ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Carbon Nanotube Field Effect Transistors Based RAM Design and Hardware Description Language Code Development

A thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Masters of Science in Electrical Engineering (Microelectronics Engineering)

By
Mengisteab Gebremedhin Yohannes

Advisor
Dr. Adeyabeba Abera

Addis Ababa, Ethiopia
August 2011.

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Carbon Nanotube Field Effect Transistors Based RAM Design and Hardware Description Language Code Development

By

Mengisteab Gebremedhin

Approval by Board of Examiners

Dr.-Ing. Getahun Mekuria	_____
Chairman (Department of Graduate Committee)	Signature
Dr. Adeyabeba Abera	_____
Advisor	Signature
Fitsum Assamnew	_____
Internal Examiner	Signature
Dr. Getachew Alemu	_____
External Examiner	Signature

DECLARATION

I, the undersigned, hereby declare that this thesis is my original work performed under the supervision of Dr. Adeyabeba Abera. It has not been presented as a thesis for a degree program in any other university and all sources of materials used for the thesis are duly acknowledged.

Name: Mengisteab Gebremedhin _____

Signature

Date of submission: July 12, 2011.

This thesis has been submitted for examination with my approval as university advisor.

Dr. Adeyabeba Abera _____

Signature

Addis Ababa, Ethiopia

July 2011.

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisor, Dr. Adeyabeba Abera for her continuous support, encouragement, and perceptive comments. This thesis would not have been possible without her support and guidance.

Most of my simulation work deserves a gratitude to ANSYS, Inc. company for their valuable software named Ansoft Simplorer.

I am also grateful to the Department of Electrical and Computer Engineering of Addis Ababa University for its financial support.

I also owe thanks to all my friends for their cooperation and support during the time of my studies.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The goal of this thesis is to develop carbon nanotube field effect transistors (CNFETs) based static random-access memory (SRAM) and implement it into a Very-high-speed integrated circuit Hardware Description Language Analog and Mixed-Signal (VHDL-AMS). To achieve this objective, a compact model of the transistor known as enhancement-mode MOSFET-like SWCNT-CNFET is used. This circuit-compatible model of CNFET is described using VHDL-AMS and tested for basic electrical characteristics. This model is valid for CNFETs with channel lengths greater than 20 nm. Based on the CNFETs a new SRAM is designed, and implemented in VHDL-AMS. The performance of the proposed SRAM cell is investigated and compared with SRAMs from conventional metal-oxide semiconductor field effect transistors (MOSFETs). The effect of substrate biasing a CNFET is also demonstrated and implemented in designing the SRAM cell. The VHDL-AMS codes of the CNFET and the SRAM are simulated in software known as Ansoft Simplorer.

The compact model of the CNFET is organized hierarchically in three main levels. The first level models the intrinsic channel just beneath the gate of the transistor. The second level builds upon the first level and models the doped source and drain regions of the CNFET. The last level represents the complete trans-capacitance model of the transistor and accounts for multiple CNTs.

The proposed SRAM cell is composed of four CNFETs and two load resistors. The driver CNFETs of the proposed SRAM cell are substrate biased. Besides, 8-bit complete SRAM architecture based on this cell is indicated. The performance analysis of the SRAM shows that it has better writing and reading speed as well as better stability when compared with SRAM from conventional MOSFETs. Specifically, the newly proposed SRAM cell has read time of twenty five pico seconds, write time of twenty pico seconds and can tolerate a noise of 120 mV at 32 nm node technology.

CHAPTER 1
INTRODUCTION

1.1 Introduction to IC Technology

Beginning from the first decade of the twentieth century electronic circuitry has undergone tremendous changes following the invention of vacuum tube. In those days, the active components (vacuum tubes like triode) and passive components (such as resistors, capacitors and inductors) of circuits were separate and distinct units connected by soldered leads. After the invention of the transistor in 1948, integrated circuits (ICs) were developed in 1959 [1]. When compared with the discrete components, ICs are advantageous due to their very small physical size and mass, high reliability, low power consumption, low cost, improved response time, higher yield and can be easily replaced.

ICs started with bipolar processes, and then they were gradually replaced by metal oxide semiconductor (MOS) and complementary metal oxide semiconductor (CMOS) devices. In the first decade of the 21st century, about 88% of the IC market was MOS based and about 8% BJT based. Of the entire semiconductor industry, about 86% were digital ICs. Semiconductor memories such as dynamic random-access memory (DRAMs), static random-access memories (SRAMs), and nonvolatile flash memories made up approximately 25% of the market, microprocessors about 25%, and other application-specific ICs (ASICs) about 20% [1].

1.1.1 CMOS Scaling

For the past forty years CMOS scaling has offered improved performance from one technology node to the next. However, as device scaling moves beyond the 32 nm node, there exist significant technology challenges. Two of the main challenges are: the considerable increase of standby power dissipation and the increasing variability in device characteristics [2]. Silicon based technology will reach its limits in 2020 when the channel length of metal-oxide semiconductor field effect transistor (MOSFET) is below 10 nm [3]. Hence, researches are being conducted in the semiconductor industry on different materials and devices to integrate with the current silicon-based technology or to substitute it in the future. Among the number of investigated solutions carbon nanotubes (CNTs) are found to be promising materials.

1.1.2 Carbon Nanotubes

CNTs are hollow cylinders of graphene. They can be utilized to build both low-resistance high-strength interconnections and highly scalable low-power carbon nanotube field effect transistors (CNFET) and single electron tunneling transistors [3]. Carbon atoms can also form long chains, branched chains, and rings that may also have chains attached to them [4]. Today, many research teams are carrying out studies about CNFET devices and their logic applications all over the world, both in industrial laboratories and in universities. One of the basic ideas is to replace the silicon MOSFETs with CNFETs to overcome most of the limitations of silicon MOSFETs.

1.1.3 RAM

Before the development of the static and dynamic integrated random access memory circuits, computers used relays or various kinds of vacuum tube arrangements to implement memory functions. In 1950s the widespread writable random-access memory was the magnetic core memory [5]. Today memory chips are being manufactured as an IC made of millions of transistors and capacitors. There are many types of memories with different ranges of speeds and storage capacity. Some operate at the highest speed and can be accessed directly by the processing unit. Others operate at slower speeds but store more data. SRAMs are the major cache memory units of digital systems such as microprocessors.

1.2 Motivation

Silicon is the dominant semiconductor material in the semiconductor industry due to its availability, formation of oxide, low cost, high yield in mass production, and properties. IC, particularly MOSFET-based IC, manufacturing is in general undergone in harmony with silicon processing; in spite of the fact that silicon suffers from low mobility and short channel effects. One way of improving MOSFETs is by replacing the bulk silicon channel material by CNT, which has high mobility. Therefore, it is of good importance to deal with these new types of transistors and to implement them to one of the most sensitive digital units namely; SRAM. As a result, design and simulation of CNFET-based SRAM is done by using VHDL-AMS.

1.3 Objective

1.3.1 General Goal

The purpose of this thesis is to develop carbon nanotube field effect transistors (CNFETs) based static random-access memory (SRAM) and implement it into a Very-high-speed integrated circuit Hardware Description Language Analog and Mixed-Signal (VHDL-AMS).

1.3.2 Specific Tasks

- Study the basic properties of carbon nanotubes and CNFETs.
- Select appropriate CNFET model.
- Develop the VHDL-AMS code of the CNFET and investigate its main characteristics.
- Use the selected CNFET model to design basic SRAM cell.
- Develop the VHDL-AMS code of the SRAM.
- Simulate the performance of the newly designed SRAM and compare it with SRAM made of conventional MOSFETs.

1.4 Outline of the Thesis

The thesis is organized in six chapters and appendices. Chapter one generally introduces the thesis by briefly covering the areas in the center of attention and presenting the motivation and the objective of the thesis. The second chapter describes the concepts underlying CNFET and its modeling. The third chapter enlightens SRAM technology emphasizing on the foundations for designing. Chapters four and five focus mainly on the works done. In Chapter 4, the CNFET and the SRAM cell are described using VHDL-AMS. The SRAM is investigated with different parameters and the results are compared with SRAM from conventional MOSFETs in Chapter 5. Finally, the core body of the paper is terminated by presenting concluding ideas and forwarding future works based on the results obtained from the analysis. Appendix A and B respectively show the main symbols and abbreviations used in the paper while Appendix C and F introduce VHDL-AMS and the software used for simulation respectively. The VHDL-AMS code of the components and VHDL-AMS package for the transistor are concisely illustrated in Appendix D and E correspondingly. A magnified view of the 8-bit SRAM is also shown on Appendix G.

CHAPTER 2
BACKGROUND THEORY

2.1 Carbon Nanotube Field Effect Transistors

Carbon nanotube field effect transistor (CNFET) refers to a field effect transistor that utilizes a single CNT or an array of CNTs as the channel material instead of bulk silicon in the traditional MOSFET structure. The operation principle of CNFET is similar to that of traditional MOSFET. CNFETs are introduced in Section 2.1.4 and CNFET compact model is explained in detail in Section 2.2.

2.1.1 Carbon

Carbon, nonmetallic chemical element, known by the symbol C, is the fundamental building block of materials in living organisms. A carbon atom has six electrons. In an isolated carbon atom there are two electrons in 1s orbital of the inner shell and four valence electrons in 2s and 2p atomic orbitals with ground state configuration $1s^2 2s^2 2p^2$. The valence shell can hold eight electrons, and atoms in general are much more stable when octet state is satisfied. The carbon atom can attain a stable configuration by forming four covalent bonds. Carbon is unique among the elements in the extent to which it forms bonds between like atoms and in the diversity of compounds that are formed. Carbon atoms form long chains, branched chains, and rings that may also have chains attached to them [4].

Carbon occurs in nature in nearly pure form in diamond and graphite [5, 6]. It is also the major component of coal, petroleum, limestone, and materials in living organisms. The density of diamond is about 3.5 grams per cubic centimeter (g/cm^3), graphite ranges from 1.9 to 2.3 g/cm^3 [5]. Diamond is one of the hardest known materials, while graphite is one of the softest. In 1985 chemists created a new allotrope of carbon by heating graphite to extremely high temperatures [6]. They named the allotrope buckminsterfullerene, after American architect R. Buckminster Fuller. The original fullerene forms molecules of 60 carbon atoms (with a molecular formula of C_{60}) [6]. The molecules are shaped like tiny soccer balls (called buckyballs). Scientists have since discovered other fullerenes, including very narrow, long tubes and the C_{70} fullerene, an elongated structure shaped more like a football but rounded on the ends [5, 6].

Graphite is black, soft and conductor of electricity. In graphite the atoms form planar or flat layers. Each layer is made up of rings containing six carbon atoms, with each carbon atom being bonded to three coplanar neighbouring atoms by strong covalent bonds of length 0.142 nm and the distance between planes is 0.335 nm [7]. Since the interlayer force is van der Waals, a single layer of graphite known as graphene can be prepared by a simple peeling of the top layer of highly oriented graphite using a tape [7]. As carbon atoms form graphene, three atomic orbitals namely 2s, $2p_x$, and $2p_y$ are hybridized into three sp^2 orbitals in the same plane while the $2p_z$ orbital remains perpendicular to other orbitals [8]. The hybridized orbitals are responsible for σ-bonds between the adjacent carbon atoms and the $2p_z$ orbital results in π-bonds out of the plane of graphene [8]. Each atom has three σ-bonds (with 120° between any two of the bonds) and belongs to three neighboring rings. The fourth electron of each atom becomes part of an extensive π-bond system [5]. A sigma bond is a bond resulting from head-on overlap of atomic orbitals while a pi bond is a bond resulting from side-on overlap of atomic orbitals [4].

Generally, electrical transport properties are determined by the electrons (holes) near the Fermi level, since only these electrons (holes) have easy access to the unoccupied (occupied) states. In graphene, the π orbitals, which lie near the Fermi level, are responsible for the electrical transport properties by forming delocalized states [8]. The π-bond does not hold electrons as tightly as the σ-bond holds the first pair; therefore, electrons in π-bonds are much more reactive than are electrons in σ-bonds.

In diamond, each carbon atom bonds tetrahedrally to four other carbon atoms to form a three-dimensional lattice. The shared electron pairs are held tightly in sigma bonds between adjacent atoms. Pure diamond is an electrical insulator, colorless and is used in industrial cutting tools [5].

2.1.2 Carbon Nanotubes

A carbon nanotube (CNT) can be viewed as a hollow cylinder formed by rolling a graphene sheet. The end-to-end rolled graphene (the CNT) has already created an explosion of research in nanotechnology since its discovery by Iijima in 1991 [7]. CNTs are electrically and thermally more conductive, chemically and biologically more active, and

mechanically stronger than graphene [9]. In addition, topological defects such as pentagons and heptagons can be incorporated into the hexagonal network to form capped, bent, toroidal, and helical nanotubes which allow electrons to be localized. A nanotube is called defect free if it is of only hexagonal network and defective if it also contains topological defects such as pentagon and heptagon or other chemical and structural defects [9]. The electron confinement along the tube circumference makes a defect-free nanotube either semiconducting or metallic with quantized conductance whereas pentagons and heptagons generate localized states [9].

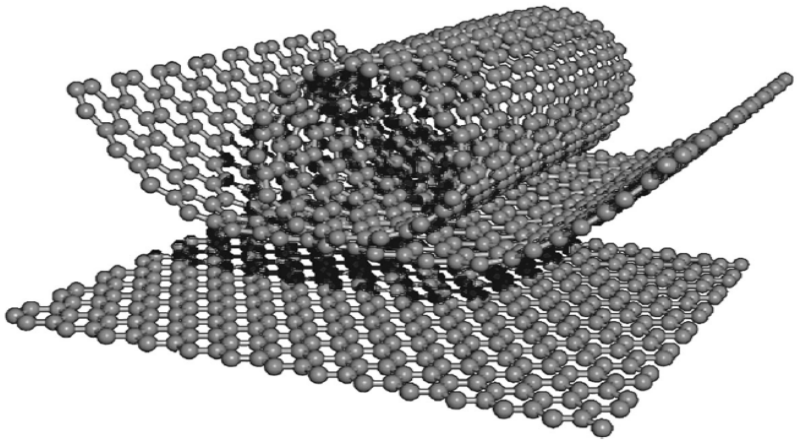

Figure 2.1 Schematic of Graphene and How it Can be Rolled to Form a CNT [10].

A single-walled carbon nanotube (SWCNT) is a hollow cylinder of a graphite sheet whereas a multi-walled carbon nanotube (MWCNT) is a group of coaxial SWCNTs. A MWCNT can also be viewed as a scrolled graphite sheet or a spiral graphite sheet, or mixture of scrolled structure and concentric shells. In general, most SWCNTs are defect-free whereas MWCNTs are relatively more defective. The diameter of SWCNT should be at least 0.4 nm large to afford strain energy and at most about 3.0 nm large to maintain tubular structure and prevent collapsing. Typical MWCNT diameter is larger than 2 nm inside and smaller than 100 nm outside [9].

A SWCNT can be uniquely characterized by a chiral vector $\mathbf{C_h}$ in terms of a set of two integers (n_1, n_2) corresponding to graphene vectors $\mathbf{a_1}$ and $\mathbf{a_2}$ as shown in Figure 2.2 [9, 11],

$$\mathbf{C_h} = n_1\mathbf{a_1} + n_2\mathbf{a_2} \qquad (2 - 1)$$

Thus, the SWCNT is constructed by rolling up the sheet such that the two end-points of the vector $\mathbf{C_h}$ are superimposed. This tube is denoted as (n_1, n_2) with diameter (D_{CNT}) given by

$$D_{CNT} = \frac{a\sqrt{n_1^2 + n_1 n_2 + n_2^2}}{\pi} \qquad (2 - 2)$$

where $a = |\boldsymbol{a_1}| = |\boldsymbol{a_2}|$ is lattice constant of graphene. The lattice constant and inter-tube spacing vary with tube diameter or in radial direction. Most experimental measurements and theoretical calculations show that the C–C bond length (d_{cc}) is 0.144 nm, $a = |\boldsymbol{a_1}| = |\boldsymbol{a_2}|$= 0.246 nm and inter-tube spacing (d_{tt}) is 0.34 nm. The tubes with $n_1 = n_2$ are commonly referred to as armchair tubes and $n_2 = 0$ as zigzag tubes. Others are called chiral tubes with the chiral angle, θ, defined as that between the vector $\mathbf{C_h}$ and the zigzag direction $\mathbf{a_1}$,

$$\theta = \tan^{-1}\left[\sqrt{3}n_2/(2n_1 + n_2)\right] \qquad (2 - 3)$$

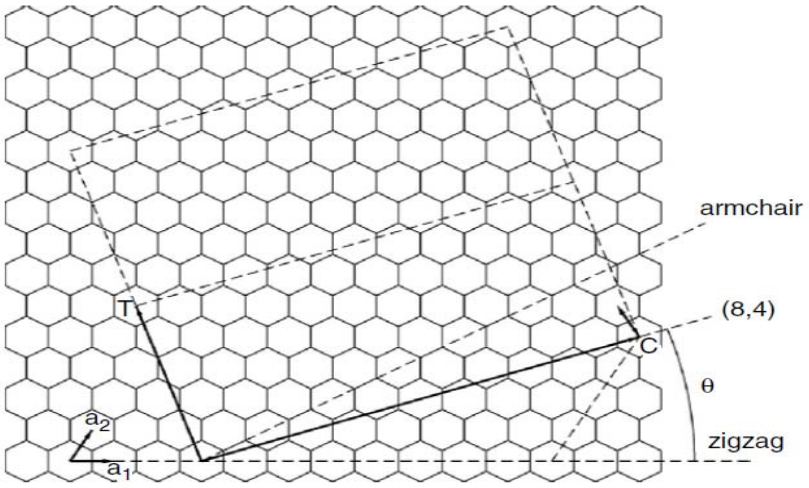

Figure 2.2 A Diagram for Construction of an (8, 4) Nanotube [9].

The electronic properties of these nanotubes are determined by their (n_1, n_2) chiral indices according to the rules [11],

$$If\ \frac{(n_1 - n_2)}{3} \begin{cases} = integer, then\ the\ nanotube\ is\ metallic, \\ \neq integer, then\ the\ nanotube\ is\ semiconducting. \end{cases} \qquad (2 - 4)$$

The armchair SWCNT is, therefore, always metallic, and for a semiconducting SWCNT of diameter (D_{CNT}) the band-gap (Eg) is inversely proportional to D_{CNT} or approximately given by [12], Eg = 0.84 eV/ D_{CNT} (nm).

2.1.3 Applications of CNTs

The direct band gap and one-dimensional band structure of CNTs make them suitable for optical and optoelectronic applications. High Young's modulus (over 1 TPa) and tensile strength of over 100 GPa are exhibited by CNTs due to the σ-bonds [9]. The strong sensitivity of CNTs to chemical interactions such as adsorption, doping, capillarity, and charge transfer makes them suitable materials to produce various electronic devices [6].

The ability of carbon nanotubes to serve as a potential alternative to existing silicon complementary metal oxide semiconductor (CMOS) technology and overcome some limitations and shortcomings of silicon technology has resulted in analysis for the past few years. CNTs are being intensively investigated as possible structures from which nanoscale transistors and logic gates might be fabricated. It is reported on CNFETs that their models are being developed; they have been fabricated; and have better performance than present silicon transistors of equivalent size. Enormous progress has been made toward many applications of CNTs, including [9]:

- Probes, sensors, and actuators for molecular imaging, sensing, and manipulation;
- Transistors, memories, and other nanoelectronic devices;
- Field emission devices for x-ray instruments, flat panel display.

2.1.4 CNFETs

Since the introduction of transistors, continuous reduction of electronic circuit size and power dissipation have been the ongoing theme in electronics industry. However as the feature size becomes smaller, scaling the silicon MOSFET becomes increasingly harder. This increasing challenge is often attributed to: quantum mechanical tunneling of carriers and difficulties to control the density and location of dopant atoms in the channel and source/drain region to provide high on/off current ratio [13].

Innovation and integration of new materials, such as high-k gate dielectrics, various metals, silicides and nitrides, has been used for the evolutionary path of CMOS device

scaling [8]. However, the active channel material has dominantly remained the same, mainly due to the scalability and manufacturability of the silicon technology. As the device dimensions, such as the channel lengths approach the sub-10 nm system, direct tunneling between source (S) and drain (D), and severe short channel effects present a fundamental challenge in continued scaling of silicon devices [8]. As a result, tremendous research efforts have recently been undertaken by various academic and industrial research groups for integrating new semiconductors as the channel material to enable

(i) more efficient transport of carriers (i.e., higher mobility) and
(ii) improved electrostatics at nanoscale (i.e., non-planar channel materials).

In most approaches, a hybrid technology is predicted, where silicon still remains the handling substrate for fabrication processing, heat transport, and mechanical support purposes, with a new semiconductor integrated for enhanced device operations or added new functionalities. One such material system is carbon nanotubes based FET known as carbon nanotube field effect transistor shown in Figure 2.3. CNFET which is three (or four) terminal device consists of a semiconduting nanotube, acting as conducting channel, bridging the source and drain contacts. The device is turned on or off electrostatically via the gate (G) [14].

CNFETs have generated a lot of interest because of their ability to serve as an alternative to existing silicon technology. Models that can accurately describe the behavior of the CNFETs are being developed so that the design and evaluation of circuits using these devices can be made. CNFET is the most promising technology to extend or complement traditional silicon technology due to three reasons [14]: First, the operation principle and the device structure are similar to CMOS devices; we can reuse the established CMOS design infrastructure. Second, we can reuse CMOS fabrication process. Third, CNFET has the best experimentally demonstrated device current carrying ability to date.

Based on the device operation mechanism, CNFETs can be categorized as the MOSFET-like CNFET and the Schottky Barrier (SB) CNFET [14]. The conductivity of SB-CNFET is governed by the majority carriers tunneling through the SBs at the end contacts. The on-current and thereby device performance of SB-CNFET is determined by the contact

resistance due to the presence of tunneling barriers at both or one of the source and drain contacts, instead of the channel conductance. The SBs at source/drain (S/D) contacts are due to the Fermi-level alignment at the metal-semiconductor interface. Both the height and the width of the SBs, and therefore the conductivity, are modulated by the gate electrostatically. SB-CNFET shows ambipolar transport behavior. The work function induced barriers at the end contacts can be made to enhance either electron or hole transport. On the other hand, MOSFET-like CNFET exhibits unipolar behavior by suppressing either electron or hole transport with heavily doped source/drain. The non-tunneling potential barrier in the channel region, and thereby the conductivity, is modulated by the gate-source bias [8, 14].

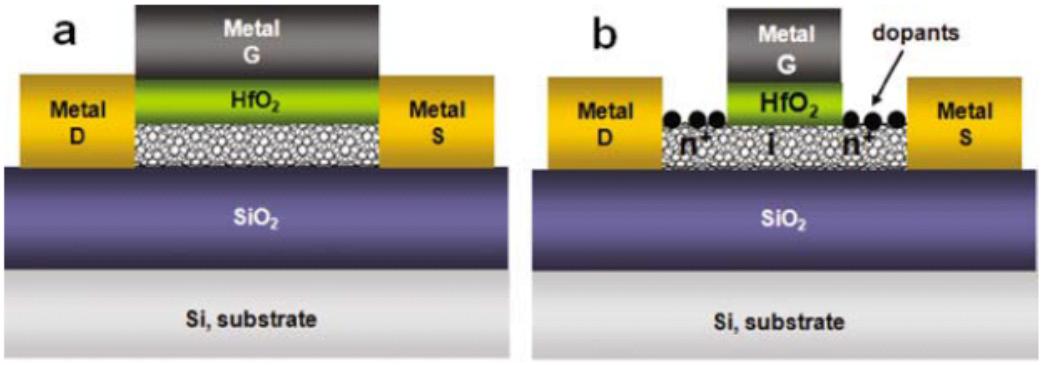

Figure 2.3 (a) SB-CNFET and (b) MOSFET-like CNFET [8].

MOSFET-like CNFETs with heavily doped S/D contacts have two main advantages over SB-CNFETs with metal contacts [8]. First, the heavily doped contacts enable a more efficient injection of carriers from S/D into the channel than the metal-contacted devices, even when SB heights are zero. Second MOSFET-like CNFETs enable lower OFF currents due to the decreased leakage at the drain. But an advantage of SB-CNFETs over MOSFET-like CNFETs is reduced parasitic resistances of the S/D contacts.

Titanium contacted SWCNT-FETs exhibit ambipolar transfer characteristics while Aluminum contacted devices are intrinsically n-type. In addition to the contact material, the diameter of nanotubes plays a key role in determining the interface energy barriers for carrier injection. This is due to the fact that the band gap of semiconducting CNTs is inversely proportional to the diameter and the chemical reactivity and surface properties of CNTs strongly depend on the nanotube diameter [8, 14, 15].

2.2 CNFET Modeling

There are generally two approaches in modeling SWCNT-FETs. The more numerically intensive NEGF (Non-Equilibrium Green's Function) approach, and a simpler compact modeling methodology based on the ballistic transport assumption [8].

2.2.1 Overview of the Compact Model

This model includes a complete transcapacitance network in order to produce better predictions of the performance of the transistor. The type of transistor described here is MOSFET-like SWCNT-FET with semiconducting SWCNTs regions forming the channel and highly doped SWCNT regions forming the source/drain extension regions. The model, indicated in Figure 2.4, is organized hierarchically in three main levels [8, 14, 16].

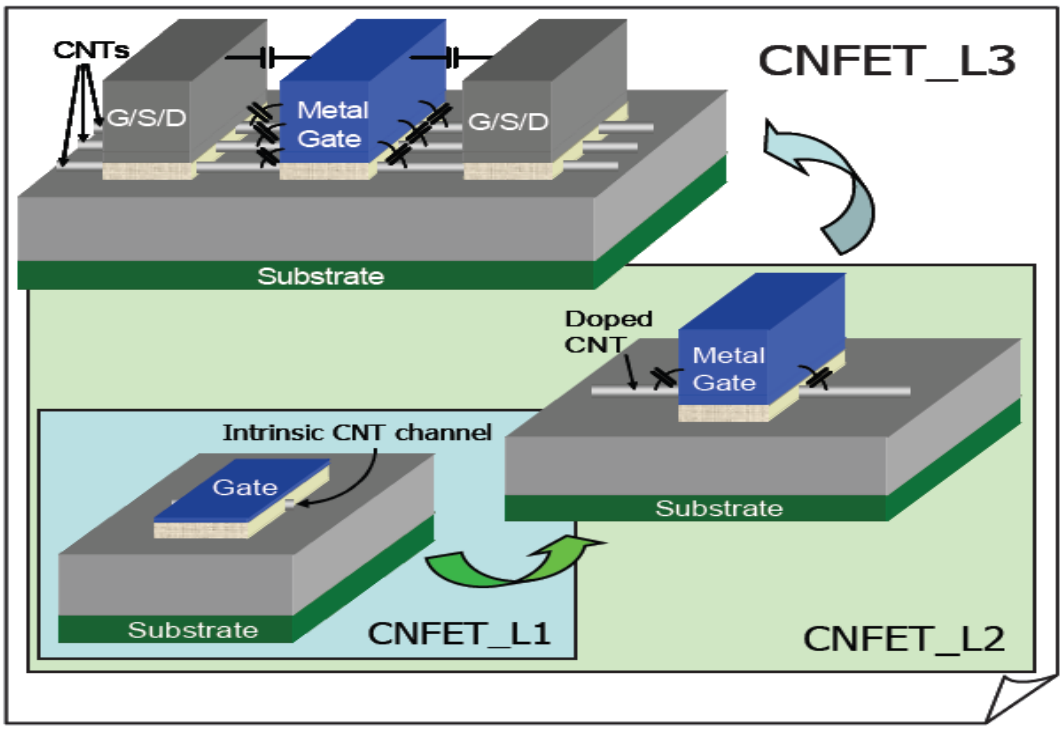

Figure 2.4 Compact SWCNT-FET Device Model [17].

In Figure 2.4 the top diagram (CNFET_L3) shows the device structure modeled, which allows multiple nanotubes per device. CNFET_L2 models the source/drain extension regions and contacts of the device and its parasitics. CNFET_L1 is the core of the model and describes the intrinsic SWCNT channel region of the CNFET.

2.2.2 Intrinsic Channel Model

This model is referred to as first level (CNFET_L1) and its circuit diagram is shown in Figure 2.5. It is the core of the model and is used to describe the portion of the SWCNT under the metal gate, which forms the SWCNT-FET channel region. It models the intrinsic channel region of the CNFET which serves as the first level modeling of the complete device model. This model includes the quantum confinement on both the circumferential and the axial directions, the acoustical/optical phonon scattering in the channel region, and the intrinsic AC behavior which is delivered by a dynamic gate capacitance network. This level assumes near-ballistic transport, with acoustic and optical phonon scattering, and includes parasitic capacitances. Three current contributions are considered [16]:

1) thermionic current from the semiconducting subbands (I_{semi});
2) current from the metallic subbands (G_{metal}); and
3) band-to-band tunneling (BTBT) current from the semiconducting subbands (I_{btbt}).

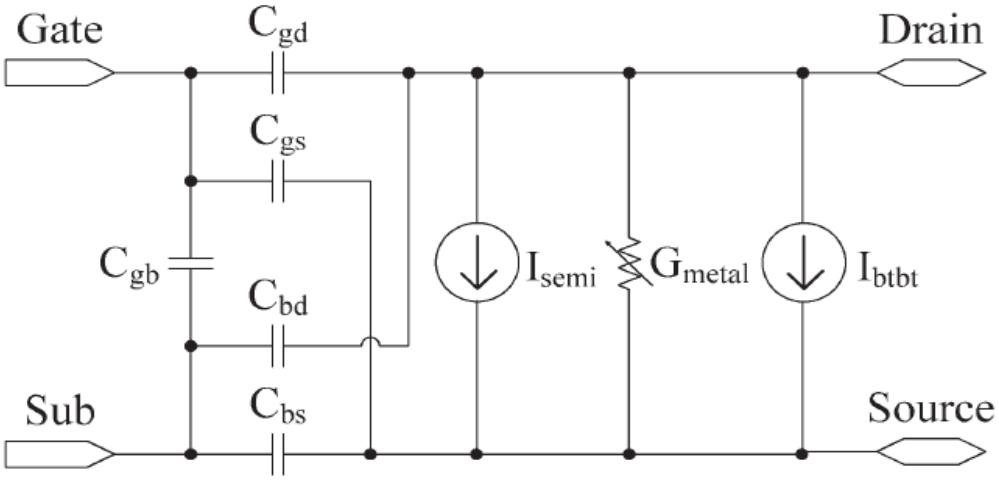

Figure 2.5 Circuit Schematic of the Intrinsic SWCNT Channel Model (CNFET_L1) [16].

2.2.2.1 I_{semi} Calculation

For SWCNT with chiralities (n_1, n_2), a finite length (L_g), lattice constant *a*, and a diameter (D_{CNT}), the energy and wave number (E-k) dispersion relation is quantized into discrete sub-states. We denote *(m, l)* as the l^{th} sub-state at the m^{th} sub-band, k_m as the wave-number of the m^{th} sub-band in circumferential direction, and k_l as the wave-number of the l^{th}

sub-state in current flow direction. The wave numbers related with semiconducting sub-bands are given by [8, 16],

$$k_m = \frac{2\pi}{a\sqrt{n_1^2 + n_1 n_2 + n_2^2}} \cdot \lambda \qquad (2-5)$$

$$\lambda = \begin{cases} \dfrac{6m - 3 - (-1)^m}{12}, & m = 1, 2, \ldots \; and \; mod\,(n_1 - n_2, 3) \neq 0 \\ m, & m = 0, 1, 2, \ldots \; and \; mod\,(n_1 - n_2, 3) = 0 \end{cases} \qquad (2-6)$$

$$k_l = \frac{2\pi}{L_g} \cdot l\,, \qquad l = 0, 1, 2, \ldots \qquad (2-7)$$

m=0 is reserved for the metallic sub-band.

Around the Fermi point with carrier energy $E_{m,l} << V_\pi$ (~3.033 eV, the carbon π-π bond energy in the tight-binding model), CNT E-k dispersion relation can be expressed as [16],

$$E_{m,l} \approx \frac{\sqrt{3}aV_\pi}{2} \cdot \sqrt{k_m^2 + k_l^2} \qquad (2-8)$$

$E_{m,l}$ is the carrier energy at the (m,l) sub-state above the intrinsic level E_i, and $E_{m,0}$ is the half band gap of the m^{th} sub-band.

The current contributed by the sub-state (m,l) is given by [8, 16],

$$J_{m,l}(V_{xs}, \Delta\Phi_B) = \frac{2q}{h}\frac{\sqrt{3}a\pi V_\pi}{L_g} \cdot \frac{k_l}{\sqrt{k_m^2 + k_l^2}} \cdot \frac{1}{1 + e^{(E_{m,l} + qV_{xs} - \Delta\Phi_B)/kT}} \qquad (2-9)$$

Where,

- V_{xs} is potential difference between node x and source,
- $\Delta\Phi_B$ is the channel surface potential change with gate/drain bias,
- q is the electronic charge of value 1.60218×10^{-19} C,
- h is the Planck's constant (6.62617×10^{-34} Js or 4.1357×10^{-15} eVs),
- k is the Boltzmann's constant (1.38066×10^{-23} J/K or 8.6174×10^{-5} eV/K),
- T is the temperature in Kelvin,
- $E_{m,l}$ is the carrier energy at the substate (m, l).

The total current contributed by all sub-states can then be obtained by [14, 16],

$$I_{semi}(V_{ch,DS}, V_{ch,GS}) = 2 \sum_{k_m,\ m=1}^{M} \sum_{k_l,\ l=1}^{L} \left[T_{LR} J_{m,l}(0, \Delta\Phi_B)\big|_{+k} - T_{RL} J_{m,l}(V_{ch,DS}, \Delta\Phi_B)\big|_{-k} \right] \qquad (2-10)$$

Where,

- $V_{ch,DS}$ and $V_{ch,GS}$ denote the Fermi potential differences near source side within the channel.
- M and L are the number of sub-bands and the number of sub-states, respectively. For typical devices with appropriate diameter range (D_{CNT} < 3 nm) and short gate length (Lg = 100 nm), only the first 2 or 3 sub-bands and the first 10-15 sub-states have a significant impact on the current using a sub-1V power supply.
- T_{LR} and T_{RL} are the transmission probability of the carriers at the sub-state (m, l) in +k branch and –k branch, respectively.

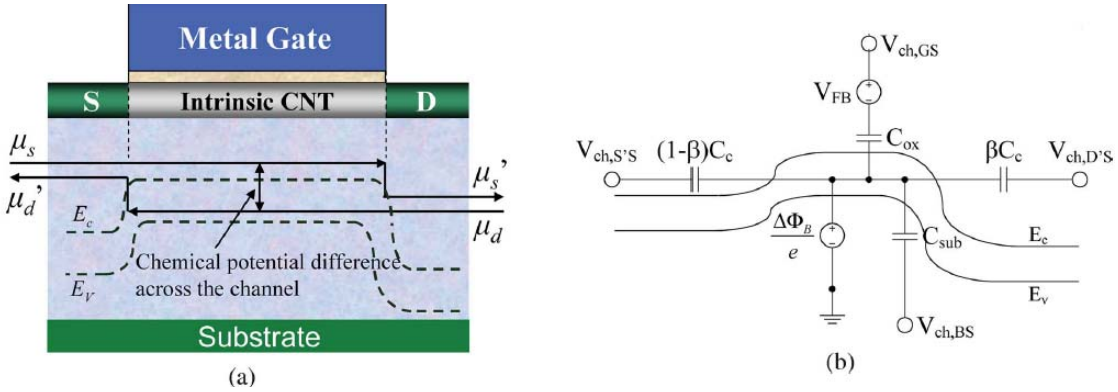

Figure 2.6 (a) Ideal CNFET with Intrinsic Channel (b) The Electrostatic Capacitor Model Used to Calculate the Channel Surface Potential Change [16].

The Fermi level profiles and the energy band diagram in the channel region with ballistic transport are illustrated in Figure 2.6(a). The potential differences μ_s-μ_s' and μ_d-μ_d' are determined by both the applied bias and the property of the source/drain (S/D) extension regions. Near-ballistic transport and ideal contacts are assumed in this model; therefore, $qV_{DS} \approx \mu_d$-μ_s. Hence, μ_s /μ_d remains almost constant in the source-channel /drain-channel region.

Referring to Figure 2.6, we can calculate the channel surface potential change $\Delta\Phi_B$ using the following charge conservation equations [8,14,16],

$$Q_{cap} = C_{ox}\left(V_{ch,GS} - V_{FB}\right) + C_{sub}V_{ch,BS} + \beta C_c V_{ch,D'S} + (1-\beta)C_c V_{ch,S'S} - (C_{ox} + C_{sub} + C_c)\frac{\Delta\Phi_B}{q} \quad (2-11)$$

$$Q_{CNT} = \frac{4q}{L_g}\sum_{k_{m,m=m0}}^{M}\sum_{k_{l,l=0}}^{L}\left[\frac{1}{1+e^{(E_{m,l}-\Delta\Phi_B)/kT}} + \frac{1}{1+e^{(E_{m,l}+qV_{DS}-\Delta\Phi_B)/kT}}\right] \quad (2-12)$$

$$Q_{CNT} = Q_{cap} \quad (2-13)$$

$$m0 = \begin{cases} 1, & mod\ (n_1 - n_2, 3) \neq 0 \\ 0, & mod\ (n_1 - n_2, 3) = 0 \end{cases} \quad (2-14)$$

Where,

- Q_{cap} is the charge induced by the electrodes,
- Q_{CNT} is the total charge induced on SWCNT surface,
- C_{ox} the capacitance between the gate and channel,
- C_{sub} the capacitance between channel and substrate,
- C_c the capacitance between channel and external drain (D') / source (S'),
- V_{FB} is the flat band voltage,
- V_{BS} is the potential difference between substrate and source,
- β is a fitting parameter.

Three scattering mechanisms in the channel region are considered: acoustic phonon scattering, optical phonon scattering, and elastic scattering. Assuming the optical phonon scattering mean free path (MFP) (λ_{op} ~ 15 nm) and the acoustic phonon scattering MFP (λ_{ap} ~ 500 nm) are constant, the effective acoustic phonon scattering MFP (l_{ap}) and the effective optical phonon scattering MFP (l_{op}) of the semiconducting sub-bands can be normalized to the available target empty states as [16],

$$l_{ap}(V_{xs}, m, l) = \frac{D_0\lambda_{ap}}{D\left(E_{m,l}\right)\left[1 - \mathrm{f}_{FD}\left(E_{m,l} + qV_{xs} - \Delta\Phi_B\right)\right]} \quad (2-15)$$

$$l_{op}(V_{xs}, m, l) = \frac{D_0 \lambda_{op}}{D(E_{m,l} - \Omega\hbar)[1 - f_{FD}(E_{m,l} + qV_{xs} - \Delta\Phi_B - \Omega\hbar)]} \quad (2-16)$$

Where,

- D_o is constant $8/(3\pi d_{cc} V_\pi)$ where d_{cc} (~0.144 nm) is the carbon-carbon bond distance,
- $\Omega\hbar$ (~0.16 eV) is the optical phonon energy that a carrier attains before an optical phonon scattering can occur,
- f_{FD} (E) is the Fermi-Dirac distribution function given by,

$$f_{FD}(E) = \frac{1}{1 + e^{E/kT}} \quad (2-17)$$

- D(E) is the CNT density of states (DOS) which is valid in the range $E_{m,l} << V_\pi$,

$$D(E) = \begin{cases} \dfrac{ED_0}{\sqrt{E^2 - E_{m,0}^2}}, & E > E_{m,0} \\ 0, & E \le E_{m,0} \end{cases} \quad (2-18)$$

The effective phonon scattering MFP is in the form of [8,14,16],

$$\frac{1}{l_{eff}(V_{xs}, m, l)} = \frac{1}{l_{ap}(V_{xs}, m, l)} + \frac{1}{l_{op}(V_{xs}, m, l)} \quad (2-19)$$

Thus the transmission probabilities in Equation (2-10) are given by,

$$T_{LR} = \frac{l_{eff}(V_{ch,DS}, m, l)}{l_{eff}(V_{ch,DS}, m, l) + L_g} \quad (2-20)$$

$$T_{RL} = \frac{l_{eff}(0, m, l)}{l_{eff}(0, m, l) + L_g} \quad (2-21)$$

For a long channel device (L_g >> 100 nm), the wave number k_l can be represented as a continuous variable. By replacing the inner summation with the integral function and assuming $T_{LR}=T_{RL}=T_m$, Equation (2-10) can be simplified as [14, 16],

$$I_{semi}(V_{ch,DS}, V_{ch,GS}) \approx \frac{4q^2}{h} \sum_{k_m,\ m=1}^{M} T_m \cdot \left[V_{ch,DS} + \frac{kT}{q} ln\left(\frac{1 + e^{(E_{m,0} - \Delta\Phi_B)/kT}}{1 + e^{(E_{m,0} + qV_{ch,DS} - \Delta\Phi_B)/kT}} \right) \right] \quad (2-22)$$

2.2.2.2 I_{btbt} Calculation

Assuming ballistic transport for the tunneling process, the BTBT current is approximated by the BTBT tunneling probability (T_{btbt}) times the maximum possible tunneling current integrating from the conduction band at drain side up to the valance band at source side as,

$$I_{btbt} = \frac{4qkT}{h} \sum_{k_m,\ m=1}^{M} T_{btbt} \left[ln\left(\frac{1 + e^{(qV_{ch,DS} - E_{m,0} - E_f)/kT}}{1 + e^{(E_{m,0} - E_f)/kT}} \right) \left(\frac{max\left(qV_{ch,DS} - 2E_{m,0}\ ,0\right)}{qV_{ch,DS} - 2E_{m,0}} \right) \right] \quad (2-23)$$

Where, E_f is the Fermi level of the doped source/drain nanotube in units of eV and T_{btbt} is given by,

$$T_{btbt} \approx \frac{\pi^2}{9} exp\left(-\frac{\pi\sqrt{m^*}\left(\eta_m 2E_{m,0}\right)^{3/2}}{q\hbar F 2^{3/2}} \right) \quad (2-24)$$

Where,

- m* is the effective electron mass given by,

$$m^* = \frac{\hbar^2}{\left({}^{\partial^2 E_{m,l}}\!/_{\partial k_l^2} \right)} \quad (2-25)$$

  Where ħ is the reduced Planck's constant (1.055×10^{-34} Js or 6.582×10^{-16} eVs),

- η_m (~ 0.5) is a fitting parameter which represents the band gap narrowing effect under high electrical field,
- F is the electrical field triggering the tunneling process near the drain side junction given by,

$$F = \frac{\left(V_{ch,DS} + \left(E_f - \Delta\Phi_B\right)/q\right)}{l_{relax}} \quad (2-26)$$

  Where, l_{relax} is a fitting parameter.

2.2.2.3 Calculation of the Parasitic Capacitances

Controlled trans-capacitance array among the four electrodes (G, S, D, B) is used to model the intrinsic AC response of the CNFET as shown in Figure 2.5. The capacitance values between the nodes in the channel region are expressed as [16]:

$$C_{gs} = \frac{L_g C_{ox}}{2} \frac{C_{Qs} + C_{Qd} + 2(1-\beta)C_c}{C_{Qs} + C_{Qd} + C_{tot}} \quad (2-27)$$

$$C_{gd} = \frac{L_g C_{ox}}{2} \frac{C_{Qs} + C_{Qd} + 2\beta C_c}{C_{Qs} + C_{Qd} + C_{tot}} \quad (2-28)$$

$$C_{bs} = C_{sg} \frac{C_{sub}}{C_{ox}} \quad (2-29)$$

$$C_{bd} = C_{dg} \frac{C_{sub}}{C_{ox}} \quad (2-30)$$

$$C_{gb} = \frac{L_g C_{ox} C_{sub}}{C_{Qs} + C_{Qd} + C_{tot}} \quad (2-31)$$

Where,

- C_{tot} is the total electrostatic coupling capacitance per unit length between channel and other electrodes given by $C_{tot} = C_{ox} + C_{sub} + C_c$;
- C_{Qs} and C_{Qd} are the quantum capacitance due to the carriers from source and drain respectively; they are expressed as:

$$C_{Qs} = \frac{4q^2}{L_g kT} \sum_{k_{m,m=m0}}^{M} \sum_{k_{l,l=0}}^{L} \left[\frac{e^{(E_{m,l} - \Delta\Phi_B)/kT}}{\left(1 + e^{(E_{m,l} - \Delta\Phi_B)/kT}\right)^2} \right] \quad (2-32)$$

$$C_{Qd} = \frac{4q^2}{L_g kT} \sum_{k_{m,m=m0}}^{M} \sum_{k_{l,l=0}}^{L} \left[\frac{e^{(E_{m,l} + qV_{ch,DS} - \Delta\Phi_B)/kT}}{\left(1 + e^{(E_{m,l} + qV_{ch,DS} - \Delta\Phi_B)/kT}\right)^2} \right] \quad (2-33)$$

Where, m0 is as expressed in equation (2-14).

2.2.3 Source/Drain Extension Model

This model is named as Level 2 (CNFET_L2) and is shown in Figure 2.7. It builds upon Level 1 by including parasitic capacitances and resistances from the highly doped source/drain extension regions; and SB resistances from the source/drain contacts to the source/drain SWCNT regions. This model includes [17]:

1) elastic scattering in the channel ($V_{ch,el}$);
2) the source/drain extension region parasitic capacitances and resistances (C_{gss}, C_{bss}, C_{gdd}, C_{bdd}, $R_{metal,s}$, $R_{semi,s}$, $R_{metal,d}$, $R_{semi,d}$); and
3) the Schottky Barrier resistance due to source/drain metal contacts (R_{sb}).

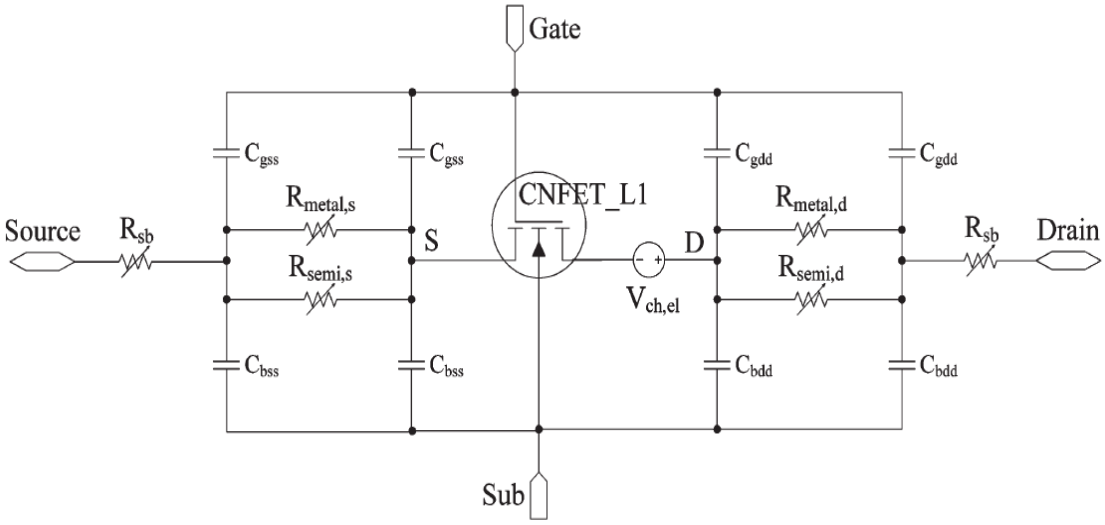

Figure 2.7 Equivalent Circuit Model for the CNFET_L2 [17].

2.2.3.1 $V_{ch,el}$ Calculation

The total potential drop (V_{DS}) across the channel region is a summation of the potential drop ($V_{ch,DS}$) due to the channel quantum resistance $R_{ch,c}$ and the potential drop ($V_{ch,el}$) over the channel resistance $R_{ch,el}$ due to the elastic scattering [17], i.e.,

$$V_{DS} = V_{ch,DS} + V_{ch,el} \tag{2 - 34}$$

Where,

- $V_{ch,DS} = I_{DS}R_{ch,c}$,
- $V_{ch,el} = I_{DS}R_{ch,el}$
- $R_{ch,el} = R_{ch,c}\ (1 - T_{ch})/T_{ch}$ ·, where T_{ch} is the transmission probability in the channel region given by, $T_{ch} = l_{eff}/(L_g + l_{eff})$.

We can assume that the effective elastic-scattering MFP l_{eff} is linearly proportional to the diameter of the CNT, i.e., $l_{eff} = \lambda_{eff}*D_{CNT}/(1.5\ nm)$, where λ_{eff} (~200 nm) is the elastic-scattering MFP for 1.5 nm in diameter CNT. Hence, the potential drop over $R_{ch,el}$ can be represented as,

$$V_{ch,el} = V_{DS}\ \frac{L_g}{L_g + \frac{D_{CNT}}{1.5nm}\lambda_{eff}} \tag{2 - 35}$$

2.2.3.2 Calculation of the Parasitic Resistances

The total effective resistance of the doped S/D region with the length of the doped source region L_s and length of the doped drain region L_d can be obtained by [14, 17],

$$R_{x,s} = \frac{L_s}{G_{x,c}\lambda_{sd}} \tag{2-36}$$

$$R_{x,d} = \frac{\eta_{deff}}{G_{x,c}} \tag{2-37}$$

where,

$$\eta_{deff} = \begin{cases} \dfrac{L_d - \lambda_{sd}}{\lambda_{sd}}, & D_{out} = 0 \\ \dfrac{L_d}{\lambda_{sd}}, & D_{out} = 1 \end{cases} \tag{2-38}$$

$$G_{semi,c}(V_c) = \frac{4q^2}{h} \sum_{k_m,\ m=1}^{2} \left[1 + \frac{kT}{qV_c} ln\left(\frac{1 + e^{\left(E_{m,0} - E_f - \Delta\Phi_s\right)/kT}}{1 + e^{\left(E_{m,0} - E_f + qV_c - \Delta\Phi_s\right)/kT}} \right) \right] \tag{2-39}$$

$$\Delta\Phi_s(V_c) = \Delta\Phi_{s,max} \frac{min\left(V_c, E_f - E_{1,0} + \Delta\Phi_{s,max}\right)}{E_f - E_{1,0} + \Delta\Phi_{s,max}} \tag{2-40}$$

$$\Delta\Phi_{s,max} \approx \begin{cases} \sqrt{Q_{Ef}^2 + E_{1,0}^2} - E_f, & E_{1,0} < E_f + \Delta\Phi_{s,max} < E_{2,0} \\ \dfrac{\sqrt{\left(E_{2,0}^2 - E_{1,0}^2\right)^2 + 2\left(E_{2,0}^2 + E_{1,0}^2\right)Q_{Ef}^2 + Q_{Ef}^4}}{2Q_{Ef}} - E_f, & E_{2,0} \le E_f + \Delta\Phi_{s,max} < E_{3,0} \end{cases} \tag{2-41}$$

$$Q_{Ef} \approx \begin{cases} 2\sqrt{E_f^2 - E_{1,0}^2}, & E_{1,0} < E_f < E_{2,0} \\ 2\sqrt{E_f^2 - E_{1,0}^2} + 2\sqrt{E_f^2 - E_{2,0}^2}, & E_{2,0} \le E_f < E_{3,0} \end{cases} \tag{2-42}$$

In Equations 2-36 and 2-37, the subscript x denotes either "semi" or "metal". λ_{sd} is the impurity scattering MFP, assumed a constant with a default value of 15 nm, for degenerately doped nanotubes. $G_{x,c}$ is the quantum conductance of doped CNT. It depends on CNT diameter, the doping level (E_f), and the S/D Fermi-level difference ($qV_c = |\mu_s - \mu_d|$). For heavily doped nanotubes of typical CNFETs, it is assumed that the doping level E_f is above the first semiconducting subband but does not exceed the third semiconducting subband.

With an applied bias at the two ends, the S/D Fermi levels split apart by V_c, which results in carrier redistribution between +k states and −k states.

In Equation 2-38, D_{out} representing the drain connectivity is equal to 0 if drain is connected to doped CNT; otherwise, it is equal to 1.

Equation 2-40 can be used to obtain the surface potential change ($\Delta\Phi_s$) which depends on the maximum surface potential change ($\Delta\Phi_{s,max}$) shown on Equation 2-41. Q_{Ef} is normalized number of carriers given by Equation 2-42.

2.2.3.3 Calculation of the Parasitic Capacitances

The four equivalent capacitances in Figure 2.7 can be expressed by [14, 17],

$$C_{gzz} = \frac{L_z C_Q C_{of}}{2\left(C_Q + C_{of} + C_{sub}\right)} \qquad (2-43)$$

$$C_{bzz} = \frac{L_z C_Q C_{sub}}{2\left(C_Q + C_{of} + C_{sub}\right)} \qquad (2-44)$$

Where,

- The subscript "z" denotes either "s" for source or "d" for drain.
- C_Q is the quantum capacitance of the doped S/D region. It can be approximated to the first order as,

  $$C_Q = [\Theta(E_f - E_{1,0}) + \Theta(E_f - E_{2,0})] \times 400 \text{ aF/μm} \qquad (2 - 45)$$

  where $\Theta(x)$ is a step function that is equal to 1 if $x > 0$; otherwise, it is equal to 0.
- C_{of} is the gate outer-fringe capacitance.

For heavily doped CNTs with the first or two subbands populated, C_Q (400 ~ 800 aF/μm) >> C_{of} (~ 30 aF/μm) > C_{sub} (~ 20 aF/μm), thus it is safe to ignore C_Q for most applications [14].

2.2.3.4 Calculation of the SB-Resistances

A simplified model to describe the series Schottky-barrier (SB) resistances between doped CNTs and metallic electrodes has been devised as [14, 17],

$$R_{sb} = \frac{1}{G_{semi,c}} \left(\frac{1}{T_{SB}} - 1\right)(1 - X_{out}) \qquad (2-46)$$

Where,

- The symbol "X" denotes either "S" (source-side SB) or "D" (drain-side SB). The two parameters S_{out} (D_{out}) represent the source/drain connectivity: equal to 0 if source (drain) is connected to doped CNT, otherwise equal to 1.
- T_{SB} is the average transmission probability through the first subband expressed as,

$$T_{SB} \approx 0.5\left(\exp\left(-\tau \cdot \Phi_1^{3/2}\right) + \exp\left(-\tau \cdot \Phi_2^{3/2}\right)\right) \qquad (2-47)$$

$$\tau \approx \frac{4\sqrt{2}W_2}{9r\Phi_2\sqrt{E_{1,0}}} \qquad (2-48)$$

$$\Phi_1 = \Phi_M - \Phi_C + E_{1,0} \qquad (2-49)$$

$$\Phi_2 = \begin{cases} \Phi_1 + |V_{sb,s}| - \Delta\Phi_S, & Source - SB \\ \Phi_1 - |V_{sb,d}| - V_c + \Delta\Phi_S, & \text{Drain} - \text{SB} \end{cases} \qquad (2-50)$$

$$W_2 = \left(\sqrt{V_{bi}} - \sqrt{E_f - E_{1,0}}\right) \cdot \sqrt{\frac{2\varepsilon_0 k_2}{eN_D}} \qquad (2-51)$$

Where,

- Φ_1 is the potential barrier seen by the carriers at the metal contact side,
- Φ_2 is the potential barrier seen by the carriers at the doped CNT side,
- $V_{sb,s}$ is the potential drop over the equivalent SB resistor at the source side,
- $V_{sb,d}$ is the potential drop over the equivalent SB resistor at the drain side,
- Φ_M is the metal work function,
- Φ_C is the CNT work function,
- E_f is doping level,
- N_D is doping density,
- V_{bi} is the build in potential with an applied bias,
- W_2 is the depletion length after bias.

Figure 2.8 shows SB resistances as functions of the current, with different metal/CNT work functions and different CNT doping levels. It can be observed that both smaller barrier height and higher CNT doping level help to reduce SB resistance significantly.

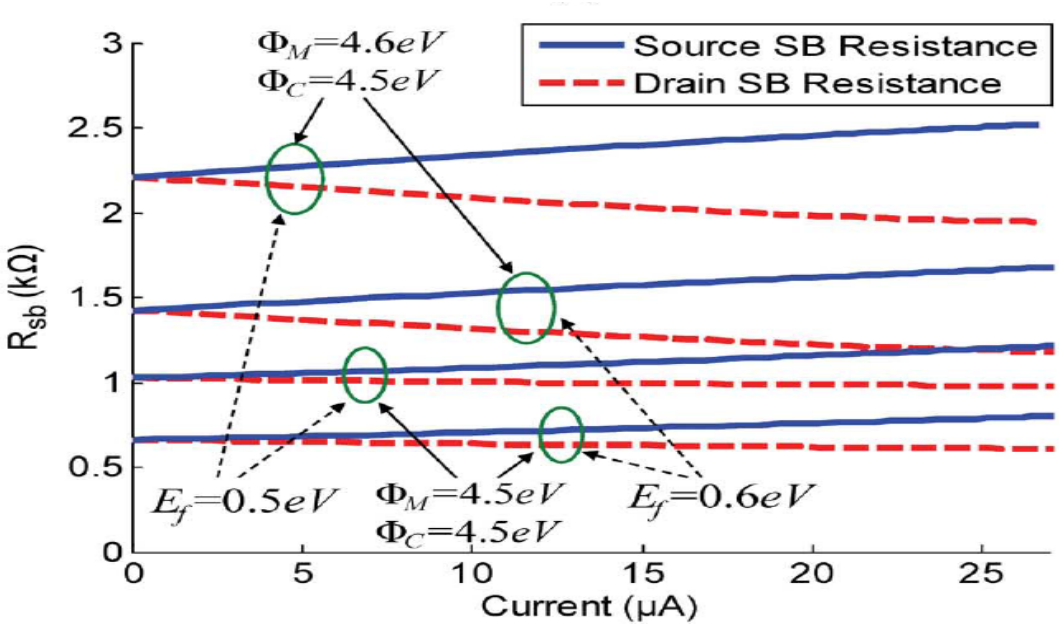

Figure 2.8 SB Resistances as Functions of the Current [17].

2.2.4 Multiple Nanotubes Model

This model, shown in Figure 2.9, is the complete model of the transistor and is referred to as Level 3 (CNFET_L3). It augments Level 2 and deals with multiple SWCNTs per device. This model includes the channel elastic scattering, the doped source/drain region, SB resistance, multiple CNTs per device and other device/circuit non-idealities. It includes the parasitic gate capacitance and the SWCNT-to-SWCNT charge screening effects for the SWCNTs within the SWCNT-FET Device. The model includes CNFET_L2 and augments it by accounting for [8, 17]:

1. inter-SWCNT charge screening by considering two cases (edge and middle) of screening effects, and
2. parasitic gate capacitances.

Considering that there are N CNTs under the gate. The CNTs can be grouped into two:

- a number of min(N, 2) CNTs at the two edges and
- the other (N − min(N, 2)) CNTs in the middle.

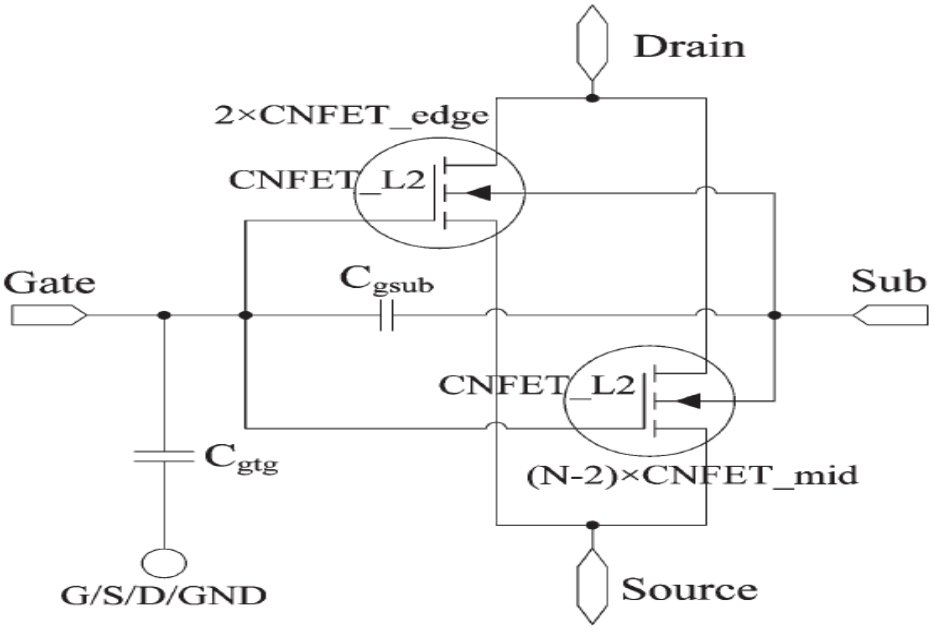

Figure 2.9 Circuit Schematic of the CNFET_L3 [17].

The gate outer-fringe capacitance (C_{of}) for multiple CNTs per gate can be obtained as the summation of the fringe capacitance between the gate and S/D CNT at the two edges (C_{of_e}) and the fringe capacitance between the gate and S/D CNT in the middle (C_{of_m}) [18].

$$C_{of} = min(N, 2) \cdot C_{of\_e} + max(N - 2, 0) \cdot C_{of\_m} \qquad (2-52)$$

Where,

$$C_{of\_e} = \frac{\pi k_2 \varepsilon_0 L_{sd}}{ln\left(\frac{\sqrt{(2h)^2 + (0.56L_{sd})^2 + s^2}}{s}\right) + \eta_1 cosh^{-1}\left(\frac{\sqrt{(2h)^2 + (0.56L_{sd})^2}}{d}\right)} \qquad (2-53)$$

$$C_{of\_m} = \frac{2\alpha}{\eta_1} \cdot C_{of\_e} + \left(1 - \frac{2\alpha}{\eta_1}\right) \cdot \frac{\pi k_2 \varepsilon_0 L_{sd}}{cosh^{-1}\left(\frac{\sqrt{(2h)^2 + (0.56L_{sd})^2}}{d}\right)} \qquad (2-54)$$

$$\eta_1 = exp\left(\frac{\sqrt{N^2 - 2N} + N - 2}{\tau_1 N}\right), \quad N \geq 2 \qquad (2-55)$$

$$\alpha = exp\left(\frac{N - 3}{\tau_2 N}\right), \quad N \geq 3 \qquad (2-56)$$

where,

- N is the number of cylinders per gate,
- τ_1 and τ_2 are fitting parameters that describe how fast the electric flux of the adjacent cylinders decreases with increasing distance,
- k_2 is the relative permittivity of the dielectric material in the bulk region which is equal to 3.9 for SiO_2,
- ε_0 is the permittivity of free space of value 8.85418×10^{-14} F/cm,
- h, s, d and L_{sd} are dimensions as indicated in Figure 2.10 and Figure 2.11 below.

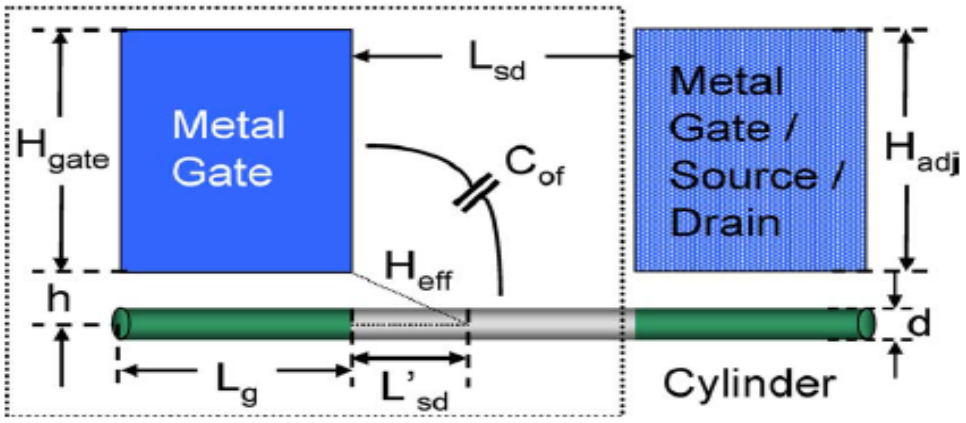

Figure 2.10 Structure Used to Calculate the Gate Outer Fringe Capacitance [18].

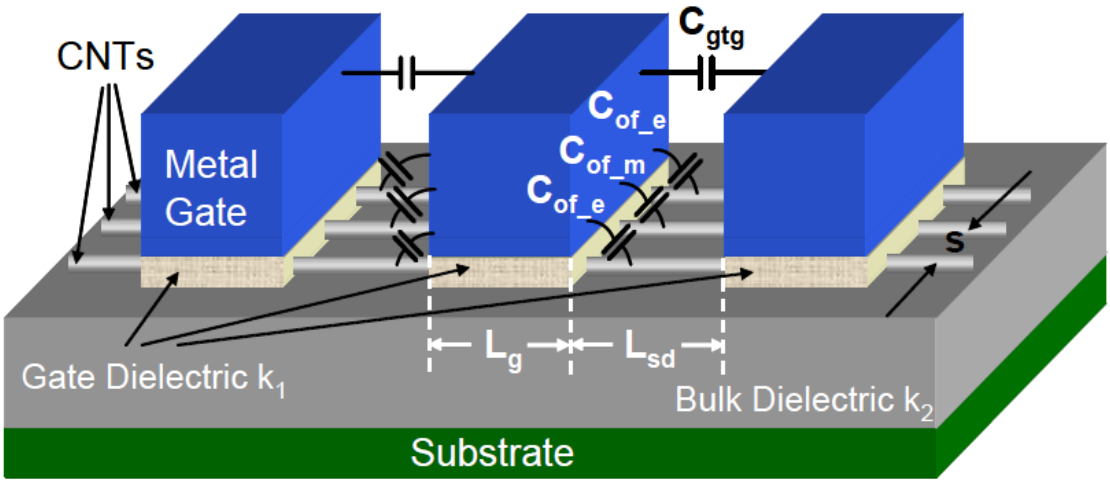

Figure 2.11 3-D Device Structure of CNFETs with Multiple Channels [18].

The capacitance between the gate and channel (C_{ox}) for multiple CNTs per gate can be grouped into the capacitance between the gate and SWCNT at the two edges (C_{ox_e}) and the capacitance between the gate and SWCNT in the middle (C_{ox_m}) [18].

$$C_{ox} = min(N, 2) \cdot C_{ox\_e} + max(N - 2,0) \cdot C_{ox\_m} \quad (2-57)$$

Where,

$$C_{ox\_e} = \frac{C_{gc\_inf} \cdot C_{gc\_sr}}{C_{gc\_inf} + C_{gc\_sr}} \quad (2-58)$$

$$C_{ox\_m} = 2C_{ox\_e} - C_{gc\_inf} \quad (2-59)$$

$$C_{gc\_sr} = \frac{4\pi k_1 \varepsilon_0}{ln\left(\frac{s^2 + 2(h-r) \cdot [h + \sqrt{h^2 - r^2}]}{s^2 + 2(h-r) \cdot [h - \sqrt{h^2 - r^2}]}\right) + \lambda_1 \cdot ln\left(\frac{(h+d)^2 + s^2}{9r^2 + s^2}\right) \cdot tanh\left(\frac{h+r}{s-d}\right)} \quad (2-60)$$

$$C_{gc\_inf} = \frac{2\pi k_1 \varepsilon_0}{cosh^{-1}\left(\frac{2h}{d}\right) + \lambda_1 \cdot ln\left(\frac{2h + 2d}{3d}\right)} \quad (2-61)$$

$$\lambda_1 = \frac{k_1 - k_2}{k_1 + k_2} \quad (2-62)$$

where,

- N is the number of CNTs per gate,
- r is the radius of the CNT,
- k_1 is the relative permittivity of the gate dielectric material,
- k_2 is the relative permittivity of the dielectric material in the bulk region,
- ε_0 is the permittivity of free space of value 8.85418×10^{-14} F/cm,
- h, s, d and L_{sd} are dimensions as indicated in Figure 2.10 and Figure 2.11.

Referring to Figure 2.10 and Figure 2.11 the coupling capacitance (C_{gsub}) between the gate and the substrate and the gate-to-gate (or gate-to-S/D) coupling capacitance per unit length (C_{gtg}) can be expressed as [17],

$$C_{gsub} \approx \frac{2\pi k_2 \varepsilon_0 L_g}{\ln\left(\frac{4H_{sub}}{H_{gate}}\right)} \quad (2-63)$$

$$C_{gtg} = \frac{k_2 \varepsilon_0 H_{gate}}{L_{sd}} + \alpha_{gtg\_sr} \cdot \frac{\pi k_2 \varepsilon_0}{\ln\left(\frac{2\pi(L_{sd} + L_g)}{2L_g + \tau_{bk} H_{gate}}\right)} \qquad (2-64)$$

where,

- H_{sub} is the insulating bulk thickness,
- H_{gate} is the gate height,
- α_{gtg_sr} is the parameter due to the screening of the adjacent gate (or S/D electrode) and interconnects,
- τ_{bk} is the factor that accounts for the effects of the back plates given by,

$$\tau_{bk} = \exp\left(2 - 2\sqrt{1 + \frac{2(H_{gate} + L_g)}{L_{sd}}}\right). \qquad (2-65)$$

CHAPTER 3
STATIC RANDOM ACCESS MEMORY

3.1 Semiconductor Memories

In electronics a memory is a mechanism that stores data for use. Most memories represent data with the binary number system. In the binary number system, numbers are represented by sequences of the two binary digits 0 and 1, which are called bits. In a computer, the two possible values of a bit correspond to the on and off states of the computer's electronic circuitry. A group of eight bits is called a byte. Memory capacity is usually quantified in terms of kilobytes, megabytes, and gigabytes.

One of the semiconductor-based memories that can be read and written by the microprocessor or other hardware devices is random access memory (RAM). The word random implies that information stored in RAM can be accessed in any order, and may be erased or written over. RAM is generally understood to refer to volatile memory since the information is lost after the power is turned off [19].

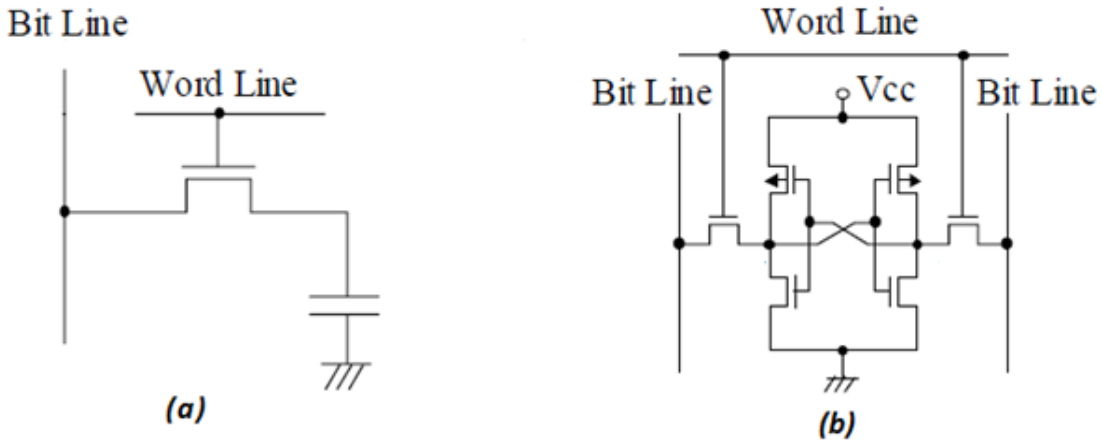

Figure 3.1 (a) One Transistor Cell of DRAM (b) Six Transistor Cell of SRAM [20].

The main types of RAMs are the static random access memory (SRAM) and dynamic random access memory (DRAM). Generally SRAMs store a bit of data in either state of a flip-flop, while DRAMs store a bit as a charge in a capacitor or a transistor gate. The capacitor charge needs to be refreshed to maintain the information since there is

leakage. Because of this refreshment DRAM is dynamic memory. But SRAM can retain the stored data without any need of periodic refreshment thus it is static memory. DRAM chips are denser than SRAM chips due to their structural simplicity. DRAM is the main RAM used in personal computers, workstations, and servers [5].

There is a growing gap between the processor cycle time and DRAM access time which in turn necessitated the introduction of several levels of caching in modern data processors. In personal computer processors such levels are often represented by L1 and L2 on-chip embedded SRAM cache memories. As the speed gap between processor, memory and mass storage continues to widen, deeper memory hierarchies have been introduced in high-end server microprocessors [21].

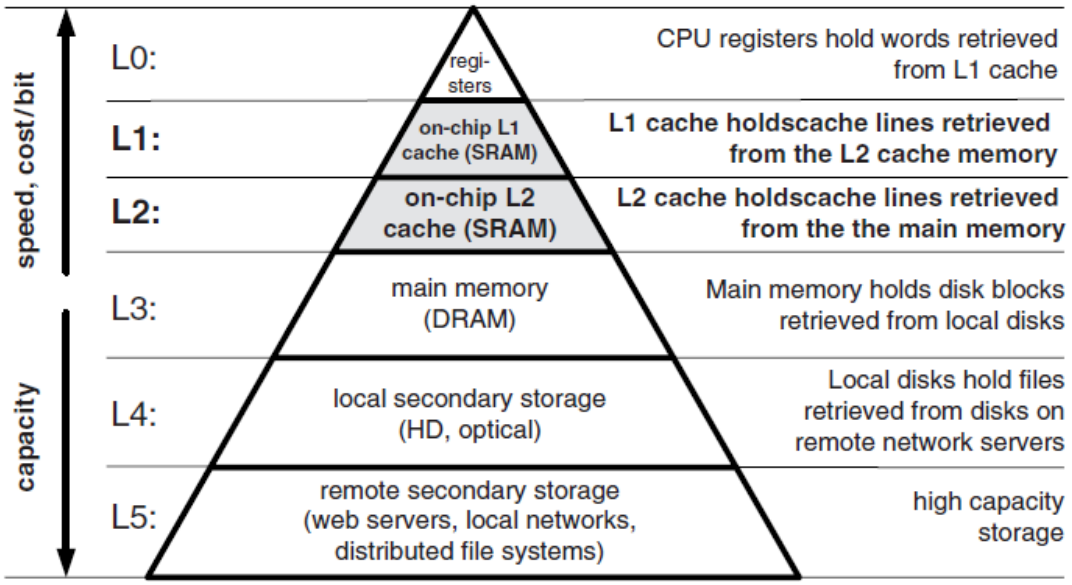

Figure 3.2 Typical Memory Hierarchy of a Personal Computer [21].

3.2 SRAM

The basic architecture of an SRAM includes one or more rectangular arrays of memory cells with support circuitry to decode addresses, and implement the required read and write operations. SRAM memory arrays are arranged in rows and columns of memory cells called wordlines and bitlines, respectively. Each memory cell in a plane has a unique

location or address defined by the intersection of a row and column. Each address is linked to a particular data input/output (I/O) pin.

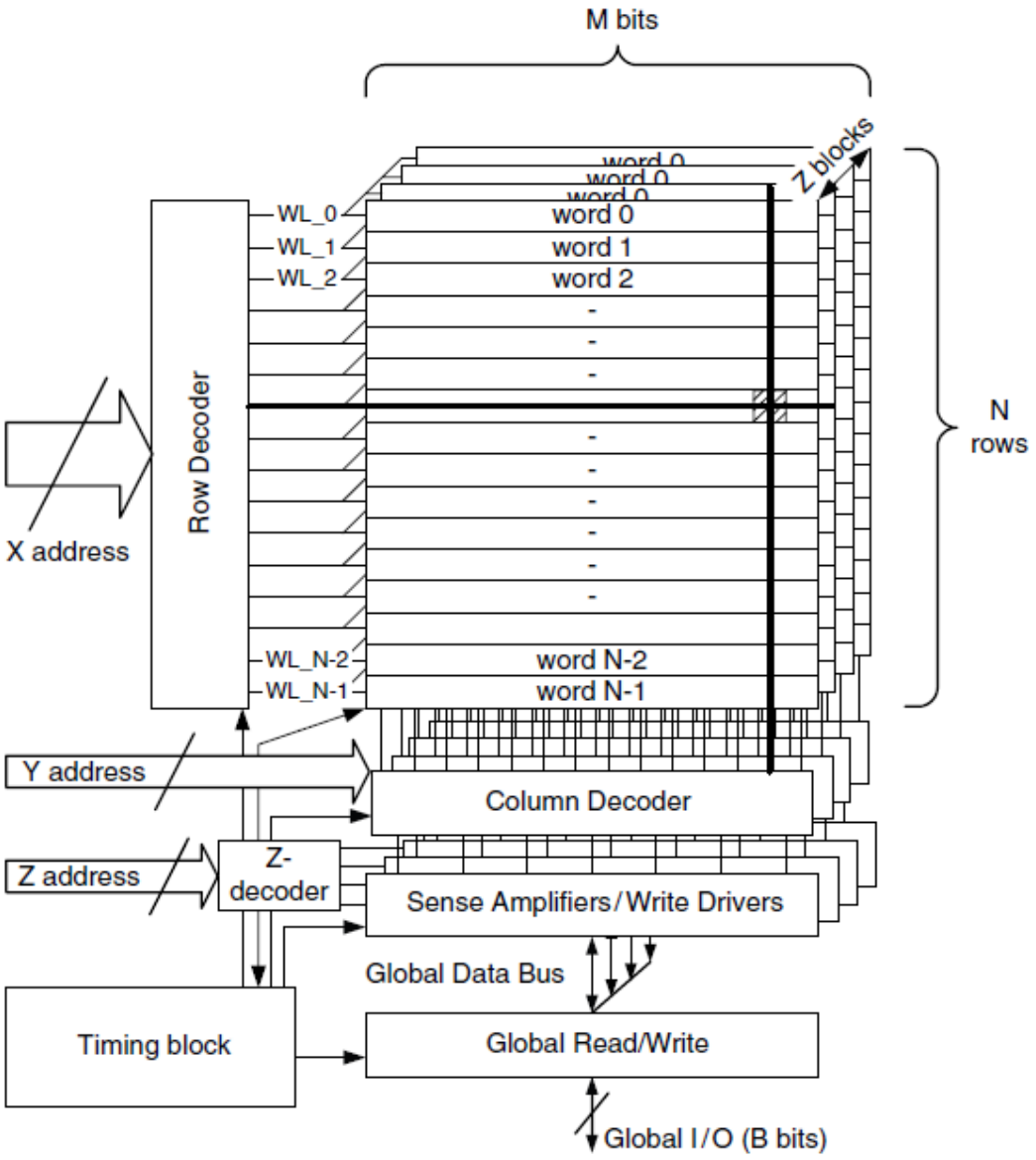

Figure 3.3 Typical SRAM Block Diagram [21].

The number of arrays on a memory chip is determined by the total size of the memory, the speed at which the memory must operate, layout, and the number of data I/Os on the chip. SRAMs can be organized as bit-oriented or word-oriented. In a bit-oriented SRAM, each address accesses a single bit, whereas in a word-oriented memory, each

address addresses a word of n bits (where the popular values of n include 8, 16, 32 or 64) [21].

Figure 3.3 shows a basic block diagram of an SRAM with four pages of NxM arrays with the corresponding I/O blocks. The SRAM core consists of a number of arrays of NxM, where N is the number of rows and M is the number of bits. A row decoder gated by appropriate timing block signal decodes X row address bits and selects one of the word lines WL_0 – WL_N-1. If an SRAM core is organized as a number of arrays in a page manner, an additional Z-decoder is needed to select the accessed page. Column decoders or column multiplexers addressed by Y address bits allow sharing of a single sense amplifier among 2, 4 or more columns. An additional Chip Select (CS) signal, introducing an extra decoding hierarchy level, is often provided in multi-SRAM chip architectures [21].

3.2.1 Memory Cell

An SRAM memory cell is a bi-stable flip-flop usually made up of four to ten transistors. The flip-flop may be in either of two states that can be interpreted by the support circuitry to be a 1 or a 0. The most common SRAM cells on the market include a silicon-based four transistor cell with a polysilicon load and a silicon-based six transistor cell. The four transistor cell with a polysilicon load is suitable for medium to high performance, this design has a relatively high leakage current, and consequently high standby current. While a six transistor memory cell is highly stable and has low leakage and standby currents. Figure 3.4 shows typical SRAM memory cells.

3.2.2 Support Circuitry

The support circuitry of the SRAM is used to read the data stored in the memory's cells, and write data to the cells. This circuitry generally includes: address logic to select rows and columns; translation logic that reads the data in a cell and sends that data to the data I/O; write logic that takes data applied at the input and stores it in a memory cell; and other control circuitries on the chip.

3.2.3 SRAM Operations

An SRAM cell can be operated in three different states: standby, reading or writing. In standby mode the cell is disconnected from the bit lines but the stored information will be

maintained by the cross coupled inverters. In reading mode the stored information will be accessed via the bit lines and delivered to the I/O port by the help of sense amplifier. The sense amplifier amplifies a small analog differential voltage developed on the bit lines by a read-accessed cell to the full swing digital output signal thus greatly reducing the time required for a read operation. Since SRAMs do not feature data refresh after sensing, the sensing operation must be nondestructive. In writing mode the data in the cell is updated by the input drivers.

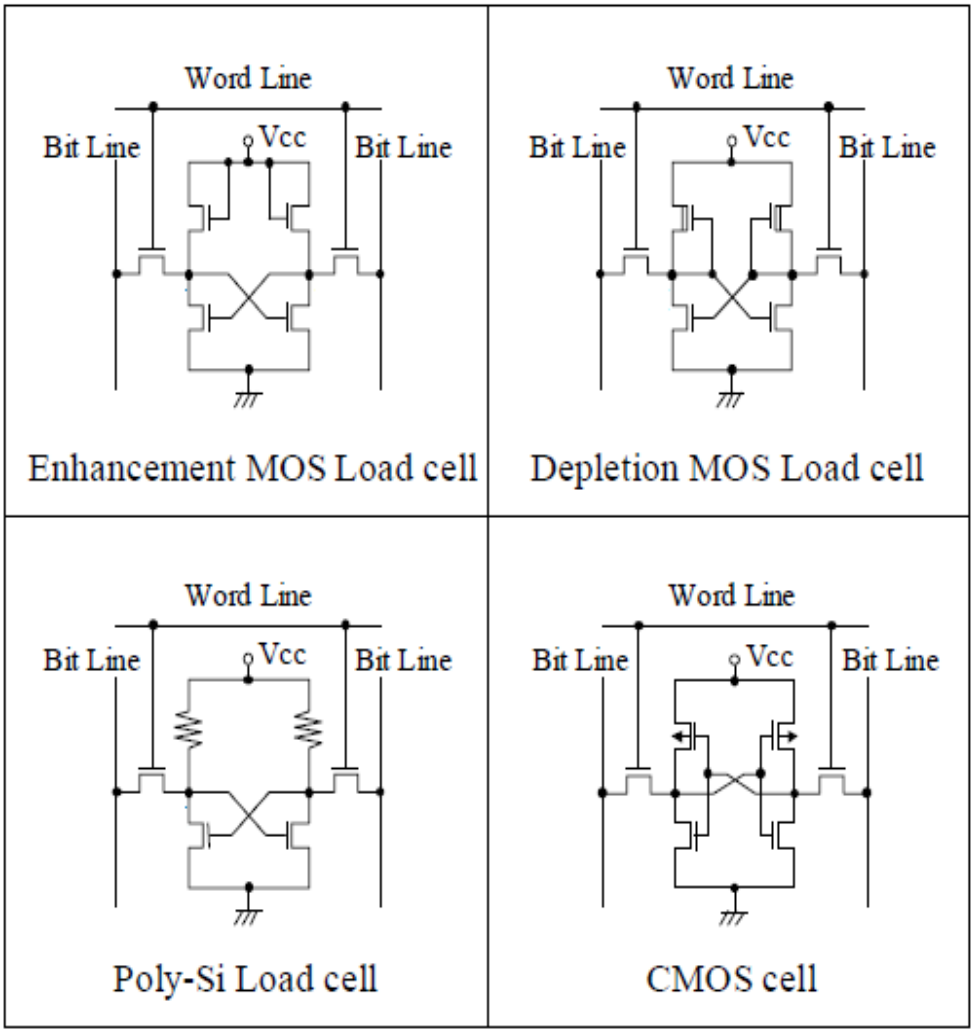

Figure 3.4 Examples of SRAM Cells [20].

3.3 SRAM Design Trends

The process technology scaling for better performance enabled embedding of millions of SRAM cells into contemporary ICs. In several applications, the embedded SRAMs can occupy the majority of the chip area and contain hundreds of millions of transistors [21]. In the early days, the development of SRAMs was focused on low-power applications, especially with very low standby and data-retention power, while increasing memory capacity with high-density technology [22]. After that, however, more emphasis had been placed on high speed rather than large memory capacity, primarily led by cache applications in high-speed micro-processor units (MPUs). These days, the memory capacity is increasing again, reaching more than 200 Mb cache [22].

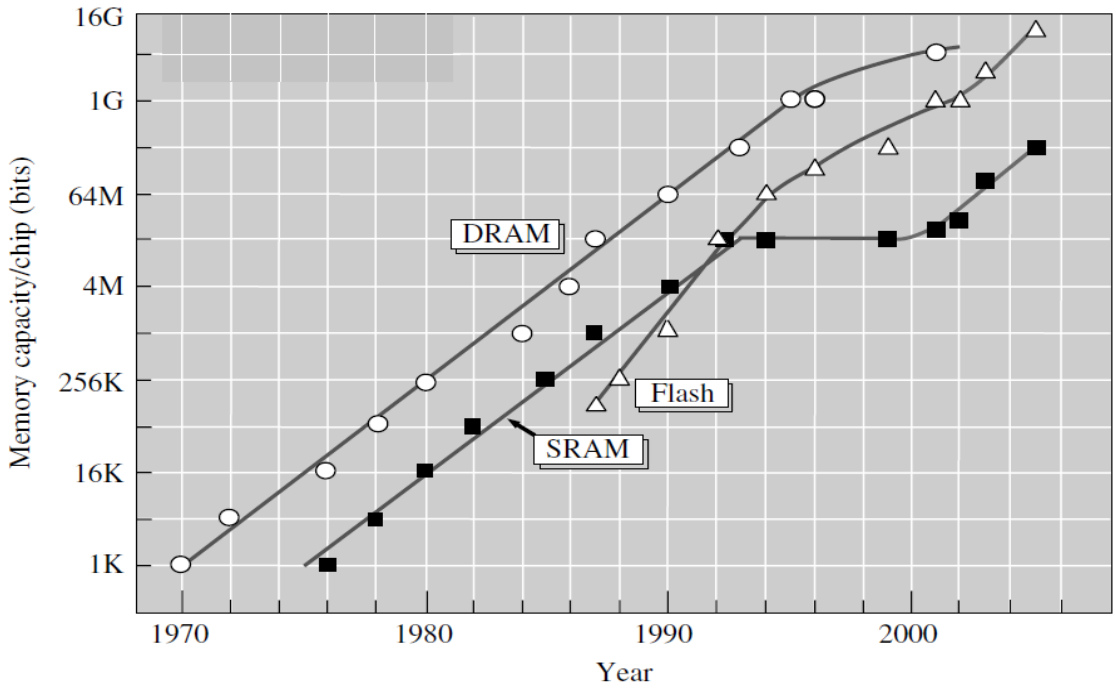

Figure 3.5 Trends in the Memory Capacity of VLSI Memories [22].

As the process technology continues to scale down, the stability of embedded SRAMs is a growing concern for designers. Large SRAM arrays that are widely used as cache memory in microprocessors and ASICs can occupy a significant portion of the chip area. In an attempt to optimize the performance/cost ratio of such chips, designers are faced with a dilemma [21],

- Large arrays of fast SRAM help to boost the system performance.
- The area impact of incorporating large SRAM arrays into a chip directly translates into a higher chip cost.

Balancing these requirements is driving the effort to minimize the footprint of SRAM cells. As a result, millions of minimum-size SRAM cells are tightly packed making SRAM arrays the densest circuitry on a chip [21]. Such areas on the chip can be especially susceptible and sensitive to manufacturing defects and process variations. In 2004 International Technology Roadmap for Semiconductors (ITRS) predicted "greater parametric yield loss with respect to noise margins for high density circuits such as SRAM arrays, which are projected to occupy more than 90% of the chip area in 2013" [21, 23]. From Figure 3.5 it can be learnt that there has been tremendous increase in the memory capacity per chip of the semiconductor memories every year except for SRAMs which have stayed with no increment of memory size per chip from 1993 to 2000. Figure 3.6 shows the technology trend in the area per cell of the semiconductor memories as predicted by ITRS in 2008.

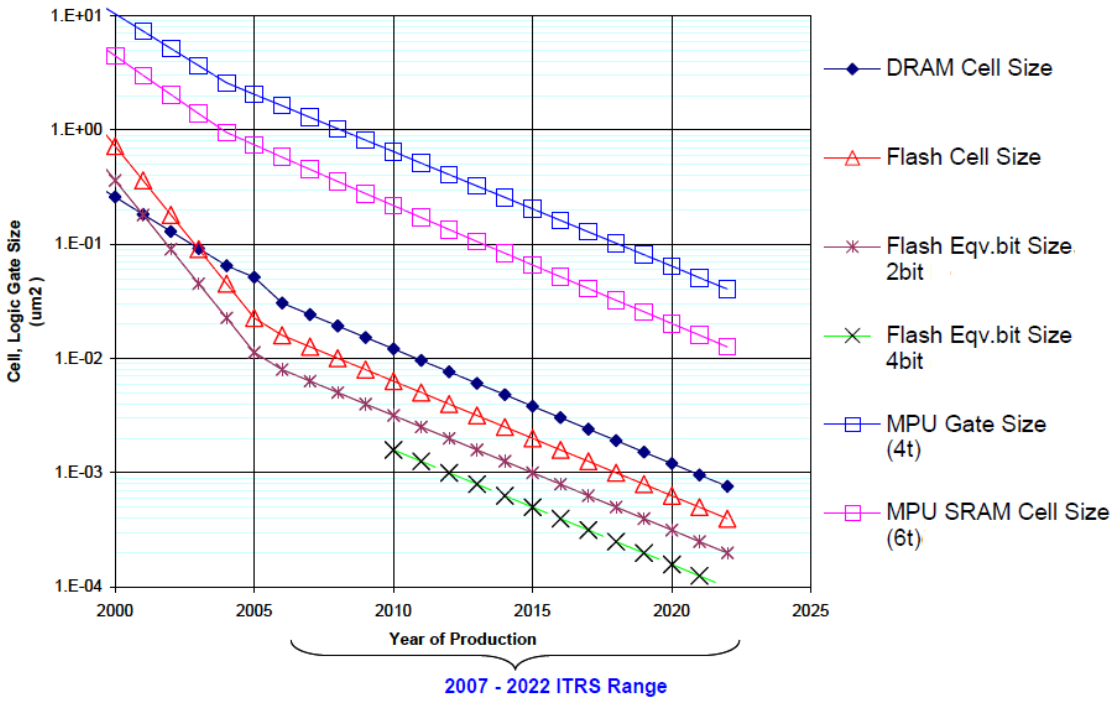

Figure 3.6 ITRS Product Technology Trends [24].

3.3.1 Static Noise Margin

Noise Margin (NM) is the maximum spurious signal that can be accepted by the device when used in a system while still maintaining the correct operation. A Static Noise Margin (SNM) is implied if the noise is a DC source. It is assumed that noise is present long enough for the circuit to react, i.e. the noise is "static" or DC.

Referring to Figure 3.7 the noise margin high (NM_H) and noise margin low (NM_L) can be defined as [21]:

$$NM_H = V_{OH} - V_{IH} \quad (3 - 1)$$

$$NM_L = V_{IL} - V_{OL} \quad (3 - 2)$$

where,

- V_{IL} is the maximum input voltage level recognized as logical "0",
- V_{IH} is the minimum input voltage level recognized as a logical "1",
- V_{OL} is the maximum logical "0" output voltage,
- V_{OH} is the minimum logical "1" output voltage.

Any inverter transfer curve, which falls into the shaded area, will have noise margins at least as good as given by the equations above. Input voltage in the range of $V_{IL} < V_{in} < V_{IH}$ may not be properly recognized by the gate and may cause a logic error.

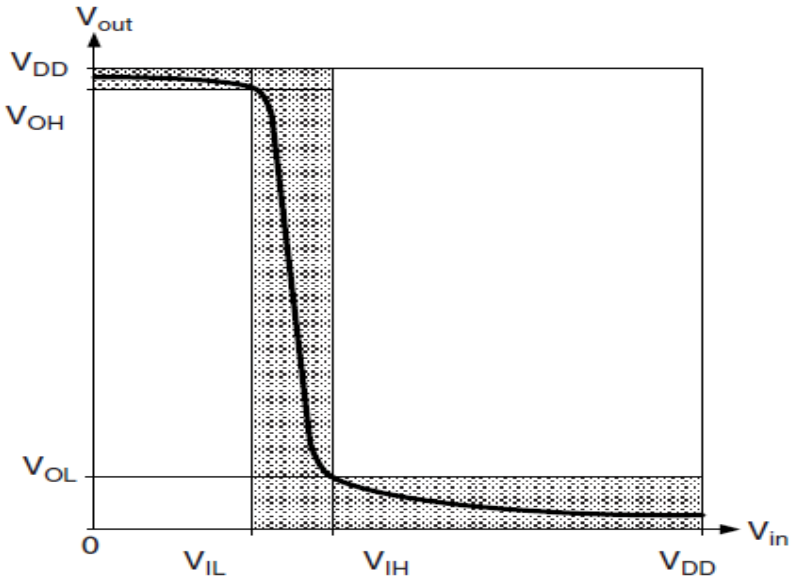

Figure 3.7 Inverter Transfer Curve [21].

3.3.2 Stability Test Setups

An accurate and consistent representation of unstable behavior of an SRAM cell is essential for SNM modeling and evaluation of circuit techniques for SRAM cell stability test. Since the SNM is a measure of SRAM cell stability, its degraded value results in cell stability fault. A stability fault model, illustrated in Figure 3.8, considers the dependence of the SNM on the resistance between node A and node B. The resistor between node A and node B represents cell stability fault model. This SRAM cell has the worst-case SNM in the read-access mode when both the bit lines are precharged and the word line is activated [21]. Each half of a read-accessed SRAM cell can be represented as an equivalent inverter, as shown in Figure 3.8(b). In a simulation environment, a cell with a resistor of a specified value between node A and node B can imitate a weak cell with a specified SNM value.

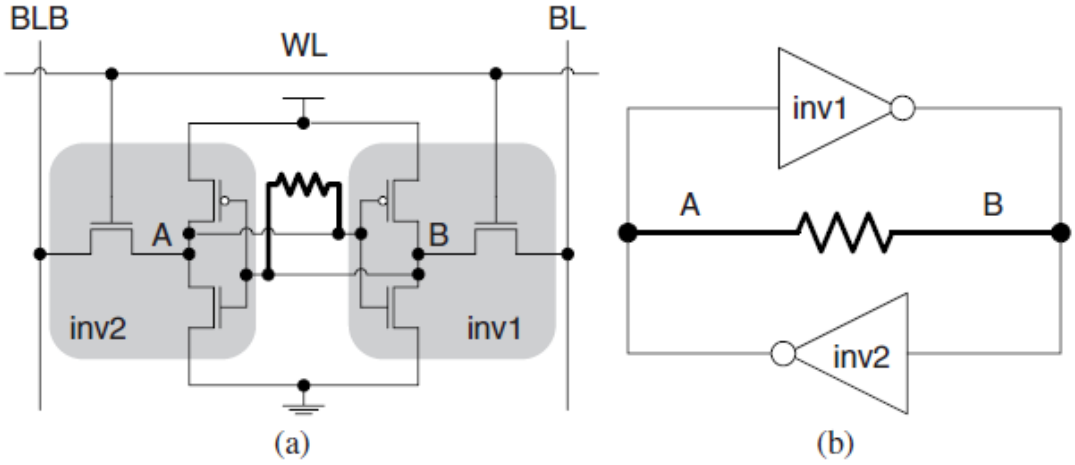

Figure 3.8 (a) Weak Cell Fault Model; (b) Its Equivalent Circuit [21].

In another approach an SRAM cell is presented as two equivalent inverters with the noise sources inserted between the corresponding inputs and outputs. Figure 3.9 shows read-accessed SRAM cell with inserted adverse polarity static noise sources (V_n). Both series voltage noise sources (V_n) have the same value and act together to upset the state of the cell, i.e. they have an "adverse" polarity to the current state of each inverter of the cell [21]. Having two adverse noise sources applied to the input of each inverter of an SRAM cell makes the value of the obtained SNM to be the worst-case SNM.

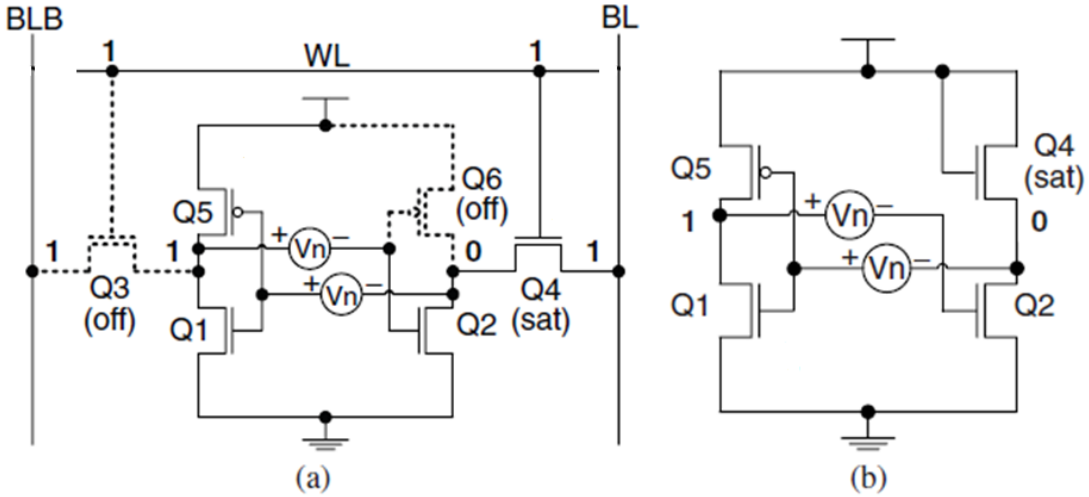

Figure 3.9 (a) SRAM Cell with Inserted Adverse Polarity Static Noise Sources; (b) Its Equivalent Circuit; adapted from [21].

CHAPTER 4
CNFET-BASED SRAM

4.1 CNFET Model

We have selected a type of CNFET known as enhancement-mode MOSFET-like SWCNT-CNFET for designing the SRAM rather than the Schottky Barrier CNFET because MOSFET-like CNFET has lower leakage current, no ambi-polar property, better efficiency of injecting carriers from S/D into the channel and better fabrication feasibility. Even though many physical aspects of the CNFET are captured in the NEGF approach, we have selected the compact model since NEGF involves very intensive calculations and it is very difficult to develop simple intuitive descriptions of the device physics. It is also difficult to use NEGF modeling to rapidly explore device design spaces. The model of the CNFET used here is based on the works by Jie Deng and H. S. Philip Wong [16, 17, 18].

4.1.1 Compact Model

The compact model of the CNFET is based on the ballistic transport assumption and its complete equivalent circuit model is indicated in Figure 4.1. The VHDL-AMS code of this transistor is also developed by mixed structural and behavioral description as can be understood from the code shown in Section 4.1.2, and Appendices D and E.

The ports of the transistor are indicated as SOURCE, DRAIN, GATE and BULK or substrate in Figure 4.1. The notation for the capacitances can be explained as follows:

Cbd = capacitance between the bulk and the drain at level one of the model
Cbs = capacitance between the bulk and the source at level one of the model
Cgb = capacitance between the gate and the bulk at level one of the model
Cgd = capacitance between the gate and the drain at level one of the model
Cgs = capacitance between the gate and the source at level one of the model
Cbdd = capacitance between the bulk and the drain at level two of the model
Cbss = capacitance between the bulk and the source at level two of the model
Cgdd = capacitance between the gate and the drain at level two of the model
Cgss = capacitance between the gate and the source at level two of the model
Cgtg = gate to gate capacitance at level three of the model
Cgsub = capacitance between the gate and the substrate at level three of the model

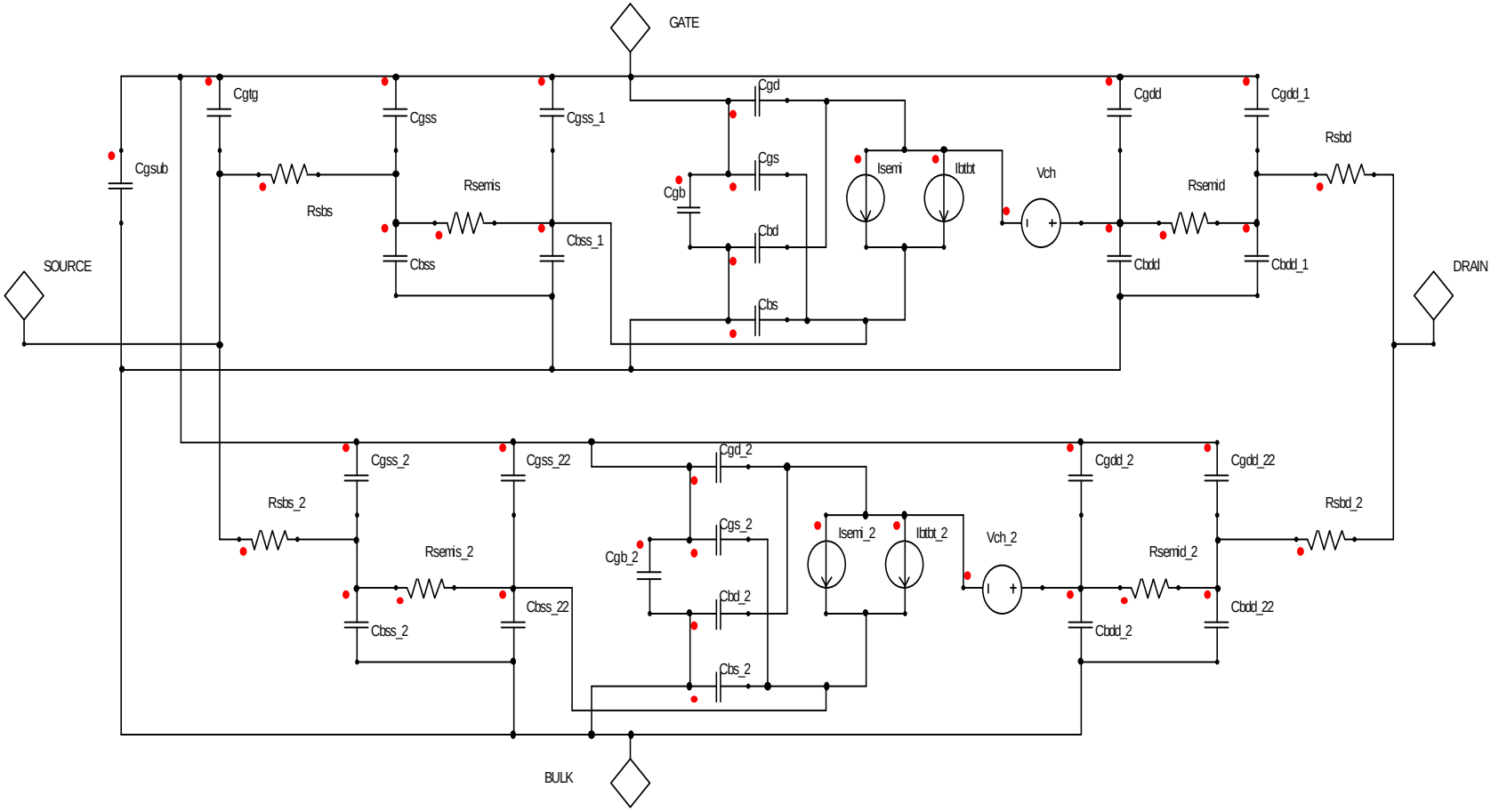

Figure 4.1 Complete Equivalent Circuit Model of the CNFET.

4.1.2 VHDL-AMS Code of the CNFET

```
---------- VHDLAMS MODEL my_ncnfet ----------
LIBRARY ieee;
USE ieee.all;
LIBRARY mgncnfet_lib;
USE mgncnfet_lib.all;
LIBRARY mgncnfet_lib02;
USE mgncnfet_lib02.cntfetpack02.all;
---------- ENTITY DECLARATION my_ncnfet ----------
ENTITY          my_ncnfet IS
   PORT(TERMINAL Drain, Gate, Source, Sub: electrical);
END ENTITY my_ncnfet;
---------- ARCHITECTURE DECLARATION arch_my_ncnfet ----------
ARCHITECTURE arch_my_ncnfet OF my_ncnfet IS
        TERMINAL Source_b: ELECTRICAL;
        TERMINAL Drain_b : ELECTRICAL;
        TERMINAL Drain_ch : ELECTRICAL;
        TERMINAL Source_L1 : ELECTRICAL;
        TERMINAL Drain_L1 : ELECTRICAL;
        TERMINAL Source_L102 : ELECTRICAL;
        TERMINAL Drain_L102 : ELECTRICAL;
        TERMINAL Drain_ch02 : ELECTRICAL;
        TERMINAL Drain_b02 : ELECTRICAL;
        TERMINAL Source_b02 : ELECTRICAL;
        QUANTITY v_sub ACROSS Sub TO ELECTRICAL_REF;
        QUANTITY v_vd ACROSS Drain_L1 TO ELECTRICAL_REF;
        QUANTITY v_vg ACROSS Gate TO ELECTRICAL_REF;
        QUANTITY v_vs ACROSS Source_L1 TO ELECTRICAL_REF;
        QUANTITY v_ddin ACROSS Drain_b TO Drain_ch;
        QUANTITY v_dinsin  ACROSS Drain_ch TO Source_L1;
BEGIN
        c01 : ENTITY mgncnfet_lib.cgss (arch_cgss)
                PORT MAP ( m => Source_b, p => Gate);
        c02 : ENTITY mgncnfet_lib.cgss (arch_cgss)
                PORT MAP ( m => Source_L1, p => Gate);
```

```
c03 : ENTITY mgncnfet_lib.cbss (arch_cbss)
        PORT MAP ( m => Sub, p => Source_b);
c04 : ENTITY mgncnfet_lib.cbss (arch_cbss)
        PORT MAP ( m => Sub, p => Source_L1);
c05 : ENTITY mgncnfet_lib.cgdd (arch_cgdd)
        PORT MAP ( m => Drain_ch, p => Gate);
c06 : ENTITY mgncnfet_lib.cbdd (arch_cbdd)
        PORT MAP ( m => Sub, p => Drain_ch);
c07 : ENTITY mgncnfet_lib.cbdd (arch_cbdd)
        PORT MAP ( m => Sub, p => Drain_b);
c08 : ENTITY mgncnfet_lib.cgdd (arch_cgdd)
        PORT MAP ( m => Drain_b, p => Gate);
r01 : ENTITY mgncnfet_lib.Rsbs (arch_rsbs)
        PORT MAP ( m => Source_b, p => Source);
r02 : ENTITY mgncnfet_lib.Rss (arch_rss)
        PORT MAP ( m => Source_L1, p => Source_b);
r03 : ENTITY mgncnfet_lib.Rsd (arch_rsd)
        PORT MAP ( m => Drain_b, p => Drain_ch);
r04 : ENTITY mgncnfet_lib.Rsbd (arch_rsbd)
        PORT MAP ( m => Drain, p => Drain_b);
e01 : ENTITY mgncnfet_lib.vchel02 (arch_vchel02)
        PORT MAP ( m => Drain_ch, p => Drain_L1);
c09 : ENTITY mgncnfet_lib.cap_cgb (arch_cap_cgb)
        PORT MAP ( m => Sub, p => Gate);
c10 : ENTITY mgncnfet_lib.cap_cgd (arch_cap_cgd)
        PORT MAP ( m => Drain_L1, p => Gate);
c11 : ENTITY mgncnfet_lib. cap_cgs (arch_cap_cgs)
        PORT MAP ( m => Source_L1, p => Gate);
c12 : ENTITY mgncnfet_lib.cap_cbd (arch_cap_cbd)
        PORT MAP ( m => Drain_L1, p => Sub);
c13 : ENTITY mgncnfet_lib.cap_cbs (arch_cap_cbs)
        PORT MAP ( m => Source_L1, p => Sub);
i01 : ENTITY mgncnfet_lib. isemi (arch_isemi)
        PORT MAP ( m => Source_L1, p => Drain_L1);
i02 : ENTITY mgncnfet_lib.ibtbt (arch_ibtbt)
```

```
        PORT MAP ( m => Source_L1, p => Drain_L1);
i03 : ENTITY mgncnfet_lib.ibtbt (arch_ibtbt)
        PORT MAP ( m => Source_L102, p => Drain_L102);
i04 : ENTITY mgncnfet_lib.isemi(arch_isemi)
        PORT MAP ( m => Source_L102, p => Drain_L102);
c14 : ENTITY mgncnfet_lib.cap_cbs (arch_cap_cbs)
        PORT MAP ( m => Source_L102, p => Sub);
c15 : ENTITY mgncnfet_lib.cap_cbd (arch_cap_cbd)
        PORT MAP ( m => Drain_L102, p => Sub);
c16 : ENTITY mgncnfet_lib. cap_cgs (arch_cap_cgs)
        PORT MAP ( m => Source_L102, p => Gate);
c17 : ENTITY mgncnfet_lib.cap_cgd (arch_cap_cgd)
        PORT MAP ( m => Drain_L102, p => Gate);
c18 : ENTITY mgncnfet_lib. cap_cgb (arch_cap_cgb)
        PORT MAP ( m => Sub, p => Gate);
e02 : ENTITY mgncnfet_lib.vchel02 (arch_vchel02)
        PORT MAP ( m => Drain_ch02, p => Drain_L102);
r05 : ENTITY mgncnfet_lib.Rsbd (arch_rsbd)
        PORT MAP ( m => Drain, p => Drain_b02);
r06 : ENTITY mgncnfet_lib.Rsd (arch_rsd)
        PORT MAP ( m => Drain_b02, p => Drain_ch02);
r07 : ENTITY mgncnfet_lib.Rss (arch_rss)
        PORT MAP ( m => Source_L102, p => Source_b02);
r08 : ENTITY mgncnfet_lib.Rsbs (arch_rsbs)
        PORT MAP ( m => Source_b02, p => Source);
c19 : ENTITY mgncnfet_lib.cgdd (arch_cgdd)
        PORT MAP ( m => Drain_b02, p => Gate);
c20 : ENTITY mgncnfet_lib.cbdd (arch_cbdd)
        PORT MAP ( m => Sub, p => Drain_b02);
c21 : ENTITY mgncnfet_lib.cbdd (arch_cbdd)
        PORT MAP ( m => Sub, p => Drain_ch02);
c22 : ENTITY mgncnfet_lib.cgdd (arch_cgdd)
        PORT MAP ( m => Drain_ch02, p => Gate);
c23 : ENTITY mgncnfet_lib.cbss (arch_cbss)
        PORT MAP ( m => Sub, p => Source_L102);
```

```
    c24 : ENTITY mgncnfet_lib.cbss (arch_cbss)
            PORT MAP ( m => Sub, p => Source_b02);
    c25 : ENTITY mgncnfet_lib.cgss (arch_cgss)
            PORT MAP ( m => Source_L102, p => Gate);
    c26 : ENTITY mgncnfet_lib.cgss (arch_cgss)
            PORT MAP ( m => Source_b02, p => Gate);
    c27 : ENTITY mgncnfet_lib.cgsub (arch_cgsub)
            PORT MAP ( m => Sub, p => Gate);
    c28 : ENTITY mgncnfet_lib.cgtg (arch_cgtg)
            PORT MAP ( m => Source, p => Gate);
END ARCHITECTURE arch_my_ncnfet;
```

The VHDL-AMS code illustrated in Section 4.1.2 above works in combination with the code for the components shown in Appendix D and the main package indicated in Appendix E. This is actually for the n-type enhancement mode SWCNT CNFET (NCNFET), similar code has been developed for the p-type CNFET (PCNFET). The code in Section 4.1.2 is structural description of the model shown in Figure 4.1 while the codes in Appendix D link the behavioral description of the components of the CNFET embodied in the package demonstrated in Appendix E. This complete code has been successfully compiled and executed in Ansoft Simplorer version 9.0. Since Ansoft Simplorer does not support vector valued communication between the package and the main structural code (Section 4.1.2), we have been compelled to develop distinct function for every component. Therefore, one has to re-declare the quantities and terminals for the components that need computed data from the package.

4.2 SRAM Based on CNFET

An SRAM cell can be composed of four to ten transistors. In this paper, we have started from the six transistors based SRAM cell structure and then arrived at a cell composed of four transistors with load resistors; based on the transistors described in Section 4.1.

4.2.1 New SRAM Cell

To help us propose a new design of CNFET-based SRAM cell, we have started with the cell indicated on Figure 4.2 below by using optimum substrate biasing scheme for each

transistor, so as to achieve the best overall performance. In this design each transistor has been properly substrate-biased for each operation i.e. reading, writing and hold modes.

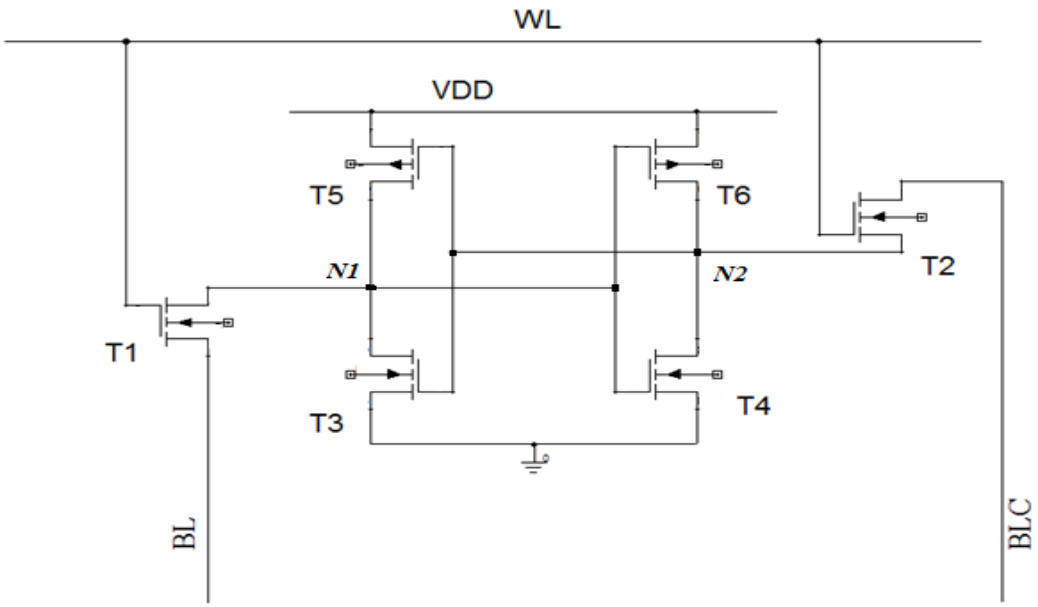

Figure 4.2 Six-Transistor SRAM Cell.

Let's have a brief description on the principles that determine the design. In this SRAM cell four transistors (T5, T6, T3 and T4) comprise cross-coupled inverters and two transistors T1 and T2 provide read and write access to the cell. In other words, T5 and T6 are the pull up or load PCNFETs, T3 and T4 are the pull down or driver NCNFETs, and T1 and T2 are the access NCNFETs. Upon the activation of the word line (WL), the access transistors connect the two internal nodes of the cell to the true bit line (BL) and the bit line bar or the complementary bit line (BLC). This SRAM cell can maintain the stored data as long as there is power supply, i.e. V_{DD}.

The two basic requirements which dictate the design of the SRAM cell are:

1. Reading operation shall not be destructive, i.e. the stored data shall also be available after reading is done.
2. The cell shall allow modification of the data during the writing operation.

Therefore, the SRAM cell must be designed such that it provides a non-destructive read operation and a reliable write operation. These two requirements impose contradicting requirements on the SRAM cell design.

Referring to Figure 4.2 and supposing that the stored data is logic '0', the transistors T4 and T5 are turned off while the transistors T3 and T6 are turned on or they can be made to operate in their linear region. As long as WL is not enabled and V_{DD} is supplied there will be no problem in maintaining the data by this cell. However, during the read operation T1 and T3 will conduct some current as a result voltage at node N1 (previously at 0 V) will increase. If the voltage at node N1 (V_{N1}) increases to greater than the threshold voltage of T4 ($V_{th,T4}$), the stored data will be changed. To avoid this destructive reading operation, the cell shall fulfill the condition that:

$$V_{N1} < V_{th,T4} \tag{4 - 1}$$

Similarly, supposing that the stored data is logic '1', the transistors T3 and T6 are turned off while the transistors T4 and T5 are turned on. This state can be maintained by the cell during the hold mode. But during the read operation T2 and T4 will conduct some current as a result voltage at node N2 (previously at 0 V) will increase. If the voltage at node N2 (V_{N2}) exceeds the threshold voltage of T3 ($V_{th,T3}$), the stored data will be changed. To avoid this destructive reading operation, the cell shall fulfill the condition that:

$$V_{N2} < V_{th,T3} \tag{4 - 2}$$

Coming to the writing operation, supposing that the stored data is logic '1', i.e. V_{N2} = 0V and $V_{N1} = V_{DD}$. During writing '0' the bit line BL is driven to 0 V. When the WL is enabled, the cell is designed to operate in accordance to Equation 4-2. But this design will not allow the data to be updated. Similarly, supposing that the stored data is logic '0', i.e. V_{N1} = 0 V and $V_{N2} = V_{DD}$. During writing '1', BLC is driven to 0 V. When the WL is enabled, the cell is designed to operate in accordance with Equation 4-1. But this design will not permit the data to be changed.

Based on the above explanation, the hold mode and reading mode has been well satisfied leaving a contradictory requirement for the writing mode. Hence, we have tried to tune the threshold voltages of the transistors for optimum functioning of the cell by varying the substrate voltage of each transistor. In order to bias the substrate of each transistor in the cell, we have first identified the threshold voltages of the transistors. Since the CNT of the CNFETs have threshold voltage (V_{th}) given by [15]:

$$V_{th} \approx \frac{\sqrt{3}aV_{\pi}}{3qD_{CNT}} \qquad (4-3)$$

Where,

- a is the carbon to carbon atom distance (~ 0.249 nm),
- V_{π} is the carbon π-π bond energy in the tight bonding model (~ 3.033 eV),
- q is electron charge, and
- D_{CNT} is the CNT diameter that can be calculated from Equation 2-2.

For an intrinsic CNT with chiral (n_1, 0) Equation 4-3 can be simplified as

$$V_{th} \approx \frac{5.5}{n_1} \qquad (4-4)$$

This implies that we can vary the chiral to control the threshold voltage of the CNFETs which in turn will affect the performance of the SRAM cell. Since the typical permitted diameters of SWCNT range from 0.6 to 2 nm, we can select n_1 as 8, 10, 11, 13, 14, 16, 17, 19, 20, 22 or 23. Therefore, we have threshold voltages of magnitude 0.687 V, 0.550 V, 0.500 V, 0.423 V, 0.393 V, 0.344 V, 0.323 V, 0.289 V, 0.275 V, 0.250 V or 0.239 V for the CNFETs.

The trials show that each of the transistors has to be biased in various ways to satisfy reading logic '0', reading logic '1', writing logic '0', writing logic '1' and hold mode. For example,

- During hold mode, the substrate of the transistors can be grounded.
- In reading logic '0', the substrate of the transistors T1, T2 and T3 shall be biased at about -0.4 V; T5 shall be biased at about 0.4 V while T4 and T6 shall be grounded.
- In reading logic '1', the substrate of the transistors T1, T2 and T4 shall be biased at about -0.4 V; T6 shall be biased at about 0.4 V while T3 and T5 shall be grounded.
- In writing logic '0' to a cell storing logic '1', the substrate of the transistors T1, T2 and T3 shall be biased at about -0.4 V; T5 and T6 shall be biased at about 0.4 V while T4 shall be grounded.

This implies complex circuitry is required to bias the substrate of each of the transistor in every operation. This complicates the structure of the SRAM which will be also infeasible to fabricate the cell. In addition, the traditional six transistors based SRAM cell occupies

more area due to the presence of the load transistors. So we have come up with a new SRAM cell composed of four transistors (two of which are substrate-biased) and resistor-loads as shown in Figure 4.3.

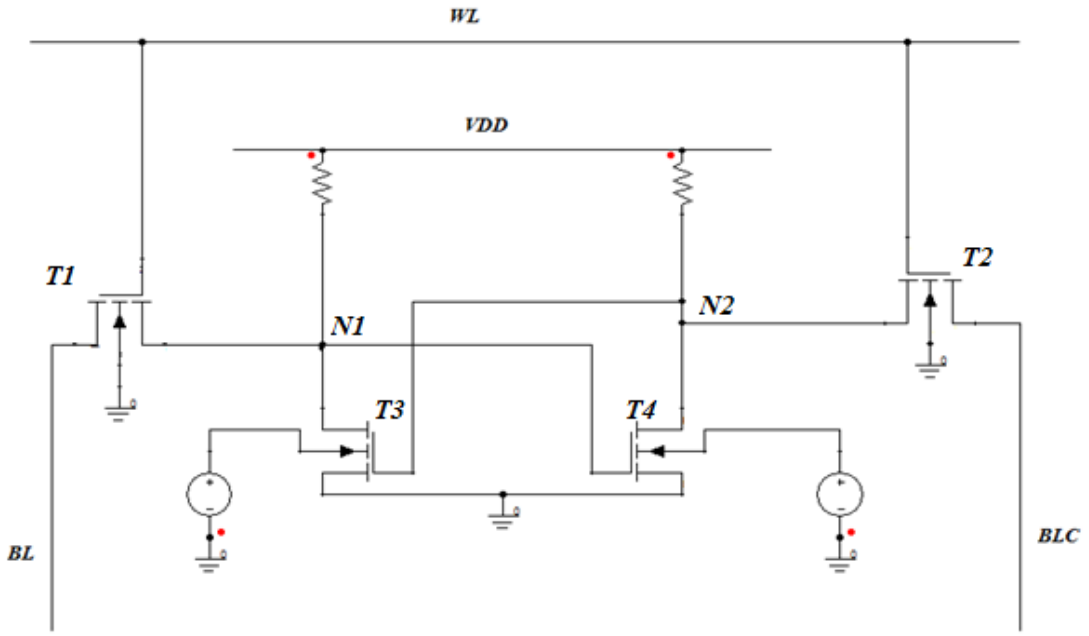

Figure 4.3 Proposed SRAM Cell.

The proposed SRAM cell indicated on Figure 4.3 is composed of two substrate-biased pull down or driver NCNFETs (transistors T3 and T4), two load resistors and two access NCNFETs (transistors T1 and T2). Upon the activation of the word line (WL), the access transistors connect the two internal nodes of the cell to the true bit line (BL) and the complementary bit line (BLC). This SRAM cell maintains the stored data as long as there is power supply, i.e. V_{DD}. The reading and writing operations of this cell can be performed by controlling the threshold voltages of the driver transistors.

For example, referring to Figure 4.3 and supposing that the stored data is logic '1', T3 is turned off while T4 is turned on. As long as WL is not enabled and V_{DD} is supplied the data will be maintained by this cell even without biasing the substrates. However, during the read operation T2 and T4 will conduct some current as a result voltage at node N2 (previously at 0 V) will increase. In order to maintain the voltage at N2 below the threshold voltage of T3, the substrate voltage of T3 can be increased. Similarly, in order to write logic

‘0’ to this cell which is currently storing logic ‘1’; the bit line BL shall be driven to 0 V, the WL shall be enabled, and the voltage at N1 can be maintained below the threshold voltage of T4 by increasing the substrate voltage of T4.

4.2.2 VHDL-AMS Code of the Cell

```
LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
USE IEEE.MATH_REAL.ALL;
LIBRARY mg_components;
USE mg_components.all;
LIBRARY mgncnfet_lib;
USE mgncnfet_lib.all;
ENTITY sram_cell IS
   PORT(TERMINAL BL, BLC, WL, VDD: ELECTRICAL);
END ENTITY sram_cell;
ARCHITECTURE struct OF sram_cell IS
        TERMINAL net_1 : ELECTRICAL;
        TERMINAL net_2 : ELECTRICAL;
        TERMINAL net_3 : ELECTRICAL;
        TERMINAL net_4 : ELECTRICAL;
BEGIN
        E01 : ENTITY mg_components.v_source01(archv_source01)
                GENERIC MAP (v_val => -0.4)
                PORT MAP ( m => net_3, p => electrical_ref);
        E02 : ENTITY mg_components. v_source01 (archv_source01)
                GENERIC MAP (v_val => -0.4)
                PORT MAP ( m => net_4, p => electrical_ref);
        U01 : ENTITY mgncnfet_lib.my_ncnfet (arch_my_ncnfet)
                PORT MAP (sub => net_4, source => electrical_ref, gate => net_1, drain => net_2);
        U02 : ENTITY mgncnfet_lib.my_ncnfet (arch_my_ncnfet)
                PORT MAP (sub => electrical_ref, source => net_2, gate => WL, drain => BL);
        U03 : ENTITY mgncnfet_lib.my_ncnfet (arch_my_ncnfet)
                PORT MAP (sub => net_3, source => electrical_ref, gate => net_2, drain => net_1);
        U04 : ENTITY mgncnfet_lib.my_ncnfet (arch_my_ncnfet)
                PORT MAP (sub => electrical_ref, source => net_1, gate => WL, drain => BLC);
```

```
        r1 : ENTITY mg_components. res_01(arch_res01)
                GENERIC MAP ( r => 60000.0)
                PORT MAP ( m => net_2, p => VDD);
        r2 : ENTITY mg_components.res_01(arch_res01)
                GENERIC MAP ( r => 60000.0)
                PORT MAP ( m => net_1, p => VDD);
END ARCHITECTURE struct;
-- Voltage Source
LIBRARY IEEE;
USE IEEE.MATH_REAL.ALL;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
 ENTITY v_source01 IS
     GENERIC(v_val: VOLTAGE);
     PORT(TERMINAL m, p: ELECTRICAL);
END ENTITY v_source01;
 ARCHITECTURE archv_source01 OF v_source01 IS
     QUANTITY v ACROSS i THROUGH m TO p;
BEGIN
     v == v_val;
END ARCHITECTURE archv_source01;
-- Resistor
LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
ENTITY res01 IS
        GENERIC(r : RESISTANCE);
        PORT ( TERMINAL p, m  : ELECTRICAL);
END ENTITY res01;
ARCHITECTURE arch_res01 OF res01 IS
   QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
   v == i*r;
END ARCHITECTURE arch_res01;
```

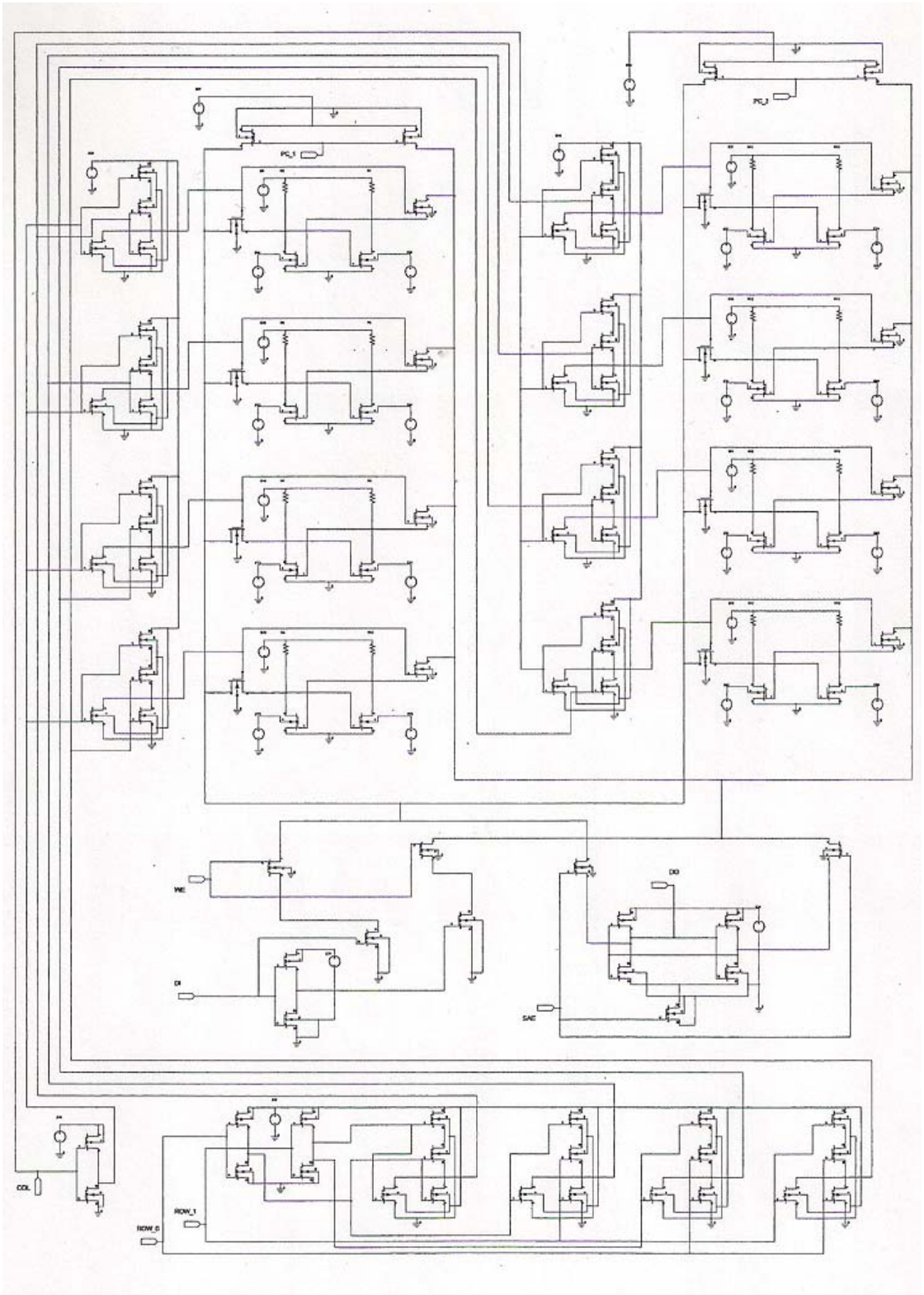

Figure 4.4 8-Bit SRAM.

4.2.3 8-Bit SRAM

An eight bit or one byte SRAM is made using the proposed cell as indicated in Figure 4.4 for the sake of completeness. In this configuration, the main signal ports that control the operations of the SRAM are shown. Row and column address input signals are used to select a memory location on the chip. The address input COL is for column selection, which can have values 0 and 1 to select the first column and second column respectively. The four rows are selected by combination of address inputs ROW_0 and ROW_1; at combinations of 00, 01, 10 and 11 for the first, second, third and fourth rows respectively. Here, the word-line of a cell is made to be activated by a combination of signals from the row and column enable control signals. During a write operation, a data signal is applied at the data input pin (DI). This data is stored in the selected memory cell when the write enable (WE) control signal is activated. During a read operation, data from the selected memory cell appears at the data output pin (DO) when the signal amplifier enable control signal (SAE) is enabled. During the read and write operations a peripheral circuit charges both the bit lines to half of the V_{DD} value when the pre-charging control signal (PC_1 or PC_2) is enabled.

CHAPTER 5

RESULTS

5.1 I-V Characteristics of the CNFET

As discussed in Chapter 2, varying the gate voltage of the CNFET changes its threshold voltage. In addition, varying the substrate voltage can also be used to control the threshold voltage of the transistor. In order to identify and understand the impact of substrate biasing the CNFETs, simulations of both PCNFET and NCNFET have been done at various gate and substrate biasing voltages. The results are indicated in Figures 5.1, 5.2, 5.3 and 5.4. From these simulation results we can infer that if the substrate voltage is decreased the threshold voltage of NCNFET decreases so the current capability becomes higher. In the other case, if the substrate voltage is increased the threshold voltage of PCNFET decreases. Besides, from Figures 5.2 and 5.3, it can be understood that the substrate can be used as an alternative gate. These simulations are done for a CNFET with CNT of chirality (14, 0) channel length of 32 nm and at a temperature of 300K.

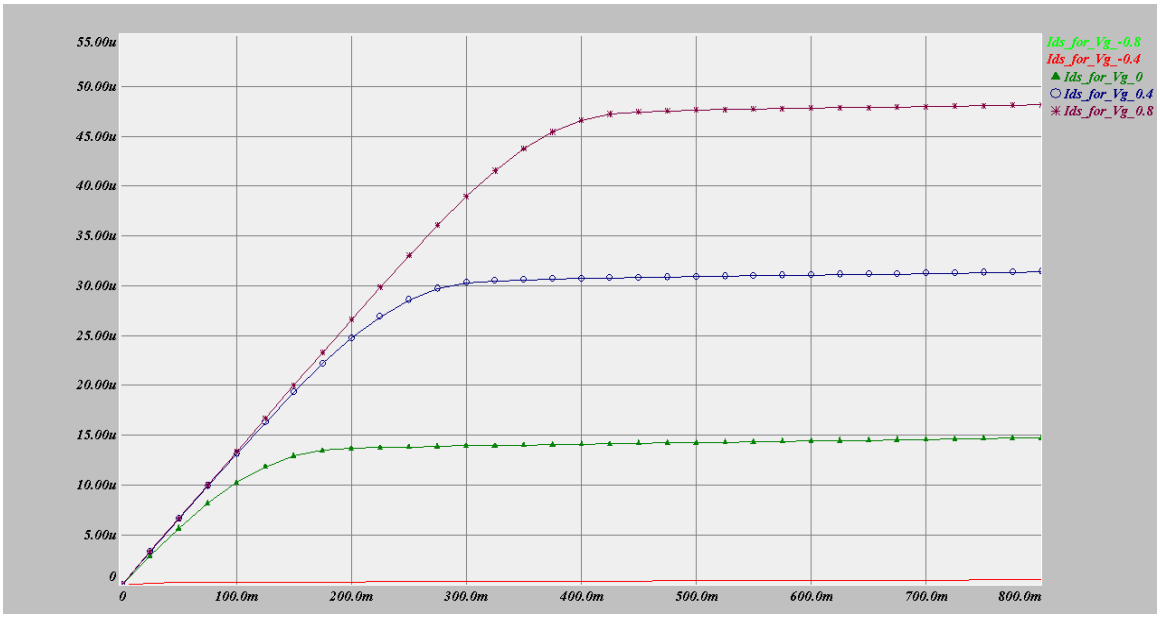

Figure 5.1 NCNFET at Substrate Voltage of -0.6 V and Varying Gate Voltage.

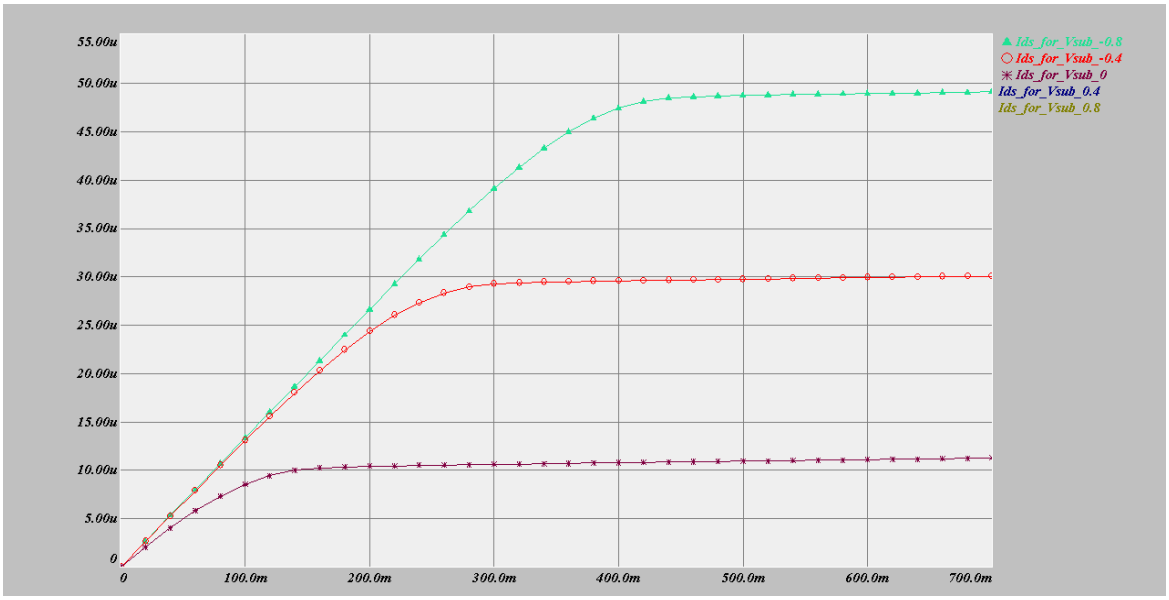

Figure 5.2 NCNFET at Gate Voltage of 0.6 V and Varying Substrate Voltage.

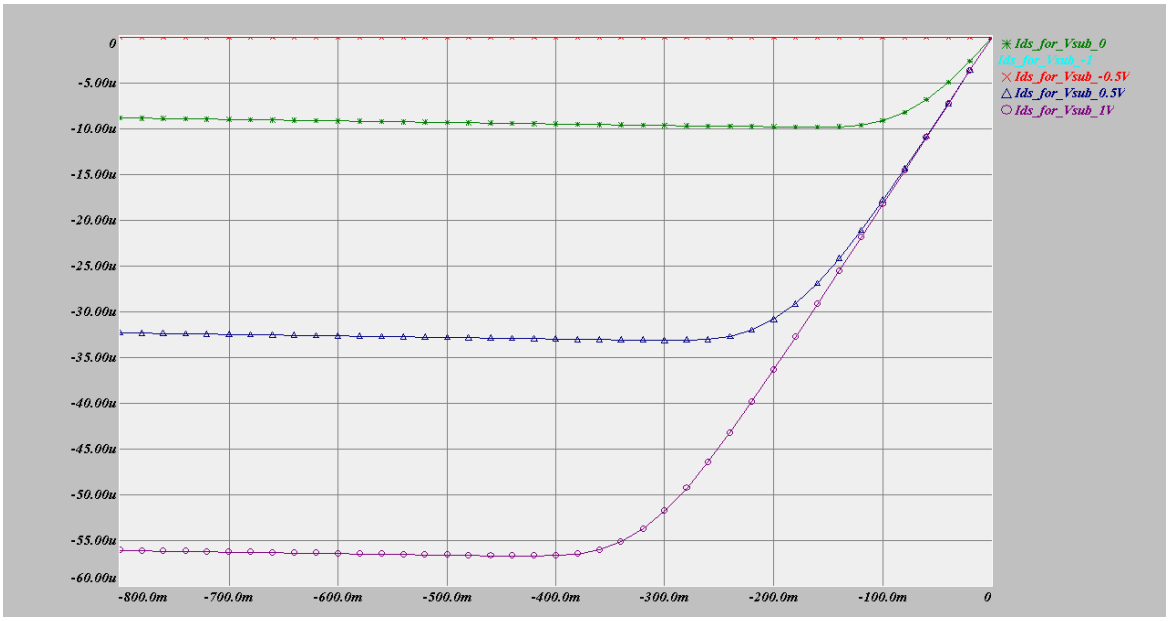

Figure 5.3 PCNFET at Gate Voltage of -0.6 V and Varying Substrate Voltage.

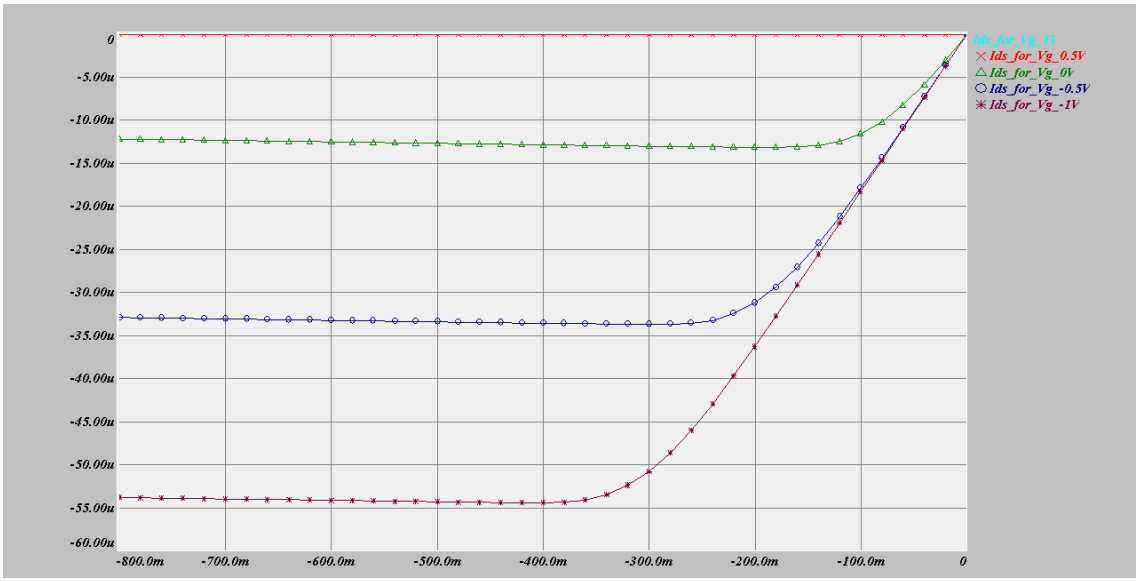

Figure 5.4 PCNFET at Substrate Voltage of 0.6 V and Varying Gate Voltage.

5.2 Read and Write Speeds

The proposed SRAM cell has been simulated for its speed in its reading and writing operations. The time required by the SRAM to change the states of the data storing nodes during its writing operation is named as write time while the time required by the SRAM to amplify the difference between the potentials of the data storing nodes during its reading operation is named as read time.

5.2.1 Simulation Results

The simulation results shown on Figures 5.5 and 5.6 have been obtained for the following values:

V_{DD} (voltage source of the SRAM cell) = 0.8 V

PC (pre-charge circuit source) = 0.4 V

R_L (load resistances) = 60 kΩ

WL (word line) = 0.6 V at 125 MHz

WE (write enable) = 0.6 V at 200 MHz

RE (read enable) = 0.6 V at 100 MHz

Substrate biasing of the driver transistors = 0.3 V.

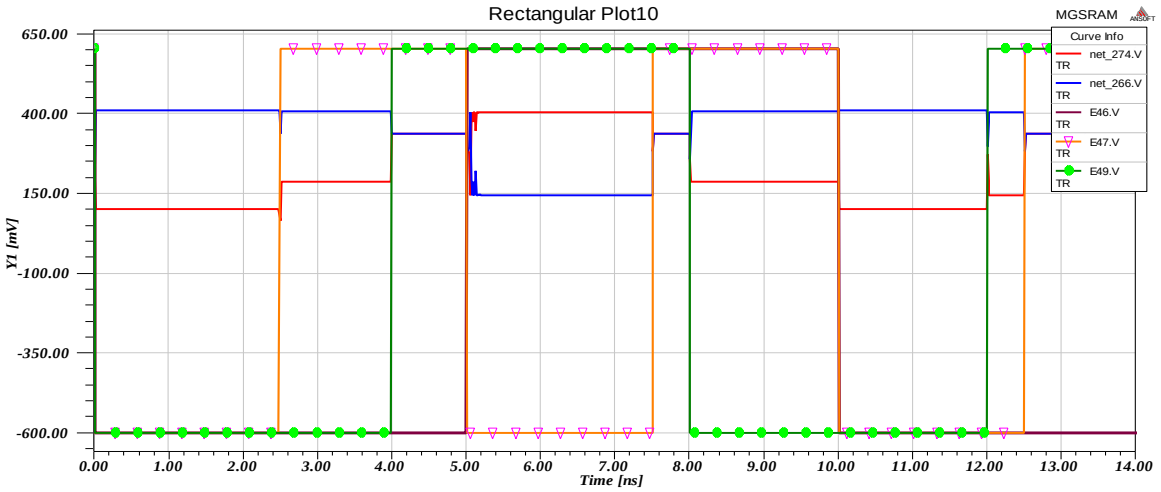

Figure 5.5 (a) The Proposed SRAM Writing '0' and '1'.

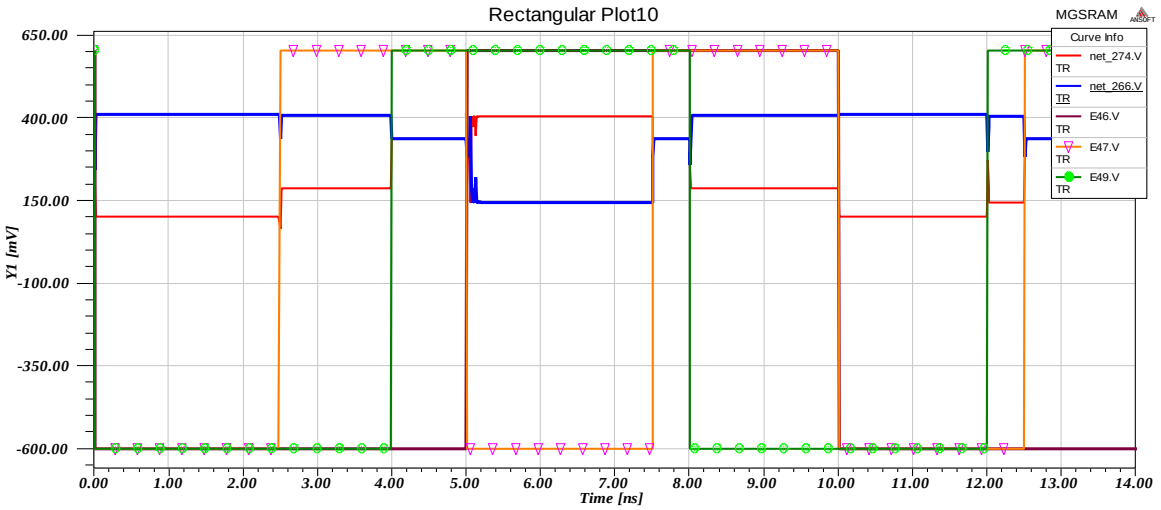

Figure 5.5 (b) The Proposed SRAM Writing '0' and '1' the BL.

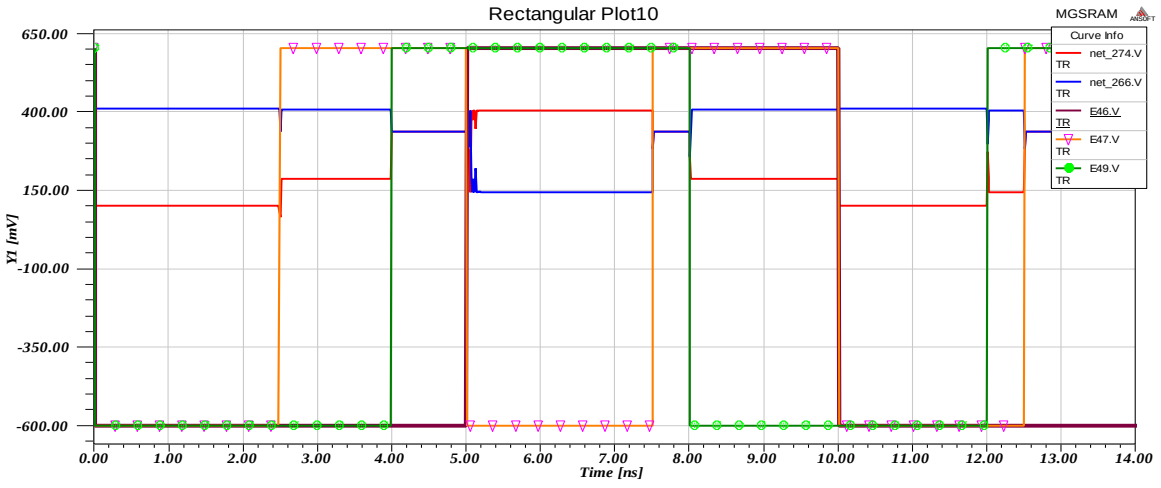

Figure 5.5 (c) The Proposed SRAM Writing '0' and '1'the DI.

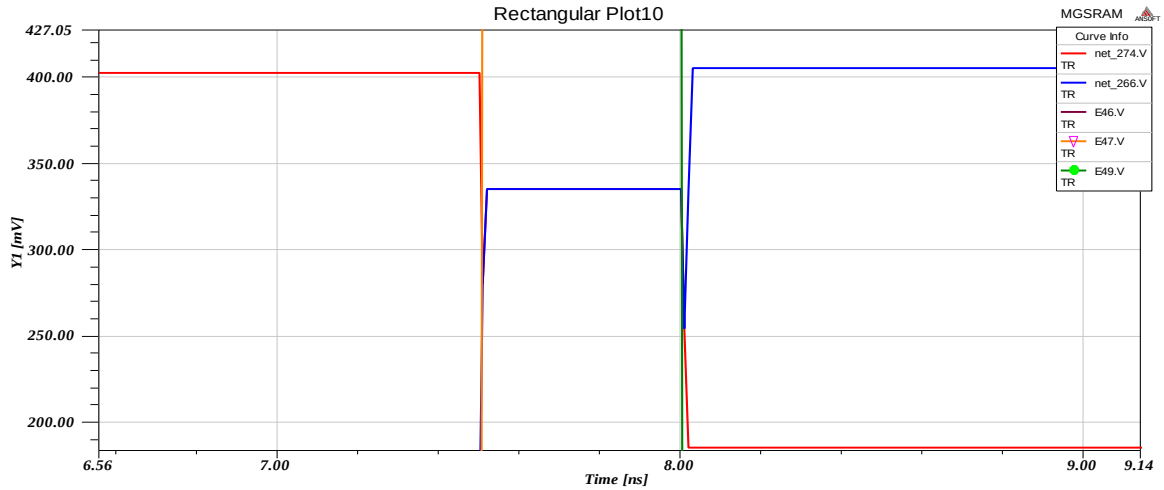

Figure 5.5 (d) The Proposed SRAM Writing '1'at about 8 ns.

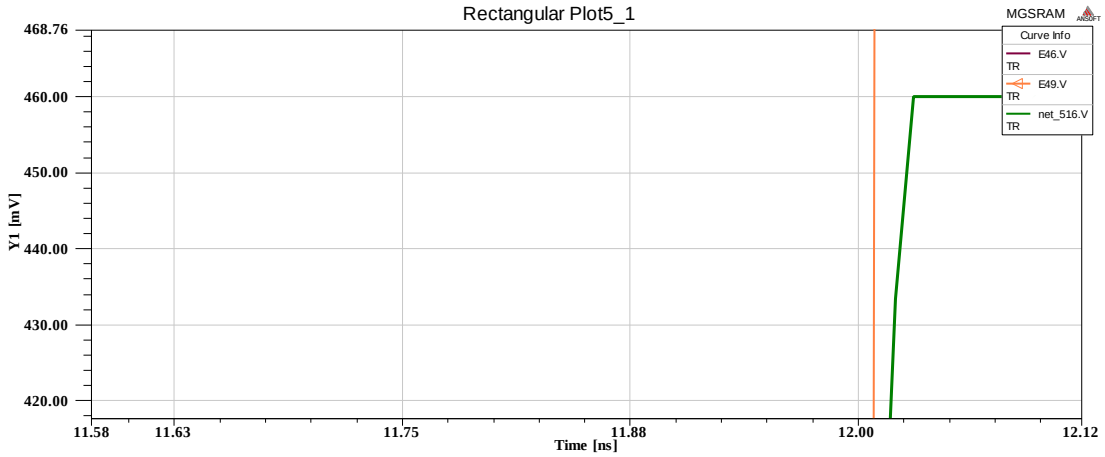

Figure 5.6 (a) The Proposed SRAM Reading '1' at about 12 ns.

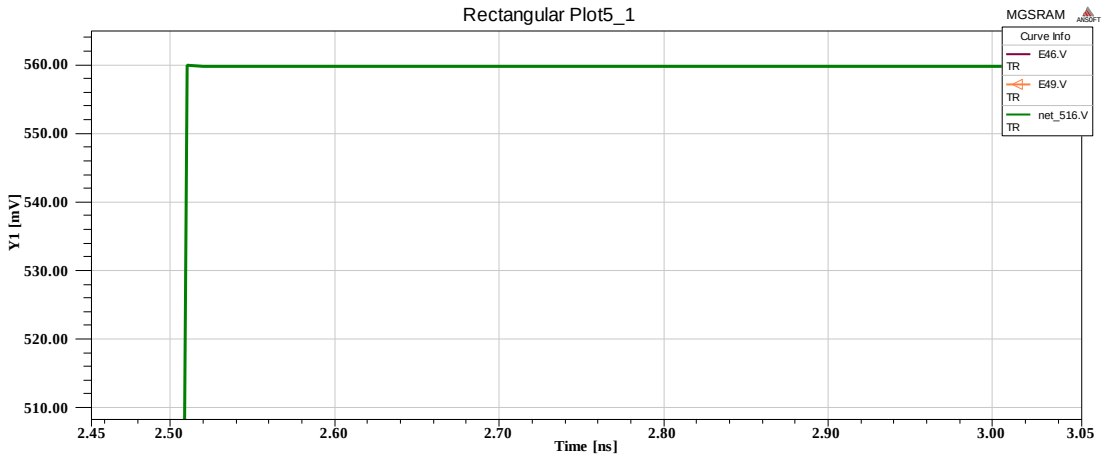

Figure 5.6 (b) The Proposed SRAM Reading '1' at about 2.5 ns.

In the results of the simulations the representations are as follows:

E47.V for WE (write enable);

E49.V for WL (word line);

Net_274.V for BLC (complementary bit line);

Net_266.V for BL (bit line);

E46.V for DI (data in);

Net_516.V for DO (data out).

The simulation results on Figure 5.5 show the proposed SRAM cell writing logic '0' and logic '1'. Here, it can be observed that the SRAM cell writes when both WE and WL are enabled and changes the data based on the input data i.e. DI. On Figure 5.5(a) the SRAM writes logic '0' at about 5 ns since both WL and WE signals are enabled and DI at this time is logic '0'. Similarly, the BL has been changed to a state of logic '1' at about 8 ns because both WL and WE signals are enabled (in the interval 7.50 – 8.00 ns) and DI in this time is logic '1'. This information (logic '1') is maintained by the SRAM cell till 12.5 ns because in the interval 8.00 – 10.00 ns only WE is enabled but WL is disabled, in the interval 10.00 – 12.00 ns both WE and WL are not enabled, and in the interval 12.00 – 12.50 ns only WL is enabled. Figure 5.5(b) and Figure 5.5(c) show simulation results emphasizing on BL and DI respectively. As shown in Figure 5.5(c), DI has a period of 10 ns and has a value of logic '0' from 0.00 – 5.00 ns and a value of logic '1' from 5.00 – 10.00 ns while BL is at logic '1' from 0.00 – 4.00 ns because WL and WE are not enabled at the same time in this interval. On Figure 5.5(d) writing logic '1' operation is zoomed to show the delay of the cell at about 8 ns. Figure 5.6(a) and (b) show reading logic '1' operation zoomed to show the delay of the cell at about 12 ns and 2.5 ns respectively.

5.2.2 Comparison with Other SRAMs

Results from [27] show that the write time of the six-CNFETs based SRAM in 20 nm technology and six-transistors based SRAM in 32 nm CMOS technology can be as low as 29.75 ps and 25.62 ps respectively. In addition, the read time of the six-CNFETs based SRAM in 20 nm technology and six-transistors based SRAM in 32 nm CMOS technology can be as low as 31.59 ps and 25.79 ps respectively. In our results, the proposed SRAM can have read time and write time as low as 25.0 ps and 20.0 ps respectively in 32 nm

technology. These show that the proposed SRAM is faster to read and write than the six-MOSFETs based SRAM and the six-CNFETs based SRAM.

5.3 Stability of the SRAM

International Technology Roadmap for Semiconductors (ITRS) [26] predicts that the supply voltage should be reduced to 0.9 V for 32 nm technology node. Hence, we have made the operational voltages below 0.9 V, specifically at V_{DD} equal to 0.8 V. The simulation setups explained in Section 3.3.2 have been used to evaluate the stability of the proposed SRAM cell. Figure 5.7b has been used as set up to simulate the faulty resistance model while Figure 5.7a has been used as set up to simulate the adverse noise that can be inoculated between the data storing nodes.

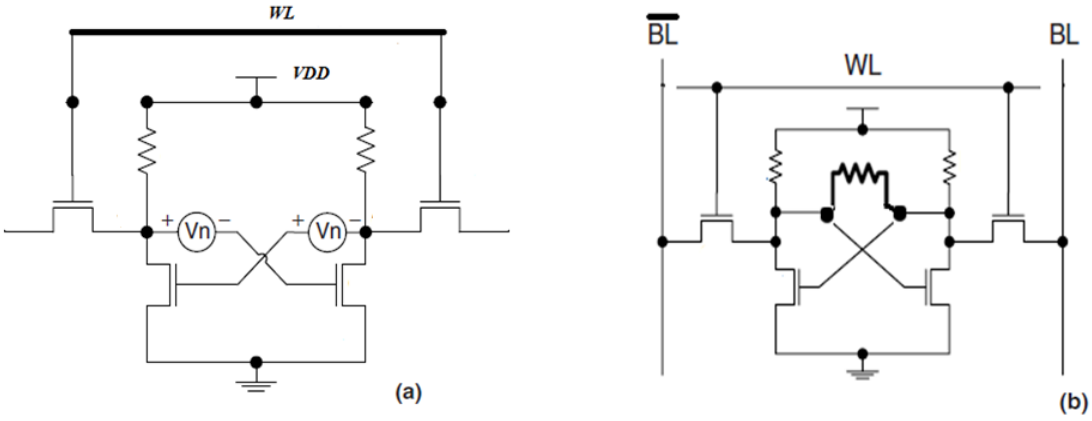

Figure 5.7 Test Setups for the Proposed SRAM.

5.3.1 Simulation Results

The simulation results indicated in Figures 5.7, 5.8 and 5.9 have been obtained for the following parameters:

V_{DD} = 0.8 V;

PC = 0.4 V;

R_L = 60 kΩ;

WL = 0.6 V at 125 MHz;

WE = 0.6 V at 200MHz;

RE = 0.6 V at 100MHz;

Substrate biasing of the driver transistors = 0.3 V.

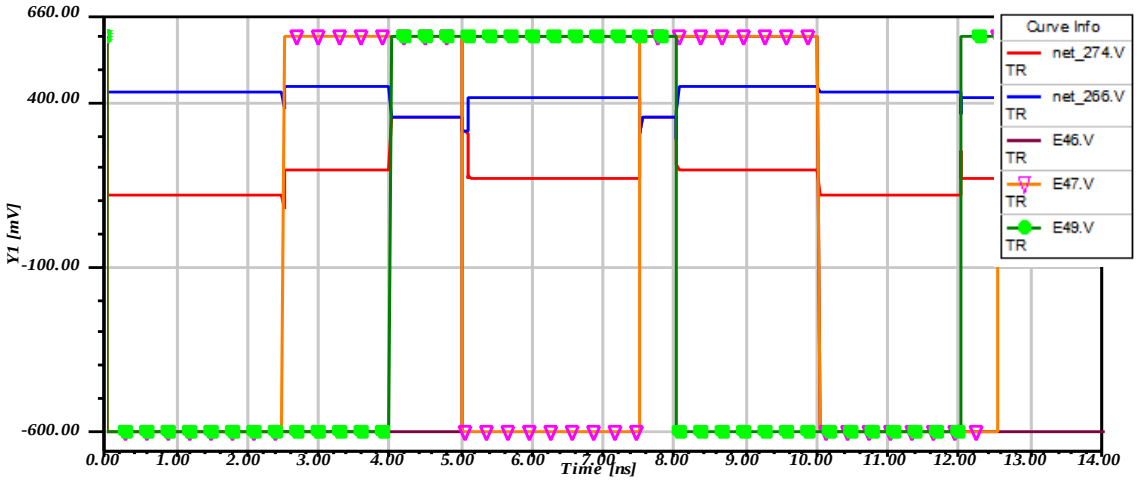

Figure 5.8 The Proposed SRAM at 1-10 MΩ Fault Resistance.

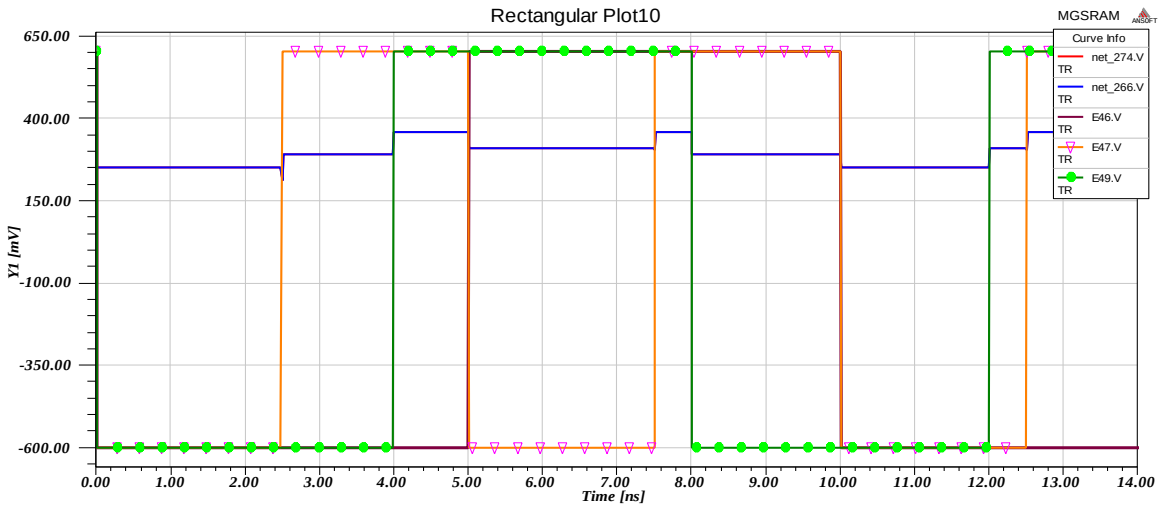

Figure 5.9 The Proposed SRAM at 100 kΩ Fault Resistance.

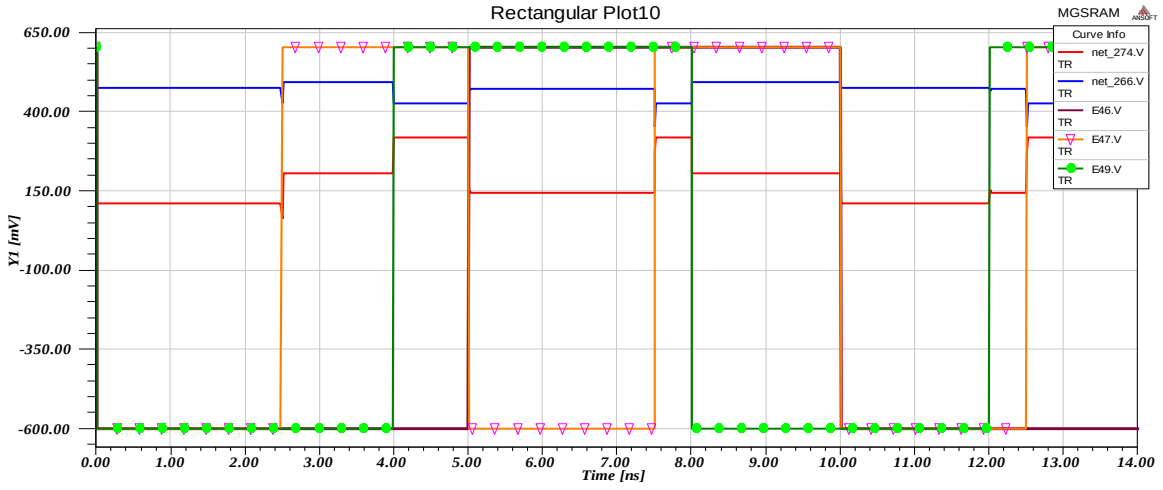

Figure 5.10 The Proposed SRAM at 130 mV Noise Voltage Source.

In the results of the simulations the representations are as follows: E46.V for DI (data in); E47.V for WE (write enable); E49.V for WL (word line); Net_274.V for BLC (complementary bit line); and Net_266.V for BL (bit line).

When the proposed SRAM is simulated for stability by fault resistance of 20 MΩ between the nodes that store the data no observable malfunction has appeared. When the proposed SRAM is simulated for fault resistances from 1-10 MΩ nothing can be written to or read from the SRAM but it has maintained a potential difference between the nodes that store data during the hold mode as shown in Figure 5.8. In Figure 5.8, since DI is logic '0' when both WE and WL are enabled in the interval 4.00 – 5.00 ns, BL is expected to be logic '0' in the interval 5.00 – 7.50 ns but has a wrong value of logic '1' due to the fault resistance. As shown in Figure 5.9, the SRAM is not able to do any of its operations at a fault resistance of 100 kΩ, i.e. it cannot be written to or read from neither can it maintain any potential between the nodes. Regarding noise toleration, the proposed SRAM can operate very well in the presence of noise with magnitude of up to 120 mV. However, at about 130 mV of noise it can't operate as memory cell as indicated in Figure 5.10. In short, the proposed SRAM can work properly in the presence of fault resistances of more than 20 MΩ and can tolerate a noise of magnitude up to 120 mV.

5.3.2 Comparison with Other SRAMs

Results from [24] show that the SNM of the six-transistors based SRAM and the loadless four-transistors based SRAM in 65 nm CMOS technology can be up to 260 mV and 230 mV respectively. Results from [27] show that the SNM of the six-CNFETs based SRAM in 20 nm technology and six-transistors based SRAM in 32 nm CMOS technology can be up to 227 mV and 85 mV respectively. In our results, the proposed SRAM can have SNM of only 120 mV and can tolerate a fault resistance down to about 20 MΩ in 32 nm technology. These show that the proposed SRAM is more susceptible to static noise than six-CNFETs based SRAM in 20 nm technology, the six-MOSFETs based SRAM and the loadless four-MOSFETs based SRAM at 65 nm CMOS technology. But it is much better than the six-transistors based SRAM in 32 nm CMOS technology. Besides, we expect better SNM and fault resistance when compared with similar architecture SRAM based on CNFETs, but we haven't got a paper that deals with similar CNFET structure.

CHAPTER 6
CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In this thesis a compact model of a type of CNFET known as enhancement-mode MOSFET-like SWCNT-CNFET was studied. The model includes the intrinsic channel model, the doped source/drain extension regions model and multiple nanotubes transistor model. The thermionic and band-to-band tunneling currents from the semiconducting sub-bands are considered as the main current contributors for the transistor in this model. It takes into consideration elastic scattering, acoustic and optical phonon scattering, parasitic capacitances and resistances, SB-resistances and screening effects among multiple nanotubes. The model was described in hardware description language known as VHDL-AMS. The VHDL-AMS code encompasses structural description of the equivalent circuit and behavioral code of the mathematical expressions in a separate package. The complete VHDL-AMS code of the transistor was used to study the electrical characteristics of the CNFET. The influence of substrate biasing on the CNFET was simulated and the results confirm that the substrate (bulk) can be used as an alternative gate or additional terminal to control the transistor.

Based on the CNFET, a new SRAM cell was developed. The proposed SRAM cell is composed of four CNFETs (two of which are substrate-biased) and two load resistors. The SRAM was implemented in VHDL-AMS. The VHDL-AMS code was simulated to study the performance of the SRAM. The simulation results at 32 nm technology show that the proposed SRAM cell is faster than the present SRAMs but less noise tolerant than most of the SRAMs. When compared with the most common six-transistors CMOS technology based SRAM cell, the proposed SRAM cell exhibits a number of advantages including reduced area occupation, faster reading and writing speeds, and more stability.

In general, we can conclude that CNFETs can be used to produce high density and fast digital devices such as the SRAM beyond 32 nm channel length. We can also predict that CNFETs outperform MOSFETs under the same conditions such as at the same operating environment, equal channel lengths and overall dimensions.

6.2 Future Works

Other peripheral circuitry and signal pins used to implement special features such as spare cells management and low power hold mode can also be designed and added to the SRAM.

Further studies can be done to develop optimum layout of the SRAM for fabrication and testing.

More accurate simulation results can be obtained by using more accurate models such as the NEGF and Monte-Carlo simulations by the help of high speed computers.

REFERENCES

1. B. G. Streetman and S. K. Banerjee, Solid State Electronic Devices, Sixth Edition, Pearson Prentice Hall, USA, 2006.
2. Z. Zhang, J. G. Delgado-Frias and J. Nyathi, "CNTFET SRAM Cell Design with Tolerance to Metallic CNTs," IEEE, 2010.
3. T. Dang, L. Anghel and R. Leveugle, "CNTFET Basics and Simulation," IEEE, 2006.
4. K. W. Whitten, R. E. Davis and M. L. Peck, General Chemistry with Qualitative Analysis, Sixth edition, Saunders Co. Ltd., 1999.
5. Microsoft Corporation, Microsoft Encarta Premium, USA, 2009.
6. P. F. Harris, Carbon Nanotubes and Related Structures: New Materials for the Twenty-first Century, Cambridge University Press, USA, 1999.
7. C. Srinivasan and R. Saraswathi, "From Graphite to Graphene and Now to Graphane," Current Science, Vol. 97, No. 3, 2009.
8. A. Javey and J. Kong, Carbon Nanotube Electronics, Springer, USA, 2009.
9. M. Meyyappan, Carbon Nanotube Science and Applications, CRC Press, USA, 2005.
10. M. Endo, T. Hayashi, Y. Kim, M. Terrones and M. Dresselhaus, "Applications of Carbon Nanotubes in the Twenty-First Century," The Royal Society, 2004.
11. H. Rafii-Tabar, Computational Physics of Carbon Nanotubes, Cambridge University Press, New York, 2008.
12. I. O'Connor, J. Liu and F. Gaffiot, "CNTFET-Based Logic Circuit Design," IEEE, 2006.
13. F. Pregaldiny, C. Lallement and J. Kammerer, "Design-oriented Compact Models for CNTFETs," IEEE, 2006.
14. J. Deng, "Device Modeling and Circuit Performance Evaluation for Nanoscale Devices: Silicon Technology Beyond 45 nm Node and Carbon Nanotube Field Effect Transistors," Stanford University, 2007.
15. H. Shahidipour, Y. Zhong, A. Ahmadi and K. Maharatna, "Effects of CNT Diameter Variability on a CNFET-Based SRAM," Asia Pacific Conference on Circuits and Systems, Malaysia, 2010.

16. J. Deng and H.S. Philip Wong, "A Compact SPICE Model for Carbon-Nanotube Field-Effect Transistors Including Nonidealities and Its Application—Part I: Model of the Intrinsic Channel Region," IEEE Transactions on Electron Devices, Vol. 54, No. 12, 2007.

17. J. Deng and H.S. Philip Wong, "A Compact SPICE Model for Carbon-Nanotube Field-Effect Transistors Including Nonidealities and Its Application—Part II: Full Device Model and Circuit Performance Benchmarking," IEEE Transactions on Electron Devices, Vol. 54, No. 12, 2007.

18. J. Deng and H.S. Philip Wong, "Modeling and Analysis of Planar-Gate Electrostatic Capacitance of 1-D FET with Multiple Cylindrical Conducting Channels," IEEE Transactions on Electron Devices, Vol. 54, No. 9, 2007.

19. S. Kang and Y. Leblebici, CMOS Digital Integrated Circuits: Analysis and Design, Third Edition, McGraw-Hill, USA, 2003.

20. S. Ikeda, "Technology for High-density and High-performance Static Random Access Memory," Tokyo Institute of Technology, Tokyo, 2003.

21. A. Pavlov and M. Sachdev, CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies, Springer, USA, 2008.

22. K. Itoh, M. Horiguchi and H. Tanaka, Ultra-Low Voltage Nano-Scale Memories, Springer, Massachusetts, USA, 2007.

23. B. Ebrahimi and A. Afzali-Kusha, "Realistic CNFET Based SRAM Cell Design for Better Write Stability," IEEE, International Symposium on Quality Electronic Design, 2009.

24. Sandeep R, N. T. Deshpande and A. R. Aswatha, "Design and Analysis of a New Loadless 4T SRAM Cell in Deep Submicron CMOS Technologies," IEEE, Second International Conference on Emerging Trends in Engineering and Technology, 2009.

25. International Roadmap Committee, "International Technology Roadmap for Semiconductors: 2008 Update Overview," ITRS, 2008.

26. M. Moradinasab, F. Karbassian and M. Fathipour, "A Comparison Study of the effects of Supply Voltage and Temperature on the Stability and Performance of CNFET and Nanoscale Si-MOSFET SRAMs," University of Tehran, Iran, 2009.

27. S. Lin, Y. Kim and F. Lombardi, "A New SRAM Cell Design Using CNTFETs," IEEE, International SoC Design Conference, 2008.

28. E. Grossar, M. Stucchi, K. Maex and W. Dehaene, "Read Stability and Write-Ability Analysis of SRAM Cells for Nanometer Technologies", IEEE Journal of Solid State Circuits, Vol. 41, No. 11, Nov. 2006.

29. B. S. Amrutur, "Design and Analysis of Fast Low Power SRAMs", Stanford University, 1999.

30. P. Avouris, J. Appenzeller, R. Martel and S. J. Wind, "Carbon Nanotube Electronics," Proceedings of the IEEE, Vol. 91, No. 11, Nov. 2003.

31. Z. Navadi, VHDL - Analysis and Modeling of Digital Systems, Second Edition, McGraw-Hill, USA, 1998.

32. P. J. Ashenden, The Designer's Guide to VHDL-AMS, Morgan Kaufmann Publishers, San Francisco, 1995.

33. ANSYS Inc., "Simplorer VHDL-AMS Tutorial," November 2010.

APPENDIX A. LIST OF SYMBOLS

C_{bd}	capacitance between the bulk and the drain at level one of the CNFET model
C_{bdd}	capacitance between the bulk and the drain at level two of the CNFET model
C_{bs}	capacitance between the bulk and the source at level one of the CNFET model
C_{bss}	capacitance between the bulk and the source at level two of the CNFET model
C_{gb}	capacitance between the gate and the bulk at level one of the CNFET model
C_{gd}	capacitance between the gate and the drain at level one of the CNFET model
C_{gdd}	capacitance between the gate and the drain at level two of the CNFET model
C_{gs}	capacitance between the gate and the source at level one of the CNFET model
C_{gss}	capacitance between the gate and the source at level two of the CNFET model
C_{gsub}	coupling capacitance between the gate and the substrate
C_{gtg}	gate-to-gate (or gate-to-S/D) coupling capacitance per unit length
C_{of}	gate outer-fringe capacitance
C_{ox}	capacitance between the gate and channel
C_Q	quantum capacitance of the doped S/D region
C_{sub}	capacitance between channel and substrate
D_{out}	drain connectivity (0 if drain is connected to doped CNT; otherwise, equal to 1)
D_{CNT}	diameter of the CNT
E_f	doping level
$E_{m,l}$	carrier energy at the substate (m, l)
$G_{x,c}$	quantum conductance of doped CNT
h	the Planck's constant (6.62617×10^{-34}Js or 4.1357×10^{-15}eVs),
I_{btbt}	band-to-band tunneling current from the semiconducting subbands
I_{semi}	thermionic current from the semiconducting subbands
k	the Boltzmann's constant (1.38066×10^{-23}J/K or 8.6174×10^{-5}eV/K)
l_{ap}	effective acoustic phonon scattering MFP
L_d	length of doped drain region
L_g	length of gate region
l_{op}	effective optical phonon scattering MFP
L_s	length of doped source region
N_D	doping density
q	the electronic charge of value 1.60218×10^{-19}C
Q_{cap}	charge induced by the electrodes
Q_{CNT}	total charge induced on SWCNT surface
R_{sb}	Schottky-barrier resistance
$R_{semi,d}$	resistance of doped drain region
$R_{semi,s}$	resistance of doped source region
S_{out}	source connectivity (0 if source is connected to doped CNT; otherwise, it is 1)
T	temperature in Kelvin

V_{bi}	build in potential with an applied bias
V_{BS}	potential difference between substrate and source
V_{xs}	potential difference between node x and source,
$\Delta\Phi_B$	the channel surface potential change with gate/drain bias,
$\Delta\Phi_s$	the surface potential change
$\Delta\Phi_{s,max}$	the maximum surface potential change
ε_0	the permittivity of free space of value 8.85418×10^{-14}F/cm,
$\Theta(x)$	a step function that is equal to 1 if x > 0; otherwise, it is equal to 0.
λ_{ap}	acoustic phonon scattering MFP
λ_{op}	optical phonon scattering mean free path
Φ_C	CNT work function
Φ_M	metal work function

APPENDIX B. ABBREVIATIONS

ASIC	Application Specific Integrated Circuit
BTBT	Band To Band Tunneling
CMOS	Complementary Metal Oxide Semiconductor
CNFET	Carbon Nanotube Field Effect Transistor
CNT	Carbon Nano Tube
CPU	Central Processing Unit
DRAM	Dynamic Random Access Memory
HDL	Hardware Description Language
I/O	Inputs and Outputs
IC	Integrated Circuit
ITRS	International Technology Roadmap for Semiconductors
IEEE	Institute of Electrical and Electronic Engineers
MFP	Mean Free Path
MOS	Metal Oxide Semiconductor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MWCNT	Multi-Walled Carbon Nanotube
NCNFET	N-type Carbon Nanotube Field-Effect Transistor
NEGF	Non-Equilibrium Green's Function
PCNFET	P-type Carbon Nanotube Field-Effect Transistor
RAM	Random Access Memory
SB	Schottky Barrier
S/D	Source/Drain
SNM	Static Noise Margin
SRAM	Static Random Access Memory
SWCNT	Single-Walled Carbon Nano Tube
VHDL	VHSIC Hardware Description Language
VHDL-AMS	VHSIC Hardware Description Language Analog and Mixed-Signal
VHSIC	Very High Speed Integrated Circuit
VLSI	Very Large Scale Integration

APPENDIX C. VHDL-AMS

VHDL-AMS (Very high-speed integrated circuit Hardware Description Language – Analog Mixed Signal) is a standardized language used for describing digital, analog, and mixed-signal systems. The Institute of Electrical and Electronics Engineers (IEEE) standardized the VHDL-1076 language as a Hardware Description Language (HDL) for digital models. The VHDL standard from 1993 was extended in 1999 for the description of analog and mixed-signal models in the form of the IEEE 1076.1 standard for VHDL-AMS.

VHDL and VHDL-AMS models consist of two parts: an entity declaration and one or more architecture descriptions [31, 32]. The entity describes the interface of the model and declares inputs, outputs, constant value parameters, conservative pins, etc. The architecture defines the behavior or structure of the model. It is possible to associate multiple architectures with an entity declaration, and only the selected architecture will be used during simulation by Ansoft Simplorer.

```
ENTITY entity_name IS
        GENERIC (generic_list); -- optional generic list
        PORT (port_list); -- input/output signal ports
END [ENTITY] name;

ARCHITECTURE architecture_name OF entity_name IS
        … declarations
BEGIN
        … sequential/simultaneous statements
END [ARCHITECTURE] [architecture_name];
```

Figure C.1 General Syntax of VHDL-AMS [32].

Port: Specifies dynamic information to be communicated between a model and its environment for all architectures. A port can be represented by a quantity, terminal, or signal. The mode of a port defines the directions of the signals on that port, and the most commonly used ones are IN, OUT and INOUT. Signal objects can use the mode IN, OUT, and INOUT; quantity objects can use IN and OUT whereas terminals have no direction mode.

Declarations: are pieces of information used in the model description. Declarations include data types, constants, signals, files, components, attributes, subprograms, and others.

Sequential statements: Statements that are executed in the order in which they appear. They define algorithms for the execution of a subprogram or process.

Simultaneous statements: Statements that are executed at the same time with respect to each other.

An important component in VHDL-AMS is **package**. A VHDL-AMS package is a way of grouping a collection of related declarations that serve a common purpose. They might be a set of subprograms that provide operations on a particular type of data, or they might just be the set of declarations needed to model a particular design. The important thing is that they can be collected together into a separate design unit that can be worked on independently and reused in different parts of a model. A VHDL-AMS package comprises a package declaration part and a package body. The declaration part represents the portion of the package that is visible outside of that package.

```
PACKAGE identifier IS
        … package_declarative_item
END [PACKAGE] [identifier];

PACKAGE BODY identifier IS
        … package_body_declarative_item
END [PACKAGE BODY] [identifier];
```

Figure C.2 Syntax Rule for VHDL-AMS Package [32].

Models suitable to be described in VHDL can be categorized into three domains: function, structure and geometry [32]. The functional domain is concerned with the operations performed by the system. In a sense, this is the most abstract domain of description, since it does not indicate how the function is implemented. The structural domain deals with how the system is composed of interconnected subsystems. The geometric domain deals with how the system is laid out in physical space. The SRAM is

described in a combination of structural and functional/behavioral methods. Examples of VHDL and VHDL-AMS codes are shown in Figure C.3 and Figure C.4 respectively.

```
--VHDL CODE FOR 16X1 BIT RAM
--Author: MENGISTEAB GEBREMEDHIN
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY ram IS
   PORT ( d_in : IN  STD_LOGIC;
        en : IN  STD_LOGIC;
        wr_rt : IN  STD_LOGIC;
        addr : IN  STD_LOGIC_VECTOR (3 DOWNTO 0);
        d_out : OUT  STD_LOGIC);
END ram;

ARCHITECTURE behavioral OF ram IS

        TYPE memcap IS ARRAY (15 DOWNTO 0) OF STD_LOGIC;
        SIGNAL mcap : memcap;

BEGIN
PROCESS (wr_rt, en, addr, mcap)
BEGIN
        IF en = '1' THEN
                IF (wr_rt = '1' AND wr_rt'event) THEN
                        mcap(CONV_INTEGER(addr)) <= d_in;
                END IF;
        END IF;
                d_out <= mcap(CONV_INTEGER(addr));
END PROCESS;
END behavioral;
```

Figure C.3 VHDL Code for RAM.

```
--VHDL-AMS CODE FOR RESISTOR
--Author: MENGISTEAB GEBREMEDHIN
LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;

ENTITY resistor01 IS
        PORT (TERMINAL p, n: ELECTRICAL);
END ENTITY resistor01;

ARCHITECTURE behavioral OF resistor01 IS
        QUANTITY res : RESISTANCE := 20.0E+4;
        QUANTITY v ACROSS i THROUGH p TO n;
BEGIN
        v == i*res;
END ARCHITECTURE behavioral;
```

Figure C.4 VHDL-AMS Code for a Resistor.

APPENDIX D. VHDL-AMS CODES OF THE COMPONENTS OF THE CNFET

```
---------- VHDLAMS MODEL cap_cbd ----------
LIBRARY IEEE;  USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;  USE mgncnfet_lib02.cntfetpack02.fetequations00;
ENTITY cap_cbd IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY cap_cbd;
ARCHITECTURE arch_cap_cbd OF cap_cbd IS
        TERMINAL Gate, Sub, Drain_b , Drain_ch, Source_L1 , Drain_L1 : ELECTRICAL;
        QUANTITY v_sub ACROSS Sub TO ELECTRICAL_REF;
        QUANTITY v_vd ACROSS Drain_L1 TO ELECTRICAL_REF;
        QUANTITY v_vg ACROSS Gate TO ELECTRICAL_REF;
        QUANTITY v_vs ACROSS Source_L1 TO ELECTRICAL_REF;
        QUANTITY v_ddin ACROSS Drain_b TO Drain_ch;
        QUANTITY v_dinsin  ACROSS Drain_ch TO Source_L1;
        QUANTITY Cap : CAPACITANCE;
        QUANTITY v ACROSS i THROUGH p TO m;
        QUANTITY C_charge : CHARGE;
BEGIN
        Cap == fetequations00 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
        C_charge == Cap * v ;
        i == C_charge'dot;
END ARCHITECTURE arch_cap_cbd;
---------- VHDLAMS MODEL cap_cbs ----------
LIBRARY IEEE;  USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;   USE mgncnfet_lib02.cntfetpack02.fetequations01;
ENTITY cap_cbs IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY cap_cbs;
ARCHITECTURE arch_cap_cbs OF cap_cbs IS
        --insert the terminals and quantities
   QUANTITY Cap : CAPACITANCE;
   QUANTITY v ACROSS i THROUGH p TO m;
   QUANTITY C_charge : CHARGE;
BEGIN
        Cap == fetequations01 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
        C_charge == Cap * v ;
        i == C_charge'dot;
END ARCHITECTURE arch_cap_cbs;
---------- VHDLAMS MODEL cap_cgb ----------
LIBRARY IEEE;   USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;   USE mgncnfet_lib02.cntfetpack02.fetequations02;
ENTITY cap_cgb IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY cap_cgb;
```

```
ARCHITECTURE arch_cap_cgb OF cap_cgb IS
        --insert the terminals and quantities
   QUANTITY Cap : CAPACITANCE;
   QUANTITY v ACROSS i THROUGH p TO m;
   QUANTITY C_charge : CHARGE;
BEGIN
        Cap == fetequations02 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
        C_charge == Cap * v ;
        i == C_charge'dot;
END ARCHITECTURE arch_cap_cgb;
---------- VHDLAMS MODEL cap_cgd ----------
LIBRARY IEEE;    USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;   USE mgncnfet_lib02.cntfetpack02.fetequations03;
ENTITY cap_cgd IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY cap_cgd;
ARCHITECTURE arch_cap_cgd OF cap_cgd IS
        --insert the terminals and quantities
   QUANTITY Cap : CAPACITANCE;
   QUANTITY v ACROSS i THROUGH p TO m;
   QUANTITY C_charge : CHARGE;
BEGIN
        Cap == fetequations03 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
        C_charge == Cap * v ;
        i == C_charge'dot;
END ARCHITECTURE arch_cap_cgd;
---------- VHDLAMS MODEL cap_cgs ----------
LIBRARY IEEE;  USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;   USE mgncnfet_lib02.cntfetpack02.fetequations04;
ENTITY cap_cgs IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY cap_cgs;
ARCHITECTURE arch_cap_cgs OF cap_cgs IS
        --insert the terminals and quantities
   QUANTITY Cap : CAPACITANCE;
   QUANTITY v ACROSS i THROUGH p TO m;
   QUANTITY C_charge : CHARGE;
BEGIN
        Cap == fetequations04 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
        C_charge == Cap * v ;
        i == C_charge'dot;
END ARCHITECTURE arch_cap_cgs;
---------- VHDLAMS MODEL cbdd ----------
LIBRARY IEEE;  USE IEEE.ELECTRICAL_SYSTEMS.ALL;
ENTITY cbdd IS
   PORT (TERMINAL p, m : ELECTRICAL);
```

```
END ENTITY cbdd;
ARCHITECTURE arch_cbdd OF cbdd IS
        CONSTANT Ldd : REAL := 32.0e-9;
        CONSTANT Csub : REAL := 20.0E-12;
        QUANTITY Cap : CAPACITANCE;
        QUANTITY v ACROSS i THROUGH p TO m;
         QUANTITY C_charge : CHARGE;
BEGIN
    Cap == 0.5*Csub*Ldd;
    C_charge == Cap * v ;
     i == C_charge'dot;
END ARCHITECTURE arch_cbdd;
---------- VHDLAMS MODEL cbss ----------
LIBRARY IEEE;   USE IEEE.ELECTRICAL_SYSTEMS.ALL;
ENTITY cbss IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY cbss;
ARCHITECTURE arch_cbss OF cbss IS
        CONSTANT Lss : REAL := 32.0e-9;
        CONSTANT Csub : REAL := 20.0E-12;
        QUANTITY Cap : CAPACITANCE;
        QUANTITY v ACROSS i THROUGH p TO m;
        QUANTITY C_charge : CHARGE;
BEGIN
    Cap == 0.5*Csub*Lss;
    C_charge == Cap * v ;
     i == C_charge'dot;
END ARCHITECTURE arch_cbss;
---------- VHDLAMS MODEL cgdd ----------
LIBRARY IEEE;    USE IEEE.ELECTRICAL_SYSTEMS.ALL;
ENTITY cgdd IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY cgdd;
ARCHITECTURE arch_cgdd OF cgdd IS
        CONSTANT Ldd : REAL := 32.0e-9;
        CONSTANT Cof : REAL := 30.0E-12;
        QUANTITY Cap : CAPACITANCE;
        QUANTITY v ACROSS i THROUGH p TO m;
        QUANTITY C_charge : CHARGE;
BEGIN
    Cap == 0.5*Cof*Ldd;
    C_charge == Cap * v ;
     i == C_charge'dot;
END ARCHITECTURE arch_cgdd;
---------- VHDLAMS MODEL cgss ----------
LIBRARY IEEE;   USE IEEE.ELECTRICAL_SYSTEMS.ALL;
```

```
ENTITY cgss IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY cgss;
ARCHITECTURE arch_cgss OF cgss IS
          CONSTANT Lss : REAL := 32.0e-9;
          CONSTANT Cof : REAL := 30.0E-12;
          QUANTITY Cap : CAPACITANCE;
          QUANTITY v ACROSS i THROUGH p TO m;
          QUANTITY C_charge : CHARGE;
BEGIN
    Cap == 0.5*Lss*Cof;
    C_charge == Cap * v ;
     i == C_charge'dot;
END ARCHITECTURE arch_cgss;
---------- VHDLAMS MODEL cgsub ----------
LIBRARY IEEE;   USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;  USE mgncnfet_lib02.cntfetpack02.fetequations09;
ENTITY cgsub IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY cgsub;
ARCHITECTURE arch_cgsub OF cgsub IS
          --insert the terminals and quantities
          QUANTITY Cap : CAPACITANCE;
          QUANTITY v ACROSS i THROUGH p TO m;
          QUANTITY C_charge : CHARGE;
BEGIN
    Cap ==  fetequations09 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
    C_charge == Cap * v ;
     i == C_charge'dot;
END ARCHITECTURE arch_cgsub;
---------- VHDLAMS MODEL cgtg ----------
LIBRARY IEEE;   USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;   USE mgncnfet_lib02.cntfetpack02.fetequations10;
ENTITY cgtg IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY cgtg;
ARCHITECTURE arch_cgtg OF cgtg IS
          --insert the terminals and quantities
          QUANTITY Cap : CAPACITANCE;
          QUANTITY v ACROSS i THROUGH p TO m;
           QUANTITY C_charge : CHARGE;
BEGIN
    Cap ==  fetequations10 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
    C_charge == Cap * v ;
     i == C_charge'dot;
END ARCHITECTURE arch_cgtg;
```

```
---------- VHDLAMS MODEL ibtbt ----------
LIBRARY IEEE;   USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;   USE mgncnfet_lib02.cntfetpack02.fetequations05;
ENTITY ibtbt IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY ibtbt;
ARCHITECTURE arch_ibtbt OF ibtbt IS
        --insert the terminals and quantities
   QUANTITY i THROUGH p TO m;
BEGIN
   i == fetequations05 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
END ARCHITECTURE arch_ibtbt;
---------- VHDLAMS MODEL isemi ----------
LIBRARY IEEE;   USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;   USE mgncnfet_lib02.cntfetpack02.fetequations06;
ENTITY isemi IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY isemi;
ARCHITECTURE arch_isemi OF isemi IS
        --insert the terminals and quantities
   QUANTITY i THROUGH p TO m;
BEGIN
i == fetequations06 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
END ARCHITECTURE arch_isemi;
---------- VHDLAMS MODEL rsbd ----------
LIBRARY IEEE;  USE IEEE.ELECTRICAL_SYSTEMS.ALL;
ENTITY rsbd IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY rsbd;
ARCHITECTURE arch_rsbd OF rsbd IS
        QUANTITY r : RESISTANCE;
        QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
        r == 1.0E3;
        v == i*r;
END ARCHITECTURE arch_rsbd;
---------- VHDLAMS MODEL rsbs ----------
LIBRARY IEEE;  USE IEEE.ELECTRICAL_SYSTEMS.ALL;
ENTITY rsbs IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY rsbs;
ARCHITECTURE arch_rsbs OF rsbs IS
        QUANTITY r : RESISTANCE;
        QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
        r == 2.0E3;
```

```
        v == i*r;
END ARCHITECTURE arch_rsbs;
---------- VHDLAMS MODEL rsd ----------
LIBRARY IEEE;   USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;   USE mgncnfet_lib02.cntfetpack02.fetequations07;
ENTITY rsd IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY rsd;
ARCHITECTURE arch_rsd OF rsd IS
        --insert the terminals and quantities
        QUANTITY r : RESISTANCE;
         QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
        r == fetequations07 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
        v == i*r;
END ARCHITECTURE arch_rsd;
---------- VHDLAMS MODEL rss ----------
LIBRARY IEEE;   USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;     USE mgncnfet_lib02.cntfetpack02.fetequations08;
ENTITY rss IS
   PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY rss;
ARCHITECTURE arch_rss OF rss IS
        --insert the terminals and quantities
        QUANTITY r : RESISTANCE;
        QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
         r == fetequations08 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
        v == i*r;
END ARCHITECTURE arch_rss;
---------- VHDLAMS MODEL vchel02 ----------
LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;
LIBRARY mgncnfet_lib02;
USE mgncnfet_lib02.cntfetpack02.fetequations11;
ENTITY vchel02 IS
        PORT (TERMINAL p, m : ELECTRICAL);
END ENTITY vchel02;
ARCHITECTURE arch_vchel02 OF vchel02 IS
        --insert the terminals
   QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
        v == fetequations11 (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
END ARCHITECTURE arch_vchel02;
```

APPENDIX E. VHDL-AMS PACKAGE OF THE CNFET

```
LIBRARY IEEE;
USE IEEE.ALL;
PACKAGE cntfetpack02 IS
FUNCTION fetequations(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN
REAL_VECTOR;
FUNCTION fetequations00(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations01(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations02(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations03(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations04(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations05(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations06(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations07(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations08(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations09(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations10(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
FUNCTION fetequations11(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL;
END PACKAGE cntfetpack02;

PACKAGE BODY cntfetpack02 IS
CONSTANT  a: REAL := 0.2495E-9;              -- The carbon lattice constant
CONSTANT CNTPos : REAL := 1.0;               -- position of CNT: 1 for edge, 0 for middle
CONSTANT Csub : REAL := 20.0E-12;        -- coupling capacitance between channel and substrate
CONSTANT d : REAL := 0.144E-9;               -- The carbon PI-PI bond distance
CONSTANT de_fac : REAL := 4.0 ;              -- fitting parameter
CONSTANT Dout : REAL := 1.0;                 -- fitting parameter
CONSTANT Efi : REAL := 0.6;                  -- The n+/p+ doped CNT fermi level (eV)
CONSTANT epso : REAL := 8.854E-12;           -- Dielectric constant in vacuum
CONSTANT FacR : REAL := 0.4;                 -- fitting parameter
CONSTANT h  : REAL := 6.626E-34;             -- Planck constant
CONSTANT h_ba : REAL := 1.06E-34;            -- reduced Planck constant
CONSTANT k  : REAL := 8.617E-5;              -- Boltzmann constant
CONSTANT Klowk : REAL := 2.0;                -- The dielectric constant of low-k material
CONSTANT Kox : REAL :=  16.0;                -- The dielectric constant of high-K oxide
CONSTANT Kgate : REAL := Kox;
CONSTANT Ksub : REAL :=  3.9;                -- The dielectric constant of substrate dielectric
CONSTANT L_relax : REAL := 40.0E-9;          -- fitting parameter
CONSTANT L_sd : REAL := 32.0e-9;             -- n+CNT source/drain full length
CONSTANT lambda_op : REAL :=  15.0E-9;    -- The Optical Phonon backscattering MFP
CONSTANT Lceff : REAL := 200.0e-9;           -- The mean free path in intrinsic CNT
CONSTANT Lch: REAL := 32.0E-9;               -- channel length
CONSTANT Ldd : REAL := L_sd;                 -- length of doped CNT drain side
CONSTANT Leff : REAL := 15.0E-9;             --  The mean free path in p+/n+ doped CNT
CONSTANT Lg : REAL := 32.0E-9;
```

```
CONSTANT Lss : REAL := L_sd;                -- length of doped CNT source side
CONSTANT n1 : REAL := 14.0;                 -- chirality vector coordinate
CONSTANT n2 : REAL := 0.0;                  -- chirality vector coordinate
CONSTANT phi_M : REAL := 4.6;               -- Metal workfunction
CONSTANT phi_S : REAL := 4.5;               -- CNT workfunction
CONSTANT photon : REAL := 0.16 ;            -- The photon energy, typical value 0.16eV
CONSTANT pi : REAL := 3.1415926;            -- the constant π
CONSTANT Pitch : REAL := 20.0E-9;           -- distance between adjacent CNTs
CONSTANT q  : REAL := 1.6E-19;              -- electronic charge
CONSTANT Rcnt : REAL := 3.3E3;              -- n+ CNT resistance
CONSTANT Rud : REAL := Rcnt*(1.0-FacR);          -- Drain side contact resistance
CONSTANT Sout : REAL := 1.0;                -- fitting parameter
CONSTANT sub_pitch : REAL := 6.4E-9;        -- sub-lithographic pitch
CONSTANT Temp: REAL := 300.0;               -- Temperature of operation in Kelvin
CONSTANT Tox : REAL := 4.0E-9;              -- thickness of high-k dielectric
CONSTANT Vpi  : REAL := 3.033;              -- carbon  π-π bond energy
CONSTANT hgate: REAL := 64.0E-10;           -- height of the gate
CONSTANT hsub: REAL := 10.0E-6;             -- thickness of the substrate
CONSTANT kT : REAL := k*TEMP;               -- The KT constant
CONSTANT tubes: REAL :=1.0;                 -- number of CNTs
CONSTANT Wgate: REAL :=sub_pitch;           -- width of gate

FUNCTION max(x,y: REAL) RETURN REAL IS
        VARIABLE z: REAL;
    BEGIN
        IF x>=y THEN
            z:=x;
        ELSE
             z:=y;
        END IF;
        RETURN z;
END FUNCTION max;

FUNCTION hspsqrt(x: REAL) RETURN REAL IS
                BEGIN
                        RETURN sgn(x)*SQRT(ABS(x));
END FUNCTION hspsqrt;

FUNCTION hspln(x: REAL) RETURN REAL IS
    BEGIN
                        RETURN sgn(x)* LOG(ABS(x))/0.43429448;
END FUNCTION;
FUNCTION Min(x,y: real) RETURN REAL IS
        variable z: real;
    BEGIN
        if x<=y then
            z:=x;
```

```
        else
            z:=y;
        end if;
        return z;
END FUNCTION Min;

FUNCTION sgn(x: REAL) RETURN REAL IS
        variable sgn: REAL;
        BEGIN
        IF (x>0.0) THEN
                        sgn := 1.0;
        ELSIf (x<0.0) THEN
                        sgn:= -1.0;
        ELSE
                        sgn := 0.0;
          END IF;
END sgn;

FUNCTION hsppow (x, y: REAL) RETURN REAL IS
        begin
                        RETURN x**y;
END FUNCTION hsppow;

FUNCTION fetequations(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN
REAL_VECTOR IS
        VARIABLE results: REAL_VECTOR(0 TO 11);
        VARIABLE cdelta_phib: REAL;
        VARIABLE Qcnt: REAL;
            --  declare all the variables similar to these
BEGIN
Lgate := min(Lg, 100.0E-9);                                     -- for short channel
dia := a*hspsqrt(hsppow(n1,2.0)+n1*n2+hsppow(n2,2.0))/pi;       -- from equation (2-2)
rad := dia/2.0;
Lgeff := 200.0*dia/1.5;
coeffj := 4.0*(q/h)*q;
Csub_tot := Csub+0.0;
offset := photon+0.0;
vds := v_vd - v_vs;
vvg := v_vg;
vvsubm := v_sub;
vvd := v_vd;
vvs := v_vs;

-- THE WAVE NUMBERS
K1 := 2.0*pi/(3.0*a*hspsqrt(hsppow(n1,2.0)+n1*n2+hsppow(n2,2.0)));   -- from equation (2-5)
K2 := 2.0*K1;                   -- from equation (2-5) and equation (2-6)
Kp1 := 2.0*pi/Lgate;            -- from equation (2-7)
```

```
Kp2:=2.0*Kp1;
Kp3:=3.0*Kp1;
Kp4:=4.0*Kp1;
Kp5:=5.0*Kp1;
Kp6:=6.0*Kp1;
Kp7:=7.0*Kp1;
Kp8:=8.0*Kp1;
Kp9:=9.0*Kp1;
Cprefactor := 2.0*pi*Kgate*epso;
CoeffE := hspsqrt(3.0)*a*Vpi/2.0;
CocoJ := hspsqrt(3.0)*a*pi*Vpi;
```

-- CARRIER ENERGIES

```
E1 := Vpi*pi/hspsqrt(3.0*(hsppow(n1,2.0)+n1*n2+hsppow(n2,2.0)));       -- from equation (2-8)
E2 := 2.0*E1;    -- band gap from equation (2-8)
E11 := CoeffE*hspsqrt(hsppow(K1,2.0)+hsppow(Kp1,2.0));
                    -- calculating E12 - E19 and E21 – E29 similar to E11 based on equation (2-8)
```

-- TO GET THE CAPACITANCE BETWEEN THE GATE AND THE CHANNEL

```
Hoxcn := Tox+rad;
lamda01 := (Kgate-Ksub)/(Kgate+Ksub);                                   -- from equation (2-62)
DCinf := (hspln(2.0*Hoxcn/dia+hspsqrt(hsppow(2.0*Hoxcn/dia,2.0)-1.0)))+
(lamda01*hspln((2.0*Hoxcn+2.0*dia)/(3.0*dia)));                         -- from equation (2-61)
DCadj :=  (0.5*hspln((hsppow(Pitch,2.0)+2.0*(Hoxcn-rad)*(Hoxcn+hspsqrt(hsppow(Hoxcn,2.0)-
hsppow(rad,2.0))))/(hsppow(Pitch,2.0)+2.0*(Hoxcn-rad)*(Hoxcn-hspsqrt(hsppow(Hoxcn,2.0)-
hsppow(rad,2.0))))))+ (0.5*lamda01*hspln((hsppow(Hoxcn+dia,2.0)+hsppow(Pitch,2.0))/
(9.0*hsppow(rad,2.0)+hsppow(Pitch,2.0)))*tanh((Hoxcn+rad)/(Pitch-dia))); -- from equation (2-60)
Cedge := Cprefactor/(DCinf+DCadj);                                      -- from equation (2-58)
Cmid := 2.0*Cedge-Cprefactor/DCinf;                                     -- from equation (2-59)
Cox := Cedge*CNTPos+Cmid*(1.0-CNTPos);                                  -- from equation (2-57)
```

-- TO GET THE CHANNEL SURFACE POTENTIAL CHANGE

```
qcnt_sub10 := 1.0/(1.0+exp((E1-cdelta_phib)/kT))+1.0/(1.0+exp((E1-cdelta_phib+(vvd-vvs))/kT));
        -- calculating sub11 – sub29 similar to qcnt _sub10 based on equation (2-12)
        -- calculating qcnt _sub as the summation of sub10 upto sub29
                        Qcap := Cox*(vvg-vvs)+Csub*(vvsubm-vvs)-(Cox+Csub)*cdelta_phib;
                        Qcnt := de_fac*q*(charge_sub)/Lgate;
                        Qcnt := Qcap;
                                -- from equations (2-11), (2-12) and (2-13)
delta_phib := cdelta_phib; -- assigning the calculated deltaphib
Ci := Cox;

En11 := E11 - E1;
        -- calculating En12 – En19 similar to En11
En21 := E21 - E2;
        -- calculating En22 – En29 similar to En21
```

-- TO GET Isemi

```
Coeff_J11 := Kp1/hspsqrt(hsppow(K1,2.0)+hsppow(Kp1,2.0));
        -- calculating the remaining coefficients similar to Coeff_J11  from equation (2-9)
fermi_s11 := 1.0/(1.0+exp((E11-delta_phib)/kT));
        -- calculating the remaining Fermi probabilities similarly from equations (2-9) and (2-10)
fermi_d11 := exp(-vds/kT)/(exp(-vds/kT)+exp((E11-delta_phib)/kT));
        -- calculating the remaining Fermi probabilities similarly from equations (2-9) and (2-10)
DOS11 := (E11-offset)/hspsqrt(abs(hsppow((E11-offset),2.0)-hsppow(E1,2.0)))*max(En11-
offset,1.0e-14);
        -- calculating the remaining density of states similarly from equations (2-16) and (2-18)
fermi_op11 := exp((offset-vds)/kT)/(exp((offset-vds)/kT)+exp((E11-delta_phib)/kT));
fermi_op11_0 := exp((offset-0.0)/kT)/(exp((offset-0.0)/kT)+exp((E11-delta_phib)/kT));
        -- calculating the distributions for optical scattering from equations (2-16) and (2-17)
l_op11 := lambda_op/(DOS11*(1.0-fermi_op11));
l_op11_0 := lambda_op/(DOS11*(1.0-fermi_op11_0));
        -- calculating the remaining optical scattering MFPs similarly from equation (2-16)
T11 := l_op11/( l_op11+Lg);
        -- calculating the remaining transmission probabilities similarly from equation (2-20)
T11_0 := l_op11_0/( l_op11_0+Lg);
        -- calculating the remaining transmission probabilities similarly from equation (2-21)
current_sub11 := (T11*fermi_s11 - T11_0*fermi_d11)*Coeff_J11;
        -- calculating the remaining current values similarly from equation (2-10)
        -- calculating current _sub as the summation of sub_currents from equation (2-10)
```

-- TO GET THE PARASITIC CAPACITANCES

```
qnt_c_sub10:=exp((E1-delta_phib)/kT)/hsppow(1.0+exp((E1-delta_phib)/kT),2.0)+exp((E1-
delta_phib+vvd-vvs)/kT)/hsppow(1.0+exp((E1-delta_phib+vvd-vvs)/kT),2.0);
        -- calculating the quantum capacitance values from equations (2-32) and (2-33)
        -- calculating qnt_charge_sum as the summation of the quantum capacitances
qnt_charge := 4.0*q*q*(qnt_charge_sum)/(Lgate*kT);          -- from equations (2-32) and (2-33)
parac_r := (Cox+Csub_tot+qnt_charge)/Cox;
```

-- TO GET Ibtbt

```
ids02 :=  coeffj*CocoJ/Lgate*(current_sub);
Efield := (vds+Efi-delta_phib)/L_relax;                    -- from equation (2-26)
meff1 := 2.0*h_ba*h_ba*K1/(hspsqrt(3.0)*a*Vpi);            -- from equation (2-25)
meff2 := 2.0*h_ba*h_ba*K2/(hspsqrt(3.0)*a*Vpi);            -- from equation (2-25)
Ef1 := 2.0*hspsqrt(2.0)*q*h_ba*Efield/(pi*E1*hspsqrt(meff1*E1));
Ef2 := 2.0*hspsqrt(2.0)*q*h_ba*Efield/(pi*E2*hspsqrt(meff2*E2));
Tbtbt1 := pi*pi/9.0*exp(-1/Ef1);                           -- from equation (2-24)
Tbtbt2 := pi*pi/9.0*exp(-1/Ef2);                           -- from equation (2-24)
Ibtbt_sub_1 := hspln((1.0+exp((vds-E1+ids02*Rud-Efi)/kT))/(1.0+exp((E1+ids02*Rud-
                Efi)/kT)))*max(vds-2.0*E1,0.0)/(vds-2.0*E1);   -- from equation (2-23)
Ibtbt_sub_2 := hspln((1.0+exp((vds-E2+ids02*Rud-Efi)/kT))/(1.0+exp((E2+ids02*Rud-
                Efi)/kT)))*max(vds-2.0*E2,0.0)/(vds-2.0*E2);   -- from equation (2-23)
```

```
-- TO GET THE PARASITIC RESISTANCES
CoeffRd:= (1.0-Dout)*(Ldd-Leff)/Leff + Dout*Ldd/Leff;          -- from equation (2-38)
CoeffRs:=Lss/Leff;
Eref := max(Efi,E1+0.01);                                      -- doping level
Qef := 2.0*(hspsqrt(hsppow(Eref,2.0)-hsppow(E1,2.0))*max(sgn(Eref-
E1),0.0)+hspsqrt(hsppow(Eref,2.0)-hsppow(E2,2.0))*max(sgn(Eref-E2),0.0));
                                                               -- from equation (2-42)
dphib_max1 := hspsqrt(hsppow(hsppow(E1,2.0)-hsppow(E2,2.0),2.0)+2.0*(hsppow(E1,2.0)+
                hsppow(E2,2.0))*hsppow(Qef,2.0)+hsppow(Qef,4.0))/(2.0*Qef)-Eref;
dphib_max2 := hspsqrt(hsppow(Qef,2.0)+hsppow(E1,2.0))-Eref;
diff_phib := dphib_max2 - dphib_max1;
phib_max:=dphib_max1*max(sgn(diff_phib),0.0)+dphib_max2*max(sgn(-diff_phib),0.0);
        -- maximum surface potential change from equation (2-41)
Vd_max:=phib_max+Eref-E1;
Vds00 := max(abs(v_ddin)/CoeffRd,1.0e-9);
delta_phib00 := (phib_max/Vd_max)*max(min(Vds00,Vd_max),1.0e-9);
        -- surface potential change from equation (2-40)
Gc:= (2.0+(kT/Vds00)*(hspln((1.0+ exp((E1-Eref-delta_phib00)/kT))/(1.0+(exp((E1-Eref-
delta_phib00)/kT))*exp(Vds00/kT)))+hspln((1.0+ exp((E2-Eref-delta_phib00)/kT))/(1.0+( exp((E2-
Eref-delta_phib00)/kT))*exp(Vds00/kT)))))*coeffj;
        -- quantum conductance of the doped CNT from equation (2-39)
Rc:=1.0/Gc;

-- TO GET Cgtg
Tbk := exp(2.0-2.0*hspsqrt(1.0+2.0*(hgate+Lg)/L_sd));
        -- a parameter based on equation (2-65)

-- TO EXPORT THE VALUES OF THE COMPONENTS

results(0):= ABS(Csg*Csub_tot/Cox);                                    -- Cbd
results(1):= ABS(Csg*Csub_tot/Cox);                                    -- Csb
results(2):= abs(Csub_tot/parac_r *Lg);                                -- Cbg
results(3):= abs((Cox-(Cox+Csub_tot)/parac_r)*Lg/2.0) ;                -- Cgd
results(4):= abs((Cox-(Cox+Csub_tot)/parac_r)*Lg/2.0);                 -- Cgs
results(5):=  coeffj*kT*(Ibtbt_sub_1*Tbtbt1+Ibtbt_sub_2*Tbtbt2);      -- Ibtbt
results(6):= coeffj*CocoJ/Lgate*(current_sub) ;                        -- Isemi
results(7):= max(CoeffRd*Rc,1.0e-4) ;                                  -- Rsd
results(8):= max(CoeffRs*Rc,1.0e-4);                                   -- Rss
results(9):= 2.0*pi*Lg*ksub*epso/hspln(4.0*hsub/hgate);                -- Cgsub
results(10):= (ksub*epso*hgate)+
(0.7*pi*ksub*epso)*(L_sd+Lg)/hspln(2.0*pi*(L_sd+Lg)/(2.0*Lg+Tbk*hgate));  -- Cgtg
results(11):= v_dinsin*Lch/(Lch+Lgeff);                                -- Vchel
RETURN results;

END FUNCTION fetequations;
```

```
FUNCTION fetequations00(v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin: REAL)  RETURN REAL
IS
VARIABLE Ivalues: REAL_VECTOR(0 TO 11);
VARIABLE i: REAL;
BEGIN
        Ivalues := fetequations (v_vg, v_vd, v_vs, v_sub, v_ddin, v_dinsin);
i := Ivalues(0);
RETURN i;
END FUNCTION fetequations00;

        --  setting similar codes for the rest functions returning results(1) up to results(11)

END PACKAGE BODY cntfetpack02;
```

APPENDIX F. SIMPLORER

Simplorer is a software package developed by ANSYS Inc. It is used to design and analyze complex technical systems. Simulation models created with Simplorer can contain circuit components from different physical domains, block elements, and state machine structures modeled in SML (Simplorer Modeling Language) as well as VHDL-AMS [33]. The various tools used for modeling, simulating, and analyzing are integrated within the Simplorer application which manages the project files, sets options for both simulation and program environment, runs analyses, and generates reports. Simplorer performs calculations for simulation models described in VHDL-AMS.

The Ansoft Simplorer interface that we have used for our works looks like the figure indicated on Figure F.1. The major Simplorer interface elements for an existing project with its associated schematics as shown in Figure F.1 can be described as:

Design Area Window: Displays one or more editor windows such as the Schematic Editor, model editors, and symbol editor. It also displays various report windows.

Menu Bar: Provides various menus that enable us to perform Simplorer tasks, such as managing project files, designs, and libraries; customizing desktop components; drawing objects; and setting and modifying project parameters and options.

Toolbars: Provides buttons that act as shortcuts for executing various commands.

Project Manager Window and Project Tree: The Project Manager window shows all the components, models, symbols, and other elements of each design in the project. Each project has its own expandable Project Tree. Many operations on the design elements such as DC analysis, transient analysis and AC analysis can be performed directly from the Project Manager window.

Properties Window: Displays the attributes of a selected object in the active model, such as the object's name, electrical or other associated physical quantities, orientation, and color. It also displays information about a selected component.

Message Manager Window: Displays error, informational, and warning messages for the active project.

Progress Window: Displays solution progress information.

Status Bar: Shows current actions and provides instructions.

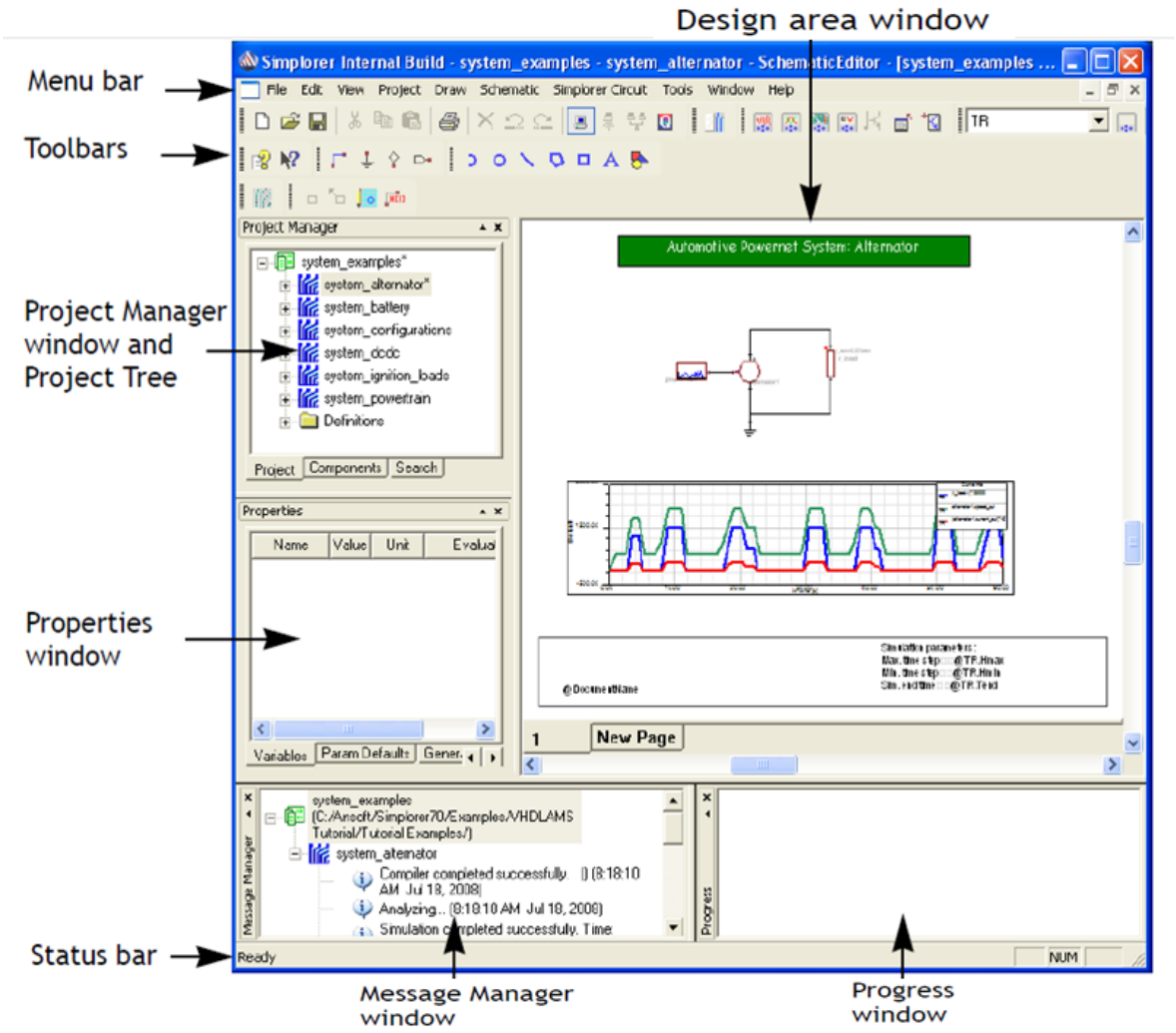

Figure F.1 Ansoft Simplorer Software Interface.

APPENDIX G: COMPLETE 8-BIT ARCHITECTURE OF THE SRAM