LOW POWER ACCURACY CONFIGURABLE MULTIPLIER USING NEW CARRY MASKABLE ADDER AND FS-GDI TECHNIQUE

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHONOLOGY

IN

VLSI AND EMBEDDED SYSTEM DESIGN

Submitted By

GOLLU RAJKUMAR

(2k19/VLS/12)

Under the supervision of

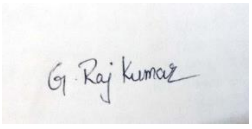
Prof. NEETA PANDEY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I Gollu Rajkumar student of M.Tech (VLSI and Embedded Systems), hereby declare that the Project Dissertation titled **"LOW POWER ACCURACY CONFIGURABLE MULTIPLIER USING NEW CARRY MASKABLE ADDER AND FULL SWING -GDI TECHNIQUE"** which is submitted by me to the Department of Electronics and Communication Engineering, Delhi Technological University, for major project, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.



Place: Delhi **GOLLU RAJKUMAR**
Date: 19.09.21 (2K19/VLS/12)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation Report titled **"LOW POWER ACCURACY CONFIGURABLE MULTIPLIER USING NEW CARRY MASKABLE ADDER AND FULL SWING -GDI TECHNIQUE"** which is submitted by **GOLLU RAJKUMAR**, **2K19/VLS/12** of Electronics and Communication Department, Delhi Technological University, Delhi for the Minor Project, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: **19.09.21**

Neeta

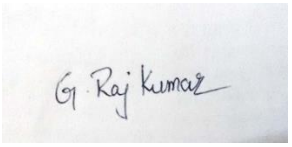
Prof. NEETA PANDEY
SUPERVISOR

ACKNOWLEDGEMENT

A successful project can never be prepared by the efforts of the person to whom the project is assigned, but it also demands the help and guardianship of people who helped in completion of the project. I would like to thank all those people who have helped me in this research and inspired me during my study.

With profound sense of gratitude, I thank Prof. Neeta Pandey my Research Supervisor, for his encouragement, support, patience and his guidance in this project work.

I take immense delight in extending my acknowledgement to my family and friends who have helped me throughout this project work.



GOLLU RAJKUMAR
(2K19/VLS/12)

ABSTRACT

Adders and multipliers are important arithmetic modules in the efficacy and performance of digital signal processor (DSP) systems. The speed and power consumption of arithmetic units have an impact on processor efficiency. It is improved by using approximation computing in arithmetic units with tolerable output degradation.

Approximate computing is a recent concept that aims to create a cost-effective design strategy while sacrificing computational quality for error-tolerant applications. Approximate computing can be used in research at both the hardware and software levels.

In mathematical and logical procedures, adding is the most important step. Approximate Computing is used in VLSI design to reduce the number of transistors, delay, and power limitations. The employment of approximate adders in error-prone applications is viable since approximate addition is considered to be an effective energy trading technology in terms of performance and accuracy.

Accuracy that can be customised Approximate Adder designs have been shown to be effective in alleviating these constraints. Using Full Swing-Modified Gate Diffusion Input (GDI) approach, a new circuit design for a Carry Maskable Adder has been presented. Proposed circuit design simulations have been simulated out in 45-nm process technology using LTSpice XVII. The results indicate 37% and 32% reduction in Power and Delay respectively.

Using the proposed Adder a low power accuracy configurable Multiplier is designed and simulated out in 45nm process technology using LTSpice XVII. The results indicate 51.2% and 37% reduction in Power and Delay respectively.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

With the evolution in technology, the number of transistors in an IC is increasing and components are getting more small in size day by day. Due to which, the requirement for much smaller implementation of circuits is growing. Since adder is the crucial logical block of arithmetic and logic unit, many researchers have tried to obtain a circuit which can perform the same functionality with fewer number of transistors. One of the computation techniques that can be employed in this area of interest, to implement circuits with lesser hardware is Approximate Computing [1-2].

The increment in the transistors count mounted on a IC is accompanied by the demand of Power reduction. Over the years, approximate computing has managed to be a encouraging method to decrease energy consumption, area and delay constraints in Physical design. But it also results in a loss of computational accuracy. Power can be reduced by upgrading the module into an approximate version. It is required to examine the different logic functionalities of the circuit when designing the inexact version of the module. As one of the elementary, but crucial blocks of many basic deigns, adders have pulled in a broad enthusiasm for upgrading and implementing approximate modules.

Because the adder is the most important digital circuit in the arithmetic and logic unit, several academics have presented their research [3]-[6] on using approximate computing to enhance the adders' performance in terms of energy and speed.

Approximate adders have become popular for use in circuits where some loss is acceptable. [4] Proposes approximation mirror adders (AMAs), which conserve energy by using a less number of transistors in a mirror adder design.

In [5] XOR/XNOR gates-based adders have been proposed implemented by pass transistors which provide a significant decrease in power with better performance.[6] utilizes transmission gates in place of pass transistors which proves to be a better replacement as unlike pass transistors, they do not suffer from signal degradation.

1.2 Project Organization:

The project work is categorized as , In the chapter 2 a literature review on different approximate adders and carry maskable full adder is discussed and in chapter 3 Gate Diffusion Input Technique is reviewed and Full Swing GDI based CMF A is proposed and in chapter 4 simulations and results of these adders are evaluated in terms of power and delay and in chapter 5 low power accuracy configurable multiplier is designed with the help of the proposed CMFA and results are evaluated in terms of power and delay and conclusion is given in chapter 6.

So, the objective of this research is to design a Adder whose Accuracy is configurable and using Full swing GDI technique and implement a low power accuracy configurable multiplier using the proposed adder.

CHAPTER 2

LITERATURE REVIEW

Approximation is done at different levels of the design. In this chapter a review on approximate adders is discussed particularly approximate adders in gate level and transistor level.

2.1 Approximate Adders in the Gate Level

Inexact Full Adders (INEXA). Three approximate adder design [3] are implemented using the gate level approximation and a lesser number of transistors. One of the full adder outputs is estimated in these inexact adders, while the other yield has the accurate result.

INEXA1. The correct sum signal is maintained in this adder, while the carryout signal is estimated. INEXA1's gate level execution is depicted in Figure 2.1. In the carry signal, the INEXA1 gives an exact sum and two errors out of eight occurrences.

$$SUM = A \text{ XOR } B \text{ XOR } C \qquad (2.1)$$

$$CARRY = \sim(SUM) \qquad (2.2)$$

Two XOR gates and one NOT gate are required to implement 1-bit INAX1 [3]. The XOR gate can only have two inputs (AND, OR, and NOT), hence it requires two AND gates, one OR gate, and one NOT gate [6]. INEXA1 is then constructed using four AND gates, two OR gates, and two NOT gates. As a result, nine basic gates are required to implement INEXA1.

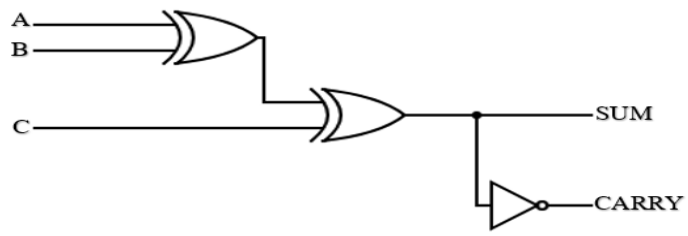


Figure 2.1: Gate level diagram of INEXA1

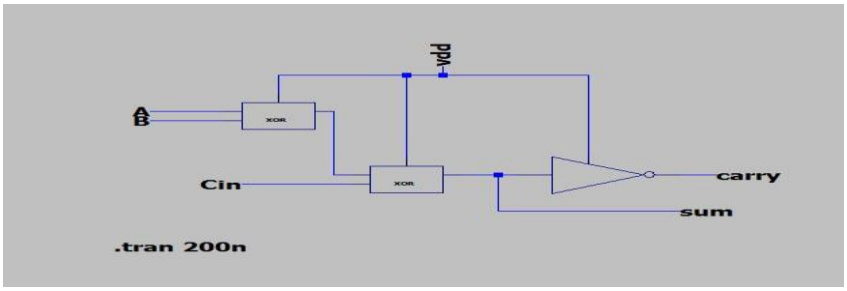


Figure 2.1(a): Schematic of INEXA1

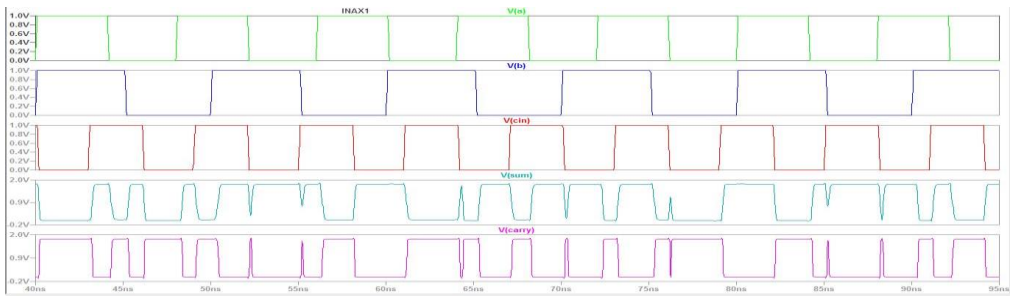


Figure 2.1(b): Transient Analysis of INEXA1

INEXA2. INEXA2 [3] is put together by keeping exact outputs in the carryout signal while using an estimated result in the output sum signal. INEXA2 replaces the second XOR gate in the accurate complete adder with an OR gate. Figure 2.2 depicts the INEXA2 gate level implementation. 2 Out of 8 occurrences, the total signal has two errors, whereas the carry signal provides an accurate output.

$$SUM = (A \text{ XOR } B) \text{ OR } C \qquad (2.3)$$

$$CARRY = AB + (A \text{ XOR } B)C \qquad (2.4)$$

1 XOR gate, 2 OR gates, and 2 AND gates are required for a 1-bit INEXA2. INEXA2 is constructed with four AND gates, a NOT gate, and 3 OR gates when the XOR is confined to 2-input basic gates. As a result, eight basic gates are required to build INEXA2.

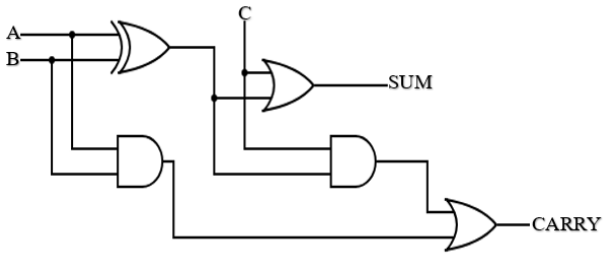


Figure 2.2 Gate level diagram of INEXA2

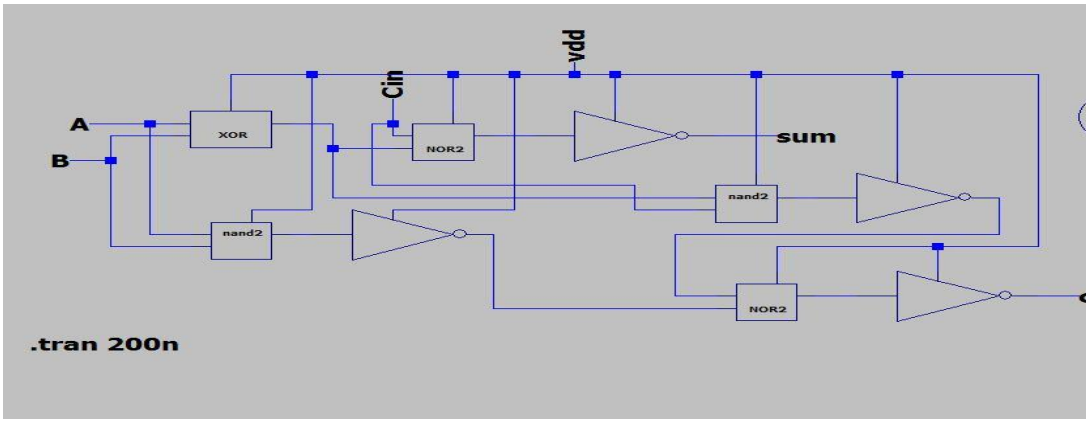


Figure 2.2(a) Schematic of INEXA2

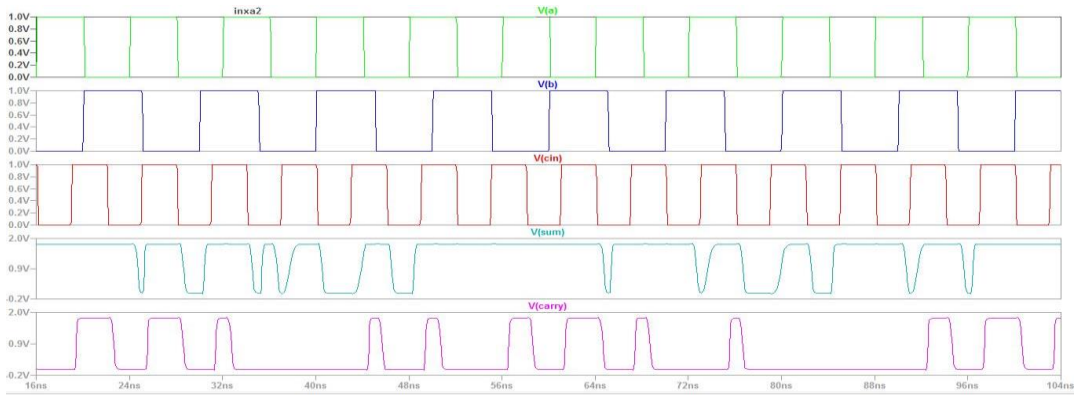


Figure 2.2(b) Transient Analysis of INEXA2

INEXA3. INEXA3 [3] is illustrated by approximating the sum signal while keeping the carry signal precise. Figure 2.3 depicts the INEXA3 gate level diagram. In the total signal, INEXA3 creates two errors out of every eight cases.

$$SUM = \sim(CARRY) \qquad (2.5)$$

$$CARRY = AB + (A \text{ XOR } B) \qquad (2.6)$$

INXA3 makes use of one XOR gate, two AND gates, one OR gate, and one NOT gate. Following the XOR gate constraint, INXA3 was redeveloped to use four AND gates, two NOT gates, and two OR gates. As a result, eight basic gates are required to build INEXA2.

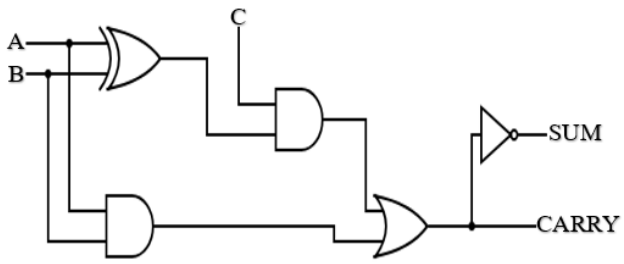


Figure 2.3 Gate level diagram of INEXA3

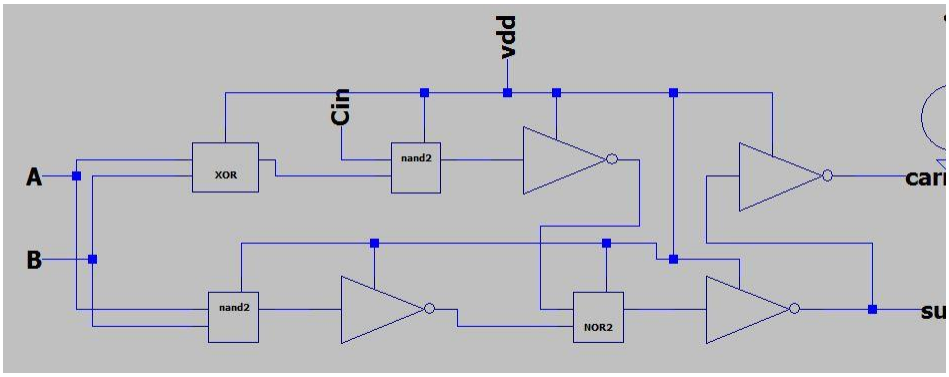


Figure 2.3(a) Schematic of INXA3

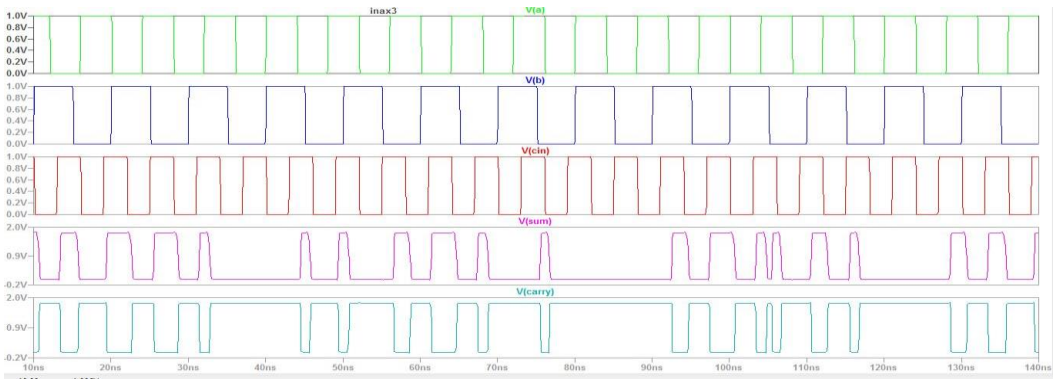


Figure 2.3(b) Transient Analysis of INXA 3

Approximate Full Adders (AXFA). AXFAs [2] are built by altering the gate level of a standard Full adder, and they are also designed so that the carryout signal output is not dependent on the carry input. When implemented in multibit adders, the common strategy utilised in these AXFAs is to diminish the length of carry lifespan. Approximation is used only in the carry signal to implement AXFAs.

The output of the carryout signal is forwarded as the input of A in AXFA1. In AXFA2, B's input is treated as the carry signal's output. In the carry signal, AXFA1 and AXFA2 cause two errors out of every eight cases

For AXFA1 SUM = A XOR B XOR C (2.7)

CARRY = A (2.8)

For AXFA2 SUM = A XOR B XOR C (2.9)

CARRY = B (2.10)

Figures 2.4 and 2. 5 show the gate level logic implementation of AXFA1 and AXFA2[2]. Two XOR gates are used to realise AXFA1 and AXFA2. As a result, eight basic gates are required to implement AXFA1 and AXFA2.

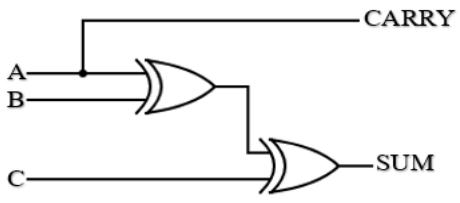


Figure 2.4: Gate level implementation of AXFA1

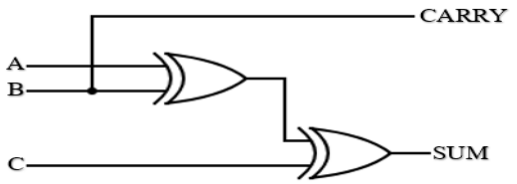


Figure 2.5: Gate level implementation of AXFA 2

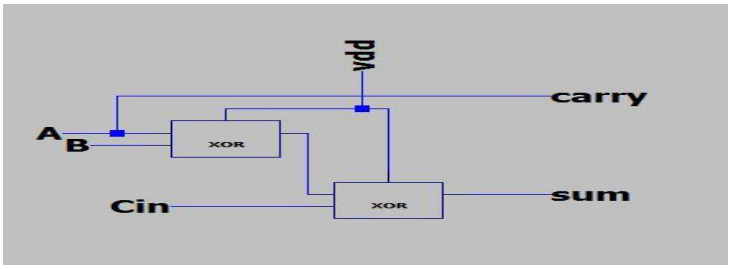


Figure 2.4 (a) Schematic diagram of AXFA1

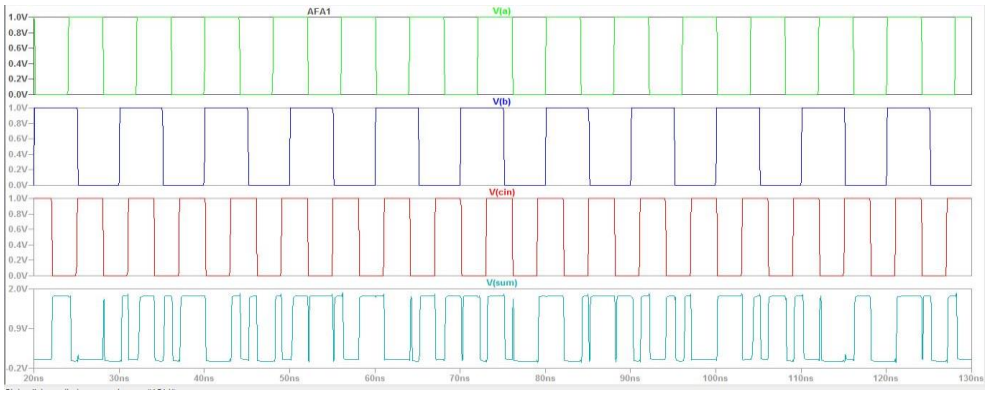


Figure 2.4(b) Transient Analysis of AXFA1

AXFA3: The output of the carryout signal is seen as A.B. in AXFA3. The graphic shows the gate level implementation in figure 2.6.

$$SUM = A \text{ XOR } B \text{ XOR } C \qquad (2.11)$$

$$CARRY = AB \qquad (2.12)$$

Two XOR gates and one AND gate are used to perform AXFA3. AXFA3 is produced by utilising 5 AND gates, 2 OR gates, and 2 NOT gates once the XOR gate is limited. As a result, nine basic gates are required to create AXFA3.

AXFA4: The carryout signal is accomplished as A+B in AXFA4 [2]. Figure 2.7 depicts the gate level circuit diagram.

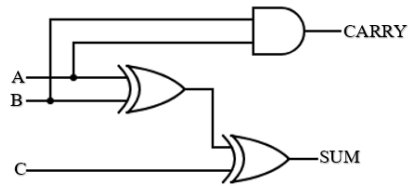


Figure 2.6 Gate level implementation of AXFA3

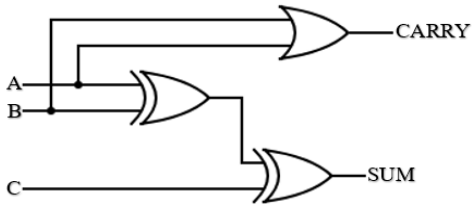


Figure 2.7 Gate level implementation of AXFA4

$$SUM = A \text{ XOR } B \text{ XOR } C \qquad (2.13)$$

$$CARRY = A+B \qquad (2.14)$$

AXFA4 is made up of two XOR gates and one OR gate. After the XOR gate is limited, AXFA4 is built using four AND gates, three OR gates, and two NOT gates. As a result, nine basic gates are required to create AXFA3.

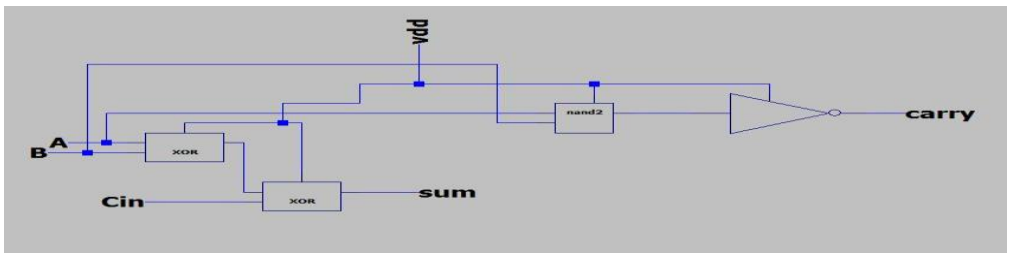


Figure 2.6(a) Schematic of AXFA 3

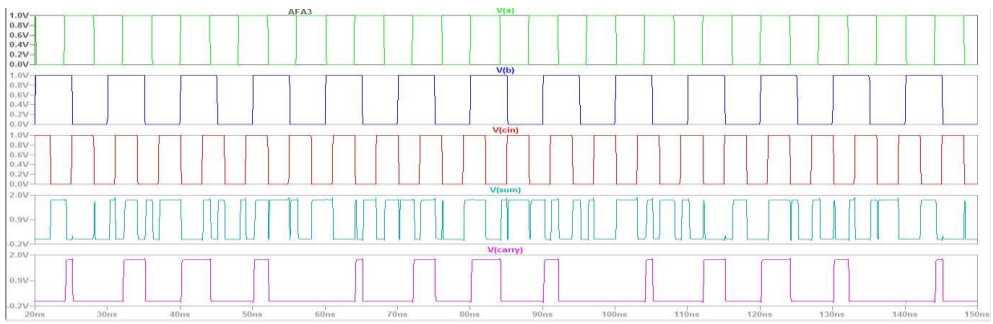


Figure 2.6(b) Transient Analysis of AXFA3

2.2 Transistor Level Approximation of Full adder

By excluding few transistors from the transistor level model of the exact full adder, the circuit complication, capacitances, and power dissipation are narrowed down.

2.2.1 Approximate Mirror Adder

One of the popularly used accurate implementation of full adder is mirror adder. The Approximate-Mirror-Adder (AMA) can be derived from this design by reduction of transistors. By removing some transistors, three approximate mirror adder designs can be obtained which have been presented in [4]. The circuits shown in figure 2.8 is capable of achieving lower power dissipation and reduced circuit complexity. Due to less transistors, the node capacitances can get charged and discharged at a faster rate, thus experiences a shorter delay. A step-by-step procedure is being followed in coming up with several inexact mirror adder cells with less transistors. One by one the transistors are removed from the conventional schematic. However, these cannot be removed in an arbitrary fashion. It is to be made sure that not even a single combination of inputs A, B and Cin results in short-circuit or open-circuit in the modified schematic of the adder. Another crucial point to be kept in mind is that the resulting simplified schematic should introduce a small amount of loss in the truth table of full adder

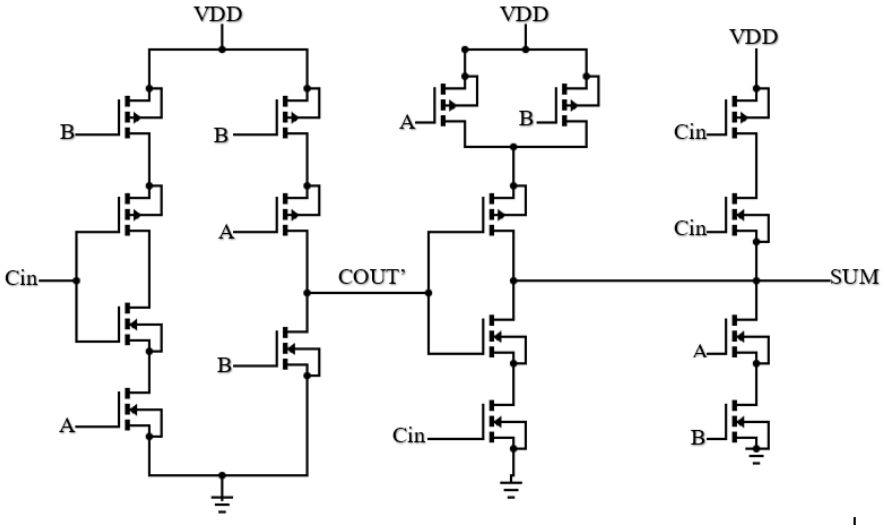


Figure2.8: Approximate Mirror Adder (AMA 1)

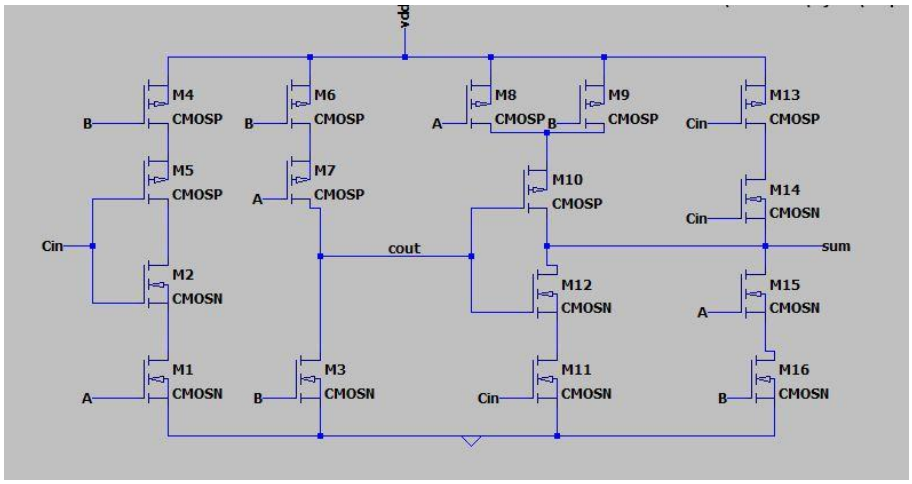


Figure 2.8(a) Schematic of Mirror Adder (AMA1)

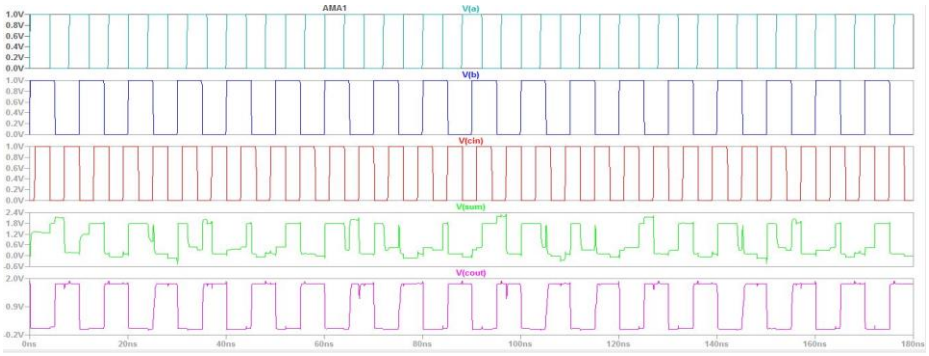


Figure 2.8(b) Transient Analysis of AMA1

2.2.2 Approximate XNOR/XOR Adder

AXA-1

Approximate XOR based Adder type 1[5] is shown in figure 2.9. In the type 1 schematic, the XOR operation is carried out by two pass transistors and an inverter connected to inputs X and Y respectively.

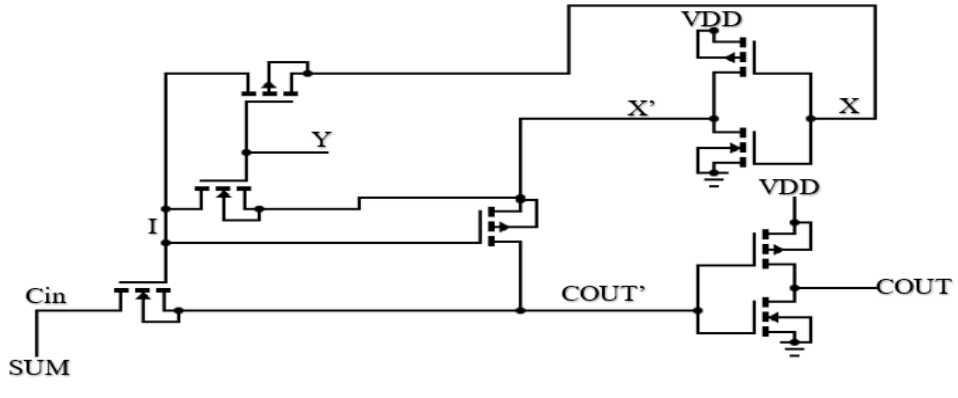


Figure2.9 Transistor implementation of Approximate XOR Adder Type-1

When the input Y equals ‘1’, I becomes equal to complement of input X and when Y is ‘0’, I equals to X i.e., I=X⊕Y. The Sum and Cout can be calculated precisely only for 4 out of 8 different combinations of the inputs. Thus the total error distance that can be attained with this circuit design is 4. The total transistor count is 8. The expression for Sum and Cout are given by:

$$SUM = Cin; \qquad (2.15)$$

$$COUT = (X \oplus Y)Cin + Y X \qquad (2.16)$$

AXA-2

The circuit design in figure 2.10 shows the implementation of the Approximate XOR based adder Type-2.[5]

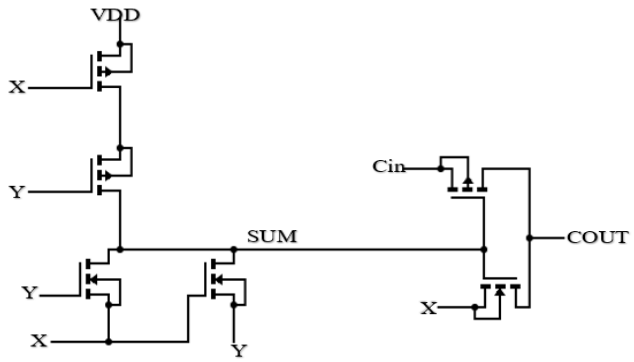


Figure2.10 Transistor implementation of Approximate XOR Adder Type-2

AXA-2 can be implemented with the help of only 6 transistors, which utilizes one pass transistor block and 4 transistors for the XNOR operation. Sum is accurate for only 4 input combinations out of the 8, while Cout can be calculated accurately for all input combinations. For this circuit design also, the overall error distance that can be attained is 4. The expressions for Sum and Carryout are:

$$SUM = (X \text{ XOR } Y)' \qquad (2.17)$$

$$COUT = (X \text{ XOR } Y)\ Cin + XY \qquad (2.18)$$

As the pass transistors suffer from the problem of signal degradation because of threshold voltage drop, the output signals for SUM and COUT do not achieve a full swing for some transitions.

In these approximate adders the transmission gates have been used in place of pass transistors as they suffer from the problem of signal distortion or degradation.

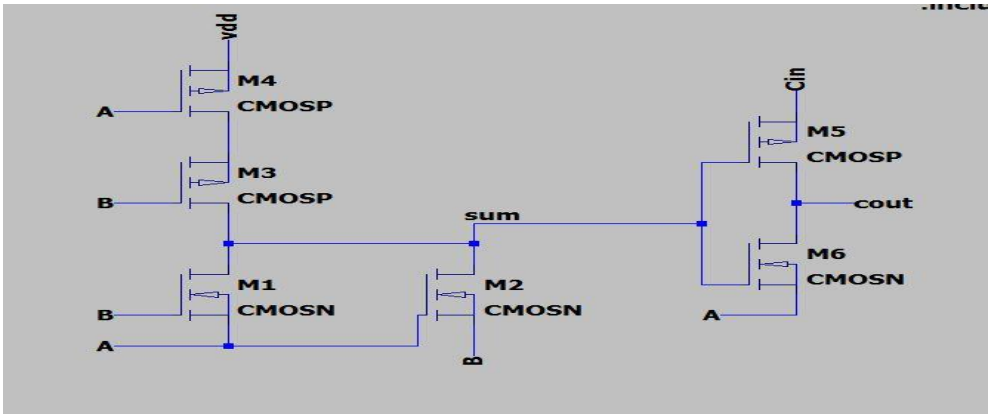


Figure 2.10(a) Schematic of Approximate XOR Adder Type-2

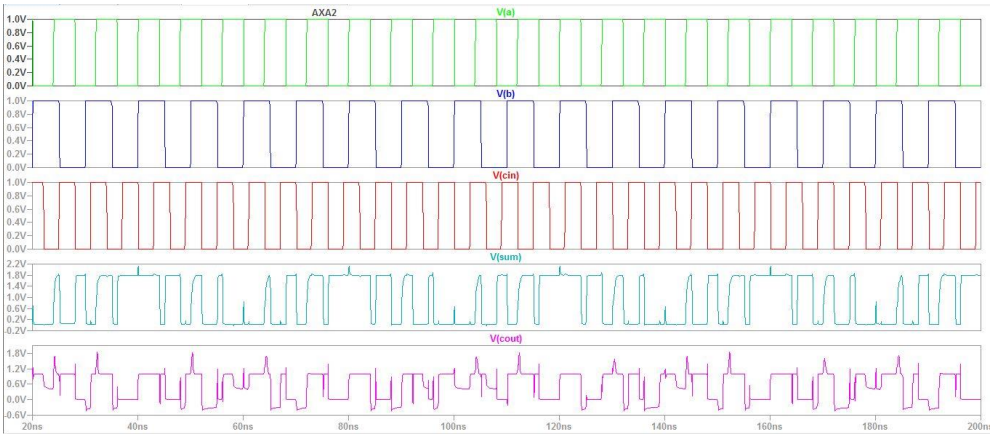
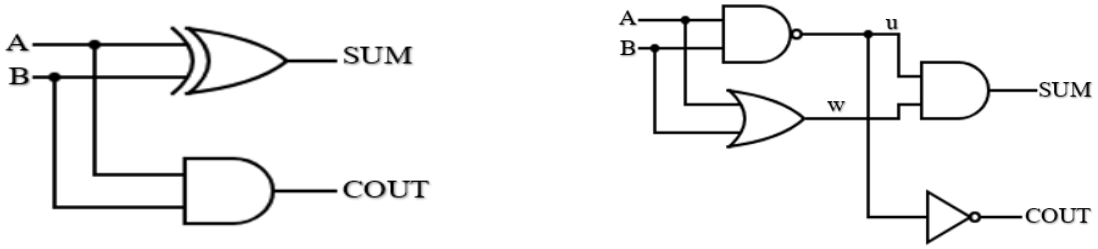


Figure 2.10(b) Transient Analysis of Approximate XOR Adder Type -2

2.3 Carry Maskable Adder

Carry Maskable Adder [9] is one such approximate adder where the accuracy can be dynamically configured. It means it can be used as both exact full adder and approximate adder during the run time in any application domains. By slightly changing the conventional adder circuit carry maskable adder circuit is designed.

Figure 2.10 depicts a traditional half adder (a). The sum is computed using an XOR gate, while the carry out is computed using an AND gate (COUT). Figure 2.10 depicts a half adder's equivalent circuit (b). The dashed frame depicts a 2-input XOR gate's comparable circuit. The truth table for the corresponding design of a half adder is depicted in Table 2.1



(a) (b)

Figure 2.11: (a) Typical Half Adder (b) corresponding circuit of a half adder

Table 2.1. Truth Table of corresponding circuit of half adder

Inputs		Internal Signals		Outputs	
X	Y	u	w	SUM	COUT
0	0	1	0	0	0
0	1	1	1	1	0
1	0	1	1	1	0
1	1	0	1	0	1

[Here X=A,Y=B]

If the mentioned signal u is 1, the SUM equals X OR Y and the carryout COUT is 0, as depicted in figure 2.10(b) and table 1. If u is manageable and can be varied between 1 and 0, the carry propagation will be masked if its set to 1, and the SUM will equal X OR Y. In other words, the SUM (= X OR Y) might be thought of as an approximate total. A command signal, which is used to control u to be X NAND Y, or to be 1, can guarantee the choice between the exact and estimated sums. We add a control signal called MX and replace the 2-input NAND gate in the dashed frame with a 3-input NAND gate. The carry-maskable half adder (CMHA) is depicted in figure 2.11.(X=A,Y=B)

When MX=0 SUM = X OR Y (2.19)

COUT = 0 (2.20)

When MX = 1 SUM = X XOR Y (2.21)

COUT = X AND Y (2.22)

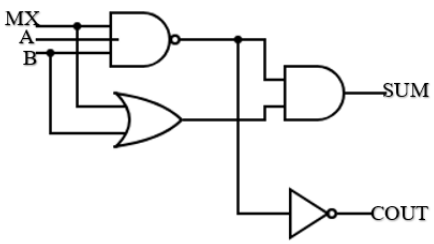


Figure 2.12 Gate level implementation of Carry Maskable Half Adder (CMHA)

A full adder, as shown in figure 2.12, requires similar considerations. When MX = 0 and Cin = 0, the SUM = X OR Y, and the carryout COUT = 0, switching actions are reduced, and dynamic energy per unit time utilization is lowered as well. Carry-maskable full adder (CMFA)[9] is the way of referring to this full adder.

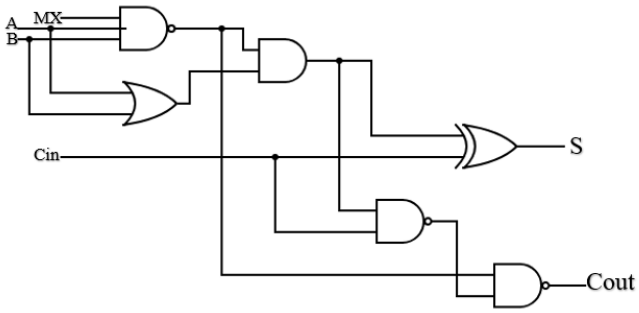


Figure 2.13: Gate level implementation of Carry Maskable Full Adder (CMFA)

An x-bit adder, which is realized using one CMHA and (x-1) CMFA, is called an x-bit CMA

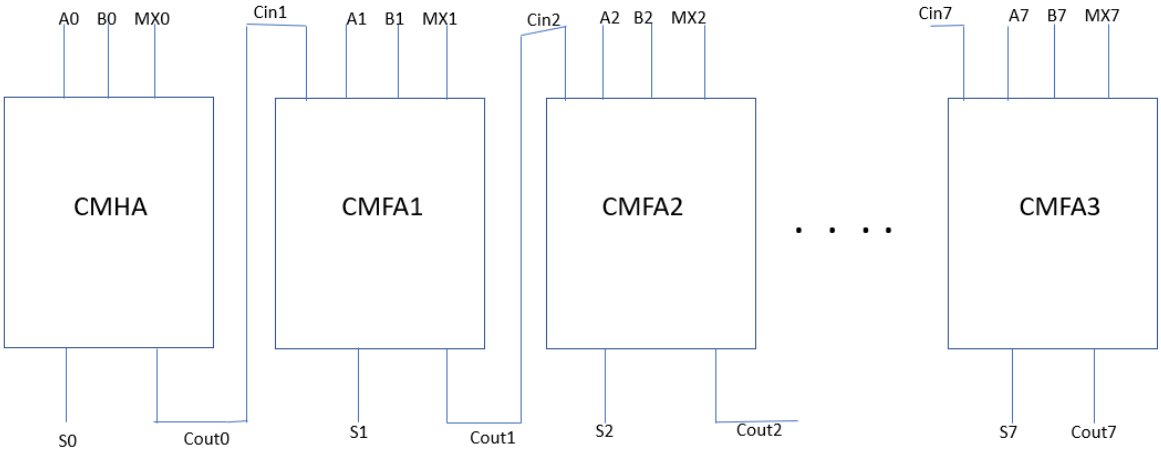


Figure 2.14: A Carry Maskable Adder of 8-bit

An example of an 8-bit CMA is shown in Figure 2.13 [9]. MX0, MX1, •••, MX7 are the eight bits that make up the carry-mask signal MX. The CMHA's sum and carry are S0 and Cout0, respectively. Cin1 and Cout0 are linked. When MX0 equals 0, S0 = A0 OR B0, and Cin1 = Cout0 = 0, Cin1 = Cout0 = 0. S0 = A0 OR B0, Cin1 = Cout0 = 0, S1 = A1 OR B1, and Cout1 = 0 when both MX1 and MX0 are 0. (Cin2 is also 0). As a result, the spread of CMHA to CMFA7 is hidden. It's worth noting that hiding the carry movement of a CMFA requires both MX and Cin to be 0.

In this chapter approximate adder designs in both gate level and transistor level are discussed and carry maskable adder is also discussed which is accuracy configurable adder. Gate level implementation of carry maskable adder is discussed.

CHAPTER 3

GDI BASED CARRY MASKABLE ADDER

In this chapter gate diffusion technique is reviewed and also the full swing gate diffusion input method which overcomes the drawbacks of basic GDI cells is discussed and in later sections of this chapter carry maskable adder is implemented with the help of this full swing Gate diffusion input technique.

3.1 Gate Diffusion Input Technique (GDI)

Gate diffusion input (GDI) is a reduced-power digital combinatorial circuit design method [10]. This method lowers the power exhaustion, propagation latency, and size of digital designs while keeping the logic architecture simple.

G (input of NMOS and PMOS to common Gate), P (input of PMOS on to the source/drain), and N (input of NMOS to the source/drain) are the three inputs to the GDI cell depicted in figure 3.1. In contrast to a CMOS inverter, the bulk of both NMOS and PMOS are connected to N or P (respectively), allowing it to be randomly driven.

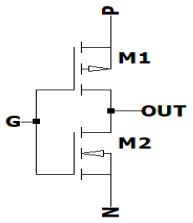


Figure 3.1 Basic GDI cell

Various Boolean functions can be realised by modifying the input settings of modified GDI cells, as shown in table 2. Many of these functions are difficult to implement in static CMOS, however with GDI, these functionalities can be obtained with only two transistors.

Not all of the functionalities are achieved in a typical p-well CMOS process, but they can be implemented completely with a success in twin-well CMOS or silicon on insulator (SOI) technology. To address this issue, a modified GDI Cell was developed, in which the majority of NMOS and PMOS are linked to ground and VDD, respectively. Figure 3.2 depicts the Modified Gate Diffusion Input (GDI) Cell [10].

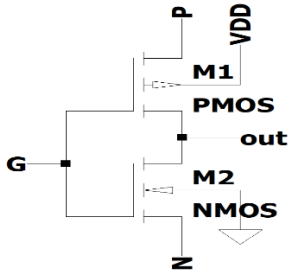


Figure 3.2 Modified GDI cell

Table 3.1. Functions achieved using Gate Diffusion Input Method

N	P	G	OUT	FUNCTION	T.C CMOS	T.C GDI
0	B	A	A'B	F1	8	2
B	1	A	A'+B	F2	8	2
1	B	A	A+B	OR	6	2
B	0	A	AB	AND	6	2
C	B	A	AC+A'B	MUX	6	2
0	1	A	A'	NOT	2	2

[T.C-Transistor Count]

When GDI cells are employed, the minimum gateway voltage reduces, which in turn lowers the current circulation and causes degradation of output swing. To address the voltage peak to peak reduction, the Swing restoration method [11] is implemented, in which a single transistor, named the Swing-Restoration (SR) transistor, is used in order to improve the output peak to peak of certain functions like F1 and F2 GDI cells. Figure 3.3 depicts the full-swing architecture of modified F1 and F2 [11] cells. Figure 28 shows an XOR gate implemented using the GDI approach [12] and with full swing.

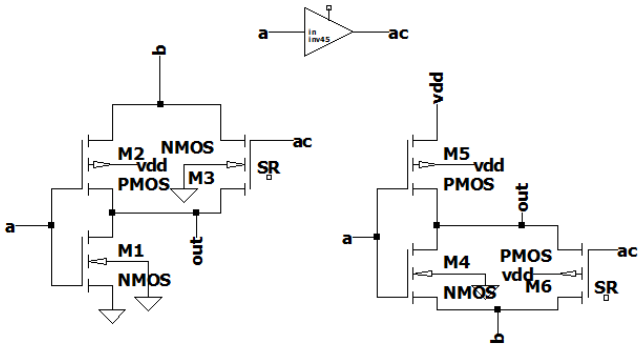


Figure 3.3 Architectures of F1 and F2 GDI cells with SR transistors

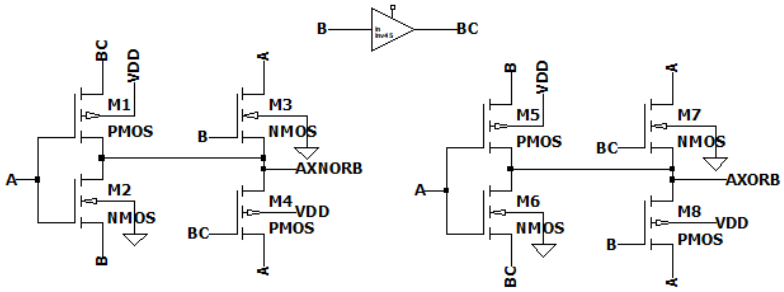


Figure 3.4. Modified XNOR and XOR gates with Full Swing using GDI [12]

Below Tables shows the truth tables and signal levels of different configurations when using basic GDI cell and Modified GDI cell and Full swing GDI Cell.

Table 3.2 Signal levels in execution of F1 = A'B

A	B	A'B	BASIC GDI	MODIFIED GDI	FULL SWING GDI
0	0	0	Weak(vtp)	Weak(vtp)	Strong
0	1	1	Strong	Strong	Strong
1	0	0	Strong	Strong	Strong
1	1	0	Strong	Strong	Strong

Table 3.3 Signal levels in execution of F2=A'+B

A	B	A'+B	BASIC GDI	MODIFIED GDI	FULL SWING GDI
0	0	1	Strong	Strong	Strong
0	1	1	Strong	Strong	Strong
1	0	0	Strong	Strong	Strong
1	1	1	Weak(VDD-Vtn)	Weak(VDD-Vtn)	Strong

Table 3.4 Signal levels in implementation of MUX=A'B+AC

A	B	C	A'B+AC	BASIC GDI	MODIFIED GDI	FULL SWING GDI
0	0	0	0	Weak(Vtp)	Weak(Vtp)	Strong
0	0	1	0	Weak(Vtp)	Weak(Vtp)	Strong
0	1	0	1	Strong	Strong	Strong
0	1	1	1	Strong	Strong	Strong
1	0	0	0	Strong	Strong	Strong
1	0	1	1	Weak(VDD-Vtn)	Weak(VDD-Vtn)	Strong
1	1	0	0	Strong	Strong	Strong
1	1	1	1	Weak(VDD-Vtn)	Weak(VDD-Vtn)	Strong

Table 3.5 Signal levels in execution of AND=A.B

A	B	AB	BASIC GDI	MODIFIED GDI
0	0	0	Weak(Vtp)	Weak(Vtp)
0	1	0	Weak(Vtp)	Weak(Vtp)
1	0	0	Strong	Strong
1	1	1	Weak(VDD-Vtn)	Weak(VDD-Vtn)

Table 3.6 Signal levels in implementation of OR= A+B

A	B	A+B	BASIC GDI	MODIFIED GDI
0	0	0	Weak(Vtp)	Weak(Vtp)
0	1	1	Strong	Strong
1	0	1	Weak(VDD-Vtn)	Weak(VDD-Vtn)
1	1	1	Weak(VDD-Vtn)	Weak(VDD-Vtn)

5.1 PROPOSED ACCURACY CONFIGURABLE ADDER USING CARRY MASK AND GDI

Full swing of F1 and F2 GDI cells modified and are used to create the basic logic gates in the suggested Carry Maskable Full Adder (CMFA), and the XOR gate is also realized utilising Full Swing-GDI technology.

With the use of this approach, the CMFA that was previously developed using 38 transistors in a conventional CMOS Process can now be realized by utilizing only 30 transistors. While the projected CMFA's gate level layout is identical to that of the CMFA (figure 2.12). Figure 3.5 depicts a block diagram of the proposed CMFA employing the full swing GDI approach.

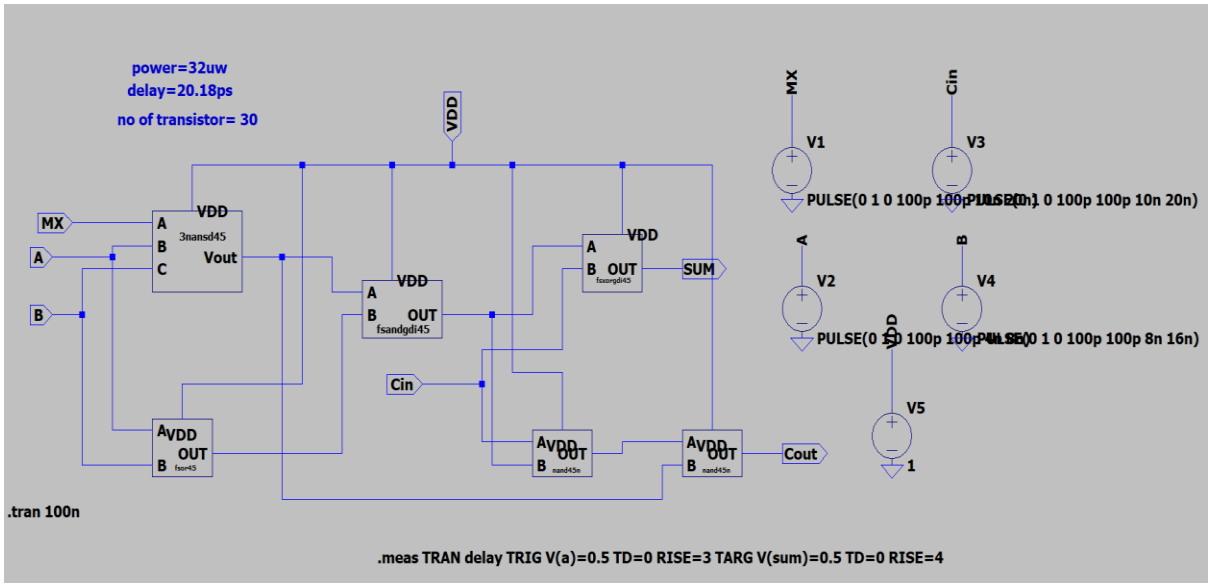


Figure 3.5 Block diagram of Full Swing GDI based Carry Maskable Full Adder(CMFA)

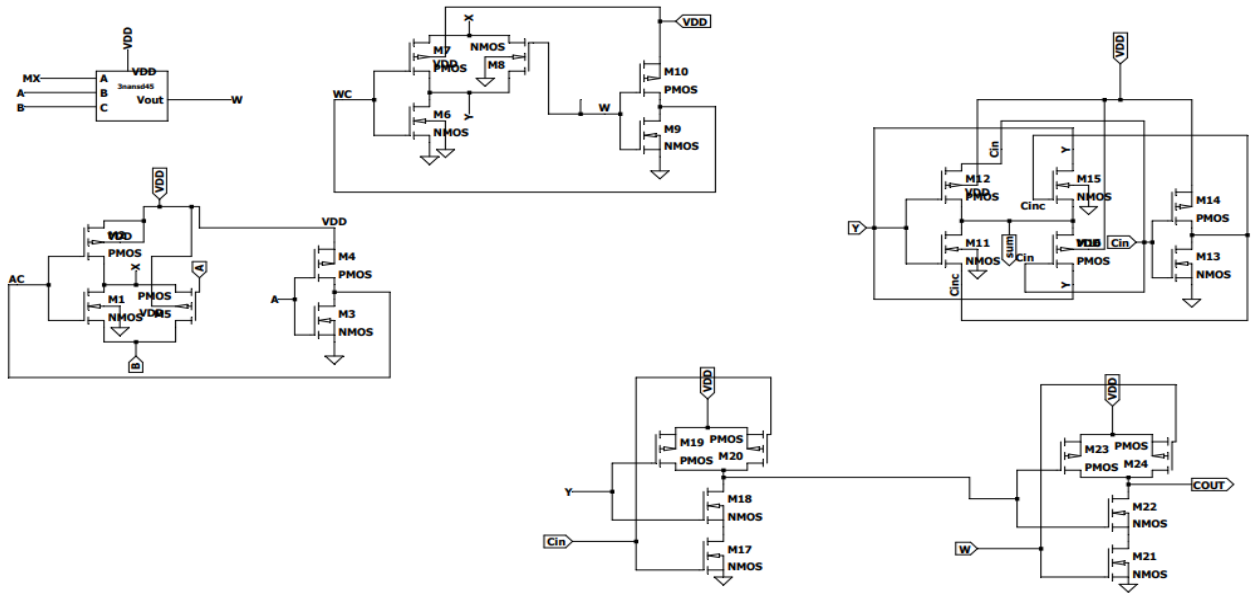


Figure 3.6 Transistor level Schematic of Full swing GDI based CMFA

CHAPTER 4

SIMULATION AND RESULTS

For a more detailed comparison, a CMOS-based carry maskable adder was simulated and compared to a planned Full swing GDI-based Carry Maskable Adder, and it was found that both adders performed correctly.

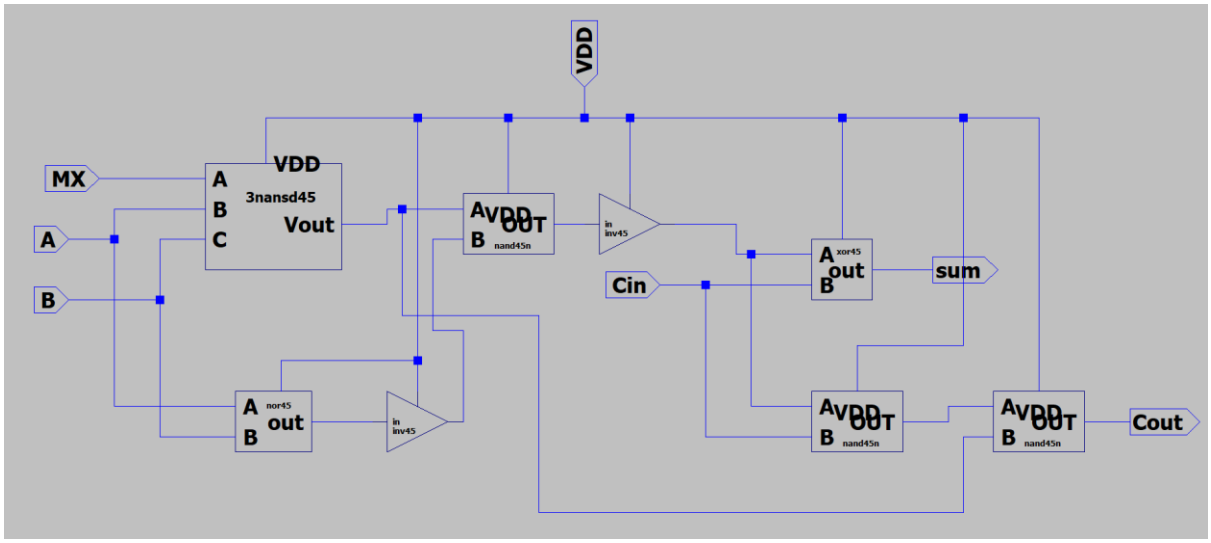


Figure 4.1. Block diagram CMOS based Carry Maskable Full adder

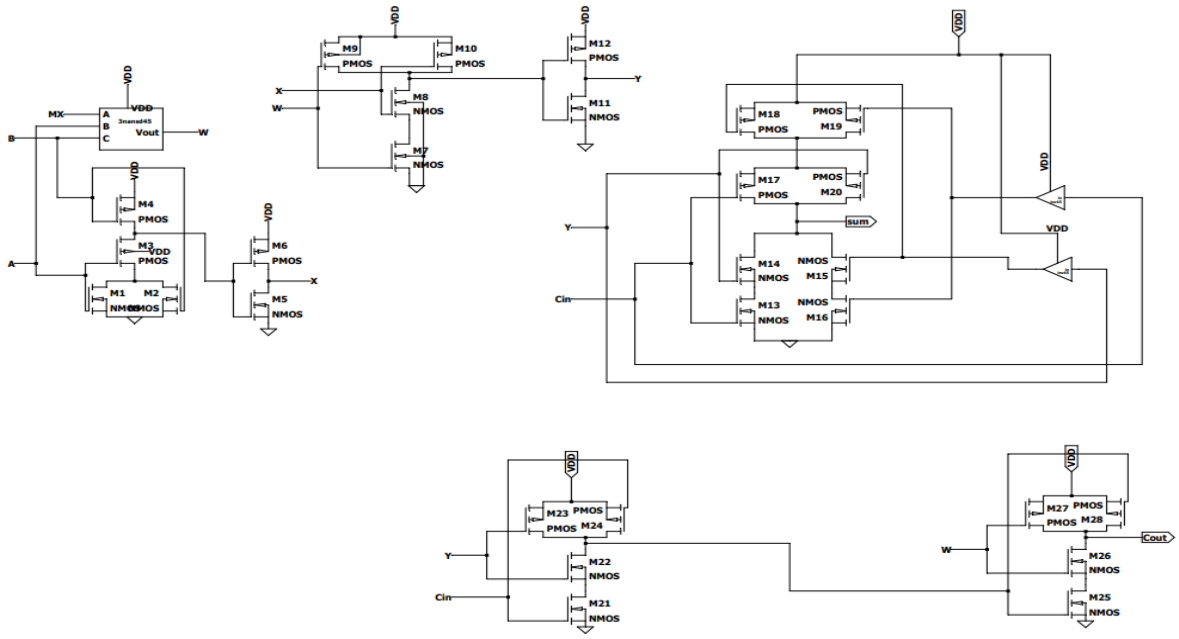


Figure 4.2. Transistor level schematic of CMOS based Carry Maskable Full Adder

These designs have been simulated out in LTSpiceXVII under 45nm process technology using 1V power supply.

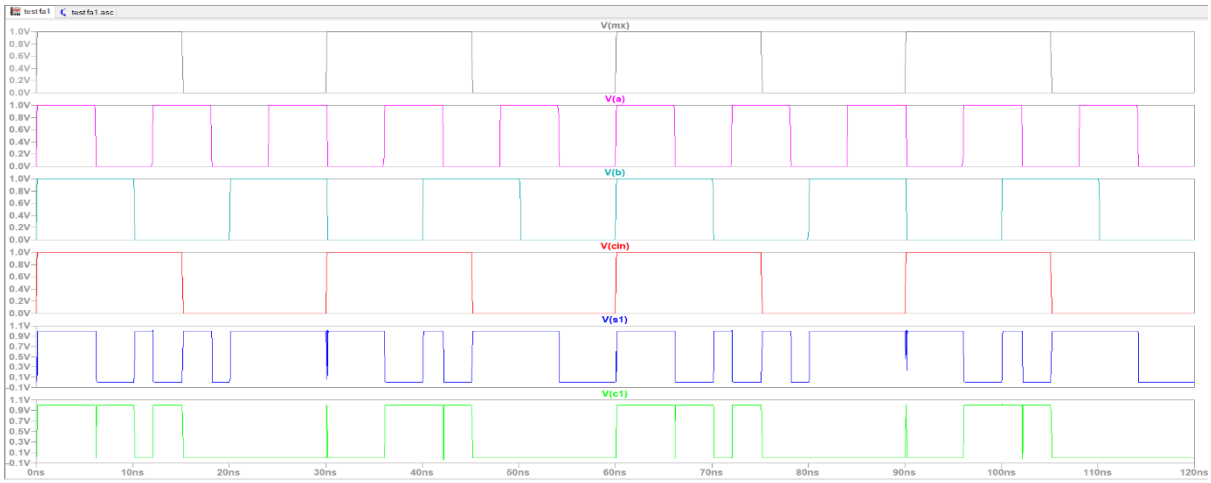


Figure 4.3. Transient analysis of CMOS based CMFA

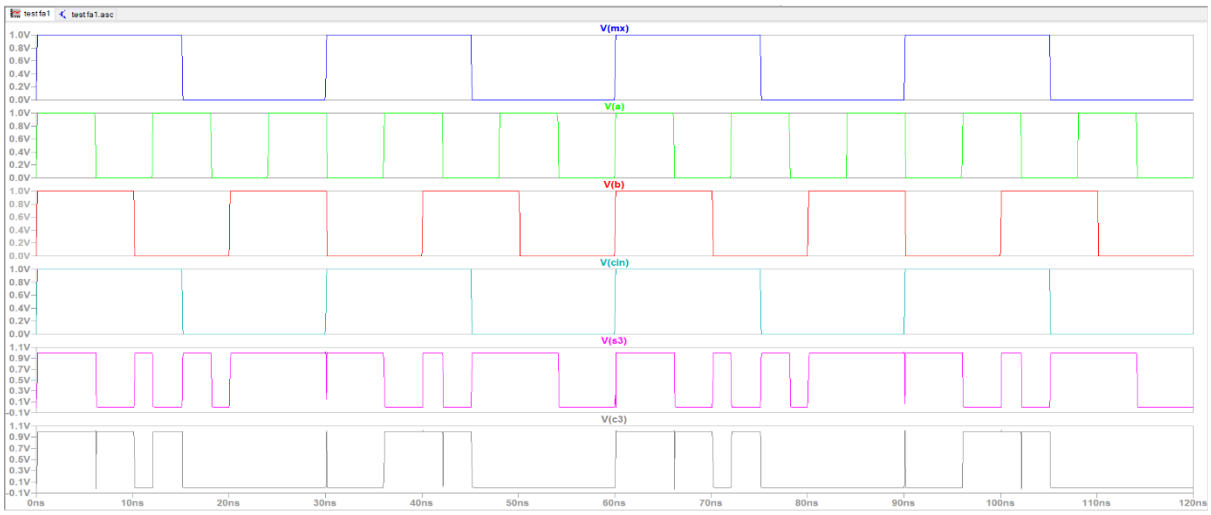


Figure 4.4. Transient analysis of Full swing GDI based CMFA

Energy consumption and time delay of both CMOS based and full swing GDI based carry maskable full adders are calculated and compared and it's observed that quite fair amount of reduction of power and delay in the Full swing GDI based CMFA. Power and delay comparison is given in table 4.1

Table 4.1 power and delay comparison

FA TYPE	Power(uW)	Delay(ps)	Number of Transistors
CMOS based CMFA	43.75	52.2	38
FS-GDI based CMFA	23.78	35.4	30

Using 1 bit full swing GDI based Carry maskable adders 4 bit adder is simulated and it's noted that it has given correct functionality with reduced number of transistors.

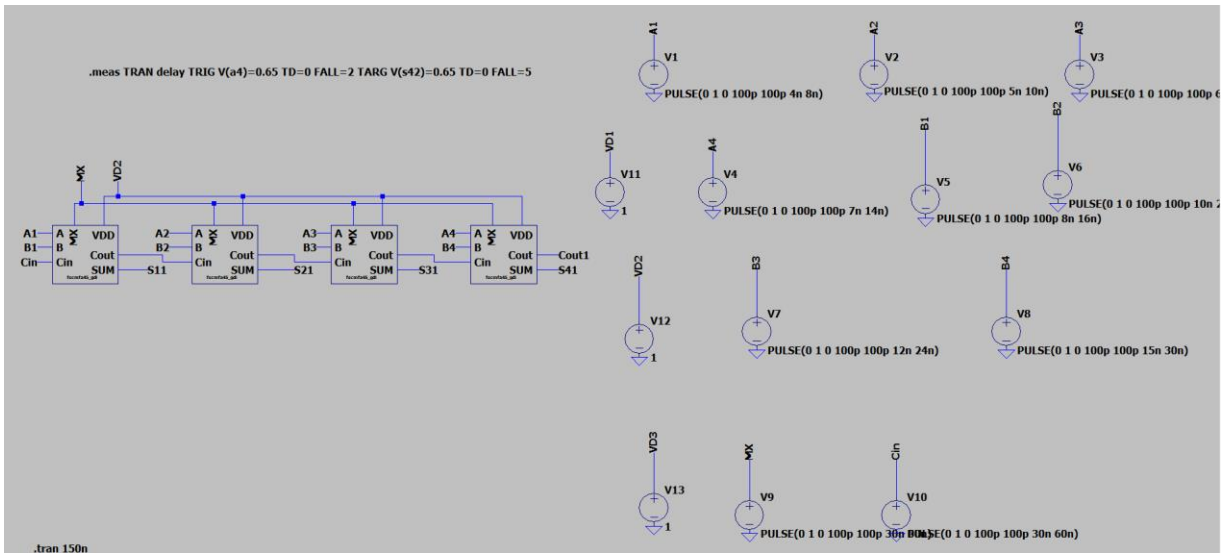


Figure 4.5: 4 bit Adder using Full Swing GDI based CMFA

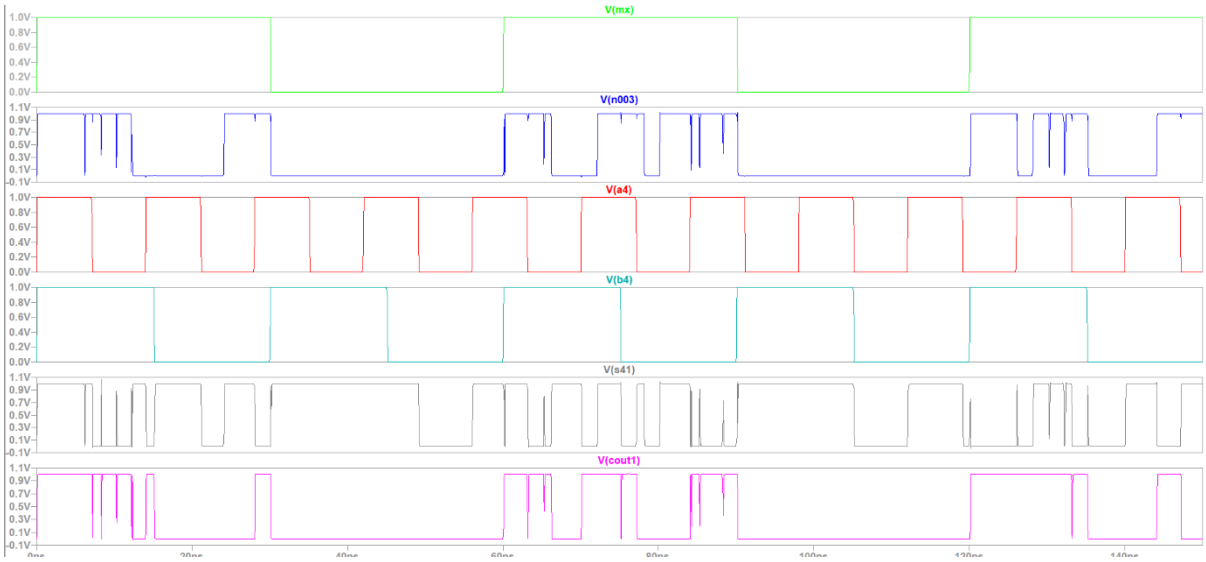


Figure 4.6 Transient analysis of 4bit adder using Full Swing GDI based CMFA

Its noted that all the adders simulated are given correct functionality in both CMOS based and Full swing GDI based and by using Full Swing GDI cells, same function is obtained with fewer transistors in comparison to the CMOS process. Therefore, the number of transistors has been reduced from 38 (in CMFA [9]) to 30 (Proposed CMFA), and so the energy consumption and delay of the proposed circuit design have also been reduced considerably.

CHAPTER 5

LOW POWER MULTIPLIER USING NEW CARRY MASKABLE ADDER AND FS-GDI TECHNIQUE

Multiplication is a fundamental function used in a wide range of applications. Approximate multiplication is a new and successful technique for balancing energy consumption, performance, and precision. In many digital signal processing applications, we employ a multiplier. Calculators, mobile phones, CPUs, and digital image processors are all designed using it.

The combinational circuit for the multiplier is designed using four AND gates and two half adders in Binary Multiplication. The multiplication will be done by the AND gates, and the partial product terms will be added by the half adders.

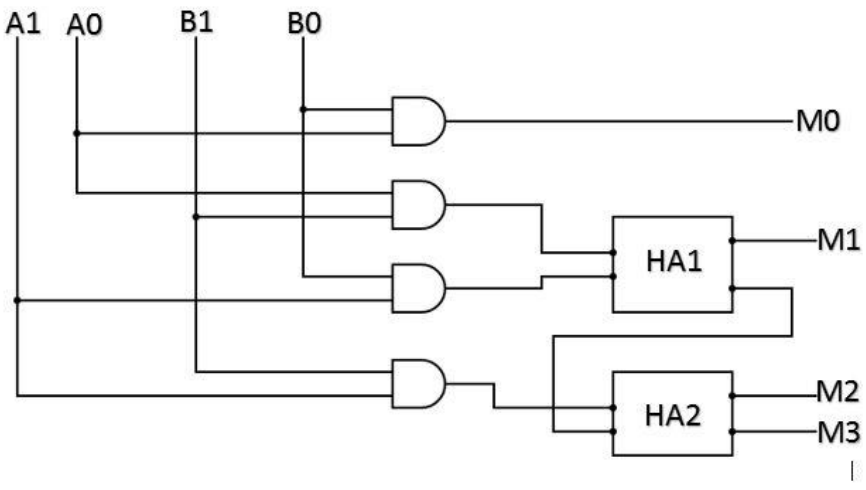


Figure 5.1: Block Diagram of two bit binary Multiplier

A low power Accuracy Configurable Multiplier is designed using the proposed Carry Maskable Adder and Full Swing GDI technique.In this multiplier the AND gates are replaced with Full Swing GDI AND gates and Adder circuits are replaced with the Proposed Carry Maskable Adder using Full Swing GDI technique. Both Conventional and Proposed Multilpier are designed on 45nm technology in LTSpiceXVII and comparisions are made

5.1 Simulation and Results:

For farer comparison CMOS based Multiplier is also simulated and compared with proposed low power accuracy configurable multiplier and its noted that both multipliers gave the correct functionality.

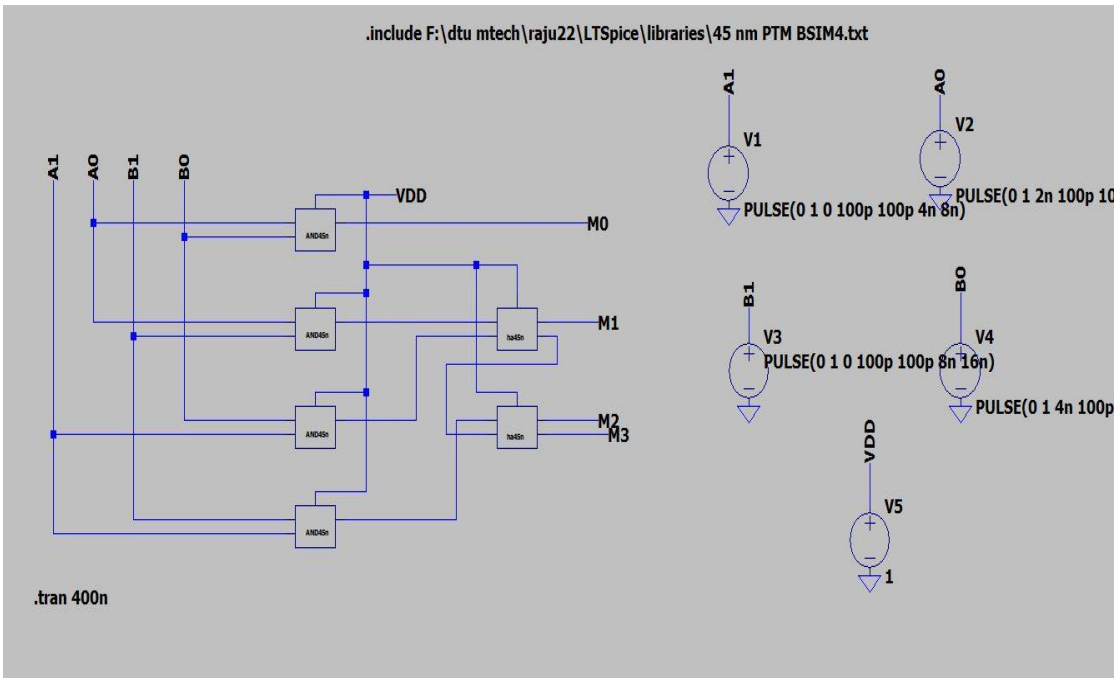


Figure5.2 Multiplier using basic CMOS Technology.

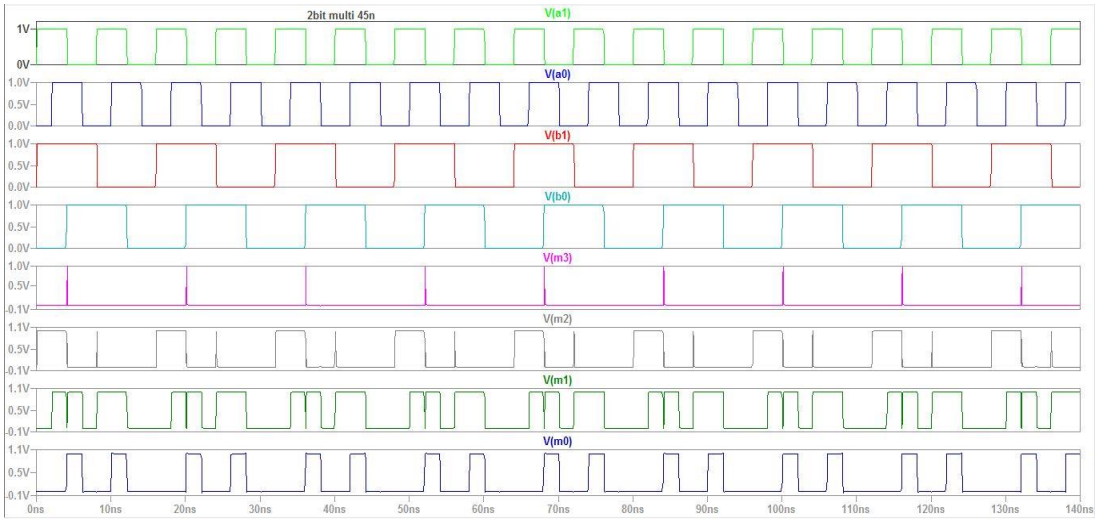


Figure 5.3: Transient Analysis of Multiplier using Basic CMOS

All the simulations have been carried out in 45-nm PTM process technology using LTSpice XVII for 1V power supply

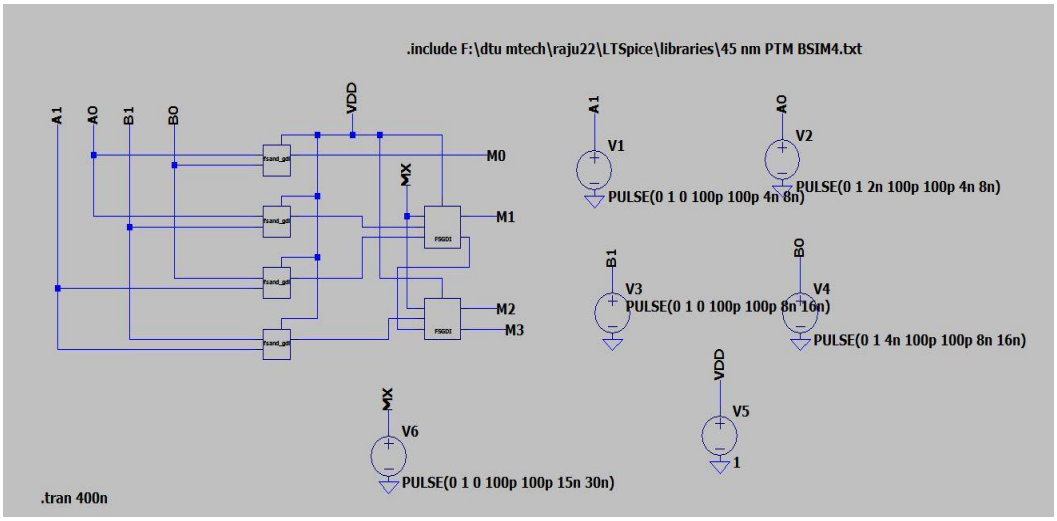


Figure 5.4: Multiplier using Proposed Carry Maskable Adder

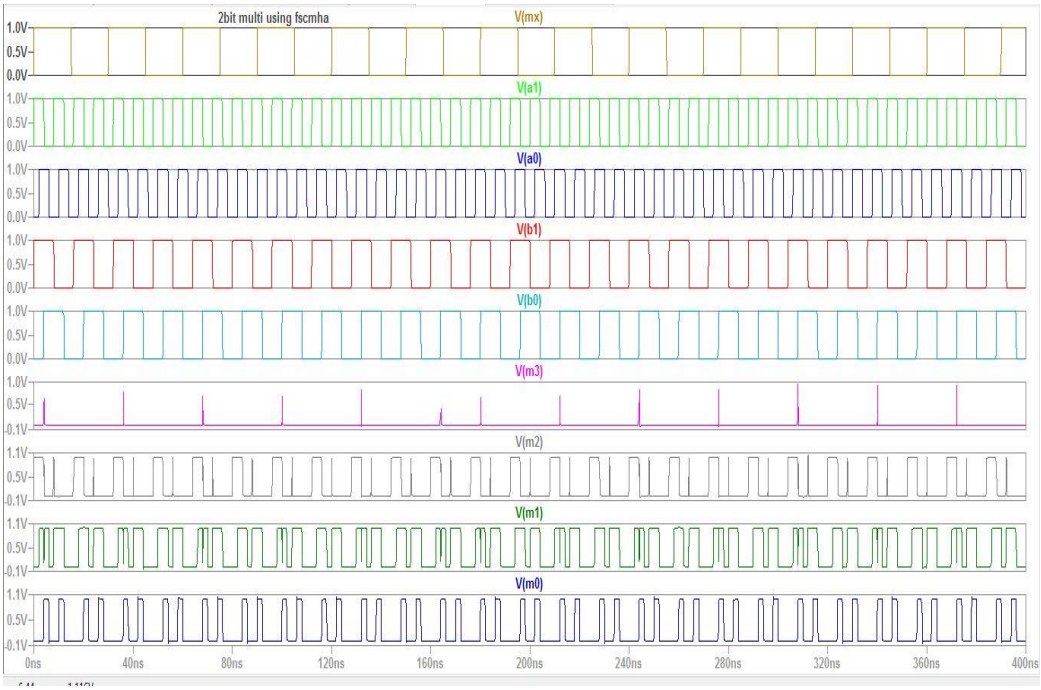


Figure 5.5: Transient Analysis of multiplier using Carry Maskable Adder.

Power and delay of both CMOS based and CMFA-full swing GDI based low power multiplier are calculated and compared and its observed that significant reduction of power and delay in the proposed multiplier Power and delay comparison is given in table 5.1

Table 5.1 power and delay comparison

FA TYPE	Power(uW)	Delay(ps)	Number of Transistors
CMOS based Multiplier	188.10	42.2	60
FS-GDI CMFA based Multiplier	99.08	25.4	54

It's noted that all the multipliers simulated are given correct functionality in both CMOS based and CMFA-Full swing GDI based and by using Full Swing GDI cells, same function is implemented with less count of transistors as compared to the CMOS process. So the power and delay of the proposed circuit have also been reduced considerably.

CHAPTER 6

CONCLUSION

In this research, a new circuit design for an Accuracy Configurable Adder using Carry Mask Technique and based on Full Swing Modified-GDI approach has been developed.

Using 45-nm technology, the improved developments of the proposed Carry Maskable Full Adder and the existing CMFA[7] were simulated and functionally verified in LTSpice. The results in Table 4.1 show that the proposed CMFA improves power and latency by an average of 37 percent and 32 percent, respectively.

And also 4 bit adders using this single bit Carry Maskable Adder and proposed CMFA using Full swing Modified GDI circuits are designed and simulated using LTSpice XVII under 45-nm PTM process technology for 1V power supply

.

Using the proposed Approximate Adder based on Full Swing Modified-GDI and Carry Maskable which is accuracy configurable a multiplier is designed and functionalities are verified and for the efficacies it is compared with basic CMOS Multiplier and its observed that improvement of power and delay by 51.2% and 37% respectively and its noted that these circuits are accuracy configurable it can be used as normal as well as approximate circuits depending on requirement of applications.

REFERENCES

[1] M. Pashaeifar, M. Kamal, and A. Afzali-Kusha, “Approximate Reverse Carry Propagate Adder for Energy-Ef□cient DSP Application,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 11, pp. 2530–2541, Nov. 2018.

[2] T. Moreau, A. Sampson, and L. Ceze, “Approximate computing: Making mobile systems more ef□cient,” IEEE Pervasive Comput., vol. 14, no. 2, pp. 9–13, Apr. 2015.

[3] H. A. F. Almurib, T. N. Kumar and F. Lombardi, "Inexact designs for approximate low power addition by cell replacement," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2016, pp. 660-665.

[4] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, “Low-power digital signal processing using approximate adders,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[4] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, “Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[5] Z. Yang, A. Jain, J. Liang, J. Han, and F. Lombardi, “Approximate XOR/XNOR-based adders for inexact computing,” in Proc. 13th IEEE Int. Conf. Nanotechnol. (NANO), pp. 690–693, Aug. 2013.

[6] Z. Yang, J. Han, and F. Lombardi, “Transmission gate-based approximate adders for inexact computing,” in Proc. IEEE/ACM Int. Symp. Nanosc. Archit. (NANOARCH), Jul. 2015, pp. 145–150.

[7] B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," DAC Design Automation Conference 2012, San Francisco, CA, 2012, pp. 820-825, doi: 10.1145/2228360.2228509.

[8] R. Ye, T. Wang, F. Yuan, R. Kumar and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," 2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, 2013, pp. 48-54, doi: 10.1109/ICCAD.2013.6691096.

[9] T. Yang, T. Ukezono and T. Sato, "A low-power configurable adder for approximate applications," 2018 19th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, 2018, pp. 347-352, doi: 10.1109/ISQED.2018.8357311.

[10] A. Morgenshtein, A. Fish, and I. Wagner, “Gate-diffusion input (GDI): a power-efficient method for digital combinatorial circuits,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems IEEE Trans. VLSI Syst., vol. 10, no. 5, pp. 566–581, 2002.

[11] Morgenshtein, A., Yuzhaninov, V., Kovshilovsky, A., & Fish, A. (2014). Full-Swing Gate Diffusion Input logic—Case-study of low-power CLA adder design. Integration, the VLSI Journal, 47(1), 62–70. doi:10.1016/j.vlsi.2013.04.002

[12] O. A. Albadry, M. A. Mohamed El-Bendary, F. Z. Amer and S. M. Singy, "Design Of Area Efficient And Low Power 4-Bit Multiplier Based On Full- swing GDI technique," 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), Aswan, Egypt, 2019, pp. 328-333, doi: 10.1109/ITCE.2019.8646341.