Comparative Analysis of Different CMOS Full Adder Using Cadence in 90 nm CMOS Process Technology

A thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

In

Electrical, Electronic and Communication Engineering

Submitted by

Meem Shahrin	**Student ID: 201416019**
Zarin Tabassum	**Student ID: 201416040**
Aniqa Ibnat Lim	**Student ID: 201416053**

Supervised By

Md. Tawfiq Amin, PhD, EME

Instructor Class B

Department of Electrical, Electronic and Communication Engineering (EECE)
Military Institute of Science and Technology
Mirpur Cantonment, Dhaka-1216

December, 2017

CERTIFICATION

The thesis titled “Comparative Analysis of Different CMOS Full Adder Using Cadence in 90 nm CMOS Process Technology” submitted by Meem Shahrin (201416019), Zarin Tabassum (201416040) and Aniqa Ibnat Lim (201416053) session 2013-2014 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical, Electronic and Communication Engineering on December 2017.

Approval of Supervisor

……………………………………

Md. Tawfiq Amin, PhD, EME
Instructor Class B
Department of Electrical, Electronic and Communication Engineering (EECE)
Military Institute of Science and Technology (MIST)
Mirpur Cantonment, Dhaka-1216

DECLARATION

We hereby declare that the thesis titled “Comparative Analysis of Different CMOS Full Adder Using Cadence in 90 nm CMOS Process Technology” is submitted to the Department of Electrical, Electronic and Communication Engineering for the partial fulfillment of the requirement for Bachelor of Science Degree on Electrical, Electronic and Communication Engineering.

We hereby declare that this thesis is our original work under the supervision of Md. Tawfiq Amin, PhD, EME and it has been written by us in its entity. We have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Authors

…………………………………….
Meem Shahrin
Student ID: 201416019

…………………………………….
Zarin Tabassum
Student ID: 201416040

…………………………………….
Aniqa Ibnat Lim
Student ID: 201416053

ACKNOWLEDGEMENT

First and foremost we would like to give our heartfelt gratitude to our parents who have been a constant pillar of support for us throughout university. This would not be possible without their encouragement and motivation.

To Md. Tawfiq Amin, PhD, EME, Instructor Class B, Department of Electrical, Electronic and Communication Engineering (EECE), Military Institute of Science and Technology (MIST), for his supervision in our understanding of beneficial advice in the digital design and designing of various adders, helping us in performance analysis, the proof readings of our rough drafts as well as for helping us in our academics. We would not have been able to do this without his effort.

We would like to extend our regards to them who directly and indirectly helped us during this journey.

ABSTRACT

This thesis paper comprises the study and performance analysis of four full adders with various topologies. Propagation delay, power dissipation and power delay product are the parameters to measure the performance of full adders. The overall performance can be improved by reducing power and delay which is done by optimizing the transistor size. For meeting the demands of tech-savvy fast progressing world of electronics, the sole focus for adder architecture is on making it more efficient.

The design and simulation is carried out for four full adders (Conventional, transmission gate, 14T & GDI based technique) for transistor count, power dissipation, delay and power delay products. It is performed in virtuoso platform, using Cadence tool with available GPDK –90nm kit having MOS Capacitance value of 50 femto farad. The width of NMOS and PMOS is 120nm and 240nm respectively. Delay estimation is obtained by taking the average propagation delay. Transmission gate full adder has sheer advantage of high speed but consumes more power. On the other hand, GDI full adder gives reduced voltage swing not being able to pass logic 1 and logic 0 completely showing degraded output. Transmission gate full adder shows better performance in terms of delay, whereas 14T full adder shows better performance considering overall factors.

CONTENTS

List of Figures

List of Tables

Chapter 01

INTRODUCTION

1.1 Introduction to Digital Design

Digital electronics or digital circuits are electronics that operate on digital signals. In contrast, analog circuits manipulate analog signals whose performance is more subject to manufacturing tolerance, signal attenuation and noise. Digital techniques are helpful because it is a lot easier to get an electronic device to switch into one of a number of known states than to accurately reproduce a continuous range of values. Digital electronic circuits are usually made from large assemblies of logic gates (often printed on integrated circuits), simple electronic representations of Boolean logic functions. In the early days of simple integrated circuits, the technology's large scale limited each chip to only a few transistors, and the low degree of integration meant the design process was relatively simple. Manufacturing yields were also quite low by today's standards. As the technology progressed, millions, then billions of transistors could be placed on one chip, and good designs required thorough planning, giving rise to new design methods. An advantage of digital circuits when compared to analog circuits is that signals represented digitally can be transmitted without degradation due to noise. For example, a continuous audio signal transmitted as a sequence of 1s and 0s, can be reconstructed without error, provided the noise picked up in transmission is not enough to prevent identification of the 1s and 0s[1].

A digital circuit is typically constructed from small electronic circuits called logic gates that can be used to create combinational logic. Each logic gate is designed to perform a function of Boolean logic when acting on logic signals. A logic gate is generally created from one or more electrically controlled switches, usually transistors but thermionic valves have seen historic use. The output of a logic gate can, in turn, control or feed into more logic gates. Integrated circuits consist of multiple transistors on one silicon chip, and are the least expensive way to make large number of interconnected logic gates. Integrated circuits are usually interconnected on a printed circuit board which is a board which holds electrical components, and connects them together with copper traces. Each logic symbol is represented by a different shape. The actual set of shapes was introduced in 1984 under IEEE/ANSI standard 91-1984.

Design started with relays. Relay logic was relatively inexpensive and reliable, but slow. Occasionally a mechanical failure would occur. Fan outs were typically about 10, limited by the resistance of the coils and arcing on the contacts from high voltages. Later, vacuum tubes were used. These were very fast, but generated heat, and were unreliable because the filaments would burn out. The first semiconductor logic family was resistor–transistor logic. This was a thousand times more reliable than tubes, ran cooler, and used less power, but had a very low fan-in of 3. Diode–transistor logic improved the fan out up to about 7, and reduced the power. Some DTL designs used two power-supplies with alternating layers of NPN and PNP transistors to increase the fan out. Transistor–transistor logic (TTL) was a great improvement over these. In early devices, fan out improved to 10, and later variations reliably achieved 20. TTL was also fast, with some variations achieving switching times as low as 20 ns. TTL is still used in some designs. Emitter coupled logic is very fast but uses a lot of power. It was extensively used for high-performance computers made up of many medium-scale components (such as the Iliac IV).

By far, the most common digital integrated circuits built today use CMOS logic, which is fast, offers high circuit density and low-power per gate. This is used even in large, fast computers, such as the IBM System z.

Over the past several years, Silicon CMOS technology has become the dominant fabrication process for relatively high performance and cost effective VLSI circuits. The revolutionary nature of these developments is understood by the rapid growth in which the number of transistors integrated on circuit on single chip. Very-large-scale integration (VLSI) is the process of creating an integrated circuit (IC) by combining thousands of transistors into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. Before the introduction of VLSI technology, most ICs had a limited set of functions they could perform. An electronic circuit might consist of a CPU, ROM, RAM and other glue logic. VLSI lets IC designers add all of these into one chip.

The electronics industry has achieved a phenomenal growth over the last few decades, mainly due to the rapid advances in large scale integration technologies and system design applications. With the advent of very large scale integration (VLSI) designs, the number of applications of integrated circuits (ICs) in high-performance computing, controls, telecommunications, image and video processing, and consumer electronics has been rising at a very fast pace

1.2 ALU in Digital Communication

In Digital Communication, every single process is made faster and more efficient since IC chips are used. With the growing technologies, Integrated Circuits are becoming smaller in size having high efficiency. One of the major problems faced in electronic circuit designing is the power dissipation. So the need for low power circuits and devices are growing day by day. And, ALU comprising different operations including adding operation is one of the prime factors for determining the performance analysis of the IC.

An arithmetic and logic unit (ALU) is a digital circuit that performs arithmetic and logical operations. The ALU is the basic building block of the CPU of a computer, and even the simplest processors contain one ALU for purposes such as maintaining timers. It is a combinational digital electronic circuit that performs arithmetic and bitwise operations on integer binary numbers. This is in contrast to a floating-point unit (FPU), which operates on floating point numbers. An ALU is a fundamental building block of many types of computing circuits, including the central processing unit (CPU) of computers, FPUs, and graphics processing units (GPUs). A single CPU, FPU or GPU may contain multiple ALUs.

The inputs to an ALU are the data to be operated on, called operands, and a code indicating the operation to be performed; the ALU's output is the result of the performed operation. In many designs, the ALU also has status inputs or outputs, or both, which convey information about a previous operation or the current operation, respectively, between the ALU and external status registers [2].

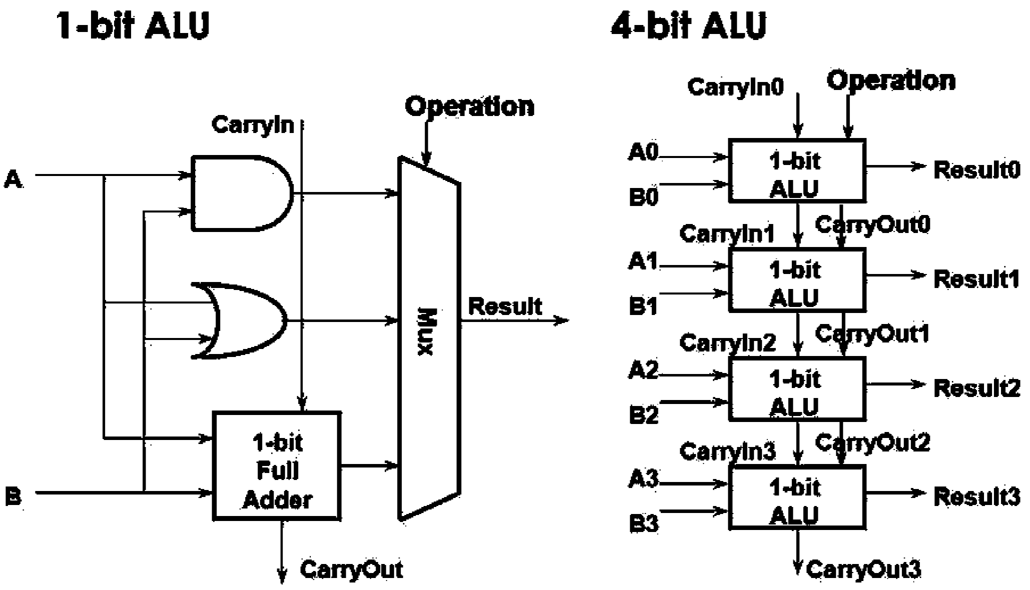


Figure 1.1 Use of Full Adder in the construction of ALU.

1.2.1 Arithmetic operations:

Add: A and B are summed and the sum appears at Y and carry-out.

Add with carry: A, B and carry-in are summed and the sum appears at Y and carry-out.

Subtract: B is subtracted from A (or vice versa) and the difference appears at Y and carry-out. For this function, carry-out is effectively a "borrow" indicator. This operation may also be used to compare the magnitudes of A and B; in such cases the Y output may be ignored by the processor, which is only interested in the status bits (particularly zero and negative) that result from the operation.

Subtract with borrow: B is subtracted from A (or vice versa) with borrow (carry-in) and the difference appears at Y and carry-out (borrow out).

Two's complement (negate): A (or B) is subtracted from zero and the difference appears at Y.

Increment: A (or B) is increased by one and the resulting value appears at Y.

Decrement: A (or B) is decreased by one and the resulting value appears at Y.

Pass through: all bits of A (or B) appear unmodified at Y. This operation is typically used to determine the parity of the operand or whether it is zero or negative, or to load the operand into a processor register [3].

1.3 CMOS (Complementary Metal Oxide Semiconductor) technology

Complementary MOSFET (CMOS) technology is widely used today to form circuits in numerous and varied applications. Today's computers, CPUs and cell phones make use of CMOS due to several key advantages. The main advantage of CMOS over NMOS and BIPOLAR technology is the much smaller power dissipation. Unlike NMOS or BIPOLAR circuits, a Complementary MOS circuit has almost no static power dissipation. Power is only dissipated in case the circuit actually switches. This allows integrating more CMOS gates on an IC than in NMOS or bipolar technology, resulting in much better performance. CMOS consists of P-channel MOS (PMOS) and N-channel MOS (NMOS). CMOS offers low power dissipation, relatively high speed, high noise margins in both states and will operate over a wide range of source and input voltages (provided the source voltage is fixed).

- **NMOS**

  NMOS is built on a p-type substrate with n-type source and drain diffused on it. In NMOS, the majority carriers are electrons. When a high voltage is applied to the gate, the NMOS will conduct. Similarly, when a low voltage is applied to the gate, NMOS will not conduct. NMOS are considered to be faster than PMOS, since the carriers in NMOS, which are electrons, travel twice as fast as the holes.

- **PMOS**

  P-channel MOSFET consists P-type Source and Drain diffused on an N-type substrate. Majority carriers are holes. When a high voltage is applied to the gate, the PMOS will not conduct. When a low voltage is applied to the gate, the PMOS will conduct. The PMOS devices are more immune to noise than NMOS devices.

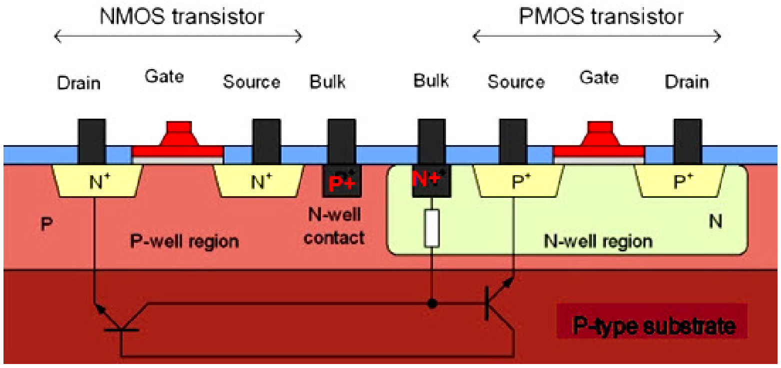


Figure1.2 CMOS transistor

It is time the well-engineered deep submicron CMOS technologies to address the challenging criteria of these emerging low-power and high-speed communication digital signal processing chips is explored. The performance of many applications as digital signal processing depends upon the performance of the arithmetic circuits to execute complex algorithms such as convolution, correlation, and digital filtering. Fast arithmetic computation cells including adders and multipliers are the most frequently and widely used circuits in very-large-scale integration (VLSI) systems [3].

1.4 Motivation

With the advancement of nanotechnology, there has been a rapid progress in the world of electronics and digital signal processing due to which there is an increasing demand for more efficient, low power consumption high performance devices ensuring high reliability and sustainability. The semiconductor industry has witnessed an explosive growth of integration of sophisticated multimedia-based applications into mobile electronics gadgetry since the last decade.

Adder being one of the core elements of complex arithmetic circuits is very important in electronics. Moreover, researchers in the past used standard implementations with various logic styles to design full-adder cells. Although they all have similar function, the way of producing the intermediate nodes and the transistor count is varied. In order to keep pace with the growing demand for nano chips, it is absolutely important to analyze and compare the different logic styles to identify, which adder shows the best performance with greater reliability promising to give significant output, which in turn can be implemented becoming highly impactful in the world of electronics.

1.5 Thesis Organization

This whole thesis paper has five chapters which is organized in the following manner:

Chapter 01 provides the knowledge of adder introduction and how it is designed based on CMOS technology. It also contains the necessity of adders in the world of electronics.

Chapter 02 illustrates different types of adders and their application. It comprises the classification of adders with the brief description of each type.

Chapter 03 represents the elaborate description of Different logic styles along with their architecture including advantages and disadvantages.

Chapter 04 shows the design and performance analysis of various types of Adders. Transient analysis, DC analysis including power and delay for each type of adder is depicted here.

Chapter 05 depicts the conclusion which holds the summary of whole thesis paper.

Chapter 02

ADDER BASICS

2.1 General Idea

An adder is a digital circuit that performs addition of numbers. In many computers and other kinds of processors adders are used in the arithmetic logic units or ALU. They are also utilized in other parts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators, and similar operations.

Although adders can be constructed for many number representations, such as binary-coded decimal or excess-3, the most common adders operate on binary numbers. In cases where two's complement or ones' complement is being used to represent negative numbers, it is trivial to modify an adder into an adder–subtractor. Other signed number representations require more logic around the basic adder.

An adder can be classified to two groups namely half adder and full adder. The half adder adds two binary digits called as augend and addend and produces two outputs as sum and carry; XOR is applied to both inputs to produce sum and AND gate is applied to both inputs to produce carry. The full adder adds 3 one bit numbers, where two can be referred to as operands and one can be referred to as bit carried in. And produces 2-bit output, and these can be referred to as output carry and sum [4].

2.1.1 Application:

1. Adders are critically important elements in processor chips.
2. They are used in floating-point arithmetic units, ALUs, memory addressing, program
3. Booth Multipliers, ALU Designing, multimedia.
4. Communication systems, Real-time signal processing like audio signal processing, counter updating, video/image processing, or large capacity data processing etc.
5. Important tool in DSP (digital signal processing).
6. For graphics related applications, where there is need of complex computations, the GPU uses optimized ALU which is made up of full adders, other circuits as well.

2.2 Half Adder

A half adder is a type of adder, an electronic circuit that performs the addition of numbers. The half adder is able to add two single binary digits and provide the output plus a carry value. It has two inputs, called A and B, and two outputs S (sum) and C (carry). The common representation uses a XOR logic gate and an AND logic gate. A half adder is used to add two single-digit binary numbers and results into a two-digit output. Half adder is the simplest of all adder circuit, but it has a major disadvantage. The half adder can add only two input bits (A and B) and has nothing to do with the carry if there is any in the input. So if the input to a half adder have a carry, then it will be neglected it and adds only the A and B bits. That means the binary addition process is not complete and that's why it is called a half adder. It is named as such because putting two half adders together with the use of an OR gate results in a full adder. In other words, it only does half the work of a full adder. The truth table of half adder is given below.

Table 2.1: Truth table of half adder

A **(INPUT)**	**B** **(INPUT)**	**C** **(OUTPUT)**	**S** **(OUTPUT)**
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

The simplified Boolean function of half adder for the two outputs can be obtained directly from the truth table. The expressions for sum (S) and carry (c) are-

$$S = A'B + AB' \qquad (2.1)$$

$$C = AB \qquad (2.2)$$

The logic diagram of this implementation using XOR and AND gate is shown below in figure 2.1.

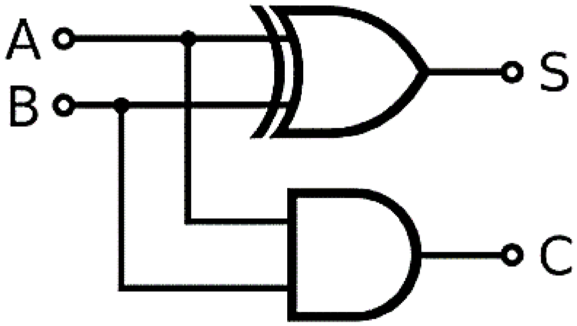


Fig 2.1 Block diagram of half adder

The adder works by combining the operations of basic logic gates, with the simplest form using only an XOR and an AND gate. This can also be converted into a circuit that only has AND, OR and NOT gates. This is especially useful since these three simpler logic gate ICs (integrated circuits) are more common and available than the XOR IC, though this might result in a bigger circuit since three different chips are used instead of just one.Now it has been cleared that 1-bit adder can be easily implemented with the help of the XOR Gate for the output 'SUM' and an AND Gate for the 'Carry'. When we need to add, two 8-bit bytes together, we can be done with the help of a full-adder logic [4].

2.2.1 Application:

1. The ALU (arithmetic logic circuitry) of a computer uses half adder to compute the binary addition operation on two bits.
2. Half adder is used to make full adder as a full adder requires 3 inputs, the third input being an input carry i.e. we will be able to cascade the carry bit from one adder to the other.
3. Ripple carry adder is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is called RIPPLE CARRY ADDER, since each carry bit "ripples" to the next full adder. The first full adder (and only the first) may be replaced by a half adder.

2.3 Full Adder

Full adder is difficult to implement than a half-adder. The difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs, whereas half adder has only two inputs and two outputs. It is a combinational circuit that forms the arithmetic sum of three input bits. The first two inputs are A and B represent the bits to be added. The third input Cin represents the carry from the previous lower significant position. The two outputs are designated by the symbols S for sum and C for carry. The binary variable S gives the value of the least significant bit of the sum. The binary variable C gives the carry. With the truth-table, the full adder logic can be implemented [5].

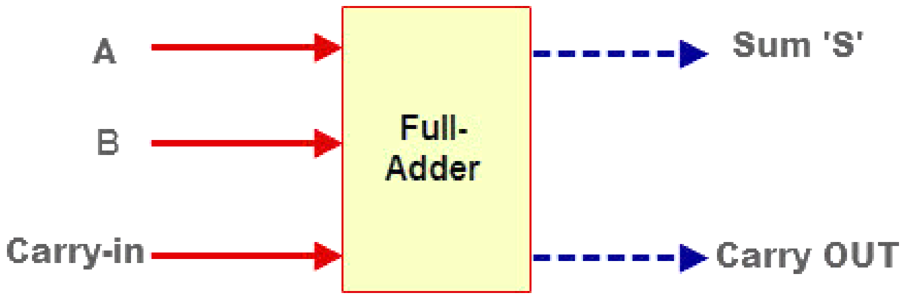


Figure 2.2 Block diagram of full adder

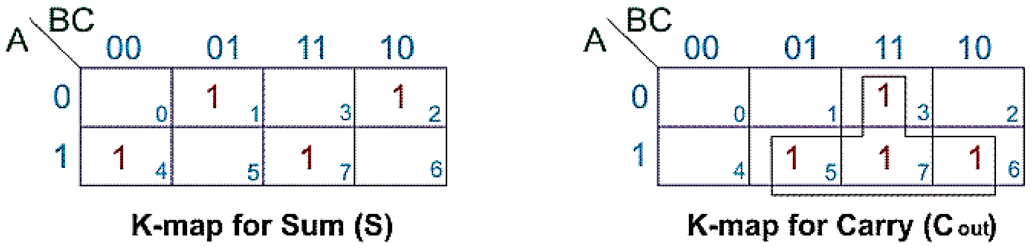
Table 2.2: Truth table of full adder

Input			Output	
A	B	Cin	S (Sum)	C (Carry)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	1	1	1

The input output logical relationship for full adder can be expressed by two Boolean functions, one for each output variable. Each output Boolean function requires a unique map for its simplification. It is called Karnaugh map given in figure 2.3. The 1's in the squares for the map of sum and carry are determined directly from the truth table of full adder. The logic diagram for the full adder implemented in sum of products is given in figure 2.4. The implementation uses following Boolean expression [4] [5].

$$S = A'B'C + A'BC' + AB'C' + ABC$$

$$C = AB + BC + CA$$



A \ BC	00	01	11	10
0	$_{0}$	$1\ _{1}$	$_{3}$	$1\ _{2}$
1	$1\ _{4}$	$_{5}$	$1\ _{7}$	$_{6}$

K-map for Sum (S)

A \ BC	00	01	11	10
0	$_{0}$	$_{1}$	$1\ _{3}$	$_{2}$
1	$_{4}$	$1\ _{5}$	$1\ _{7}$	$1\ _{6}$

K-map for Carry (C_{out})

Figure 2.3 Karnaugh maps for full adder

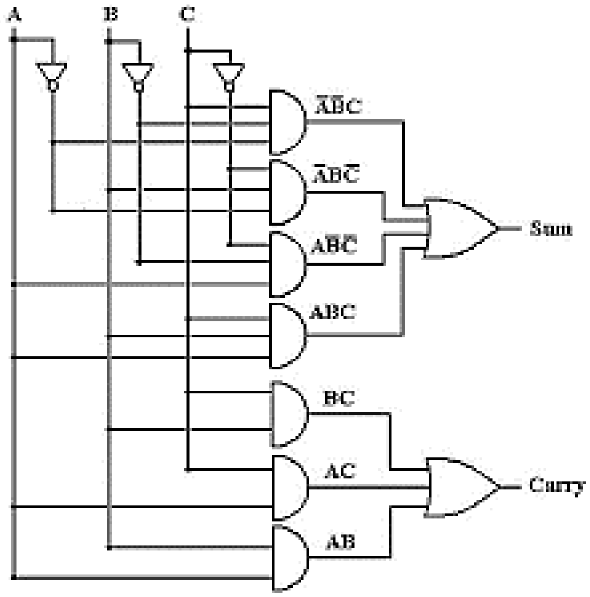


Figure 2.4 Implementation of full adder in sum of products

Other configuration of full adder can be developed by manipulating the equation. Generally the equation of full adder is used as below and the figure 2.4 is normally used as logic diagram.

$$S = A \oplus \mathrm{B} \oplus \mathrm{C}$$

$$C = AB + Cin(A \oplus \mathrm{B})$$

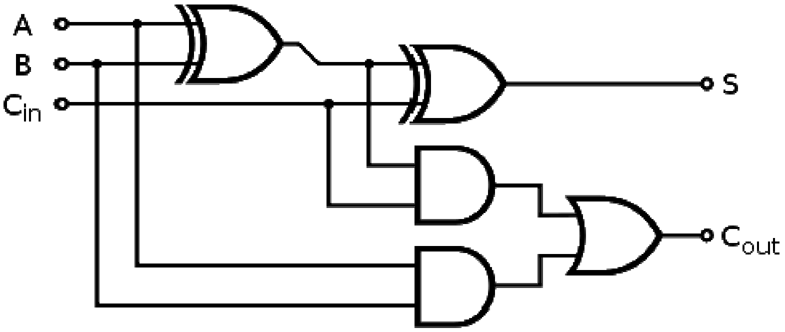


Figure 2.5 Logic diagram of full adder

A full adder can be implemented with the help of two half adder circuits [4]. At first, a half adder will be used to add A and B to produce a partial Sum S1 and a second half adder logic can be used to add Cin to the Sum produced by the first half adder to get the final S and C output.

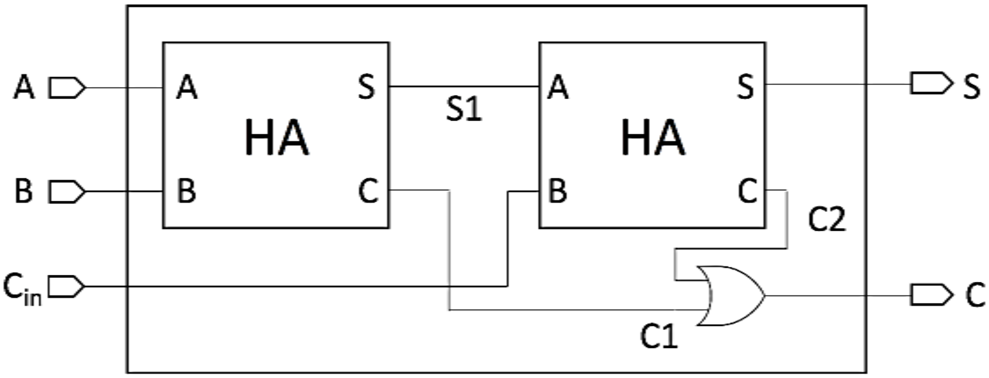


Figure 2.6 Block diagram of full adder using two half adder

2.4 Various Types of Adders

Different adder topologies together with versatile hardware synthesis are rudiments for a high productivity in integrated circuit designs. Many different adder architectures for speeding up binary addition have been studied and proposed over the last decades. This section presents the design of adder topologies. Ripple Carry Adder (RCA),Carry Skip Adder (CSkA), Carry Increment Adder (CIA), Carry Look Ahead Adder (CLA), Carry Save Adder (CSA), Carry Select Adder (CSlA) and Carry Bypass Adder (CBA) are discussed below [6]

2.4.1 Ripple-carry adder

It is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is called a ripple-carry adder, since each carry bit "ripples" to the next full adder [6]. The first (and only the first) full adder may be replaced by a half adder (under the assumption that Cin = 0).The carry-in must travel through n XOR-gates in adders and n carry-generator blocks to have an effect on the carry-out. The layout of a ripple-carry adder is simple, which allows fast design time; however, the ripple-carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder. The gate delay can easily be calculated by inspection of the full adder circuit. Each full adder requires three levels of logic. In a 32-bit ripple-carry adder, there are 32 full adders, so the critical path (worst case) delay is 3 × 31 (from input to later adder) + 2 (in later adder) = 95 gate delays. The general equation for the worst-case delay for a n-bit carry-ripple adder is

$$Tcra(n) = (n-1).Tc + Ts = (n-1).3D + 2D = (3n-1)D$$

The delay from bit position 0 to the carry-out is a little different:

$$Tcra[0: cout] = Tcra[co: cn](n) = n.3D = 3nD$$

A design with alternating carry polarities and optimized AND-OR-Invert gates can be about twice as fast [7].

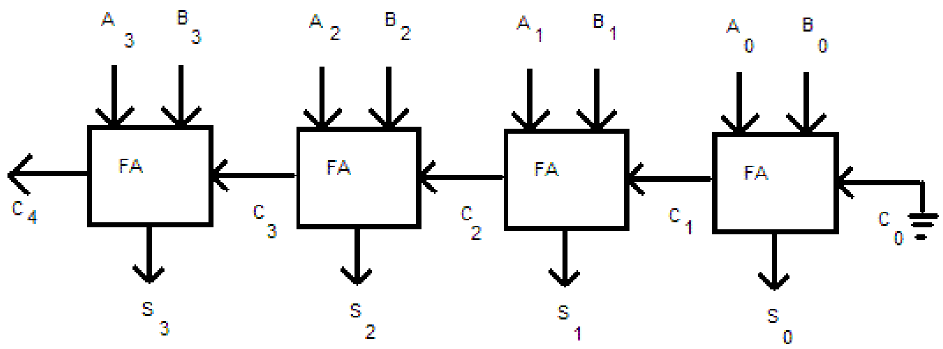


Figure 2.7 Block diagram of 4 bit Ripple-carry adder

2.4.2 Carry Lookahead Adder

Carry look-ahead adder is designed to overcome the latency introduced by the rippling effect of the carry bits. The propagation delay occurred in the parallel adders can be eliminated by carry lookahead adder. This adder is based on the principle of looking at the lower order bits of the augends and addend if a higher order carry is generated. This adder reduces the carry delay by reducing the number of gates through which a carry signal must propagate [9]. Carry lookahead depends on two things: Calculating for each digit position, whether that position is going to propagate a carry if one comes in from the right and combining these calculated values to be able to deduce quickly whether, for each group of digits, that group is going to propagate a carry that comes in from the right. The net effect is that the carries start by propagating slowly through each 4-bit group, just as in a ripple-carry system, but then moves 4 times faster, leaping from one lookahead carry unit to the next. Finally, within each group that receives a carry, the carry propagates slowly within the digits in that group. This adder consists of three stages: propagate block/generate block, sum generator and carry generator. A simple 4-bit generalized Carry Look Ahead circuit that combines with the 4-bit Ripple Carry Adder is shown below.

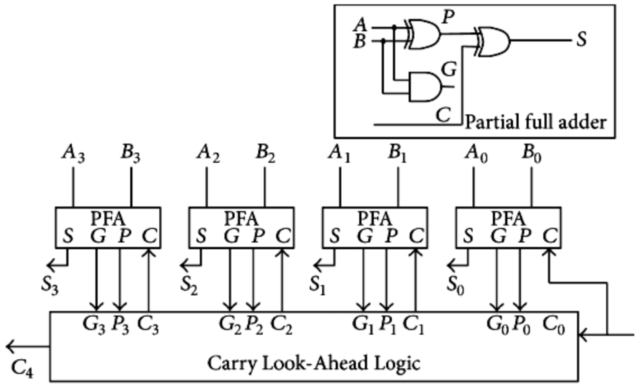


Figure 2.8 Block diagram of Carry Lookahead adder

For each bit in a binary sequence to be added, the Carry Look Ahead Logic will determine whether that bit pair will generate a carry or propagate a carry. This allows the circuit to "pre-process" the two numbers being added to determine the carry ahead of time. Then, when the actual addition is performed, there is no delay from waiting for the ripple carry effect (or time it takes for the carry from the first Full Adder to be passed down to the last Full Adder) [7].

2.4.3 Carry Increment Adder

The standard Carry Increment Adder (CIA) consists of RCA's and incremental circuitry. The incremental circuit is designed using half address in ripple carry chain with a sequential order. The addition operation is done by separating the total number of bits in to group of 4bits and addition operation is performed by several 4-bit RCA's. Instead of computing two partial sums for each group and selecting the correct one, only one partial sum is calculated and incremented if necessary, according to the input carry. Thus the second adder and the multiplexers in the carry-select scheme can be replaced by a much smaller incremental circuit and the modified architecture is the Carry Increment Adder (CIA). For example, an 8-bit CIA comprises of two 4-bit RCA. The first block of RCA adds first 4-bits to produce 4-bit partial sum and a carry output. Thus, first 4-bit of sum of CIA is directly obtained from first block of RCA. And the carry output of first RCA block is given as input to the Cin of incremental circuit. Incremental circuit consists of Half Adders (HA). Hence, the partial sum obtained from the second RCA block is given to incremental circuit. The block diagram representation of an 8-bit CIA_RCA is as shown in Figure 2.8[7].

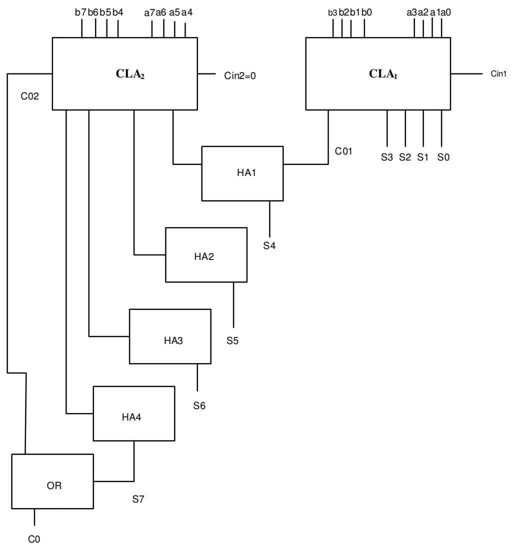


Figure 2.9 Block diagram of Carry Increment Adder

2.4.4 Carry Skip Adder

A carry-skip adder (also known as a carry-bypass adder) is an adder implementation that improves on the delay of a ripple-carry adder with little effort compared to other adders. The improvement of the worst-case delay is achieved by using several carry-skip adders to form a block-carry-skip adder [6]. As the name indicates, Carry Skip Adder uses skip logic in the propagation of carry.It consists of two logic gates. It is designed to speed up the addition operation by adding a propagation of carry bit around a portion of entire adder.

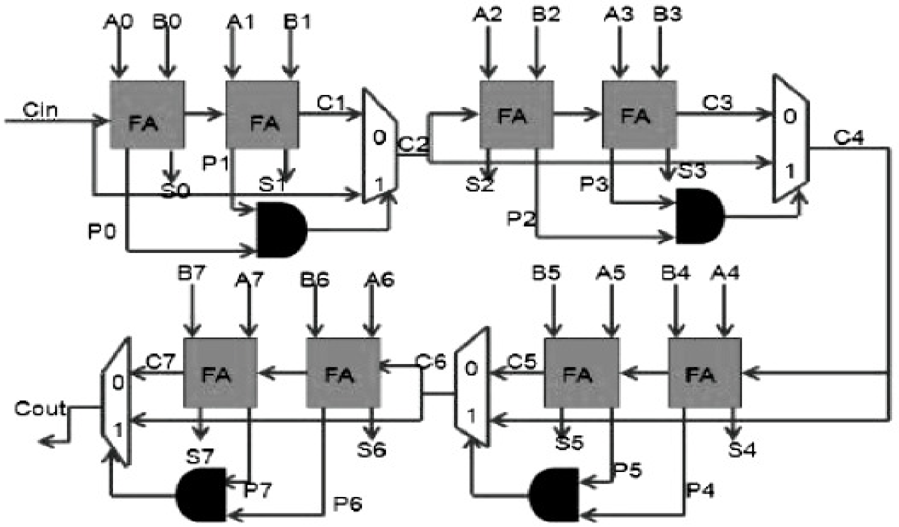


Figure2.10 Block diagram of Carry Skip Adder

A carry-skip adder consists of a simple ripple carry-adder with a special speed up carry chain called a skip chain. Carry skip adder is a fast adder compared to ripple carry adder when addition of large number of bits take place; carry skip adder provides a good compromise in terms of delay, along with a simple and regular layout. A carry-skip adder is designed to speed up a wide adder by aiding the propagation of a carry bit around a portion of the entire adder. Actually the ripple carry adder is faster for small values of N. However the industrial demands these days, which most desktop computers use word lengths of 32 bits like multimedia processors, makes the carry skip structure more interesting. The crossover point between the ripple-carry adder and the carry skip adder is dependent on technology considerations and is normally situated 4 to 8 bits [7].

2.4.5 Carry Bypass Adder

As in a ripple-carry adder, every full adder cell has to wait for the incoming carry before an outgoing carry can be generated. This dependency can be eliminated by introducing an additional bypass (skip) to speed up the operation of the adder. An incoming carry Ci,0=1 propagates through complete adder chain and causes an outgoing carry C0,7=1 under the conditions that all propagation signals are 1. This information can be used to speed up the operation of the adder, as shown in Figure 2.11. When BP = P0P1P3P4P5P6P7P8 = 1, the incoming carry is forwarded immediately to the next block through the bypass and if it is not the case, the carry is obtained via the normal route. If (P0P1P3P4P5P6P7 = 1) then C0,7 = Ci,0 else either Delete or Generate occurred. Hence, in a CBA the full adders are divided into groups, each of them is "bypassed" by a multiplexer if its full adders are all in propagated [7].

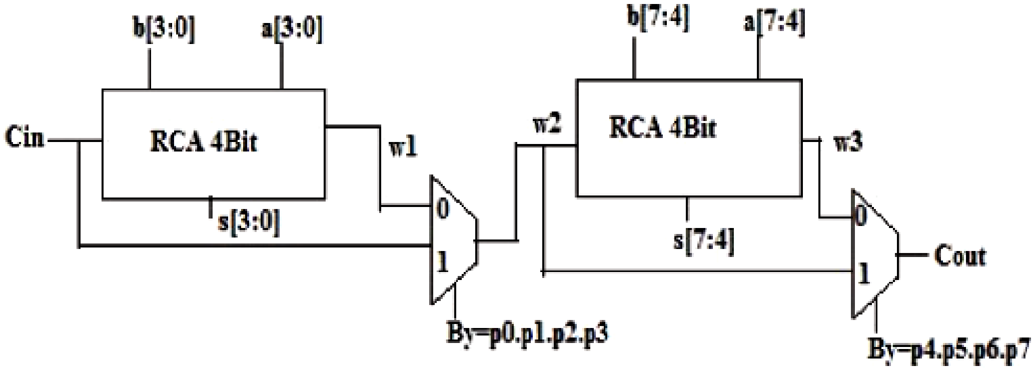


Figure2.11 Block diagram of Carry Bypass Adder

2.4.6 Carry Select Adder

A carry-select adder is divided into sectors, each of which – except for the least-significant performs two additions in parallel, one assuming a carry-in of zero, the other a carry-in of one. A four bit carry select adder generally consists of two ripple carry adders and a multiplexer. The carry-select adder is simple but rather fast. Adding two n-bit numbers with a carry select adder is done with two adders (two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known. The design schematic of Carry Select Adder is shown in Figure 2.12. A carry-select adder speeds 40% to 90%faster than RCA by performing additions in parallel and reducing the maximum carry path [7].

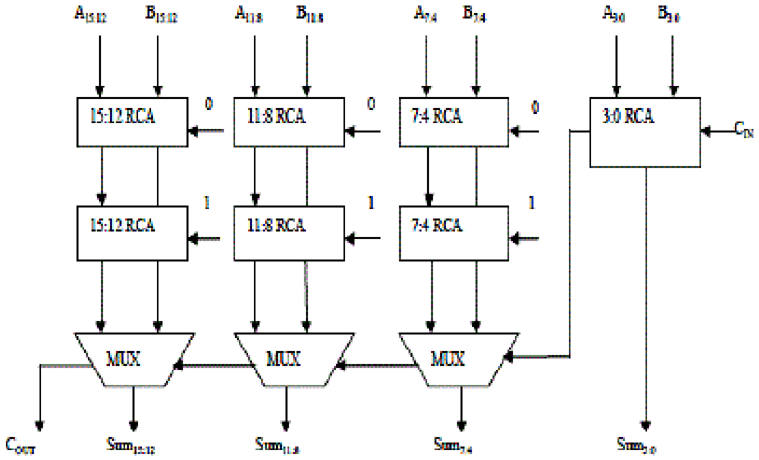


Figure 2.12 Block diagram of Carry Select Adder

2.4.7 Carry Save Adder

The carry-save adder reduces the addition of 3 numbers to the addition of 2 numbers. The propagation delay is 3 gates regardless of the number of bits. The carry-save unit consists of n full adders, each of which computes a single sum and carries bit based solely on the corresponding bits of the three input numbers. The entire sum can then be computed by shifting the carry sequence left by one place and appending a 0 to the front (most significant bit) of the partial sum sequence and adding this sequence with RCA produces the resulting n+1-bit value. This process can be continued indefinitely, adding an input for each stage of full adders, without any intermediate carry propagation. These stages can be arranged in a binary tree structure, with cumulative delay logarithmic in the number of inputs to be added, and invariant of the number of bits per input. The main application of carry save algorithm is, well known for multiplier architecture is used for efficient CMOS implementation of much wider variety of algorithms for high speed digital signal processing .CSA applied in the partial product line of array multipliers will speed up the carry propagation in the array [7].

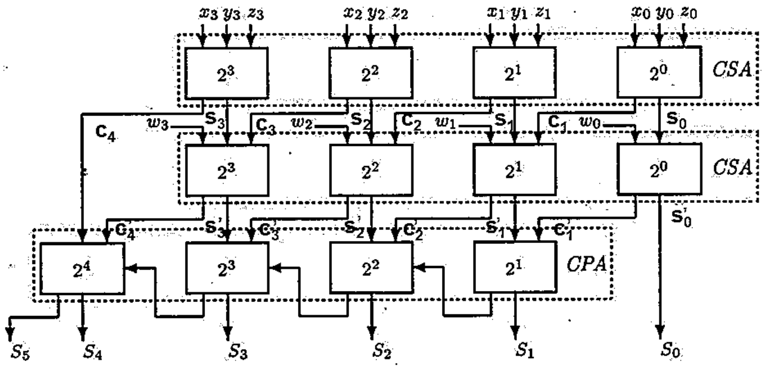


Figure2.13 Block diagram of Carry SaveAdder

Chapter 03

ANALYSIS OF VARIOUS CMOS FULL ADDER

3.1 General Idea

There are various design styles for implementing full adder. Different logic styles can be investigated from different points of view. Evidently, they tend to favor one performance aspect at the expense of others. In other words, it is different design constraints imposed by the application that each logic style has its place in the cell library development. Even a selected style appropriate for a specific function may not be suitable for another one. For example, static approach presents robustness against noise effects, so automatically provides a reliable operation. The issue of ease of design is not always attained easily. Pass transistor logic style is known to be a popular method for implementing some specific circuits such as multiplexers and XOR-based circuits, like adders. On the other hand, dynamic logic facilitates the realization of fast, small and complex gates. However, this advantage is gained at the expense of parasitic effects such as load sharing, which makes the design process hazardous [8][9].In this chapter different logic style for designing a 1 bit full adder is presented.

3.2 Conventional CMOS Full Adder

Conventional CMOS full adder is designed with 28 transistors. The structure is constructed using regular CMOS design consists of PMOS pull-up and NMOS pull-down transistors. The pull up network consists of PMOS transistors and pull down network consists of NMOS devices. The function of pull up network is to provide connection between gate output and V_{dd}, anytime the output of the gate is meant to be high. Similarly, function of Pull down Network is to provide connection between gate output and GND anytime the output of the gate is meant to be low. The pull up network and pull down network are mutually exclusive to each other. [10] To implement the full adder circuit logic equation (3.1) and (3.2) are taken and translated into complementary CMOS circuit. Some logic manipulations helps to reduce the transistor count.

The manipulation of equations for designing 28 transistor full adder is given below:

$Sum = ABC + AB'C' + A'BC' + A'B'C$

$= ABC + AB'C' + AB'C' + A'BC' + A'B'C$

$= ABC + AA'B' + AB'B'C' + AA'C' + AC'B'C' + A'C'B + BB'C'C' + A'B'C$

$+B'B'CC'$

$= ABC + A'(AB' + AC' + BC' + B'C) + B'C'(AB' + AC' + BC' + B'C)$

$= ABC + A'(AB' + AC' + BC' + B'C)(A' + B'.C')$

$= ABC + A'(AB' + AC' + BB' + BC' + B'C + CC')(A' + B'.C')$

$= ABC + (A + B + C)(B' + C').(A' + B'C')$

$= ABC + (A + B + C)\overline{BC}.\overline{A.(B + C)}$

$= ABC + (A + B + C)\overline{BC + A(B + C)}$

$= ABC + (A + B + C)\overline{BC + AB + CA}$

$Sum = ABC + (A + B + C)\overline{Cout}$ (3.1)

$Cout = AB + BC + CA$

$= AB + C(A + B)$(3.2)

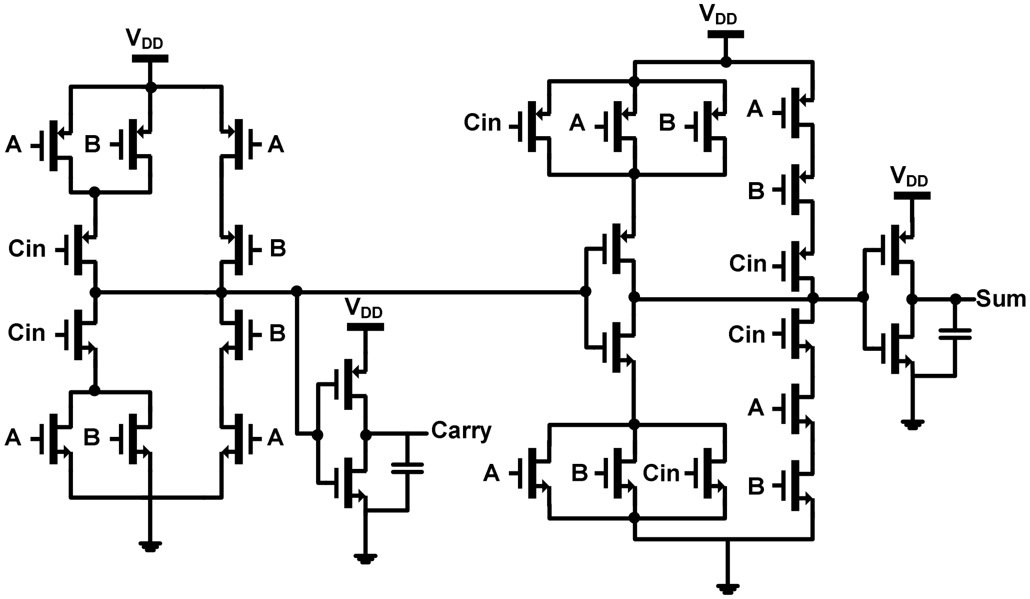


Figure 3.1 Block diagram of conventional CMOS full adder

3.2.1 Advantages of C-CMOS full adder

- The layout of CMOS is straight-forward and efficient due to complementary transistor pairs.
- It shows robustness against voltage scaling and transistor sizing.
- C-CMOS style gives improved quality of output in spite of transistor sizing and voltage scaling.
- It also gives a full voltage swing, which is needed in complex designs.
- More on, the layout of this design is area efficient and simple attributing to the PMOS NMOS transistor pairs and little variety of interconnecting wires [11].

3.2.2 Drawbacks of C-CMOS full adder

- At the output stage transistors are present in series, which decrease the driving capability of the circuit.
- Extra buffers are required for suitable compensation.
- The existence of the PMOS block is a major drawback because it has low mobility compared to the NMOS [10].
- The main drawback of C-CMOS full adder includes the usage of large number of transistor resulting in increased area overhead [10] [11].

3.3 Transmission Gate Full Adder

In conventional full adder circuits, we use CMOS technology i.e. PMOS and NMOS are used as a switch in complementary mode. Such applications of NMOS and PMOS as a switch is called pass transistor logic. Transmission gate logic circuit is a special kind of pass-transistor logic circuit. It is built by connecting a PMOS transistor and a NMOS transistor in parallel, which are controlled by complementary control signals. Transmission gate full adder circuits consists of transmission gates and requires 20 transistors to implement the logic of full adder [12]. The adder is designed based on X-OR gate which is realized by transmission gate. Less no. of transistor are required for this design approach compared to conventional full adder.

3.3.1 Transmission Gate

A transmission gate is defined as an electronic element that will selectively block or pass a signal level from the input to the output. This solid-state switch is comprised of a

PMOS transistor and NMOS transistor. The transmission gate consists of two MOSFETs, one n-channel responsible for correct transmission of logic zeros, and one p-channel, responsible for correct transmission of logic ones. A transmission gate has three inputs, called source, NMOS gate, and PMOS gate; and it has one output, called drain. The two transistors, an NMOS and a PMOS are connected in parallel configuration.

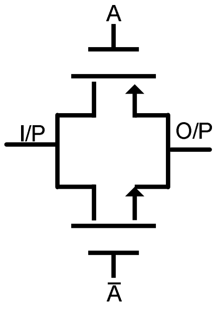


Figure 3.2 Transmission Gate

When the control input A is a logic zero (negative power supply potential), the gate of the NMOS is also at a negative supply voltage potential. The gate terminal of the PMOS is caused by the inverter, to the positive supply voltage potential. Accordingly, neither of the two transistors will conduct and the transmission gate turns off. When the control input A is a logic one, the gate terminal of the NMOS is located at a positive supply voltage potential & the gate terminal of the PMOS is now at a negative supply voltage potential. At this condition the transmission gate conducts [13].

3.3.2 Circuit description

The transmission gate full adder is shown in fig 3.3[13] - [15]. The circuit architecture can be divided into two parts i.e. sum and carry. The sum circuit is based on XOR logic which can be understood from the equation of sum of the full adder which is given as

$$Sum = A \oplus B \oplus C$$

The XOR logic can be applied into the transmission gate to form the sum circuit of the adder. The XOR gate is a digital logic gate that gives a true (1/HIGH) output when the number of true inputs is odd. Either the inputs are false (0/LOW) or both are true, a false output results. The truth table of XOR gate is given below

Table 3.1: Truth table of XOR gate

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

First 2 sets of transmission gate are taken to get the logic output Y which is A XOR B. The implementation is shown in Figure 3.3.

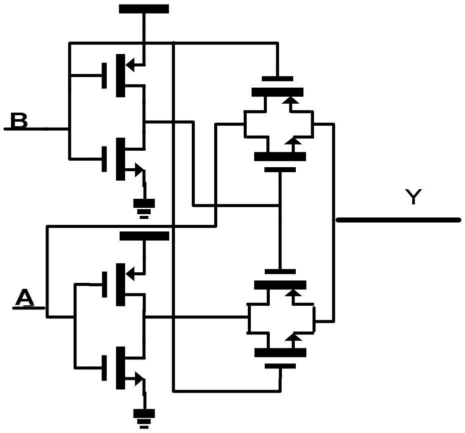


Figure 3.3 First step of generation process of sum in transmission gate full adder

So, from figure 3.3 it is realized that if B is considered the controlling input i.e. the gate input and A is given input to the I/P(source) terminal, then when B=0, both NMOS &PMOS of the lower transmission gate are off and the upper transmission gate is on. At this time output in O/P terminal is equal to the input i.e. A. When B=1 both NMOS & PMOS of the lower transmission gate turn on and the output is inverted to the input A. At this time the upper transmission gate remains off.

The output Y which is obtained along with the carry input (Cin) are then given as input to another two sets of transmission gate to finally obtain the sum output.

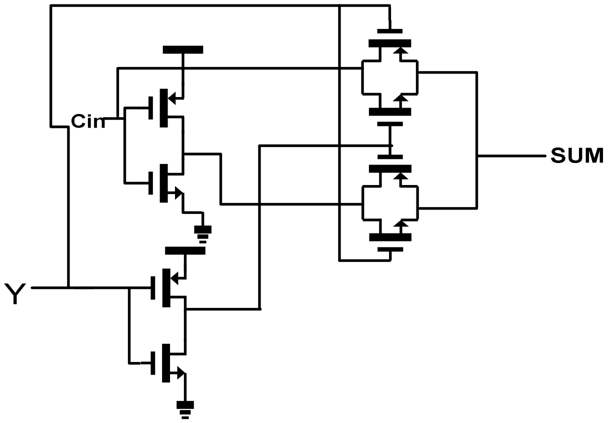


Figure 3.4 Second step of generation process of sum in transmission gate full adder

For the carry part it is needed to analyze the truth table of full adder given in section 2.4.

Table 3.2: Analysis of truth table of full adder for carry output

A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

It is seen that if the inputs A & B are same then C_{out} =A=B and

if A & B are different then C_{out}=C_{in}.

The mentioned two logic are again applied to two set of transmission gate to obtain the carry part of the circuit.The overall block diagram of the 20T transmission gate full adder is given below in Figure3.5.

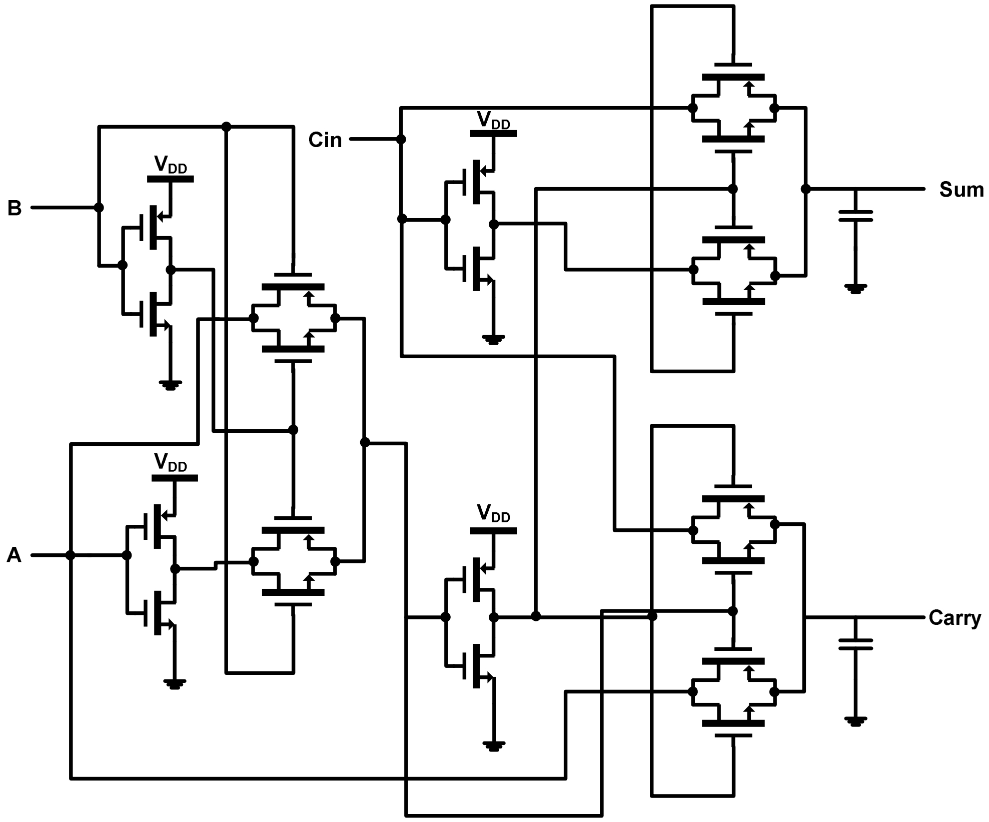


Figure3.5 Transmission gate full adder

3.3.3 Advantages of transmission gate full adder

- The architecture of transmission gate full adder is simpler than conventional CMOS full adder.
- Reduced number of transistors occupies small area which leads to minimize the interconnection of the wires [14].
- The degradation voltage level at output problem can be overcome by using this technique.
- It is the fastest adder so far been reported.

3.3.4 Drawbacks of transmission gate full adder

- Power consumption is greater in transmission gate full adder than conventional full adder.
- Driving capability is lower compared to other design topologies [13].

3.4 14 Transistor Full Adder

For faster operation another design technique is 14 transistor full adder. It uses pass transistor logic and double pass transistor logic to implement the adder cell. 14 transistors are used in the architecture. This circuit occupies less area in comparison with conventional full adder.

3.4.1 Pass transistor logic

Pass transistor logic (PTL) describes several logic families used in the design of integrated circuits. It reduces the count of transistors used to make different logic gates, by eliminating redundant transistors. Transistors are used as switches to pass logic levels between nodes of a circuit, instead of as switches connected directly to supply voltages. In conventional logic families input is applied to gate terminal of transistor but in PTL it is also applied to source/drain terminal [16].

In NMOS pass transistor primary inputs drive both gate and source/drain terminals. NMOS switch closes when the gate input is high. It is noted that NMOS transistors pass a strong 0 but a weak 1. PMOS switch closes when the gate input is low. PMOS transistor passes a strong 1 but a weak 0.

3.4.2 Double pass transistor logic

The delay of pass transistor networks increases quadratically with the number of stages and as a result, some intermediate buffers should be used to make strong VDD and ground. All these problems arise from the fact that NMOS transistors cannot pass VDD faithfully to the other side. The solution is using a complementary PMOS transistor in parallel with NMOS to generate a strong VDD at the output. This structure is double pass transistor logic (DPL).DPL uses both PMOS and NMOS devices in the pass transistor network to avoid non full swing problems. Double pass-transistor logic shows improved circuit performance at reduced supply voltage. Its symmetrical arrangement and double-transmission characteristics improves the gate speed without increasing the input capacitance [17].

3.4.3 Circuit description

The logic diagram of 14 transistor full adder is shown in Figure 3.6.

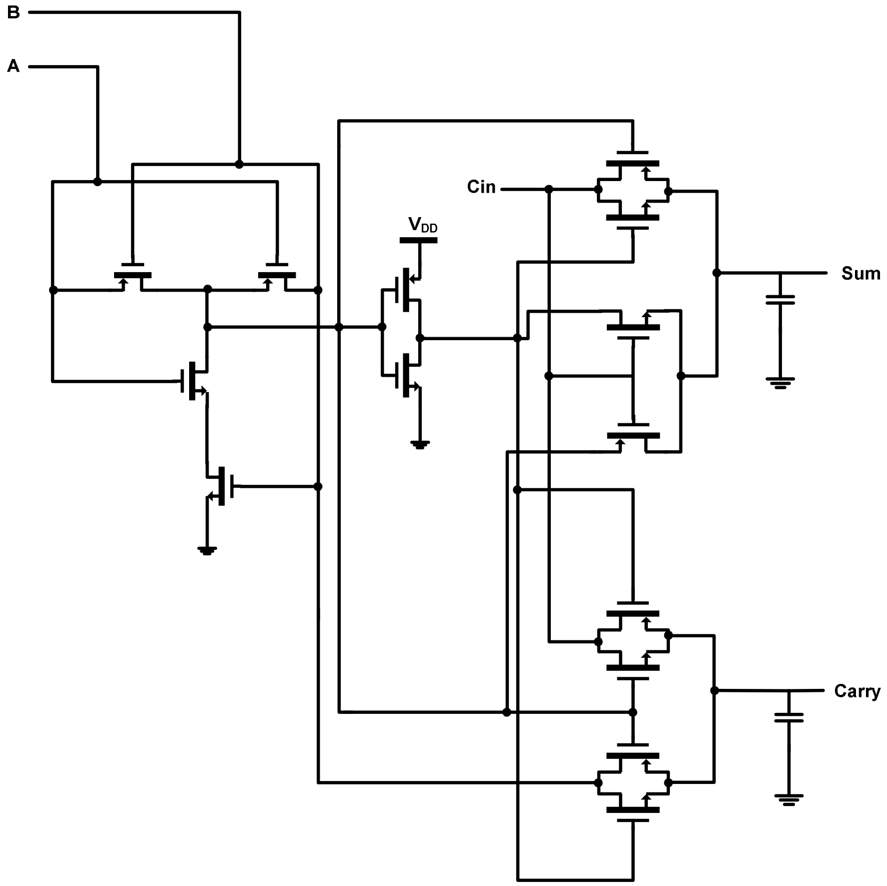


Figure 3.6 14T Full Adder

The 14T full adder contains a 4 transistor PTL XOR gate; an inverter and two transmission gates based multiplexer designs for sum and Cout signals. This circuit has 4 transistor XOR which in the next stage is inverted to produce XNOR. These XOR and XNOR are used simultaneously with the Cin to generate sum and Cout [18] [19].

By looking at the truth table of full adder it is seen that, if A XOR B gives a logic 0 output ,then sum=Cin. If A XOR B gives logic 1 output then sum=Cin'. This logic is applied while designing the sum circuit. For generating the sum output a transmission gate and two pass transistors (N & P) are used. Cin is given as input to the transmission gate and it also drives the gate of the N & P pass transistors. The XOR & XNOR drives the gates of the transmission gate.

For generating carry output two transmission gates are used. The logic behind the carry circuit is same as the carry circuit of transmission gate (20T) full adder explained in section 3.3.2. 14T full adder is a design modification of transmission gate full adder with reduction in transistor number. In transmission gate full adder 4 inverters are used whereas in 14T full adder there is only 1 inverter which results in reduction of 6 transistors [17].

3.4.4 Advantages of 14T full adder

- The circuit comprises less area than conventional CMOS full adder [19].
- The operation speed is faster [13].

3.4.5 Drawbacks of 14T full adder

- The power dissipation is more in this adder than conventional CMOS adder.

3.5 Gate Diffusion Input (GDI) Full Adder

GDI (Gate diffusion input) is a new low power design technique. It allows implementation of a wide range of complex logic function by using only two transistors. This method is suitable for design of fast, low power circuits using reduced number of transistor. This method is suitable for design of fast, low power circuits, using reduced number of transistor while improving power characteristics [19]-[23].

3.5.1 Basics of GDI cell

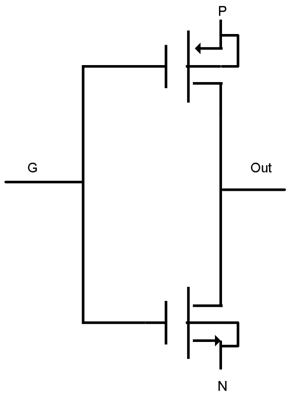


Figure 3.7 GDI cell

Table 3.3: Logic function of the basic GDI cell

N	P	G	OUT	Function
‘0’	B	A	A’B	F1
B	‘1’	A	A’+B	F2
‘1’	B	A	A+B	OR
B	‘0’	A	AB	AND
C	B	A	A’B+AC	MUX
‘0’	‘1’	A	A’	NOT

The basic GDI cell shown in Fig 3.7 was proposed by Morgenshtein [18]. The main difference between the CMOS and GDI based design is that the source of the PMOS in a GDI cell is not connected to VDD and the source of the NMOS is not connected to GND. This feature gives the GDI cell two extra input pins for use which makes the GDI design more flexible than CMOS design. GDI cell consists of three inputs - G (common gate input of NMOS and PMOS), P (input to the source/drain of PMOS) and N (input to the source/drain of NMOS). Bulks of both NMOS and PMOS are connected to N and P respectively. Table 3.3 shows different logic functions implemented by GDI logic [23] based on different input values. So by using GDI technique various logic functions with less power and high speed as compared to conventional CMOS design can be implemented.

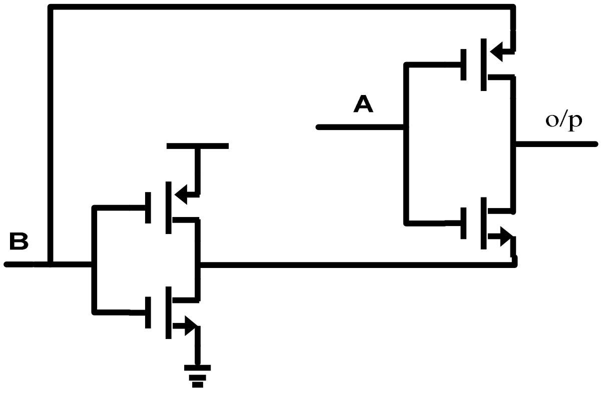


Figure 3.8 GDI based XOR gate

One of the main building blocks of full adder circuit is XOR gate. So if we can optimize XOR gate then it can improve the overall performance of the 1 bit full adder circuit.Figure 3.8 shows the implementation of XOR gate using GDI technique.It uses less number of transistors as compared to conventional design of XOR gate using CMOS logic.

In the circuit operation, When A& B both are at logic 0, NMOS transistor is switched OFF and PMOS transistor is switched ON. Therefore, the output, which is approximately equal to |Vtp|, is obtained at the output, where Vtp is the threshold voltage of PMOS transistor. However, when A & B both are at logic 1, the NMOS transistor becomes ON and PMOS transistor becomes OFF and passes ground potential at the output.

When A=0 & B=1, PMOS transistor is switched ON and NMOS transistor is switched OFF. Therefore, VDD passes through PMOS transistor. On the contrary, the case occurs when A= 1 & B=0 NMOS turns ON and PMOS turns OFF resulting in NMOS passes a poor '1' signal which is about VDD-Vtn at the output, Vtn denotes the threshold voltage of NMOS transistor.

3.5.2 Circuit description

The GDI based full adder is designed using XOR gate described in section 3.4.1. 10 transistors are required for this architecture. The block diagram of the adder is given below in Figure 3.9.

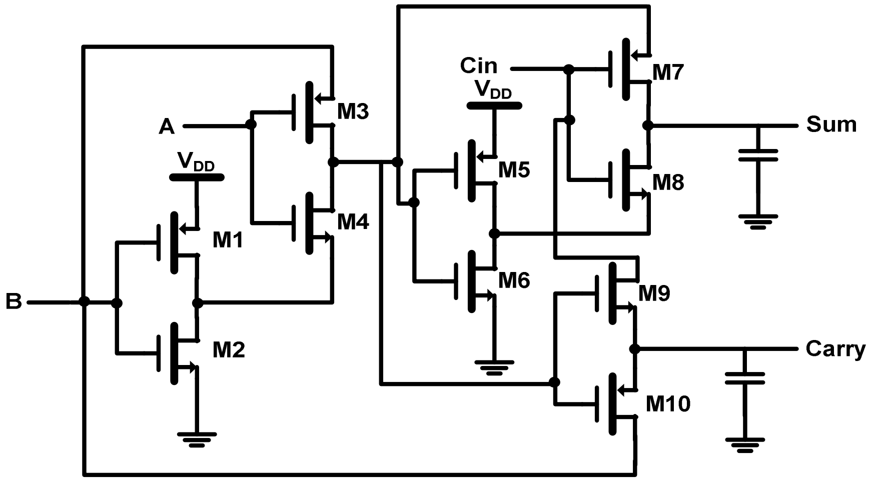


Figure 3.9 GDI based full adder

In figure, M1, M3, M5, M7, M10 are PMOS transistors and M2, M4, M6, M8, M9 are NMOS transistors [19].

3.5.3 Advantages of GDI full adder

- It is a low power design technique which offers the implementation of the logic function with fewer numbers of transistors.
- GDI gates provide reduced voltage swing at their outputs, the reduction in voltage swing is beneficial to power consumption.
- The two extra input pins which makes the GDI design more flexible than CMOS design.
- The technique is very power efficient without huge amount of transistor count [19]-[23].

3.5.4 Drawbacks of GDI full adder

- The major problem of a GDI cell is that it requires twin-well CMOS or silicon on insulator (SOI) process to realize so it is more expensive to comprehend a GDI chip.
- If only standard p-well CMOS process is used, the GDI scheme will face the problem of lacking driving capability which makes it more expensive and difficult to realize as a feasible chip.
- Reduced voltage swing at output may lead to slow switching in the case of cascaded operation.
- At low VDD operation, the degraded output may even cause circuit malfunction.
- As the allotment of supply and ground to PMOS and NMOS is not fixed in case of GDI, therefore, problem of low voltage swing arises [19].

Chapter 04

SIMULATION AND RESULT

4.1 General Idea

For simulating and comparing the performance of 1 bit full adder with different design topologies, CADENCE Design Suite 6.1.6 for GPDK 90nm CMOS technology using Virtuoso environment at room temperature was used. For the analysis, the supply voltage V_{DD} was set at 1.2Vfor each of the full adder. During the simulation, the inputs (A, B, C) were given pulsating DC known as V_{PULSE}, where the pulses were varied from 0V to 1.2V. Here, for comparison purpose the width and length of PMOS and NMOS were set at an identical value for each of the adder. Length was set at 100nm. Width of PMOS and NMOS were set at 240 nm and 120 nm respectively. Delay was calculated by doing transient analysis. By dc analysis the power consumptions by the adder circuits were obtained.

4.2 Truth Table Verification

Figure 4.1, 4.2, 4.3 & 4.4 verify the truth table of various full adder. The supply voltage is set at 1.2 Volt. Transient analysis is done for the verification. From the analysis, the outputs are obtained with respect to time by adding the three inputs. The figures show the three inputs A, B, C and the corresponding outputs sum and carry after adding the inputs. During the analysis the stop time was taken 120ns.

4.2.1 Conventional CMOS full adder

Figure 4.1 shows the transient analysis C-CMOS adder topology. The truth table of 1 bit full adder is successfully verified which is shown in the figure below. Some glitches are noticed in the output waveform. As in conventional C-CMOS full adder, the number of transistors are higher than the other simulated topologies, so the glitches are more visible in the output curves.

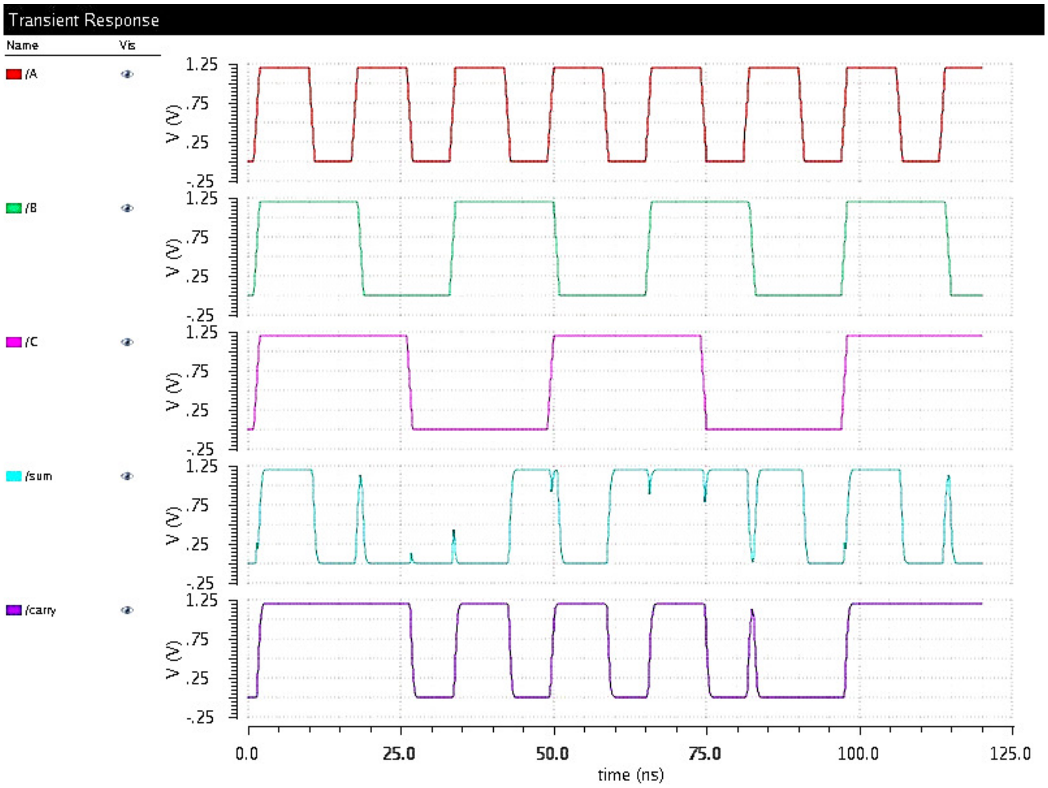


Figure 4.1 Transient analysis of conventional CMOS full adder

4.2.2 Transmission gate full adder

Figure 4.2 shows the transient analysis of transmission gate full adder. The truth table of 1 bit full adder is successfully verified. Glitches are noticed in the output waveform, but for this topology as the number of transistors is lower than the conventional full adder so the glitches are less evident in the output curves.

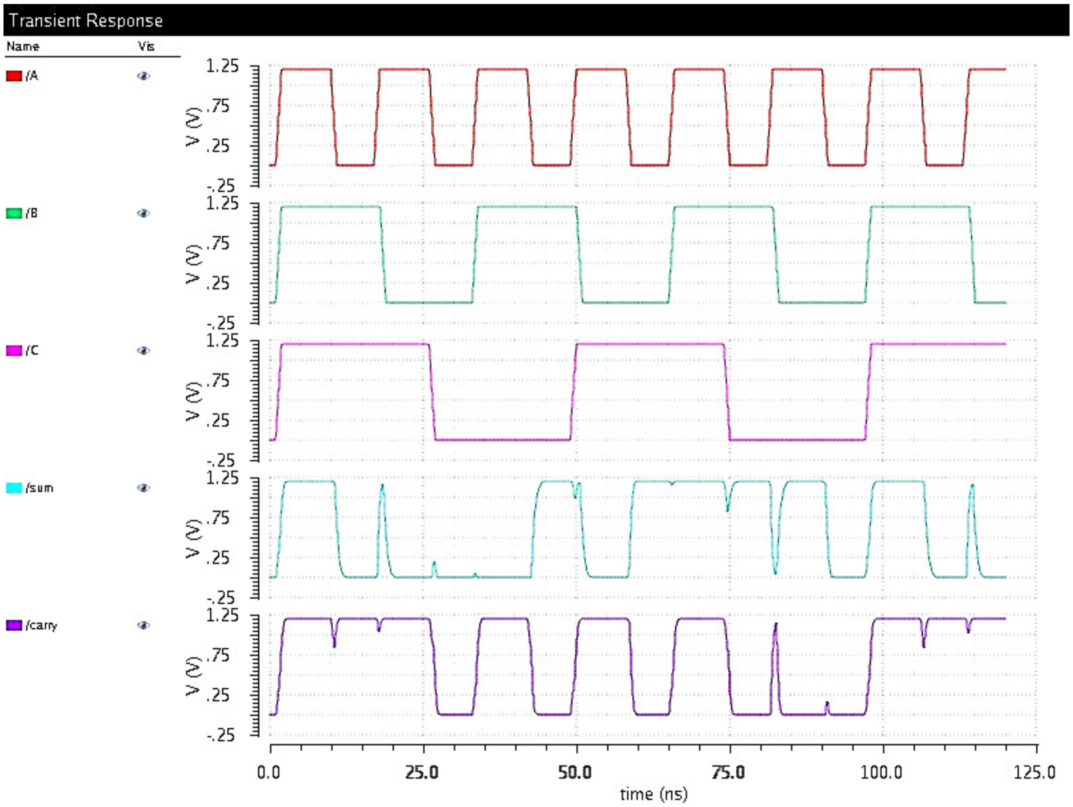


Figure 4.2 Transient analysis of transmission gate full adder

4.2.3 Fourteen transistor full adder

Figure 4.3 shows the transient analysis of fourteen transistor full adder. The truth table of 1 bit full adder is tested. In fourteen transistor full adder, the glitches observed are very less due to lesser transistor count.

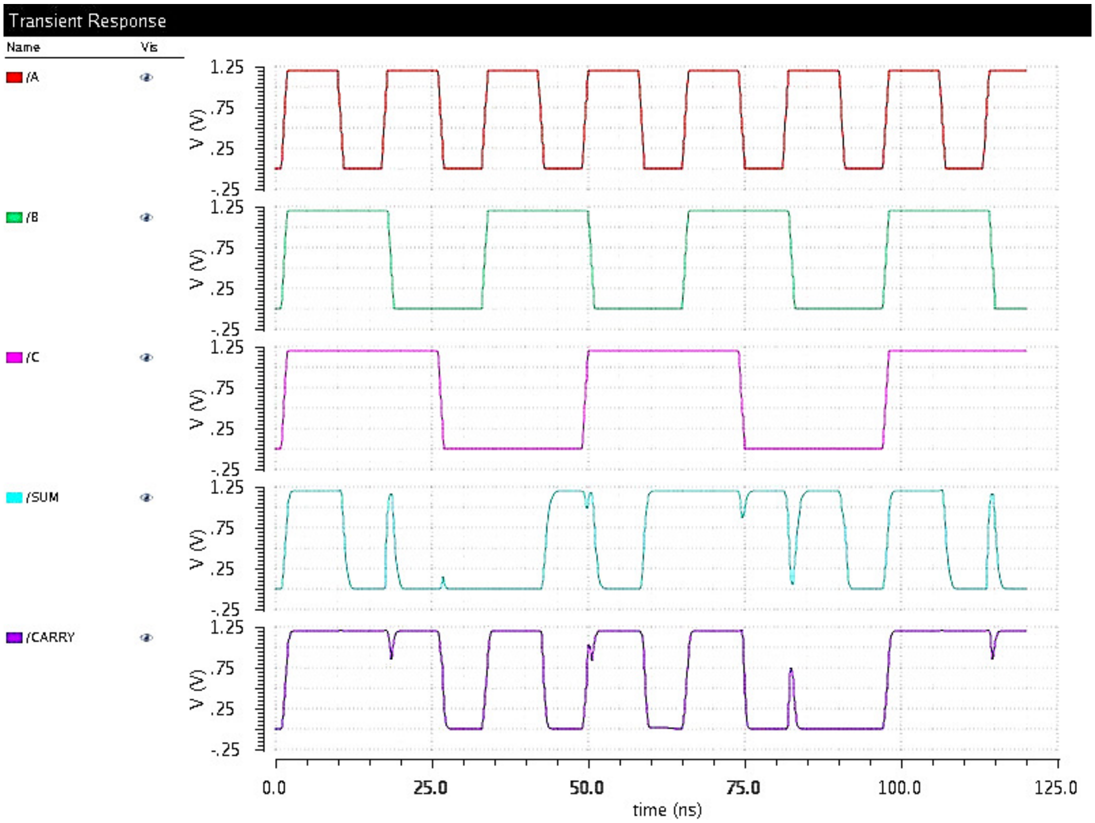


Figure 4.3 Transient analysis of fourteen transistor full adder

4.2.4 Gate diffusion input (GDI) full adder

Figure 4.4 shows the transient analysis of gate diffusion input full adder. The truth table of 1 bit full adder is verified. In the GDI full adder though the number of transistors are less than all other simulated full adder topologies, but still due to the configuration of GDI it is not possible to get the full swing of 1 and 0 in the output waveform. So in this topology the glitches seen in the output are the highest.

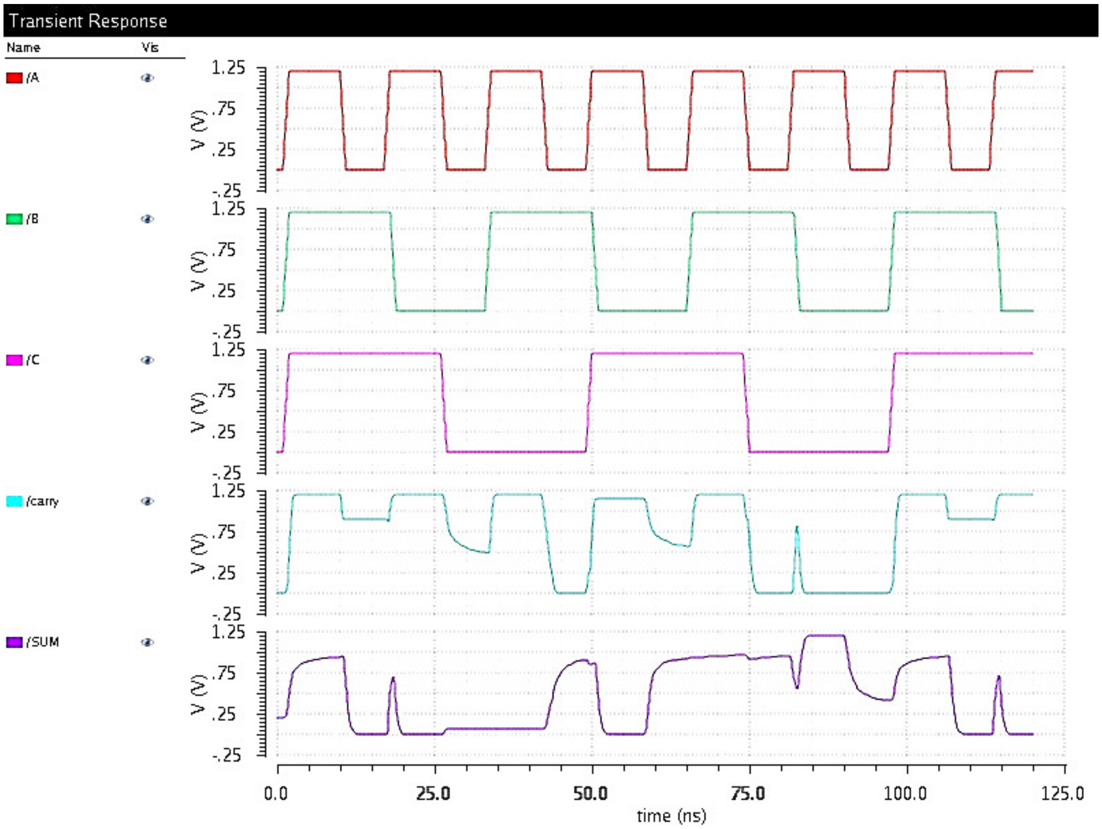


Figure 4.4 Transient analysis of gate diffusion input full adder

4.3 Delay Estimation

In digital electronics the propagation delay is the length of time which starts when the input to a logic gate becomes stable and valid to change, to the time that the output of that logic gate is stable and valid to change.

Propagation delay time, t_{pd}= maximum time from theinput crossing 50% to the output crossing 50%

Rise time, t_r= time for a waveform to rise from 20% to80% of its steady-state value

Fall time, t_f= time for a waveform to fall from 80% to20% of its steady-state value

Rising propagation delay, t_{PLH}= maximum rising time from theinput crossing 50% to the output crossing 50%.

Falling propagation delay, t_{PHL}= maximum falling time from theinput crossing 50% to the output crossing 50%

The average propagation delay,

$$t_{pd}=(t_{PLH}+t_{PHL})/2 \qquad (4.1)$$

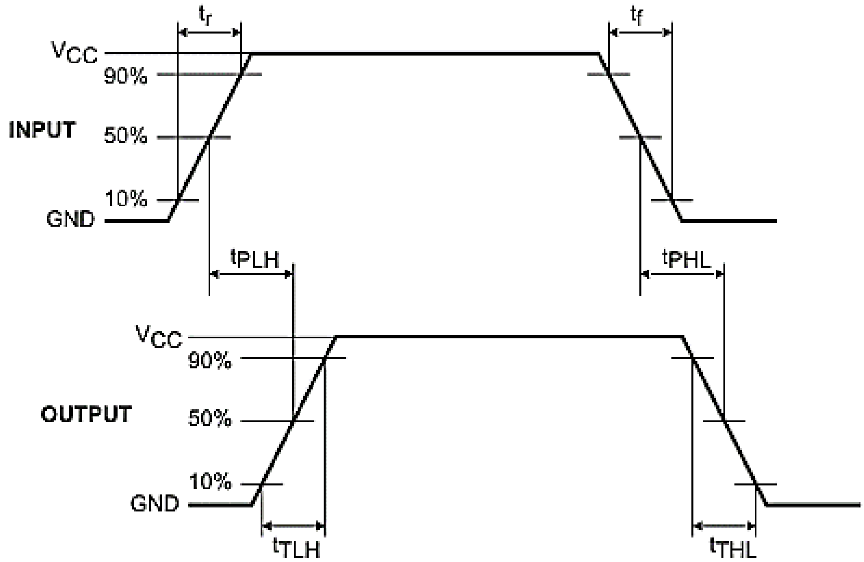


Figure 4.5 Delay Estimation

Reducing gate delays in digital circuits allows them to process data at a faster rate and improve overall performance. The determination of the propagation delay of a combined circuit requires identifying the longest path of propagation delays from input to output and by adding each t_{pd} time along this path.

In this paper propagation delay is calculated by performing transient analysis on various full adder topologies. A close observation on the transient analyzed waveform gives the rise time and fall time for the output. By using them average propagation delay is calculated from equation 4.1. Delay estimation for different adder topologies are shown in the following sections.

4.3.1 Propagation delay of conventional CMOS full adder

Here, this graph is representing the propagation delay of conventional CMOS full adder of input to sum. From the graph it is seen that, the green pulse is representing the "input" and the blue one is representing the corresponding output which is the "sum". Also the rising and falling propagation delays are also shown. By using these two delays, the average propagation delay for sum is calculated, which is shown below.

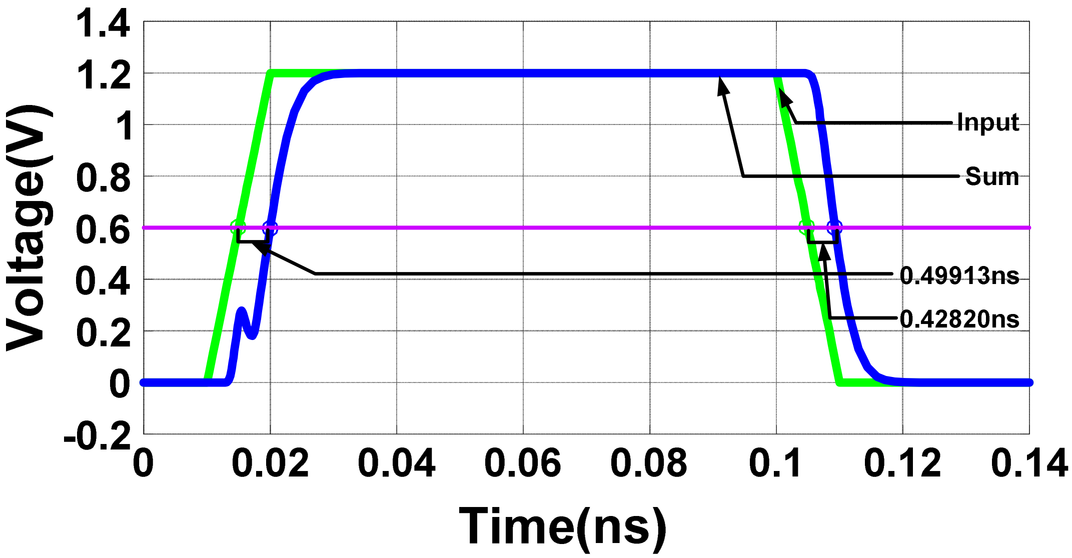


Figure 4.6 Propagation delay of input to sum

Here,

Rising propagation delay, t_{PLH}= 0.49913ns

Falling propagation delay, t_{PHL}=0 .42820ns

From equation 4.1,

Average propagation delay for sum = (0.49913+0.42820)/2 ns

= 0.463665 ns

Now, this graph is representing the propagation delay of conventional CMOS full adder of input to carry. From the graph it is seen that, the red pulse is representing the "input" and the magenta one is representing the corresponding output which is the "carry". Also the rising and falling propagation delays are also shown. By using these two delays, the average propagation delay for carry is calculated, which is shown below.

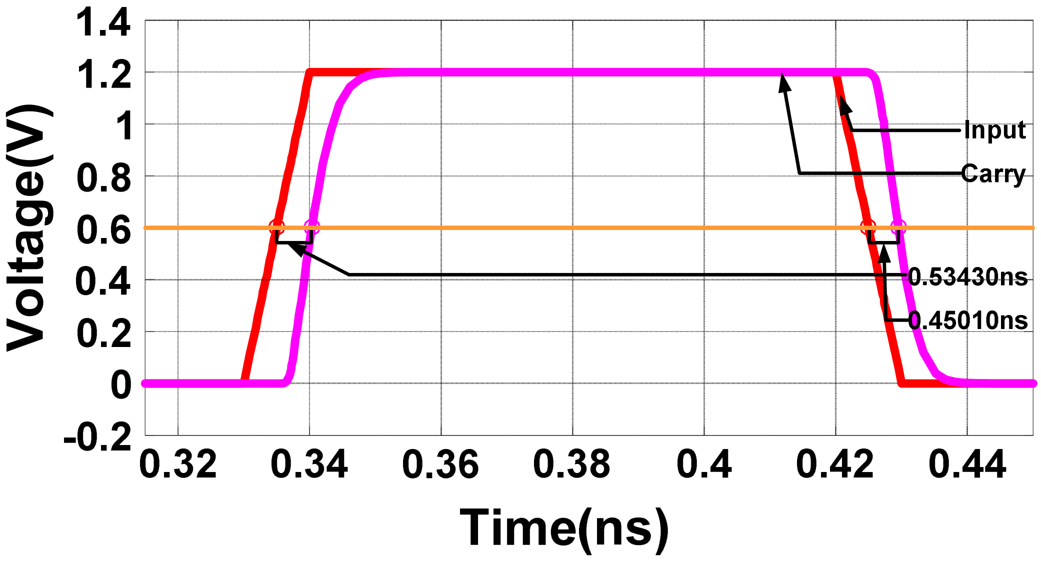


Figure 4.7 Propagation delay of input to carry

Here,

Rising propagation delay, t_{PLH}= 0.53430 ns

Falling propagation delay, t_{PHL}= 0 .45010 ns

From equation 4.1,

Average propagation delay for carry = (0.53430+0 .45010)/2 ns

= 0.4922ns

4.3.2 Propagation delay of transmission gate full adder

Here, this graph is representing the propagation delay of transmission gate full adder of input to sum. From the graph it is seen that, the green pulse is representing the "input" and the blue one is representing the corresponding output which is the "sum". Also the rising and falling propagation delays are also shown. By using these two delays, the average propagation delay for sum is calculated, which is shown below.

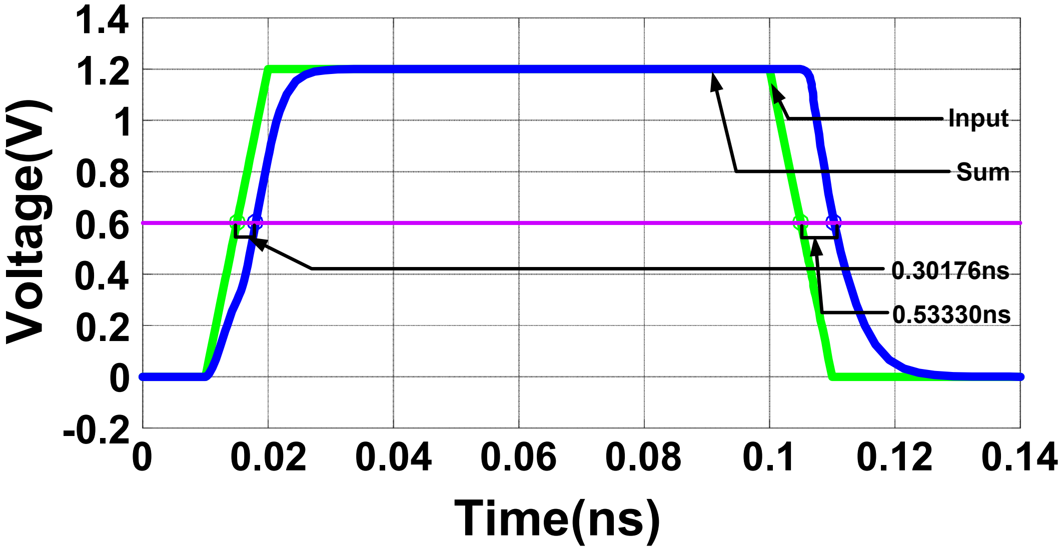


Figure 4.8 Propagation delay of input to sum

Here,

Rising propagation delay, t_{PLH}= 0.30176 ns

Falling propagation delay, t_{PHL}= 0.53330 ns

From equation 4.1,

Average propagation delay for sum = (0.30176+0.53330)/2 ns

= 0.41753 ns

Now, this graph is representing the propagation delay of transmission gate full adder of input to carry. From the graph it is seen that, the red pulse is representing the "input" and the magenta one is representing the corresponding output which is the "carry". Also the rising and falling propagation delays are also shown. By using these two delays, the average propagation delay for carry is calculated, which is shown below.

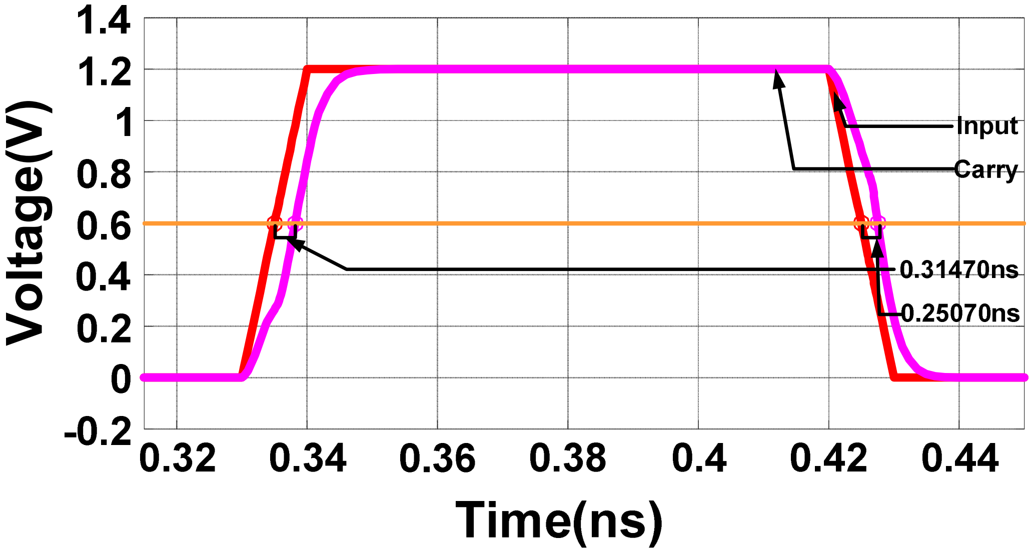


Figure 4.9 Propagation delay of input to carry

Here,

Rising propagation delay, t_{PLH}= 0.31470 ns

Falling propagation delay, t_{PHL}= 0.25070 ns

From equation 4.1,

Average propagation delay for carry = (0.31470+0.25070)/2 ns

= 0.2827ns

4.3.3 Propagation delay of fourteen transistor full adder

Here, this graph is representing the propagation delay of fourteen transistor full adder of input to sum. From the graph it is seen that, the green pulse is representing the "input" and the blue one is representing the corresponding output which is the "sum". Also the rising and falling propagation delays are also shown. By using these two delays, the average propagation delay for sum is calculated, which is shown below.

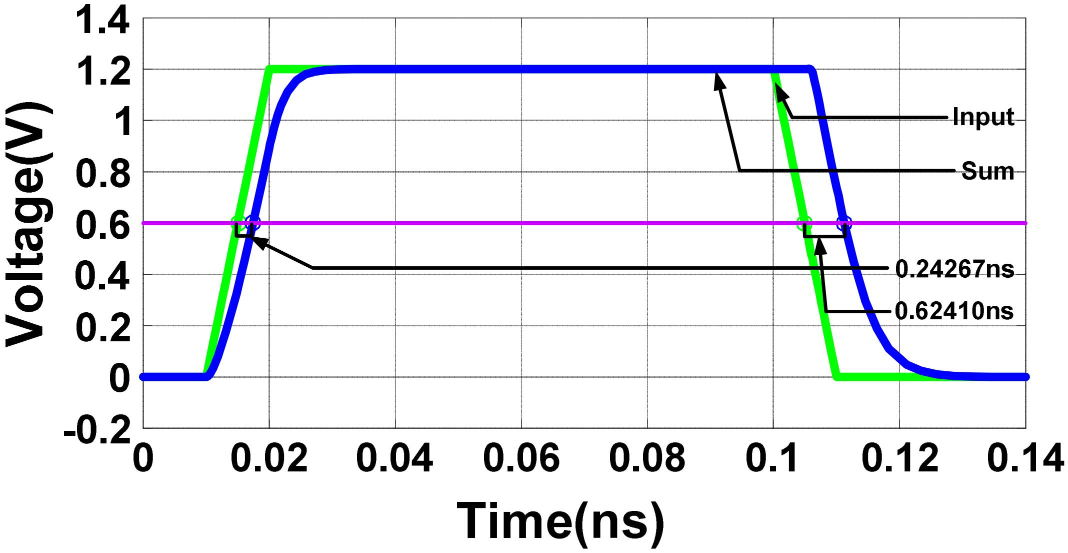


Figure 4.10 Propagation delay of input to sum

Here,

Rising propagation delay, t_{PLH}= 0.24267 ns

Falling propagation delay, t_{PHL}= 0.62410 ns

From equation 4.1,

Average propagation delay for sum = (0.24267+0.62410)/2 ns

= 0.433385 ns

Now, this graph is representing the propagation delay of fourteen transistor full adder of input to carry. From the graph it is seen that, the red pulse is representing the "input" and the magenta one is representing the corresponding output which is the "carry". Also the rising and falling propagation delays are also shown. By using these two delays, the average propagation delay for carry is calculated, which is shown below.

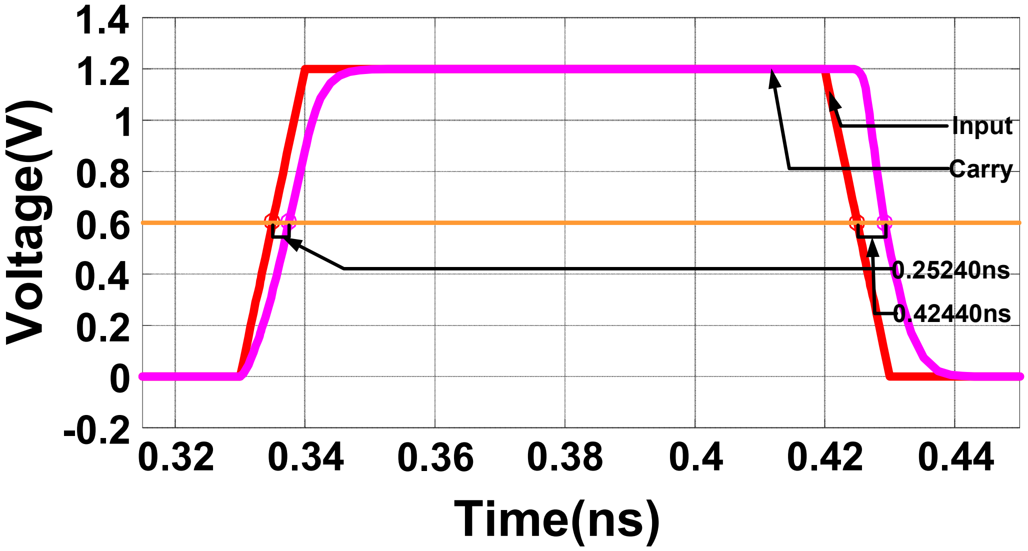


Figure 4.11 Propagation delay of input to carry

Here,

Rising propagation delay, t_{PLH}= 0.25240 ns

Falling propagation delay, t_{PHL}= 0.42440 ns

From equation 4.1,

Average propagation delay for carry = (0.25240+0.42440)/2 ns

= 0.3384 ns

4.3.4 Propagation delay of gate diffusion input full adder

Here, this graph is representing the propagation delay of gate diffusion input full adder of input to sum. From the graph it is seen that, the green pulse is representing the "input" and the blue one is representing the corresponding output which is the "sum". Also the rising and falling propagation delays are also shown. By using these two delays, the average propagation delay for sum is calculated, which is shown below.

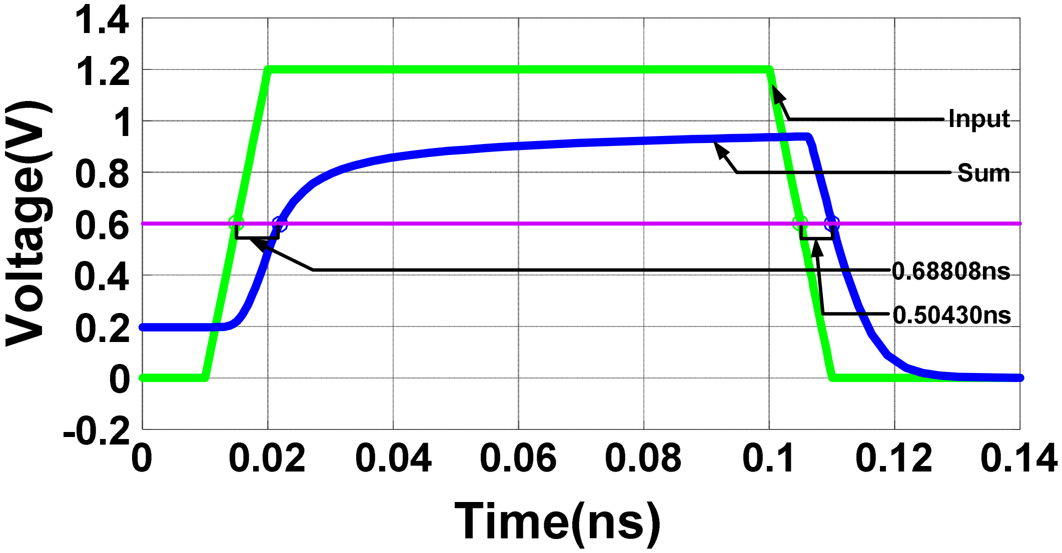


Figure 4.12 Propagation delay of input to sum

Here,

Rising propagation delay, t_{PLH}= 0.68808 ns

Falling propagation delay, t_{PHL}= 0.50430 ns

From equation 4.1,

Average propagation delay for sum = (0.68808+0.50430)/2 ns

= 0.59619 ns

Now, this graph is representing the propagation delay of fourteen transistor full adder of input to carry. From the graph it is seen that, the red pulse is representing the "input" and the magenta one is representing the corresponding output which is the "carry". Also the rising and falling propagation delays are also shown. By using these two delays, the average propagation delay for carry is calculated, which is shown below.

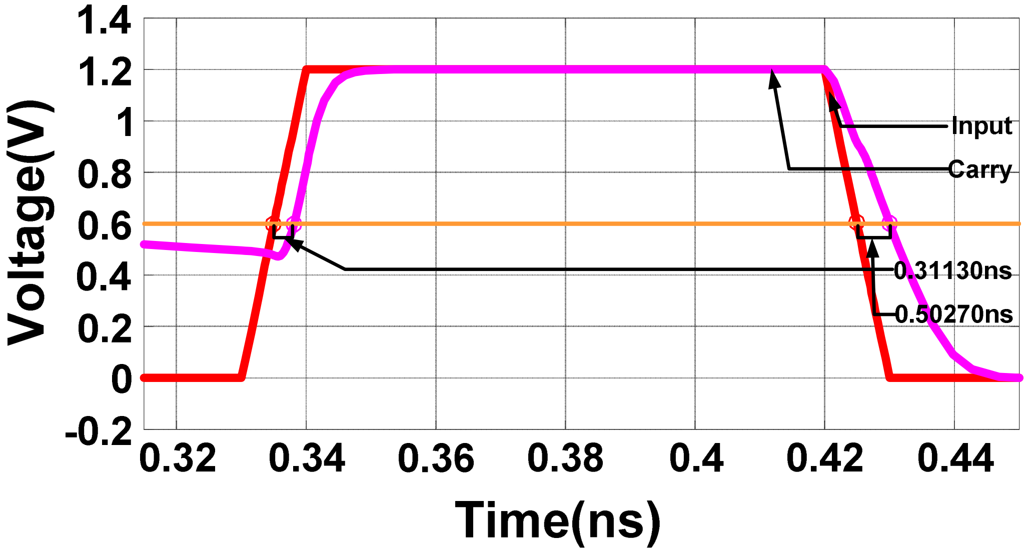


Figure 4.13 Propagation delay of input to carry

Here,

Rising propagation delay, t_{PLH}= 0.31130 ns

Falling propagation delay, t_{PHL}= 0.50270 ns

From equation 4.1,

Average propagation delay for carry = (0.31130+0.50270)/2 ns

= 0.407 ns

4.4 Power Consumption Analysis

The power consumed in an adder circuit can be categorized into two portions, one is the static power consumption and the other one is the dynamic power consumption. Static power consumption indicates how much power is consumed due to leakage currents, also indicates the power consumption caused by biasing currents of the full adder circuit. The occurring of dynamic power consumption is due to charging and discharging of the capacitor. It is observed that, in most of the cases dynamic power consumption is greater than static power consumption.

CADENCE simulation environment is a tremendous platform for calculating power consumption. By performing DC analysis, the power consumptions for various full adder topologies can be obtained. By doing DC analysis in CADENCE, the direct values of voltage and current are obtained at the power sources. Then by using the equation,

Power consumed = supplied voltage * generated current, the power consumptions are calculated for every single full adder topologies.

With the variation in full adder design style and the transistor count power consumption for different adder topologies also vary. The table below shows the values of power of different adder topologies which are simulated in CADENCE.

Table 4.1: Power consumption of different adder topologies

Full Adder Type	**Power Consumption (watt)**
Conventional	99.3182E-9
Transmission Gate	34.1499E-8
Fourteen Transistor	18.7115E-8
Gate Diffusion Input	27.9492E-7

4.5 Comparative analysis

Comparative analysis of various full adder based on different parameters are presented in table 4.2 and graphical representation is also shown for the results.

Table 4.2: Comparative Analysis of simulation output

Serial no.	Adder Type	No. of Transistors	Supply Voltage (V)	Technology	Delay (ns)		Power (Watt)	PDP (J)
					Input to sum Delay (ns)	Input to carry Delay (ns)		
1	CMOS	28	1.2	90nm	0.4636	0.4922	99.31E-9	4.61E-17
2	TGA	20	1.2	90nm	0.4175	0.2827	34.14E-8	1.43E-18
3	14T	14	1.2	90nm	0.4333	0.3384	18.71E-8	8.11E-19
4	GDI	10	1.2	90nm	0.5961	0.4070	279.4E-8	1.67E-15

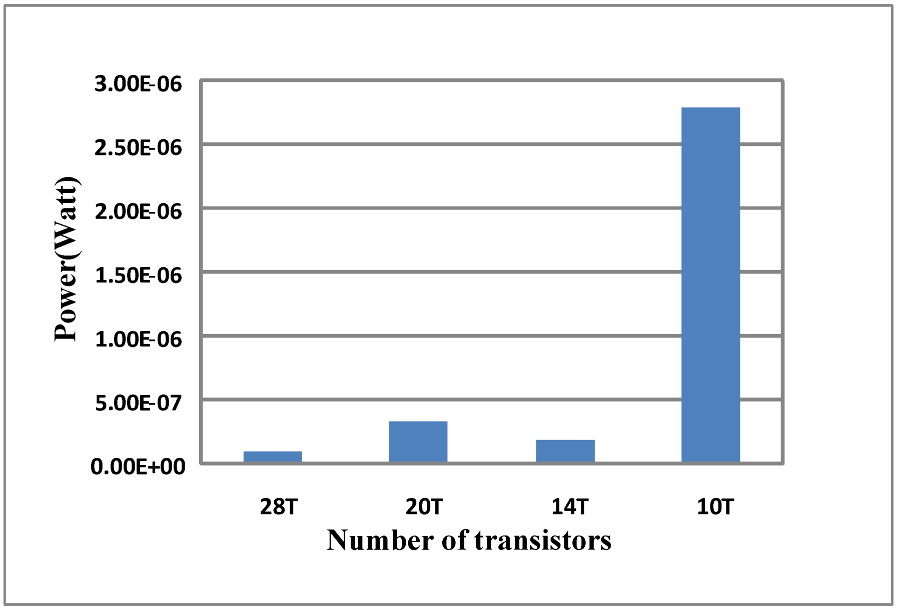


Fig 4.14 Graphical representation of power

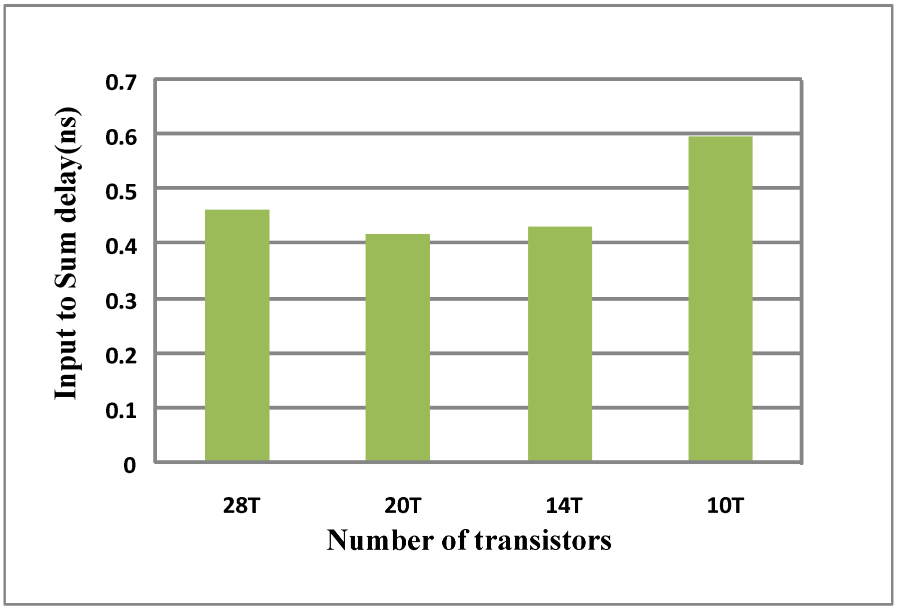


Fig 4.15 Graphical representation of average sum delay

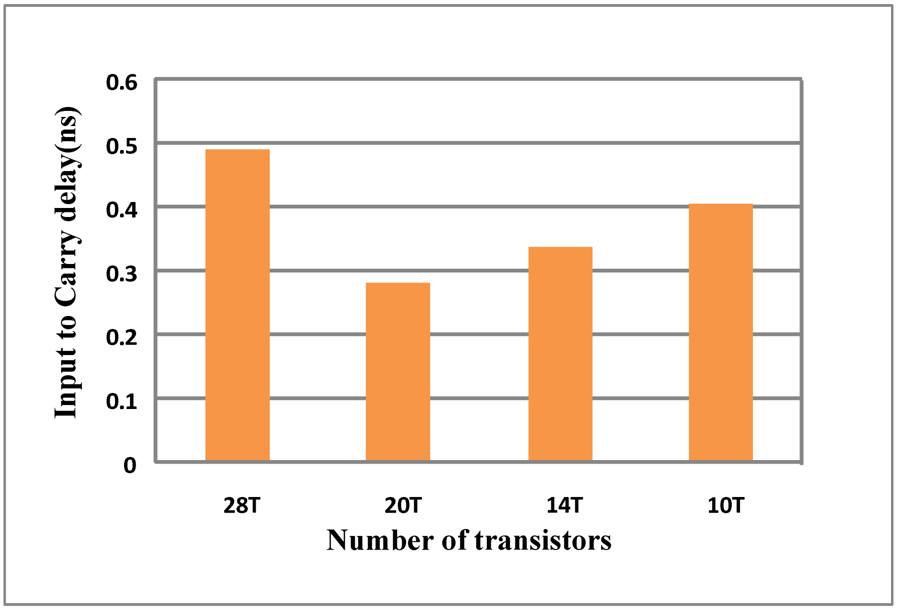


Fig 4.16 Graphical representations of average carry delay

Table 4.2 summarizes the comparison of various types of 1 bit full adder topologies. The table compares among different adder on the basis of the adder topology, transistor count, average propagation delay and the power consumption. As for each design there is a tradeoff between power and delay so PDP is taken for comparing overall performance of each adder.

From the performance analysis, it can be concluded that, if power consumption is taken as the main factor for design purpose then Conventional full adder is the best choice. If delay is the main factor, then transmission gate full adder is preferred. If area (Transistor Count) is considered as the main factor, then the designers can prefer GDI full adder design which has lesser Area. If PDP (Power Delay Product)is considered then 14T full adder is suggested.

Chapter 5

CONCLUSION

5.1 Analytical Observation

The main concern of our thesis has been comparing various full adders with different logic styles, to figure out, among the full adders which one promises to show the best performance. Since low-power and high-speed is one of the challenging criteria for digital signal processing chips, so it is compulsory that the device with high reliability and sustainability is chosen. There are various design styles for implementing full adder. From different points of view, different logic styles can be investigated. One selected style appropriate for a specific function may not be suitable for another one. For decades, researchers have invested their time to indicate various full adders at transient level, with different design topologies. Every time with new evaluation, verification and analysis, new observations are coming forward. Achieving the better performance with greater stability based on power and delay is the prime focus in most recent research works. Resonating with the same purpose, we also wanted to prove and demonstrate with necessary explanation provided in various chapters.

In this thesis paper, we described the performance of four full adders namely, Conventional full adder with 28 transistor, Transmission gate full adder with 20 transistor, 14 transistor adder using pass transistor logic & GDI based full adder comprising of 10 transistor. Each of which has both merits and demerits. All the matters are focused in a way so that it is beneficial for designing and useful in digital communication system.

Cadence 6.1.2 software is used to make the designs more accurate by fulfilling every aspect of design requirement. We also tried to improve the performance by changing the width and length of each topology. Simulation was done based on several parameters like full adder design topology, transistor count, average propagation delay, power consumption, power delay product. After simulation it was observed that, in terms of delay carried out by transient analysis, transmission gate full adder showed best performance among all others. Then, by dc analysis for power consumption, where obtained results showed greater performance for Conventional full adder. As power consumption depends on the transistor count, so theoretically the adder with lowest transistor count must show better result. But when we evaluated the power delay product by multiplying the values for delay and power, 14 transistor full adder showed better performance.

5.2 Future Work

The following works can be carried out in the future:

1) These full adder topologieshave been done in Cadence Virtuoso 90nm CMOS technology butin future 45 nm CMOS technology should be used.
2) In the near future, analysis should be more focused on reducing the noise that has not been done in this thesis paper.
3) In order to design the adders with more precision, layout design should be followed with various simulating tools provided in Cadence software.
4) It was observed from the simulation result that 14T shows better performance in terms of PDP, which is comprised of two techniques: Pass transistor and transmission gate logic. So, in the coming future, incorporating two or more different logic styles, a new adder can be designed which may give better performance.

Reference

[1] S. Ramola, “Digital Communication-Technology and Advancements”, in *Research India Publications* ,Volume 4, Number 4 (2014), pp. 367-374.

[2] A. J. Raj, and A. P. V, “Performance Improvement of ALU using D3 L Logic,” Saintgits College of Engineering, Kottayam-689648, India.

[3] N. H. E. Weste, and D. M. Harris,” CMOS VLSI Design A Circuits and Systems Perspective,” *Introduction,* chap.1, pg. 1-54

[4] M. M. Mano, ”Digital Logic And Computer Design,” *Combinational Logic*, chap.4,pg.114-149.of Adder Topologies

[5] T. L. Floyd, “Digital Fundamentals”, *Functions of Combinational Logic*, chap. 6, pg. 313-385

[6] J. Kaur, and L. Sood,” Comparison Between Various Types of Adder Topologies,” IJCST Vol. 6, Issue 1, 2015.

[7] Uma, V. Vijayan, and M. Mohanapriya, S. Paul, “Area, Delay and Power Comparison of Adder Topologies”, in *International Journal of VLSI Design & Communication Systems*, Vol. 3, No. 1, Feb. 2012, pp.153-168.

[8] M. K. M, A. Lateef, and A. Pagi, Ulaganathan,”Analysis of Various Full-Adder Circuits in Cadence”, in *International Journal of Computer Applications*, June 27,2016,pp 30-37

[9] K. Navi, and O. Kavehei, “Low-Power and High-Performance 1-Bit CMOS Full-Adder Cell”, in *JOURNAL OF COMPUTERS*, Vol. 3, No. 2, Feb. 2008, pp. 48-54.

[10] S. Panda1, A. Banerjee, and B. Maji, Dr.A.K.Mukhopadhyay, “Power and Delay Comparison in between different types of Full Adder Circuits,” in *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* Vol. 1, Issue 3, September 2012,pp. 168-172.

[11] S. Chauhan, and T. Sharma, “Full Adder Circuits using Static Cmos Logic Style: A Review”, in *International* Journal *of Computer Applications*

[12] D. Chaudhuri, A. Nag and S. Bose, “Low Power Full Adder Circuit Implemented In Different Logic,” in *National Conference on Emerging Technology and Applied Sciences-2014 (NCETAS 2014)* February,West Bengal, India,pp.124-129.

[13] A. Maheshwari, and S. Luthra, “Low Power Full Adder Circuit Implementation using Transmission Gate”, in *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 4, Issue 7, pp. 183-185, 2015.

[14] Y. S. G. Reddy, and V.V.G.S. R. Prasad, “Power Comparison Of CMOS And Adibatic Full Adder Circuits,” in *International Journal of Scientific & Engineering Research*, Volume 2, Issue 9, September-2011, ISSN 2229-5518.

[15] C. H. Chang, J. Gu, and M. Zhang, “A review of 0.18 um full adder performances for tree structured arithmetic circuits, in *IEE transactions on very large scale integration (vlsi) systems*, Vol. 13, No. 6, June 2015

[16] D. A. Pucknell, and K. Eshraghian, “ Basic VLSI Design”, *Basic Electrical Properties of MOS and BiCMOS Circuits*, chap. 2, pg. 25-55

[17] C. Khedhiri, M. Karmani, and B. Hamdi, “ A Differential Double Pass Transistor Logic Unit”, in *IJCSI International Journal of Computer Science Issues,* Vol. 9, Issue 2, No 1, March 2012.

[18] A. M. Shams, T. K. Darwish, and M. A. Bayoumi, “Performance analysisof low-power 1-bit CMOS full adder cells,” in IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 10, no. 1, pp. 20–29, Feb. 2002

[19] Manisha, and Archana, “A comparative study of full adder using static cmos logic style”, in *International Journal of Research in Engineering and Technology,* Vol. 3, Issue 6, June 2014

[20] P. Kumar, S. Mishra, and A. Singh, “Study of Existing Full Adders and To Design a LPFA (Low Power Full Adder)”, in*International Journal of Engineering Research and Applications (IJERA) ,*Vol. 3, Issue 3, May-Jun 2013, pp.509-513.

[21] M. Shoba, and R. Nakkeeran, “GDI based full adders for energy efficient arithmetic applications”,Engineering Science and Technology,an International Journal,2015.

[22] A. Rajender, Mr. P. Ramesh, and M. Tech, “Design& Study of a Low Power High Speed Full Adder Using 10Transistor GDI Multiplexer, in *International Journal And Magazine Of Engineering,Technology,Management And Research*,Vol.3,September 2016,pp.568-572.

[23] A. Saberkari, and S. B. Shokouhi , “A Novel Low-Power Low-Voltage CMOS 1-Bit Full Adder Cell with the GDI Technique”, presented at ME - INTERTECH Conference, 2006