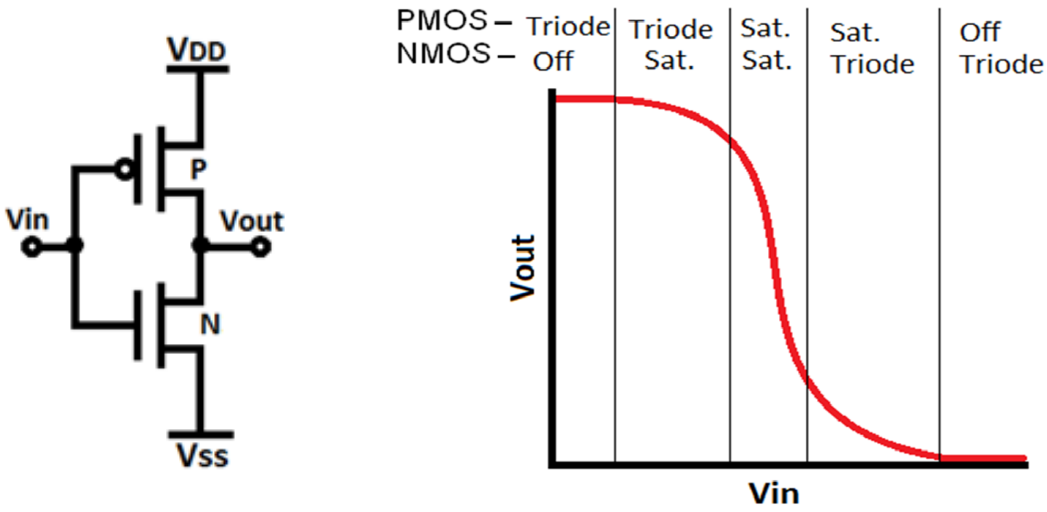


Figure 3.8: Highlighted Latch in Sense Amplifier

and P_3-N_3. To ensure that the transistors take up the smallest area, the transistor length will be held at the minimum value for the CMOS process, while the widths are varied to meet the other requirements. Looking at the voltage transfer characteristic for different sizing options can help to determine the sizes that are most suitable for the inverter. Figure 3.9b shows a sample voltage transfer characteristic for an inverter. Figure 3.9a shows the basic inverter circuit diagram.



(a) Inverter

(b) Opperating Regions

Figure 3.9: Inverter Circuit Diagram and Charictoristic Voltage Curve

Based on the curve above, the sense amplifier will have transistors that are both off, and in the triode region once the amplifier has latched. When a MOSFET is in the triode region, it is acting like a variable resistor whose value is controlled by the total voltage applied to the gate. Once the circuit has latched, no static current will flow because there will be no path from V_{DD} to V_{SS}, as one PMOS and one NMOS transistor will be off in each inverter.

For the latch, a fast response is desired so that the transistors enter triode mode quickly because no static current flows once the amplifier has latched. To make the response as fast as possible, the current should not be limited by the latch. As Equations 3.1 and 3.2 show, a larger current and faster latching can be made possible by having the width (W) larger than the length (L). Because of this, the inverter transistors will be sized for a larger width, and a minimum length.

One of the most important points on the inverter voltage transfer characteristic curve is the inverter threshold voltage or the voltage which is halfway between V_{DD} and ground. This point is also the intersection point on the inverter butterfly curve, which is used to determine the inverters stability. The technique for determining the stability of the latch inverter is exactly the same as the technique used to determine the SRAM cell inverter that was described in Section 2.2.7.

The static noise margin is defined as the maximum voltage disturbance at the input of the latch that the device can handle without changing states [17]. This is one of the most important considerations when sizing an latch. The noise margin should be as even as possible to ensure that the sense amplifier will latch correctly when it is exposed to different bitline voltages. The best sizing for the latch will therefore be with inverters that have comparable noise margins. As a starting point for the sizing analysis, it is estimated that the speed of the free charge carriers in PMOS devices, known as the hole mobility, is approximately half the speed of the free charge carries in NMOS devices, known as electron mobility. Therefore to size a stable inverter, the PMOS device must be nearly double the size of the NMOS device.

Figure 3.10 shows the voltage transfer characteristic for different inverter transistor widths/length ratios where $k_n = \mu_n C_{ox} \frac{W_n}{L_n}$ and $k_P = \mu_p C_{ox} \frac{W_p}{L_p}$. Since $\mu_n \approx 2\mu_p$, the width of the PMOS device needs to be approximately double the width of the PMOS device in order for kp to equal kn. By varying the relative sizes of the transistors, different amplifier transfer characteristics can be found. As the sizing changes, one transistor will stay in saturation mode longer than the other.

With a smaller width, the current density in the MOSFETs conducting channel will increase and cause pinch-off to occur sooner, which will shift the threshold voltage, or monostable point of the inverter.

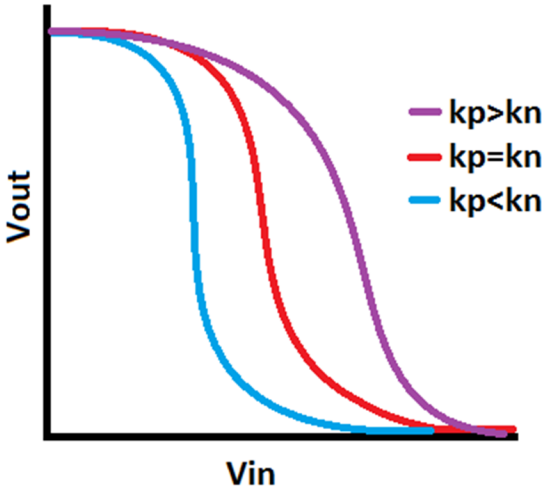
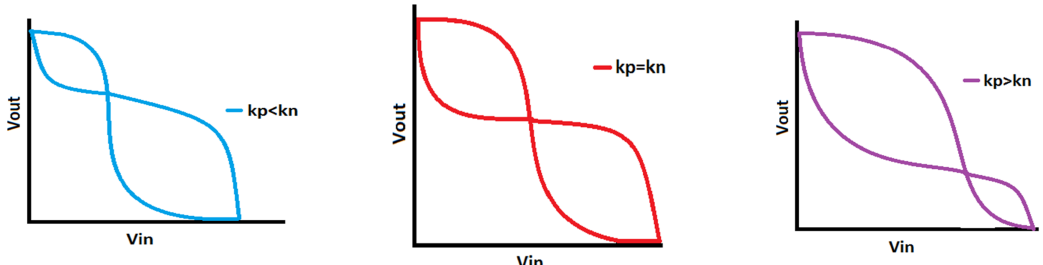


Figure 3.10: Inverter Curves

Changing the inverter sizes directly affects the inverter threshold voltage, which also affects the stability of the inverter. Figure below shows the static noise margins for the three inverter curves in Figure 3.10. The two inverters with bad static noise margins will tend to latch as one value much more easily than as the other, greatly increasing the chance of incorrectly latching or switching states, providing extraneous values. The purple figure will tend to latch to a logic high, and the blue figure will tend to produce a logic low.



(a) Bad SNM

(b) Good SNM

(c) Bad SNM

Figure 3.11: Example Static Noise Margin Curves

Cadence Simulation Sizing Analysis

To find the optimum sizes for the sense amplifier, multiple simulations known as parametric analyses were run using Cadence Design Suite. To size the inverter, the smallest NMOS device size was chosen, and a parametric analysis was run to vary the PMOS device size while monitoring the voltage transfer characteristic. The PMOS device was then chosen based on the size that led to the inverter threshold voltage at exactly the halfway point as this will produce an even latch and a high noise margin.

To determine the voltage transfer characteristic, the input voltage of the inverter was swept from V_{SS} = 0V to V_{DD} = 3V. The results for the DC sweep may be seen below in Figure 3.12 where the NMOS device was held at its smallest allowable value and the PMOS device length was held at its smallest allowable value, and its width was swept from half the size of the NMOS device to four times the size of the NMOS device.

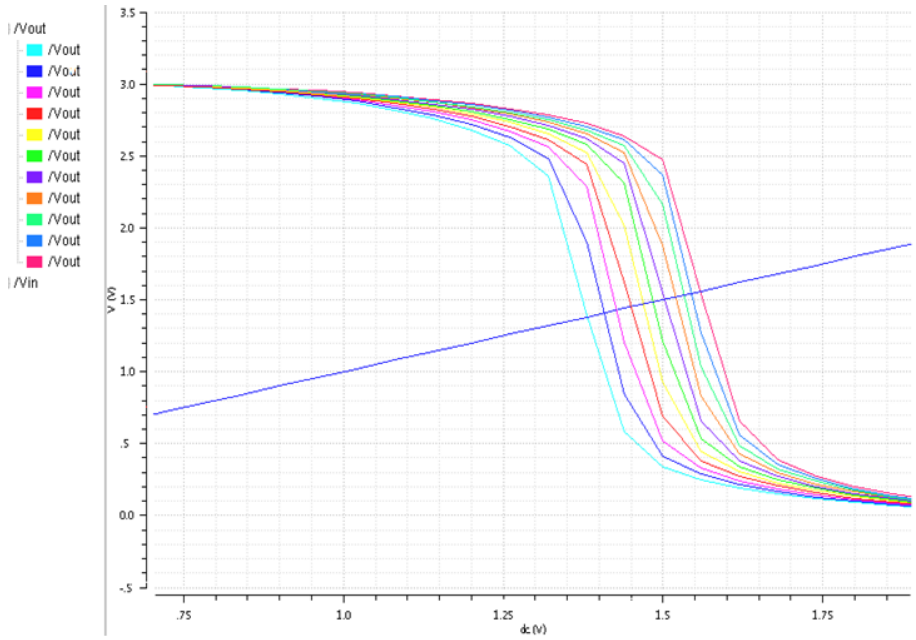


Figure 3.12: Inverter Parametric Analysis Results

After running the parametric analysis, the PMOS width that crossed closest to midway was chosen for the latch. The voltage transfer characteristic for the inverter with the final sizing can be seen below in Figure 3.13. The threshold voltage for the inverter is extremely close to the halfway point, which means that it has a very good static noise margin and will be stable.

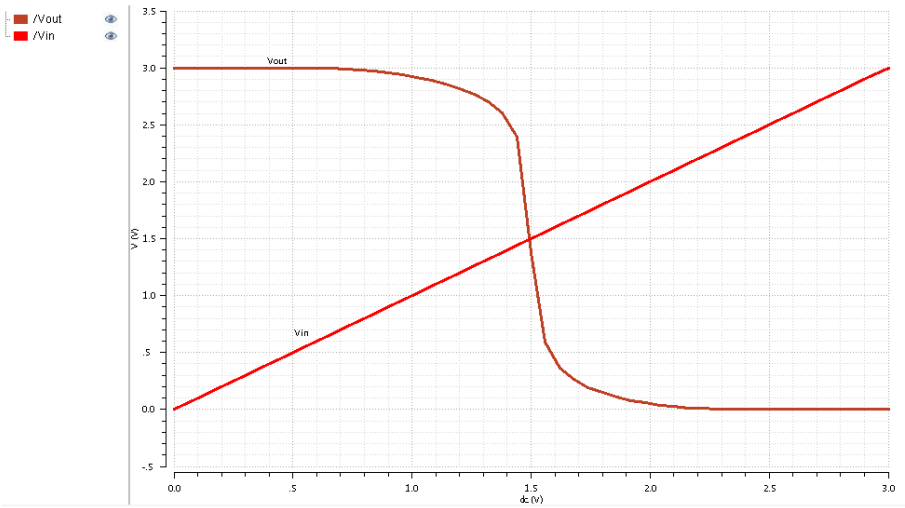


Figure 3.13: Final Inverter Voltage Transfer Characteristic Curve

The static noise margin curve may be seen in Figure 3.14. The even lobes on the butterfly curve confirm that the noise margin will be high.

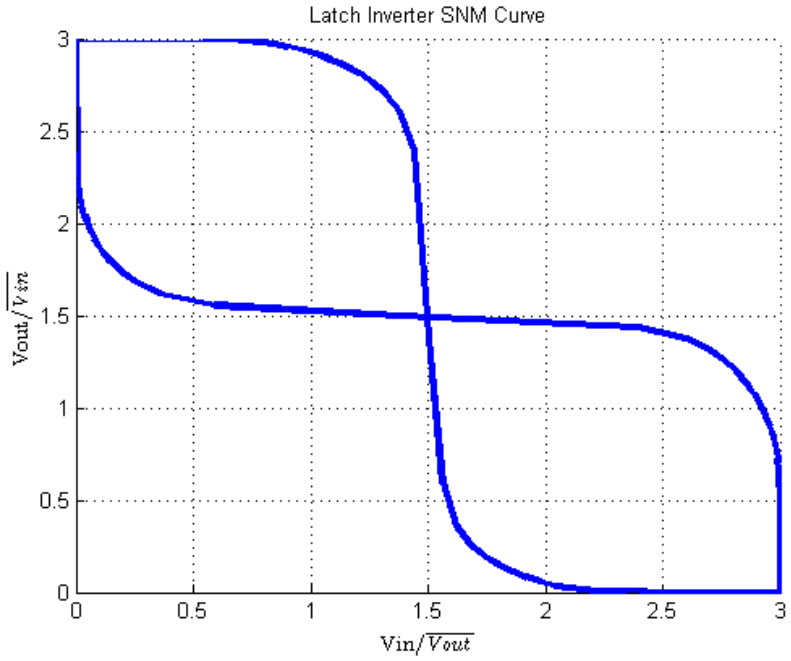


Figure 3.14: Inverter Static Noise Margin

3.3.3 The Access Transistor and Current Source

he next part of the sizing analysis is to choose the PMOS access transistor size (P_1) and the NMOS current source transistor size (N_1). The circuit diagram for the sense amplifier with these two transistors highlighted may be seen in Figure 3.15.

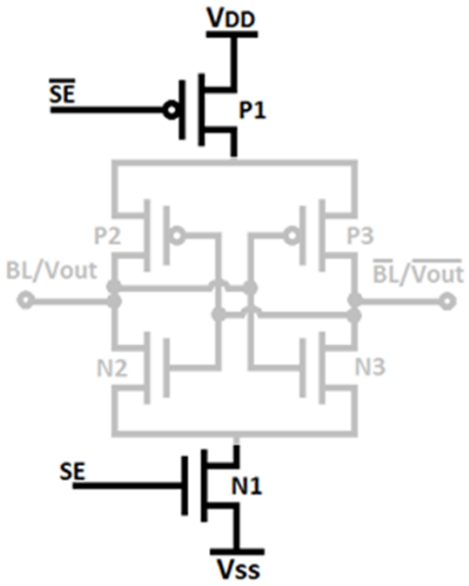


Figure 3.15: Current Source and Access Transistor

The PMOS access transistor connects the amplifier to the power supply V_{DD}. If the PMOS device were not enabled, the amplifier would not turn on and would therefor not be able to function. When the amplifier is enabled, the PMOS transistor should turn on as quickly as possible to decrease the access time. The size of this transistor can be freely varied as it is not the device setting the current limit for the system. To allow for the fastest enable as possible, the PMOS transistor will have a larger length (L) than width (W), which will allow as large of a drain current to flow as the limiting current source allows. The relationship between the drain current and $\frac{W}{L}$ ratio may be seen above in Section 3.3.1 in Equations 3.1 and 3.2 and Equations 3.3 and 3.4.

The NMOS current source size is chosen differently than the other five MOSFETs. When the amplifier is enabled, this device creates a constant current flow through the other transistors in the amplifier. One way of reducing the power consumption of the sense amplifier as a whole is to limit the current that is allowed to flow through the system before the amplifier latches. In

order to do this, the MOSFET is sized so that the width (W) is larger than the length (L) which effectively limits the current and reduces the power consumption. This relationship effect may be seen above in Section 3.3.1 in Equations 3.1 and 3.2 and Equations 3.3 and 3.4. Unfortunately there is a trade off between the power consumption and access time. If the width is made too large, the low amount of current through the amplifier will cause the amplifier to latch slower, which will increase the access time. Because of this issue, the power and the access time will both be considered while choosing the current source and access transistor sizes.

To size the current source and access transistor, a series of parametric analyses are run using an Ocean script through Cadence Design Suite. Figure 3.16 shows a basic flowchart for the ocean script. The NMOS width and PMOS length were held constant at the minimum value for the process technology, then the other variable was swept through multiple sizing options. By using Ocean, the timing delay and power consumption for each iteration of size were recorded in an output file. Using Matlab, the access time and power consumption were then plotted for each size iteration.

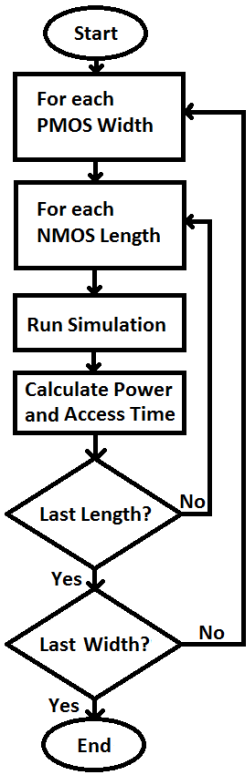
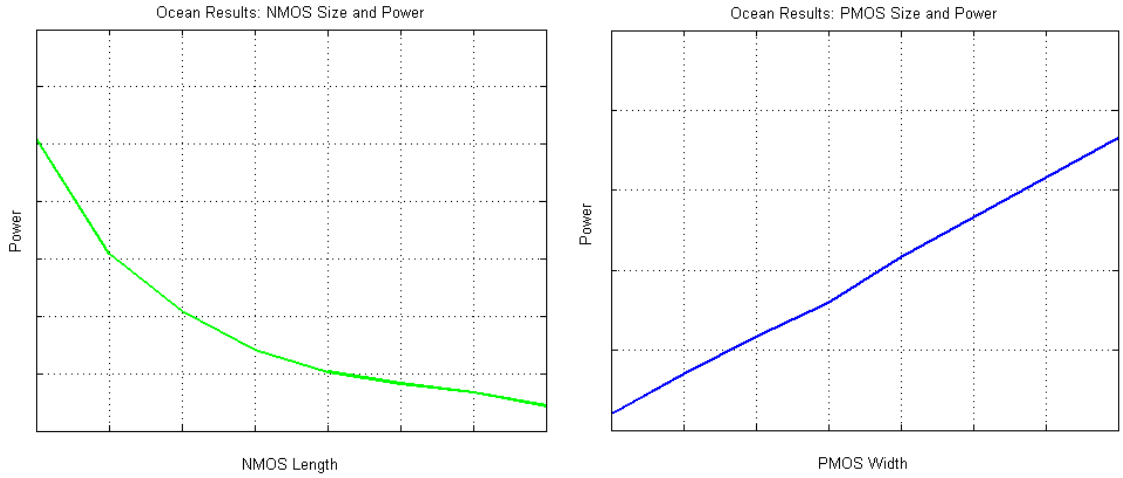


Figure 3.16: Ocean Flowchart

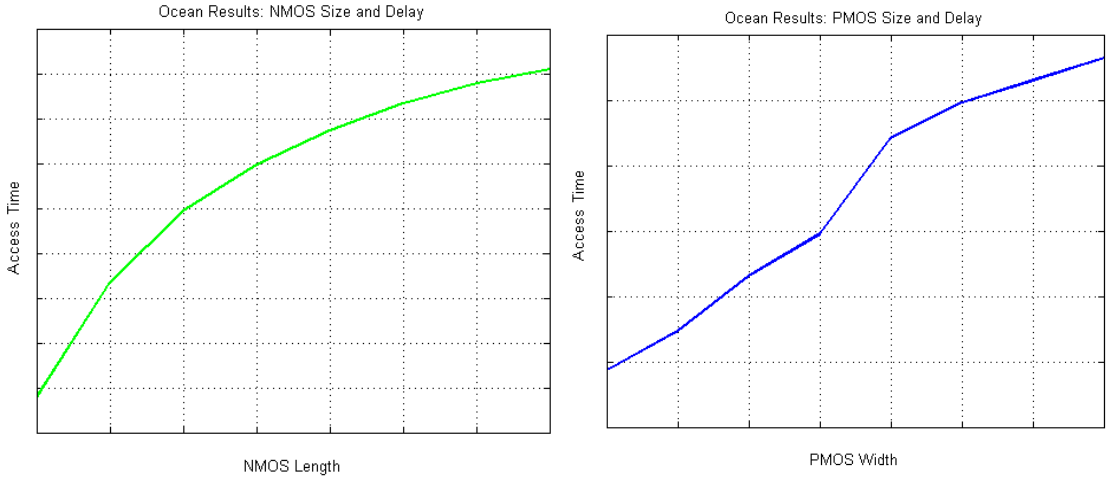
Figure 3.17 shows the overall affect of the device sizes on the power consumption. Figure 3.18a shows that as the NMOS Length increases, the power consumption decreases and Figure 3.17b shows that as the PMOS Width increases, the power consumption increases. Figure 3.18 shows the overall affect of the device sizes on the delay, or access time. Figure 3.18a shows that as the NMOS Length increases, the access time increases and Figure 3.17b shows that as the PMOS Width increases, the access time increases.



(a) Affect of NMOS Length on Power

(b) Affect of PMOS Width on Power

Figure 3.17: Size and Power Consumption



(a) Affect of NMOS Length on Access Time

(b) Affect of PMOS Width on Access Time

Figure 3.18: Size and Access Time

After plotting the power and delay from the sizing analysis with respect to the changing size, the results were compared to each other directly. Figure 3.19 shows the resulting trends from the sizing analysis.The solid green lines represents the changing NMOS length, and the solid blue lines represents the changing PMOS length. As the NMOS length increases, the power consumption decreases, but the access time increases. This is what was expected, since the current will be limited with a larger length, as was explained in Equations 3.1 and 3.3. As the PMOS width

increases, the power consumption and access time both increase. Since the PMOS device is acting as a switch, a smaller width will allow the device to turn on faster, which will cause a decrease in power consumption and access time. Values for the NMOS length and PMOS width that are a good trade off between the power consumption and the access time were then selected. During testing and layout, these values may be changed slightly as needed since the trend of the power consumption and access time is known.

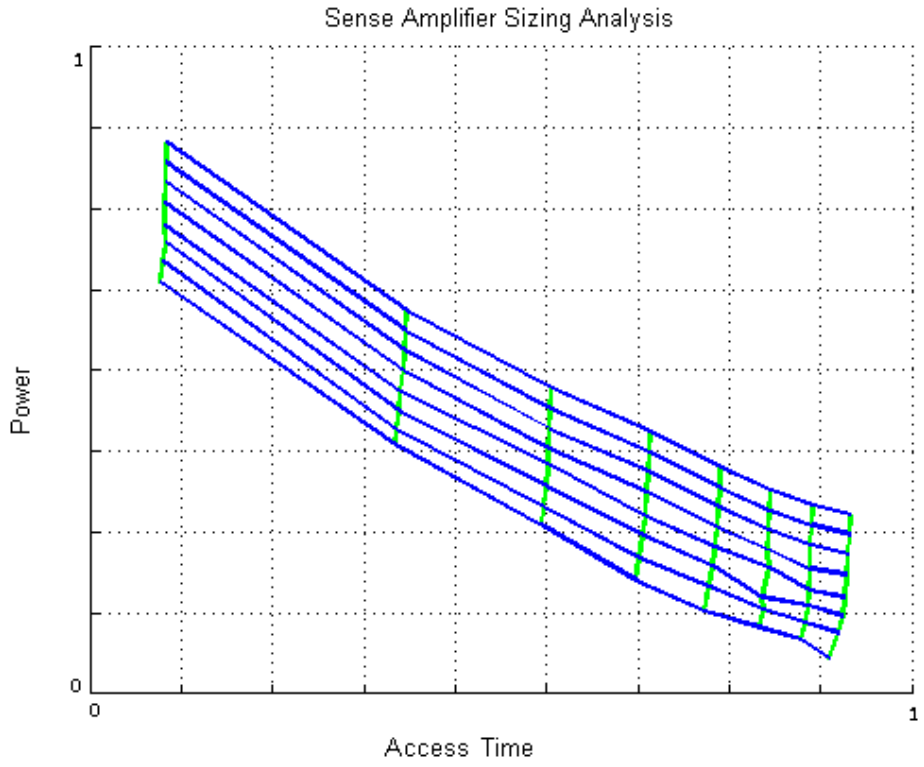


Figure 3.19: Current Source and Access Transistor Sizing

Chapter 4

SRAM Results

After completing the sizing analysis for the new sense amplifier, the design must be throughly tested both on its own and as a part of the SRAM test chip that will be fabricated. It is extremely important that all components that interact with the new sense amplifier are checked and simulated carefully. If the timing is incorrect in any area, the SRAM chip will not function as it should, which will cause errors and possibly destroy the chip. To prepare for fabrication, a physical layout of the design must be created as well. The final chip that the sense amplifier will be integrated with has similar components to those explained in Chapter 2. Figure 4.1 shows an overview of the final chip along with its components.

The test chip will be synchronous with an external clock source. This clock will be used to determine all the timing for the rest of the SRAM chip. Other inputs include the chip enable, which determines if the SRAM is in active mode and able to read or write values or if the chip is just holding the stored data; input address, to determine which wordline is selected; Read/Write enable, which determines if data is being read from or written to the cells; and the input data which will be stored in the cells. The one output from the chip will be the output data that is read from the SRAM cells by the sense amplifiers.

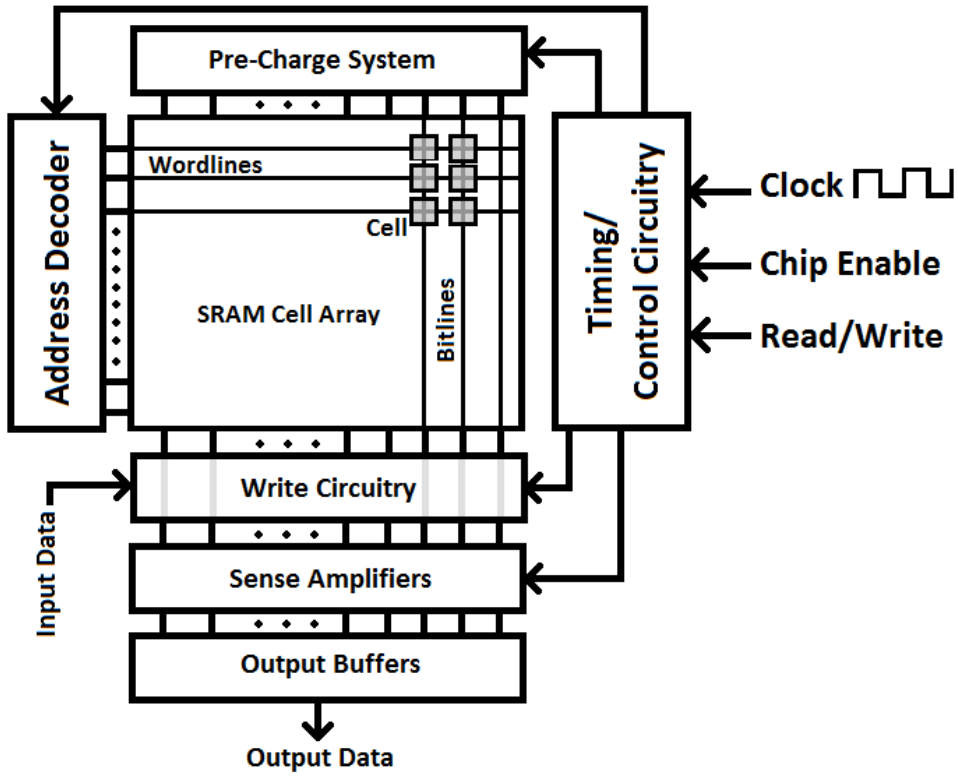


Figure 4.1: SRAM Final Block Diagram

4.1 Layout

After the transistor level design of the SRAM chip is complete, it must be transferred into a silicon level design. This is known as the layout and will be used during the fabrication process. The layout is exactly what will be created on the silicon wafer during fabrication. The layout for each unit of the SRAM chip is first created by itself, and then successively connected to the rest of the design. For example, the layout for a single sense amplifier will first be created on its own, then this layout will be copied and connected to each column of SRAM cells. The total number of sense amplifiers in the final layout will be the same as the SRAMs word size, as each pair of bitlines can share the read and write peripherals.

For this design, the total layout width of the sense amplifier needed to be equal to the width of one column within the memory matrix. This allows the amplifiers to be aligned with the bitlines. To begin the layout, both n-type and p-type wells are placed and the transistors for the amplifier

are placed. The approximate area for each transistor is its width times its length. After the individual transistors are placed, connections are made by placing vias and running metal layers. In order to decrease the risk fabrication defects, certain distances must be kept between layers of the same material. Along with the six transistors that make up the basic latched sense amplifier, an output buffer stage is also included in the overall sense amplifier layout.

After the transistors have been placed and connected, a number of tests must be run on the completed layout. The first test that is completed is the DRC, or Design Rule Checking test [24]. This series of tests goes through the layout and checks to make sure that no design rules have been broken during the creation of the layout. Some of the layout rules are caused by manufacturing limits such lithography and etching. These rules determine things such as the minimum width and spacing that must exist between the same net. This example may be seen in Figure 4.2. Other elements have a set size that must be used, such as for contacts and vias. If larger areas are needed, they must be created by repetition. Figure 4.3 shows an example of this requirement.

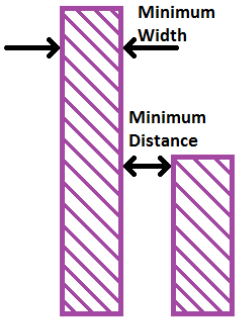


Figure 4.2: DRC Example

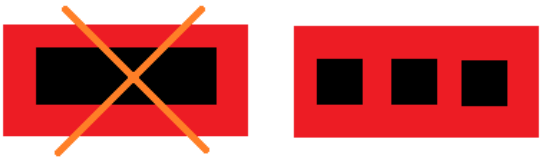
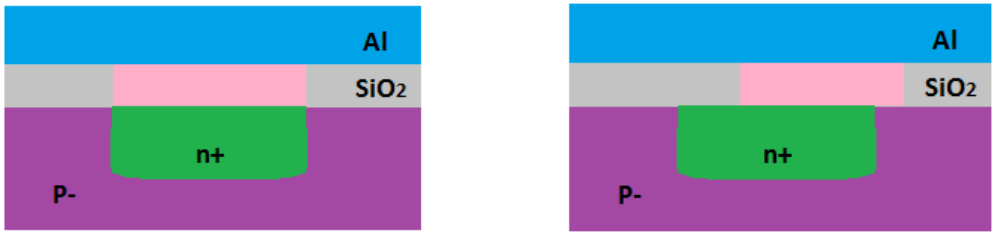


Figure 4.3: Incorrect and Correct Setup for Larger Structure with Set Size

Another criteria that must be met is the minimum enclosure or overlap that is allowed between nets. This requirement is due to the alignment precision of different masks [24]. Figure 4.4a shows an example of a MOSFET drain or source with a large contact attached. In this figure, the contact is perfectly aligned. Figure 4.4b shows an example of the same MOSFET drain or source with a slight process variation that has caused a misalignment in the contact mask. Since the contact is so large, this slight deviation has caused a short circuit between the drain or source and the substrate. To counteract this issue, a smaller contact is used, as is shown in Figure 4.5. If the

minimum enclosure or overlap rule were not used, there would be a higher likelihood of shorts between certain levels, as was shown above.



(a) Ideal Wide Contact (b) Misaligned Contact Mask

Figure 4.4: Issues of Large Contact Area

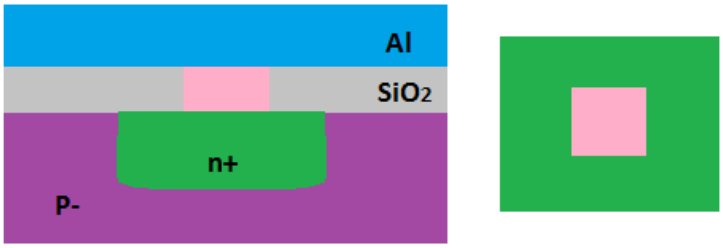


Figure 4.5: DRC Overlap Rule

In addition to the DRC testing, LVS or Layout Vs Schematic testing must also be performed [24]. This variety of testing compares the silicon layout to the original schematic. Errors such as incorrect MOSFET sizes and broken connections are analyzed. During the fabrication process, the schematic is not considered, so it is extremely important that the layout matches the actual schematic. After these levels of testing are completed, the layout for the single sense amplifier is ready to be added into the next step of the design. Each step must be tested using the DRC and LVS techniques, so that the final layout is an accurate representation of the design.

4.2 Testing

After completing the design and sizing analysis for the latch based sense amplifier, the functionality and performance of the amplifier must be thoroughly tested. Testing will be broken up into three main phases: unit testing, integration testing, and system testing.

Unit testing involves isolating and testing individual parts of a system. Here, the functionality and electrical performance of the sense amplifier will be explored. The sense amplifier will be modeled in Cadence Virtuoso Design Suite using the transistor sizing determined in Section 3.3.2 and Section 3.3.3. The functionality of this design will be verified and the electrical characteristics analyzed. The performance of the amplifier will be defined by the slew rate and access time found in this stage.

In the integration testing phase, multiple blocks are interfaced with one another and tested. The goal of this phase is to verify the timing and analog functionality of the sense amplifier. To do this, the amplifier is integrated into the final design and multiple tests are run. This includes testing with paracitics, verifying driver voltage response, and finalizing timing for the system.

Finally, the entire chip will be tested as a whole during system testing. This final testing will be as close to the real world conditions as can be simulated. Verilog will be used to run rapid tests that cover aspects such as Write/Hold/Read cycles, logic levels, and multiple cell outputs for the entire SRAM chip. The results of this test are the final verification of functionality and performance before the chip is fabricated.

4.2.1 Unit Testing

The goal of the initial unit testing is to verify the functionality of the sense amplifier with the final transistor sizes that were chosen in section 3.3. The simulations were performed using Cadence Virtuoso Design Suite.

To unit test the sense amplifier, a very simplified SRAM test setup is created. This setup consists of the latch based sense amplifier and a dummy cell. The dummy cell is a single SRAM cell connected by the bitlines to the amplifier inputs. This test ensures that the amplifier will actually read out the correct value stored in an SRAM cell. Parasitic capacitances are added to model the capacitance in the bitlines and at the output. These capacitor values are approximated by looking at a previous layout for a preexisting SRAM chip that is similar to the final top-level design that the new sense amplifier will be added to. To run the simulations, a logic high is pre-stored in the dummy SRAM cell. Power is supplied to the cell so that the value is held for a certain amount of time, then the cell is enabled, allowing a voltage difference between the bitlines

to develop at the amplifier inputs. After this, the amplifier is enabled and the values are read from the cell and examined at the output. This procedure is then repeated with the opposite logic level stored in the dummy cell.

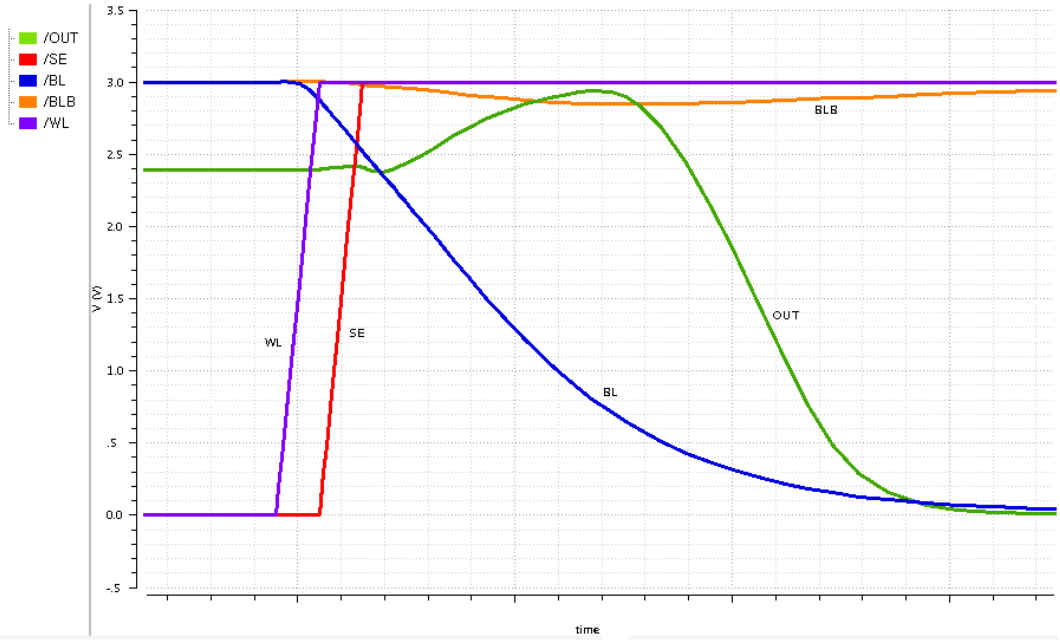


Figure 4.6: Unit Testing Results

The output response of the sense amplifier with a logic low stored in the dummy cell may be seen in Figure 4.6. Before starting the simulation, the bitlines are precharged and a logic low is stored in the dummy cell. First, the wordline (purple line, labeled WL) is brought high, enabling the SRAM dummy cell. This allows the bitline voltage (blue line, labeled BL) to begin decreasing. Once a sufficient voltage difference has developed between bitline and $\overline{bitline}$ (orange line, labeled BLB), the sense amplifier is enabled by bringing the sense enable signal (red line, labeled SE) high. At this point, the amplifier output (green line, labeled out) latches, but there is a delay before seeing the final value at the output because of the propagation delay.

4.2.2 Integration Testing

After the unit testing is complete, the new amplifier design is integrated in to the top-level chip, and is tested with individual system blocks. First, parasitics were extracted from the layout and added to the output, the bitlines, the timing circuitry, and the data inputs. Its especially

important to include the parasitics because the timing of the system is very fast. Next, a variety of simulations were run to verify the timing response. These were performed to make sure that there was never a direct path from the supply voltage to ground—for example, verifying that wordlines were never held high during a precharge cycle. After this, the timing for enabling the sense amplifier was determined. A large enough voltage difference must develop between the bitlines before the sense amplifier is enabled. If the amplifier is enabled too soon, there is a higher chance that it will misread, causing incorrect data to appear at the output. To determine the timing for enabling the sense amplifier, the discharge rates of the bitlines were examined under varying conditions. The worst-case scenario was used to determine the final timing for enabling the sense amplifier. By determining the timing in this manner, the amplifier should work reliably even with slight process variations under varying operating conditions.

Figure 4.7 shows the final analog testing results for the integration testing. For this simulation, two cells were written to and read from. This verifies that the write cycle is performed correctly, that the cell is storing the correct values, and that the sense amplifier is correctly displaying the stored value at the output.

The first line (brown) shows the least significant bit of the address bus, as all the other address lines are held low. This selects the wordline that will be enabled. For this simulation, two wordlines are used. The second line (green) shows the read/write enable signal. When the signal is low, the SRAM is being written to and when it is high, the SRAM is being read from. The third line (blue) shows the data input, which is the data that is being written to the cells. For this example, the other data inputs are held at a logic low, so only the least significant bit within the word is shown. By looking at the first three lines, it can be seen that a logic high is written to the first cell when address one is selected and a logic low is written to the first cell when address zero is selected. The fourth line (red) shows the signal that is used to enable the sense amplifier. During the write cycle, the sense amplifier is disabled, so the signal stays at a logic low. During the read cycle, the signal is brought high, enabling the sense amplifier. This allows the data to appear at the output, which can be seen on the fifth line (pink). The sense enable signal is brought back to a logic low for a short interval of time to allow the bitlines to be precharged. By looking at the output, it can be seen that the sense amplifier correctly reads the two values stored in the SRAM cells.

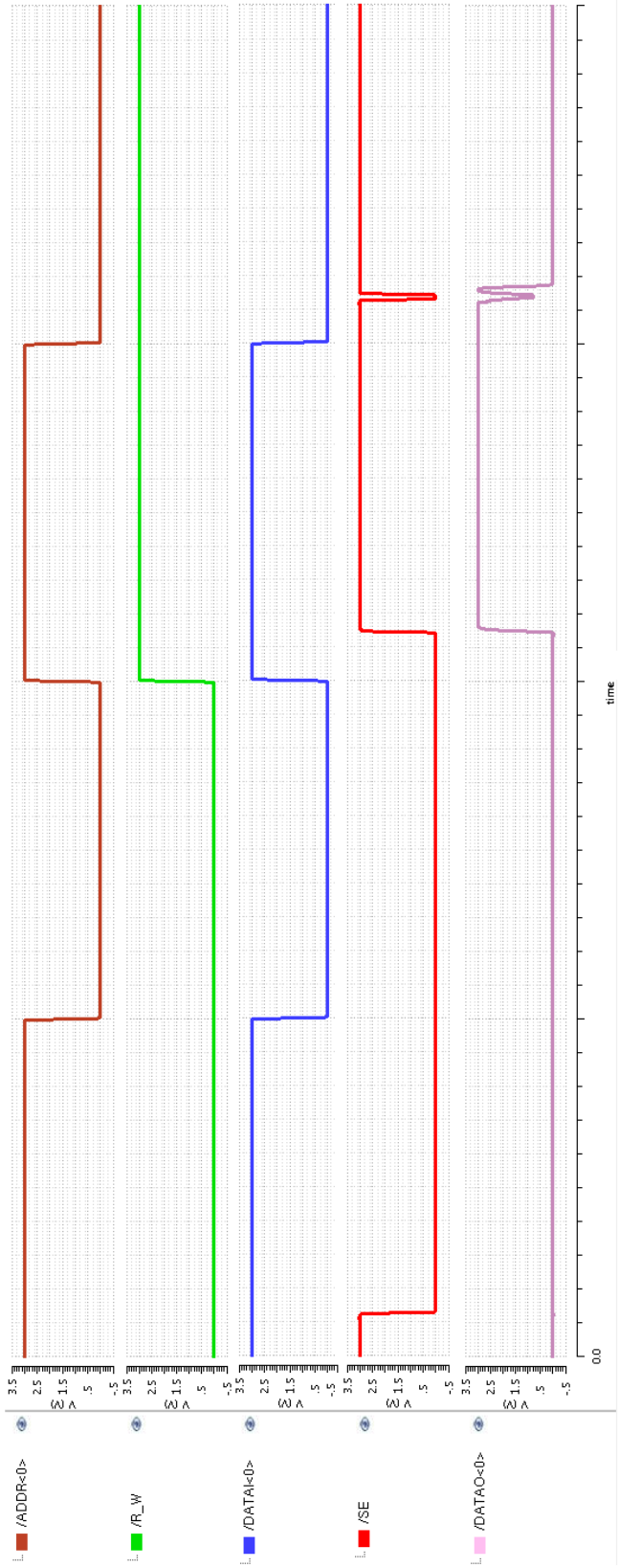


Figure 4.7: Integration Testing Results

4.2.3 System Testing

System testing is the final phase of testing before the chip is fabricated. At this point the analog functionality of the circuit has been tested and the functionality of the SRAM chip as a whole needs to be verified. Instead of using a voltage test-bench in Cadence, a Verilog script is used to execute a test sequence on the top-level schematic. The Verilog file is there to supply the test input signals and read the values that appear at the output. When this runs, an output file will be created with the address locations and the data that is read to and written from each address location. This allows a direct comparison to verify that the chip is accurately writing, storing, and reading values.

Figure 4.8 shows an example output from the system testing. In this example, the clock signal may be seen on the top line (labeled CLK). The read/write signal (labeled R_W) may be seen on the second line. The chip is being written to when this signal is at a logic low, and is being read from when the signal is at a logic high. The third line shows the chip enable signal (labeled CSN). When the signal goes high, the chip is disabled, and is not able to have values written to it, or have values read from it. In this state the chip should continue to store the values within the cells as long as the device is powered. When this signal is brought back to a logic low the SRAM should continue to function as normal during read and write cycles. the fourth line (labeled adr_s8) shows the address that is being selected. The fifth line (labeled din_s8) shows the data that is being written to the SRAM cells during a write cycle. The sixth line (labeled dout_s8) shows the data that is being read from the SRAM cells during the read cycle. By comparing the data that is written to the cells and the data that is read out of the SRAM cells, the functionality of the SRAM cell may be verified.

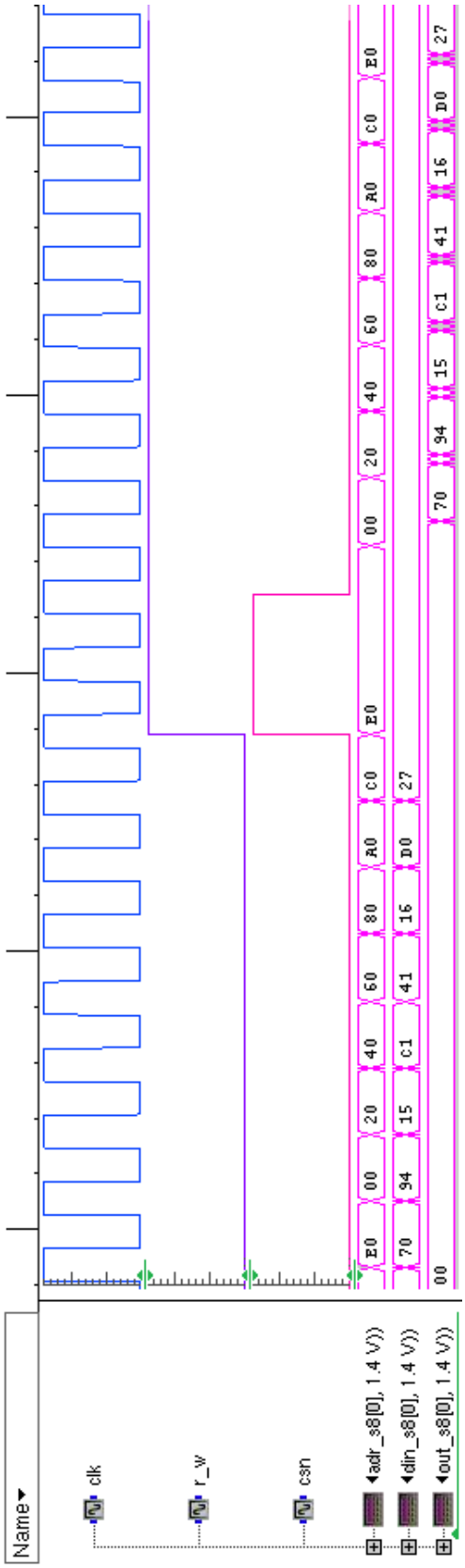


Figure 4.8: System Testing Results

4.3 Final Comparison

After completing the design and layout for the new sense amplifier and finishing all testing, the new design is compared to the preexisting sense amplifier. For these comparative tests, the same top-level design is used and only the sense amplifier is changed, allowing for a direct comparison of sense amplifier performance. The two main design goals of the new sense amplifier were to reduce the overall power consumption of the SRAM chip and to decrease the access time during a read cycle.

To test these two parameters, two identical simulations will be performed using Cadance Virtuoso Design Suite. For the tests, two read/write cycles will take place. First, a logic low will be written to the least significant bit of the first address location in the SRAM chip. All other data lines will be held at a logic low. This will be directly followed by a read cycle, where the logic low will be read at the output. Next, a logic high will be written to the same location. Again, all other data lines will be held at a logic low. This will be directly followed by a read cycle, where the logic high will be read at the output. After running this simulation, the access time will be measured for both the logic low and logic high. This will be measured from the 90% point on the address line to the 90% point on the output. To measure the power consumption, the average will be taken for the supply voltage multiplied by the supply current.

To compare the results, the output of both the new latch-based sense amplifier and the preexisting differential pair based sense amplifier are combined onto the same plot. Figure 4.9 shows the comparison for reading a logic high.

In this figure, the least significant bit of the address may be seen as the black line. When this signal goes high, the SRAM cells are enabled which creates a bias between the bitlines. The blue line is the sense amplifier enable signal. When this signal goes high, the amplifier is enabled and is able to read values out of the SRAM cells. The pink line is the output from the latch based sense amplifier, and the green line is the output from the differential sense amplifier. This figure clearly shows that the latch based design has a shorter access time. This means that the goal to improve the access time for the sense amplifier has been met.

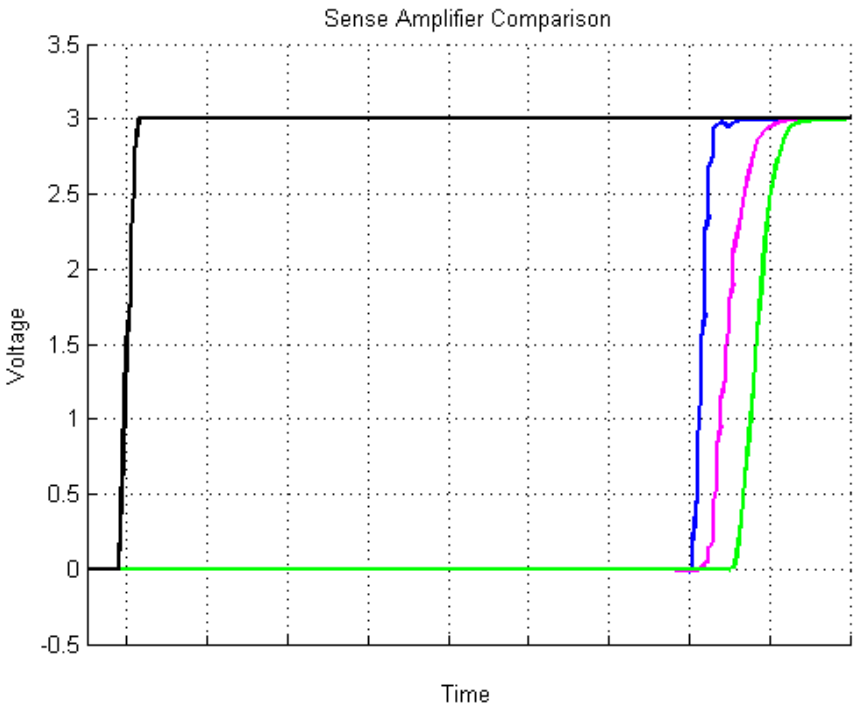


Figure 4.9: Access Time Reading a Logic High

Chapter 5

SRAM Conclusion

During this project, a new SRAM sense amplifier was designed, sized, simulated, and laid-out so that it can be fabricated. This new design is latch based which decreases the power consumption during the read cycle, as no quiescent current will flow once the amplifier has latched and read the value that is stored within an SRAM cell. The new sense amplifier also has a faster access time than the differential sense amplifier which was previously used in Allegro's SRAM chip.

After determining the amplifier requirements and choosing the most suitable design, the transistors were sized by running multiple parametric sizing analyses to minimize the access time and power consumption. The sense amplifier then went through multiple testing phases both by itself and as part of the top-level schematic to verify its functionality. The final simulations were completed using conditions that were as close to real life as possible by adding in parasitic capacitances that were extracted from the SRAM layout. The layout was completed for the top-level SRAM design and was checked against the schematic to verify that the two were identical. The layout was also checked to ensure that it met all of the design constraints for the CMOS process technology.

Although the SRAM test chip with the new sense amplifier design did not return from fabrication within the time frame of this project, a test chip will still fabricated. In the future, the physical SRAM chip design will be tested once it returns from fabrication. These tests will be used to compare the designs functionality to the simulation results and will check for faults and deviations. After any necessary changes are made, the sense amplifier design may be phased into production.

Chapter 6

LED Display

6.1 Overall Setup

The Cube size that we decided to go with is 15 LEDs long by 9 LEDs tall by 5 LEDs deep. This shape can be seen below:

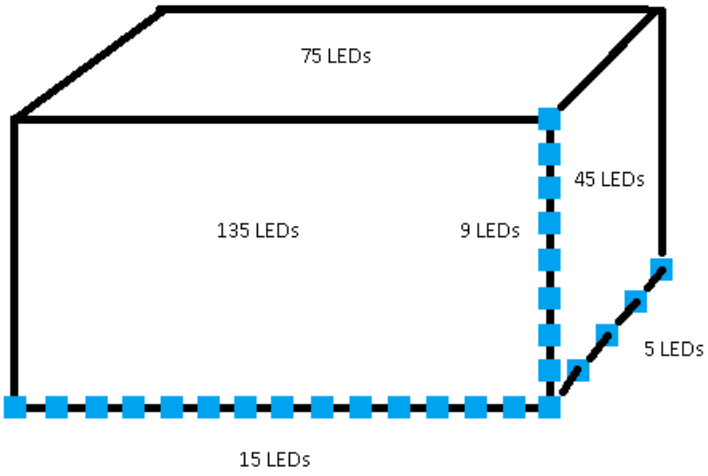


Figure 6.1: LED Display Setup

To fully access all of the LEDs in the cube we must connect the cathodes of the LEDS to the LED drivers and the anodes to the MOSFETS. There are a couple different ways that the leads of the LEDs can be connected. First off the cathodes of an entire column of LEDs will be connected,

a column will be considered the string of LEDs that are connected by the cathodes with the final cathode attached to a pin on the LED Drivers. A layer will be considered the LEDs that are connected by the anodes to a switching MOSFET. The layers and columns must cross in a way that will allow us to turn on only one LED when one column and one layer is on. The three ways that the layers and columns can be set up are shown below.

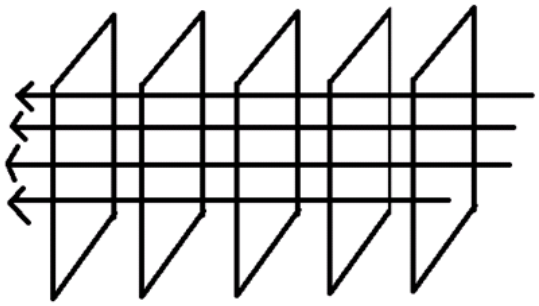


Figure 6.2: 15 Layers by 45 Columns

The Figure above shows vertical layers from side to side with horizontal LED columns. This option would have 45 led columns and 15 layers. 45 LED columns would require 3 LED drivers. Having 15 layers means that its possible that 15 LEDs might need to be turned on at the same time by the same LED driver pin. This option would require a minimum of 20 pins so this option is not as good of an option as the following ones are.

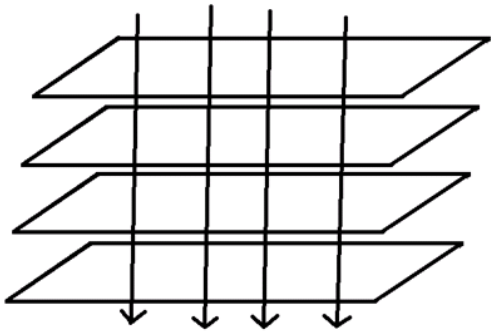


Figure 6.3: 9 Layers by 75 Columns

The figure above shows horizontal layers from top to bottom with vertical LED columns. This option requires 75 LED columns and 9 layers. 75 columns means that we would need 5 LED

drivers. Having 9 Layers also means that it might need to turn on 9 LEDs at a time by the same pin which still requires a lot of current. This option would require around 12 pins on the microcontroller which is substantially lower.

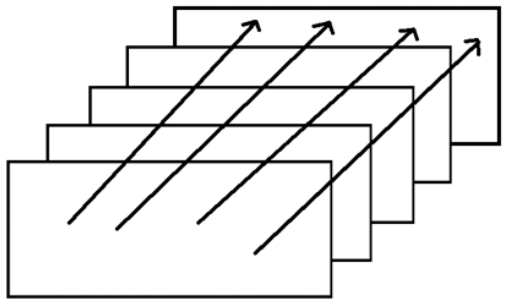


Figure 6.4: 5 Layers by 135 Columns

The figure above shows the third and final option. This option has vertical layers from front to back with horizontal columns from front to back as well. This option requires 135 LED columns which would be 9 LED drivers. It would also need 5 layers. Having only 5 layers means that only 5 LEDs might need to be turned on at the same time by the same driver pin which is less current needed than that of the second option and especially the first option. Also, this option will only need around 8 or 9 pins on the microcontroller

6.2 Flashing

All of the above schemes are possible to make work, so we cant really cut any of them out just yet. All of the systems above will need to incorporate some sort of flashing scheme, but they will be for different reason and at different speeds.

The first option with the 15 layers and 45 columns will need to flash at a fast speed. The reason that it will need to do this is because a letter or symbol cant just be drawn on to it. This is because we cant access specific LEDs without turning on some LEDs that we do not want to turn on. So in order to fix this LED flashing will need to be used. We will unfortunately have to continuously be switching the LED column and layer depending on what we want to do. If we want to move a letter across the cube we will need to turn on the LEDs that start the first part of the letter along with that layer and for the next position we will need to flash the layers and

columns so that an LED that we dont want to turn on wont turn on. The following images will show what needs to happen:

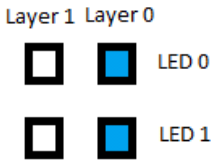


Figure 6.5: Drawing an L: Step 1

Right now we have LED column 0 and LED column 1 on with only layer 0 on. If we want the letter L if we just turn on layer 1 and layer 0 with LED columns 0 and 1on as well, we will get the following:

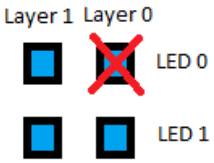


Figure 6.6: Drawing an L: Step 2

So now we realize that we need to flash the layers so that not all the LEDs are on, but with just flashing the layers we will still get an undesired LED to turn on:

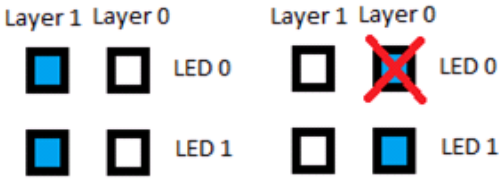


Figure 6.7: Drawing an L: Errors

We now realize that we need to flash which LED column is on at certain times. In the below image the steps taken to make sure that the right LEDs are on is shown.

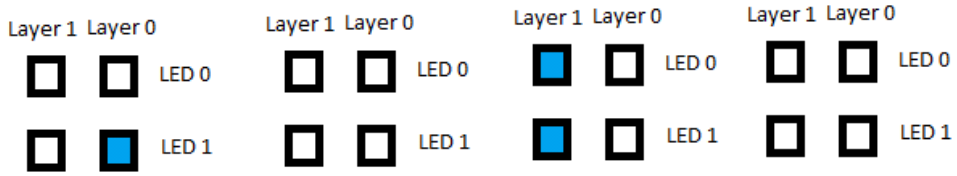


Figure 6.8: Drawing an L: Flashing

At first only layer 0 and LED column 1 are turned on, then they must be turned off, then layer 1 and both LED columns need to be turn on and then turn back off to get back to the original image. This is the same problem that the second option has as well. The big difference between the first two options and the third option is that the third option will allow us to pinpoint exactly the LED that we need to make up the letter, the only thing changing a layer will do is change the depth of the letter in the cube. This allows for extremely easy maneuvering of letters throughout the space.

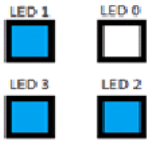


Figure 6.9: Drawing an L: Final

As shown above if we want to show an L all we have to do is send in the bit stream 1110b and turn on just one layer and that will appear. All we would have to do to make it change its depth to have a waving scroll throughout the space is change which layer is turned on. The best way to make this option work would be to set up a group of positions that a letter could possibly be in. that way you can show the Letter at any part of the space by just choosing its position and a layer.

6.3 Position and Layer Scheme

This positioning scheme can be seen on a bigger scale in the image below, where it shows the letter A scrolling through the same layer of the structure. The Letter can be turned on as a whole by selecting the LEDs that it needs by looking at position 0.

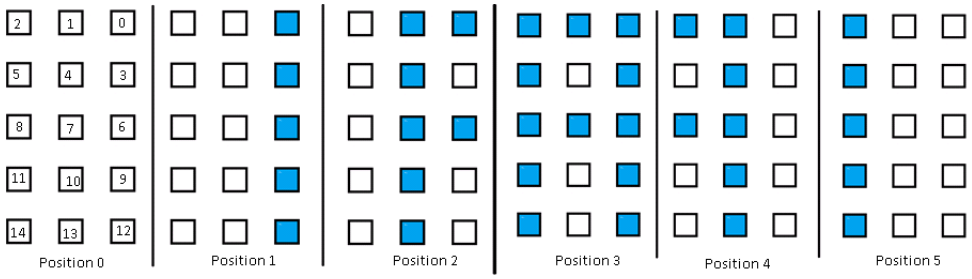


Figure 6.10: Scrolling Letter A

These can be turned on and switched by following the following LED column sequence:

- Position 0 No Columns
- Position 1 Columns 0, 3, 6 , 9, 12
- Position 2 Columns 0, 1, 4, 6, 7, 10, 13
- Position 3 Columns 0, 1, 2, 3, 5, 6, 7, 8, 9, 11, 12, 14
- Position 4 Columns 1, 2, 4, 7, 8, 10, 13
- Position 5 Columns 2, 5, 8, 11, 14

Next the layering scheme can be seen in the image below were part of the letter A can be seen and then by manipulating the layer and position number the LED displays the A somewhere completely different.

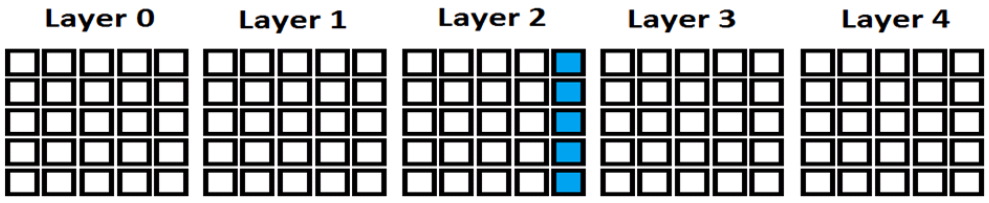


Figure 6.11: Switching Layers: Step 1

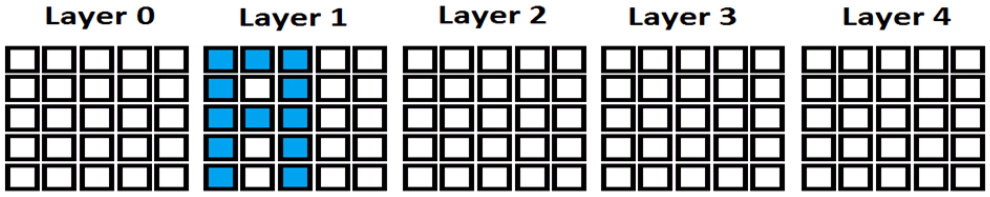


Figure 6.12: Switching Layers: Final

6.4 Construction of LED Structure

After careful review of the three previously stated options for LED configuration the second option is chosen. Although the addressing scheme for the last option is the most viable, the construction of the LED cube becomes a lot more difficult, requiring a lot more bending of the LED leads, Not only that, but also, either a lot of jumper wires or a PCB Board would need to be attached to the back. This wouldnt a lot for a free standing structure which could be seen from all sides. After these considerations the second option was chosen. This is done because the cathodes of the LED Cube could be attached into the same PCB board that had the drivers on it. Because the cathodes were facing downwards the PCB would be able to be hid underneath the structure allowing for a full viewing experience.

To begin construction of the LED structure all of the cathodes were bent vertically perpendicular to the anode, having the bend being very close to the LED itself. Next, the anode would be bent so that it would be horizontally perpendicular to the anode, after many of these bend were made soldering connections were done to connect nine cathodes together and then 15 anodes together. Once started, it became apparent that it would be very difficult to make all of these

soldering connections. Thinking carefully, an alternative method of soldering the 675 needed LEDs together was realized. Below the different stages of the process can be seen, starting with the front and side view of an untouched LED.

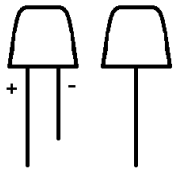


Figure 6.13: LED Front and Side View

The first step would be to bend the cathode in the same fashion as stated above, this can be seen in the image below.

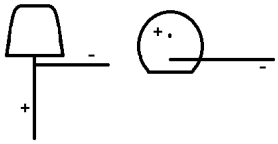


Figure 6.14: LED First Bend Side and Bottom View

Once many of these bends had been made, the soldering connections that would attach the cathodes could be made. Nine LEDs cathodes would be connect together as shown in the image below.

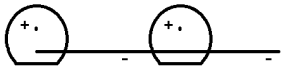


Figure 6.15: LED First Solder Bottom View

After many of these LED chains were constructed the next bend in the wires could be made. The next step was to bend the anodes over the connections made between the LEDs at their cathodes. This bend can be seen in a single LED below.

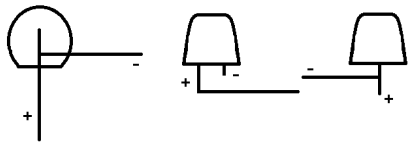


Figure 6.16: LED Second Bend Bottom and Side Views

Once all nine of the anodes for that chain had been made it would look similar to the image below but with nine LEDs instead of two.

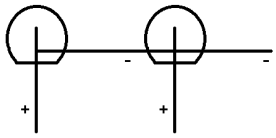


Figure 6.17: LED Second Bend with First Solder

Next, these chains of nine LEDs were bent and looking like the image above, the anodes could be soldered together. Fifteen, of these nine LED long chains would need to be soldered together. They would look like the image below.

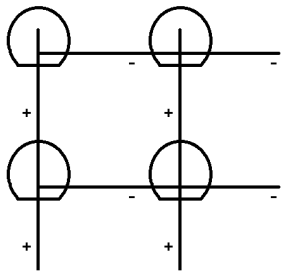


Figure 6.18: Final Soldering

6.5 Code Explanation

To code the microcontroller the program Code Composer Studio (CCS) was used. To begin the coding process all needed coding libraries had to be referenced, because the microcontroller was only being used for its digital inputs and outputs, the only library needed was msp430.h. next, a list of all the function prototypes is added so that the composer can correctly compile all of the text. To strt of the actual coding the main function is written. This is where the order of all the

functions can be found. It tells the compiler in what order and how to do specific parts of the task at hand.

When the first function, setsinkCode(), is entered the program begins to map the locations of the sink pins into an array. The next function that is entered is setTimer(), this is where the clock is configured so the system is running at the needed frequency. After, the setupSPI() function is entered this is where the serial peripheral interface is configured. Next, both the pins must be configured to be digital outputs. To do this the main function enters configPins() and then setLayers(), here using functions like PSEL, PDIR, and POUT, pins can be turned into outputs. To make specific pins on those ports outputs they must be first turned to 0 to become a digital IO with the PDIR function and then turned to a 1 in the PSEL function using hexadecimal values. In the setLayers() function layers are characterized by a certain hexadecimal value and the specific pin that they are on need to be made into a digital output as explained above.

Next, there are a series of functions which the programmer can now use to turn on specific LEDs. These functions include turnonLayer() and sendData(). turnonLayer() does exactly as its name says and will turn on a specific layer by taking in a character and using it as a case for each layer. The sendData() function takes in a hexadecimal value and will send it to the current sinks to turn on a specific column of LEDs. The combination of both turnonLayers() and sendData() allows for complete control of which LEDs will turn on.

Bibliography

[1] A. Pavlov and M. Sachdev, *CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies*. Springer, 2010.

[2] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 6th ed., ser. Oxford Series in Electrical and Computer Engineering. Oxford University Press, USA, 2009.

[3] T. L. Floyd, *Digital Fundamentals*, 8th ed. Pearson, 2003.

[4] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits, A Design Perspective*, 2nd ed., ser. Prentice Hall Electronics and VLSI Series, C. G. Sodini, Ed. Pearson, 2003.

[5] B. Wicht, *Current Sense Amplifiers for Embedded SRAM in High-Performance System-on-a-Chip Designs*, ser. Springer Series in Advanced Microelectronics. Springer, 2003, vol. 12.

[6] D. Hodges, H. Jackson, and R. Saleh, *Analysis and Design of Digital Integrated Circuits*. McGraw-Hill, 2003, ch. Semiconductor Memory Design, pp. 359–399.

[7] B. Mohammad, P. Dadabhoy, K. Lin, and P. Bassett, "Comparative Study of Current Mode and Voltage Mode Sense Amplifier Used for 28nm SRAM," in *Microelectronics (ICM), 2012 24th International Conference on*, Dec 2012, pp. 1–6.

[8] P. Mazumder and K. Chakraborty, *Testing and Testable Design of High-Density Random-Access Memories*. Kluwar Academic Publishers, 1996.

[9] J. Kim, N. Hardavellas, K. Mai, B. Falsafi, and J. C. Hoe, "Multi-bit Error Tolerant Caches Using Two-Dimensional Error Coding," in *ACM/IEEE International Symposium on Microarchitecture*, 2007.

[10] V. D. Agrawal. (2005) Lecture 14alt memory test. Accessed 2013-9-13. [Online]. Available: http://www.eng.auburn.edu/ agrawvd/COURSE/FULL/lec15.pp

[11] S. T. Inc., *Soft Test: The Fundamentals of Memory Testing.* Soft Test, Incorporated, 2004, ch. Chapter Two - SRAM, pp. 2.1– 2.22.

[12] S. Hamdioui, A. J. van de Goor, and M. Rodgers, "March SS: A Test for All Static Simple RAM Faults," in *Proc. IEEE International Workshop on Memory Technology, Design and Testing (MTDT 2002)*, 2002, pp. 95–100.

[13] R. Dekker, F. Beenker, and L. Thijssen, "A realistic fault model and test algorithms for static random access memories," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 9, no. 6, pp. 567–572, Jun 1990.

[14] S. Mourad and Y. Zorain, *Principles of Testing Electronic Systems.* John Wiley and Sons, 2000, ch. Memory Testing, pp. 279–302.

[15] Tektronix, "The XYZs of Logic Analyzers Primer," Tech. Rep., 2008.

[16] Agilent Technologies, inc., "Feeling Comfortable with Logic Analyzers: Application Note 1337," Tech. Rep., 2006.

[17] E. Seevinck, F. J. List, and J. Lohstroh, "Static noise margin analysis of MOS SRAM cells," *IEEE JSSC*, 1987.

[18] B. H. Calhoun and A. P. Chandrakasan, "Static Noise Margin Variation for Sub-threshold SRAM in 65-nm CMOS," *IEEE JSSC*, vol. 41, no. 7, 2006.

[19] E. Grossar, M. Stucchi, and K. Maex, "Read Stability and Write-Ability Analysis of SRAM Cells for Nanometer Technologies," *IEEE JSSC*, vol. 41, no. 11, 2006.

[20] Z. Guo, A. Carlson, L.-T. Pang, K. Duong, T.-J. K. Liu, and B. Nikolic, "Large-Scale Read/Write Margin Measurement in 45nm CMOS SRAM Arrays," in *IEEE Symp. VLSI Circuitse*, 2008.

[21] V. Champac, J. Castillejos, and J. Figueras, "IDDQ Testing of Opens in CMOS SRAMs," *Journal of Electronic Testing: Theory and Applications*, vol. 15, pp. 53–62, 1999.

[22] T. Kobayashi, K. Nogami, T. Shirotori, and Y. Fujimoto, "A current-controlled latch sense amplifier and a static power-saving input buffer for low-power architecture," *Solid-State Circuits, IEEE Journal of*, vol. 28, no. 4, pp. 523–527, Apr 1993.

[23] B. Wicht, T. Nirschl, and D. Schmitt-Landsiedel, "A yield-optimized latch-type SRAM sense amplifier," in *Solid-State Circuits Conference, 2003. ESSCIRC '03. Proceedings of the 29th European*, Sept 2003, pp. 409–412.

[24] P. Fischer. (2013) VLSI Design Lecture 6: Design Rules. Accessed 2014-3-3. [Online]. Available: http://sus.ziti.uni-heidelberg.de/Lehre/VLSIVorlesung1213/VLSI_Fischer_06_Designrules.pdf