Automation of Memory Scrambling Checker and Evaluation of Memory Characterization Flow Tools

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

In

Electronics & Communication Engineering

(Communication Engineering)

By

MANANKUMAR D. SUTHAR

(12MECC29)

Electronics & Communication Branch

Electrical Engineering Department

Institute of Technology

Nirma University, Ahmedabad-382 481

Automation of Memory Scrambling Checker and Evaluation of Memory Characterization Flow Tools

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

In

Electronics & Communication Engineering

(Communication Engineering)

By

MANANKUMAR D. SUTHAR

(12MECC29)

Under the Internal Guidance of

Prof. Bhupendra Fataniya

and

External Guidance of

Mr. Atul Bhargava

Electronics & Communication Branch

Electrical Engineering Department

Institute of Technology

Nirma University, Ahmedabad-382 481

May 2014

Declaration

This is to certify that

1. The thesis comprises of my original work towards the degree of Master of Technology in Communication Engineering at Nirma University and has not been submitted elsewhere for a degree.

2. Due acknowledgement has been made in the text to all other material used.

MANANKUMAR D. SUTHAR

CERTIFICATE

This is to certify that the Major Project entitled **"Automation of Memory Scrambling Checker and Evaluation of Memory Characterization Flow Tools"** submitted by **Mr. MANANKUMAR D. SUTHAR (12MECC29)**, towards the partial fulfillment of the requirements for the degree of **Master of Technology in Communication Engineering** of **Nirma University of Science and Technology, Ahmedabad** is the record of work carried out by him under our supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for the examination. The results embodied in this major project, to the best of our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Bhupendra Fataniya	**Dr. D. K. Kothari**
Internal Guide	**Program Co-ordinator**
Dr.P.N. Tekwani	**Dr K Kotecha**
Head of EE Department	**Director, IT**

Acknowledgement

Accomplishment of any project requires the hard work and efforts of many people. This project work would not have been possible without the kind support and help of many individuals and organizations.

I would like to express my sincere gratitude to **Mr. Bhupendra Fataniya**, internal guide from Nirma University. I wish to express my deep sense of gratitude to my project guide **Mr. Atul Bhargava**, for his invaluable guidance, encouragement, useful suggestions and readiness to resolve any point of confusion by mutual discussion. I would take this opportunity to thank **Dr. P. N. Tekwani** (Head of Department, Electrical Engineering), and **Dr. Dilip Kothari** (Professor and Program Coordinator, M.Tech - EC (Communication Engineering)) and **Dr. Ketan Kotecha** (Director, Nirma University, Ahmedabad) for allowing me to undertake this project and for their continuous guidance, support and enthusiasm. I would also like to thank all the faculties at **Nirma University (Communication Engineering)**, for their vision and relentless effort, support, and encouragement to provide me with this excellent opportunity to carry out my project work in such a highly renowned and esteemed organization, **STMicroelectronics**. I am equally thankful to **STMicroelectronics** for providing me the invaluable exposure to the industry and the current market trends.

I would also like to express my sincere gratitude to **Ms. Radhika Gupta** and **Ms. Monika Rawat** for all their valuable guidance during the project. My experience at STMicroelectronics would be incomplete without the mention of my colleagues and team members who were there to attend to my queries/ issues cordially and enthusiastically.

Finally, I would like to express my heartfelt thanks to my parents for their blessings and for their constant love and support.

- MANANKUMAR D. SUTHAR

Abstract

Semiconductor Memories are circuits that store digital information in large quantity, hence are vital subsystem in modern integrated circuits. The ever-increasing demand for low priced memories with low power consumption, high speed operation, high density and small package size has driven the fabrication technology and memory development towards more compact design rules and consequently towards higher data storage densities. The objective is to study the architecture, working and characterization of SRAM memory and also to automate the memory scrambling checker.

Memory Scrambling Checker (MSC) calculates the scrambling equations of memory. There is difference between logical and actual locations of memory cells in memory. So to validate both addresses, MSC is used. It gives the coordinates of particular memory cell in output file. MSC was having some bugs while running at higher word, bit combinations. So bug fixing was required.

Characterization of memory means to get information about its behaviour in terms of different timing (access time, setup time, hold time etc), power (dynamic, static and leakage) and capacitances. This help in evaluating the performance of memory and improves upon the design. The tool used for characterization is Memory characterization flow (MCF). Also the vendor has proposed a new tool, so ST is checking its reliability against its own tool to reduce the efforts of its engineers.

STMicroelectronics At A Glance

- A world leader in providing the semiconductor solutions that help our customers improve quality of life for everyone, both today and in the future
- Among the world's largest semiconductor companies
- A leading Integrated Device Manufacturer serving all electronics segments
- A leading technology innovator (around 12,000 researchers approx. 21,500 patents)
- Key strengths in Multimedia Convergence, Power Applications and Sensors
- Rich, balanced portfolio (ASICs, Application-Specific Standard Products and Multi-Segment Products)
- A pioneer and visionary leader in sustainability
- President and CEO: Carlo Bozotti
- 2011 revenue $9.73 billion , Q3 2011 revenue $2.44 billion ,2011 revenue $7.54 billion
- Approximately 50,000 employees including STEricssion at December 31, 2011
- Advanced research and development centers in 10 countries
- 12 main manufacturing sites

- Corporate Headquarters Geneva, Switzerland
- Global presence with sales offices all around the world
- Public since 1994 - shares traded on New York Stock Exchange (NYSE: STM), Euronext Paris, and Borsa Italiana
- Created as SGS-THOMSON Microelectronics in June 1987, from merger of SGS Microelettronica (Italy) and Thomson Semiconducteurs (France)
- Renamed STMicroelectronics in May 1998

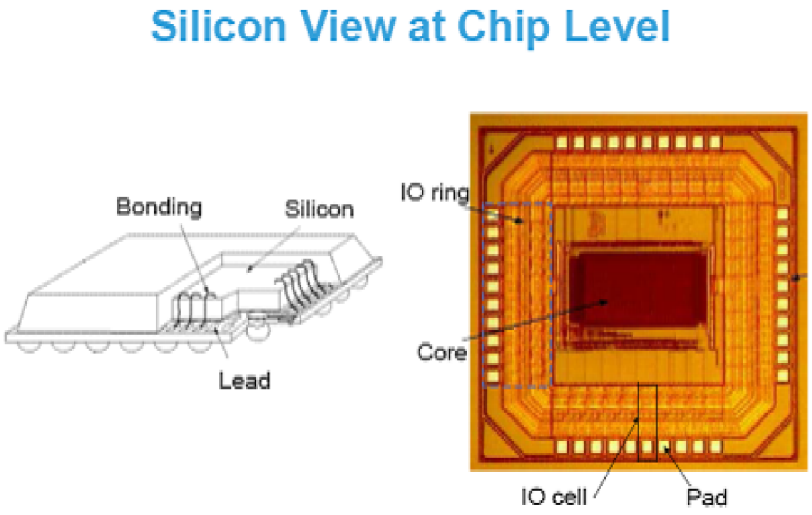
Group Introduction

The TR&D is a group in ST which works on integrated products (IPs). Its work is to provide quality Library Solutions and services to divisions, in the company's drive in time to market, IP Reuse and Super Integration. There are various types of IPs:

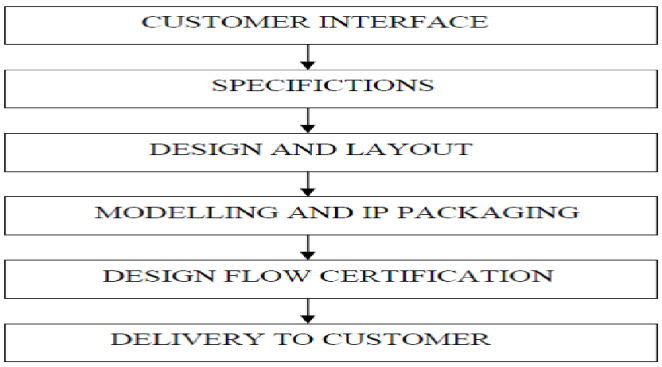
- Standard Cells : It is the layout of basic sequential and combinational circuits which are used to make bigger circuits. It is the basic building block of all the circuits. It is a component within an integrated circuit with known functional and timing characteristics, which can be used as an element in building a larger circuit. Example of standard cells include Inverter, Flip-flop (FD), Latch (LD), Multiplexer (MUX), NAND, NOR, AND, OR, XOR, XNOR, Full Adder (FA), Half Adder (HA).

- Memories : A device or an electrical circuit used to store a single bit (0 or 1) is called a memory cell. Examples of memory cell are flip flop, a charged capacitor etc. Semiconductor memories are capable of storing large amount of Digital information. The amount of memory required in a particular system depends on the type of application but the no. of transistors required for storage of data are always much larger than the no of transistors used for logic operations and other purposes. The ever increasing demand of high storage capacity has driven the fabrication technology and memory development towards more compact design rules and consequently towards higher data storage densities.

- I/Os (input/output cells) : I/O's are the input/output cells, placed on the periphery of the chip. Any I/P signal which comes from off-chip environment (external voltages are at a typical voltage Vdde level of 2.5V, 3.3V or 5V) is connected to the core (logic part of the chip) is connected through them and vice versa.



- Developement Flow : The customer, as per his requirements, gives the specifications to the flow team. These requirements are analyzed and as the needs, the features required for the design are noted. These are sent to the Back End Team for the designing. After the design is ready, it is checked for the specifications and then sent to the Front End for the characterization and functional modeling. ıtem Concept Of Library : A library is collection of cells all of



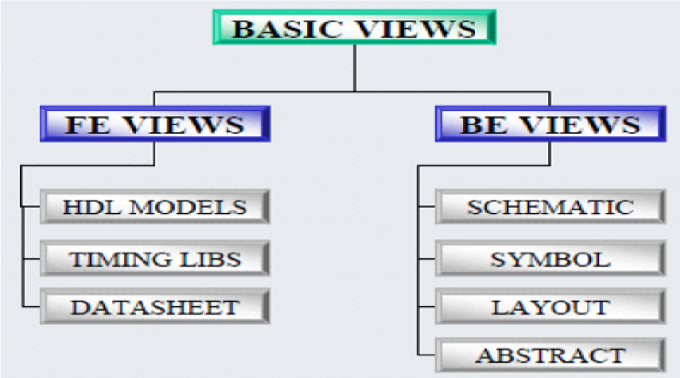
which reference the same technology file. The cells are the smallest functional unit. Attached to every library is a technology file. A technology file is a large data file that specifies, in one central location, all of the technology-dependent parameters associated with that particular library. The purpose of library is

to reduce the design cycle of the designer.

- Basic building blocks: Digital (AND, OR); Analog (amplifier).
- Complex building block: Digital (Microcontroller); Analog (PLL, DAC).
- Memories: Single or dual port SRAM, ROM etc.

The cells are delivered as a set of different views. These views are used by different tools in a given Design flow. These cells are used until the physical implementation.

- Cell Views : A particular representation of a cell is referred to as a view. Each cell may have a layout view, schematic view, symbolic view, timing view, etc. Each view object has attached property objects that are specific to the view, such as grid units, scale and display of axes etc. A cell is delivered as a set of view and each view is used by a different tool in a given design flow.



- BE (Back End) VIEWS: Views Related to the Physical Design of a Cell
- FE (Front End) VIEWS: Views Related to the Timings/Modeling of the Cell the various views can also be classified as Primary and Secondary views.

Contents

List of Figures

Chapter 1

Introduction

The electronics industry has achieved a phenomenal growth over the last few decades, mainly due to the rapid advances in integration technologies and large-scale systems design. The use of integrated circuits in high-performance computing, telecommunication and consumer electronics has been growing a very fast pace. The increasing need for high computational and information processing power of these application is the driving force for fast development of this field. Complementary metal oxide semiconductor (CMOS) digital integrated circuits are the enabling technology for the modern information age because of their intrinsic features in low-power consumption, large noise margins, and ease of design. CMOS integrated circuits have been widely used to develop random access memory (RAM) chips, microprocessor chips, digital signal processor (DSP) chips, and application-specific integrated circuits (ASIC) chips.

For VLSI circuit design, however, it is important that the design be done in the context of global optimization(in terms of area, power, speed) with proper boundary conditions. In fact, the beauty of integrated circuits is that the final design goal is the concerted performance of all interconnected transistors, and not of individual transistors. No matter how well an individual transistor performs, if the technology fails to have equally good interconnects, the total performance can be very poor due to large parasitic capacitances and resistances; these translate into a large delay in

the interconnection lines between transistors or logic gates.

Computer aided design (CAD) tools are used for design automation and optimization. Computer simulation is, and will continue to be, an essential part of the design process, both for performance verification and for fine-tuning of circuits. However, the emphasis on simulation must be well-balanced with the emphasis on hands-on-design and analytical estimates, so that the significance of the latter is not overwhelmed by the extensive use of computer-aided techniques. In addition to the transistor-level circuit design issues, the accurate prediction and reduction of interconnect parasitics has become a very significant topic in high performance digital integrated circuits, especially for deep sub-micron technologies. Digital systems require the capability of storing and retrieving large amounts of information at high speeds. Memories are circuits or systems that store digital information in large quantity, hence are vital components in modern integrated circuits. In present scenario demand is for low-priced memories with low-power consumption, high-speed operation, high density, and small package size. The semiconductor markets have embraced the fact that the architecture of the memory structure has a considerable impact on the performance of the system. Over the years, technology advances have been driven by memory designs of higher and higher density.

1.1 Motivation

A device or an electrical circuit used to store a single bit (0 or 1) is called a memory cell. Semiconductor memories are capable of storing large amount of digital information. The amount of memory required in a particular system depends on the type of application. The no. of transistor required for storage of data are always much larger than the number of transistors used for logic operations and other purposes. The ever increasing demand of high storage capacity has driven the fabrication technology and memory development towards more compact design rules and consequently towards higher data storage densities. The memory data

storage capacity of a single chip doubles almost after every two years. The number of data bits stored per unit area is one of the key criteria that determine the overall storage capacity and hence the memory cost per bit. Another important issue is the memory access time i.e. the time taken to store or retrieve data in the memory array. The access time determines the memory speed. Static and Dynamic Power consumption of the memory array is a significant factor to be considered in the design because of the increasing demand of the low power applications.Also another important point to be considered is the physical and logical locations of the cells in the memory array. This helps in various electric rule checks. During my training my prime focus was on SRAM memories.

Memory Scrambling means that the logical structure of a memory array is different from its physical internal structure. The tool used to map these physical and logical locations of the memcell is Memory Scrambling Checker (MSC). It is an inhouse tool of STM. When a set of inputs(required word-bits) is given to it , MSC provides the logical and physical addresses for these inputs. Each memcell is toggled consequently and addresses are obtained internally. In case of large memory block instead of mapping the entire core, one can simply go for some standard locations provided internally in MSC for mapping addresses. Also one more important requirement for shrinking technologies was xa simulation which is faster than nanosim simulation(done previously).

Memory characterization means to obtain the information about the behaviour of memory in terms of different timings, power and capacitances. When a set of specified inputs is applied to the memory, it includes running simulation and then doing measurements from the simulated values .This help in evaluating the performance of memcell and improves upon the design. The tool used for characterization is Memory characterization flow (MCF).This is an inhouse tool of STM. In case of a large memory block where number of cuts at different pvt's is to be characterized , then it is very difficult to characterize the whole cut as in case of a single inverter. So for that purpose we need a frame work which governs all the activities and fulfil

the requirement automatically. This framework is called MCF.

1.2 Objective of Work

The objective of the project work was to design and develop MSC and migrate it to XA simulator from nanosim , study the architecture, working and characterization of SRAM memory . The work dealt with the addition of new modes in MSC setup i.e. various formats in which a stimuli can be provided to the msc to obtain various word-bit combinations and three such modes were added.Also MSC was migrated to XA as new and upcoming technologies were simulated with XA. The present MSC setup gives results in less duration as compared to previous setup and also saves on the manual time involved in writing the stimuli file as its being generated internally now. Even the memcell being accessed are refreshed after obtaining the address locations. Another aspect of the work was the changes done in the setup of Characterization tool. During the training period, changes were done in the previous task as well as new tasks were added in the characterisation setup. The present MCF setup gives simulated results in less duration as compared to the previous setup. This allows customer to get any memory cut's datasheet having cut's different timings, power, and leakage and dimensional (height, width and areas), value with all pins diagrams and timing diagrams in very short duration.

1.3 Organization of Report

In Chapter 2 different types of storage memories have been discussed. The prime focus of discussion in this chapter is SRAM. 6T-SRAM memory cell is explained in detail in terms of design, read and write operation along with read and write circuitry. SRAM architecture is also explained in terms of main subcomponents of the chip. Different types of SRAM architecture has been analysed in terms of area, power and speed. Also the design foundation (from schematic to layout) is included.

In Chapter 3 basics of Memory Scrambling , structure of Memory Scrambling Checker and results of kind of work done are shown.

In Chapter 4 basics of Memory Characterization , structure of Memory Characterization Flow and results of kind of work done are shown.

In Chapter 5 basics of other work regarding to script in shell and perl is described.

Chapter 2

Background and Related Work

The data storage devices can be classified by a wide variety of aspects, but most frequently, they are divided by technology into the semiconductor types and the moving media types.

2.1 Data storage devices : An Overview

The five basic semiconductor types are bipolar, N-channel and P-channel MOS, complementary MOS and charge coupled devices (CCD)[1].

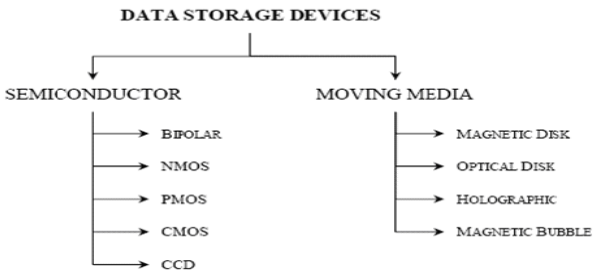


Figure 2.1: Classification of data storage devices

The moving media types include magnetic disk, optical disk and holographic storage. While magnetic bubbles are not mechanical, they require equipment for supplying a magnetic field for operation and are thus considered with the mechanical

types. Since the semiconductor memories are decreasing in cost per bit faster than the other types of data storage, various attempts to configure them for the disk application are occurring.

2.2 Introduction to Memories

The ideal memory would be low cost, high performance, high density, with low power dissipation, random access, non-volatile, easy to test, highly reliable, and standardized throughout the industry. Those memory technologies, which did not offer these advantages to some extent, were one by one successfully challenged by the MOS memories. Unfortunately a single memory having all these characteristics has not yet been developed, although each of the characteristics is held by one or another of the MOS memories. Thus, MOS memories have dominated the world of memories.

The Semiconductor memories are classified according to the type of data storage and data access.

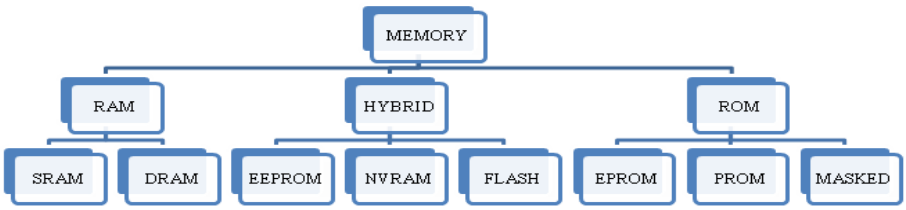


Figure 2.2: Classification of data storage devices

2.2.1 Classification of Memories

1. **ROM**

These are read only memory. ROM is a non volatile memory that is, the stored data is not lost even when the power supply is not switched off and refresh operation is not required. Memories in the ROM family are distinguished by the methods used to write new data to them (usually called programming), and the number of times they can be rewritten. The ROM is classified into three main categories:

- **PROM**

  These are programmable ROM in which data is written electrically after the chip is fabricated. The device programmer writes data to the device one word at a time by applying an electrical charge to the input pins of the chip. Once a PROM has been programmed in this way, its contents can never be changed. If the code or data stored in the PROM must be changed, the current device must be discarded. As a result, PROMs are also known as one-time programmable (OTP) devices[1].

- **EPROM**

  These are programmable ROM in which data is written electrically after the chip is fabricated. The device programmer writes data to the device one word at a time by applying an electrical charge to the input pins of the chip. Once a PROM has been programmed in this way, its contents can never be changed. If the code or data stored in the PROM must be changed, the current device must be discarded. As a result, PROMs are also known as one-time programmable (OTP) devices.

2. **HYBRID**

As memory technology has matured in recent years, the line between RAM and ROM has blurred. Now, several types of memory combine features of both. These devices do not belong to either group and can be collectively referred to as hybrid memory devices. Hybrid memories can be read and written as

desired, like RAM, but maintain their contents without electrical power, just like ROM. Two of the hybrid devices, EEPROM and flash, are descendants of ROM devices. These are typically used to store code. The third hybrid, NVRAM, is a modified version of SRAM. NVRAM holds persistent data.

- **EEPROM**

  EEPROMs are electrically-erasable-and-programmable. Internally, they are similar to EPROMs, but the erase operation is accomplished electrically, rather than by exposure to ultraviolet light. Any byte within an EEPROM may be erased and rewritten. Once written, the new data will remain in the device forever–or at least until it is electrically erased. The primary tradeoff for this improved functionality is higher cost and longer writes cycles than RAM.

- **FLASH RAM**

  Flash memory combines the best features of the memory devices described thus far. Flash memory devices are high density, low cost, nonvolatile, fast (to read, but not to write), and electrically reprogrammable. These advantages are overwhelming and, as a direct result, the use of flash memory has increased dramatically in embedded systems. From a software viewpoint, flash and EEPROM technologies are very similar. The major difference is that flash devices can only be erased one sector at a time, not byte-by-byte. Typical sector sizes are in the range 256 bytes to 16KB. Despite this disadvantage, flash is much more popular than EEPROM and is rapidly displacing many of the ROM devices as well.

3. **RAM**

The Read/Write memory is commonly known as Random Access Memory (RAM). Read/Write(R/W) memory must permit the modification (writing) of data bits stored in the memory array, as well as their retrieval (reading)

on demand. .Unlike sequential access memory any cell can be accessed with nearly equal access time. The stored data is volatile i.e. the stored data is lost when the power supply is switched off.

RAMs are classified into two main categories.

- **DRAM**

  Dynamic random-access memory (DRAM) is a type of random-access memory that stores each bit of data in a separate capacitor within an integrated circuit. The capacitor can be either charged or discharged; these two states are taken to represent the two values of a bit, conventionally called 0 and 1. Since capacitors leak charge, the information eventually fades unless the capacitor charge is refreshed periodically. Because of this refresh requirement, it is a dynamic memory as opposed to SRAM and other static memory. The advantage of DRAM is its structural simplicity: only one transistor and a capacitor are required per bit, compared to four or six transistors in SRAM. This allows DRAM to reach very high densities. Unlike flash memory, DRAM is volatile memory (cf. non-volatile memory), since it loses its data quickly when power is removed. The transistors and capacitors used are extremely small; billions can fit on a single memory chip.

- **SRAM**

  Static random access memory (SRAM) can retain its stored information as long as power is supplied. This is in contrast to dynamic RAM (DRAM) where periodic refreshes are necessary or non-volatile memory where no power needs to be supplied for data retention, as for example flash memory. The term “random access” means that in an array of SRAM cells each cell can be read or written in any order, no matter which cell was last accessed. It is used when we require high speed because SRAM devices offer extremely fast access times (approximately

four times faster than DRAM) but are much more expensive to produce. Generally, SRAM is used only where access speed is extremely important.

2.3 SRAM Cell Description

The basic static RAM cell is consists of two cross-coupled inverters and two access transistors. The access transistors are connected to the wordline at their respective gate terminals, and the bitlines at their source/drain terminals[1].

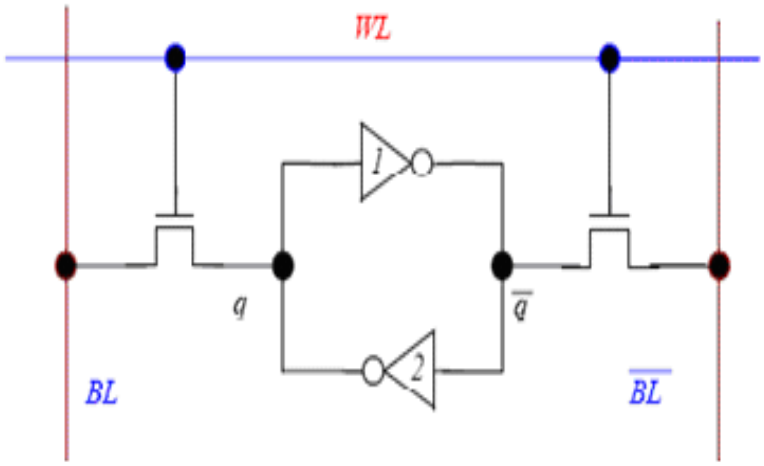


Figure 2.3: Static Memcell

The wordline is used to select the cell while the bitlines are used to perform read or write operations on the cell. Internally, the cell holds the stored value on one side and its complement on the other side. For reference purposes, assume that node q holds the stored value while node q' holds its complement. The two complementary bitlines are used to improve speed and noise rejection properties.The wordline is used to select the cell while the bitlines are used to perform read or write operations on the cell. Internally, the cell holds the stored value on one side and its complement on the other side. For reference purposes, assume that node q holds the stored value while node q bar holds its complement. The two complementary bitlines are used

to improve speed and noise rejection properties.

2.3.1 Voltage Transfer Characteristics

The Voltage Transfer Characteristics (VTC) conveys the key cell design considerations for read and writes operation. In the cross-coupled configuration, the stored values are represented by the two stable states in the VTC[1].

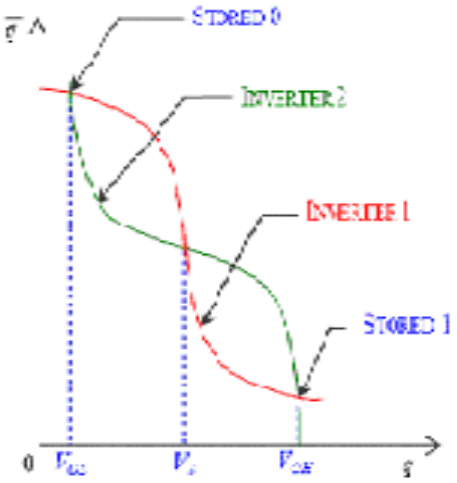


Figure 2.4: Voltage Transfer Characteristics

The cell will retain its current state until one of the internal nodes crosses the switching threshold, VS. When this occurs, the cell will flip its internal state. Therefore, during a read operation, its current state must not be disturbed, while during the write operation the internal voltage is forced to swing past VS to change the state.

2.3.2 6T SRAM Cell

The six transistor (6T) static memory cell in CMOS technology is used in majority of the designs, today. The cross-coupled inverters, M1, M5 and M2, M6, act

as the storage element. Major design effort is directed at minimizing the cell area and power consumption so that millions of cells can be placed on a chip.

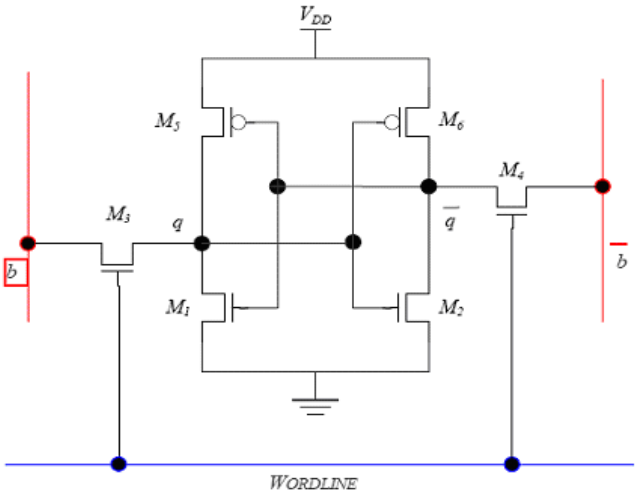


Figure 2.5: 6T Memcell

The steady-state power consumption of the cell is controlled by sub-threshold leakage currents, so a larger threshold voltage is often used in memory circuits.

2.3.3 Read operation

For a ”0” is stored on the left side of the cell, and a ”1” on the right side in the 6T RAM cell, M1 is on and M2 is off. Initially, b and b’ are precharged to a high voltage around VDD by a pair of column pull-up transistors. The row selection line, held low in the standby state, is raised to VDD which turns on access transistors M3 and M4. Current begins to flow through M3 and M1 to ground. The resulting cell current slowly discharges the capacitance Cbit. Meanwhile, on the other side of the cell, the voltage on b ? remains high since there is no path to ground through M2.The difference between b and b’ is fed to a sense amplifier to generate a valid low output, which is then stored in a data buffer[1].

Upon completion of the read cycle, the wordline is returned to zero and the

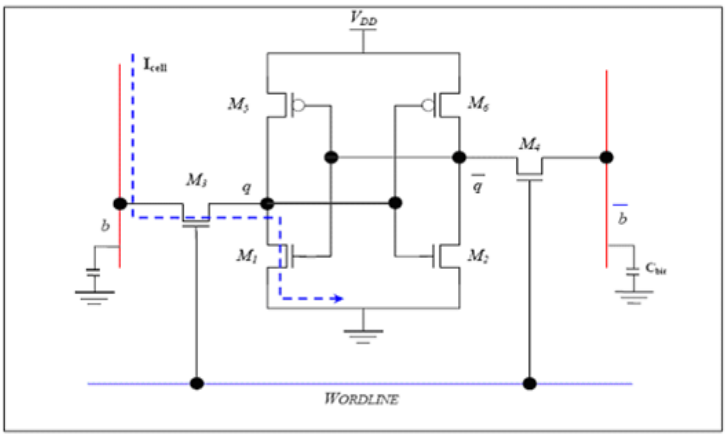


Figure 2.6: Read operation of Memcell

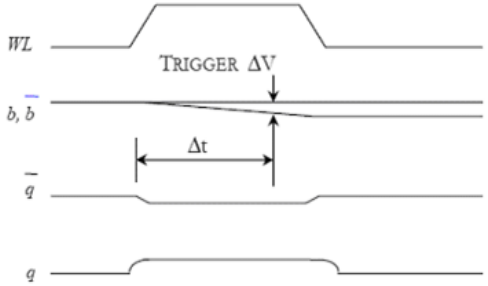


Figure 2.7: Waveform of Read operation

column lines (bit lines) can be precharged back to a high value. When designing the transistor sizes for read stability, it is ensured that the stored values are not disturbed during the read cycle. The problem is that, as current flows through M3 and M1, it raises the output voltage at node q which could turn on M2 and bring down the voltage at node q ?. The voltage at node q bar may drop a little but it should not fall below VS. To avoid altering the state of the cell when reading, the voltage at node q is controlled by sizing M1 and M3 appropriately. This is accomplished by making the conductance of M1 about 3 to 4 times that of M3 so

that the drain voltage of M1 does not rise above VTN. In theory, the voltage should not exceed VS, but this design must be carried out with due consideration of process variations and noise. In effect, the read stability requirement establishes the ratio between the two devices.

2.3.4 Write operation

The operation of writing 0 or 1 is accomplished by forcing one bitline, either b or b, low while the other bitline remains at about VDD. For SRAM cell taken above, to write 1, b is forced low, and to write 0, b bar is forced low.

The cell must be designed such that the conductance of M4 is several times larger than M6 so that the drain of M2 is pulled below VS. This initiates a regenerative effect between the two inverters. Eventually, M1 turns off and its drain voltage rises to VDD due to the pull up action of M5 and M3. At the same time, M2 turns on and assists M4 in pulling output q ? to its intended low value. When the cell finally flips to the new state, the row line can be returned to its low standby level.

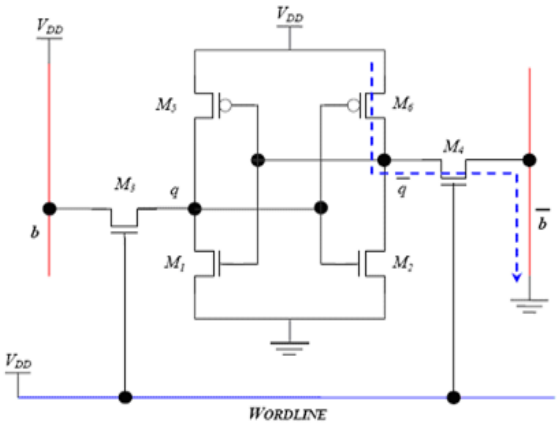


Figure 2.8: Write operation of waveform

The design of the SRAM cell for a proper write operation involves the transistor pair M6 M4. When the cell is first turned on for the write operation, they form a pseudo- NMOS inverter. Current flows through the two devices and lowers the

voltage at node q bar from its starting value of VDD. The design of device sizes is based on pulling node q bar below VS to force the cell to switch via the regenerative action.

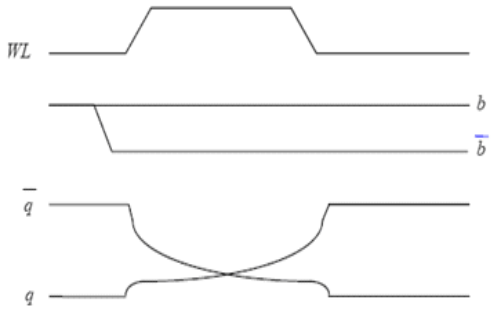


Figure 2.9: Waveform of Write operation

2.4 Timing Diagram

Timing diagrams specify the minimum required and maximum expected timing requirements for system actions. The two sets of timing symbols are self-explanatory, one being the standard for timing symbols and the other older one in widespread usage. The operation of the SRAM starts with the detection of an address change in the address register. An address change activates the SRAM circuits, the internal timing circuit generates the control clocks, and the decoders select a single memory cell.

At write, the memory cell receives a new datum from the data input buffers; at read, the sense amplifier detects and amplifies the cell signal and transfers the datum to the output buffer. Data input/output and write/read are controlled by output enable OE and write enable WE signals. A chip enable signal CE allows for convenient applications in clocked systems.

In some systems, power consumption may be saved by the use of the power down

signal PD. The power down circuit controls the transition between the active and standby modes. In active mode, the entire SRAM is powered by the full supply voltage; in standby mode, only the memory cells get a reduced supply voltage. In some designs, the memory-internal timing circuit remains powered and operational also during power down.

2.4.1 Read cycle diagram

It is important to understand the characteristics of the memories, to understand them better. The best indicator of these characteristics is the data sheet specification for the particular memory.

In a memory system, there are signals going from the processor via the bus into the inputs of the memory and signals coming from the outputs of the memory onto the bus and to the processor.

Inputs from the system processor to the memory include:

- Addresses, which indicate the memory locations selected.
- Write enable, which chooses between read and write mode and controls writing of new information into the memory.
- Chip select(s), which select one memory out of several in a system. If a chip select is off, the memory is deselected.
- Output enable, which can be used to control the output buffer.
- Data input(s), to be written into the memory.

Outputs from the memory include:

- Data output(s) being read from the memory.

Some memories such as SRAM with wider data path can have a common pin for input and output.

A timing diagram for a basic read cycle during which the system reads out information that is stored in a static RAM (in fig. 05) for a wide bus, common I/O SRAM with chip enable, and output enable functions. It consists of

- System selects the RAM by turning the chip select on ((CS) bar low).
- System sets the correct addresses (A set).
- System turns the output enable on ((OE) bar low).

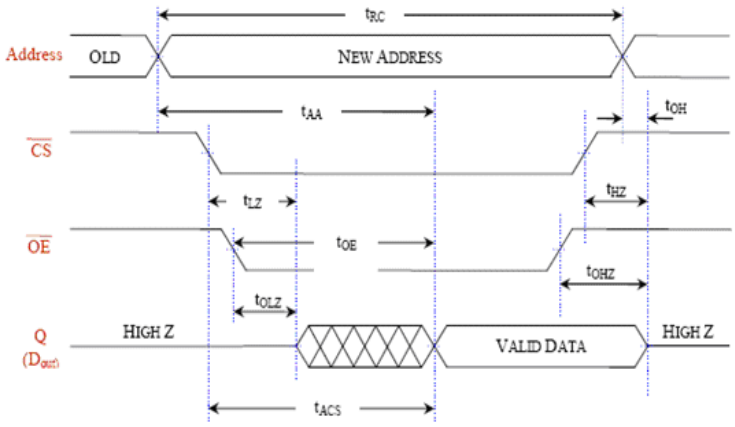


Figure 2.10: Timing diagram of Read cycle

- System must make sure that the time that old data from other sources is still on the common I/O bus is less than the minimum of output enable low to output active t_{OLZ} or chip enable low to output active t_{LZ}.
- The system must wait a minimum time of address access time t_{AA} in order to be sure of correct data.

The expansions of the acronyms for read timing parameters are

t_{RC} : Read Cycle Time

t_{AA} : Address Access Time

t_{ACS} : Chip Enable Access Time

t_{OE} : Output Enable Access Time

t_{OH} : Output Hold from Address Change

t_{LZ} : Chip Enable Low to Output Enable

t_{OLZ} : Output Enable Low to Output Enable

t_{HZ} : Chip Enable High to Output Enable - Z

t_{OHZ} : Output Enable High to Output Enable - Z

After the maximum required wait time (which is the minimum time the system must wait) the system may read the information stored in the memory.

2.4.2 Basics of Write cycle Diagram

A simple write cycle for changing data in an SRAM (writing into it) is shown below.

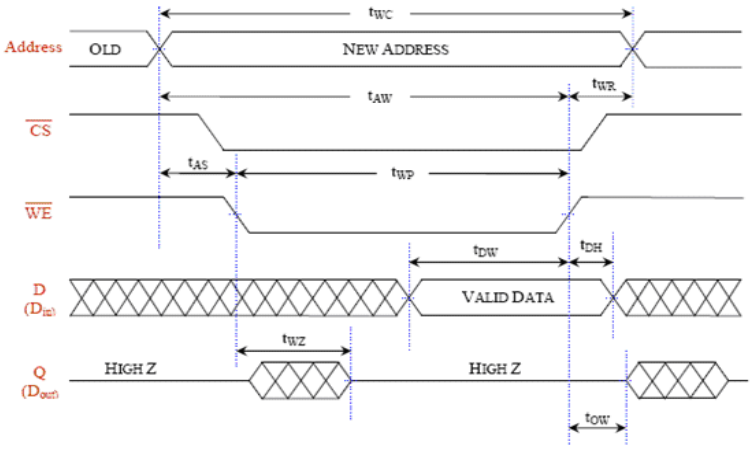


Figure 2.11: Timing diagram of Write cycle

It consists of

- System sets the correct addresses (A set).
- System selects the RAM by turning the chip select on ((CS)bar low).

- System waits a minimum required amount of time after changing the addresses for the RAM to do internal 'set up' of the addresses t_{AS}, and then turns the write enable on ((WE)bar low).

- System waits a minimum required amount of time after turning on the write enable (t_{WZ}) for the memory to disable the data output driver' Q' in preparation for using these lines for data input.

- System inputs the new data and waits a minimum required amount of time for the memory to write the data before turning off the write enable (t_{DW}).

- System waits a minimum required amount of time after turning the write enable on before turning it off (t_P). This is to be sure the write enable pulse width is wide enough for correctly writing the data into the RAM.

The expansions of the acronyms for read timing parameters are

t_{WC} : Write Cycle Time

t_{AS} : Address Setup Time

t_{AW} : Address Valid to End of Write

t_{WP} : Write Pulse Width

t_{DW} : Data Valid to End of Write

t_{DH} : Data Hold Time

t_{WZ} : Write Enable Low to Output High - Z

t_{OW} : Write Enable High to Output Active

t_{WR} : Write Recovery Time

The RAM is now ready to begin the next cycle. It should be clear that the minimum timings are periods when the system must wait for the RAM to do something, and the maximum timings are guaranteed limits within which the system will act.

Minimum access times for reading and writing are not necessarily the same, but for simplicity of design, most systems specify a single time for both reading and

writing. For semiconductor read-write memories, the read access time is typically 50 to 80

For internal timing, SRAMs apply a high number of clock impulses generated from an address change or chip enable signal in asynchronous systems, or and from the systems' master clock in synchronous systems. Synchronous and asynchronous operation relates to memory system interface modes rather than to chip-internal operation modes.

2.5 Different Types of SRAM Architectures

The memory architecture gives the idea about the organization which is used to benefit in speed, area, or power.

2.5.1 Basic Architecture

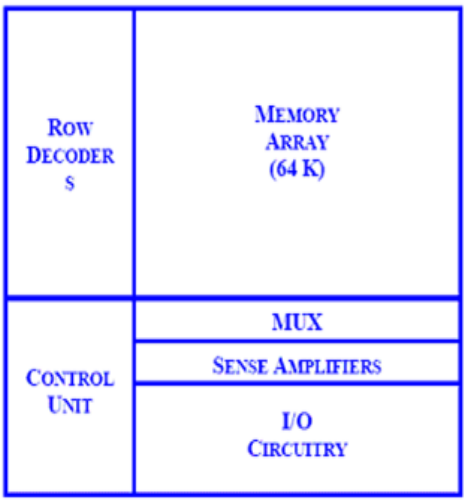


Figure 2.12: Floorplan of basic SRAM Architecture

In the conventional architecture, when the selected word line is high, all the cells connected to the wordline in the row are active,thus dissipation increases. In the

basic architecture, the two major factors which contribute to the read access are the bit access time and the word line access time. When the size of the SRAM increases, the number of cells connected to the word line increases. Therefore, the wordline delay increases because of the increase in the wordline capacitance. These two factors can be improved by reducing the bit line capacitance and the word line capacitance, but this is achieved only after using a different architecture.

2.5.2 Split Core Architecture

In this type of architecture, reduction is performed by splitting the memory array in smaller blocks. Reduction in the RC delay is observed because of the split bank, but the activation of a wordline activates the entire cell in both of the core areas. So certainly, there is need of a different architecture, which could also provide some advantage in terms of power dissipation.

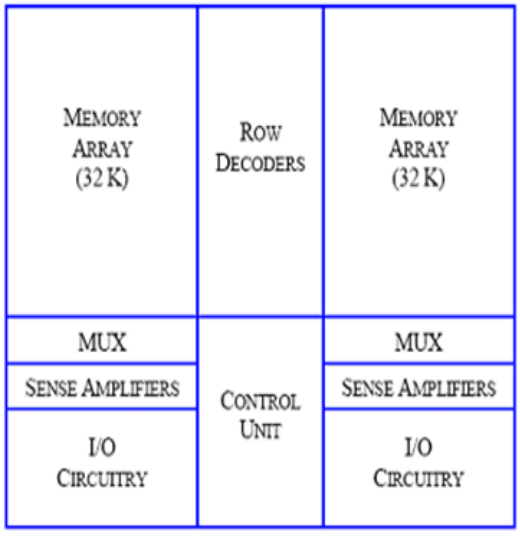


Figure 2.13: Floorplan of Split Core SRAM Architecture

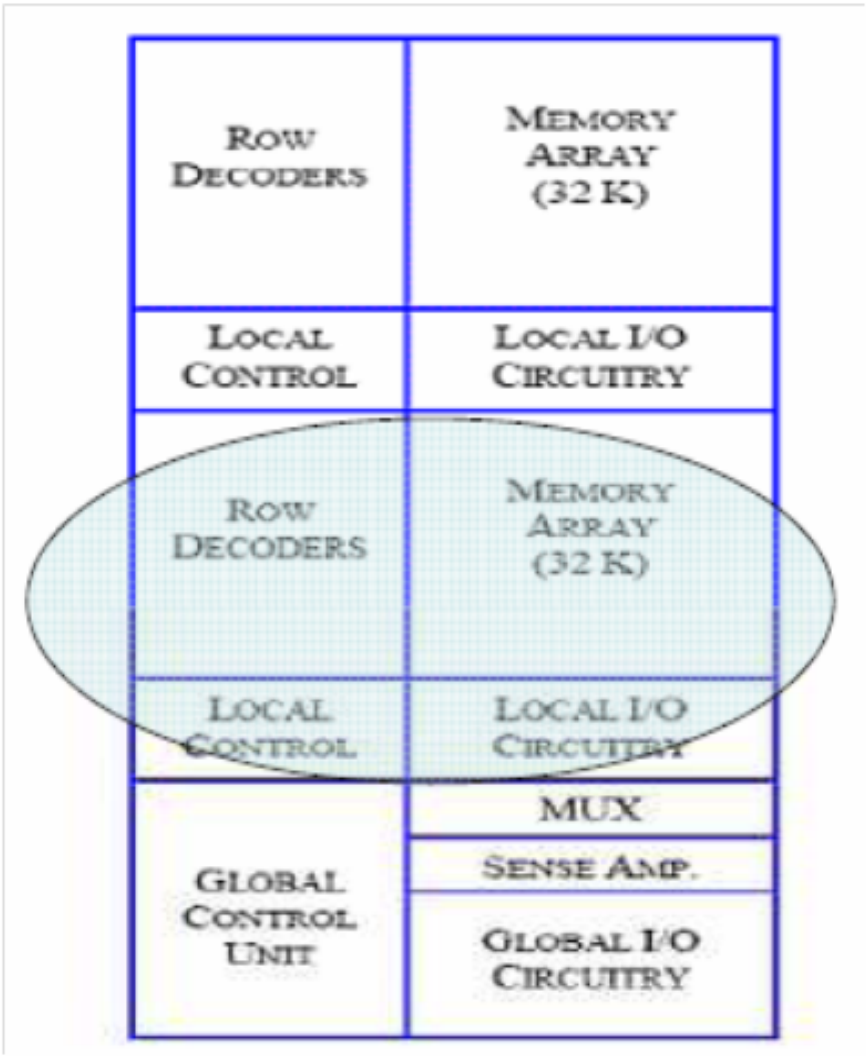


Figure 2.14: Floorplan for a64k SRAM cut with Bank type architecture (Bank =2)

2.5.3 Bank Architecture

In this type of architecture the control and the I/O sections are divided. So for a selected wordline, the cells of only one bank are activated. Also in case of bitline, the numbers of cells activated are reduced. Thus, a significant improvement is observed in case of wordline cap and the bit line cap. There is also reduction in the power consumption to a very significant value. But in this type of architecture, the area used is more and hence a less dense memory is obtained.

2.5.4 Redundancy Feature

Redundancy feature is an optional feature added in the memory architecture. This feature is like a on chip repair mechanism. There are two redundant rows provided as shown in Figure with only consecutive row correctibilty[2] . Incase memcells of any row fails ,that particular row will be replaced by redundant row.

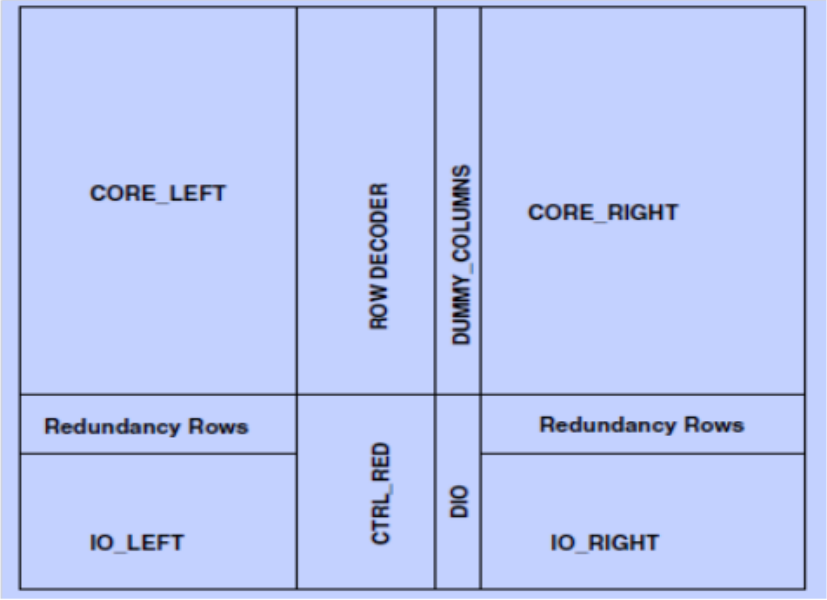


Figure 2.15: Floorplan for a cut with Redundancy(Bank=1)

2.6 A literature survey

2.6.1 Power Consumption in SRAM

With continuous advancements in technologies, the SRAM memories have undergone changes with respect to following parameters:

- Decrease in geometric cell size
- Increased transistor density

- Higher complexities of the peripheral and control circuitry and
- High frequency

Such circuits consume an excessive amount of power and generate increased amount of heat. In case of reduced power processors, memories contribute significantly to the system level power consumption by taking a share of 43

The circuits with more power dissipation are more susceptible to run-time failure and reliability problems. In addition, at increased temperatures, high power processors tend to create several silicon failures. As per studies, component failure rate double every 10C increase in temperature.

The solution for the problem is either to pursue expensive packaging or apply cooling strategies. However, another better option is to restrict the extensive heat generation.

The power consumption in SRAM can be divided in two modes of its operation i.e. Active and Standby. The power consumption in active mode is in the following sections

- The core area, it is the main location of power consumption in the SRAM Memory.
- The I/O section which uses power during precharge, multiplexer toggling, sensing and output driving also have a significant percentage of power consumption.
- The others sections which include predecoded line toggling and remaining periphery, the control and row decoder section have a small usage of power.

During the standby mode, the power consumption is very low which is used for the purpose of data retention. The main source in this mode is the leakage current in the Memcell.

The following techniques can be deployed for low power operation of SRAM memories:

1. Capacitance reduction of wordline and bitlines. This helps in reduction of main power consumption in the active mode of operation of memory.

2. Leakage current reduction by utilizing higher threshold voltage devices in the core. This factor helps in reduction of power usage in both of the active and standby modes.

3. Operating voltage reduction. This also needs to improve the periphery circuits accordingly.

4. AC current reduction by using new decoding schemes.

5. DC current reduction by improving pulse operation techniques for wordlines and periphery.

The analysis of SRAM based on various architectures focuses the first point above, i.e. the nerve point of maximum power consumption.

Reduction in power dissipation provides following advantages[3]:

- Better system efficiency is achieved.
- Performance of the system is improved.
- Reliability in enhanced.
- Overall cost is reduced.

2.6.2 High Speed Operation Of SRAM

As the technology improving day by day, similarly the trend toward higher memory density and large storage capacity is continued to push the leading edge of system design. For years, Memories have been the bottlenecks for achievement of faster systems. Today the SRAM is to be designed to match the processor speed with the main memory and improve the apparent speed of the main memory. The fastest low densities MOS static RAM have migrated into this area[2].

The SRAM access path delay (for read operation) can be split in to three parts :

1. Delay from clock implementation to wordline activation
2. Time taken from wordline activation to sense on action of the sense amplifier and
3. Time from sense on to availability of data at the output.

The main delay is in the second stage, where the delay is due to discharge of bit line. Therefore, a due consideration is given to it.

During write operation the delay components are

- Delay from clock implementation to wordline activation
- Time from data at the input to data written at the desired location.

This is always a compromise between the density and speed in memory array. If the important issue is speed then we have to take care of following parameters.

1. An important parameter is the memory access time i.e. memory access time, i.e., the time required to store and/or retrieve a particular data bit in the memory array. Access time is the important parameter to determine the performance criterion of the memory array.
2. To further speed up the read process, analog amplifiers are used. The tiny difference is rapidly converted into a logic level, without waiting until BL and (BL) bar reach their final voltage.

2.6.3 Low Power Approaches In SRAM

This presents the latest development in low-power circuit techniques and methods for static random access memories. The key techniques in power reduction in both

active and standby modes are: capacitance reduction by using divided word line structure, selective precharging scheme, pulse word line, ac current reduction by multistage decoding, operating voltage reduction coupled with low power sensing with sense amplifier.

2.6.4 Memory Compilers

Different compiler used in STM have different features like single port, dual port, high density etc. Each compiler specifies a range (min -max) of primary parameters like words, bits, mux and secondary parameters like bank, redundancy etc. A different combination of these parameters corresponds to different cuts i.e memory instance. A customer can choose a cut (with the parameters defined in the range of the compiler) depending on his requirement.

Memory Compiler uses predefined building blocks or leaf cells and connectivity information to compile the cut of user-specified size. The compiler generates the layout, behavioural level models, schematic symbols, and a layout abstraction to place and route. A program can automatically complete the creation of all the models in different levels.

While designing a compiler all the possible leaf cells have to be designed so that both the largest possible and smallest possible memory cut configurations can be achieved. It is the job of the tool or script to join the leaf cell to generate the schematics for the required combinations. While characterization, it is enough to extract the values for the most critical path.

Characteristics of memory depends upon the primary ,the secondary parameters and operating conditions. Different cuts will have different values for power consumption, timings, area etc. Different options offers an interesting timing-power-density-leakage trade-off. A higher bank value typically improves intrinsic memory access delays. However, it also means higher area and leakage of memory. A higher mux option in place of a lower mux is used for a gain in timing performance. While

a smaller mux option is able to provide better timing performance with a higher bank option.

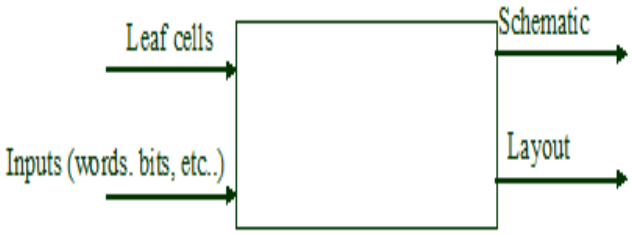


Figure 2.16: Memory Compiler

Chapter 3

Memory Scrambling Checker(MSC)

Scrambling means that the logical structure, as seen by the user from the outside of the chip, differs from the physical or topological internal structure of the chip. The consequence is that logically adjacent addresses may not be physically adjacent (this is called address scrambling)[4]

Address scrambling To minimize silicon area and delay, internal address lines (word-lines and bit-lines) are frequently scrambled.

A memory array includes a plurality of storage cells and a selection device which selects a storage cell for physical access due to a logical address supplied via an address bus. Scrambling allocates a storage cell in the memory array in predictable fashion to a logical address transmitted to one of the selection devices, the cell is then being physically accessed. The software-only detection of memory errors typically incurs a very high performance overhead, because the detection tool has to intercept memory accesses (loads and/or stores). For each access, the checker must find the status of accessed memory location (state lookup), e.g. to find whether the location is allocated, initialized, stores a return address, etc. The checker then verifies if the access is allowed by the location's status (state check), and possibly changes

the status of the memory location for future accesses (state update), for example to mark an uninitialized location as initialized if the access is a write[5].

While doing the electric check of the memory chip , if we there is some problem in reading or writing in a memcell then from the layout we know physically which memcell is in error but logically which location is in error can't be determined. *Memory Scrambling Checker (MSC)* resolves this issue. It provides mapping of physical and logical addresses and results are provided with the compiler for reference whenever required.

3.1 Directory Structure

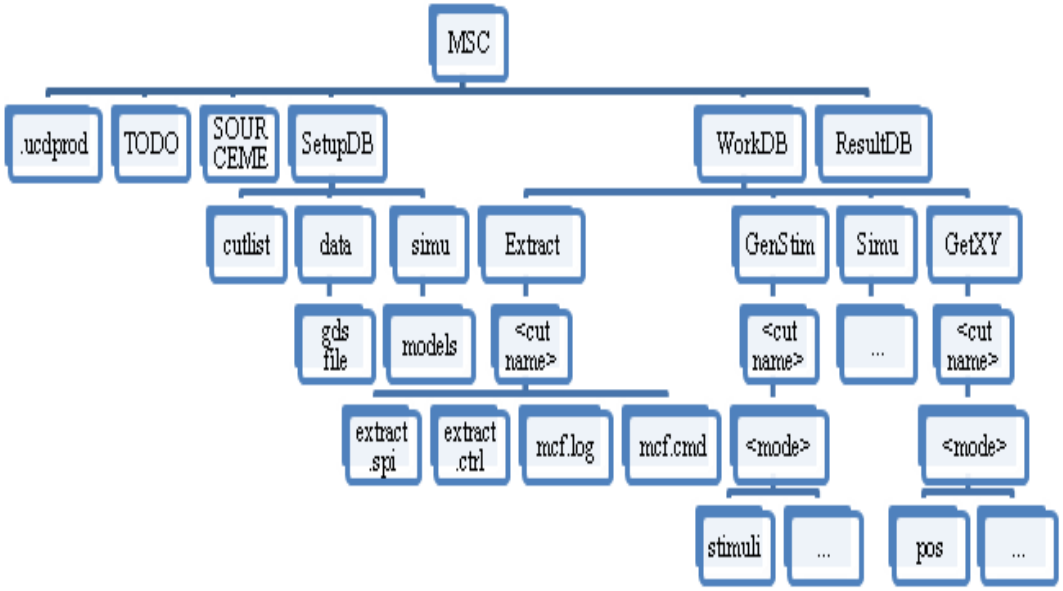


Figure 3.1: Directory Structure of MSC

Figure 3.1 shos the entire directory structure which is required as a setup before launching the MSC.

3.1.1 Setup files

- **.ucdprod**

  This file contains all the tools with their version which are needed for the simulation.

- **ToDo**

  ToDo is a file which needs to be sourced at a particular terminal where simulation is to be done. This file will set an environment variables which will be used while running the simulation. The environment variable contains path of the spice models and memcell ,and other important tools required.

- **Sourceme**

  Sourceme is a file which needs to be sourced at a particular terminal where simulation is to be done. This file will set an environment variables which will be used while running the simulation. The environment variables contains path of msc to be used and the simulator used.

3.1.2 Input Directory

- **SetupDB**

  This directory contains input configuration files for MSC. Those files are very useful to generate the appropriate coordinates in the output directories.

  - **Cutlist**

    Cutlist contains the number of cuts on which the simulation needs to be performed. It consists of parameters like cut name , words , bits , mux , banks , supply voltage , memcell name and input ports. (memcell name and input ports can be obtained from the extrcated netlist of the particular cut)

- **Data**

  It contains the gds file obtained by streaming out of the layout. It's a netlist file for the layout.

- **Simu**

  It contains the models file which includes all the .lib files that define the various instances used in the design of memory compiler. e.g. nfet , pfet , diodes etc.

3.1.3 Output Directories

1. **WorkDB**

   This directory contains input configuration files as well the output files or simulation result files.

   - **Exract**

     It contains a sub directory with the name as that of the cut ¡cut name¿. Further in that directory following input files are there.

     - **Extract.spi**

       It the extracted netlist file obtained after running lvs (layout vs schematic) check for the cdl and gds of required cuts of a memory compiler. Note: Ensure that the units of width and length of mosfets is in microns and the mosfets have same name as defined in the model files.

     - **Extract.ctrl**

     - **Mcf.log**

     - **Mcf.cmd**

       The latter three are the blank files that are to be created for futher processing during simulation. Following directories are created after launching the msc command.

- **GenStim**

  It contains a sub directory with ¡cut name¿ followed by a directory with mode and then following files:

  - **Stimuli**

    This file is generated by msc flow depending upon the mode specified. It provides the word-bit combinations for which physical locations i.e the xy coordinates are to be obtained.

  - **Mcf.log**

    This is log file generated during the stimuli generation task.

  - **Mcf.cmd**

    This file defines the task of stimuli generation.

- **Simu**

  This directory contains the msc simulation result files and intermediate files required for simulation and are generated by the tool itself depending on the inputs. For instance:

  - **Mcf.log**

    It is the log file for complete msc run from simulation to generating logical addresses.

  - **Simu.log**

    It is the log file depicting the simulation progress.

  - **Simu.err**

    This file consists of logical addresses of all the memcells being accessed during stimulation or the memcell corresponding to the specified word-bit combination alongwith the time its being executed.

  - **Paths**

    It picks up the logical path of the memcells accessed from simu.err file.

- **Vect.dvf** This file provides the address locations being toggled alongwith the clock cycle in binary form.
- **Gen.spi**

  It contains the voltage supplies to the various ports in the top level of design.
- **Simu.cfg**

  It contains all the options required to run the simulation i.e. various simulation commands to create required files to be used by msc to generate results.

- **GetXY**

  It contains a sub directory with ¡cut name¿ followed by a directory with mode and further it contains the pos file.
  - **Pos**

    It contains the word-bit combinations alongwith the corresponding XY coordinates.

2. **ResultDB**

   It is an outout directory that contains directory with cut name containing the file with XY coordinates alongwith word-bit. Its similar to pos file in WorkDB.

3.2 Methodology

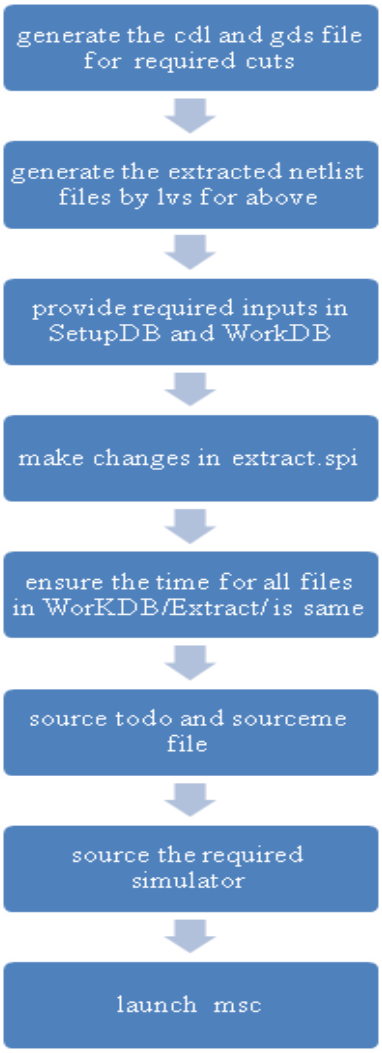
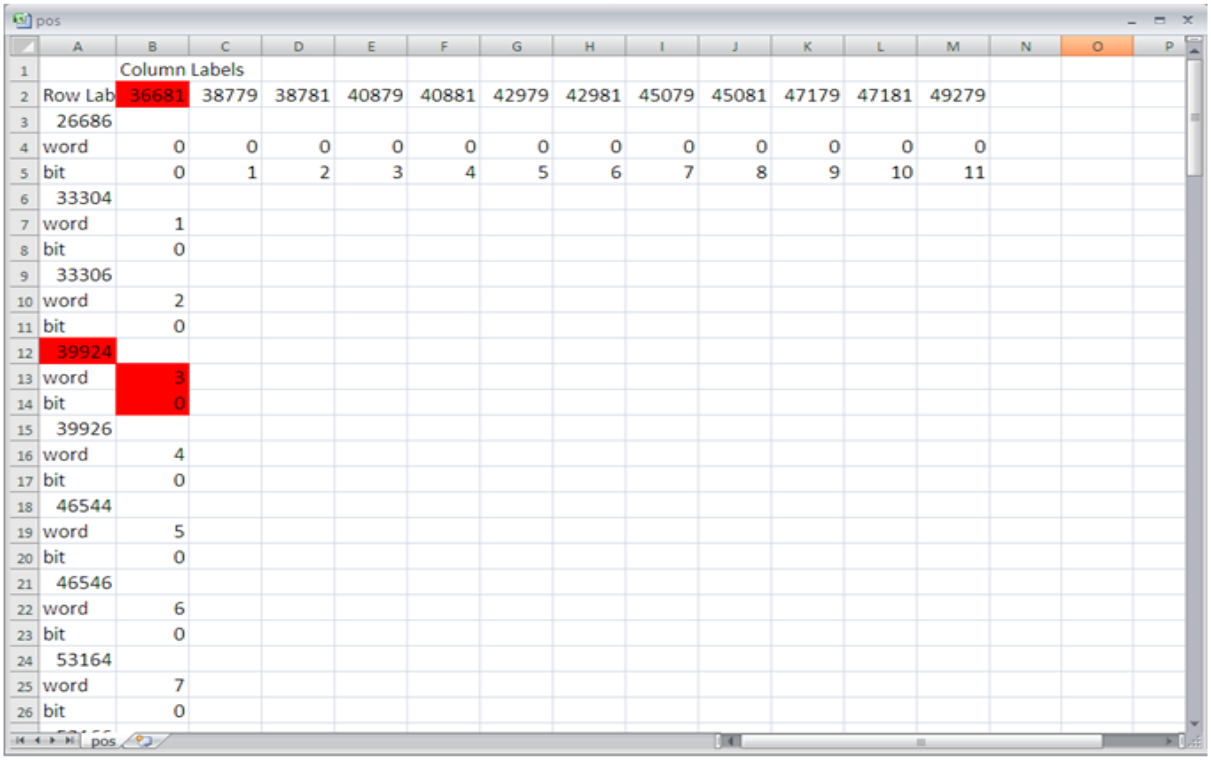


Figure 3.2: Flow of MSC methodology

3.3 Result



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		Column Labels														
2	Row Lab	36681	38779	38781	40879	40881	42979	42981	45079	45081	47179	47181	49279			
3	26686															
4	word	0	0	0	0	0	0	0	0	0	0	0	0			
5	bit	0	1	2	3	4	5	6	7	8	9	10	11			
6	33304															
7	word	1														
8	bit	0														
9	33306															
10	word	2														
11	bit	0														
12	39924															
13	word	3														
14	bit	0														
15	39926															
16	word	4														
17	bit	0														
18	46544															
19	word	5														
20	bit	0														
21	46546															
22	word	6														
23	bit	0														
24	53164															
25	word	7														
26	bit	0														

Figure 3.3: Location of memcell from MSC Results

Figure 3.3 shows the excel sheet prepared from the MSC results for 12x46m1(words=12, bits=46, mux=1). The row labels indicate the x coordinate and coloumn labels indicate the y coordinate for memcells being accessed during scrambling. A word-bit pair and its x-y coordinates are highlighted in red.

Chapter 4

Memory Characterization Flow(MCF)

Semiconductor memories are capable of storing large amount of Digital information. The ever increasing demand of low priced memories with low power consumption, high speed operation, high density and small package size has driven the fabrication technology & memory development towards more compact design rules. The memory data storage capacity of a single chip doubles almost after every two years. The number of data bits stored per unit area is one of the key criteria that determine the overall storage capacity & hence the memory cost per bit. Another important issue is the memory access time i.e. the time taken to store or retrieve data in the memory array. The access time determines the memory speed. Static & Dynamic Power consumption of the memory array is a significant factor to be considered in the design because of the increasing demand of the low power applications.[3]

Memory characterization means to obtain the information about the behaviour of memory in terms of different timings, power and pin cap. "Performance Characteristics" of SRAM is categorized as:

Timing Characterization: Setup, Hold, Access, Cycle time performance of

a given memory at any given PVT.

Power Characterization: Dynamic power, Stand-by Power & Leakage (Static) power of given memory at any given PVT

Pin Cap Characterization: Capacitance offered by input pins of a memory to the stage driving it.

Cut is a memory instance made up of a number of memcells which are made of transistors in turn. Every cut has some primary parameters. These parameters are words, bits, mux, other features are also there which define a cut like Bank, redundancy etc. These parameters depend upon the range and architecture of particular memory compiler. This is a step prior to the fabrication of a particular memory. Whenever customer uses memory in a chip , he checks whether the area , height ,width of memory of required words, bits , mux and capacity or any other feature can be fit into the space available and the timings can be met by processor and power consumption and leakage that can be allowed or not.

For the purpose of providing the various timings like cycle time , address setup and hold time , power, leakage etc. to customer using it ,the memory have to be characterized and to characterize the memory , it is to be simulated . In simulation , a stimuli (types of waveform which are applied to different inputs of memory) is given to a netlist. Netlist is basically whole memory circuit written in text form, it is defined in netlist which type of transistors , resistances , capacitors are connected at different places. Simulator is a tool which pass this stimuli (input) to netlist (memory in text form) and output waveforms are obtained. And using another tool measurement of different timings, power and leakage is done. Now the discussion of how the structure of memory is there is shown.

4.1 Directory structure of MCF

Figure 4.1 shows the directory structure of Memory Characterization Flow (MCF) is shown.[1]

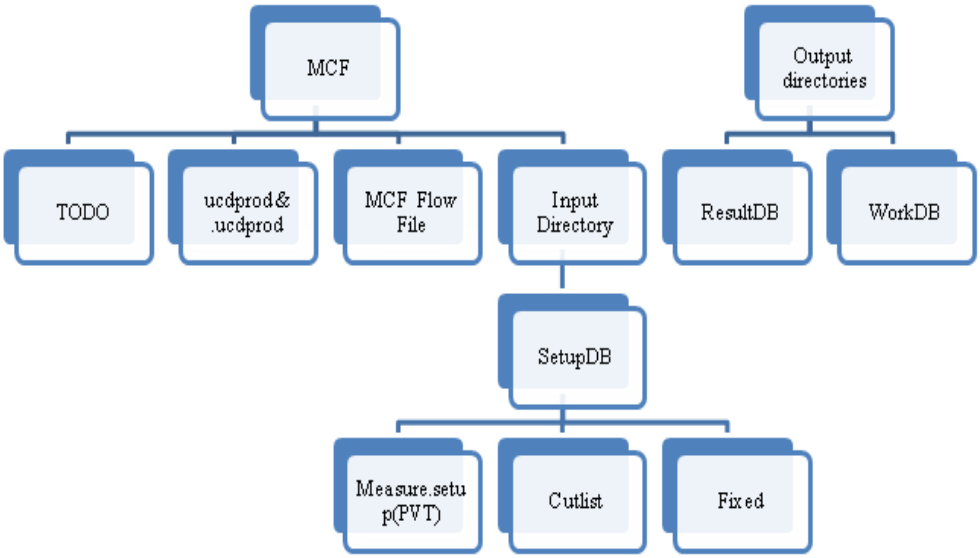


Figure 4.1: Directory structure of MCF

4.1.1 Setup files

- **ToDo**

  ToDo is a file which needs to be sourced at a particular terminal where simulation is to be done. This file will set an environment variables which will be used while running the simulation. The environment variable contains the path of license file of the simulator ,path of the spice models and memcell and path of other important tools required for simulation.

- **.ucdprod**

  This file contains all the tools with their version which are needed for the simulation.

- **ucdprod**

  This file contains the path of all the tools (where they are downloaded) which are defined in the .ucdprod.

4.1.2 Input Dirctory

- **SetupDB**

  This directory contains input configuration files for mcf. It contains two files are :-

  - **Cutlist**

    Cutlist contains the number of cuts on which the simulation needs to be performed. It consist of primary parameters (word, bits, mux) and secondary parameters (bank,redundancy etc) , which are used to define a cut. A cutlist can have a large number of cuts but the simulation is run only on the corner cuts in order to save time and resources.

  - **Measure.Setup**

    This file contains the different PVT condition at which the simulation needs to be done .PVT stands for Process , Voltage and Temperature. The process can be min, max and typ. For ex. Min means slow devices with large Vt , Process (Kind of process for N transistors) and Process (Kind of process for P transistors) are both defined in the file. Different supply voltages are also defined in this file. Temperature on which the device is supposed to work is also defined in this file. We also define for which task the simulation needs to be run .For ex. if have to run simulation only for timing ,it is defined in this file.

4.1.3 Output Directories

1. **WorkDB**

   The output directory WorkDB contains many sub directories . Each subdirectories contains the result files and intermediate files (stimuli files, cutlist, measure.setup, measure.script, analysis, header etc) and error log file used for the particular task (mentioned in the flow file)being run. So, all the output

files which are generated during the execution of flow are in the WorkDB. There are different types of directories for different tasks. The directories are named according to task,name and other neccessray parameters (specific to a memory) etc. One of important file in the WorkDB is the lbf.log file. The lbf.log file contains all the information starting from the time the command for mcf starts executing to the end of the execution. The lbf.log file contains the errors and warnings (which may help in debugging any error occurred during execution of the task.) as well as the results. This file also tells about the runtime of each simulator for the specific task. The result files present in subdirectory of the particular task are measure.result (in which the values of the measured parameters is given) and waveform files(with extension specific to a compiler).

2. **ResultDB**

   This is the output directory which contains only the results files . In the ResultDB there different files. The number of files in the ResultDB is equals to the total number of combinations of cuts defined in the cutlist and pvt conditions defined in measure.setup. For example if there are three cuts defined in the cutlist and two pvts defined in measure.script then ResultDB will have six files, each having data for each cut under different pvt conditions. These files contain only name of the parameter and its value mentioned in measure.result.

4.2 The Working of MCF

MCF takes an input setup (in directory SetupDB) and characterizes instances defined in cutlist file at the PVTs defined in measure.setup file. It starts the process by first preparing a "netlist" or spi for the exact instance and then simulates it with the help of other input files defined in the mcf setup.The various steps of MCF are as follow :

1. **Generation of spi or netlist**

   A netlist is needed for simulation. Generation of netlist has been described below

   - A netlist is generated from the schematic of the memory compiler (without extracted information).
   - This netlist is named as pre layout.
   - The pre layout contains all the leaf cells that may be needed for characterization.
   - A leaf cell is a sub circuit description of different blocks, present in the memory compiler. Eg: It can be the 'Core' block of the memory and it can also be as simple as a NAND gate. Then, from the layout of the memory compiler another set of netlists are generated , named as post layout netlist
   - The post layout also contains some leaf cells.
   - The number of leaf cells in the pre layout is more than the number of leaf cells in the post layout because the leaf cells of the post layout are bigger than the leaf cells of pre layout and many hierarchy levels may be flattened in one big post-layout leafcell.

     After this, the mcf tool function operates on the pre layout and the post layout alongwith basi.spi to produce a file named template.spi.From this template.spi and the file command.be as well as the cutlist , a spi file for each cut is generated . A spi file contains the netlist for the memory under characterisation which simulated with the simulator. For every cut & for every PVT defined in the cutlist and measure.setup file respectively will have a different spi file.

     The above process to is done by defining different task which are called from the specific characterization task .

2. **Simulation**

   A Circuit file is generated by the mcf tool which includes the spi file, analysis file, header file, stimuli , option file and other parameters required for simulation.Simulators used were Eldo (tool of Mentor Graphics), xa (tool of synopsis) etc. Eldo is considered as the golden simulator to generate the output. This is considered as golden because its results are assumed to be correct aand is used as a benchmark for other simulators . Simulators take spi, stimuli, analysis and option file as an input for simulation. On running mcf setup, a simulation is done by using the desired simulator. The number of simulations would be for all the combinations of cutlist and measure.setup entries.

3. **Measurements**

   The simulation produces some waveforms as outputs. The waveforms are stored in the filename with a specific extension (depends on the simulator used). Once the simulation is completed, the desired measurements are done on the simulated waveform using a tool and the results which we need to be measured are kept in the result files. The parameters to be measured are mentioned in measure.script. All the the result files are kept in the a separate directory.

4. **ResultDB**

   This is an output directory formed when the simulation are over. This directory only contains result files for a combination of cuts and pvts mentioned in mcf setup. The result file contains all the values of parameter used for characterization.The result file present here is processed in a specific format and then given to customers.

4.3 Flowchart of MCF Working

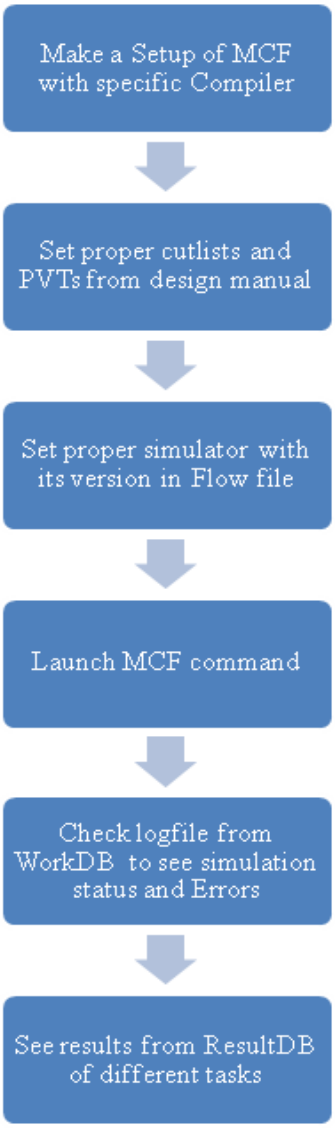


Figure 4.2: Flowchart to work with MCF

4.4 Results

1						XA(e-2010.12-sp2)			XA(g-2012.06)		XA vs. XA	
2	CUT	PVT	Variable_name	Ref_value	MP_value	DIFF	%DIFF	MP_value	DIFF	%DIFF	DIFF	%DIFF
3	16x4m2b1	TT	taa	0.35299	0.352409	-0.0006	-0.1646	0.35183	-0.0012	-0.3284	-0.00058	-0.16457
4	16x4m2b1	TT	tcycle	0.559964	0.559892	-0.0001	-0.0129	0.558798	-0.0012	-0.2082	-0.00109	-0.19578
5	16x4m2b1	TT	tah	0.178748	0.176163	-0.0026	-1.4461	0.174658	-0.0041	-2.2879	-0.0015	-0.86168
6	16x4m2b1	TT	tas	0.17605	0.174776	-0.0013	-0.7233	0.175035	-0.001	-0.5762	0.000259	0.14797
7	16x4m2b1	TT	tdh	0.137954	0.136533	-0.0014	-1.0304	0.13633	-0.0016	-1.1778	-0.0002	-0.1489
8	16x4m2b1	TT	tds	0.11287	0.111177	-0.0017	-1.5001	0.110658	-0.0022	-1.9592	-0.00052	-0.46901
9	16x4m2b1	TT	tds	0.108551	0.106871	-0.0017	-1.548	0.106624	-0.0019	-1.7753	-0.00025	-0.23166
10	16x4m2b1	TT	tds	0.09863	0.097303	-0.0013	-1.345	0.096954	-0.0017	-1.6991	-0.00035	-0.35996
11	16x4m2b1	SS	tds	0.230602	0.226519	-0.0041	-1.7707	0.227968	-0.0026	-1.1423	0.001449	0.635616
12	16x4m2b1	TT	tds	0.107998	0.106532	-0.0015	-1.3574	0.106252	-0.0017	-1.6167	-0.00028	-0.26352
13	32x4m4b1	TT	taa	0.356577	0.356943	0.0004	0.1027	0.35646	-0.0001	-0.0326	-0.00048	-0.1355
14	64x4m8b1	TT	taa	0.357602	0.358767	0.0012	0.3259	0.3582	0.0006	0.1671	-0.00057	-0.15829
15	16x144m2b1	SS	taa	0.889507	0.888328	-0.0012	-0.1325	0.892199	0.0027	0.3027	0.003871	0.433872
16	32x72m4b1	FF	taa	0.246022	0.246035	0	0.0053	0.245975	0	-0.019	-6E-05	-0.02439
17	2048x4m2b1	TT	taa	0.516612	0.515545	-0.0011	-0.2066	0.515867	-0.0007	-0.1441	-0.00032	0.05987
18	64x4m8b1	SS	tcycle	1.10087	1.095274	-0.0056	-0.5083	1.09585	-0.005	-0.456	0.000576	0.052562
19	16x144m2b1	TT	tcycle	0.690021	0.688666	-0.0014	-0.1964	0.688799	-0.0012	-0.1771	0.000133	0.019309
20	32x72m4b1	FF	tcycle	0.365757	0.363852	-0.0019	-0.5209	0.364595	-0.0012	-0.3178	0.000743	0.203788
21	512x4m2b1	TT	tcycle	0.67616	0.675235	-0.0009	-0.1368	0.67565	-0.0005	-0.0754	0.000415	0.061422
22	2048x4m2b1	FF	tcycle	0.363656	0.361913	-0.0017	-0.4792	0.362173	-0.0015	-0.4077	-0.00026	0.061265
23	32x4m4b1	SS	tah	0.266123	0.265871	-0.0003	-0.0944	0.262751	-0.0034	-1.2669	-0.00312	-1.18744
24	64x4m8b1	SS	tah	0.264705	0.265205	0.0005	0.1887	0.259799	-0.0049	-1.8536	-0.00541	-2.08084
25	16x144m2b1	TT	tah	0.180877	0.178834	-0.002	-1.1295	0.176943	-0.0039	-2.175	-0.00189	-1.06871
26	16x144m2b1	SS	tah	0.271022	0.270874	-0.0001	-0.0547	0.26607	-0.005	-1.8272	-0.0048	-1.80554
27	32x72m4b1	TT	tah	0.180171	0.17783	-0.0023	-1.2995	0.175609	-0.0046	-2.5323	-0.00222	-1.26474
28	512x4m2b1	TT	tah	0.180392	0.178028	-0.0024	-1.3104	0.176267	-0.0041	-2.2869	-0.00176	-0.99905
29	512x4m2b1	SS	tah	0.27026	0.269946	-0.0003	-0.1164	0.266952	-0.0033	-1.2239	0.002994	0.087983
30	512x4m2b1	FF	tah	0.120406	0.120414	0	0.0062	0.119208	-0.0012	-0.9953	0.001206	0.08478

Figure 4.3: Result by MCF for single compiler

Figure 4.4 shows the results which are taken after characterize memory compiler for timing with different spice simulators. It also shows values of different timing parameters like access time, hold time, setup time etc.

Chapter 5

Scripts

5.1 msc-genvect

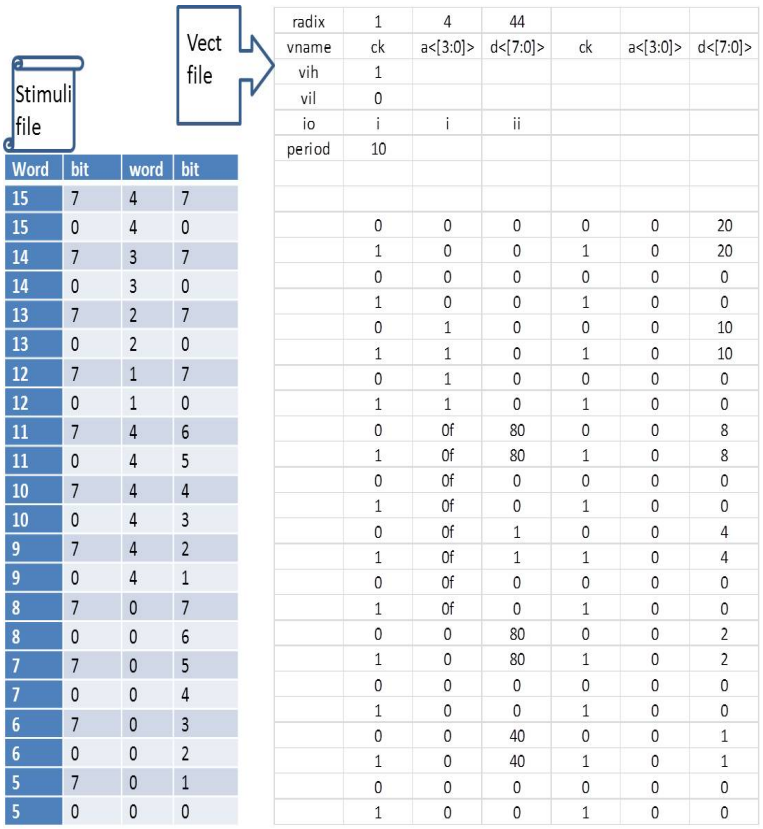
This script generates vector file from stimuli file. Stimuli file gives the word-bit combination that needs to be excited during memory scrambling to check the errors. But compiler can not take input in human readable format. It needs to be converted in some machine readable format which is called as vect file.

From vect file, the simulator stimulates given word-bit combination and the report is generated in path file, pos file and other files that are human readable.

5.2 Pos vs. bitmap comparison

Pos file and bitmap files are generated from two different sources. Pos file contains limited information about the x-y combinations corresponding to given word-bit combinations. Where as bitmap file contains information about muxes, banks, redundancy rows and x-y coordinates of given word-bit combination. Bitmap files are generated from mathematical equations and it shows pure ideal condition. We can say that it is textual representation of how the memory should be arranged in ideal situation.

This scripts checks any discrepancy in both the files and reports it.



Word	bit	word	bit
15	7	4	7
15	0	4	0
14	7	3	7
14	0	3	0
13	7	2	7
13	0	2	0
12	7	1	7
12	0	1	0
11	7	4	6
11	0	4	5
10	7	4	4
10	0	4	3
9	7	4	2
9	0	4	1
8	7	0	7
8	0	0	6
7	7	0	5
7	0	0	4
6	7	0	3
6	0	0	2
5	7	0	1
5	0	0	0

radix	1	4	44			
vname	ck	a<[3:0]>	d<[7:0]>	ck	a<[3:0]>	d<[7:0]>
vih	1					
vil	0					
io	i	i	ii			
period	10					
	0	0	0	0	0	20
	1	0	0	1	0	20
	0	0	0	0	0	0
	1	0	0	1	0	0
	0	1	0	0	0	10
	1	1	0	1	0	10
	0	1	0	0	0	0
	1	1	0	1	0	0
	0	0f	80	0	0	8
	1	0f	80	1	0	8
	0	0f	0	0	0	0
	1	0f	0	1	0	0
	0	0f	1	0	0	4
	1	0f	1	1	0	4
	0	0f	0	0	0	0
	1	0f	0	1	0	0
	0	0	80	0	0	2
	1	0	80	1	0	2
	0	0	0	0	0	0
	1	0	0	1	0	0
	0	0	40	0	0	1
	1	0	40	1	0	1
	0	0	0	0	0	0
	1	0	0	1	0	0

Figure 5.1: Stimuli File and Corresponding vect file

5.3 msc - checkxy

This script check these types of errors:

1. Multiple Word bit combinations pointing to same coordinates
2. Single (word, bit) combination corresponds to Multiple mem cell locations - short circuit
3. (word, bit) combination does not point to any location in the core - open circuit

After toggling the bitcells simulator generates a report which has the following information, 1) time stamp when the cell is toggled, 2) (x,y) coordinates where the toggling has actually took place, 3) whole hierarchy of the bitcell.

word	bit	x	y	word	bit	x	y
0	0	42687	45586	97	22	102116	156884
0	1	35914	45586	97	23	108734	156884
0	2	35914	45586	97	24	108736	156884
0	3	42536	45586	97	25	115354	156884
0	4	42536	45586	97	26	115356	156884
0	5	49154	45586	97	27	121974	156884
0	6	49154	45586	97	28	121976	156884
0	7	55774	45586	97	29	128594	156884
0	8	55776	45586	97	30	128596	156884
0	9	62394	45586	255	119	434894	335384
0	10	62396	45586	255	120	434896	335384
0	11	69014	45586	255	121	441514	335384
0	12	69016	45586	255	122	441516	335384
0	13	75634	45586	255	123	448134	335384

Figure 5.2: Pos file

From this information we can extract the three types of errors mentioned above. This extraction task has been done through a script written in perl.

5.4 Masis Tool: GDS vs .masis file comparison

The actual gdsvsmasis.tcl script calls various tools including mpt (memory pattern generator), vpg (verification pattern generator), xa or eldo simulator, hercules lvs extractor to replicate the whole task covered by MSC tool.

But few of the tool are not supported in ST, also the tool calling mechanism is different in ST environment. The methodology used by MASIS tool is totally different from that is being used by ST. So the understanding of MASIS tool is going on. Simultaneously the script that is being run by MASIS tool is being changed as per the understanding. Tool calling methods have been changed, unsupported tools were removed and their counter parts were inserted in the script.

#CUT PROVIDED NAME : ST_SPSMALL_256x128m1_bH							
#CUT PARAMETER NAME : ST_SPSMALL_256x128m1_bH							
#ADDRESS	BIT	ROW	COLUMN	LOWERLEF	LOWERLEF	UPPERRIG	UPPERRIG
0	1	0	1	32605	45585	35915	46635
0	2	0	2	35915	45585	39225	46635
0	3	0	3	39225	45585	42535	46635
0	4	0	4	42535	45585	45845	46635
0	5	0	5	45845	45585	49155	46635
0	6	0	6	49155	45585	52465	46635
0	7	0	7	52465	45585	55775	46635
0	8	0	8	55775	45585	59085	46635
0	9	0	9	59085	45585	62395	46635
0	10	0	10	62395	45585	65705	46635
0	11	0	11	65705	45585	69015	46635

Figure 5.3: CSV file

csvadd	posadd	csvbit	posbit	xpos	llx(x1)	urx(x2)	ypos	lly(y1)	ury(y2)
0	0	2	2	35914	35915	39225	45586	45585	46635
0	0	3	3	42536	39225	42535	45586	45585	46635
0	0	6	6	49154	49155	52465	45586	45585	46635

Figure 5.4: Errorneous Rows - Pos vs CSV comparison file

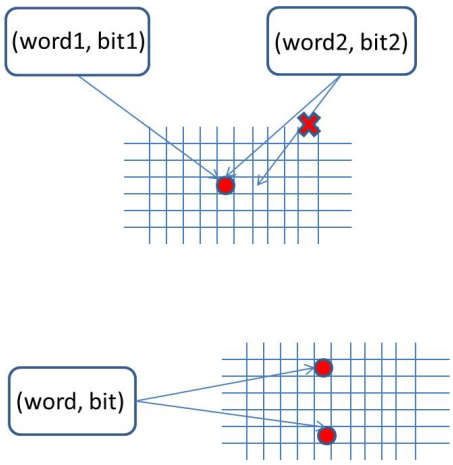


Figure 5.5: Types of errors that can occur in memory scrambling

Conclusion

This project has greatly increased my knowledge in the field of design flow. I have learnt a lot about the architecture, working and characterization of the memory, need of physical and logical addresses of memcell in a memory compiler.

One can run simulation on any kind of stimuli on the newer MSC setup. It has the flush/refresh cycle in it which refreshes the memory array before accessing the next memcell. Testing of a memory chip requires the user to know about the locations(physical and logical). The latest MSC tool verifies various toggling errors, it cross checks bitmap file vs MSC output, hence the reliability has incrased greatly. It includes learning basic Perl and Shell languages.

Also there are some changes done in the setup of Characterization tool (MCF) which will make the characterization process fast. Simulators from various vendors are cross checked on a common platform, their gain and losses were calculated and feeded back. This has increased my confidence and sense of responsibility a lot.

Future scope of work

With the shrinking technology the complexities in Design Flow are increasing, hence human efforts have to be increased. Here automation comes into the picture. Automation can be implemented at various design stages in Memory Design flow. It also enhances the efficiency and reduces human efforts.

The enhancement in MSC and MCF can be done as per follow :

- MSC
  - Improvement in MSC accordingly to the shrinking Technology
  - Runtime improvement
  - To make MSC compatible with multiple simulators
  - Testing the newer tool MASIS to reduce the designer's efforts nearly to zero
- MCF
  - Runtime improvement
  - Efforts reduction
  - Adding flexibility with options

References

[1] ST Internal Modules

[2] A.Hardi, B. Bhaumik, P. Pradhan, R. Varambally, " A Low Power 256 KB SRAM Design,"

[3] Ad. J. van de Goor , Ivo Schanstra "Address and Data Scrambling: Causes and Impact on Memory Tests," IEEE International Workshop on Electronic Design, Test and Applications (DELTA.02), 2002.

[4] Sung-Mo Kang, Yusuf Leblebici, CMOS Digital Integrated Circuit.

Annexures

- **Glossary**

  1. **Memcell**: It is the basic component present in the memory array organisation. Individuals memory cells are arranged in an array of horizontal rows and vertical column. Each cell is capable of storing one bit of information.
  2. **View**: A particular representation of cell in industry standard format. Each cell may have a layout view, schematic view, timing view etc. Each view specializes in providing a particular information of cell, in a standard or usable format. Some views are related to physical design of cell and some are related to Performance of cellThus enabling designers, to accurately analyze any given design, which in turn helps in optimizing the design performance
  3. **Library**: is a collection of various views that are useful in designing a chip
  4. **PVT**: The abbreviation stands for Process, Voltage and Temperature. The operating condition cannot be a fixed value,

thus any design/cell must be qualified to perform under a range of PVTs. Since all the three have direct impact on cell's performance (speed, power consumption, leakage etc.), operating conditions are an important consideration for designers.

5. **Technology**: Defined as the process in which the silicon is being manufactured. It is the minimum gate length possible.
6. **Critical Path**: The longest path traversed by an input signal to reach at output node in the circuit to produce the desired output. This path decides the maximum delay for an input signal to travel from input node to output node. If this delay is characterised then all other delays would be less than this and need not be characterised .
7. **UNICAD**: stands for 'Unified Computer Aided Design'. UNICAD is a concept deployed in ST to enable the combined usage of different tools from various vendors and ST's in house tools/product ensuring the compatibility between them.
8. **Cut**: A 'cut' is the name given to memory serving (user given) set of specifications. Memory designed in ST caters to a range of words/bits/mux etc.
9. **Generators**: A generator is a UNICAD product which supports the creation of a view (& related task) in accordance to the user given inputs. The user given input includes the

specifications of memory (i.e. cut), operating condition & demanded view. Generators are categorized into FE, BE and Model, where each of them supports a defined set of view.

10. **FE Generator**: Front-End Generator is unicad product which caters to characterization specific view generation (e.g. datasheet, apache, timing etc.)
11. **BE Generator**: Back-End Generator is a unicad product which caters to physical views generation (e.g. CDL, GDS etc.)
12. **Model Generator**: Model Generator is unicad product which caters to behavioural views generation(e.g. Verilog)
13. **Compiler**: All the three generators FE, BE and Model corresponding to same memory design constitute a compiler for the given memory.
14. **Aspect ratio**: is the ratio of row to the column
15. **Cdl**: Its the netlist file of the schematic defining the circuit and the subcircuits used in the design. It defines all the pins and ports and their connections in spice format.
16. **Gds**: Its the layout netlist file defining the circuit and the subcircuits and the layers used for all interconnects and instances.