

LOW LEAKAGE SRAM CELL WITH IMPROVED STABILITY FOR IOT APPLICATION

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electronics and Communication Engineering

by

ATLURI SONY (38130020)

CHODA AHALYA REDDY (38130047)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MAY - 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with “A” grade by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in



DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of ATLURI SONY (38130020), CHODA AHALYA REDDY (38130047) who carried out the project entitled “**LOW LEAKAGE SRAM CELL WITH IMPROVED STABILITY IOT APPLICATION**” under our supervision from November 2021 to April 2022 .

Internal Guide

Mrs.K.SRILATHA.,M.E.,(Ph.D).,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

We **ATLURI SONY** and **CHODA AHALYA REDDY** here by declare that the Project Report entitled “LOW LEAKAGE SRAM CELL WITH IMPROVED STABILITY IOT APPLICATIONS” done by us under the guidance of Mrs. **K.SRILATHA, M.E., (Ph.D).**, Assistant Professor, Department of Electronics and Communication Engineering is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **ELECTRONICS AND COMMUNICATIONENGINEERING**.

1)

2)

DATE:

PLACE: CHENNAI

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey thanks to **Dr.N.M.NANDHITHA.,M.E.,Ph.D.**, Dean, School of ELECTRICAL AND ELECTRONICS and **Dr.T.RAVI.,M.E.,Ph.D.**, Head of the Department, Dept. of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Mrs.K.SRILATHA.,M.E.,(Ph.D).**, Assistant Professor, Dept of Electronics and Communication Engineering for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of us project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

A novel 8-transistor (8T) static random access memory cell with improved data stability in sub threshold operation is designed. The proposed single-ended with dynamic feedback control 8T static RAM (SRAM) cell enhances the static noise margin (SNM) for ultra low power supply. It achieves write SNM of 1.4x and 1.28x as that of iso area 6T and read-decoupled 8T (RD-8T), respectively, at 300 mV. The standard deviation of write SNM for 8T cell is reduced to 0.4x and 0.56x as that for 6T and RD-8T, respectively. It also possesses another striking feature of high read SNM $\sim 2.33x$, $1.23x$, and $0.89x$ as that of 5T, 6T, and RD-8T, respectively. The cell has hold SNM of 1.43x, 1.23x, and 1.05x as that of 5T, 6T, and RD-8T, respectively. The write time is 71% lesser than that of single-ended asymmetrical 8T cell. The proposed 8T consumes less write power 0.72x, 0.6x, and 0.85x as that of 5T, 6T, and iso area RD-8T, respectively. The read power is 0.49x of 5 T, 0.48% of 6T, and 0.64x of RD-8T. The power/energy consumption of 1-kb 8T SRAM array during read and write operations is 0.43x and 0.34x, respectively, of 1-kb 6T array. These features enable ultralow power applications of 8T. The proposed architecture of this paper area and power consumption analysis using tanner tool.

CONTENTS

CHAPTER	TITLE	PAGE
No.		
	ABSTRACT	5
	LIST OF ABBREVIATIONS	9
	LIST OF FIGURES	10
1	INTRODUCTION	11-24
	1.2 WHY VLSI?	
	1.3 STRUCTURED DESIGN	
	1.4 WHAT IS VLSI	
	1.5 HISTORY & EVOLUTION	
	1.6 FUTURE OF VLSI	
	1.6.1 System Design	
	1.6.2 Switches	
	1.7 Semiconductors and Doping	
	1.7.1 Fabrication Technology	
	1.7.2 Conventional Approach to Digital Design	
	1.7.3 Design of VLSI	
	1.8 VLSI AND SYSTEMS	
	1.9 APPLICATIONS OF VLSI	
	1.9.1 Electronic system in cars	
	1.9.2 Digital electronics control VCRs	
	1.9.3 Transaction processing system, ATM	
	1.9.4 Personal computers and Workstations	
	1.9.5 Medical electronic systems	
	1.10 ASIC	

	<i>1.10.1 ASIC DESIGN FLOW</i>	
2	LITERATURE SURVEY	25
3	EXISTING SYSTEM	26
4	PROPOSED SYSTEM	27-31
5	RESULT AND DISCUSSION	32-54
	<i>5.1 XILINX</i>	
	<i>5.2 XILINX ISE 13.1</i>	
	<i>5.2.1 Simulation</i>	
	<i>5.2.2 Synthesis</i>	
	<i>5.2.3 Procedure</i>	
	5.3 LANGUAGES	
	<i>5.3.1 VHDL & VERILOG</i>	
	<i>5.3.2 History of the IEEE 1364 Verilog standard</i>	
	<i>5.3.3 MODELING ENHANCEMENTS</i>	
	<i>5.3.4 VERILOG</i>	
	<i>5.3.5 DEFINITION OF CONSTANTS</i>	
	<i>5.3.6 SYNTHESIZABLE CONSTRUCTS</i>	
	<i>5.3.7 INITIAL AND ALWAYS</i>	
	<i>5.3.8 FORK/JOIN</i>	
	<i>5.3.9 FOUR-VALUED LOGIC</i>	
	<i>5.3.10 SYSTEM TASKS</i>	

6	SUMMARYANDCONCLUSION	
	6.1 Summary	55
	6.2 Conclusion	56
7	REFERENCE	57-58

LIST OF ABBREVIATIONS

SRAM	STATIC RANDOM ACCESS MEMORY
RWL	READ WORLD LINE
RBL	READ BIT LINE
WWL	WRITE WORD LINE
WBL	WRITE BIT LINE

LIST OF FIGURES

FIGURENUMBER	FIGURENAME	PAGENO
4.1	Schematic diagram	30

CHAPTER-1

INTRODUCTION

The portable microprocessor controlled devices contain embedded memory, which represents a large portion of the system-onchip (SoC). These portable systems need ultralow power consuming circuits to utilize battery for longer duration.

The power consumption can be minimized using nonconventional device structures, new circuit topologies, and optimizing the architecture. Although, voltage scaling has led to circuit operation in sub threshold regime with minimum power consumption, but there is a disadvantage of exponential reduction in performance. The circuit operation in the sub threshold regime has paved path toward ultralow power embedded memories, mainly static RAMs (SRAMs).

However, in sub threshold regime, the data stability of SRAM cell is a severe problem and worsens with the scaling of MOSFET to sub nanometer technology. Due to these limitations it becomes difficult to operate the conventional 6-transistor (6T) cell at ultralow voltage (ULV) power supply .

1.1 INTRODUCTION OF VLSI:

VLSI Design presents state-of-the-art papers in VLSI design, computer-aided design, design analysis, design implementation, simulation and testing. Its scope also includes papers that address technical trends, pressing issues, and educational aspects in VLSI Design. The Journal provides a dynamic high-quality international forum for original papers and tutorials by academic, industrial, and other scholarly contributors in VLSI Design.

The development of microelectronics spans a time which is even lesser than the average life expectancy of a human, and yet it has seen as many as four generations. Early 60's saw the low density fabrication processes classified under Small Scale Integration (SSI) in which transistor count was limited to about 10. This rapidly gave way to Medium Scale Integration in the late 60's when around 100 transistors could be placed on a single chip.

It was the time when the cost of research began to decline and private firms started entering the competition in contrast to the earlier years where the main burden was borne by the military. Transistor-Transistor logic (TTL) offering higher integration densities outlasted other IC families like ECL and became the basis of the first integrated circuit revolution. It was the production of this family that gave impetus to semiconductor giants like Texas Instruments, Fairchild and National Semiconductors. Early seventies marked the growth of transistor count to

about 1000 per chip called the Large Scale Integration.

By mid-eighties, the transistor count on a single chip had already exceeded 1000 and hence came the age of Very Large Scale Integration or VLSI. Though many improvements have been made and the transistor count is still rising, further names of generations like ULSI are generally avoided. It was during this time when TTL lost the battle to MOS family owing to the same problems that had pushed vacuum tubes into negligence, power dissipation and the limit it imposed on the number of gates that could be placed on a single die.

The second age of Integrated Circuits revolution started with the introduction of the first microprocessor, the 4004 by Intel in 1972 and the 8080 in 1974. Today many companies like Texas Instruments, Infineon, Alliance Semiconductors, Cadence, Synopsys, Celox Networks, Cisco, Micron Tech, National Semiconductors, ST Microelectronics, Qualcomm, Lucent, Mentor Graphics, Analog Devices, Intel, Philips, Motorola and many other firms have been established and are dedicated to the various fields in "VLSI" like Programmable Logic Devices, Hardware Descriptive Languages, Design tools, Embedded Systems etc.

In 1980s, hold-over from outdated taxonomy for integration levels obviously influenced from frequency bands, i.e. HF, VHF, and UHF. Sources disagree on what is measured (gates or transistors)

SSI - Small-Scale Integration (0-102)

MSI - Medium-Scale Integration (102 -103)

LSI - Large-Scale Integration (103 -105)

VLSI - Very Large-Scale Integration (105 - 107)

ULSI - Ultra Large-Scale Integration (≥ 107)

VLSI Technology, Inc. was a company which designed and manufactured custom and semi-custom ICs. The company was based in Silicon Valley, with headquarters at 1109 McKay Drive in San Jose, California. Along with LSI Logic, VLSI Technology defined the leading edge of the application-specific integrated circuit (ASIC) business, which accelerated the push of powerful embedded systems into affordable products. The company was founded in 1979 by a trio from Fairchild Semiconductor by way of Synertek - Jack Balletto, Dan Floyd, and Gunnar Wetlesen - and by Doug Fairbairn of Xerox PARC and Lambda (later VLSI Design) magazine.

Alfred J. Stein became the CEO of the company in 1982. Subsequently VLSI built its first fab in San Jose; eventually a second fab was built in San Antonio, Texas. VLSI had its initial public offering in 1983, and was listed on the stock market as (NASDAQ: VLSI). The company was later acquired by Philips and survives to this day as part of NXP Semiconductors.

The first semiconductor chips held two transistors each. Subsequent advances added more and more transistors, and, as a consequence, more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now retrospectively as small-scale integration (SSI), improvements in technique led to devices with hundreds of logic gates, known as medium-scale integration (MSI). Further improvements led to large-scale integration (LSI), i.e. systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and billions of individual transistors.

At one time, there was an effort to name and calibrate various levels of large-scale integration above VLSI. Terms like ultra-large-scale integration (ULSI) were used. But the huge number of gates and transistors available on common devices has rendered such fine distinctions moot. Terms suggesting greater than VLSI levels of integration are no longer in widespread use.

As of early 2008, billion-transistor processors are commercially available. This is expected to become more commonplace as semiconductor fabrication moves from the current generation of 65 nm processes to the next 45 nm generations (while experiencing new challenges such as increased variation across process corners). A notable example is NVidia's 280 series GPU. This GPU is unique in the fact that almost all of its 1.4 billion transistors are used for logic, in contrast to the Itanium, whose large transistor count is largely due to its 24 MB L3 cache. Current designs, as opposed to the earliest devices, use extensive design automation and automated logic synthesis to lay out the transistors, enabling higher levels of complexity in the resulting logic functionality. Certain high-performance logic blocks like the SRAM (Static Random Access Memory) cell, however, are still designed by hand to ensure the highest efficiency (sometimes by bending or breaking established design rules to obtain the last bit of performance by trading stability)[citation needed]. VLSI technology is moving towards radical level miniaturization with introduction of NEMS technology. A lot of problems need to be sorted out before the transition is actually made.

1.2 WHY VLSI?

Integration improves the design, lowers the parasites, which means higher speed and lower power consumption and physically smaller. The Integration reduces manufacturing cost - (almost) no manual assembly.

The course will cover basic theory and techniques of digital VLSI design in CMOS technology. Topics include: CMOS devices and circuits, fabrication processes, static and dynamic logic structures, chip layout, simulation and testing, low power techniques, design tools and

methodologies, VLSI architecture. We use full-custom techniques to design basic cells and regular structures such as data-path and memory. There is an emphasis on modern design issues in interconnect and clocking. We will also use several case-studies to explore recent real-world VLSI designs (e.g. Pentium, Alpha, PowerPC Strong ARM, etc.) and papers from the recent research literature. On-campus students will design small test circuits using various CAD tools. Circuits will be verified and analyzed for performance with various simulators. Some final project designs will be fabricated and returned to students the following semester for testing.

Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors.

The first semiconductor chips held one transistor each. Subsequent advances added more and more transistors, and, as a consequence, more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now known retrospectively as "small-scale integration" (SSI), improvements in technique led to devices with hundreds of logic gates, known as large-scale integration (LSI), i.e. systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and hundreds of millions of individual transistors.

At one time, there was an effort to name and calibrate various levels of large-scale integration above VLSI. Terms like Ultra-large-scale Integration (ULSI) were used. But the huge number of gates and transistors available on common devices has rendered such fine distinctions moot. Terms suggesting greater than VLSI levels of integration are no longer in widespread use. Even VLSI is now somewhat quaint, given the common assumption that all microprocessors are VLSI or better.

As of early 2008, billion-transistor processors are commercially available, an example of which is Intel's Montecito Itanium chip. This is expected to become more commonplace as semiconductor fabrication moves from the current generation of 65 nm processes to the next 45 nm generations (while experiencing new challenges such as increased variation across process corners). Another notable example is NVIDIA's 280 series GPU. This microprocessor is unique in the fact that its 1.4 Billion transistor count, capable of a teraflop of performance, is almost entirely dedicated to logic (Itanium's transistor count is largely due to the 24MB L3 cache). Current

designs, as opposed to the earliest devices, use extensive design automation and automated logic synthesis to lay out the transistors, enabling higher levels of complexity in the resulting logic functionality. Certain high-performance logic blocks like the SRAM cell, however, are still designed by hand to ensure the highest efficiency (sometimes by bending or breaking established design rules to obtain the last bit of performance by trading stability).

The original business plan was to be a contract wafer fabrication company, but the venture investors wanted the company to develop IC (Integrated Circuit) design tools to help fill the foundry.

Thanks to its Caltech and UC Berkeley students, VLSI was an important pioneer in the electronic design automation industry. It offered a sophisticated package of tools, originally based on the 'lambda-based' design style advocated by Carver Mead and Lynn Conway. VLSI became an early vendor of standard cell (cell-based technology) to the merchant market in the early 80s where the other ASIC-focused company, LSI Logic, was a leader in gate arrays. Prior to VLSI's cell-based offering, the technology had been primarily available only within large vertically integrated companies with semiconductor units such as AT&T and IBM.

VLSI's design tools eventually included not only design entry and simulation but eventually cell-based routing (chip compiler), a data-path compiler, SRAM and ROM compilers and a state machine compiler. The tools were an integrated design solution for IC design and not just point tools, or more general purpose system tools. A designer could edit transistor-level polygons and/or logic schematics, then run DRC and LVS, extract parasites from the layout and run Spice simulation, then back-annotate the timing or gate size changes into the logic schematic database. Characterization tools were integrated to generate Frame Maker Data Sheets for Libraries. VLSI eventually spun off the CAD and Library operation into Compass Design Automation but it never reached IPO before it was purchased by Avanti Corp.

VLSI's physical design tools were critical not only to its ASIC business, but also in setting the bar for the commercial EDA industry. When VLSI and its main ASIC competitor, LSI Logic, were establishing the ASIC industry, commercially-available tools could not deliver the productivity necessary to support the physical design of hundreds of ASIC designs each year without the deployment of a substantial number of layout engineers. The companies' development of automated layout tools was a rational "make because there's nothing to buy" decision. The EDA industry finally caught up in the late 1980s when Tangent Systems released its Tan-Cell and Tan-Gate products. In 1989, Tangent was acquired by Cadence Design Systems (founded in 1988).

Unfortunately, for all VLSI's initial competence in design tools, they were not leaders in semiconductor manufacturing technology. VLSI had not been timely in developing a 1.0 μm

manufacturing process as the rest of the industry moved to that geometry in the late 80s. VLSI entered a long-term technology partnership with Hitachi and finally released a 1.0 μm process and cell library (actually more of a 1.2 μm library with a 1.0 μm gate).

As VLSI struggled to gain parity with the rest of the industry in semiconductor technology, the design flow was moving rapidly to a Verilog HDL and synthesis flow. Cadence acquired Gateway, the leader in Verilog hardware design language (HDL) and Synopsys was dominating the exploding field of design synthesis. As VLSI's tools were being eclipsed, VLSI waited too long to open the tools up to other fabrications and Compass Design Automation was never a viable competitor to industry leaders. Meanwhile, VLSI entered the merchant high speed static RAM (SRAM) market as they needed a product to drive the semiconductor process technology development. All the large semiconductor companies built high speed SRAMs with cost structures VLSI could never match. VLSI withdrew once it was clear that the Hitachi process technology partnership was working.

ARM Ltd was formed in 1990 as a semiconductor intellectual property licensor, backed by Acorn, Apple and VLSI. VLSI became a licensee of the powerful ARM processor and ARM finally funded processor tools. Initial adoption of the ARM processor was slow. Few applications could justify the overhead of an embedded 32 bit processor. In fact, despite the addition of further licensees, the ARM processor enjoyed little market success until they developed the novel 'thumb' extensions. Ericsson adopted the ARM processor in a VLSI chipset for its GSM handset designs in the early 1990s. It was the GSM boost that is the foundation of ARM the company/technology that it is today.

Only in PC chipsets, did VLSI dominate in the early 90s. This product was developed by five engineers using the 'Mega cells' in the VLSI library that led to a business unit at VLSI that almost equaled its ASIC business in revenue. VLSI eventually ceded the market to Intel because Intel was able to package-sell its processors, chipsets, and even board level products together. VLSI also had an early partnership with PMC, a design group that had been nurtured of British Columbia Bell. When PMC wanted to divest its semiconductor intellectual property venture, VLSI's bid was beaten by a creative deal by Sierra Semiconductor. The telecom business unit management at VLSI opted to go it alone. PMC Sierra became one of the most important telecom ASSP vendors.

Scientists and innovations from the 'design technology' part of VLSI found their way to Cadence Design Systems (by way of Redwood Design Automation). Compass Design Automation (VLSI's CAD and Library spin-off) was sold to Avant! Corporation, which itself was acquired by Synopsys.

1.3 STRUCTURED DESIGN:

Structured VLSI design is a modular methodology originated by Carver Mead and Lynn Conway for saving microchip area by minimizing the interconnect fabrics area. This is obtained by repetitive arrangement of rectangular macro blocks which can be interconnected using wiring by abutment. An example is partitioning the layout of an adder into a row of equal bit slices cells. In complex designs this structuring may be achieved by hierarchical nesting.

Structured VLSI design had been popular in the early 1980s, but lost its popularity later because of the advent of placement and routing tools wasting a lot of area by routing, which is tolerated because of the progress of Moore's Law. When introducing the hardware description language KARL in the mid' 1970s, Reiner Hartenstein coined the term "structured VLSI design" (originally as "structured LSI design"), echoing Edsger Dijkstra's structured programming approach by procedure nesting to avoid chaotic spaghetti-structured programs.

1.4 WHAT IS VLSI?

VLSI stands for "Very Large Scale Integration". This is the field which involves packing more and more logic devices into smaller and smaller areas.

- Simply we say Integrated circuit is many transistors on one chip.
- Design/manufacturing of extremely small, complex circuitry using modified semiconductor material
- Integrated circuit (IC) may contain millions of transistors, each a few mm in size
- Applications wide ranging: most electronic logic devices

In olden days, when huge computers made of vacuum tubes could occupy an entire dedicated rooms and could do about 360 multiplications of 10 digit numbers in a second. Modern day computers are getting smaller, faster, and cheaper and more power efficient for every progressing second. The electronic miniaturizing started when the occurrence of semiconductor transistor by Bardeen (1947-48) and then the Bipolar Transistor by Shockley (1949) in the Bell Laboratory.

The first IC (Integrated Circuit) was invented by Jack Kilby in 1958, in the form of a Flip Flop our ability to pack more and more transistors onto a single chip has doubled roughly every 18 months, in accordance with the Moore's Law. Such exponential or increasing development had

never been seen in any other field and still it is continuing in major areas of research work.

1.5 HISTORY & EVOLUTION:

The development of microelectronics spans a time which is even lesser than the average life expectancy of a human, and yet it has seen as many as four generations. In early 60's, the low density fabrication processes classified under Small Scale Integration (SSI) in which transistor count was limited to about 10. This rapidly gave way to Medium Scale Integration in the late 60's when around 100 transistors could be placed on a single chip. It was the time when the cost of research began to decline and private firms started entering the competition. Transistor-Transistor logic (TTL) offering higher integration densities than other IC families like ECL and became the basis of the first integrated circuit revolution. Early seventies marked the growth of transistor count to about 1000 per chip called the Large Scale Integration. By mid-eighties, the transistor count on a single chip had already exceeded 1000 and hence came the age of Very Large Scale Integration or VLSI. Though many improvements have been made and the transistor count is still rising, further names of generations like ULSI are generally avoided.

The second age of Integrated Circuits revolution started with the introduction of the first microprocessor, the 4004 by Intel in 1972 and the 8080 in 1974. Devices, Intel, Philips, Motorola and many other firms have been established and are dedicated to the various fields in "VLSI" like Programmable Logic Devices, Hardware Descriptive Languages, Design tools, Embedded Systems etc.

1.6 FUTURE OF VLSI:

Generally, VLSI technology is used in the devices like computers, cell phones, digital cameras and any electronic gadget. There are certain key issues that serve as active areas of research and are constantly improving as the field continues to mature. VLSI is dominated by the CMOS technology and much like other logic families, this too has its limitations which have been battled and improved upon since years. By taking the example of a processor, the process technology has rapidly shrunk from 180 nm in 1999 to 60nm in 2008 and now it stands at 45nm and attempts are being made to reduce it for 32nm. As the number of transistors increase, the power dissipation is

increasing and also the noise. Heat is generated per unit area. New alternatives like Gallium Arsenide technology are becoming an active area of research, future of VLSI seems to change for every little moment.

History of Scale Integration

- Late 40s Transistor invented at Bell Labs
- Late 50s First IC (JK-FF by Jack Kilby at TI)
- Early 60s Small Scale Integration (SSI)
- 10s of transistors on a chip
- Late 60s Medium Scale Integration (MSI)
- 100s of transistors on a chip
- Early 70s Large Scale Integration (LSI)
- 1000s of transistor on a chip
- Early 80s VLSI 10,000s of transistors on a Chip (later 100,000s & now 1,000,000s)
- Ultra LSI is sometimes used for 1,000,000s
- SSI - Small-Scale Integration (0-102)
- MSI - Medium-Scale Integration (102-103)
- LSI - Large-Scale Integration (103-105)
- VLSI - Very Large-Scale Integration (105-107)

ULSI - Ultra Large-Scale Integration (≥ 107)

1.6.1 System Design:

Create a high-level (Behavioral) representation of your system

Tools: Verilog, VHDL, System C

- Synthesizable (PLD's and/or ASIC)
- Non-synthesizable
- Moraine future lectures

1.6.2 Switches:

Digital equipment is largely composed of switches are

1. Switches can be built from many technologies
2. Relays (from which the earliest computers were built)

3. Thermionic valves

4. Transistors

The perfect digital switch would have the following

1. Switch instantly
2. Use no power
3. Have an infinite resistance when off and zero resistance when on

1.7 Semiconductors and Doping:

By adding trace amounts of certain materials to semiconductors alters the crystal structure and can change their electrical properties in particular it can change the number of free electrons or holes. N-Type semiconductor has free electrons and dopant is (typically) phosphorus, arsenic, antimony. P-Type semiconductor has free holes dopant is (typically) boron, indium, and gallium. Dopants are usually implanted into the semiconductor using Implant Technology, followed by thermal process to diffuse the dopants. IC Technology is mainly used for Speed / Power performance of available technologies and the microelectronics evolution and SIA Roadmap and Semiconductor Manufacturers 2001 Ranking.

Metal-oxide-semiconductor (MOS) and related VLSI technology
PMOS, NMOS, CMOS, Bi CMOS, GaAs.

Basic MOS Transistors are implemented as minimum line width, transistor cross section, Charge inversion channel, Source connected to substrate, Enhancement vs. Depletion mode devices, pMOS are 2.5 time slower than nMOS due to electron and hole motilities.

1.7.1 Fabrication Technology:

It is Silicon of extremely high purity and chemically purified then grown into large crystals. Wafer is type of crystal which is sliced into substrates, and its diameter is currently 150mm, 200mm, 300mm and wafer thickness <1mm and also surface is polished to optical smoothness. Wafer is then ready for processing. Each wafer will yield many chips and the chip die size varies from about 5mmx5mm to 15mmx15mm. A whole wafer is processed at a time. Different parts of each die will be made P-type or N-type (small amount of other atoms intentionally introduced - doping -implant). Interconnections are made with metal insulation and material used is typically SiO₂, SiN. New materials are being investigated (low-k dielectrics).

In CMOS Fabrication p-well process, n-well process and twin-tub process all the devices on

the wafer are made at the same time. After the circuitry has been placed on the chip, the chip is over-glassed (with a passivation layer) to protect it only those areas which connect to the outside world will be left uncovered (the pads). The wafer finally passes to a test station test probes send test signal patterns to the chip and monitor the output of the chip. The yield of a process is the percentage of die which pass this testing, the wafer is then scribed and separated up into the individual chips. These are then packaged and Chips are 'binned' according to their performance.

1.7.2 Conventional Approach to Digital Design

Digital ICs of SSI and MSI types have become universally standardized and have been accepted for use. Whenever a designer has to realize a digital function, he uses a standard set of ICs along with a minimal set of additional discrete circuitry.

1.7.3 Design of VLSI

The complexity of VLSI is being designed and used today, which makes the manual approach to be impractical. Design automation is the order of the day. With the rapid technological developments in the last two decades, the status of VLSI technology is characterized by the following

1. A steady increase in the size and hence the functionality of the ICs.
2. A steady reduction in feature size and hence increase in the speed of operation as well as gate or transistor density.
3. A steady improvement in the predictability of circuit behavior.
4. A steady increase in the variety and size of software tools for VLSI design.

The above developments have resulted in a proliferation of approaches to VLSI design. We briefly describe the procedure of automated design flow. The aim is more to bring out the role of a Hardware Description Language (HDL) in the design process. An abstraction based model is the basis of the automated design.

The model divides the whole design cycle into various domains. With such an abstraction

through a division process the design is carried out in different layers. The designer at one layer can function without bothering about the layers above or below. The thick horizontal lines separating the layers in the figure signify the compartmentalization. As an example, let us consider design at the gate level. The circuit to be designed would be described in terms of truth-tables and state tables. With these as available inputs, he has to express them as Boolean logic equations and realize them in terms of gates and flip-flops. In turn, these form the inputs to the layer immediately below. Compartmentalization of the approach to design in the manner described here is the essence of abstraction; it is the basis for development and use of CAD tools in VLSI design at various levels.

The design methods at different levels use the respective aids such as Boolean equations, truth tables, state transition table, etc. But the aids play only a small role in the process. To complete a design, one may have to switch from one tool to another, raising the issues of tool compatibility and learning new environments.

1.8 VLSI AND SYSTEMS:

These advantages of integrated circuits translate into advantages at the system level are

- **Smaller physical size:** Smallness is often an advantage in itself—consider portable televisions or handheld cellular telephones.
- **Lower power consumption:** Replacing a handful of standard parts with a single chip reduces total power consumption. Reducing power consumption has a ripple effect on the rest of the system: a smaller, cheaper power supply can be used; since less power consumption means less heat, a fan may no longer be necessary; a simpler cabinet with less shielding for electromagnetic shielding may be feasible, too.
- **Reduced cost:** Reducing the number of components, the power supply requirements, cabinet costs, and so on, will inevitably reduce system cost. The ripple effect of integration is such that the cost of a system built from custom ICs can be less, even though the individual ICs cost more than the standard parts they replace.

Understanding why integrated circuit technology has such profound influence on the design of digital systems requires understanding both the technology of IC manufacturing and the economics of ICs and digital systems.

1.9 APPLICATIONS OF VLSI

VLSI having applications in various domains such as electronic systems, medical, communication, digital signal processing. Some of them are listed as below

1.9.1 Electronic system in cars:

Electronic systems now perform a wide variety of tasks in daily life. Electronic systems in some cases have replaced mechanisms that operated mechanically, hydraulically, or by other means; electronics are usually smaller, more flexible, and easier to service. In other cases electronic systems have created totally new applications. Electronic systems perform a variety of tasks, some of them are visible while some are hidden. Personal entertainment systems such as portable MP3 players and DVD players perform sophisticated algorithms with remarkably little energy.

1.9.2 Digital electronics control VCRs:

Digital electronics compress and decompress video, even at high-definition data rates, on-the-fly in consumer electronics. Low-cost terminals for Web browsing still require sophisticated electronics, despite their dedicated function.

1.9.3 Transaction processing system, ATM :

Electronic systems in cars operate stereo systems and displays; they also control fuel injection systems, adjust suspensions to varying terrain, and perform the control functions required for anti-lock braking systems.

1.9.4 Personal computers and Workstations :

Personal computers and workstations provide word-processing, financial analysis, and games. Computers include both central processing units and special-purpose hardware for disk access, faster screen display, etc.

1.9.5 Medical electronic systems:

Medical electronic systems measure bodily functions and perform complex processing algorithms to warn about unusual conditions. The availability of these complex systems, far from overwhelming consumers, only creates demand for even more complex systems.

1.20 ASIC

An Application-Specific Integrated Circuit (ASIC) is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use. For example, a chip designed solely to run a cell phone is an ASIC. Intermediate between ASICs and industry standard integrated circuits, like the 7400 or the 4000 series, are application specific standard products (ASSPs).

As feature sizes have shrunk and design tools improved over the years, the maximum complexity (and hence functionality) possible in an ASIC has grown from 5,000 gates to over 100 million. Modern ASICs often include entire 32-bit processors, memory blocks including ROM, RAM, EEPROM, Flash and other large building blocks. Such an ASIC is often termed a SOC (system-on-a-chip). Designers of digital ASICs use a hardware description language (HDL), such as Verilog or VHDL, to describe the functionality of ASICs.

Field-programmable gate arrays (FPGA) are the modern-day technology for building a breadboard or prototype from standard parts; programmable logic blocks and programmable interconnects allow the same FPGA to be used in many different applications. For smaller designs and/or lower production volumes, FPGAs may be more cost effective than an ASIC design even in production.

1.20.1 ASIC DESIGN FLOW:

As with any other technical activity, development of an ASIC starts with an idea and takes tangible shape through the stages of development as shown in Fig 1 and in Fig 2. The first step in this process is to expand the idea in terms of behavior of the target circuit.

The design is tested through a simulation process; it is to check, verify, and ensure that what is wanted is what is described. Simulation is carried out through dedicated tools. With every simulation run, the simulation results are studied to identify errors in the design description. The errors are corrected and another simulation run carried out. Simulation and changes to design description together form a cyclic iterative process, repeated until an error-free design is evolved.

Design description is an activity independent of the target technology or manufacturer. It results in a description of the digital circuit. To translate it into a tangible circuit, one goes through the physical design process. The same constitutes a set of activities closely linked to the manufacturer and the target technology.

CHAPTER-2

LITERATURE SURVEY

A Single-Ended With Dynamic Feedback Control 8T Subthreshold SRAM Cell

[C. B. Kushwah](#); [S. K. Vishvakarma](#) 26 January 2015 A novel 8-transistor (8T) static random access memory cell with improved data stability in subthreshold operation is designed. The proposed single-ended with dynamic feedback control 8T static RAM (SRAM) cell enhances the static noise margin (SNM) for ultralow power supply. It achieves write SNM of 1.4× and 1.28× as that of isoarea 6T and read-decoupled 8T (RD-8T), respectively, at 300 mV.

A Single-Ended with Dynamic Feedback Control 8T Subthreshold SRAM Cell

R.Swathi, T.Bhavani, Mr.Devireddy Venkatarami Reddy 2017 The Static Random Access Memory (SRAM) has become a major component in many VLSI Chips due to their large storage density and small access time. SRAM has become the topic of substantial research due to the rapid development for low power. SRAM plays a most substantial role in the microprocessor world, but as the technology is scaling down in Nano meters

A Single-Ended with Dynamic Feedback Control 10T Subthreshold SRAM Cell

Motla Sushma 2 Mr.B.Kishore Kumar This paper implements the design of Single ended dynamic feedback control 10T sub threshold cell. The 10T can operate at very low power levels such as ultra low power ranges. The switching speed of 10T is very high as compared to the proposed method 8T cell. The proposed single-ended with dynamic feedback control 8T static RAM (SRAM) cell enhances the static noise margin (SNM) for ultralow power supply. The proposed 8T cell has less leakage current when compared to the 6T, 5T, RD-8T. The designs 6T, 5T, RD-8T, 8T and 10T are implemented at different voltage levels by using Tanner-Tool in 90nm technology and compared.

Low Leakage SRAM Cell with Improved Stability for IoT Applications Chusen Duari*, Shilpi Birla in 2020 In the present scenario, the SRAM array occupies a significant area within the chip .The memory array is responsible for the consumption of a major amount of power of the total power consumed by the chip .The primary component of consumed power in SRAM cell based with submicron MOSFET is leakage power, which is mainly consisting of the sub-threshold current component

CHAPTER-3

EXISTING SYSTEM

The portable microprocessor controlled devices contain embedded memory, which represents a large portion of the system-on-chip (SOC). These portable systems need ultralow power consuming circuits to utilize battery for longer duration. The power consumption can be minimized using nonconventional device structures, new circuit topologies, and optimizing the architecture. Although, voltage scaling has led to circuit operation in sub threshold regime with minimum power consumption, but there is a disadvantage of exponential reduction in performance.

The circuit operation in the sub threshold regime has paved path toward ultralow power embedded memories, mainly static RAMs (SRAMs). However, in sub threshold regime, the data stability of SRAM cell is a severe problem and worsens with the scaling of MOSFET to subnanometer technology. Due to these limitations it becomes difficult to operate the conventional 6-transistor (6T) cell at ultralow voltage (ULV) power supply.

In addition, 6T has a severe problem of read disturb. The basic and an effective way to eliminate this problem is the decoupling of true storing node from the bit lines during the read operation.

This read decoupling approach is utilized by conventional 8-transistor cell which offers read static noise margin (RSNM) comparable with hold static noise margin (HSNM). However, RD-8T suffers from leakage introduced in read path. This leakage current increases with the scaling thereby, increasing the probability of failed read/write operations. Similar cells that maintain the cell current without disturbing the storage node are also proposed.

Disadvantages:

- Low SNM
- High power consumption

CHAPTER-4

PROPOSED SYSTEM

To make a cell stable in all operations, single-ended with dynamic feedback control (SE-DFC) cell is presented in Fig. 1(a). The single-ended design is used to reduce the differential switching power during read-write operation. The power consumed during switching/ toggling of data on single bit line is lesser than that on differential bit-line pair. The SE-DFC enables writing through single n- MOS in 8T. It also separates the read and write path and exhibits read decoupling.

The structural change of cell is considered to enhance the immunity against the process-voltage-temperature (PVT) t improves the static noise margin (SNM) of 8T cell in sub threshold/near-threshold region. The proposed 8T has one cross coup led inverter pair, in which each inverter is made up of three cascaded transistors.

These two stacked cross-coup led inverters: M1-M2-M4 and M8-M6-M5 retain the data during hold mode. variations. It improves the static noise margin (SNM) of 8T cell in sub threshold/near-threshold region. The proposed 8T has one cross coup led inverter pair, in which each inverter is made up of three cascaded transistors. These two stacked cross-coup led inverters: M1-M2-M4 and M8-M6-M5 retain the data during hold mode. The write word line WWL) controls only one n-MOS transistor M7, used to transfer the data from single write bit line (WBL).

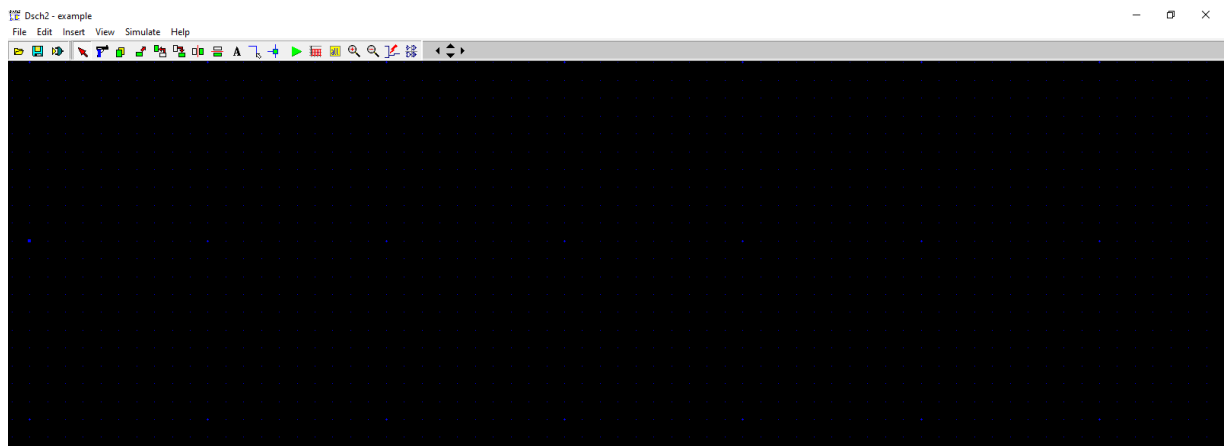
A separate read bit line (RBL) is used to transfer the data from cell to the output when read word line (RWL) is activated. Two columns biased feedback control signals: FCS1 and FCS2 lines are used to control the feedback cutting transistors: M6 and M2, respectively.

Schematic diagram

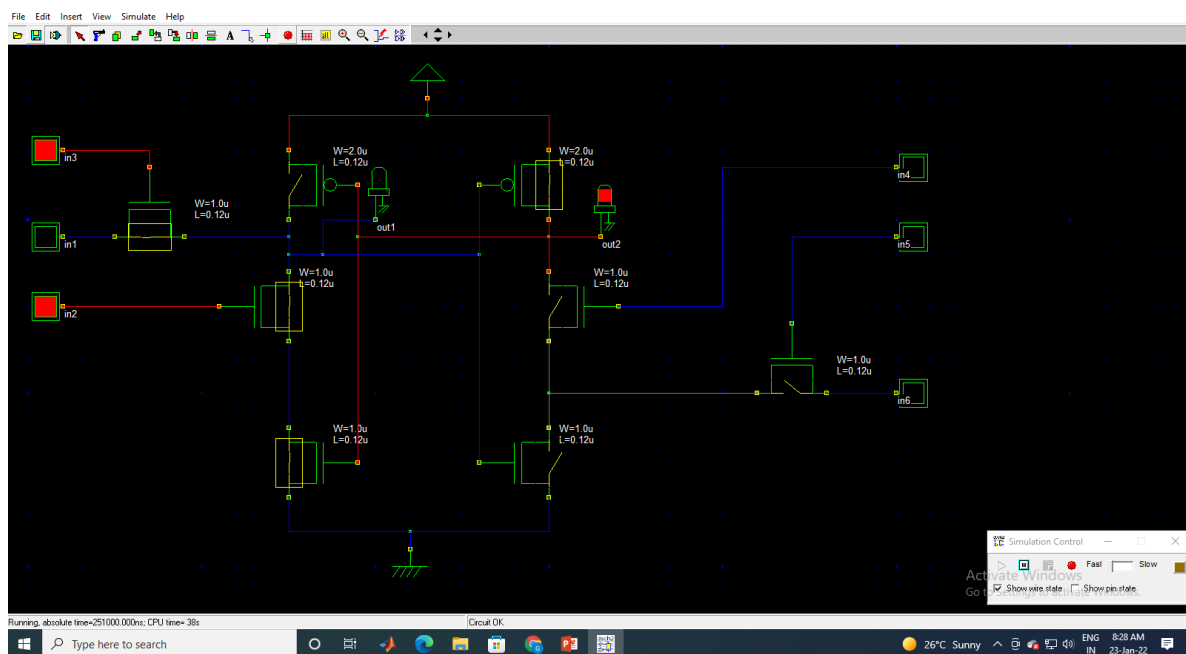
ternary gates (TNOT, TNAND, TNOR) using injected voltage method.

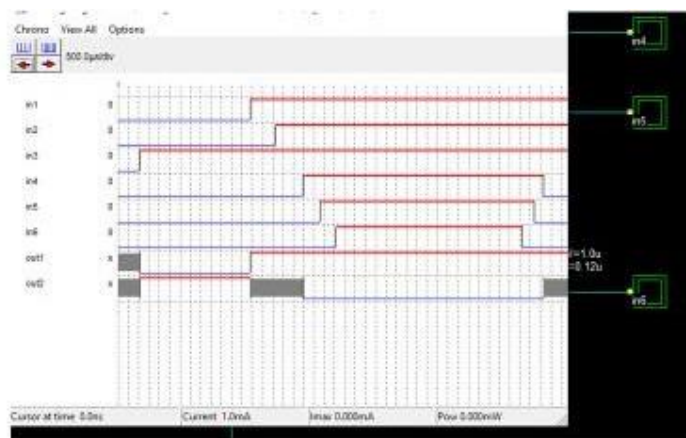
The binary CMOS logic is exploited to achieve the ternary logic values. The performance analysis of the ternary gates in terms of rise time, fall time and power dissipation is examined using Tanner Tool, version 13.02. The prominent subsets (S-Edit, L-Edit, T-Spice and W-Edit) of the tanner tool are used to derive the various device parameters and further verify the functionality of the gates. The layouts of the designed gates are also presented.

Tanner tool



Schematic diagram





CHAPTER-5

RESULT AND DISCUSSIONS

5.1 XILINX:

Xilinx, Inc. is the world's largest supplier of programmable logic devices, the inventor of the field programmable gate array (FPGA) and the first semiconductor company with a fabless manufacturing model.

Xilinx designs, develops and markets programmable logic products including integrated circuits (ICs), software design tools, predefined system functions delivered as intellectual property (IP) cores, design services, customer training, field engineering and technical support. Xilinx sells both FPGAs and CPLDs programmable logic devices for electronic equipment manufacturers in end markets such as communications, industrial, consumer, automotive and data processing.

Xilinx's FPGAs have even been used for the ALICE (A Large Ion Collider Experiment) at the CERN European laboratory on the French-Swiss border to map and disentangle the trajectories of thousands of subatomic particles.

The Vertex-II Pro, Virtex-4, Virtex-5, and Virtex-6 FPGA families are particularly focused on system-on-chip (SOC) designers because they include up to two embedded IBM PowerPC cores. The ISE Design Suite is the central electronic design automation (EDA) product family sold by Xilinx. The ISE Design Suite features include design entry and synthesis supporting Verilog or VHDL, place-and-route (PAR), completed verification and debug using Chip Scope Pro tools, and creation of the bit files that are used to configure the chip. XST-Xilinx Synthesis Technology performs device specific synthesis for Cool Runner XPLA3/-II and XC9500/XL/XV families and generates an NGC file ready for the CPLD fitter.

5.2 XILINX ISE 13.1:

Xilinx is the most important tool and in this tool we can perform both simulation and synthesis.

5.2.1 Simulation:

In this process we are going to verify our required output to get the simulation process firstly we have to implement a top module (combination of all modules) and then in the simulation behavior we can simulate the results.

5.2.2 Synthesis:

Synthesis process defines converting Verilog code into gate level which creates a net list.

5.2.3 Procedure:

- Click project navigator
- Create new project
- Selection of FPGA

Create new source

- Select source type (Verilog module)
- Coding
- Declaration of inputs and output
- Sources for implementation

Synthesize - XST

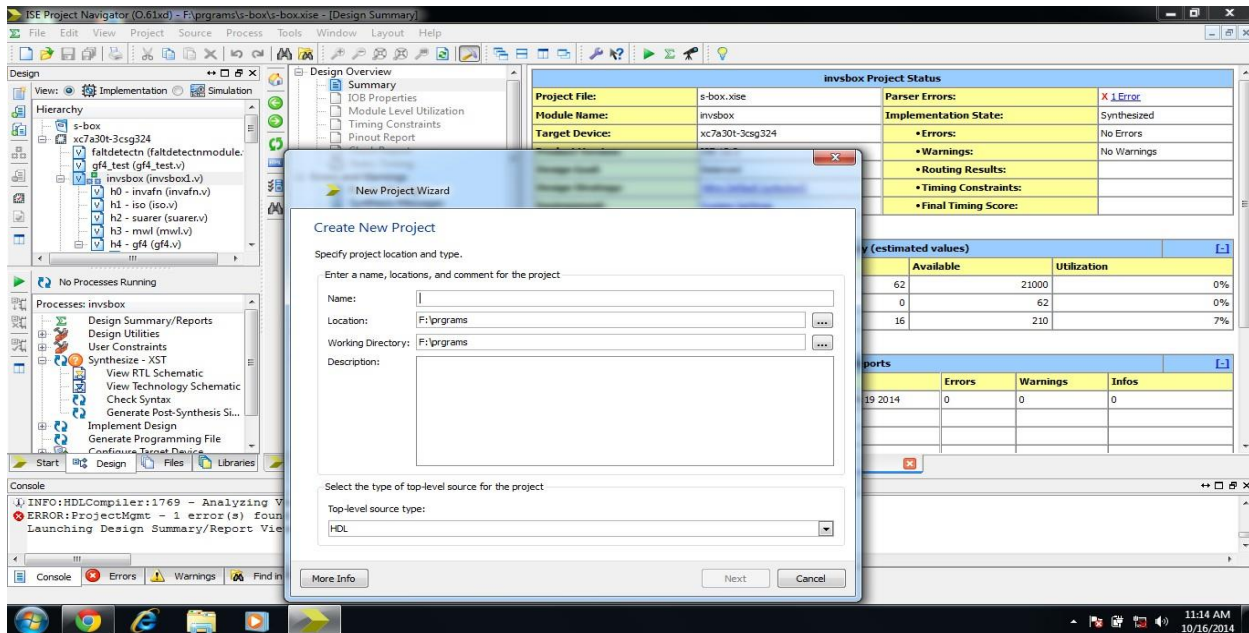
- Check syntax
- View design summary
- View RTL schematic
- View technology schematic
- Sources for behavioral simulation

Create new source

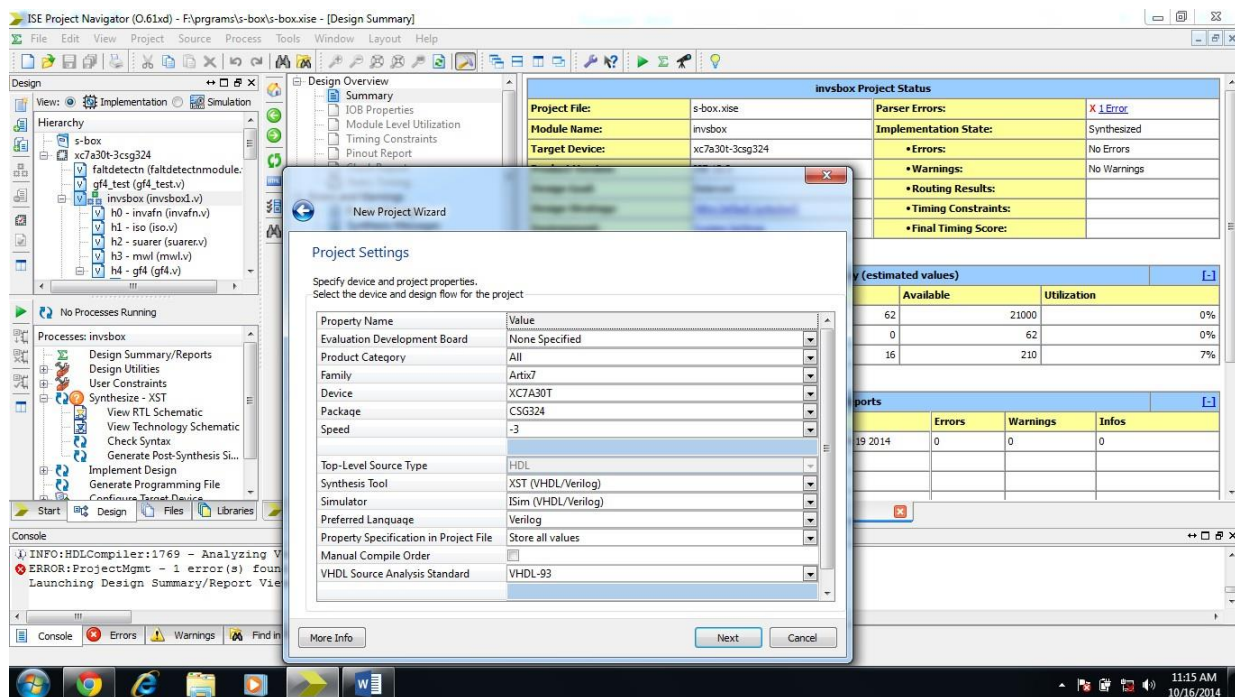
- Select source type(Verilog text fixture)
- Write test bench code
- Xilinx ISE simulator
- Behavioral check syntax

5.2.3 Xilinx snapshots:

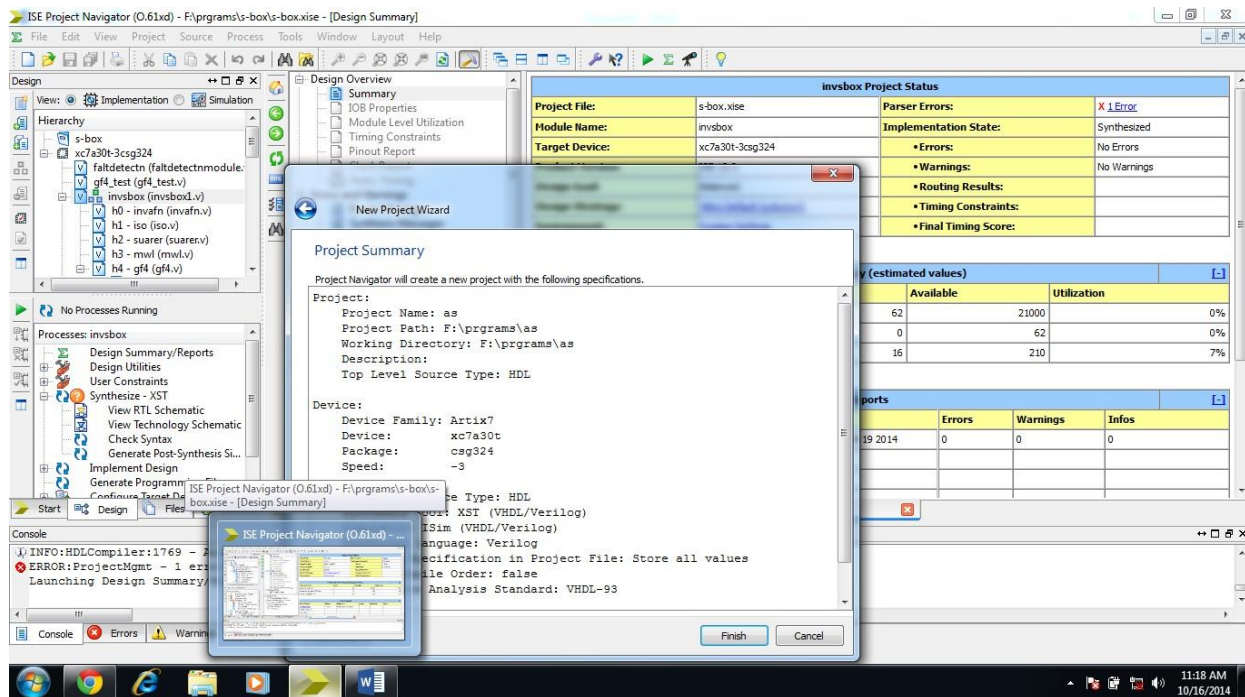
1. To create new project in xilinx we should open the file menu, click on new project then it will open the dialog box as below in that type the filename click on next



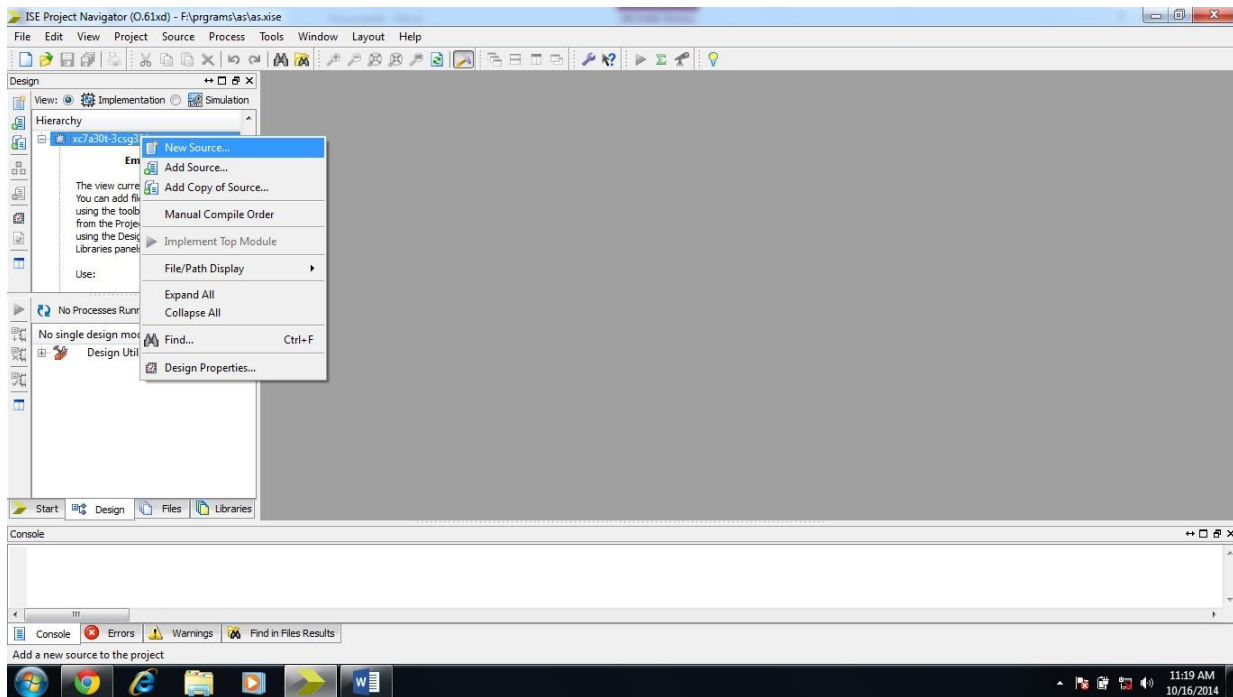
2. Then it displays one more dialog box which will give us the specifications of the project, click on next



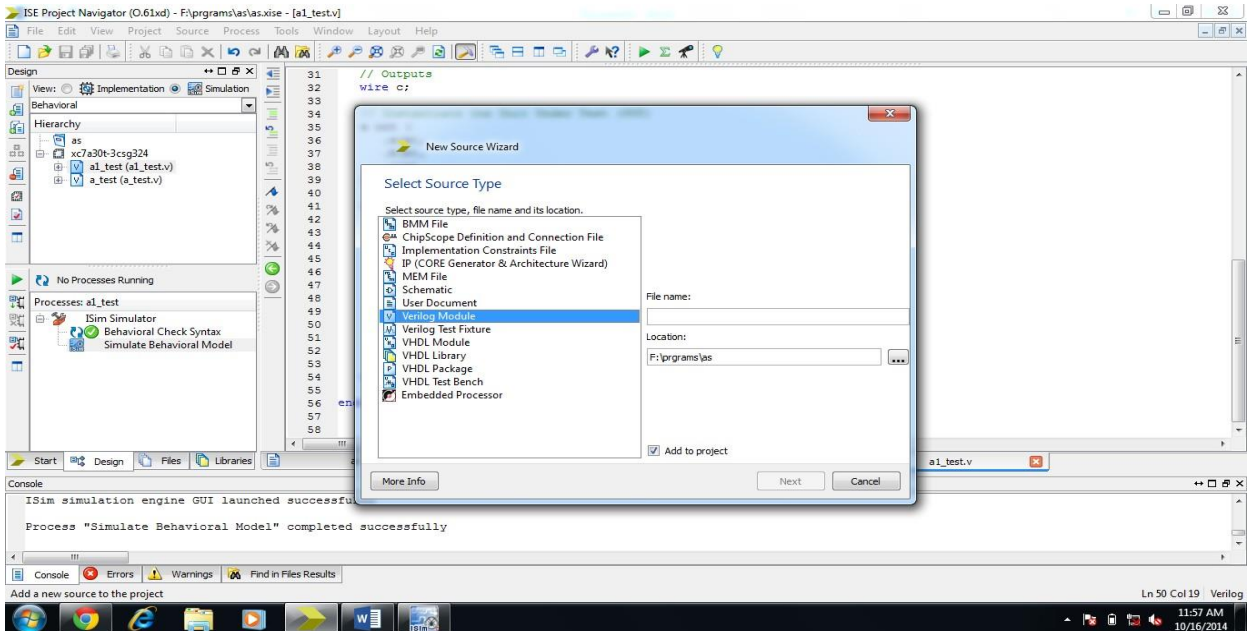
3. Then it again displays a dialogue box as shown below with the created project description and click finish to complete the process of creating new project



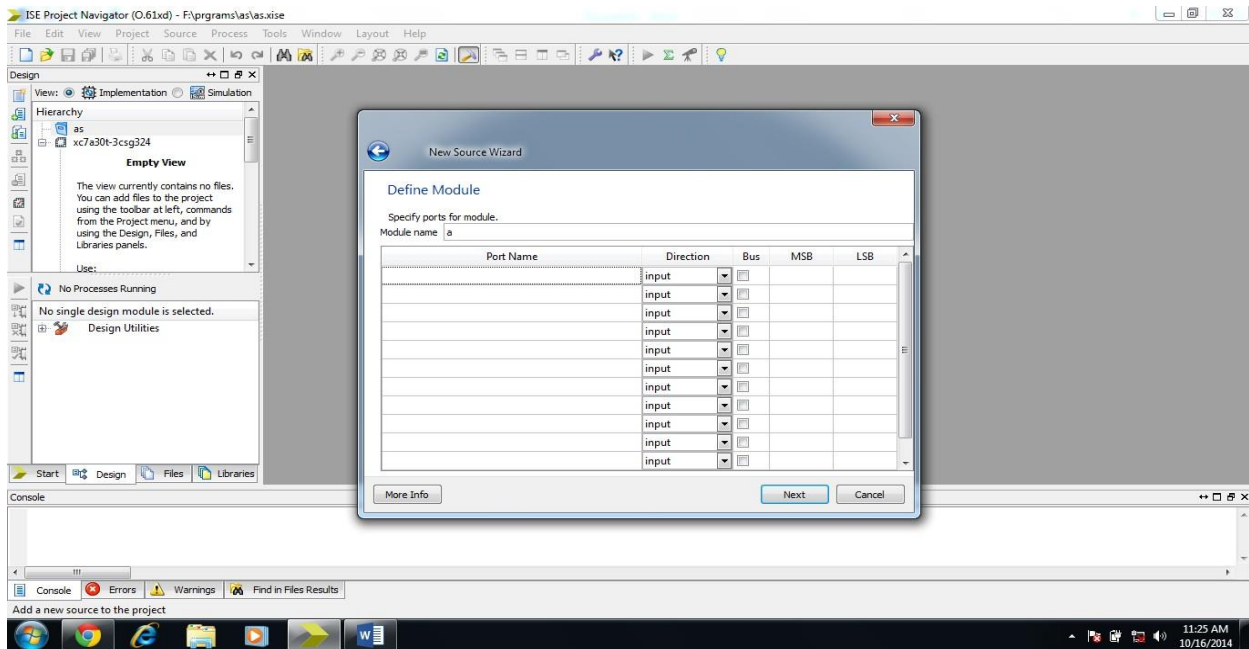
4. Now project with specified name is created then create the verilog files in the project. To create files, right click on the project that will show options like as shown below



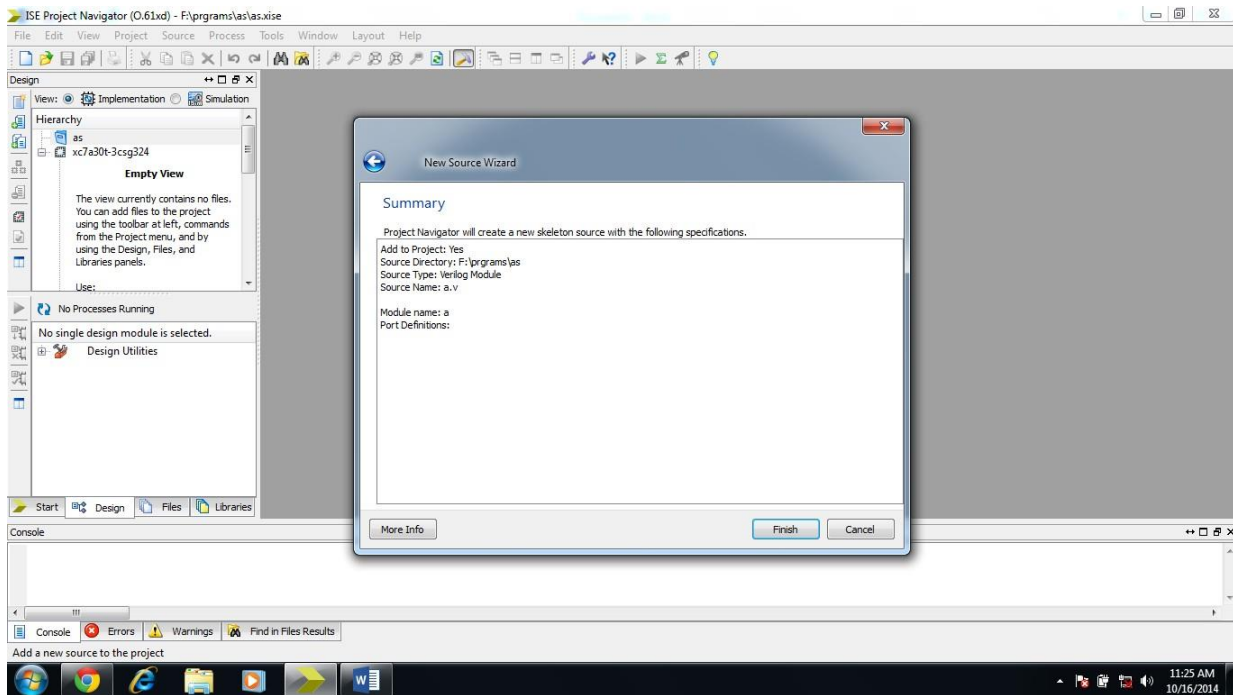
5. From the given options select new source then it displays dialogbox which is containing of list of fileformat now we want to create verlogfile so select veilog module,and give the name to the file. Then click on next



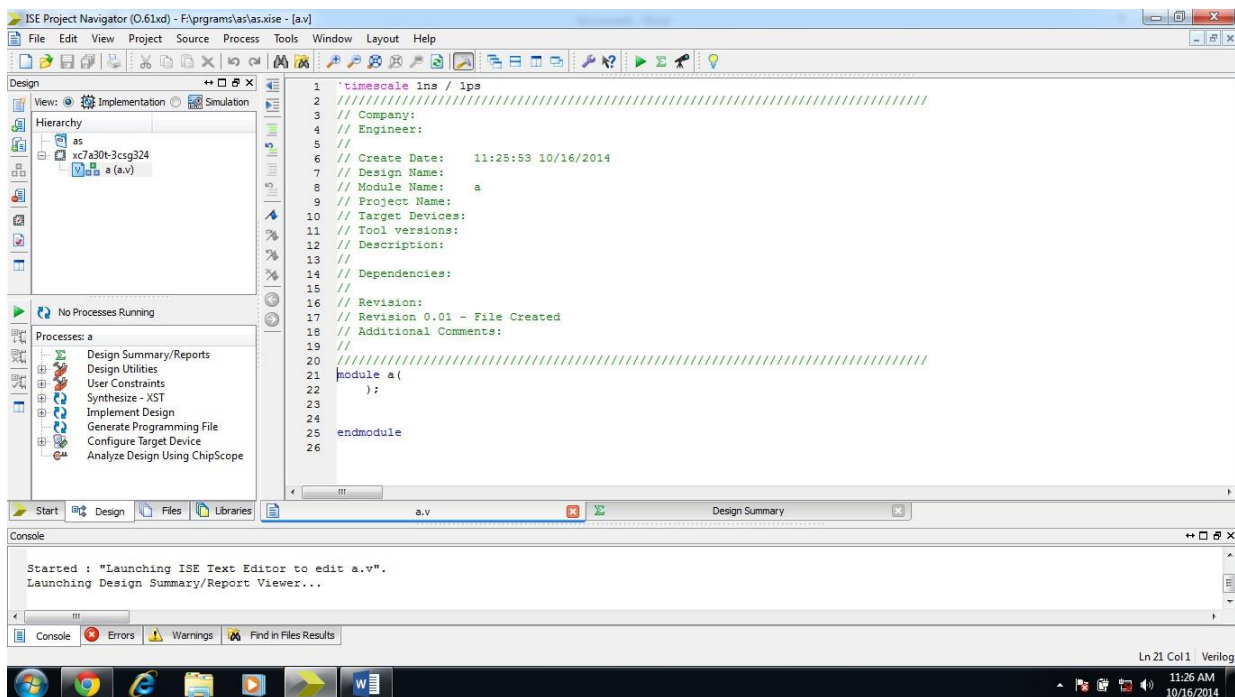
6. Then it will ask us to select inputs, outputs and inouts. We can specify our inputs and outputs here else we may also specify as part of programme depend upon the user requirement, click on next



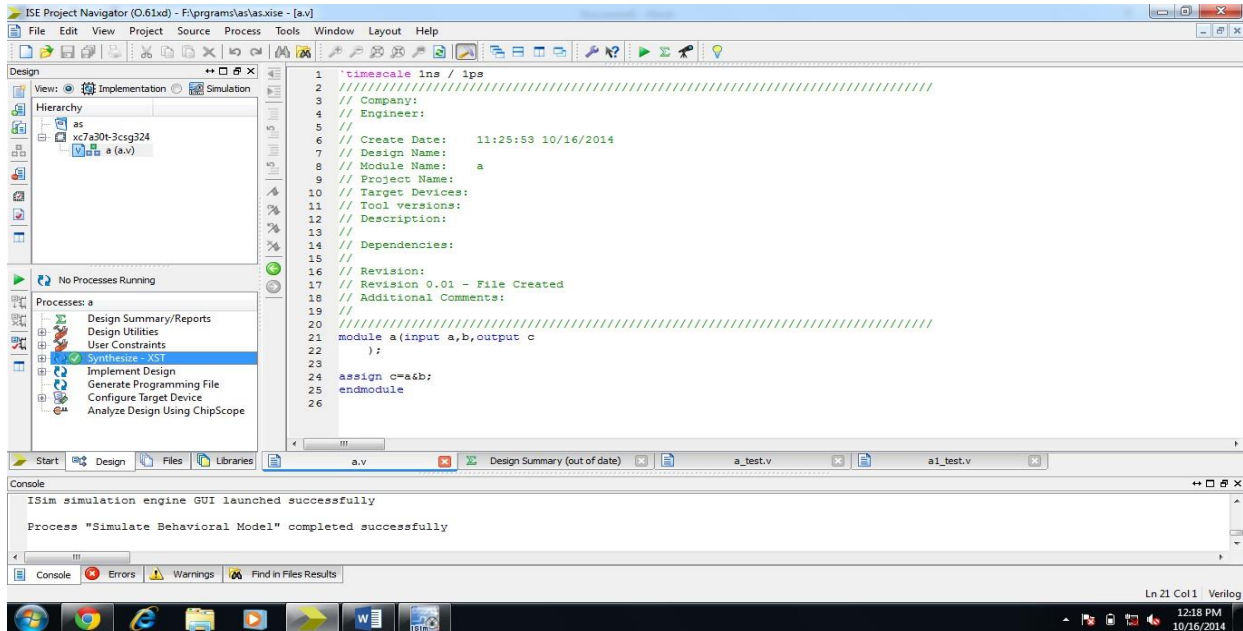
7. It will again displays a dialogbox by fiving details of filename etc, click on next



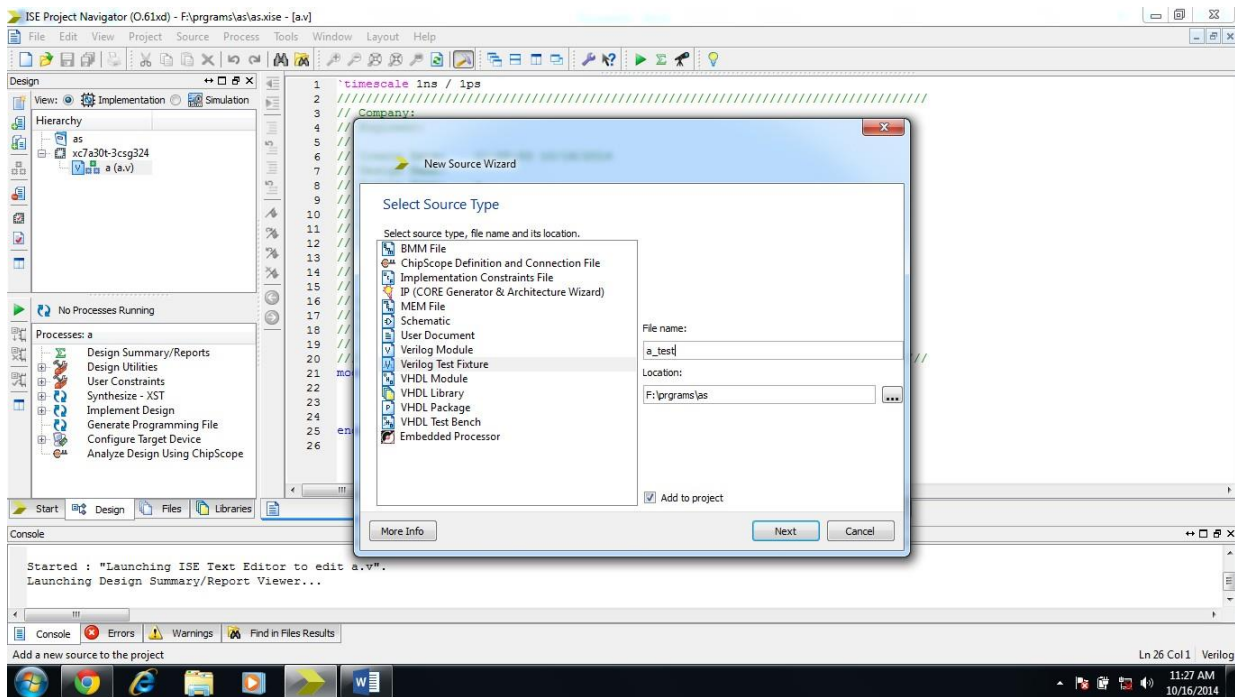
8. It will open a white space in the project window containing filename the double click on the file name so that it will displays respective file window ,where we should write the code



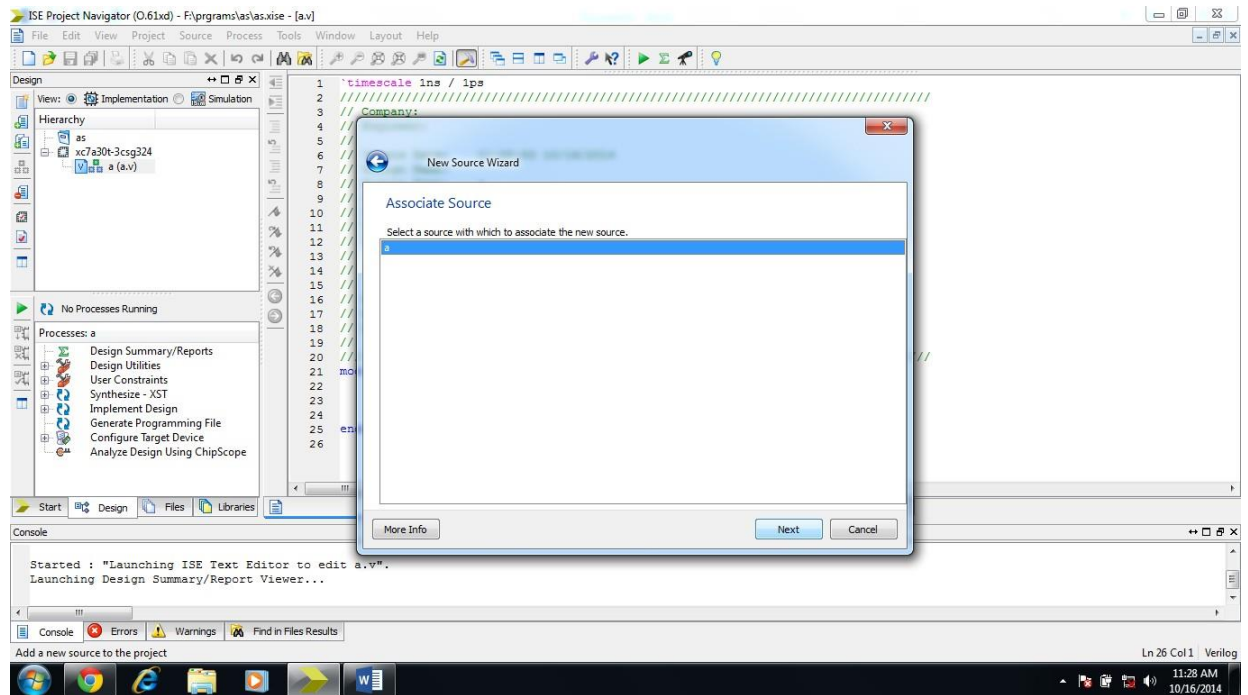
9. After completion writing code select the file name and click on synthesis which will check for errors, if there are any errors in syntax or design errors are checked and shown in the below of file window



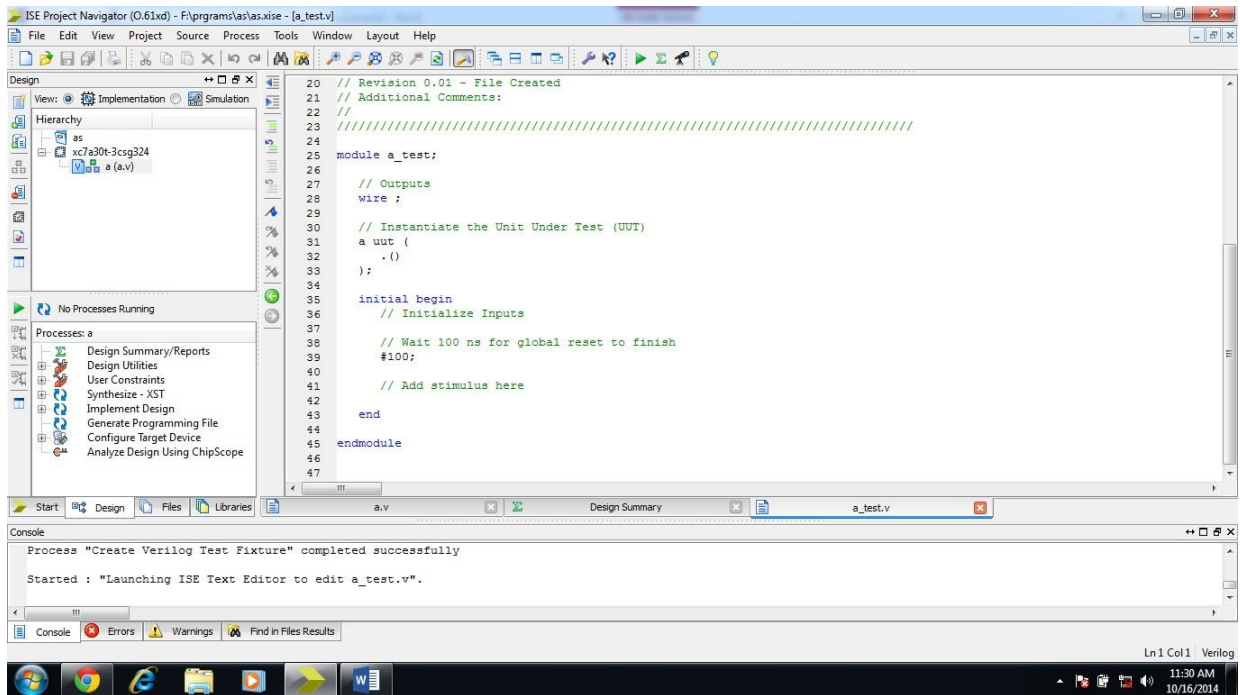
10. After successful synthesis we should have to create test bench file with extension as test, for that again right click on the file name as shown below, give filename



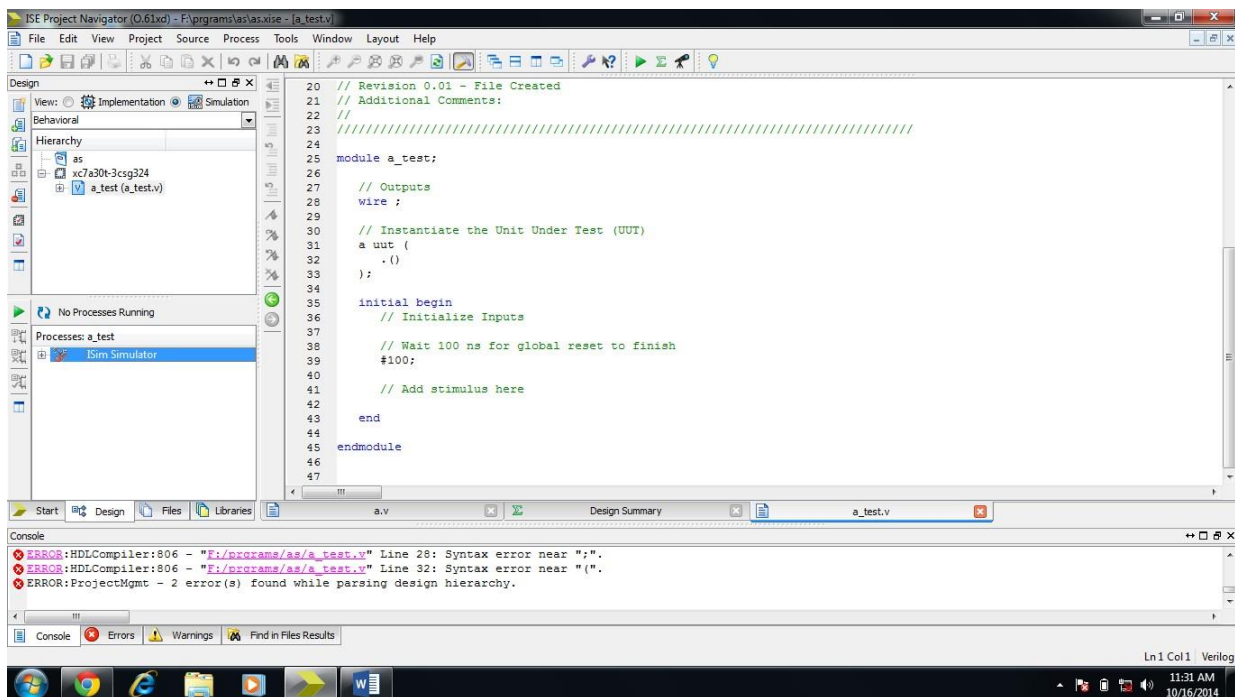
11. If there are list files then select file for which we are creating the test bench. Click on next



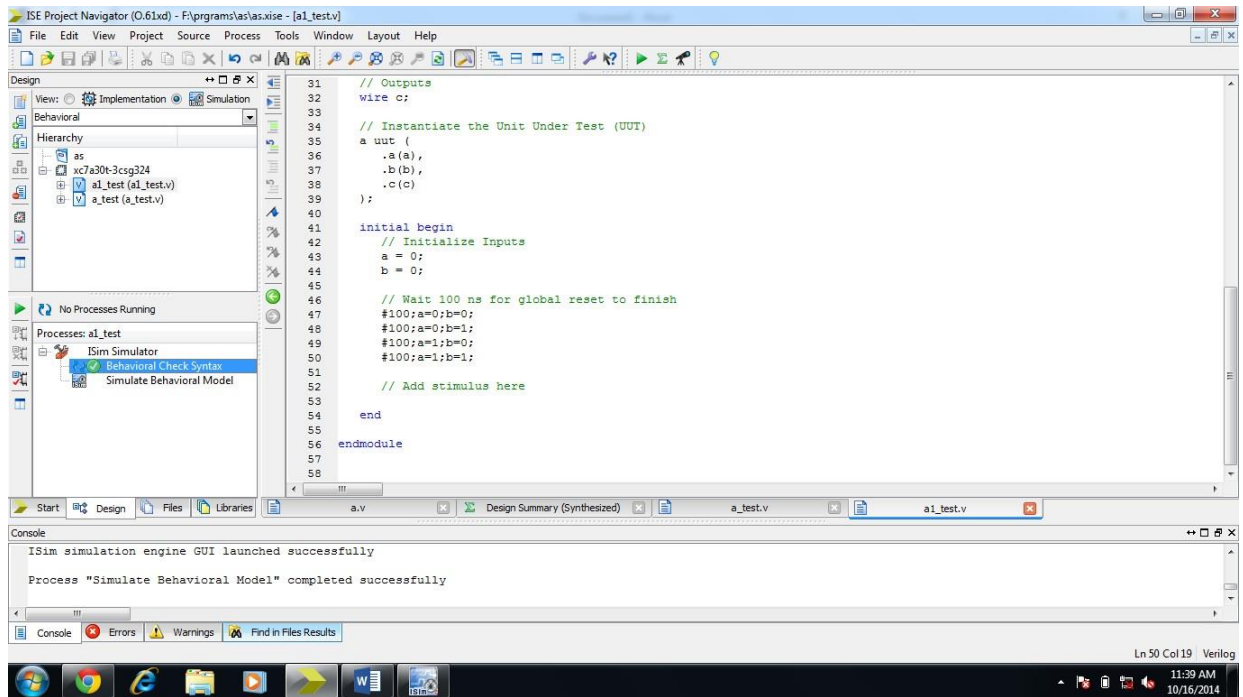
12. It again gives a testbench file in the project window, then give required inputs



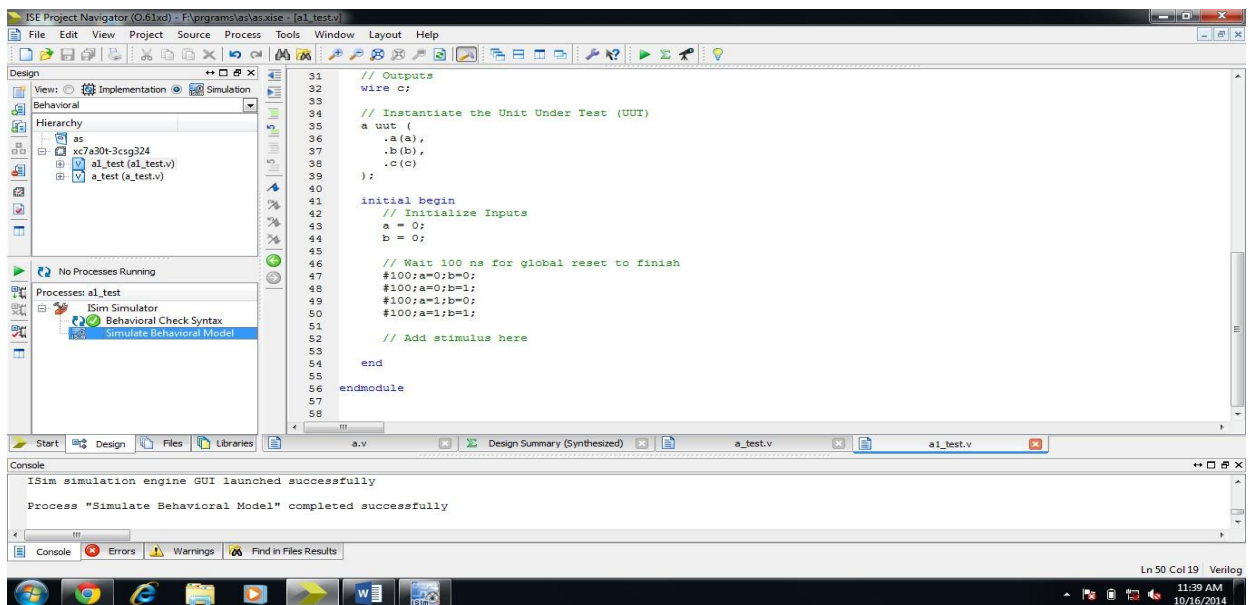
13. select simulation from the view bar in the project window above the hierarchy window as follows.



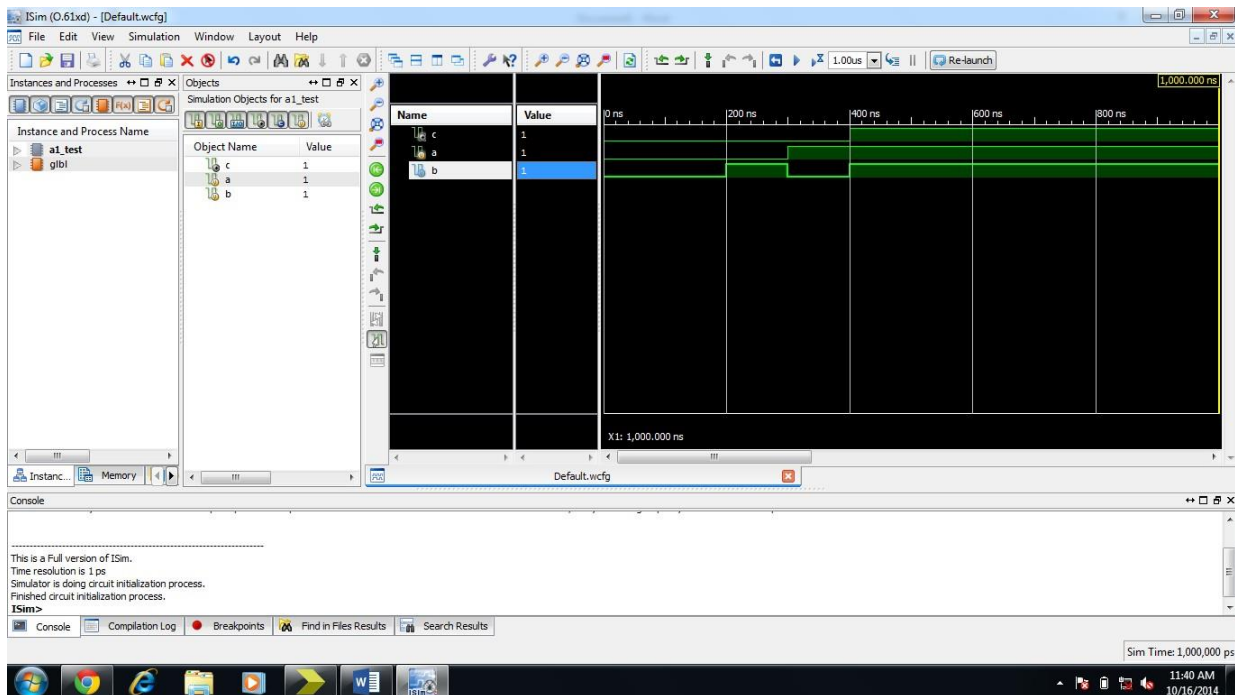
14. Double click on Isim Simulator it will expand as follows click on behavioral check syntax and it will check for syntax errors in test bench file



15. click on simulate behavioral model, it will displays wave form for in response to the inputs given in the test bench file



16. That wave form window having option to zoom out, zoom in to analyze the wave form clearly in order to understand behavior of design



5.3 LANGUAGES

5.3.1 VHDL & VERILOG

Both VHDL and Verilog are well established hardware description languages. They have the advantage that the user can define high-level algorithms and low-level optimizations (gate-level and switch-level) in the same language. A basic example of VHDL code, the evaluation of the Fibonacci series, is shown below, and it is a good example of the points made above. The code itself is reasonably straightforward for a software programmer to understand, provided that he/she understands that this is a truly parallel language and all lines are executing “at once”. It is also straightforward to simulate a simple design of this nature. However it is surprisingly difficult to implement it in hardware and this difficulty is a direct result of I/O issues. As noted above for a design to work in hardware access is required to resources that are external to the FPGA, such as memory, and an FPGA is, by its very nature, unaware of the components to which it is connected. If you want to retrieve a value from main memory and use it on the FPGA then you need to instantiate a memory controller. While systems such as the Cray XD1 provide cores for communicating with memory, such cores are still complex and unfamiliar to software programmers. Our early experiences with VHDL have indicated that it should only be used for FPGA development if you are in a position to work closely with experienced hardware designers throughout the development process.

Verilog HDL is one of the two most common Hardware Description Languages (HDL) used by integrated circuit (IC) designers. The other one is VHDL. HDL's allows the design to be simulated earlier in the design cycle in order to correct errors or experiment with different architectures. Designs described in HDL are technology-independent, easy to design and debug, and are usually more readable than schematics, particularly for large circuits. Verilog can be used to describe designs at four levels of abstraction:

- Algorithmic level (much like c code with if, case and loop statements).
- Register transfer level (RTL uses registers connected by Boolean equations).
- Gate level (interconnected AND, NOR etc.)
- Switch level (the switches are MOS transistors inside gates).

The language also defines constructs that can be used to control the input and output of simulation. More recently Verilog is used as an input for synthesis programs which will generate a gate-level description (a netlist) for the circuit. Some Verilog constructs are not synthesizable. Also the way the code is written will greatly affect the size and speed of the synthesized circuit. Most readers will want to synthesize their circuits, so no synthesizable constructs should be used only for test benches. These are program modules used to generate I/O needed to simulate the rest of the design. The words “not synthesizable” will be used for examples and constructs as needed that do not synthesize.

Verilog (released as IEEE 1364-2001) adds many significant enhancements to the Verilog language, which add greater support for configurable IP modeling and deep-submicron accuracy, and development of design management. Other enhancements make Verilog easier to use. These changes will affect everyone who uses the Verilog language, as well as those who implement Verilog software tools. This paper will review and highlight the main features added to the Verilog standard for the IEEE 1364-2001 update. The focus will be on new simulation and synthesis constructs. Where possible, status regarding Synopsys support for the new features will also be noted.

5.3.2 History of the IEEE 1364 Verilog standard:

The Verilog Hardware Description Language was first introduced in 1984, as a proprietary language from Gateway Design Automation. The original Verilog language was designed to be used with a single product, the Gateway Verilog-XL digital logic simulator. In 1989, Gateway Design Automation was acquired by Cadence Design Systems. In 1990, Cadence released the Verilog Hardware Description Language and the Verilog Programming Language Interface (PLI) to the public domain. Open Verilog International (OVI) was formed to control the public domain Verilog, and to promote its usage. Cadence turned over to OVI the Frame Maker source files containing most, but not all, of the Cadence Verilog-XL user's manual. This document became OVI's Verilog 1.0 Reference Manual. In 1993, OVI released its Verilog 2.0 Reference Manual, which contained a few enhancements to the Verilog language, such as array of instances, and major enhancements to the Verilog PLI. OVI then submitted a request to the IEEE to formally standardize Verilog 2.0. The IEEE formed a standards working group to create the standard, and, in 1995, IEEE 1364-1995 became the official Verilog standard. It is important to note that for Verilog-1995, the IEEE standards working group did not consider any enhancements to the Verilog language. The goal was to standardize the Verilog language the way it was being used at that time. The IEEE working group also decided not to create an entirely new document for the IEEE 1364 standard. Instead, the OVI Frame Maker files were used to create the IEEE standard.

Since the origin of the OVI manual was Gateway's Verilog-XL user's manual, the IEEE 1364-1995 and IEEE 1364-2001 Verilog language reference manuals are still organized somewhat like a user's guide.

Goals for Verilog standard Work on the IEEE 1364-2001 Verilog standard began in January 1997. Three major goals were established:

- Enhance the Verilog language to help with today's deep-submicron and intellectual property modeling issues.
- Ensure that all enhancements were both useful and practical, and that simulator and synthesis
- Vendors would implement Verilog-2000 in their products. Correct any errata or ambiguities in the IEEE 1364-1995 Verilog Language Reference Manual.

5.3.3 MODELING ENHANCEMENTS:

Many enhancements improve the ease and accuracy of writing synthesizable RTL models. Other enhancements allow models to be more scalable and re-usable. With the exception of the following paragraph, only changes which add new functionality or syntax are listed here. Verilog also contains many clarifications to Verilog-1995, which do not add new functionality. Notes are added to the sub-sections indicating Synopsys support with Presto and VCS at the time this paper was completed. Since the inception of Verilog in 1984, the term "register" has been used to describe the group of variable data types in the Verilog language. "Register" is not a keyword, it is simply a name for a class of data types, namely: reg, integer, time, real, and realtime. The use of term "register" is often a source of confusion for new users of Verilog, who sometimes assume that the term implies a hardware register (flip-flops).

5.3.4 VERILOG:

Hardware description languages such as Verilog differ from software programming languages because they include ways of describing the propagation of time and signal dependencies (sensitivity). There are two assignment operators, a blocking assignment (=), and a non-blocking (<=) assignment. The non-blocking assignment allows designers to describe a state-machine update without needing to declare and use temporary storage variables. Since these concepts are part of Verilog's language semantics, designers could quickly write descriptions of large circuits in a relatively compact and concise form. At the time of Verilog's introduction (1984), Verilog represented a tremendous productivity improvement for circuit designers who were already using graphical schematic capture software and specially written software programs to document

and simulate electronic circuits.

The designers of Verilog wanted a language with syntax similar to the C programming language, which was already widely used in engineering software development. Like C, Verilog is case-sensitive and has a basic preprocessor (though less sophisticated than that of ANSI C/C++). Its control flow keywords (if/else, for, while, case, etc.) are equivalent, and its operator precedence is compatible. Syntactic differences include variable declaration (Verilog requires bit-widths on net/reg types), demarcation of procedural blocks (begin/end instead of curly braces {}), and many other minor differences.

A Verilog design consists of a hierarchy of modules. Modules encapsulate design hierarchy, and communicate with other modules through a set of declared input, output, and bidirectional ports. Internally, a module can contain any combination of the following: net/variable declarations (wire, reg, integer, etc.), concurrent and sequential statement blocks, and instances of other modules (sub-hierarchies). Sequential statements are placed inside a begin/end block and executed in sequential order within the block. However, the blocks themselves are executed concurrently, making Verilog a dataflow language.

Verilog's concept of 'wire' consists of both signal values (4-state: "1, 0, floating, undefined") and strengths (strong, weak, etc.). This system allows abstract modeling of shared signal lines, where multiple sources drive a common net. When a wire has multiple drivers, the wire's (readable) value is resolved by a function of the source drivers and their strengths. Subsets of statements in the Verilog language are synthesizable. Verilog modules that conform to a synthesizable coding style, known as RTL (register-transfer level), can be physically realized by synthesis software. Synthesis software algorithmically transforms the (abstract) Verilog source into a net list, a logically equivalent description consisting only of elementary logic primitives (AND, OR, NOT, flip-flops, etc.) that are available in a specific FPGA or VLSI technology. Further manipulations to the net list ultimately lead to a circuit fabrication blueprint (such as a photo mask set for an ASIC or a bit stream file for an FPGA).

5.3.5 DEFINITION OF CONSTANTS:

The definition of constants in Verilog supports the addition of a width parameter. The basic syntax is:

<Width in bits>'<base letter><number>

Examples:

- 12'h123 - Hexadecimal 123 (using 12 bits)
- 20'd44 - Decimal 44 (using 20 bits - 0 extension is automatic)
- 4'b1010 - Binary 1010 (using 4 bits)

- 6'o77 - Octal 77 (using 6 bits)

5.3.6 SYNTHESIZABLE CONSTRUCTS:

There are several statements in Verilog that have no analog in real hardware, e.g. \$display. Consequently, much of the language cannot be used to describe hardware. The examples presented here are the classic subset of the language that has a direct mapping to real gates.

5.3.7 INITIAL AND ALWAYS:

There are two separate ways of declaring a Verilog process. These are the always and the initial keywords. The always keyword indicates a free-running process. The initial keyword indicates a process executes exactly once. Both constructs begin execution at simulator time 0, and both execute until the end of the block. Once an always block has reached its end, it is rescheduled (again). It is a common misconception to believe that an initial block will execute before an always block. In fact, it is better to think of the initial-block as a special-case of the always-block, one which terminates after it completes for the first time.

5.3.8 FORK/JOIN:

The fork/join pair are used by Verilog to create parallel processes. All statements (or blocks) between a fork/join pair begin execution simultaneously upon execution flow hitting the fork. Execution continues after the join upon completion of the longest running statement or block between the fork and join.

5.3.9 FOUR-VALUED LOGIC:

The IEEE 1364 standard defines a four-valued logic with four states: 0, 1, Z (high impedance), and X (unknown logic value). For the competing Verilog HDL, a dedicated standard for multi-valued logic exists as IEEE 1164 with nine levels

5.3.10 SYSTEM TASKS:

System tasks are available to handle simple I/O, and various design measurement functions. All system tasks are prefixed with \$ to distinguish them from user tasks & functions. It presents a short list of the most often used tasks. It is by no means a comprehensive list.

5.3.11 PROGRAM LANGUAGE INTERFACE:

The PLI provides a programmer with a mechanism to transfer control from Verilog to a program function written in C language. It is officially deprecated by IEEE Std 1364-2005 in favor of the newer Verilog Procedural Interface, which completely replaces the PLI. The PLI enables Verilog to cooperate with other programs written in the C language such as test harnesses, instruction set simulators of a microcontroller, debuggers, and so on. For example, it provides the C functions tf_putlongp() and tf_getlongp() which are used to write and read the argument of the Current Verilog task or function, respectively.

CHAPTER 6

SUMMARYANDCONCLUSION

6.1 SUMMARY

The portable microprocessor controlled devices contain embedded memory, which represents a large portion of the system-on chip (SoC). These portable systems need ultralow power consuming circuits to utilize battery for longer duration. The power consumption can be minimized using nonconventional device structures, new circuit topologies, and optimizing the architecture. Although, voltage scaling has led to circuit operation in sub threshold regime with minimum power consumption, but there is a disadvantage of exponential reduction in performance. The circuit operation in the sub threshold regime has paved path toward ultralow power embedded memories, mainly static RAMs (SRAMs). However, in sub threshold regime, the data stability of SRAM cell is a severe problem and worsens with the scaling of MOSFET to sub nanometer technology. Due to these limitations it becomes difficult to operate the conventional 6-transistor (6T) cell at ultralow voltage (ULV) power supply.

6.2 CONCLUSION

An 8T SRAM cell with high data stability (high μ and low σ) that operates in ULV supplies is presented. We attained enhanced SNM in subthreshold regime using SE-DFC and read decoupling schemes. The proposed cell's area is twice as that of 6T. Still, it's better built-in process tolerance and dynamic voltage applicability enables it to be employed similar to cells (8T, 9T, and 10T) along with area overhead. The proposed 8T cell has high stability and can be operated at ULV of 200- 300 mV power supplies. The advantage of reduced power consumption of the proposed 8T cell enables it to be employed for battery operated SoC design. Future and applications of the proposed 8T cell can potentially be in low/ULV and medium frequency operation like neural signal processor, subthreshold processor, wideoperating-range IA-32 processor, fast Fourier transform core, and low voltage cache operation.

REFERENCES

- [1] C. Kushwah and S. K. Vishvakarma, "Ultra-low power sub-threshold SRAM cell design to improve read static noise margin," in *Progress in VLSI Design and Test (Lecture Notes in Computer Science)*, vol. 7373. Berlin, Germany: Springer-Verlag, 2012, pp. 139-146
- [2] C. B. Kushwah, D. Dwivedi, and N. Sathisha, "8T based SRAM cell and related method," U.S. Patent IN920 130 218 US1, May 30, 2013
- [3] M.-H. Tu et al., "A single-ended disturb-free 9T subthreshold SRAM with cross-point data-aware write word-line structure, negative bit-line, and adaptive read operation timing tracing," *IEEE J. Solid-State Circuits*, vol. 47, no. 6, pp. 1469-1482, Jun. 2012.
- [4] C. B. Kushwah, S. K. Vishvakarma, and D. Dwivedi, "Single-ended sub-threshold FinFET 7T SRAM cell without boosted supply," in *Proc. IEEE Int. Conf. IC Design Technol. (ICICDT)*, May 2014, pp. 1-4.
- [5] C. B. Kushwah and S. K. Vishvakarma, "A sub-threshold eight transistor (8T) SRAM cell design for stability improvement," in *Proc. IEEE Int. Conf. IC Design Technol. (ICICDT)*, May 2014, pp. 1-4.
- [7] M.-F. Chang, S.-W. Chang, P.-W. Chou, and W.-C. Wu, "A 130 mV SRAM with expanded write and read margins for subthreshold applications," *IEEE J. Solid-State Circuits*, vol. 46, no. 2, pp. 520-529, Feb 2011.
- [8] I. Carlson, S. Andersson, S. Natarajan, and A. Alvandpour, "A high density, low leakage, 5T SRAM for embedded caches," in *Proc. 30th Eur. Solid-State Circuits Conf.*, Sep. 2004, pp. 215-218.
- [9] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "A variation-tolerant sub-200 mV 6-T subthreshold SRAM," *IEEE J. Solid-State Circuits*, vol. 43, no. 10, pp. 2338-2348, Oct. 2008.

[10] S. A. Tawfik and V. Kursun, "Low power and robust 7T dual-Vt SRAM circuit," in Proc. IEEE Int. Symp. Circuits Syst., May 2008, pp. 1452- 1455.