Design and Performance Benchmarking of Hybrid FinFET/MTJ-based Logic-In-Memory 7:2 compressor for high-speed application

by

Rajdeep Parmar

202011042

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY

in

INFORMATION AND COMMUNICATION TECHNOLOGY

to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

June, 2022

Declaration

I hereby declare that

(i) the thesis comprises my original work towards the degree of Master of Technology in Information and Communication Technology at DA-IICT and has not been submitted elsewhere for a degree,

(ii) due acknowledgment has been made in the text to all the reference material used.

Signature of Student

Certificate

This is to certify that the thesis work entitled Design and Performance Benchmarking of Hybrid FinFET/MTJ based logic-in-memory 7:2 compressor has been carried out by Rajdeep Parmar (202011042) for the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology under my/our supervision.

20/07/2022

Dr. Yash Agrawal

Thesis Supervisor

Acknowledgments

I would like to thank Dr. Yash Agrawal for his invaluable assistance and insights leading to the writing of this thesis. My sincere thanks also go to Dr. Kavicharan Mummaneni from NIT, Silchar for their patience and understanding. With their noble guidance, support, encouragement, and enthusiasm, they have helped me continue and successfully finish this research. Dr. Sreeja Rajendran, Dr. Yash Agrawal, and Dr. Biswajit Mishra, for their valuable feedback on the betterment of the research work.

I would like to thank DAIICT for assisting me throughout the year and for their invaluable support and guidance.

Contents

Abstract

Spintronics can be a very good candidate for the replacement of complementary metal-oxide-semiconductor (CMOS) technology. A digital signal processor (DSP) has a multiplier as a key element in the circuit. Multiplier having larger propagation delay becomes the bottleneck for the chip designed because of the component being in the critical path due to larger delay. So, the parallel multiplier is used to reduce the propagation delay in which partial products have to be added which can be done by the compressor. Compressors with less propagation delay give an advantage and are more efficient. This thesis work focuses on the 7:2 compressor which has 7 inputs and 2 outputs giving compression from 7 to 2. This proposed work is implemented using the Logic-in-memory architecture of spintronics. Given work uses magnetic tunnel junction (MTJ). This proposed work gives a lesser propagation delay compared to work [1]. This work has a reduction of ~11.9% in the propagation delay with nearly PDP. This thesis work can also be used with logic-in memory (LIM) architecture because of the Non-volatile nature of the STT-MTJ. This proposed work is based on the FinFET/MTJ structure simulated on HSpice software with 16nm technology of FinFET. For a favorable comparison, the circuits are redesigned. The 7:2 compressor circuit proposed in [1] is redesigned using the 16nm FinFET technology. This circuit based on solely FinFET is then compared with the FinFET/MTJ-based hybrid structure. The performance benchmark of the given work gives 244.36ps of the 7:2 compressor. The performance benchmarking of the FinFET/MTJ-based hybrid structure with the conventional 7:2 compressor shows an improvement in the propagation delay. Given the proposed 7:2 compressor will be used in the high-speed application multiplication. Although the application of the design is not limited to multiplication, it can also be used for image compression.

List of Principal Symbols

r	Radius
T_{OX}	Thickness of Oxide
R_P	Parallel state of MTJ
R_{AP}	Velocity of light
W	Width
L	Length
θ_0	Initial magnetic angle
θ_c	Critical angle
K	Anisotropic constant
α	Damping constant
M_{s0}	Default saturation magnetization
H_0	Intrinsic magnetic field
F	Tunneling conductivity
I_{LB}	Left-branch current
I_{RB}	Right-branch current
R_L	Left-branch resistance
R_R	Right-branch resistance
I_{th}	Threshold current
I_c	Critical current

List of Acronyms

DSP	Digital signal processor
STT	Spin transfer torque
MTJ	Magnetic tunnel junction
PCSA	Pre-charge sense amplifier
LIM	Logic-in-memory
CNTFET	Carbon nanotube field-effect transistors
FeFET	Ferroelectric field-effect transistors
PMOS	P-channel metal oxide semiconductor
NMOS	N-channel metal oxide semiconductor
CMOS	Complementary metal oxide semiconductor
FM	Ferromagnetic layer
NM	Non-magnetic layer
P-state	Parallel state
AP-state	Anti-parallel state

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Motivation

Silicon technology is about the exploration and manipulation of the charge of an electron. Silicon technology uses the transportation of charge. The transportation of charge gives current in the path and can be used in the further part of the circuit. In silicon technology, there are many devices like bipolar junction transistor (BJT), metal oxide semiconductor transistor (MOS), and complementary metal-oxide-semiconductor transistors (CMOS) that plays important role in the designing of circuits like microprocessors, chip, etc. As Moore's law is about the integration of the number of transistors. Moore predicted that the number of transistors integrated per chip will double every 18 months which can be seen in Figure 1.1. This continuous scaling of the transistors gives performance enhancements and gives continuous sets of goals for the advancements in technology. But due to this continuous scaling, it also gave some short-channel effects at the near 10nm regime of the transistors. These short-channel effects bottleneck the performance of the given chip. Because at near 10nm regime technology of the transistor short-channel effects like quantum tunneling effect, leakage current, mobility degradation, hot carrier effects, drain induced barrier lowering occurs. This led to many serious problems increasing static power dissipation being volatile. With this continuous scaling of the silicon devices, the frequency of the operation increases but this higher frequency increment gives rise to dynamic power dissipation. Due to high power consumption, the temperature of the chip increases in turn giving less performance. This scaling down of the power supply gives a better solution but this comes at the expense of silicon area and higher efficient power gating techniques. So, these different short-channel effects downgrade the performance of the chip at a near 10nm regime. In continuation of better performance improvements, different technology has to be explored. So, many researches are going on for the replacement of the current silicon technology. Many good replacement candidates for current silicon technology are ferroelectric FET (FeFET) [2], quantum computers, graphene and carbon nanotubes, nanomagnetic

logic, and spintronics [3]. These technologies are replacements for charge-based technology.

Among these technologies, spintronics is about the exploration of the spin of the electron. The discovery of giant magnetoresistance led to the development of the spintronics field. Spintronics devices tend to consume less power because of their near-zero standby current and are non-volatile. Spintronics is a good replacement for silicon technology though it has the capability to integrate it with current silicon technology. Giving easy integration capability with our current electronics circuits-based devices. As silicon technology loses its stored charge giving leakage current, spintronics uses the magnetization of an electron that has an indefinite retention time. So, circuits like magnetic arithmetic logic unit [4], and multilevel cell STT-MRAM [5,6] based on spintronics give low power consumption, non-volatility, lower read-time, and easy integration with CMOS. Being a good replacement for CMOS technology, this thesis work will explore the spintronics for the development of the 7:2 compressor.

Current silicon technology uses von-Neuman architecture in which the central processing unit and memory are different units so data travels back and forth between two units. When reaching near 10nm regime interconnect delay plays role in propagation delay. The spintronics structure used in this work is the logic-in-memory (LIM) structure. LIM structure consists of in this case FinFET/MTJ. LIM structure gives an advantage over the von Neumann structure because the logic unit and memory unit are in each other.

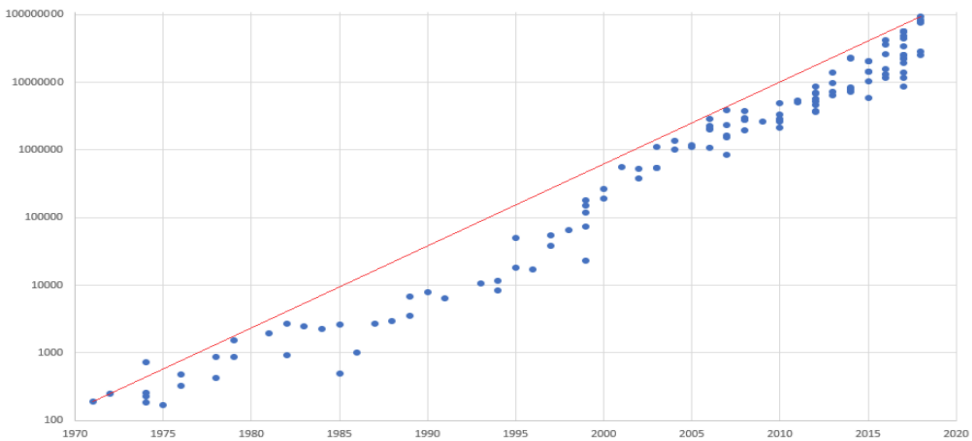


Figure 1.1 Moore's law [7]

1.2 Basics

Spintronics is about the exploration of the spin degree of the electron. There are 5 basics type of materials named as diamagnetic, paramagnetic, ferromagnetic, ferrimagnetic, and antiferromagnetic materials. Ferromagnetic material has magnetic properties. Each electron of the ferromagnetic material will have either spin-up or spin-down which can be shown in Figure 1.2. As shown in Figure 1.2, when this type of electron is in ferromagnetic material it gives net magnetization. Magnetization is the material property where it gives direction (being a vector) expressing the magnetic dipole moments. Depending upon the spin-up or spin-down electron ferromagnetic material gives 2 directions of the magnetization being 180^{o} apart from each other. The development of the giant magnetoresistance (GMR) effect led to the development of spintronics and further development of the tunnel magnetoresistance (TMR) effect led to the development of the magnetic tunnel junction (MTJ) device. MTJ device consists of 3 layers of material. There will be two ferromagnetic materials having a nonmagnetic material in-between. As there are two ferromagnetic materials is there in the MTJ, each one will have a magnetization on the easy axis. MTJ in its current state of the art uses bi-directional current(I) for defining its state.

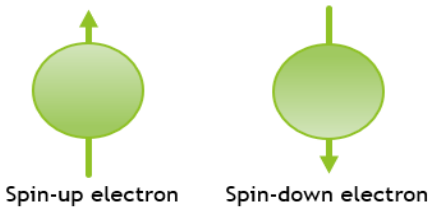


Figure 1.2 Spin of the electron

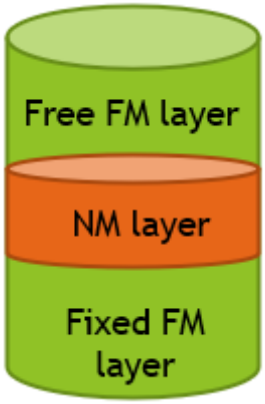


Figure 1.3 Magnetic Tunnel Junction (MTJ) [8]

$$TMR = \frac{R_{AP} - R_P}{R_P}$$

Using MTJ logic-in-memory structure can be developed. LIM structure is a hybrid where CMOS technology and MTJ are used. MTJ can be fabricated with current technology.

LIM structure compared to current silicon technology uses low power dissipation because of non-volatile. LIM structure has a faster reading capability and can be integrated with CMOS. LIM structure has infinite endurance. [8 - 10].

1.3 Organization of the thesis

This thesis is organized into the following chapters.

In Chapter 2, the basic background of the Thesis has been introduced and discussed. This chapter will focus on the background of FinFET, and Spintronics. Based on spintronics, there are many different devices that have been discussed in this section. For the switching of the MTJ, there are many different techniques like FIMS, TAS, and STT will be discussed.

The model of the STT-MTJ is important in circuit design. Chapter 3 will discuss the compact modeling used in this scope of work. The discussion on the compact modeling of the STT-MTJ will be continued with the equivalent RC network describing the behavior of the STT-MTJ. Model parameters that are important in the design of the model will be discussed. Model parameters value will be based on the material discussed in this.

Chapter 4 focuses on the explanation of the hybrid FinFET/MTJ circuit structure. This chapter will later give state-of-art circuit design and their performance benchmarking like magneto resistive random-access memory (MRAM), and magnetic full adder (MFA). For the sensing of the state of the resistance Pre-charge sense amplifier will be required which will be explained.

Chapter 5 will propose a 7:2 compressor circuit for the DSP, image compression, and image multiplication. Then Circuit of the 7:2 compressor has been proposed and performance benchmarking has been done on the conventional 7:2 compressor and the result is discussed.

Chapter 6 will discuss the result and give a conclusion on the proposed 7:2 compressor. The chapter will give references used in this thesis work and chapter 7 will contain all the references which are used in this scope of work.

Chapter 8 discusses the publication of this paper to be submitted.

Chapter 9 gives appendices on the code of the proposed 7:2 compressor.

Chapter – 2
Background

2.1 FinFET

As continuous scaling down of the CMOS led to many short-channel effects degrading the performance of the CMOS technology, different technology has been proposed. In this thesis, FinFET is being used having a technology node of 16nm. FinFET is having multi-gate transistor structures. FinFET transistor cross-section is shown in Figure 2.1. As shown in Figure 2.1, the source/drain structure forms the fin structure. Gate will cover the different sides of the FinFET's channel giving more control over the channel. As the gate is covering three sides of the channel, it provides good electrostatic control.

FinFET is a multi-gate MOSFET non-planar device. FinFET gives more controllability over the channel, a better subthreshold slope, smaller drain-induced barrier lowering (DIBL), and a higher current.

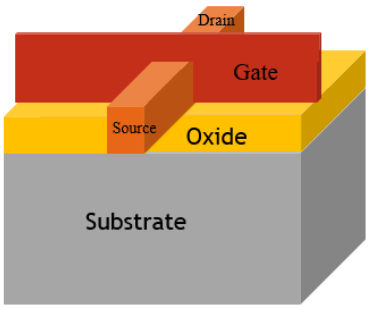


Figure 2.1 FinFET cross-section

2.2 Spintronics

Spintronics is about the exploration of the spin of the electron. In spintronics technology, the spin degree of the electron will be used. There are five types of material named ferromagnetic material, diamagnetic material, paramagnetic material, ferrimagnetic material, an antiferromagnetic material. In spintronics, ferromagnetic materials will be the main focus. Each electron in the ferromagnetic material has a spin degree of freedom. Their spin can be either spin-up or spin-down shown in Figure 2.2. This spin of the electron constitutes the magnetization of the ferromagnetic material.

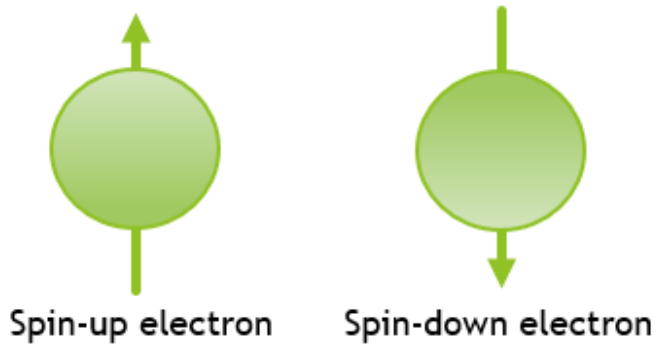


Figure 2.2 Spin of the electron (a) spin-up (b) spin-down

Based on the spin of the electron and magnetization of the material, a giant magnetoresistance effect has been observed by Albert Fert and Peter Grunberg. The giant magnetoresistance (GMR) effect shows that in a ferromagnetic material, the transportation of the electron can be affected by the magnetization of the material or vice versa. Due to the exploration of the GMR effect, the development of the spintronics devices continued.

There are mainly three phenomena: spin polarization, spin relaxation, and magnetoresistance which can be used for the spintronics basics.

Spin polarization occurs when the number of spin-up electrons is not equal to the number of spin-down electrons. This imbalance creates either magnetic moment in the material. When a current is passing through the magnetic material, because of the magnetization of the material spin-polarized current will be generated. When this current goes through another magnetic material, spin relaxation will occur which will try to make the balance of the spin-polarized electrons. Magnetoresistance will be used to detect the spin state of the spintronics device.

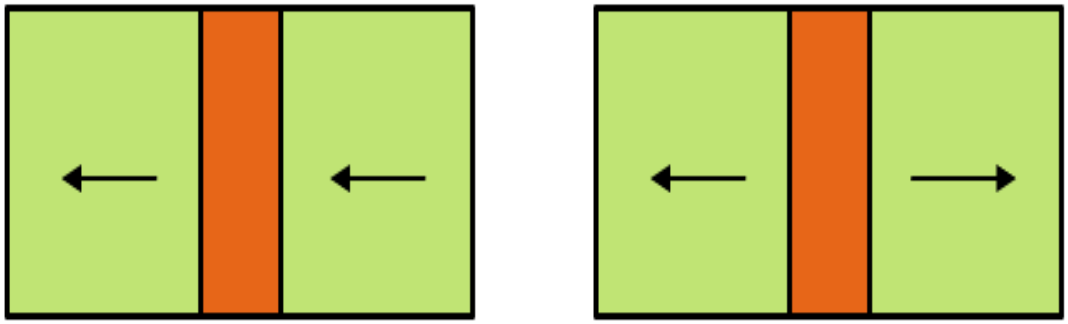


Figure 2.3 Spin Valve (a) Parallel state (b) Anti-parallel state [9]

GMR effect can be explained by spin-valve which is shown in Figure 2.3. Two ferromagnetic layers will be having non-magnetic metal layer in-between. The ferromagnetic layer (FM layer) will be having magnetization direction. When both the FM layer is having the same magnetization direction, then this state is known as the parallel state(P) as shown in Figure 2.3(a). When spin-up electrons are passed through the parallel state of the spin valve then the scattering of the electron will be less indicating low resistance. This low resistance will indicate a parallel state (P). Opposite to the P state, when the magnetization of the FM layer is opposite to each other is known as the Anti-parallel state (AP). In the AP state, the scattering of the electron will occur giving high resistance. GMR effect can be used in the read-head of the hard disk.

2.3 Spintronics devices

Using the spin-valve concept, many devices can be magnetic tunnel junctions (MTJ), GMR sensors, and spin transistors. In this scope of work, MTJ will be used and will mainly focus on hybrid logic and memory circuits.

2.3.1 Magnetic tunnel junction (MTJ)

2.3.1.1 MTJ structure

MTJ structure consists of two ferromagnetic layers and one thin insulating layer. When the non-magnetic metal layer is replaced with a thin oxide layer, this structure is known as the MTJ structure.

Magnetic coercivity [11] is a measure of the ability of a ferromagnetic material to withstand an external magnetic field. One of the FM layers will have high magnetic coercivity indicating magnetization in only one direction known as the fixed FM layer. The other layer which is the free FM layer will have low magnetic coercivity and the magnetization of the free layer can be changed by any other means.

2.4 Switching of the MTJ

(a) Field-induced magnetic switching (FIMS)

Field-induced magnetic switching will be having two current lines as shown in Figure 2.4. These current lines are orthogonal to each other and word line current (I_w) and bit line current (I_b). These orthogonal current lines create two magnetic fields that will be created named bit line magnetic field (H_b) and word line magnetic field (H_w). These current lines will require more than 10mA to generate magnetic fields. Due to the high current requirement, this method requires will have high power consumption having the drawback of speed.

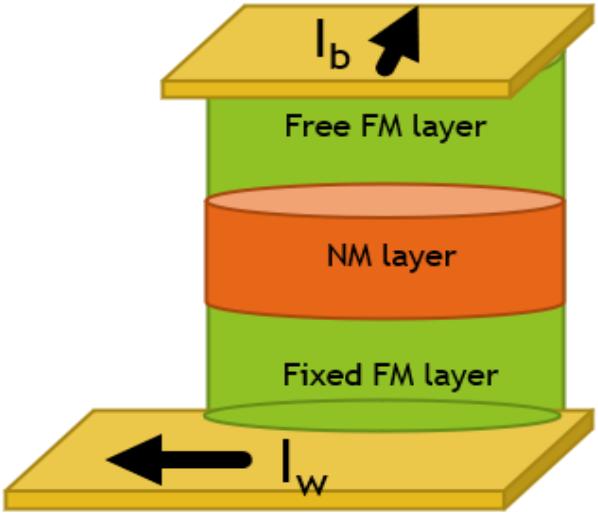


Figure 2.4 Field-induced magnetic switching (FIMS) [10]

(b) Thermally assisted switching (TAS)

An anti-ferromagnetic layer is added above the MTJ structure. This layer will flow current through it. Firstly, heat current (I_h) will flow through the anti-ferromagnetic layer which will heat the structure. After the I_h, bit line current (I_b) will flow through generating switching field. TAS will have relatively lower power consumption compared to FIMS. But due to higher heat current (I_h) cooling processes are required which will lower the speed so can't be used in the high-speed logic application.

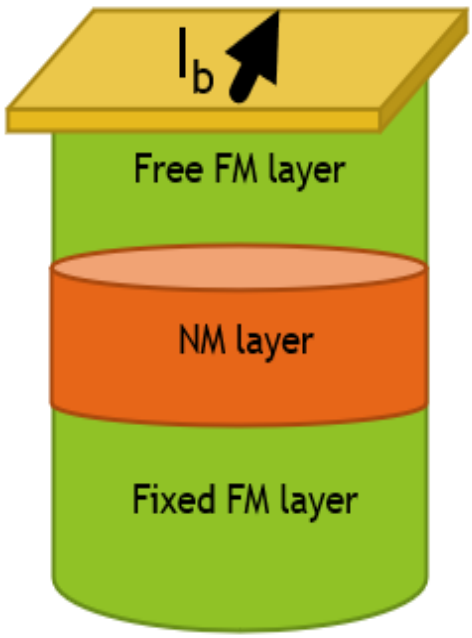
(c) Spin transfer torque (STT)

Spin transfer torque was proposed by Berger and Slonczewski in 1996 [12-13]. This effect shows spin-polarized current passing through the MTJ structure will affect the magnetization. This will create local magnetization which will change the magnetization of the free FM layer.

When the MTJ current (I_{MTJ}) is greater than the critical current (I_c), this will exert the STT-changing magnetization of the free layer. One of the FM layers will be used to generate spin-polarized current acting as a polarizer.

A bi-directional current will be used to change the state of the MTJ from a Parallel state (P) to an Anti-parallel state (AP) or vice versa.

STT will have less power consumption compared to the techniques previously discussed. In this thesis work, STT will be used to change the state of the spintronics devices and will be applied in further implementation of the hybrid LIM structure.



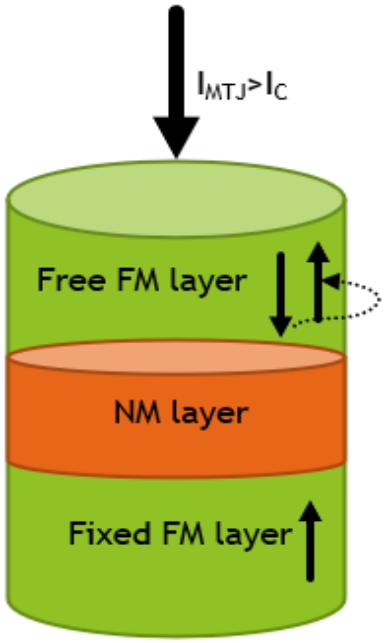


Figure 2.5 Thermally assisted switching [10] Figure 2.6 Spin transfer torque [10]

Chapter – 3
Compact modelling of MTJ

3.1 Introduction

In this chapter, a discussion on the MTJ compact model is done. This given compact model [14] considers different physical parameters. In this chapter, the compact model which is based on STT switching can be explained by the Landau-Lifshitz-Gilbert equation. First, different parameters are discussed relating to the physical behavior of the material, and based on the parameters this chapter will discuss the resistance of the MTJ based on the state of the MTJ. This model can be used in a SPICE-like simulator which gives the ability to integrate with current CMOS technology. This model is scalable with process and operation conditions. This model transfers the fundamental Landau-Lifshitz-Gilbert (LLG) equation into a passive RC network.

3.2 STT-MTJ Basics

STT-MTJ has three layers as mentioned in the chapter. These three layers are named as Fixed FM layer, NM layer, and Free FM layer. The magnetic orientation of the free FM layer will change based on the current passed through the MTJ. The radius (r) will indicate the radius of the MTJ and the thickness of the NM layer will be shown by T_{ox}. FM layer will have a magnetic moment ($\underset{M}{\rightarrow}$) in the given direction. Using the bi-directional current spin of the electron is polarized by the fixed layer changing the magnetic moment of the free FM layer. The dynamics of the magnetic moment will be defined by the LLG equation:

$$\frac{\overrightarrow{dM}}{dt} = -\gamma\mu_0\left(\underset{M}{\rightarrow}\right) \text{ x } \underset{H_{eff}}{\longrightarrow} - \gamma\frac{2K}{M_S^2}\left(\underset{M}{\rightarrow} . \underset{u_{ea}}{\longrightarrow}\right).\left(\underset{M}{\rightarrow} \text{ x } \underset{u_{ea}}{\longrightarrow}\right) + \frac{\propto}{M_S}\underset{M}{\rightarrow} \text{ x } \frac{\overrightarrow{dM}}{dt} + \eta\frac{\mu_B I}{eV} \qquad (3.1)$$

This LLG equation gives different torque components of the FM layer. Different torque components like Zeeman torque, anisotropic torque, damping torque, and spin-transfer torque can be observed in the LLG equation shown above.

3.3 Dynamics of the STT-MTJ

The magnetic moment of the MTJ is three-dimensional where Zeeman torque and anisotropic torque contribute to the rotation in the plane perpendicular to the easy axis whereas damping torque and spin-transfer torque is responsible for the switching in the easy plane. So, by this magnetic moment can be separated into two equations as shown in the below equations where the LLG equation [10] will be divided into scalar equations indicating two planes.

$$M_s \frac{d\Phi}{dt} = -\gamma\mu_0 M_s H \sin\theta - 2\gamma K \sin\theta\cos\theta \quad (3.2)$$

$$M_s \frac{d\theta}{dt} = -\alpha M_s \frac{d\Phi}{dt} + \eta \frac{\mu_B I}{eV} \quad (3.3)$$

$$M_s \frac{d\theta}{dt} = -\alpha\gamma\mu_0 M_s H \sin\theta - 2\alpha\gamma K \sin\theta\cos\theta + \eta \frac{\mu_B I}{eV} \quad (3.4)$$

θ and Φ indicate magnetic angles. Depending upon the eq. 4 there are three points giving a physical map for the compact model. These points are Threshold current, Critical angle, and Critical current.

Threshold current: This will differentiate the switching of the MTJ. When the current flowing through the MTJ is greater than the I_{th} then dθ/dt will be > 0 changing the magnetic angle and in turn state of the MTJ. If $I<I_{th}$ then dθ/dt will be < 0 then the solution of the MTJ switching will be indeterministic so the solution to the equation will be given by the dθ/dt = 0.

Critical angle: This angle gives the value of the magnetic moment where it needs to reach. If θ at time = τ is smaller than θc then the damping torque may pull θc back to 0^o.
When I = 0, there are three points of dθ/dt where 0^o and 180^o are stable.

Critical current (I_c): I_c is the minimum current required to change the angle to θc.

Table 3.1 Models of critical points

Critical point	Equation
θ_{th}	$cos^{-1}[\frac{\sqrt{(\mu_0 M_s H)^2 + 32K^2} - \mu_0 M_s H}{8K}]$
I_{th}	$\alpha\gamma\ (eV)\eta\mu B[\mu_0 M_s H \sin\theta_{th} + K \sin 2\theta_{th}]$
θ_c	$cos^{-1}\left[-\frac{\mu_0 M_s H}{2K}\right]$
I_c	$(\pi r^2)\ (2.1 \times 10^{-4}\ M_s/\tau + 0.34 M_s\ H + 4.26 \times 10^{10})$

3.4 TMR Model of the MTJ

The reading of the state will be done by the resistance of the MTJ. When the MTJ is in R_P then the current flowing through the MTJ will indicate free FM layer and fixed FM layer are in the same direction and R_{AP} will indicate anti-parallel. Tunnel magneto-resistance is defined by the below equation.

$$TMR = \frac{R_{AP} - R_P}{R_P} \tag{3.5}$$

Different resistance values in the transition of the MTJ will be defined by the following equation:

$$R(\theta) = 2R_P \left(\frac{1+TMR}{2+TMR+TMR*\cos\theta}\right) \tag{3.6}$$

3.5 Equivalent RC circuit

Based on the LLG equation sin θ can be approximated as θ, 1, -θ, when θ is close to 0^0, 90^0, 180^0. So, the LLG equation can be expressed as a linear function of the θ, and this can be shown as an equivalent RC network. This circuit shows the output node as θ while Rs indicates functions of the critical point shown in table 3.2 and C is a constant.

From the network and as shown in Figure 3.1 there will be four different regions depending on the critical point.

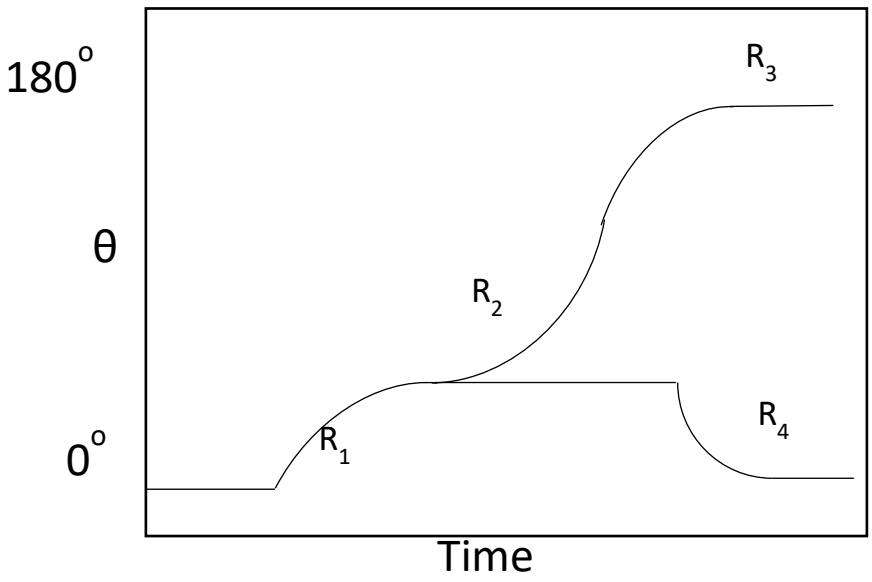


Figure 3.1 The regional RC network [14]

Region 1: When θ is close to 0 at the beginning of the current pulse, sin θ ~ θ and cos θ ~ 1. In this region, damping torque will resist the change in θ indicating the R_1C network is negative feedback. V_1 will charge the output node.

Region 2: When θ is close to 90^0 while θ exceeds the threshold angle θ_{th}, sin θ can be approximated as 1 and cos θ as $90^0 - \theta$. This can be indicated from the LLG equation and the R_2 negative resistance equation can be derived from the LLG equation which is indicated in table 3.3. R_2 will indicate an exponential increase in the magnetic angle. The magnetic angle θ will be 180^0 if the applied input current pulse stays long enough if it is not then region 3 and region 4 will indicate the behavior.

Region 3: When θ > critical angle θ_c while the current pulse ends, the damping torque will try to complete the switching obtaining the R_3C equation shown in table 3.3.

Region 4: When θ < critical angle θ_c while the current pulse ends, damping torque will try to make magnetic angle $0^{0,}$ and switching will fail to obtain the R_4C equation shown in table 4.1.

These equations with TMR equation (3.5) will define the behavior of the MTJ model. This model was developed using Verilog-AMS language and simulated in HSpice software.

The model parameter for the STT-MTJ has been discussed in table 3.3. This table will give details on the parameter used in the model and the code. This value is based on the material used.

Table 3.2 Formulas for the RC elements

Term	Equation
V_1	$\theta_{th} + a(I - I_{th})/(\pi r^2)$
R_1	K_1 $(\mu 0 M s H + 2K)/Ms$
R_2	K_2 [$1 - b(I - I_{th})/(\pi r^2)$] K / M_s
R_3	K_3 $(2K - \mu_0 M_s H)$
R_4	K_4 $(\mu_0 M_s H + 2K) M_s$

Table 3.3 Model Parameter

Symbol	Name in code	Unit	Value	Description
r	**r**	**M**	**20n**	**Radius of STT-MTJ**
Tox	**tox**	**m**	**0.85n**	**Oxide thickness of STT-MTJ**
$\theta 0$	**Ini**	**Rad**	**[0,π]**	**Initial magnetic angle**
K	**K**	**T*A/m**	**1.13E5**	**Anisotropic constant**
α	**a**	**1**	**0.027**	**Damping constant**
Ms0	**Ms0**	**A/m**	**4.81E5**	**Default saturation magnetization**
H0	**H0**	**A/m**	**1.5E5**	**Intrinsic magnetic field**
F	**F**	**$\Omega^{-1}m^{-1}$**	**221**	**Tunneling conductivity**
TMR0	**TMR0**	**1.5**	**1.5**	**Default tunnel magneto-resistance**

Chapter – 4
Hybrid FinFET/MTJ circuit structure

4.1 Introduction

As discussed earlier in the previous section, that compact model can be used to generate the spintronics-based circuit design. This hybrid structure gives performance improvement over conventional silicon technology.

In this chapter, various circuit design has been proposed. Firstly, the sensing scheme has been discussed for the sensing state of the MTJ. In the sensing scheme, a Pre-charge sense amplifier (PCSA) has been discussed in this scope. PCSA will have low power consumption and low read-sensing time.

The spintronics-based hybrid structure has been discussed later on. In this chapter, 1-bit magnetic memory (MRAM) and 1-bit magnetic full adder (MFA) have been discussed. Performance benchmarking of MRAM and MFA will be compared with the conventional silicon-based memory and full-adder.

4.2 Hybrid FinFET/MTJ structure

As Von Neumann's architecture is having logic units and memory units are different so the speed of the architecture is limited to the speed at which the CPU can get the data from the memory. Logic-in-memory (LIM) architecture will have a logic unit and a memory unit near each other. This will reduce the delay required to fetch the data.

Logic-in-memory architecture is shown in Figure 4.1. LIM structure consists of 3 circuits. 1. Sense amplifier 2. FinFET logic tree 3. MTJ. A sense amplifier will be used to sense the state of the MTJ. FinFET logic tree will evaluate the output based on the circuit designed and MTJ is used for storing the information.

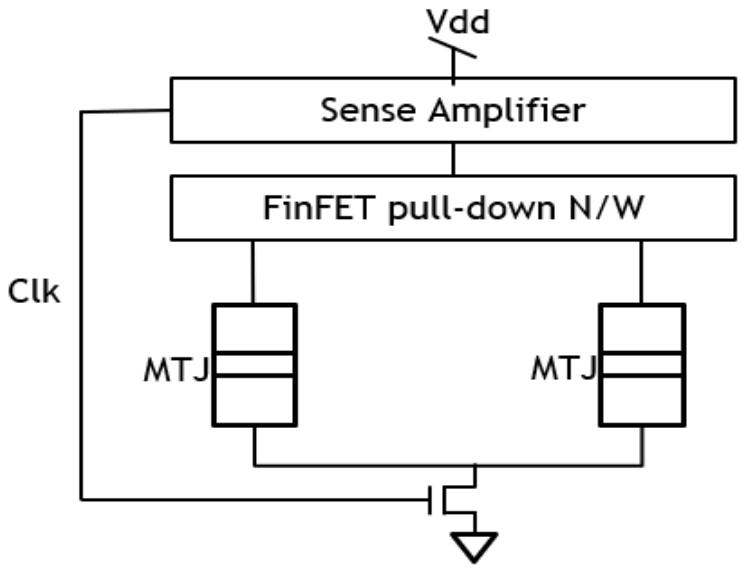


Figure 4.1 Logic-in-memory architecture [15]

4.3 Pre-charge sense amplifier (PCSA) sensing circuit

As continuous scaling down of CMOS technology occurs, this led to many short-channel effects that downgrade the performance of the chip. Using the spintronics technology, a hybrid structure is developed. MTJ used in the hybrid structure has a resistance state in it. For the integration of the hybrid structure with CMOS technology, this resistance state has to be converted to a logic voltage. This section will give a pre-charge sense amplifier (PCSA) [15]. for sensing the resistance state of the MTJ.

4.3.1 Structure of sensing circuit

The structure of the pre-charge sense amplifier (PCSA) is shown in Figure 4.2. PCSA works in two modes. 1. Pre-charge phase 2. Evaluation phase. The sensing operation of PCSA in detail is as follows.

During the pre-charge phase, when CLK = 0 then P-channel FinFET M1 and M4 will be turned on. As this FinFET M1, and M4 are in ON state, Q, $\bar{Q}$ will be charged to VDD. Here, Q, $\bar{Q}$ are output nodes indicating complementary output.

During the Evaluation phase, when CLK = 1 then N-channel FinFET M2, M3, M5, M6, and M7 will be turned on depending upon the MTJ state. There will be two

branch currents named left-branch current (I_{LB}) and right-branch current (I_{RB}) will flow through the left branch and right branch. If the left branch is having less resistance path, then $\bar{Q}$ will be discharged showing output 0 and as the right branch is having high resistance path then Q will be having output 1.

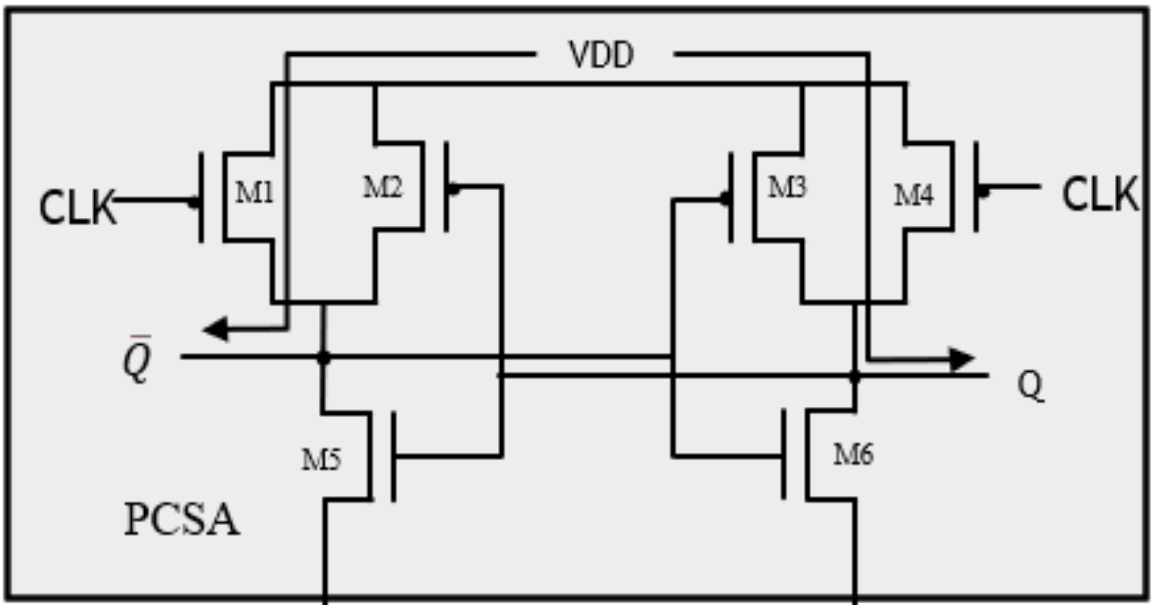


Figure 4.2 Pre-charge sense amplifier [15]

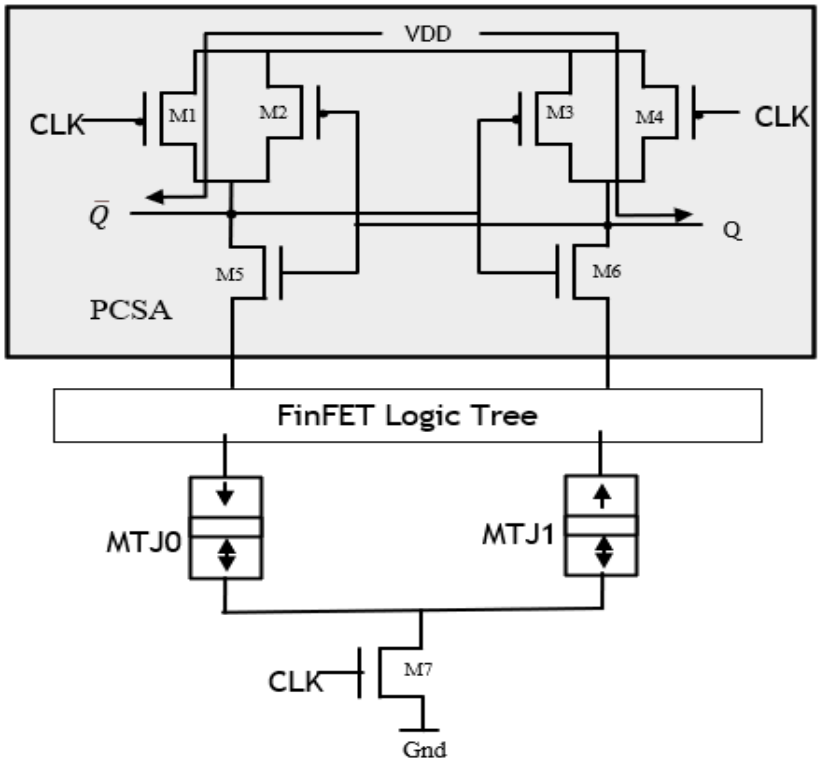


Figure 4.3 Pre-charge phase of PCSA [15]

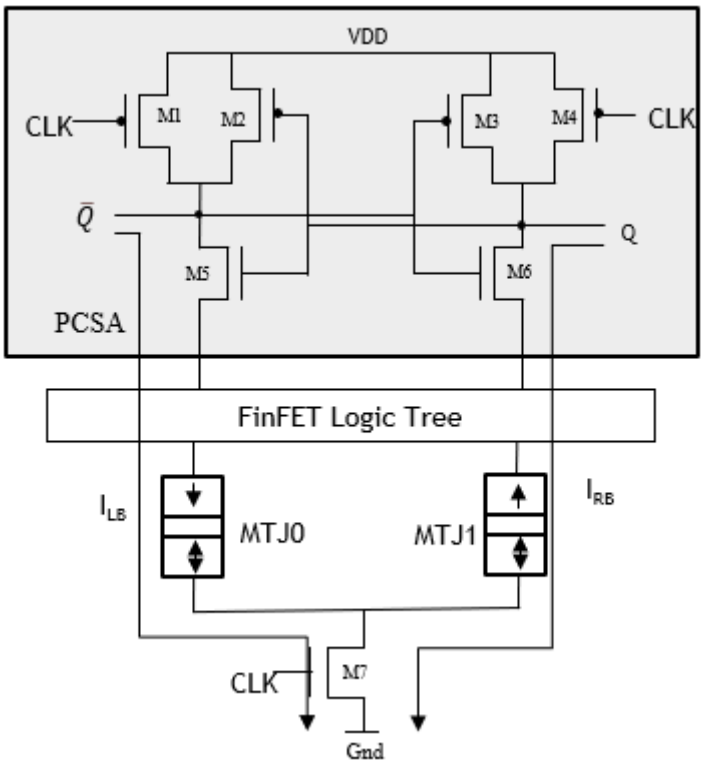


Figure 4.4 Evaluation phase of PCSA [15]

4.4 1-bit magneto resistive random-access memory (MRAM)

As current technology is having a scaling-related issue, SRAM, and DRAM will downgrade the performance of the memory. Spintronics provides less power consumption and because of the non-volatile nature of MTJ, it provides better performance over current silicon technology. Magneto resistive random-access memory (MRAM) has an indefinite endurance due to non-volatility. The circuit diagram and structure are shown in the below section.

4.4.1 1-bit MRAM

The structure of the 1-bit MRAM is shown in Figure 4.5. It consists of 1MTJ-1MOS. There will be three nodes named bit line, word line, and source line. The bit line and source line will be applied to the voltage source where a change of state and reading of the state is controlled by the word-line. These three modes of operation are named as 1. Writing operation 2. Hold operation 3. Read operation.

1. Writing operation – When the wordline is given logic '1' then it will form a close path between the bit line and source line. When the bi-directional current (I_{MTJ}) > critical current (I_C) is passing through the bit line and source line the state of MTJ will change as shown in Figure 4.6(a). This operation will write the state of the MTJ which is known as the writing operation.

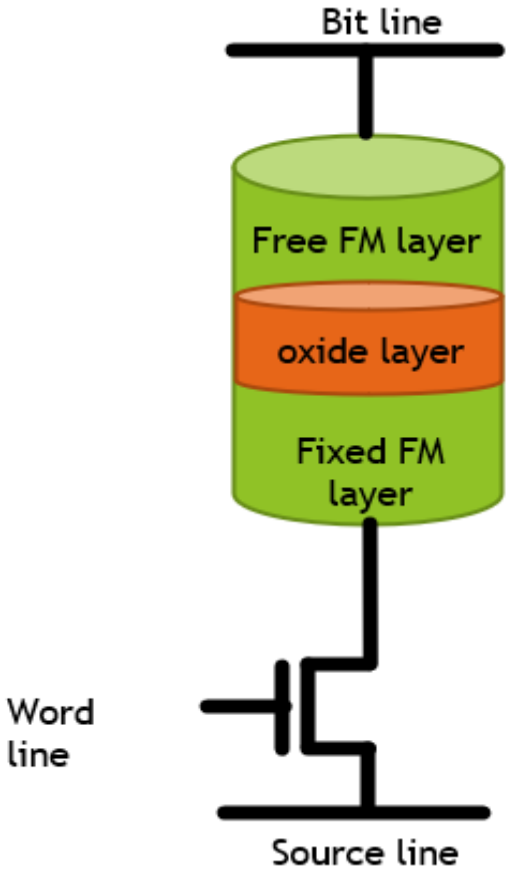


Figure 4.5 1-bit MRAM [6]

2. Hold operation – When the word line is given logic '0' then there will be no close path between the bit line and source line. As there is no current passing through the MTJ state the MTJ will remain in its state indicating non-volatile nature. So, the power consumption will be nearly zero.

3. Read operation – When logic '1' is applied and current less than the critical current (I_C) is passing through the MTJ and the sense amplifier can be used for sensing the state of the MTJ which can be seen in Figure 4.6(b).

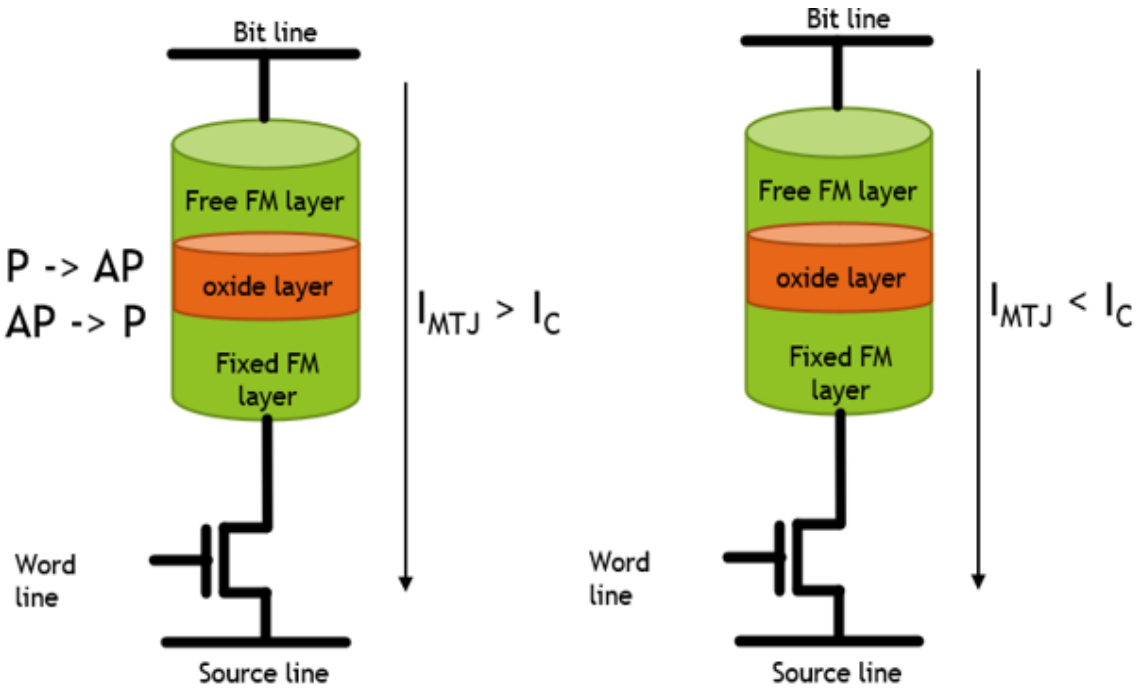


Figure 4.6 Operations of MRAM [6]

(a)Write operation of 1-bit MRAM (b) Read operation of 1-bit MRAM

4.4.2.1 1-bit complete MRAM cell

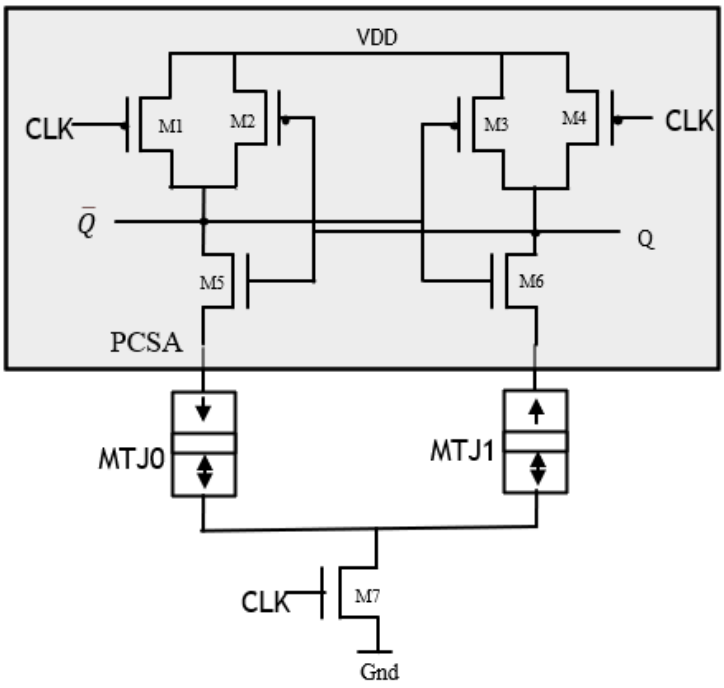


Figure 4.7 1-bit Complete MRAM cell

The structure of the complete MRAM cell is shown in Figure 4.7. Consider MTJ0 is in the AP state and MTJ1 is in the P state. When CLK = 0 then P-channel FinFET M1, M4 will be in ON state so output Q, $\overline{Q}$ will be charged to VDD in the pre-charge phase. When CLK = 1 then N-channel FinFET M2, M3, M5, M6, and M7 will be in the ON state. There will be two currents passing through the left arm and right arm. As MTJ0 is in AP state and MTJ1 is in P state low resistance path will be from the right arm, giving $I_{RB} > I_{LB}$ so Q will be discharged giving output Q = 0 and output $\overline{Q} = 1$.

Similarly, for the P and AP states, it can be seen in Table 4.1 output $\overline{Q}$ will be discharged showing output 0 and Q will give output 1.

Table 4.1 The output of MRAM cell

MTJ0 state	**MTJ1 state**	**Left-arm resistance**	**Right arm resistance**	**Current comparison**	**Q**	**$\overline{Q}$**
AP	P	R_{AP}	R_P	$R_L > R_H$	0	1
P	AP	R_P	R_{AP}	$R_L < R_H$	1	0

4.4.2.2 Simulation of MRAM cell – The simulation of the MRAM cell is shown in Figure 4.8. MTJ0 is in AP state and MTJ1 is in P state. Here, v(clk) is input, and v(q), v(qbar) is output.

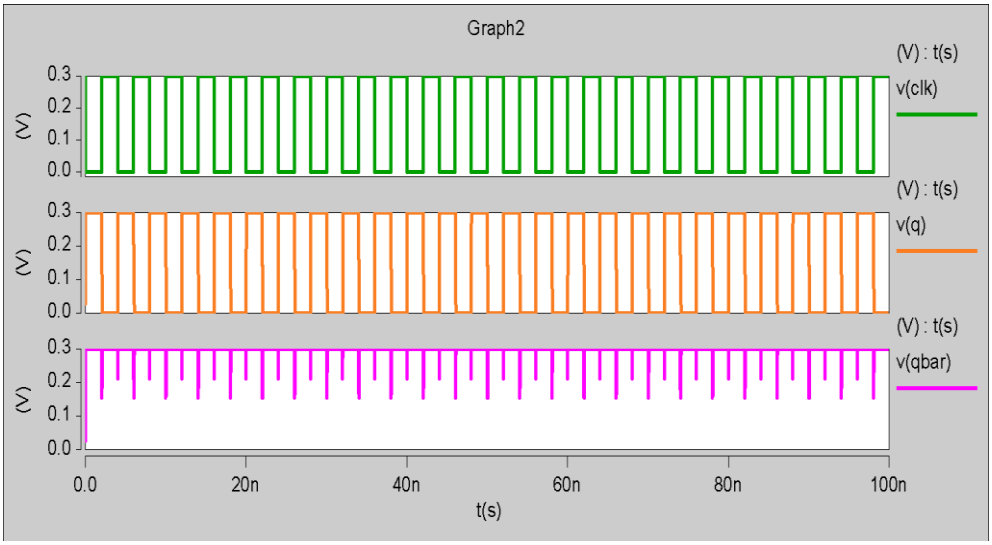


Figure 4.8 Simulation of MRAM cell

SRAM consists of 12 transistors [16]. STT-MRAM has a delay of 96.911ps while SRAM [16] has a delay of 386.1ps showing an improvement of around 290ps. SRAM is volatile while STT-MRAM is non-volatile having a retention time of >10 years. As STT-MRAM is non-volatile, power consumption will be less compared to SRAM.

Table 4.2 Performance comparison between SRAM and MRAM

Parameter	SRAM (12T) [16]	STT-MRAM
Delay	386.1ps	96.911ps
Power	5.26nW	18.001nW
PDP	$2030.886 * 10^{-21}$	$1744.49 * 10^{-21}$
Volatility	Volatile	Non-volatile
Retention time	Till power-up	>10 years

4.5 1-bit magnetic full adder (MFA)

Logic-in-memory (LIM) Structure of the 1-bit magnetic full adder (MFA) is shown in Figure 4.9. Inputs are A, B, and C_{in} and outputs are Sum, C_{out}. C_{in} is stored in MTJ. The sum can be realized using the FinFET structure as shown in Figure 4.9(a). When MOS is in OFF condition, it shows R_{OFF} MOS resistance and in ON condition, it shows R_{ON} resistance. When CLK = 0 then output Q, $\overline{Q}$ will be charged to VDD in the precharge phase and in the evaluation phase when CLK = 1 then output Q, $\overline{Q}$ will be evaluated based on the input A, B, and C_{in} applied to the circuit.

As described in Figure 4.9, the truth table for each test case has been given in table 4.3. A, B, and C are the inputs, and the Resistance comparison between the left-arm and right-arm is shown based on which output will be calculated as 0 or 1 and shown in Sum, C_{out}. If R_L is greater than the R_R then the current through the right arm will flow more compared to the left-arm giving 0.

The equation of the full adder is shown below in which C_{in} is stored in MTJs.

Sum = A ^ B ^ C_{in}

Carry = AB + BC_{in} + AC_{in}

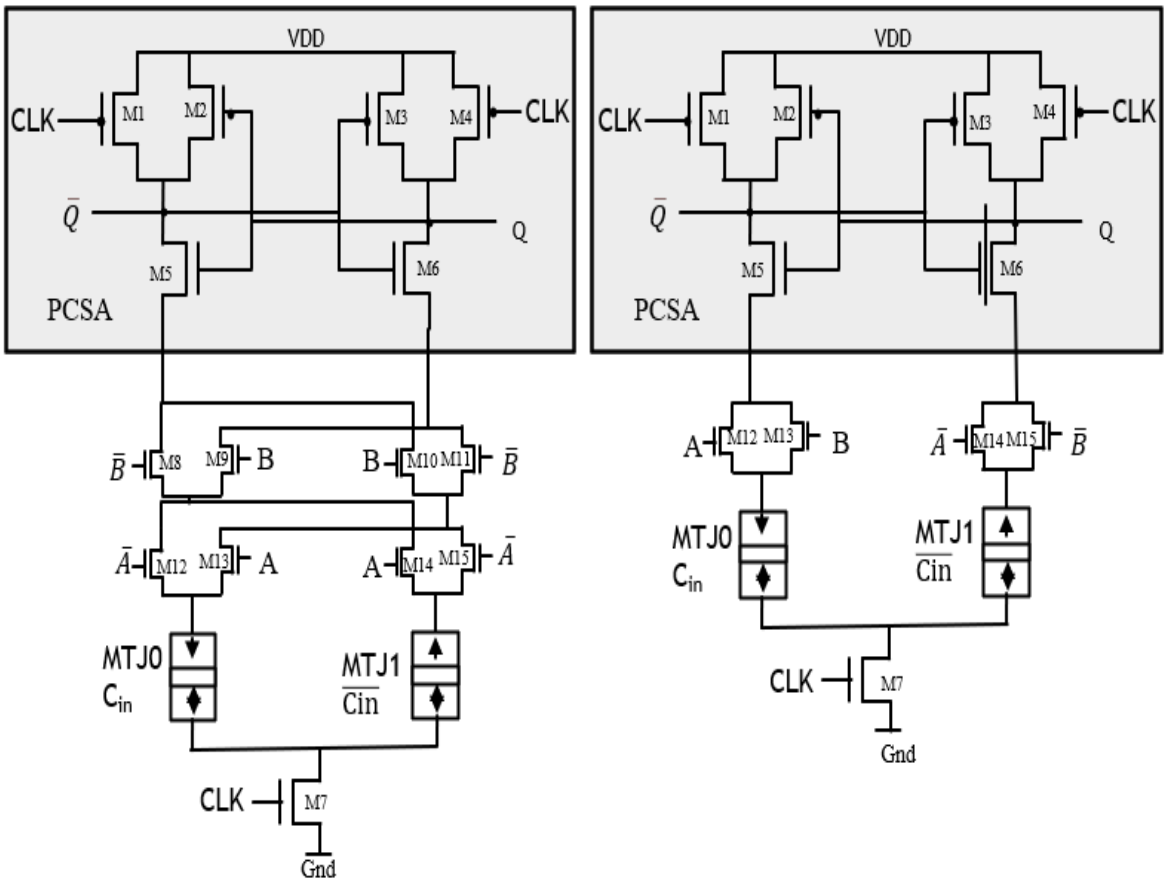


Figure 4.9 Magnetic full adder (MFA) [19] (a) Sum (b) C_{out}

Table 4.3. Truth table and resistance comparison of MFA

A	B	C	Resistance comparison for sum	Resistance comparison for C_{out}	Sum (Q)	C_{out} (Q)
0	0	0	$R_L > R_R$	$R_L > R_R$	0	0
0	0	1	$R_L < R_R$	$R_L > R_R$	1	0
0	1	0	$R_L < R_R$	$R_L > R_R$	1	0
0	1	1	$R_L > R_R$	$R_L < R_R$	0	1
1	0	0	$R_L < R_R$	$R_L > R_R$	1	0
1	0	1	$R_L > R_R$	$R_L < R_R$	0	1
1	1	0	$R_L > R_R$	$R_L < R_R$	0	1
1	1	1	$R_L < R_R$	$R_L < R_R$	1	1

4.5.1 Simulation and performance analysis of MFA

The simulation of the MFA is shown in Figure 4.10. MTJ0 will indicate C_{in} is in P state and MTJ1 will indicate AP state. As shown in Figure 4.10 when A=1, B=1, and C_{in}=1 then the output sum will indicate Sum = 1 and C_{out} = 1.

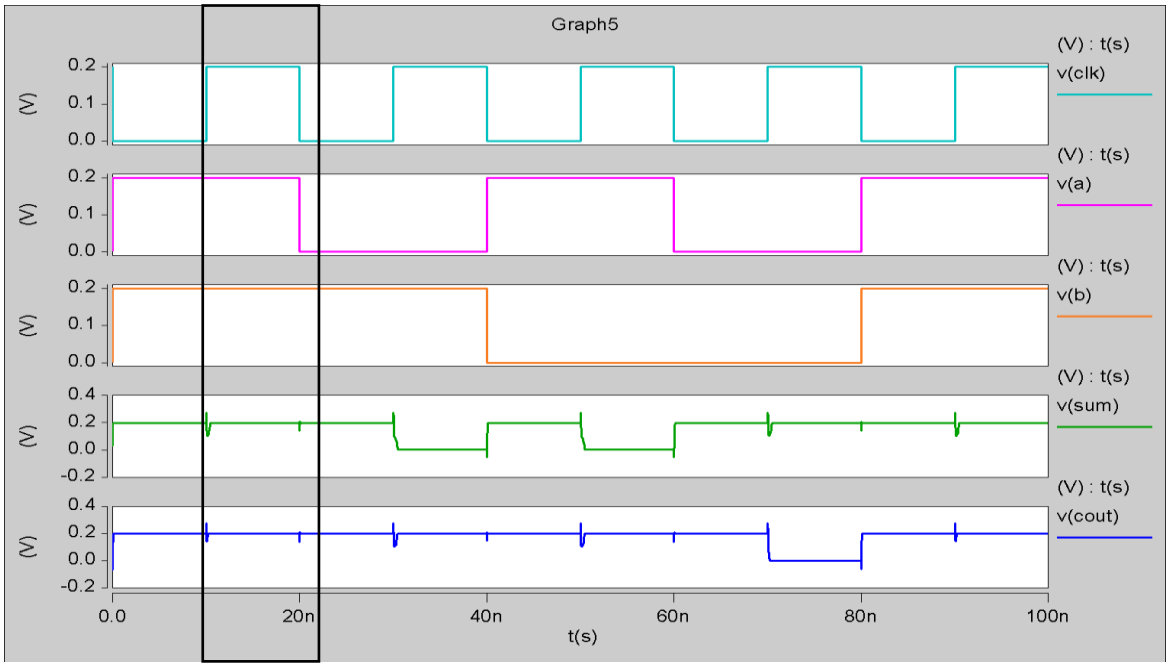


Figure 4.10 Simulation of MFA circuit

Table 4.4 Performance benchmarking with FinFET-based FA

Performance parameter	Conventional FinFET full adder	Magnetic full adder
Delay	51.673ps	86.722ps
Power	37.228nW	11.659nW
Power delay product (PDP)	1.923 aJ	1.01 aJ
Die area	46 MOS	26 MOS + 4 MTJ

Conventional full adder is redesigned for fair comparison and performance parameter like delay, power, PDP, and area is shown in table 4.4. Magnetic full adder is designed using LIM structure having delay of 86.722ps with 11.659nW power consumption. Power consumption is having power improvement of 68% compared to conventional FinFET full adder. Magnetic full adder has an improvement of 47.47% compared to conventional full adder.

Chapter – 5
Proposed 7:2 compressor

5.1 Introduction

A digital signal processor (DSP) requires fast adders and multipliers for high-speed computing systems. Adders and multipliers are on a critical path so better-performing adders and multipliers will give better performance. Multiplier generates the partial product and this partial product will be added. The compressor is used for the reduction of partial products. It is used for parallel multiplication. The multiplier can be used in digital signal processors (DSP), image compression [17], and image multiplication [17]. 5:2 compressor [17] was proposed giving an energy-efficient design.

As shown in Figure 5.1, when n-bit is multiplied with m-bit. This generates partial product. These partial products need to be added. A proposed 7:2 compressor will be used to add these partial products.

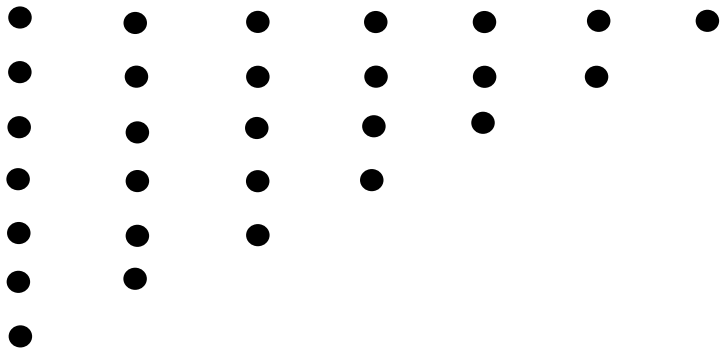


Figure 5.1 Partial product of the multiplier

The proposed 7:2 compressor is used to compress these 7 partial products and 2 outputs will be generated. In this proposed 7:2 compressor design, this thesis work uses FinFET/MTJ-based hybrid structure for the designing of the different circuits of the 7:2 compressor. Then circuit is discussed and the simulation output is shown discussing output.

Then this performance benchmarking of the given 7:2 compressor is given where the conventional 7:2 compressor circuit is redesigned for a fair comparison. Performance benchmarking of the given circuit shows that an 11.9% propagation delay has been observed with the same PDP. So, this proposed 7:2 compressor can be used in high-speed applications.

5.2 Conventional 7:2 compressor

The structure of the conventional 7:2 compressor is shown in Figure 5.2. As shown in Figure 5.2, there are 7 inputs as I1 to I7 and the final 2 outputs as Sum, Carry. C_{out1} to C_{out4} will be forwarded to the next stage for the addition of partial products. This circuit was designed using 16-nm FinFET having parameters as shown in Table 5.1. The HSpice simulator is used for the simulation of the redesigned circuit.

Table 5.1 Input parameter of 7:2 compressor

Device	Parameter	Value
FinFET	Length (L)	16nm
	VDD	0.3
	Node	16nm

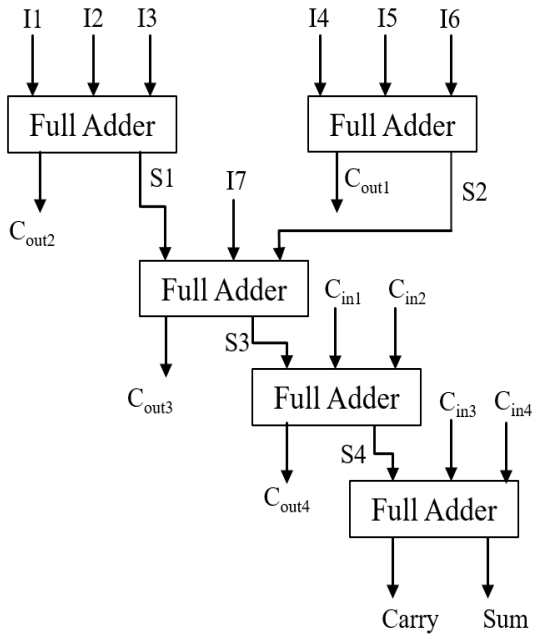


Figure 5.2 conventional 7:2 compressor [1]

5.3 Proposed 7:2 compressor circuit

As discussed in the previous section, the proposed 7:2 compressor will be designed using a logic-in-memory architecture where a hybrid FinFET/MTJ structure is used. This hybrid structure will consist of a sense amplifier [18], FinFET logic tree, and MTJ which was discussed in chapter 4 previously. From the conventional 7:2 compressor, outputs can be given by this equation [1] below.

$$S1 = I1 \wedge I2 \wedge I3 \quad (5.1)$$

$$S2 = I4 \wedge I5 \wedge I6 \quad (5.2)$$

$$S3 = I1 \wedge I2 \wedge I3 \wedge I4 \wedge I5 \wedge I6 \wedge I7 \quad (5.3)$$

$$S4 = I1 \wedge I2 \wedge I3 \wedge I4 \wedge I5 \wedge I6 \wedge I7 \wedge C_{in1} \wedge C_{in2} \quad (5.4)$$

$$Cout1 = I4*I5 + I5*I6 + I4*I6 \quad (5.5)$$

$$Cout2 = I1*I2 + I2*I3 + I1*I3 \quad (5.6)$$

$$Cout3 = S1*S2 + S1*I7 + S2*I7 \quad (5.7)$$

$$Cout4 = S3*C_{in1} + S3*C_{in2} + C_{in1}*C_{in2} \quad (5.8)$$

$$Sum = I1 \wedge I2 \wedge I3 \wedge I4 \wedge I5 \wedge I6 \wedge I7 \wedge C_{in1} \wedge C_{in2} \wedge C_{in3} \wedge C_{in4} \quad (5.9)$$

$$Carry = S4*C_{in3} + S4*C_{in4} + C_{in3}*C_{in4} \quad (5.10)$$

From these equations, it can be inferred that the output C_{out1}, C_{out2}, C_{out3}, C_{out4}, Sum, and Carry circuits will be designed. As per the circuit shown in Figure 5.2 and the equation mentioned above output can be divided into 6 circuits.

These 6 circuits will be individually designed with the hybrid structure explained in chapter 4 given in Eq. 5.5, 5.6, 5.7, 5.8, 5.9, 5.10.

5.3.1 Structure of C_{out1} circuit

The c_{out1} circuit is shown in Figure 5.3. The implementation of C_{out1} is the simplest circuit to implement. From Eq. (5.5) and the implementation of MFA given by [20], this circuit can be implemented by 2 FinFET [20] in parallel as shown in Figure 5.3.

I4 will be stored in MTJ0 and $\bar{I4}$ will be stored in MTJ1. When I4 is in AP state and $\bar{I4}$ is in P state, I5 =1, I6 =0 then, in this case, M12, and M15 will be turned on in the

evaluation phase giving I_{LB} and I_{RB} current flowing through the left arm and right arm. Here, $I_{LB} > I_{RB}$ so cout1 will be discharged faster giving $C_{out1} = 0$ and $\bar{C}_{out1} = 1$ at the output.

$$Cout1 = I4*I5 + I5*I6 + I4*I6 \quad (5.5)$$

Table 5.2 Resistance comparison and output of C_{out1}

I5	I6	Resistance comparison between left-arm and right-arm	C_{out1}
0	0	$R_L > R_R$	0
0	1	$R_L > R_R$	0
1	0	$R_L > R_R$	0
1	1	$R_L < R_R$	1

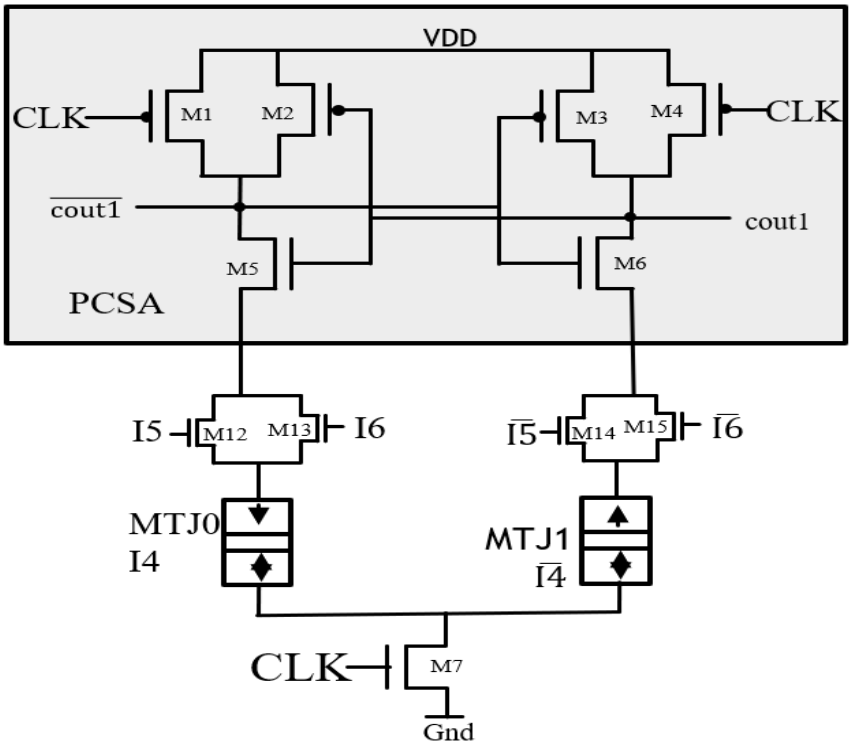


Figure 5.3 C_{out1} of proposed 7:2 compressor

5.3.2 Simulation result of C_{out1}

Simulation of C_{out1} is shown in Figure 5.4 where R(MTJ0) shows the resistance of the MTJ0 i.e., R_{AP} or R_P., and V(q) and V(qbar) will show the complementary output.

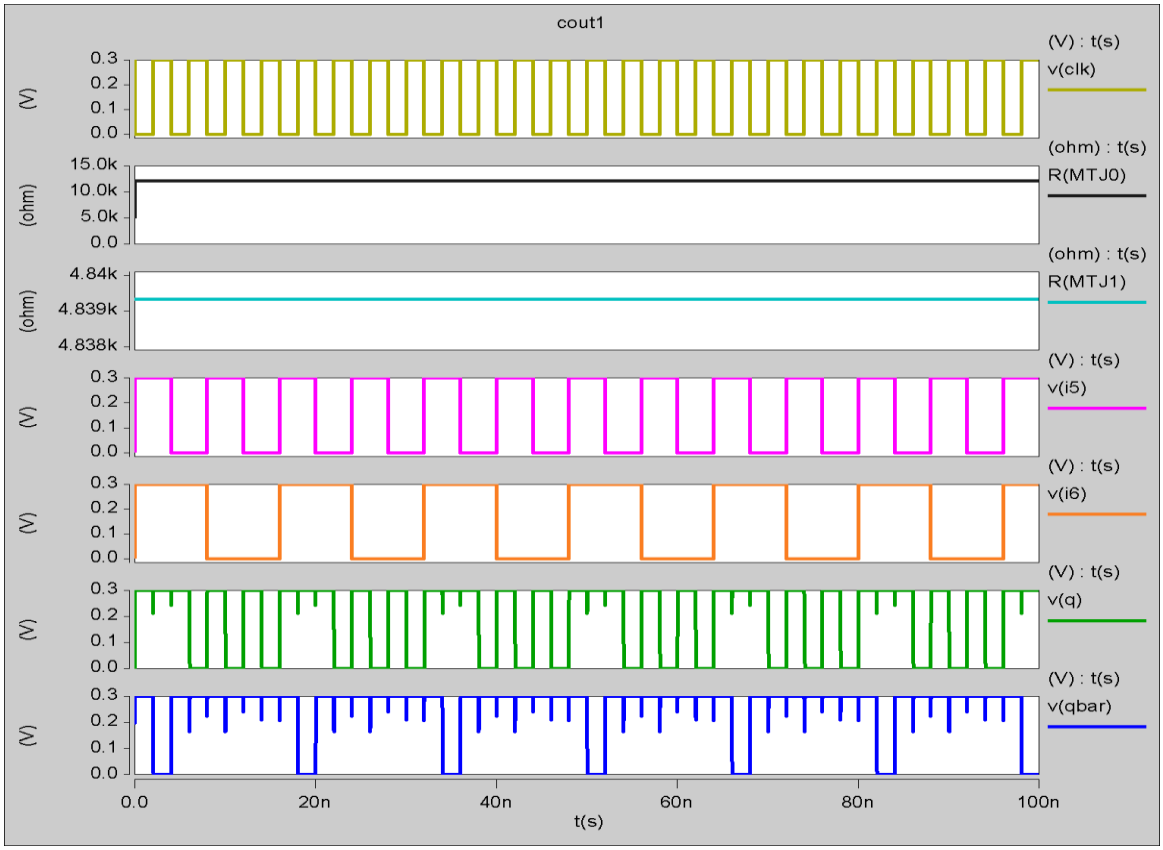


Figure 5.4 Simulation of C_{out1} of proposed 7:2 compressor

5.3.3 Structure of C_{out2} circuit

As shown in section 4.3.1, the same way the circuit of C_{out2} can be constructed. Figure 5.5 shows the structure of the C_{out2} circuit where I2 and I3 are input and I1 will be stored in MTJ0 and MTJ1. Table 5.3 shows the resistance comparison and output C_{out2}.

$$Cout2 = I1*I2 + I2*I3 + I1*I3 \quad (5.6)$$

Table 5.3 Resistance comparison and output of C_{out2}

I2	I3	Resistance comparison between left-arm and right-arm	C_{out2}
0	0	$R_L > R_R$	0
0	1	$R_L > R_R$	0
1	0	$R_L > R_R$	0
1	1	$R_L < R_R$	1

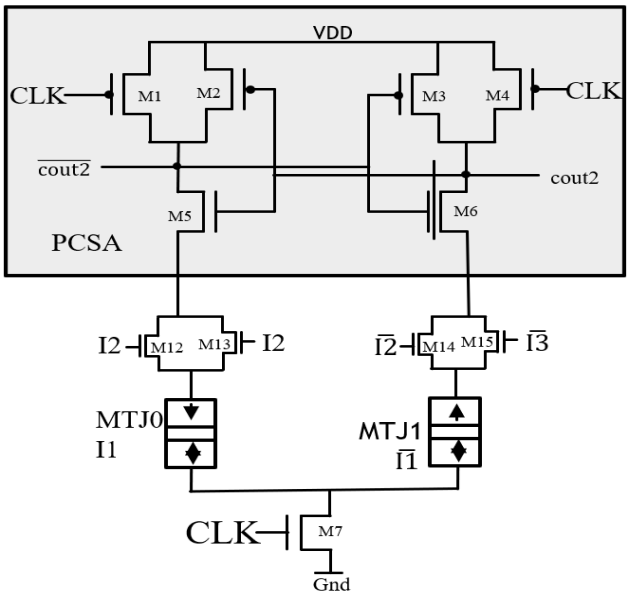


Figure 5.5 Cout2 circuit of proposed 7:2 compressor

5.3.4 Simulation result of C_{out2}

Simulation of C_{out2} is shown in Figure 5.6 where R(MTJ0) shows the resistance of the MTJ0 i.e., R_{AP} or R_P which has stored value of I1 and V(q) and V(qbar) will show the complementary output C_{out2}.

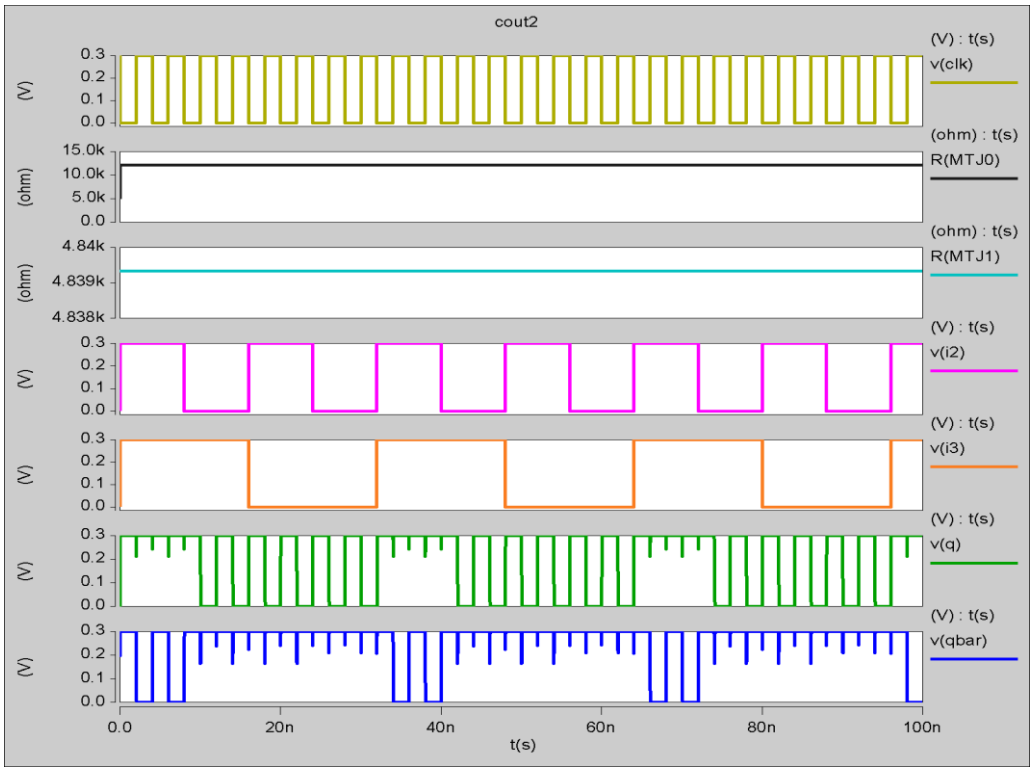


Figure 5.6 Simulation of C_{out2} of proposed 7:2 compressor

5.3.5 Structure of C_{out3}

As discussed previously, from Eq. (5.7) S1 and S2 will be in parallel. S1 and S2 can be implemented as shown in Figure 5.7. C_{out3}'s structure can be shown in Figure 5.7. When I1=1, I2=1, and I3=0 are given to the circuit where I7 is in AP state and $\overline{I7}$ is in P state, then $I_{LB} > I_{RB}$ giving output cout3 = 1, $\overline{cout3}$ = 0.

$$S1 = I1 \wedge I2 \wedge I3 \quad (5.1)$$

$$S2 = I4 \wedge I5 \wedge I6 \quad (5.2)$$

$$Cout3 = S1*S2 + S1*I7 + S2*I7 \quad (5.7)$$

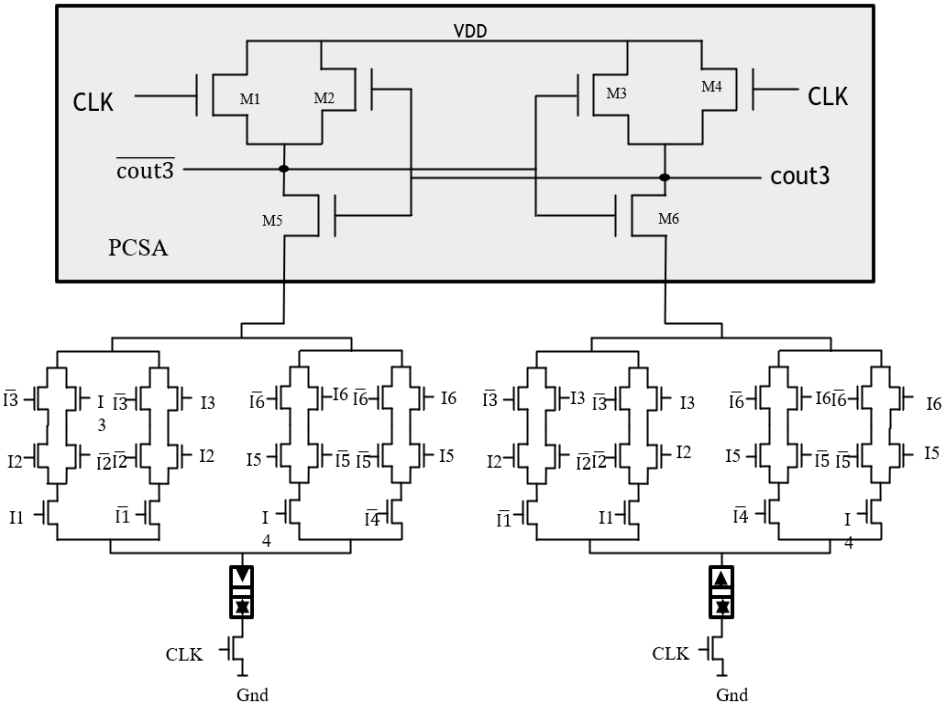


Figure 5.7 C_{out3} of the proposed 7:2 compressor

5.3.6 Simulation of the C_{out3}

Simulation of C_{out3} is shown in Figure 5.8 where R(MTJ0) shows the resistance of the MTJ0 and R(MTJ1) indicates the resistance of the MTJ1 i.e., R_{AP} or R_P which has a stored value of I1 and V(q) and V(qbar) will show the complementary output C_{out3}. V(clk) is the input indicating the pre-charge and evaluate phase. I1 to I6

shows the input of the cout3 circuit. Input I7 will be stored in the MTJ0 and MTJ1 as a resistance of the MTJ.

5.3.7 Structure of C_{out4}

From Eq. (5.8), it can be observed that when either C_{in1} or C_{in2} is 1 then output C_{out4} will be evaluated based on S3. When both C_{in1} and C_{in2} are 0 then the output will be 0. When both C_{in1} and C_{in2} are 1 then the output will be 1. This can be given as a branch where C_{in1} and C_{in2} are in series giving output 1 otherwise output will be evaluated based on S3 which is shown in Figure 5.8.

$$Cout4 = S3*C_{in1} + S3*C_{in2} + C_{in1}*C_{in2} \quad (5.8)$$

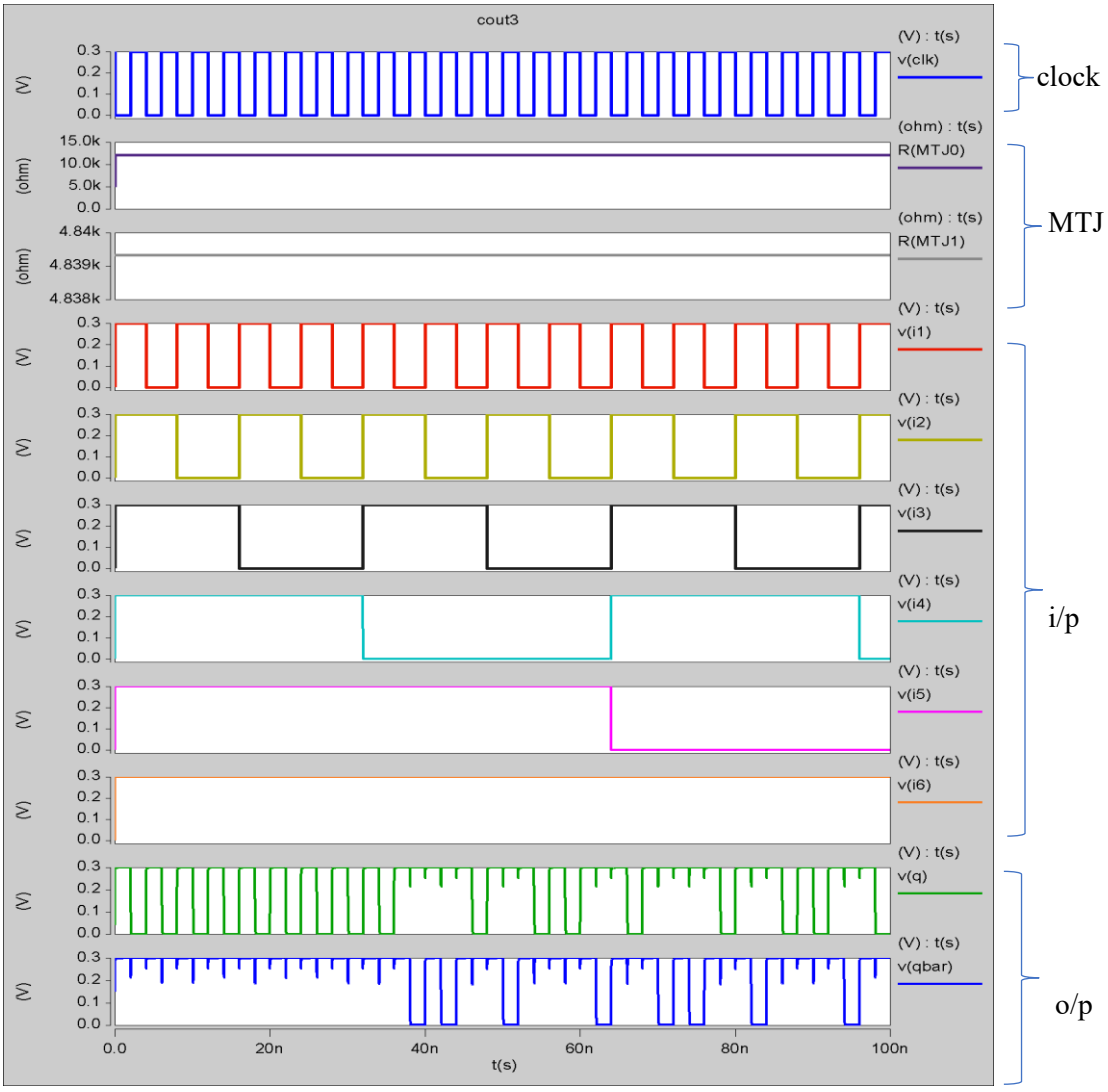


Figure 5.8 Simulation result of C_{out4} of proposed 7:2 compressor

As discussed in chapter 3 XORing of the A, B can be observed that stacking of the structure will give XORing of the input. In the same way, for S3 XORing between I1, I2, I3, I4, I5, and I6 is shown in Figure 5.8.

From Eq. (5.8), it can be observed that when either C_{in1} or C_{in2} is 1 then output C_{out4} will be evaluated based on S3. When both C_{in1} and C_{in2} are 0 then the output will be 0. When both C_{in1} and C_{in2} are 1 then the output will be 1. This can be given as a branch where C_{in1} and C_{in2} are in series giving output 1 otherwise output will be evaluated based on S3 which is shown in Figure 5.8.

$$Cout4 = S3*C_{in1} + S3*C_{in2} + C_{in1}*C_{in2} \quad (5.8)$$

As discussed in chapter 3 XORing of the A, B can be observed that stacking of the structure will give XORing of the input. In the same way, for S3 XORing between I1, I2, I3, I4, I5, and I6 is shown in Figure 5.8.

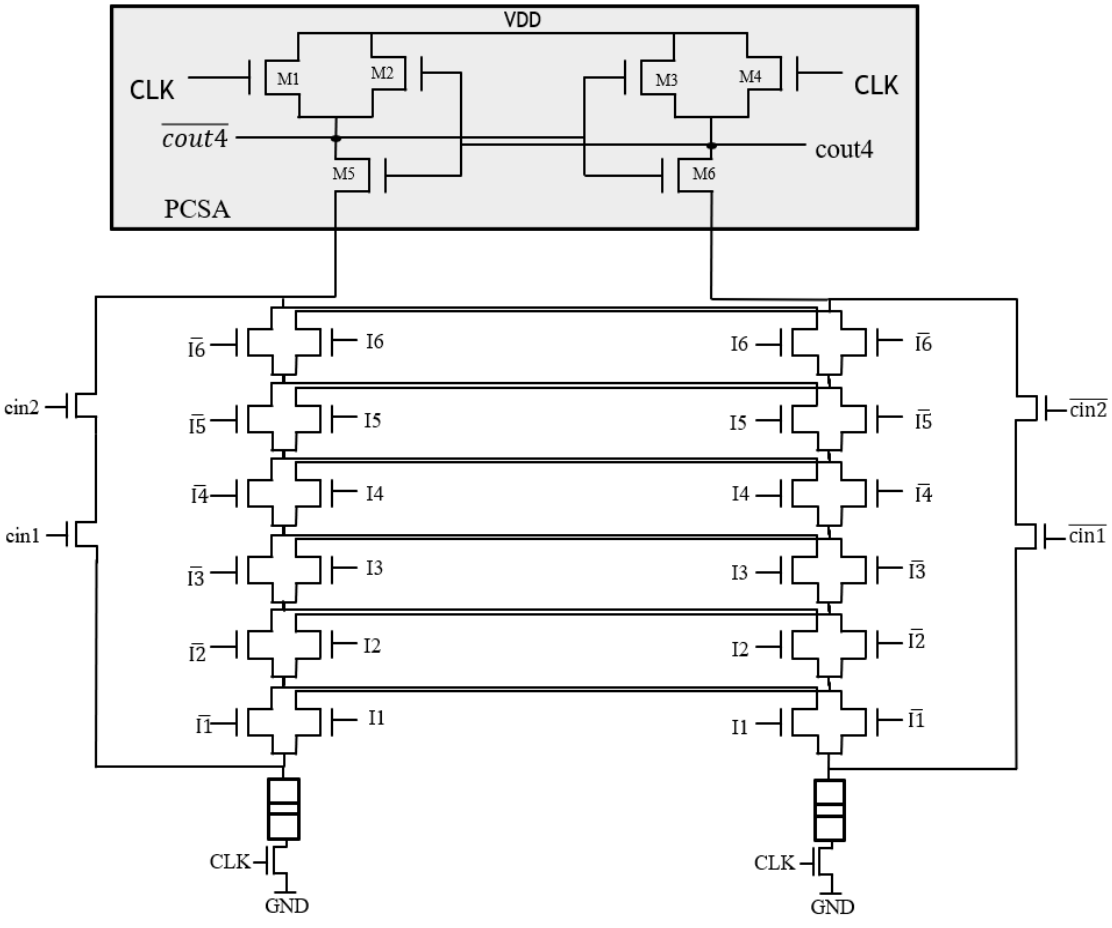


Figure 5.9 C_{out4} of the proposed 7:2 compressor

5.3.9 Simulation of the C_{out4}

Simulation of C_{out4} is shown in Figure 5.10 where R(MTJ0) shows the resistance of the MTJ0 i.e. R_{AP} or R_P which has a stored value of I7 and V(q) and V(qbar) will show the complementary output of C_{out4}. Clock, input, and output are shown in below simulation graph. Clock accounts for the pre-charge and evaluates phase and different input stimuli are applied to the circuit and output will be observed which are complementary to each other.

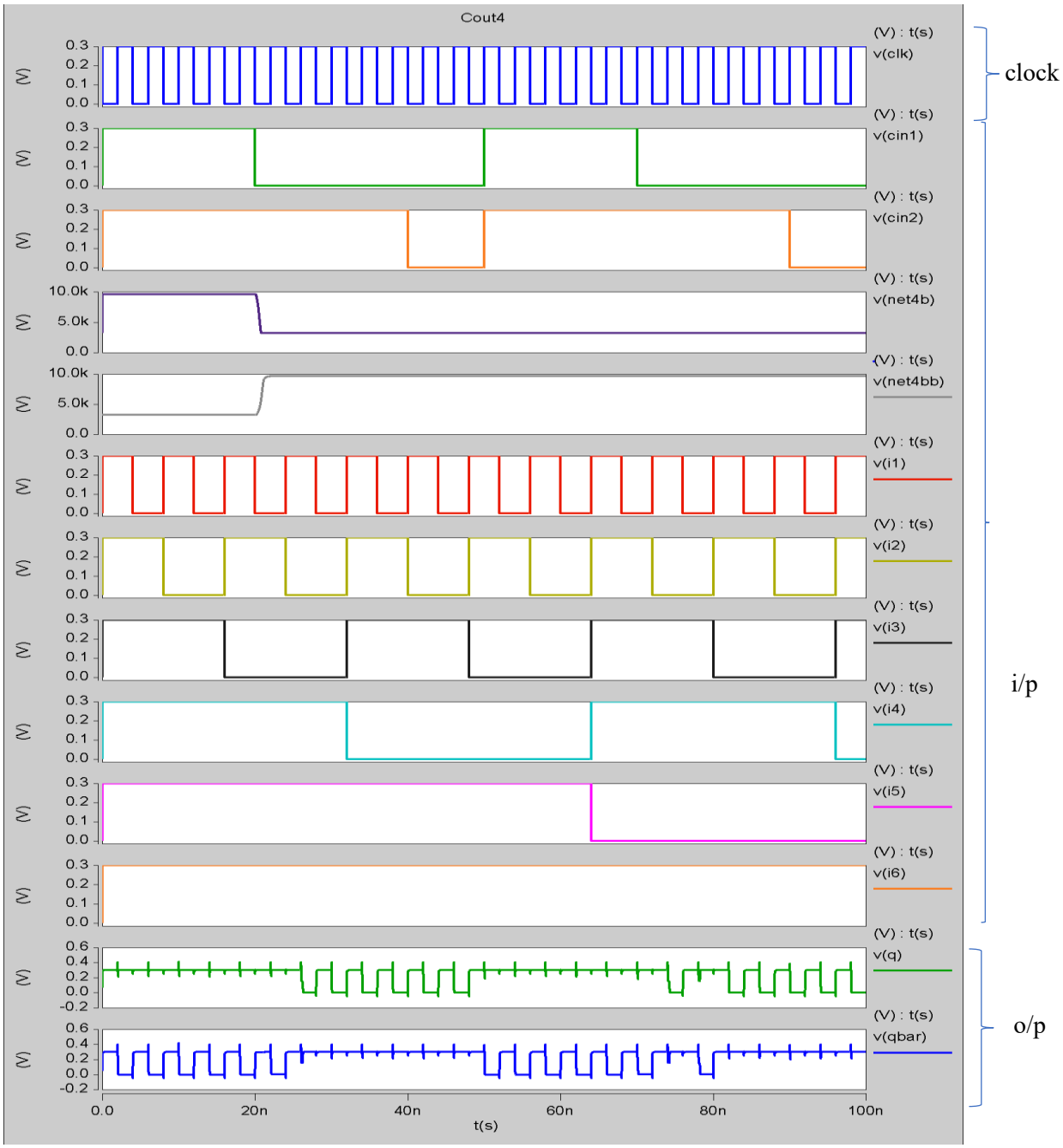


Figure 5.10 Simulation of C_{out4} of proposed 7:2 compressor

5.3.10 Structure of Sum circuit of 7:2 compressor

As discussed in chapter 3 XORing of the A, B can be observed that stacking of the structure will give XORing of the input. Same way, for Sum XORing between I1, I2, I3, I4, I5, I6, C_{in1}, C_{in2}, C_{in3}, and C_{in4} stacking of the structure as shown in Figure 5.11 can be designed. I7 will be stored in MTJ0 and MTJ1.

$$Sum = I1 \wedge I2 \wedge I3 \wedge I4 \wedge I5 \wedge I6 \wedge I7 \wedge C_{in1} \wedge C_{in2} \wedge C_{in3} \wedge C_{in4} \quad (5.9)$$

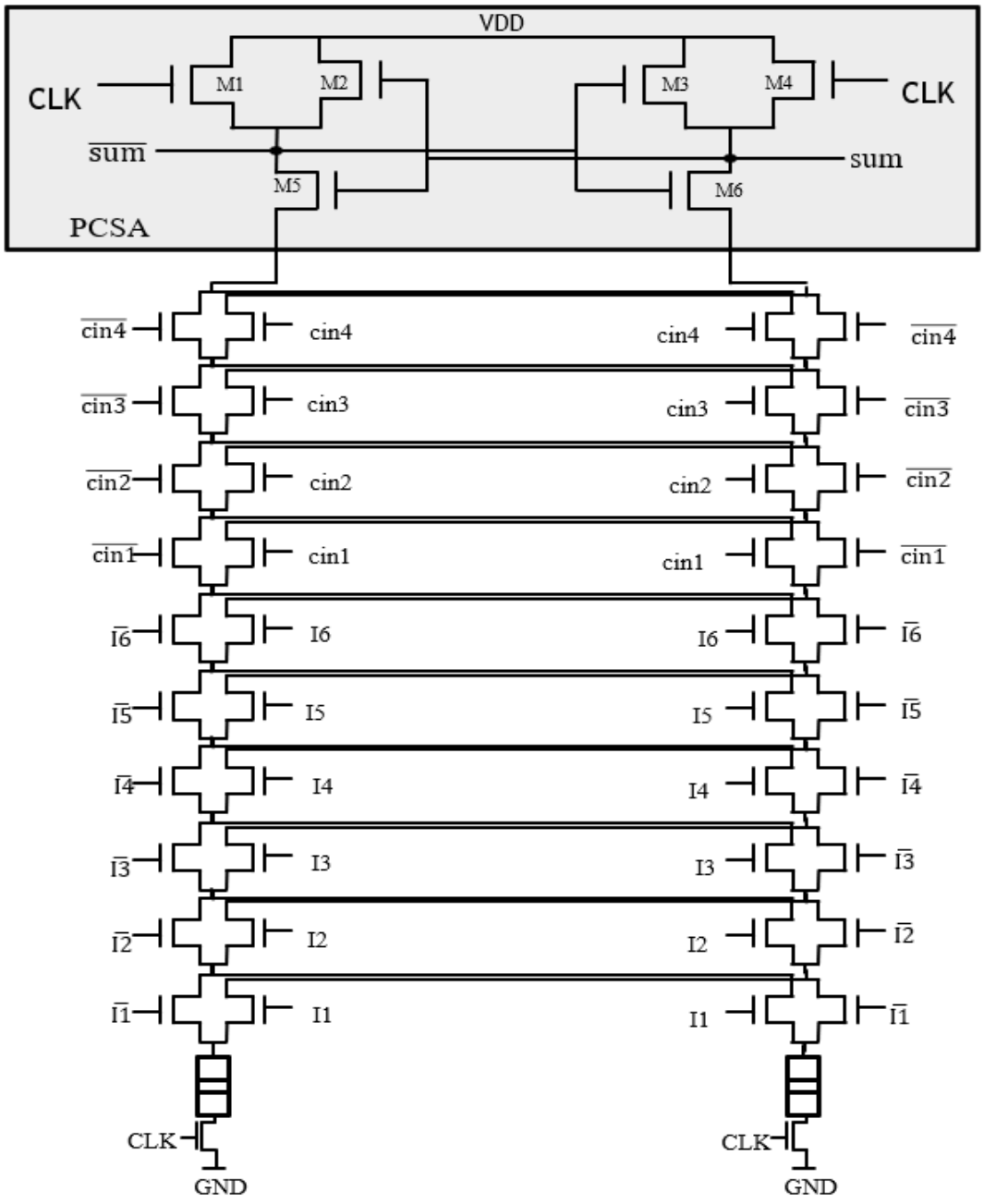


Figure 5.11 Sum of proposed 7:2 compressor

5.3.11 Simulation of the Sum

Simulation of Sum is shown in Figure 5.12 where R(MTJ0) shows the resistance of the MTJ0 i.e., R_{AP} or R_P which has a stored value of I7, and V(q) and V(qbar) will show the complementary output of sum.

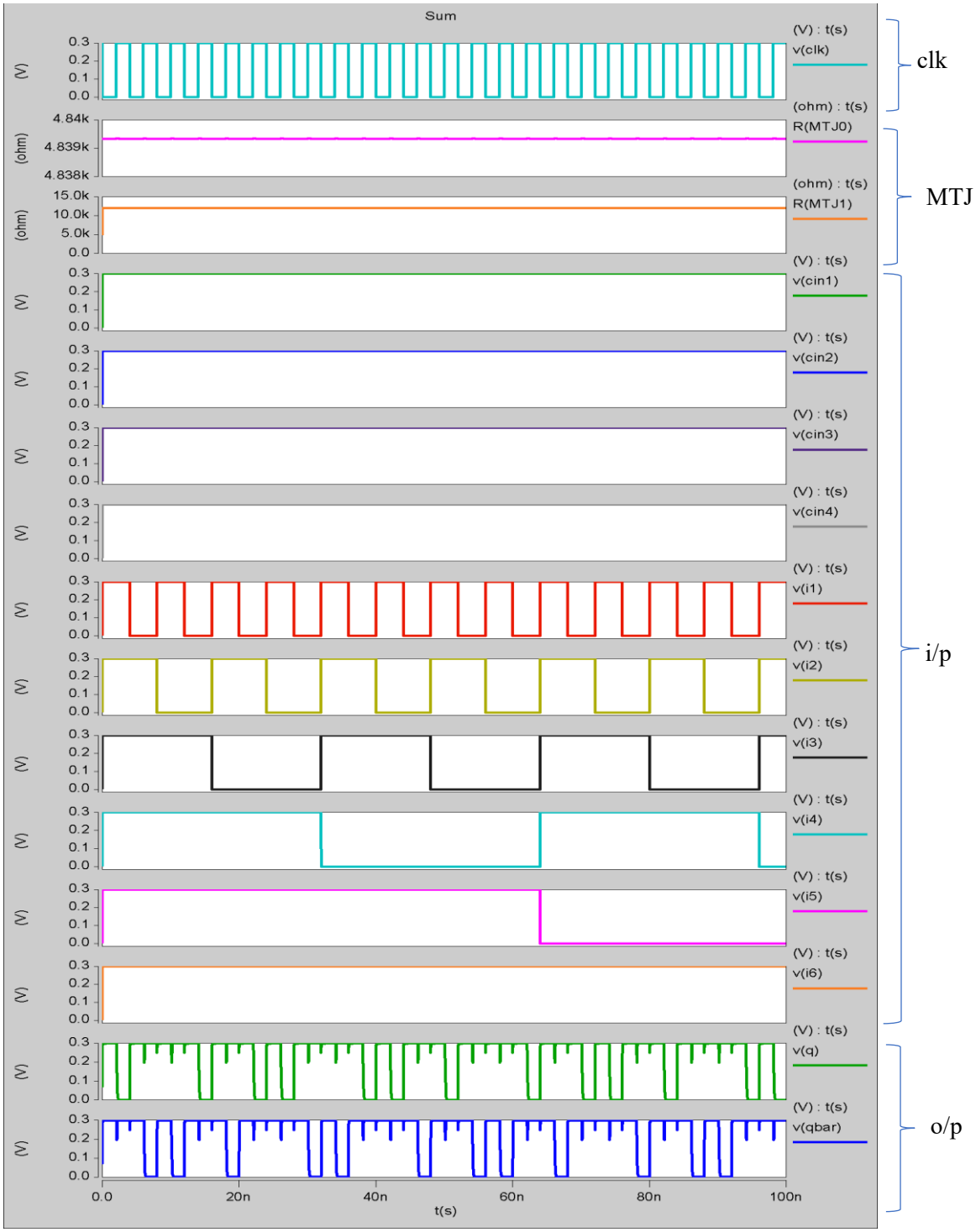


Figure 5.12 Simulation of the sum of the proposed 7:2 compressor

5.3.12 Structure of Carry of the proposed 7:2 compressor

From Eq. (5.10), when both C_{in3} and C_{in4} are 1 then the output carry will be 1 and when both are 0 then the output will be 0. When either of C_{in3} and C_{in4} is 1 then output carry will be evaluated based on S4. The structure of the carry is given in Figure 5.13.

$$S4 = I1 \wedge I2 \wedge I3 \wedge I4 \wedge I5 \wedge I6 \wedge I7 \wedge C_{in1} \wedge C_{in2} \quad (5.4)$$

$$Carry = S4*C_{in3} + S4*C_{in4} + C_{in3}*C_{in4} \quad (5.10)$$

As discussed in chapter 3 XORing of the A, B can be observed that stacking of the structure will give XORing of the input. Same way, for S4 XORing between I1, I2, I3, I4, I5, I6, C_{in1}, and C_{in2} can be designed using stacking of the structure as shown in Figure 5.13.

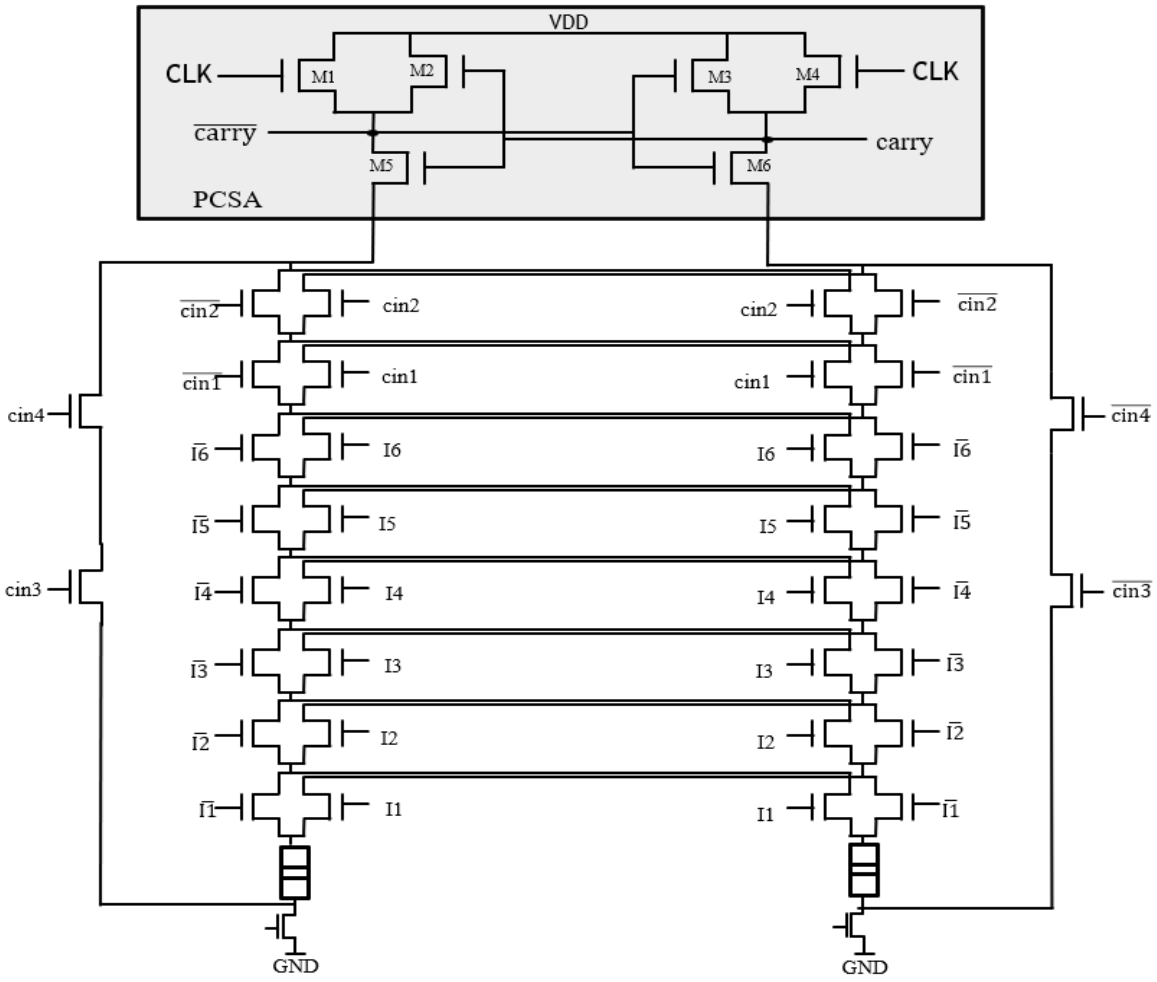


Figure 5.13 Carry of proposed 7:2 compressor

5.3.13 Simulation of the Carry

Simulation of carry is shown in Figure 5.14 where R(MTJ0) shows the resistance of the MTJ0 i.e., R_{AP} or R_P which has a stored value of I7 and V(q) and V(qbar) will show the complementary output of carry.

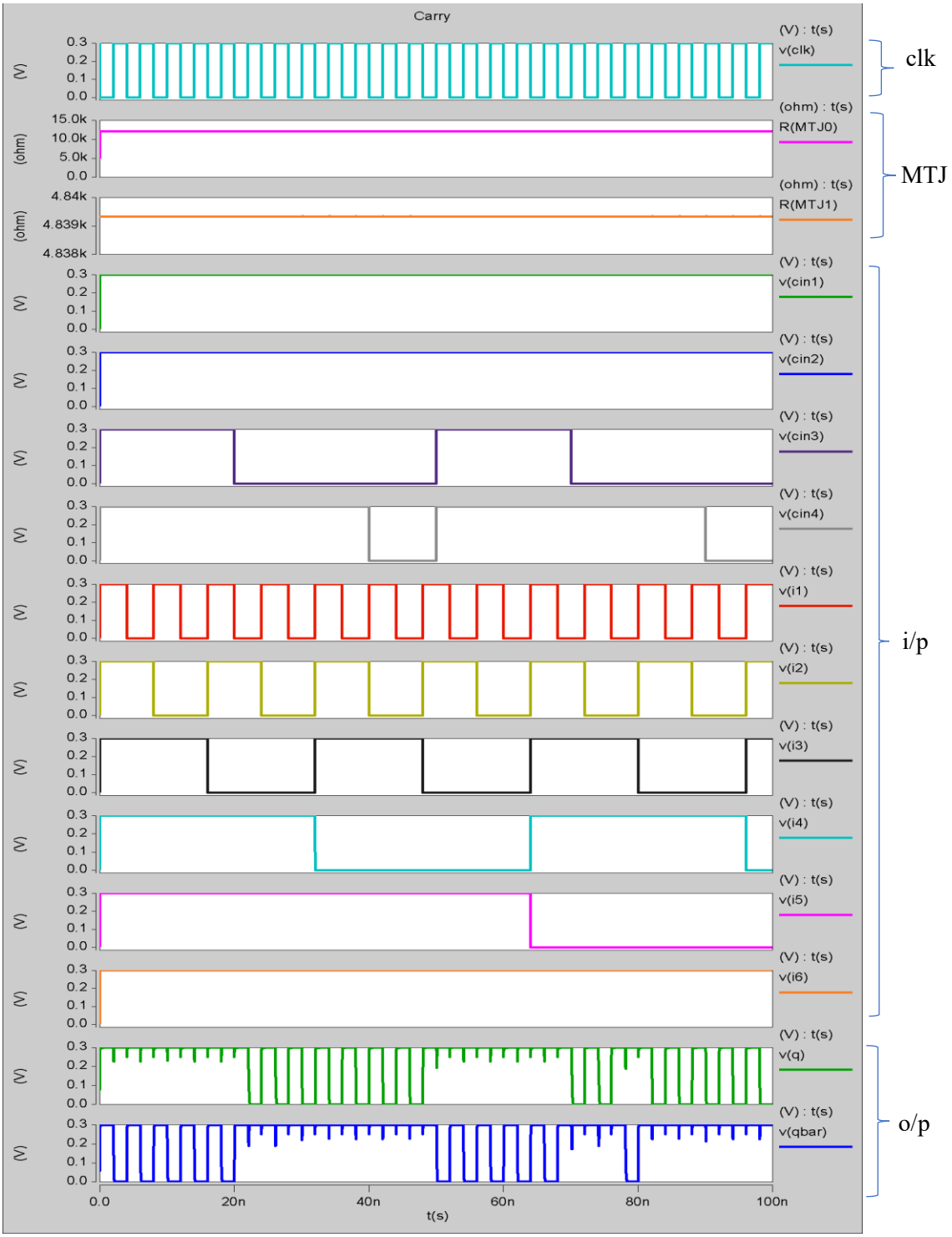


Figure 5.14 Simulation of carry of proposed 7:2 compressor

5.4 Results

In this section, the performance benchmarking of the conventional and given proposed 7:2 compressor has been discussed. For a fair comparison, the 7:2 compressor circuit proposed in [1] is redesigned using 16-nm technology. The proposed 7:2 compressor circuit is designed using 16-nm FinFET technology and a radius of 20nm for the MTJ.

Table – 5.4 Performance comparison between conventional 7:2 compressor and FinFET/MTJ-based 7:2 compressor

Parameter	**Full adder based 7:2 compressor [1]**	**FinFET/MTJ based proposed 7:2 compressor**
Delay	277.4ps	244.36ps
Power	169.07nW	217.15nW
Power delay product (PDP)	46.9 aJ	53.1 aJ

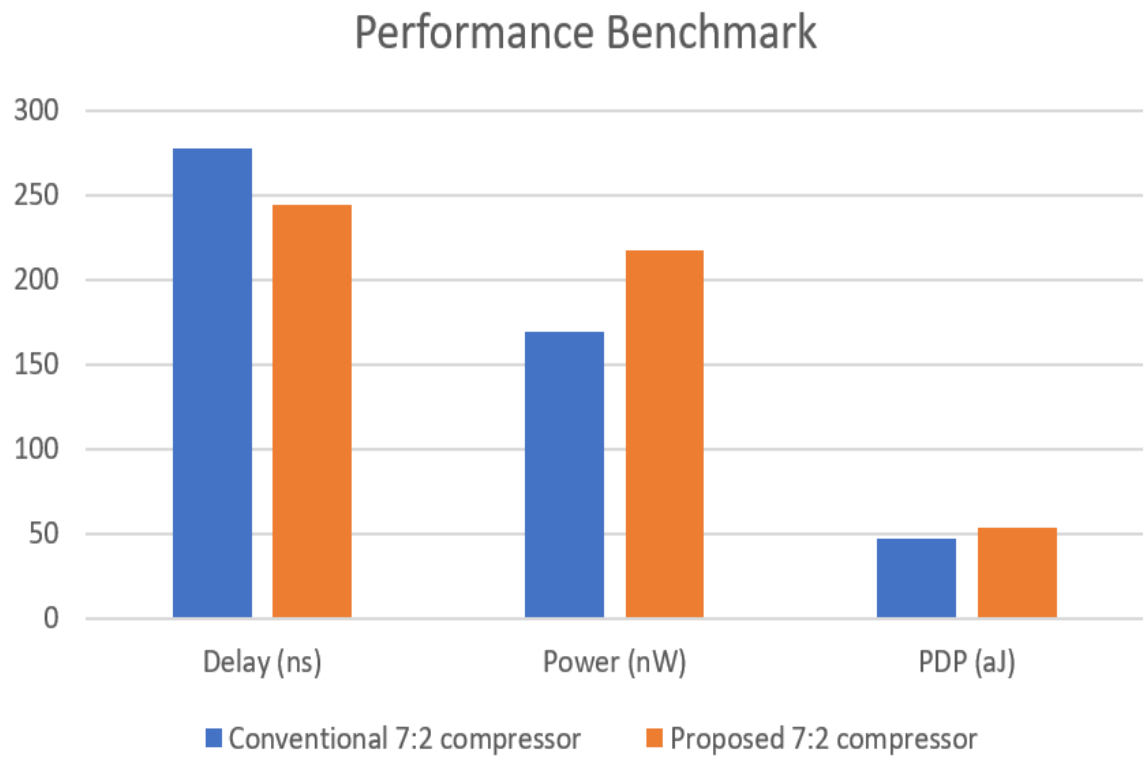


Figure 5.15 Performance benchmarking between conventional 7:2 compressor [1] and proposed 7:2 compressor

The proposed work shows a critical propagation delay of 244.36ps compared to the conventional 7:2 compressor's 277.4ps propagation having nearly the same PDP. Given circuit shows an improvement of 11.9% in the delay compared to redesigned circuit of the 7:2 compressor which was proposed in [1]. The performance benchmarking Table 5.4 is shown. Figure 5.15 shows the comparison and performance benchmarking between conventional and proposed 7:2 compressor. From Figure 5.15, It can be observed that the proposed 7:2 compressor performs better in terms of delay.

Chapter - 6
Discussions and Conclusion

6.1 Discussion of Results

Adders and multipliers are on a critical path. So, these become a bottleneck for the performance of the digital signal processor (DSP). Having less propagation delay increases the performance of the chip. So, this thesis work focuses on reducing propagation delay and focuses on developing the logic-in-memory (LIM [20]) structure. This proposed 7:2 compressor gives a performance improvement of 11.9% in propagation delay compared to a conventional 7:2 compressor. So, this proposed 7:2 compressor gives 244.36ps propagation delay which can be used in the high-speed application. This 7:2 compressor can be used in image compression, multiplier, and image multiplier.

6.2 Conclusion

In this paper, a design and analysis of a FinFET/MTJ-based 7:2 compressor are presented. Compared to the FinFET-based 7:2 compressor [1], this hybrid exact 7:2 compressor design gives performance improvement in terms of delay of around 11.92%. All of the simulations are done in Hspice with 16nm FinFET technology. So, this 7:2 compressor can be considered an alternative to the conventional 7:2 compressor design for high-speed applications. Furthermore, this work can be extended for the implementation of a 7:2 compressor using spin-hall effect MTJ. Using this spin-hall MTJ this work can give more performance improvement compared to this work.

References

[1] M. Rouholamini, O. Kavehie, A. P. Mirbaha, S. J. Jasbi, and K. Navi, "A New Design for 7:2 Compressors," 2007 IEEE/ACS International Conference on Computer Systems and Applications, 2007, pp. 474-478, DOI: 10.1109/AICCSA.2007.370924.

[2] X. Yin, X. Chen, M. Niemier, and X.S. Hu, "Ferroelectric fetsbased nonvolatile logic-in-memory circuits," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 1, pp. 159-172, Oct 2018.

[3] S. Verma, A. A. Kulkarni, and B. K. Kaushik, "Spintronics-Based Devices to Circuits: Perspectives and challenges.," in IEEE Nanotechnology Magazine, vol. 10, no. 4, pp. 13-28, Dec. 2016, DOI: 10.1109/MNANO.2016.2606683.

[4] P. Barla, V.K. Joshi, and S. Bhat, "A Novel Low Power and Reduced Transistor Count Magnetic Arithmetic Logic Unit Using Hybrid STT-MTJ/CMOS Circuit," IEEE Access, vol. 8, pp. 6876-6889, Jan 2020.

[5] S. Prajapati, and B.K. Kaushik, "Area and Energy Efficient Series Multilevel Cell STT-MRAMs for Optimized Read–Write Operations," IEEE Transactions on Magnetics, vol. 55, no. 1, pp. 1-10, Nov. 2018.

[6] L. Xue, B. Wu, B. Zhang, Y. Cheng, P. Wang, C. Park, J. Kan, S.H. Kang, and Y.Xie, "An adaptive 3T-3MTJ memory cell design for STTMRAM-based LLCs," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 3, pp. 484-495, Jan 2018.

[7] Eric Martin, "Moore's Law is Alive and Well", https://medium.com/predict/moores-law-is-alive-and-well-eaa49a450188 (accessed Oct 24, 2019).

[8] Kim, J., Paul, A., Crowell, P.A., Koester, S.J., Sapatnekar, S.S., Wang, J.-P., Kim, C.H. "Spin-based computing: device concepts, current status, and a case study on a high-performance microprocessor", Proc. IEEE **103**(1), 106–130 (2014). https://doi.org/10.1109/JPROC.2014.2361767.

[9] Verma, S., Kulkarni, A.A., Kaushik, B.K. "Spintronics-based devices to circuits: perspectives and challenges", IEEE Nanatechnol. Mag. **10**(4), 13–28 (2016). https://doi.org/10.1109/MNANO.2016.2606683

[10] Yue Zhang., “Compact modeling and hybrid circuit design for spintronic devices based on current-induced switching. Micro and nanotechnologies/Microelectronics”, University Paris Sud - Paris XI, 2014.

[11] “Magnetic Coercivity”, https://en.wikipedia.org/wiki/Coercivity, (accessed May 8, 2022).

[12] J. C. Slonczewski, “Current-driven excitation of magnetic multilayers”, Journal of Magnetism and Magnetic Materials, vol. 159, pp. L1-L7, 1996.

[13] L. Berger, “Emission of spin waves by a magnetic multilayer traversed by a current”, *Phys. Rev. B*, vol. 54, pp. 9353-9358, 1996.

[14] Z. Xu, C. Yang, M. Mao, K. B. Sutaria, C. Chakrabarti, Y. Cao, “Compact modeling of STTMTJ devices,” Solid-State Electronics, vol. 102, pp. 76-81, Dec. 2014.

[15] W. Kang et al., "Separated Precharge Sensing Amplifier for Deep Submicrometer MTJ/CMOS Hybrid Logic Circuits," in IEEE Transactions on Magnetics, vol. 50, no. 6, pp. 1-5, June 2014, Art no. 3400305, doi: 10.1109/TMAG.2013.2297393.

[16] “High performance PMOS and NMOS” , http://ptm.asu.edu/, (accessed January 6, 2012).

[17] M. Ahmadinejad, and M.H. Moaiyeri, “Energy-efficient magnetic 5: 2 compressors based on SHE-assisted hybrid FinFET/MTJ logic,” Journal of Computational Electronics, vol. 19, no. 1, pp. 1-6, Jan 2020. https://doi.org/10.1007/s10825-019-01441-0.

[18] Y. Gang, W. Zhao, J.O. Klein, C. Chappert, and P. Mazoyer, “A high reliability,low-power magnetic full adder,” IEEE Transactions on Magnetics, vol. 47, no. 11, pp. 4611-4616, May 2011.

[19] S. Sinha, G. Yeric, V. Chandra, B. Cline, and Y. Cao, "Exploring sub-20nm FinFET design with Predictive Technology Models," DAC Design Automation Conference 2012, 2012, pp. 283-288, doi: 10.1145/2228360.2228414.

[20] E. Garzón, B. Zambrano, T. Moposita, R. Taco, L. Prócel and L. Trojman, "ReconFigurable CMOS/STT-MTJ Non-Volatile Circuit for Logic-in-Memory Applications," 2020 IEEE 11th Latin American Symposium on Circuits & Systems *(LASCAS)*, 2020, doi: 10.1109/LASCAS45839.2020.9069027.

Publication

- **To be published**

R. Parmar, Y. Agrawal., K. Mummaneni, "Design and performance benchmarking of hybrid FinFET/MTJ based LIM 7:2 compressor".

Appendices

I. Code of STT

```
// Spin Transfer Torque (STT) MRAM model card

`include "disciplines.vams"
`include "constants.vams"


// Spin Transfer Torque
module hys(bl,sl,m,n);

        inout bl,sl,m,n;
        electrical bl,sl,m,n;

        parameter real r = 45e-9 from (0:inf);       // Radius of STT-MTJ
        parameter real r0= 45e-9 from (0:inf);       // Default radius of STT-MTJ
        real i1,h1;

analog
begin
        i1 = I(bl,sl);                  // Input current through STT-MTJ
        if (i1<0)
                if (V(m) >= 3.14)
                        I(m,n) <+ 0;
                else
                        I(m,n) <+ 1.29e18*(r0*r0/r/r)*i1;     // Spin Transfer Torque
regulated by transfer efficiency
        else
                if (V(m) <= 0.0001)
                        I(m,n) <+ 0;
                else
```

```
                    I(m,n) <+ 1.94e18*(r0*r0/r/r)*i1;

end
endmodule


// Damping
module ani(m,n);

    inout m,n;
    electrical m,n;

    parameter real K = 1.13e5;          // Anisotropic constant
    parameter real a = 0.027;           // Damping constanct
    parameter real gamma = 1.76e11;         // Gyromagnetic ratio
    parameter real mu0 = 1.26e-6;           // Vacuum permeability
    parameter real Ms0 = 4.56e5;            //        Default        saturation
magnetization
    parameter real H0 = 1.5e5;          // Intrinsic magnetic field
    parameter real r = 16e-9 from (0:inf);              // Radius of STT-MTJ
    real Ms;

analog
begin

    Ms = Ms0*(4*(1-1/((4*r/0.24e-9)-1))*exp(-26/3/((4*r/0.24e-9)-1))-3);    //
Saturation magnetization
    I(m,n)  <+  a*gamma*(mu0*Ms*H0*sin(V(m))+2*K*cos(V(m))*sin(V(m)));
    // Damping torque

end
endmodule
```

```
// Capacitance used in simulation
module caps1(p,n);

        inout p,n;
        electrical p,n;

        parameter real Ms0 = 4.81e5 ; // Default saturation
magnetization
        parameter real r = 45e-9 from (0:inf); // Radius of STT-MTJ
        parameter real r0= 45e-9; // Default radius of STT-MTJ
        real Ms;

analog
begin

        Ms =  Ms0*(4*(1-1/(4*r/0.24e-9-1))*exp(-26/3/(4*r/0.24e-9-1))-3); //
Saturation magnetization
        I(p,n) <+ Ms*ddt(V(p,n));

end
endmodule


// The component to indicate the resistance of STT-MTJ
module vcr0(c, o5, o6);

        input c;
        output o5, o6;
        electrical c,o5,o6;

        parameter real r=20e-9;//45e-9 from (0:inf); // Radius of
STT-MTJ
        parameter real tox=0.85e-9; // Oxide thickness of STT-MTJ
        parameter real F =221; //332;//221;
```

```
        parameter real phi = 0.4;
        parameter real A = 3.1416*r*r;              // Area of STT-MTJ
        parameter real TMR0 = 1.5;//1.2;            // Default tunnel magneto-
resistance
        real Rp, R;

analog
begin

        Rp = tox/(F*sqrt(phi)*A)*exp(1.025*tox*sqrt(phi));      // STT-MTJ
resistance in parallel state
        R = 2*Rp*(TMR0+1)/(TMR0+2+TMR0*cos(V(c)));              // STT-MTJ
resistance in transition
        V(o5,o6) <+ R;

end
endmodule


// Model STT-MTJ as a resistor
module vcr(c, o5, o6);

        input c;
        output o5, o6;
        electrical c,o5,o6;

        parameter real r=20e-9;//45e-9;
        parameter real tox=0.85e-9;
        parameter real F = 221;//332;//211;
        parameter real phi = 0.4;
        parameter real A = 3.1416*r*r;
        parameter real TMR0 = 1.5;// 2;
        real Rp, R;
```

```
analog
begin

        Rp = tox/(F*sqrt(phi)*A)*exp(1.025*tox*sqrt(phi));
        R = 2*Rp*(TMR0+1)/(TMR0+2+TMR0*cos(V(c)));
        V(o5,o6) <+ I(o5,o6)*R;

end
endmodule
```

II. Code of C_{out1} of proposed 7:2 compressor

```
*cout1 of 7:2 compressor

//////////file inclusion/////////////
.hdl 'stt.txt'
.include '16nm_nfet.txt'
.include '16nm_pfet.txt'
$.include '16nm_bulk.txt'
//////////end of file inclusion////////////////


//////////parameter initializatoin/////////
.param r1=20n
.param tox1= 0.85n
.param inip=0
.param iniap=3.14
.param width= 16n
.param length=16n
//////////end of parameter initialization/////////

$mn1 drain gate source body nmos w = width l = length        $ transistor

//////////////////pre charge sense amplifier///////////////
```

```
mpfet1 vdd clk qbar vdd pfet width=width l=length
mpfet2 vdd q qbar vdd pfet width=width l=length
mpfet3 vdd clk q vdd pfet width=width l=length
mpfet4 vdd qbar q vdd pfet width=width l=length
mnfet1 qbar q nx1 0 nfet width=width l=length
mnfet2 q qbar nx1bar 0 nfet width=width l=length
//////////////////pre charge sense amplifier///////////////

//////////////////logic block//////////////////////

mnl5 nx1 i5 net2b 0 nfet width=width l=length
mnl6 nx1 i6 net2b 0 nfet width=width l=length
mnl5bar nx1bar i5bar net2bb 0 nfet width=width l=length
mnl6bar nx1bar i6bar net2bb 0 nfet width=width l=length



//////////////////////end of logic block///////////////



//////////////////clock ///////////////

mnclkl blb clk 0 0 nfet w=width l=length
mnclkr blbb clk 0 0 nfet w=width l=length

//////////////////end of clock/////////////////

///////////////MTJ1/////////////////

xM1b net1b net2b net3b 0 hys r=r1
xM2b net3b 0 ani r=r1
```

```
xC1b net3b 0 caps1 r=r1
xRoutb net3b net4b 0 vcr0

xRwriteb net3b blb net1b vcr r=r1 tox=tox1
//////////////end of MTJ1/////////////

////////////MTJ2/////////////////

xM1bb net1bb net2bb net3bb 0 hys r=r1
xM2bb net3bb 0 ani r=r1
xC1bb net3bb 0 caps1 r=r1
xRoutbb net3bb net4bb 0 vcr0

xRwritebb net3bb blbb net1bb vcr r=r1 tox=tox1
///////////////end of MTJ2////////////

//////////initialization of MTJ///////////
//i4
.ic V(net3b) = iniap
.ic V(net3bb) = inip

/////////end of initialization of MTJ/////

////////////supply/////////
.param vddpap=1
.param vddapp=-1
.param vddc = 1
.param vddv = 0.3

Vvdd vdd 0 vddv

Vclk clk 0 pulse(vddv 0 0 0 0 2n 4n)

Vi5 i5 0 pulse(0 vddv 0 0 0 4n 8n)
```

```
Vi5bar i5bar 0 pulse(vddv 0 0 0 0 4n 8n)
Vi6 i6 0 pulse(0 vddv 0 0 0 8n 16n)
Vi6bar i6bar 0 pulse(vddv 0 0 0 0 8n 16n)




////////////end of supply////////////

//////////////transient response////////////
.measure TRAN iavg AVG i(vvdd) FROM=0 TO=100e-9

$the result will be stored on the .lis file, someting like this:
$iavg=10e-6

$If you want you can also do the power calculation with hspice:
.measure TRAN power PARAM='iavg*vddv'



.probe tran I(vvqblb)
.plot tran power
.probe tran power
.print power
.tran 1p 100n
.option post=2
.option gmin=1e-9
.option gmindc=1e-9
.end
////////////end of transient response//////////
```

III. Code of C_{out2} of proposed 7:2 compressor

```
*cout2 of 6:2 compressor

///////////file inclusion/////////////
```

```
.hdl 'stt.txt'
.include '16nm_nfet.txt'
.include '16nm_pfet.txt'
$.include '16nm_bulk.txt'
//////////end of file inclusion////////////////


//////////parameter initializatoin//////////
.param r1=20n
.param tox1= 0.85n
.param inip=0
.param iniap=3.14
.param width= 16n
.param length=16n
//////////end of parameter initialization/////////

$mn1 drain gate source body nmos w = width l = length        $ transistor

//////////////////pre charge sense amplifier///////////////

mpfet1 vdd clk qbar vdd pfet width=width l=length
mpfet2 vdd q qbar vdd pfet width=width l=length
mpfet3 vdd clk q vdd pfet width=width l=length
mpfet4 vdd qbar q vdd pfet width=width l=length
mnfet1 qbar q nx1 0 nfet width=width l=length
mnfet2 q qbar nx1bar 0 nfet width=width l=length
//////////////////pre charge sense amplifier///////////////

//////////////////logic block////////////////////////

mnl2 nx1 i2 net2b 0 nfet w=width l=length
mnl3 nx1 i3 net2b 0 nfet w=width l=length
mnl2bar nx1bar i2bar net2bb 0 nfet w=width l=length
mnl3bar nx1bar i3bar net2bb 0 nfet w=width l=length
```

```
//////////////////////end of logic block///////////////


//////////////////clock ////////////////

mnclkl blb clk 0 0 nfet w=width l=length
mnclkr blbb clk 0 0 nfet w=width l=length

////////////////end of clock////////////////

///////////////MTJ1/////////////////

xM1b net1b net2b net3b 0 hys r=r1
xM2b net3b 0 ani r=r1
xC1b net3b 0 caps1 r=r1
xRoutb net3b net4b 0 vcr0

xRwriteb net3b blb net1b vcr r=r1 tox=tox1
//////////////end of MTJ1/////////////

////////////MTJ2//////////////////

xM1bb net1bb net2bb net3bb 0 hys r=r1
xM2bb net3bb 0 ani r=r1
xC1bb net3bb 0 caps1 r=r1
xRoutbb net3bb net4bb 0 vcr0

xRwritebb net3bb blbb net1bb vcr r=r1 tox=tox1
///////////////end of MTJ2////////////

/////////initialization of MTJ///////////
//i1
.ic V(net3b) = iniap
```

```
.ic V(net3bb) = inip

/////////end of initialization of MTJ/////

//////////////supply//////////
.param vddpap=1
.param vddapp=-1
.param vddv = 0.3

Vvdd vdd 0 vddv
Vclk clk 0 pulse(vddv 0 0 0 0 2n 4n)

Vi2 i2 0 pulse(0 vddv 0 0 0 8n 16n)
Vi2bar i2bar 0 pulse(vddv 0 0 0 0 8n 16n )
Vi3 i3 0 pulse(0 vddv 0 0 0 16n 32n)
Vi3bar i3bar 0 pulse(vddv 0 0 0 0 16n 32n )

//////////////end of supply//////////////

///////////////transient response//////////////
.measure TRAN iavg AVG i(vvdd) FROM=0 TO=100e-9

$the result will be stored on the .lis file, someting like this:
$iavg=10e-6

$If you want you can also do the power calculation with hspice:
.measure TRAN power PARAM='iavg*vddv'



.probe tran I(vvqblb)
.plot tran power
.probe tran power
.print power
.tran 1p 100n
```

```
.option post=2
.option gmin=1e-9
.option gmindc=1e-9
.end
////////////end of transient response//////////
```

IV. Code of C_{out3} of proposed 7:2 compressor

```
*cout3 of 7:2 compressor

//////////file inclusion/////////////
.hdl 'stt.txt'
.include '16nm_nfet.txt'
.include '16nm_pfet.txt'
$.include '16nm_bulk.txt'
//////////end of file inclusion////////////////




//////////parameter initializatoin/////////
.param r1=20n
.param tox1= 0.85n
.param inip=0
.param iniap=3.14
.param width= 16n
.param length=16n
/////////end of parameter initialization/////////

$mn1 drain gate source body nmos w = width l = length        $ transistor

//////////////////pre charge sense amplifier///////////////

mpfet1 vdd clk qbar vdd pfet width=width l=length
mpfet2 vdd q qbar vdd pfet width=width l=length
```

```
mpfet3 vdd clk q vdd pfet width=width l=length
mpfet4 vdd qbar q vdd pfet width=width l=length
mnfet1 qbar q nx1 0 nfet width=width l=length
mnfet2 q qbar nx1bar 0 nfet width=width l=length
//////////////////pre charge sense amplifier///////////////

//////////////////logic block//////////////////////////

mnl1 nd1l i1 net2b 0 nfet width=width l=length
mnl2 nd2l i2 nd1l 0 nfet width=width l=length
mnl3bar nx1 i3bar nd2l  0 nfet width=width l=length
mnl2bar nd2lbar i2bar nd1l 0 nfet width=width l=length
mnl3 nx1 i3 nd2lbar 0 nfet width=width l=length
mnl1bar nd1bar i1bar net2b 0 nfet width=width l=length
mnll2bar nd2llbar i2bar nd1bar 0 nfet width=width l=length
mnll3bar nx1 i3bar nd2llbar 0 nfet width=width l=length
mnll2 nd2lbar i2 nd1bar 0 nfet width=width l=length
mnll3 nx1 i3 nd2ll 0 nfet width=width l=length


mnl4 nd4l i4 net2b  0 nfet width=width l=length
mnl5 nd5l i5 nd4l  0 nfet width=width l=length
mnl6bar nx1 i6bar nd5l  0 nfet width=width l=length
mnl5bar nd5lbar i5bar nd4l 0 nfet width=width l=length
mnl6 nx1 i6 nd5lbar 0 nfet width=width l=length
mnl4bar net2b i4bar nd4lbar 0 nfet width=width l=length
mnll5bar nd5llbar i5bar nd4lbar 0 nfet width=width l=length
mnll6bar nx1 i6bar nd5llbar 0 nfet width=width l=length
mnll5 nd5ll i5 nd4lbar 0 nfet width=width l=length
mnll6 nx1 i6 nd5ll 0 nfet width=width l=length

mnr1bar nr1bar i1 net2bb  0 nfet width=width l=length //c
mnr2bar nr2bar i2bar nr1bar 0 nfet width=width l=length
mnr3bar nx1bar i3bar nr2bar 0 nfet width=width l=length
```

```
mnr2 nr2 i2 nr1bar 0 nfet width=width l=length
mnr3 nx1bar i3 nr2 0 nfet width=width l=length
mnr1 nr1 i1bar net2bb 0 nfet width=width l=length //c
mnrr2bar nrr2bar i2bar nr1  0 nfet width=width l=length
mnrr3 nx1bar i3 nrr2bar 0 nfet width=width l=length
mnrr2 nrr2 i2 nr1 0 nfet width=width l=length
mnrr3bar nx1bar i3bar nrr2 0 nfet width=width l=length

mnrr4bar nr4bar i4 net2bb 0 nfet width=width l=length //c
mnr5 nr5 i5 nr4bar 0 nfet width=width l=length
mnr6 nx1bar i6 nr5 0 nfet width=width l=length
mnr5bar nr5bar i5bar nr4bar 0 nfet width=width l=length
mnr6bar nx1bar i6bar nr5bar 0 nfet width=width l=length
mnrr4 nrr4 i4bar net2bb 0 nfet width=width l=length //c
mnrr5 nrr5 i5 nrr4 0 nfet width=width l=length
mnrr6bar nx1bar i6bar nrr5 0 nfet width=width l=length
mnrr5bar nrr5bar i5bar nrr4 0 nfet width=width l=length
mnrr6 nx1bar i6 nrr5bar 0 nfet width=width l=length




//////////////////////end of logic block///////////////


//////////////////clock ////////////////

mnclkl blb clk 0 0 nfet w=width l=length
mnclkr blbb clk 0 0 nfet w=width l=length

////////////////end of clock/////////////////

///////////////MTJ1//////////////////
```

```
xM1b net1b net2b net3b 0 hys r=r1
xM2b net3b 0 ani r=r1
xC1b net3b 0 caps1 r=r1
xRoutb net3b net4b 0 vcr0

xRwriteb net3b blb net1b vcr r=r1 tox=tox1
//////////////end of MTJ1/////////////

////////////MTJ2/////////////////

xM1bb net1bb net2bb net3bb 0 hys r=r1
xM2bb net3bb 0 ani r=r1
xC1bb net3bb 0 caps1 r=r1
xRoutbb net3bb net4bb 0 vcr0

xRwritebb net3bb blbb net1bb vcr r=r1 tox=tox1
//////////////end of MTJ2////////////

//////////initialization of MTJ///////////
///x7
.ic V(net3b) = iniap
.ic V(net3bb) = inip

/////////end of initialization of MTJ/////

//////////////supply//////////
.param vddpap=-1
.param vddapp=1
.param vddv = 0.3

Vvdd vdd 0 vddv
Vclk clk 0 pulse(vddv 0 0 0 0 2n 4n)
Vi1 i1 0 pulse(0 vddv 0 0 0 4n 8n)
Vi1bar i1bar 0 pulse(vddv 0 0 0 0 4n 8n )
```

```
Vi2 i2 0 pulse(0 vddv 0 0 0 8n 16n)
Vi2bar i2bar 0 pulse(vddv 0 0 0 0 8n 16n )
Vi3 i3 0 pulse(0 vddv 0 0 0 16n 32n)
Vi3bar i3bar 0 pulse(vddv 0 0 0 0 16n 32n )
Vi4 i4 0 pulse(0 vddv 0 0 0 32n 64n)
Vi4bar i4bar 0 pulse(vddv 0 0 0 0 32n 64n )
Vi5 i5 0 pulse(0 vddv 0 0 0 64n 128n)
Vi5bar i5bar 0 pulse(vddv 0 0 0 0 64n 128n)
Vi6 i6 0 pulse(0 vddv 0 0 0 128n 256n)
Vi6bar i6bar 0 pulse(vddv 0 0 0 0 128n 256n)

$Vi7 net2b blb pwl(0ns 0 19ns 0 19.1ns vddpap 20.1ns vddpap 20.2ns 0 23ns 0)
$Vi7bar net2bb blbb pwl(0ns 0 19ns 0 19.1ns vddapp 20.1ns vddapp 20.2ns 0 23ns
0)
$Vi7 net2b blb pulse(vddpap 0 0 0 0 20n 256n)
$Vi7bar net2bb blbb pulse(vddapp 0 0 0 0 20n 256n)
////////////end of supply////////////

//////////////transient response////////////
.measure TRAN iavg AVG i(vvdd) FROM=0 TO=500e-9

$the result will be stored on the .lis file, someting like this:
$iavg=10e-6

$If you want you can also do the power calculation with hspice:
.measure TRAN power PARAM='iavg*vddv'



.probe tran I(vvqblb)
.plot tran power
.probe tran power
.print power
.tran 1p 100n
.option post=2
```

```
.option gmin=1e-9
.option gmindc=1e-9
.end
////////////end of transient response//////////
```

V. Code of C_{out4} of proposed 7:2 compressor

```
*cout4 of 7:2 compressor

//////////file inclusion/////////////
.hdl 'stt.txt'
.include '16nm_nfet.txt'
.include '16nm_pfet.txt'
$.include '16nm_bulk.txt'
//////////end of file inclusion////////////////


//////////parameter initializatoin/////////
.param r1=20n
.param tox1= 0.85n
.param inip=0
.param iniap=3.14
.param width= 16n
.param length=16n
//////////end of parameter initialization/////////

$mn1 drain gate source body nmos w = width l = length        $ transistor

//////////////////pre charge sense amplifier///////////////

mpfet1 vdd clk qbar vdd pfet width=width l=length
mpfet2 vdd q qbar vdd pfet width=width l=length
mpfet3 vdd clk q vdd pfet width=width l=length
mpfet4 vdd qbar q vdd pfet width=width l=length
```

```
mnfet1 qbar q nx6 0 nfet width=width l=length
mnfet2 q qbar nx6bar 0 nfet width=width l=length
//////////////////pre charge sense amplifier///////////////

//////////////////logic block//////////////////////////

mnl1bar nx1 i1bar net2b 0 nfet width=width l=length
mnl1 nx1bar i1 net2b 0 nfet width=width l=length
mnll1 nx1 i1 net2bb 0 nfet width=width l=length
mnll1bar nx1bar i1bar net2bb 0 nfet width=width l=length

mnl2bar nx2 i2bar nx1 0 nfet width=width l=length
mnl2 nx2bar i2 nx1 0 nfet width=width l=length
mnll2 nx2 i2 nx1bar 0 nfet width=width l=length
mnll2bar nx2bar i2bar nx1bar 0 nfet width=width l=length

mnl3bar nx3 i3bar nx2 0 nfet width=width l=length
mnl3 nx3bar i3 nx2 0 nfet width=width l=length
mnll3 nx3 i3 nx2bar 0 nfet width=width l=length
mnll3bar nx3bar i3bar nx2bar 0 nfet width=width l=length

mnl4bar nx4 i4bar nx3 0 nfet width=width l=length
mnl4 nx4bar i4 nx3 0 nfet width=width l=length
mnll4 nx4 i4 nx3bar 0 nfet width=width l=length
mnll4bar nx4bar i4bar nx3bar 0 nfet width=width l=length

mnl5bar nx5 i5bar nx4 0 nfet width=width l=length
mnl5 nx5bar i5 nx4 0 nfet width=width l=length
mnll5 nx5 i5 nx4bar 0 nfet width=width l=length
mnll5bar nx5bar i5bar nx4bar 0 nfet width=width l=length

mnl6bar nx6 i6bar nx5 0 nfet width=width l=length
mnl6 nx6bar i6 nx5 0 nfet width=width l=length
mnll6 nx6 i6 nx5bar 0 nfet width=width l=length
```

mnll6bar nx6bar i6bar nx5bar 0 nfet width=width l=length

mnlcin2 nx6 cin2 ndcin1 0 nfet width=width l=length
mnlcin1 ndcin1 cin1 net2b 0 nfet width=width l=length
mnllcin2bar nx6bar cin2bar ndcin1bar 0 nfet width=width l=length
mnllcin1bar ndcin1bar cin1bar net2bb 0 nfet width=width l=length

//////////////////////end of logic block///////////////

//////////////////clock ///////////////

mnclkl blb clk 0 0 nfet w=width l=length
mnclkr blbb clk 0 0 nfet w=width l=length

////////////////end of clock////////////////

///////////////MTJ1/////////////////

xM1b net1b net2b net3b 0 hys r=r1
xM2b net3b 0 ani r=r1
xC1b net3b 0 caps1 r=r1
xRoutb net3b net4b 0 vcr0

xRwriteb net3b blb net1b vcr r=r1 tox=tox1
//////////////end of MTJ1/////////////

////////////MTJ2/////////////////

xM1bb net1bb net2bb net3bb 0 hys r=r1
xM2bb net3bb 0 ani r=r1

```
xC1bb net3bb 0 caps1 r=r1
xRoutbb net3bb net4bb 0 vcr0

xRwritebb net3bb blbb net1bb vcr r=r1 tox=tox1
///////////////end of MTJ2////////////

//////////initialization of MTJ///////////
//i7
.ic V(net3b) = iniap
.ic V(net3bb) = inip


/////////end of initialization of MTJ/////

/////////////supply//////////
.param vddpap=-1
.param vddapp=1
.param vddv = 0.3

Vvdd vdd 0 vddv
Vclk clk 0 pulse(vddv 0 0 0 0 2n 4n)
Vi1 i1 0 pulse(0 vddv 0 0 0 4n 8n)
Vi1bar i1bar 0 pulse(vddv 0 0 0 0 4n 8n )
Vi2 i2 0 pulse(0 vddv 0 0 0 8n 16n)
Vi2bar i2bar 0 pulse(vddv 0 0 0 0 8n 16n )
Vi3 i3 0 pulse(0 vddv 0 0 0 16n 32n)
Vi3bar i3bar 0 pulse(vddv 0 0 0 0 16n 32n )
Vi4 i4 0 pulse(0 vddv 0 0 0 32n 64n)
Vi4bar i4bar 0 pulse(vddv 0 0 0 0 32n 64n )
Vi5 i5 0 pulse(0 vddv 0 0 0 64n 128n)
Vi5bar i5bar 0 pulse(vddv 0 0 0 0 64n 128n)
Vi6 i6 0 pulse(0 vddv 0 0 0 128n 256n)
Vi6bar i6bar 0 pulse(vddv 0 0 0 0 128n 256n)
$Vcin1 cin1 0 pulse(0 vddv 0 0 0 256n 512n)
```

$Vcin1bar cin1bar 0 pulse(vddv 0 0 0 0 256n 512n)
$Vcin2 cin2 0 pulse(0 vddv 0 0 0 512n 1024n)
$Vcin2bar cin2bar 0 pulse(vddv 0 0 0 0 512n 1024n)
Vcin3 cin3 0 pulse(0 vddv 0 0 0 1024n 2048n)
Vcin3bar cin3bar 0 pulse(vddv 0 0 0 0 1024n 2048n)
Vcin4 cin4 0 pulse(0 vddv 0 0 0 2048n 4096n)
Vcin4bar cin4bar 0 pulse(vddv 0 0 0 0 2048n 4096n)
$Vi7 net2b blb pwl(0ns 0 20ns 0 20.1ns vddpap 21.1ns vddpap 21.2ns 0 200ns 0 200.1ns vddapp 201.1ns vddapp 201.2ns 0)
$Vi7bar net2bb blbb pwl(0ns 0 20ns 0 20.1ns vddapp 21.1ns vddapp 21.2ns 0 200ns 0 200.1ns vddpap 201.1ns vddpap 201.2ns 0)
$Vi7 net2b blb pulse(vddpap vddapp 0 0 0 4096n 8192n)
$Vi7bar net2bb blbb pulse(vddapp vddpap 0 0 0 4096n 8192n)
Vcin1 cin1 0 pulse(0 vddv 0 0 0 20n 50n)
Vcin1bar cin1bar 0 pulse(vddv 0 0 0 0 30n 50n)
Vcin2 cin2 0 pulse(0 vddv 0 0 0 40n 50n)
Vcin2bar cin2bar 0 pulse(vddv 0 0 0 0 10n 50n)
////////////end of supply////////////

//////////////transient response////////////
.measure TRAN iavg AVG i(vvdd) FROM=0 TO=100e-9

$the result will be stored on the .lis file, someting like this:
$iavg=10e-6

$If you want you can also do the power calculation with hspice:
.measure TRAN power PARAM='iavg*vddv'

.probe tran I(vvqblb)
.plot tran power
.probe tran power
.print power
.tran 1p 100n

```
.option post=2
.option gmin=1e-9
.option gmindc=1e-9
.end
////////////end of transient response//////////
```

VI. Code of Sum of proposed 7:2 compressor

```
*sum of 7:2 compressor

//////////file inclusion/////////////
.hdl 'stt.txt'
.include '16nm_nfet.txt'
.include '16nm_pfet.txt'
$.include '16nm_bulk.txt'
//////////end of file inclusion////////////////


//////////parameter initializatoin/////////
.param r1=20n
.param tox1= 0.85n
.param inip=0
.param iniap=3.14
.param width= 16n
.param length=16n
//////////end of parameter initialization/////////

$mn1 drain gate source body nmos w = width l = length        $ transistor

//////////////////pre charge sense amplifier///////////////

mpfet1 vdd clk qbar vdd pfet width=width l=length
mpfet2 vdd q qbar vdd pfet width=width l=length
mpfet3 vdd clk q vdd pfet width=width l=length
```

```
mpfet4 vdd qbar q vdd pfet width=width l=length
mnfet1 qbar q nx10 0 nfet width=width l=length
mnfet2 q qbar nx10bar 0 nfet width=width l=length
//////////////////pre charge sense amplifier///////////////

//////////////////logic block//////////////////////////

$mnl1bar nx1 i1bar net2b 0 nfet width=width l=length
$mnl1 nx1bar i1 net2b 0 nfet width=width l=length
$mnll1 nx1 i1 net2bb 0 nfet width=width l=length
$mnll1bar nx1bar i1bar net2bb 0 nfet width=width l=length

mnl2bar nx2 i2bar net2b 0 nfet width=width l=length
mnl2 nx2bar i2 net2b 0 nfet width=width l=length
mnll2 nx2 i2 net2bb 0 nfet width=width l=length
mnll2bar nx2bar i2bar net2bb 0 nfet width=width l=length

mnl3bar nx3 i3bar nx2 0 nfet width=width l=length
mnl3 nx3bar i3 nx2 0 nfet width=width l=length
mnll3 nx3 i3 nx2bar 0 nfet width=width l=length
mnll3bar nx3bar i3bar nx2bar 0 nfet width=width l=length

mnl4bar nx4 i4bar nx3 0 nfet width=width l=length
mnl4 nx4bar i4 nx3 0 nfet width=width l=length
mnll4 nx4 i4 nx3bar 0 nfet width=width l=length
mnll4bar nx4bar i4bar nx3bar 0 nfet width=width l=length

mnl5bar nx5 i5bar nx4 0 nfet width=width l=length
mnl5 nx5bar i5 nx4 0 nfet width=width l=length
mnll5 nx5 i5 nx4bar 0 nfet width=width l=length
mnll5bar nx5bar i5bar nx4bar 0 nfet width=width l=length

mnl6bar nx6 i6bar nx5 0 nfet width=width l=length
mnl6 nx6bar i6 nx5 0 nfet width=width l=length
```

mnll6 nx6 i6 nx5bar 0 nfet width=width l=length
mnll6bar nx6bar i6bar nx5bar 0 nfet width=width l=length

mnl7bar nx7 i7bar nx6 0 nfet width=width l=length
mnl7 nx7bar i7 nx6 0 nfet width=width l=length
mnll7 nx7 i7 nx6bar 0 nfet width=width l=length
mnll7bar nx7bar i7bar nx6bar 0 nfet width=width l=length

mnlcin1bar nx8 cin1bar nx7 0 nfet width=width l=length
mnlcin1 nx8bar cin1 nx7 0 nfet width=width l=length
mnllcin1 nx8 cin1 nx7bar 0 nfet width=width l=length
mnllcin1bar nx8bar cin1bar nx7bar 0 nfet width=width l=length

mnlcin2bar nx9 cin2bar nx8 0 nfet width=width l=length
mnlcin2 nx9bar cin2 nx8 0 nfet width=width l=length
mnllcin2 nx9 cin2 nx8bar 0 nfet width=width l=length
mnllcin2bar nx9bar cin2bar nx8bar 0 nfet width=width l=length

mnlcin3bar nx10 cin3bar nx9 0 nfet width=width l=length
mnlcin3 nx10bar cin3 nx9 0 nfet width=width l=length
mnllcin3 nx10 cin3 nx9bar 0 nfet width=width l=length
mnllcin3bar nx10bar cin3bar nx9bar 0 nfet width=width l=length

//////////////////////end of logic block///////////////

//////////////////clock ///////////////

mnclkl blb clk 0 0 nfet w=width l=length
mnclkr blbb clk 0 0 nfet w=width l=length

////////////////end of clock////////////////

```
///////////////MTJ1/////////////////

xM1b net1b net2b net3b 0 hys r=r1
xM2b net3b 0 ani r=r1
xC1b net3b 0 caps1 r=r1
xRoutb net3b net4b 0 vcr0

xRwriteb net3b blb net1b vcr r=r1 tox=tox1
//////////////end of MTJ1/////////////

////////////MTJ2/////////////////

xM1bb net1bb net2bb net3bb 0 hys r=r1
xM2bb net3bb 0 ani r=r1
xC1bb net3bb 0 caps1 r=r1
xRoutbb net3bb net4bb 0 vcr0

xRwritebb net3bb blbb net1bb vcr r=r1 tox=tox1
///////////////end of MTJ2////////////

//////////initialization of MTJ///////////
.ic V(net3b) = iniap
.ic V(net3bb) = inip

/////////end of initialization of MTJ/////

/////////////supply//////////
.param vddpap=-1
.param vddapp=1
.param vddv = 0.3

Vvdd vdd 0 vddv
Vclk clk 0 pulse(vddv 0 0 0 0 2n 4n)
Vi1 i1 0 pulse(0 vddv 0 0 0 4n 8n)
```

```
Vi1bar i1bar 0 pulse(vddv 0 0 0 0 4n 8n )
Vi2 i2 0 pulse(0 vddv 0 0 0 8n 16n)
Vi2bar i2bar 0 pulse(vddv 0 0 0 0 8n 16n )
Vi3 i3 0 pulse(0 vddv 0 0 0 16n 32n)
Vi3bar i3bar 0 pulse(vddv 0 0 0 0 16n 32n )
Vi4 i4 0 pulse(0 vddv 0 0 0 32n 64n)
Vi4bar i4bar 0 pulse(vddv 0 0 0 0 32n 64n )
Vi5 i5 0 pulse(0 vddv 0 0 0 64n 128n)
Vi5bar i5bar 0 pulse(vddv 0 0 0 0 64n 128n)
Vi6 i6 0 pulse(0 vddv 0 0 0 128n 256n)
Vi6bar i6bar 0 pulse(vddv 0 0 0 0 128n 256n)
Vcin2 cin2 0 pulse(0 vddv 0 0 0 256n 512n)
Vcin2bar cin2bar 0 pulse(vddv 0 0 0 0 256n 512n )
Vcin3 cin3 0 pulse(0 vddv 0 0 0 512n 1024n)
Vcin3bar cin3bar 0 pulse(vddv 0 0 0 0 512n 1024n )
$Vcin1 net2b blb pulse(vddpap vddapp 0 0 0 1024n 2048n)
$Vcin1bar net2bb blbb pulse(vddapp vddpap 0 0 0 1024n 2048n )
////////////end of supply////////////

//////////////transient response////////////
.measure TRAN iavg AVG i(vvdd) FROM=0 TO=100e-9

$the result will be stored on the .lis file, someting like this:
$iavg=10e-6

$If you want you can also do the power calculation with hspice:
.measure TRAN power PARAM='iavg*vddv'



.probe tran I(vvqblb)
.plot tran power
.probe tran power
.print power
.tran 1p 100n
```

```
.option post=2
.option gmin=1e-9
.option gmindc=1e-9
.end
////////////end of transient response//////////
```

VII. Code of Carry of proposed 7:2 compressor

```
*sum of 6:2 compressor

//////////file inclusion/////////////
.hdl 'stt.txt'
.include '16nm_nfet.txt'
.include '16nm_pfet.txt'
$.include '16nm_bulk.txt'
//////////end of file inclusion////////////////


//////////parameter initializatoin/////////
.param r1=20n
.param tox1= 0.85n
.param inip=0
.param iniap=3.14
.param width= 16n
.param length=16n
//////////end of parameter initialization/////////

$mn1 drain gate source body nmos w = width l = length        $ transistor

//////////////////pre charge sense amplifier///////////////

mpfet1 vdd clk qbar vdd pfet width=width l=length
mpfet2 vdd q qbar vdd pfet width=width l=length
mpfet3 vdd clk q vdd pfet width=width l=length
```

```
mpfet4 vdd qbar q vdd pfet width=width l=length
mnfet1 qbar q nx8 0 nfet width=width l=length
mnfet2 q qbar nx8bar 0 nfet width=width l=length
//////////////////pre charge sense amplifier///////////////

//////////////////logic block//////////////////////////

mnl1bar nx1 i1bar net2b 0 nfet width=width l=length
mnl1 nx1bar i1 net2b 0 nfet width=width l=length
mnll1 nx1 i1 net2bb 0 nfet width=width l=length
mnll1bar nx1bar i1bar net2bb 0 nfet width=width l=length

mnl2bar nx2 i2bar nx1 0 nfet width=width l=length
mnl2 nx2bar i2 nx1 0 nfet width=width l=length
mnll2 nx2 i2 nx1bar 0 nfet width=width l=length
mnll2bar nx2bar i2bar nx1bar 0 nfet width=width l=length

mnl3bar nx3 i3bar nx2 0 nfet width=width l=length
mnl3 nx3bar i3 nx2 0 nfet width=width l=length
mnll3 nx3 i3 nx2bar 0 nfet width=width l=length
mnll3bar nx3bar i3bar nx2bar 0 nfet width=width l=length

mnl4bar nx4 i4bar nx3 0 nfet width=width l=length
mnl4 nx4bar i4 nx3 0 nfet width=width l=length
mnll4 nx4 i4 nx3bar 0 nfet width=width l=length
mnll4bar nx4bar i4bar nx3bar 0 nfet width=width l=length

mnl5bar nx5 i5bar nx4 0 nfet width=width l=length
mnl5 nx5bar i5 nx4 0 nfet width=width l=length
mnll5 nx5 i5 nx4bar 0 nfet width=width l=length
mnll5bar nx5bar i5bar nx4bar 0 nfet width=width l=length

mnl6bar nx6 i6bar nx5 0 nfet width=width l=length
mnl6 nx6bar i6 nx5 0 nfet width=width l=length
```

```
mnll6 nx6 i6 nx5bar 0 nfet width=width l=length
mnll6bar nx6bar i6bar nx5bar 0 nfet width=width l=length

mnlcin1bar nx7 cin1bar nx6 0 nfet width=width l=length
mnlcin1 nx7bar cin1 nx6 0 nfet width=width l=length
mnllcin1 nx7 cin1 nx6bar 0 nfet width=width l=length
mnllcin1bar nx7bar cin1bar nx6bar 0 nfet width=width l=length

mnlcin2bar nx8 cin2bar nx7 0 nfet width=width l=length
mnlcin2 nx8bar cin2 nx7 0 nfet width=width l=length
mnllcin2 nx8 cin2 nx7bar 0 nfet width=width l=length
mnllcin2bar nx8bar cin2bar nx7bar 0 nfet width=width l=length

mnlcin4 nx8 cin4 ndcin4 0 nfet width=width l=length
mnlcin3 ndcin4 cin3 net2b 0 nfet width=width l=length
mnllcin4bar nx8bar cin4bar ndcin4bar 0 nfet width=width l=length
mnllcin3bar ndcin4bar cin3bar net2bb 0 nfet width=width l=length


//////////////////////end of logic block////////////////


//////////////////clock ////////////////

mnclkl blb clk 0 0 nfet w=width l=length
mnclkr blbb clk 0 0 nfet w=width l=length

//////////////////end of clock/////////////////

///////////////MTJ1/////////////////

xM1b net1b net2b net3b 0 hys r=r1
xM2b net3b 0 ani r=r1
xC1b net3b 0 caps1 r=r1
```

```
xRoutb net3b net4b 0 vcr0

xRwriteb net3b blb net1b vcr r=r1 tox=tox1
//////////////end of MTJ1//////////////

////////////MTJ2/////////////////

xM1bb net1bb net2bb net3bb 0 hys r=r1
xM2bb net3bb 0 ani r=r1
xC1bb net3bb 0 caps1 r=r1
xRoutbb net3bb net4bb 0 vcr0

xRwritebb net3bb blbb net1bb vcr r=r1 tox=tox1
///////////////end of MTJ2////////////

//////////initialization of MTJ///////////
//i7
.ic V(net3b) = iniap
.ic V(net3bb) = inip


/////////end of initialization of MTJ/////

////////////supply/////////
.param vddpap=1
.param vddapp=-1
.param vddv = 0.3

Vvdd vdd 0 vddv
Vclk clk 0 pulse(vddv 0 0 0 0 2n 4n)
Vi1 i1 0 pulse(0 vddv 0 0 0 4n 8n)
Vi1bar i1bar 0 pulse(vddv 0 0 0 0 4n 8n )
Vi2 i2 0 pulse(0 vddv 0 0 0 8n 16n)
Vi2bar i2bar 0 pulse(vddv 0 0 0 0 8n 16n )
```

```
Vi3 i3 0 pulse(0 vddv 0 0 0 16n 32n)
Vi3bar i3bar 0 pulse(vddv 0 0 0 0 16n 32n )
Vi4 i4 0 pulse(0 vddv 0 0 0 32n 64n)
Vi4bar i4bar 0 pulse(vddv 0 0 0 0 32n 64n )
Vi5 i5 0 pulse(0 vddv 0 0 0 64n 128n)
Vi5bar i5bar 0 pulse(vddv 0 0 0 0 64n 128n)
Vi6 i6 0 pulse(0 vddv 0 0 0 128n 256n)
Vi6bar i6bar 0 pulse(vddv 0 0 0 0 128n 256n)
Vcin1 cin1 0 pulse(0 vddv 0 0 0 256n 512n)
Vcin1bar cin1bar 0 pulse(vddv 0 0 0 0 256n 512n )
Vcin2 cin2 0 pulse(0 vddv 0 0 0 512n 1024n)
Vcin2bar cin2bar 0 pulse(vddv 0 0 0 0 512n 1024n )
Vcin3 cin3 0 pulse(0 vddv 0 0 0 20n 50n)
Vcin3bar cin3bar 0 pulse(vddv 0 0 0 0 30n 50n )
Vcin4 cin4 0 pulse(0 vddv 0 0 0 40n 50n)
Vcin4bar cin4bar 0 pulse(vddv 0 0 0 0 10n 50n )

////////////end of supply////////////

//////////////transient response////////////
.measure TRAN iavg AVG i(vvdd) FROM=0 TO=100e-9

$the result will be stored on the .lis file, someting like this:
$iavg=10e-6

$If you want you can also do the power calculation with hspice:
.measure TRAN power PARAM='iavg*vddv'



.probe tran I(vvqblb)
.plot tran power
.probe tran power
.print power
.tran 1p 100n
```

```
.option post=2
.option gmin=1e-9
.option gmindc=1e-9
.end
//////////////end of transient response////////////
```