JES
Journal of Engineering Sciences

Vol 10, Issue 12, DEC/2019
ISSN NO:0377-9254

DESIGN OF N- BIT ADVANCED ADDERS USING MODIFIED MULTIPLEXERS

[1]VENKATA SAI DEDEEPYA BOYA, [2]M. GANESH KUMAR, [3]S. MOHAN DAS

[1]M.Tech Student, [2]Assistant Professor, [3]Associate Professor

DEPT OF ECE

SVR Engineering College, Nandyal

ABSTRACT

Adders play a vital role in the digital signal processing systems. The design of 32-bit adders is of high importance because 32-bit architecture is common and widely used in many digital systems and processors. In this paper, the design and the implementation of various 32-bit adders like Ripple Carry Adder (RCA), Carry Increment adder (CINA) and Carry bypass adder (CBYA) for different full adder cells is done using the Microwind tool. The results are obtained by executing circuits designed by various gates in Microwind and DSCH software.

I. INTRODUCTION

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip.These signal processing applications not only demand great computation capacity but also consume considerable amount of energy. While performance and Area remain to be the two major design tolls, power consumption has become a critical concern in today's VLSI system design[]. The need for low-power VLSI system arises from two main forces. First, with the steady growth of operating frequency and processing capacity per chip, large currents have to be delivered and the heat due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these portable devices.

Addition usually impacts widely the overall performance of digital systems and a crucial arithmetic function. In electronic applications adders are most widely used. Applications where these are used are multipliers, DSP to execute various algorithms like FFT, FIR and IIR. Wherever concept of multiplication comes adders come in to the picture. As we know millions of instructions per second are performed in microprocessors. So, speed of operation is the most important constraint to be considered while designing multipliers. Due to device portability miniaturization of device should be high and power consumption should be low. Devices like Mobile, Laptops etc. require more battery backup.

So, a VLSI designer has to optimize these three parameters in a design. These constraints are very difficult to achieve so depending on demand or application some compromise between constraints has to be made. Ripple carry adders exhibits the most compact design but the slowest in speed. Whereas carry look ahead is the fastest one but consumes more area. Carry select adders act as a compromise between the two adders. In 2002, a new concept of hybrid adders is presented to speed up addition process by Wang et al. that gives hybrid carry look-ahead/carry select adders design. In 2008, low power

multipliers based on new hybrid full adders is presented.

DESIGN of area- and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.

The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum.

1.1. NEED FOR LOW POWER DESIGN

The design of portable devices requires consideration for peak power consumption to ensure reliability and proper operation. However, the time averaged power is often more critical as it is linearly related to the battery life. There are four sources of power dissipation in digital CMOS circuits: switching power, short-circuit power, leakage power and static power. The following equation describes these four components of power:

$$P_{avg} = P_{switching} + P_{short-circuit} + P_{leakage} + P_{static} \quad (2.1)$$

$$= \alpha C_L V_{dd} V_s f_{ck} + I_{sc} V_{dd} + I_{leakage} V_{dd} + I_{static} V_{dd} \quad (2.2)$$

P switching is the switching power. For a properly designed CMOS circuit, this power component usually dominates, and may account for more than 90% of the total power. α denotes the transition activity factor, which is defined as the average number of power consuming transitions that is made at a node in one clock period. Vs is the voltage swing, where in most cases it is the same as the supply voltage, Vdd. CL is the node capacitance. It can be broken into three components, the gate capacitance, the diffusion capacitance, and the interconnect capacitance. The interconnect capacitance is in general a function of the placement and routing. fck is the frequency of clock. The switching power for static CMOS is derived as follows.

During the low to high output transition, the path from Vdd to the output node is con-ducting to charge CL. Hence, the energy provided by the supply source is

$$I(t) = \frac{V_s}{R} e^{-t/RC_L}$$

$$E = \int_0^\infty V_{dd} I(t) dt \quad (2.3)$$

where is the current drawn from the supply. Here, R is the resistance of the path between the Vdd and the output node. Therefore, the energy can be rewritten as

$$E = C_L V_{dd} V_s \quad (2.4)$$

During the high to low transition, no energy is supplied by the source. Hence, the average power consumed during one clock cycle is

$$P = \frac{E_{percycle}}{T} = C_L V_{dd} V_s f_{ck} \quad (2.5)$$

Eq. (2.4) and Eq. (2.5) estimate the energy and the power of a single gate only. From a system point of view, α is used to account for the actual number of gates switching at a point in time.

P short circuit is the short-circuit power. It is a type of dynamic power and is typically much smaller than P switching. Isc is known as the direct-path short circuit current. It refers to the conducting current from power supply directly to ground when

both the NMOS and PMOS transistors are simultaneously active during switching.

P leakage is the leakage power. I leakage refers to the leakage current. It is primarily determined by fabrication technology considerations and originates from two sources. The first is the reverse leakage current of the parasitic drain-/source-substrate diodes. This current is in the order of a few femto amperes per diode, which translates into a few microwatts of power for a million transistors. The second source is the sub threshold current of MOSFETs, which is in the order of a few nano amperes. For a million transistors, the total subthreshold leakage current results in a few milli watts of power.

P static is the static power and Istatic is static current. This current arises from circuits that have a constant source of current between the power supplies such as bias circuitries, pseudo-NMOS logic families. For CMOS logic family, power is dissipated only when the circuits switch, with no static power consumption.

Energy is independent of the clock frequency. Reducing the frequency will lower the power consumption but will not change the energy required to perform a given operation, as depicted by Eq. (2.4) and Eq. (2.5). It is important to note that the battery life is determined by energy consumption, whereas the heat dissipation considerations are related to the power consumption.

There are four factors that influence the power dissipation of CMOS circuits. They are technology, circuit design style, architecture, and algorithm. The challenge of meeting the contradicting goals of high performance and low power system operation has motivated the development of low power process technologies and the scaling of device feature sizes.

Design considerations for low power should be carried out in all steps in the design hierarchy, namely 1) Fundamental, 2) material, 3) device, 4) circuit, and 5) system.

LOW VOLTAGE

Power consumption is linearly proportional to voltage swing (Vs) and supply voltage (Vdd) as indicated in Eq. (2.5). For most CMOS logic families, the swing is typically rail-to-rail. Hence, power consumption is also said to be proportional to the square of the supply voltage, Vdd. Therefore, lowering the Vdd is an efficient approach to reduce both energy and power, presuming that the signal voltage swing can be freely chosen. This is, however, at the expense of the delay of circuits. The delay, td, can be shown to be proportional to $V_{dd}/(V_{dd} - V_T)^{\gamma}$ .The exponent γ is between 1 and 2. It tends to be closer to 1 for MOS transistors that are in deep sub-micrometer region, where carrier velocity saturation may occur. γ increases toward 2 for longer channel transistors.

The current technology trends are to reduce feature size and lower supply voltage. Lowering Vdd leads to increased circuit delays and therefore lower functional throughput. Smaller feature size, however, reduces gate delay, as it is inversely proportional to the square of the effective channel length of the devices. In addition, thinner gate oxides impose voltage limitation for reliability reasons. Hence, the supply voltage must be lowered for smaller geometries. The net effect is that circuit performance improves as CMOS technologies scale down, despite of the Vdd reduction. Therefore, the new technology has made it possible to fulfill the contradicting requirements of low-power and high throughput.

The various techniques that are currently used to scale the supply voltage include optimizing the technology and device reliability, trading off area for low power in architecture driven approach, and exploiting the concurrency possibility in algorithmic transformations. Hence, the voltage scaling is limited by the threshold voltage Vth.

In applications such as digital processing, where the throughput is of more concern than the speed, architecture can be designed to reduce the supply voltage at the expense of speed without throughput degradation. Hence, the performance of the system can be maintained.

1.3 LANGUAGE AND TOOLS USED

- Microwind and DSCH
- Verilog (HDL)

1.4 ADVANATGES

- Low power consumption
- Less area (less complexity)
- More speed compare regular CSLA

1.5 APPLICATIONS

- Arithmetic logic units
- High Speed multiplications
- Advanced microprocessor design
- Digital signal process

II. ADDERS

RIPPLE CARRY ADDERS (RCA)

Concatenating the N full adders forms N bit Ripple carry adder. In this carry out of previous full adder becomes the input carry for the next full adder. It calculates sum and carry according to the following equations. As carry ripples from one full adder to the other, it traverses longest critical path and exhibits worst-case delay. Si = Ai xor Bi xorCi

Ci+1 = Ai Bi + (Ai + Bi) Ci; where i = 0, 1, …, n-1

RCA is the slowest in all adders (O (n) time) but it is very compact in size (O (n) area). If the ripple carry adder is implemented by concatenating N full adders, the delay of such an adder is 2N gate delays from Cin to Cout. The delay of adder increases linearly with increase in number of bits. Block diagram of RCA is shown in figure 1.

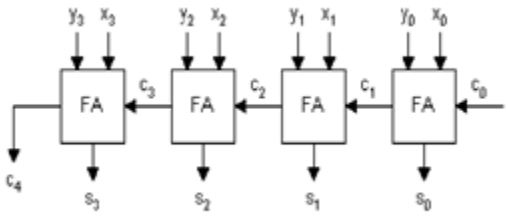


Figure:1.Block diagram of RCACARRY SKIP ADDER (CSKA)

A carry skip divides the words to be added in to groups of equal size of k-bits. Carry Propagate *pi* signals may be used within a group of bits to accelerate the carry propagation. If all the pi signals within the group are pi=1, carry bypasses the entire group as shown in figure 2.

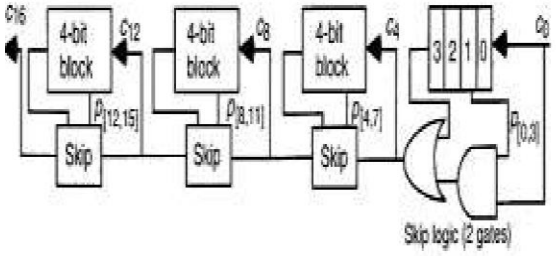
P = pi * pi+1 * pi+2 *… pi+k

In this way delay is reduced as compared to ripple carry adder. The worst-case carry propagation delay in a N-bit carry skip adder with fixed block width *b*, assuming that one stage of ripple has the same delay as one skip, can be derived:

TCSKA = (b -1)+0.5+(N/b-2)+(b -1) = 2b + N/b – 3.5 Stages

Block width tremendously affects the latency of adder. Latency is directly proportional to block width. More number of blocks means block width is less, hence more delay. The idea behind Variable Block Adder (VBA) is to minimize the critical path

delay in the carry chain of a carry skip adder, while allowing the groups to take different sizes. In case of carry skip adder, such condition will result in more number of skips between stages.



Such adder design is called variable block design, which is tremendously used to fasten the speed of adder. In the variable block carry skip adder design we divided a 32-bit adder in to 4 blocks or groups. The bit widths of groups are taken as; First block is of 4 bits, second is of 6 bits, third is 18 bit wide and the last group consist of most significant 4 bits.

Table 1 shows that the logic utilization of carry skip and variable carry skip 32-bit adder. The power and delay, which are obtained also given in the table1. From table it can be observed that variable block design consumes more area as gate count and number of LUT's consumed by variable block design is more than conventional carry skip adder.

CARRY SELECT ADDER (CSA)

The carry select adder comes in the category of conditional sum adder. Conditional sum adder works on some condition. Sum and carry are calculated by assuming input carry as 1 and 0 prior the input carry comes. When actual carry input arrives, the actual calculated values of sum and carry are selected using a multiplexer. The conventional carry select adder consists of k/2 bit adder for the lower half of the bits i.e. least significant bits and for the upper half i.e. most significant bits (MSB's) two k/ bit adders. In MSB adders one adder assumes carry input as one for performing addition and another assumes carry input as zero. The carry out calculated from the last stage i.e. least significant bit stage is used to select the actual calculated values of output carry and sum. The selection is done by using a multiplexer. This technique of dividing adder in to stages increases the area utilization but addition operation fastens. The block diagram of conventional k bit adder is shown in figure 3.

III. EXISTING SYSTEM

CARRY SELECT ADDER

DESIGN of area- and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated in to the next position .The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. However ,the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input cin=0 and cin=1, then the final sum and carry are selected by the multiplexers (mux).The basic idea of this work is to use Binary to Excess-1 Converter(BEC) instead of RCA with cin=1 in the regular CSLA to achieve lower area and power consumption The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure.

The carry select adder comes in the category of conditional sum adder. Conditional sum adder works on some condition. Sum and carry are calculated by assuming input carry as 1 and 0 prior the input carry comes. When actual carry input arrives, the actual calculated values of sum and carry are selected using a multiplexer. The conventional carry select adder consists of k/2 bit adder for the lower half of the bits i.e. least significant bits and for the upper half i.e. most significant bits (MSB's) two k/ bit adders. In MSB adders one adder assumes carry input as one for performing addition and another assumes carry input as zero. The carry out calculated from the last stage i.e. least significant bit stage is used to select the actual calculated values of output carry and sum. The selection is done by using a multiplexer. This technique of dividing adder in to stages increases the area utilization but addition operation fastens.

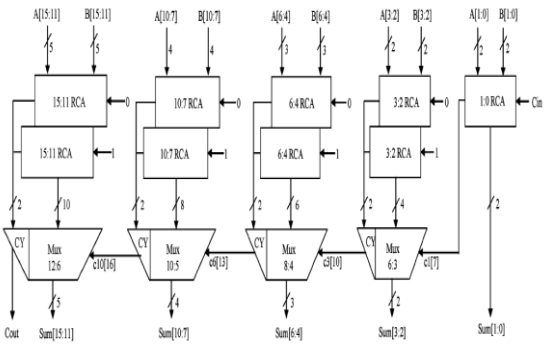


Fig.2.Regular 16-b SQRT CSLA.

IV. MULTIPLEXER

In electronics, a multiplexer (or MUX) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line.[1] A multiplexer of 2n inputs has n select lines, which are used to select which input line to send to the output.[2] Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth.[1] A multiplexer is also called a data selector. They are used in CCTV, and almost every business that has CCTV fitted, will own one of these.

An electronic multiplexer makes it possible for several signals to share one device or resource, for example one A/D converter or one communication line, instead of having one device per input signal.

On the other hand, a de multiplexer (or demux) is a device taking a single input signal and selecting one of many data-output-lines, which is connected to the single input. A multiplexer is often used with a complementary demultiplexer on the receiving end.[1]

An electronic multiplexer can be considered as a multiple-input, single-output switch, and a demultiplexer as a single-input, multiple-output switch.[3] The schematic symbol for a multiplexer is an isosceles trapezoid with the longer parallel side containing the input pins and the short parallel side containing the output pin.[4] The schematic on the right shows a 2-to-1 multiplexer on the left and an equivalent switch on the right. The wire connects the desired input to the output.

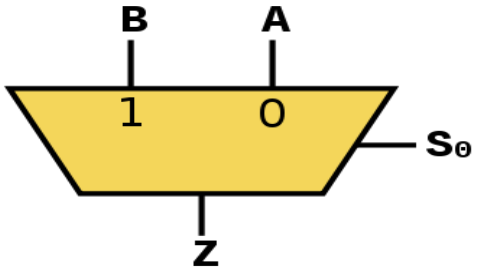
In digital circuit design, the selector wires are of digital value. In the case of a 2-to-1 multiplexer, a logic value of 0 would connect I_0 to the output while a logic value of 1 would connect I_1 to the output. In larger multiplexers, the number of selector pins is equal to $\lceil \log_2(n) \rceil$ where n is the number of inputs.

For example, 9 to 16 inputs would require no fewer than 4 selector pins and 17 to 32 inputs would require no fewer than 5 selector pins. The

binary value expressed on these selector pins determines the selected input pin.

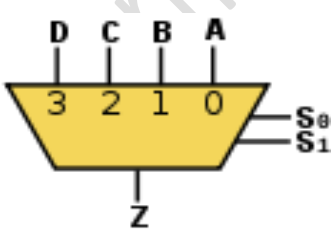
A 2-to-1 multiplexer has a boolean equation where A and B are the two inputs, S is the selector input, and Z is the output:

$$Z = (A \cdot \overline{S}) + (B \cdot S)$$



This truth table shows that when $S=0$ then $Z=A$ but when $S=1$ then $Z=B$. A straightforward realization of this 2-to-1 multiplexer would need 2 AND gates, an OR gate, and a NOT gate.

Larger multiplexers are also common and, as stated above, require $\lceil \log_2(n) \rceil$ selector pins for n inputs. Other common sizes are 4-to-1, 8-to-1, and 16-to-1. Since digital logic uses binary values, powers of 2 are used (4, 8, 16) to maximally control a number of inputs for the given number of selector inputs.



4-to-1 mux

The boolean equation for a 4-to-1 multiplexer is:

$$F = (A \cdot \overline{S_0} \cdot \overline{S_1}) + (B \cdot S_0 \cdot \overline{S_1}) + (C \cdot \overline{S_0} \cdot S_1) + (D \cdot S_0 \cdot S_1)$$

DELAY AND AREA EVALUATION OF BASIC BLOCKS

The AND, OR, and Inverter (AOI) implementation of an XOR gate is shown in Fig.below. The gates between the dotted lines are performing the operations in parallel and the numeric representation of each gate indicates the delay contributed by that gate. The delay and area evaluation methodology considers all gates to be made up of AND, OR, and Inverter, each having delay equal to 1 unit and area equal to 1 unit. We then add up the number of gates in the longest path of a logic block that contributes to the maximum delay.

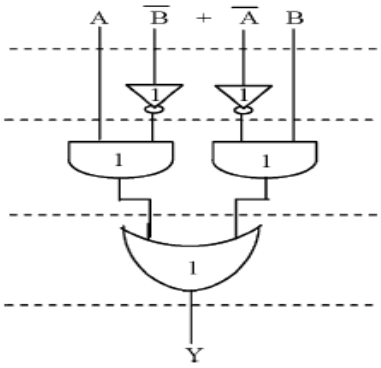


Fig 3.Area and Delay evaluation of xor gate

The area evaluation is done by counting the total number of AOI gates required for each logic block. Based on this approach, the CSLA adder blocks of 2:1 mux, Half Adder (HA), and FA are evaluated and listed in Table I.

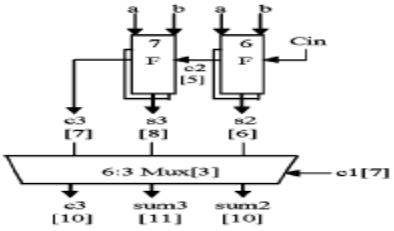
JES
Journal of Engineering Sciences
Vol 10, Issue 12, DEC/2019
ISSN NO:0377-9254

TABLE I
DELAY AND AREA COUNT OF THE BASIC BLOCKS OF CSLA

Adder blocks	Delay	Area
XOR	3	5
2:1 Mux	3	4
Half adder	3	6
Full adder	6	13

DELAY AND AREA EVALUATION OF REGULAR CSLA

The structure of the 16-b regular SQRT CSLA has five groups of different size RCA. The delay and area evaluation of each group are shown in Fig, in which the numerals within specify the delay values, e.g., sum2 requires 10 gate delays. The steps leading to the evaluation are as follows.



Group2 for delay and area evaluation

1) The group2 has two sets of 2-b RCA. Based on the consideration of delay values of Table I, the arrival time of selection input of 6:3 mux is earlier than and later than. Thus ,is summation of and 2) Except for group2, the arrival time of mux selection input is always greater than the arrival time of data outputs from the RCA's. Thus, the delay of group3 to group5 is determined, respectively as follows: 3) The one set of 2-b RCA in group2 has 2 FA for and the other set has 1 FA and 1 HA for . Based on the area count of Table I, the total number of gate counts in group2 is determined as follows:

$$\text{Gate count} = 57\ (\text{FA} + \text{HA} + \text{Mux})$$
$$\text{FA} = 39(3 * 13)$$
$$\text{HA} = 6(1 * 6)$$
$$\text{Mux} = 12(3 * 4).$$

4) Similarly, the estimated maximum delay and area

TABLE III
DELAY AND AREA COUNT OF REGULAR SQRT CSLA GROUPS

Group	Delay	Area
Group2	11	57
Group3	13	87
Group4	16	117
Group5	19	147

of the other groups in the regular SQRT CSLA are evaluated and listed

PROBLEM WITH REGULAR CARRY SELECT ADDER

- The problem in CSLA design is the number of full adders are increased then the circuit complexity also increases.
- The number of full adder cells are more thereby power consumption of the design also increases
- Number of full adder cells doubles the area of the design also increased.

MODIFIED REGULAR CSLA USING BEC

The main idea of this work is to use BEC instead of the RCA with Cin=1in order to reduce the area and power consumption of the regular CSLA. To replace the n bit RCA, an n+1 bit BEC is required. A structure and the function table of a 4 bit BEC are show in Fig and Table II respectively.

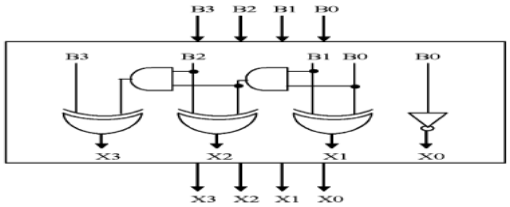


Figure:4. 4-bit BEC

TABLE II
FUNCTION TABLE OF THE 4-b BEC

B[3:0]	X[3:0]
0000	0001
0001	0010
⋮	⋮
1110	1111
1111	0000

Fig. above illustrates how the basic function of the CSLA is obtained by using 4-bit BEC together with the mux. One input of the 8:4 mux gets as it input((B3, B2, B1, and B0) and another input of the mux is the BEC output. This produces the two possible partial results in parallel and the mux is used to select either the BEC output or the direct inputs according to the control signal Cin. The importance of the BEC logic stems from the large silicon area reduction when the CSLA with large number of bits are designed. The Boolean expressions of the 4-bit BEC is listed as (note the functional symbols NOT, & XOR)

$$X0 = \sim B0$$
$$X1 = B0 \hat{} B1$$
$$X2 = B2 \hat{} (B0 \,\&\, B1)$$
$$X3 = B3 \hat{} (B0 \,\&\, B1 \,\&\, B2).$$

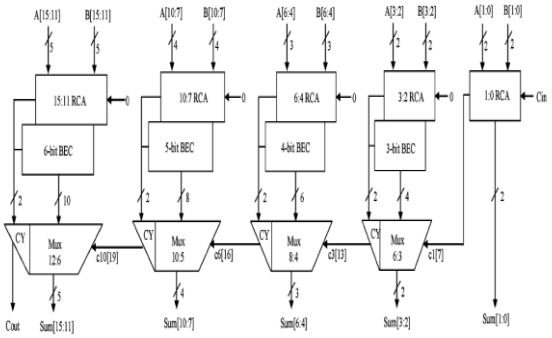


Figure:5.Modified SQRT Carry Select Adder The parallel RCA with Cin=1 is replaced with BEC

DELAY AND AREA EVALUATION OF MODIFIED CSLA

The structure of the proposed 16-b SQRT CSLA using BEC for RCA with Cin=1 to optimize the area and power is shown in above fig. We again split the structure into five groups. The delay and area estimation of each group are shown in Fig below. The steps leading to the evaluation are given here.

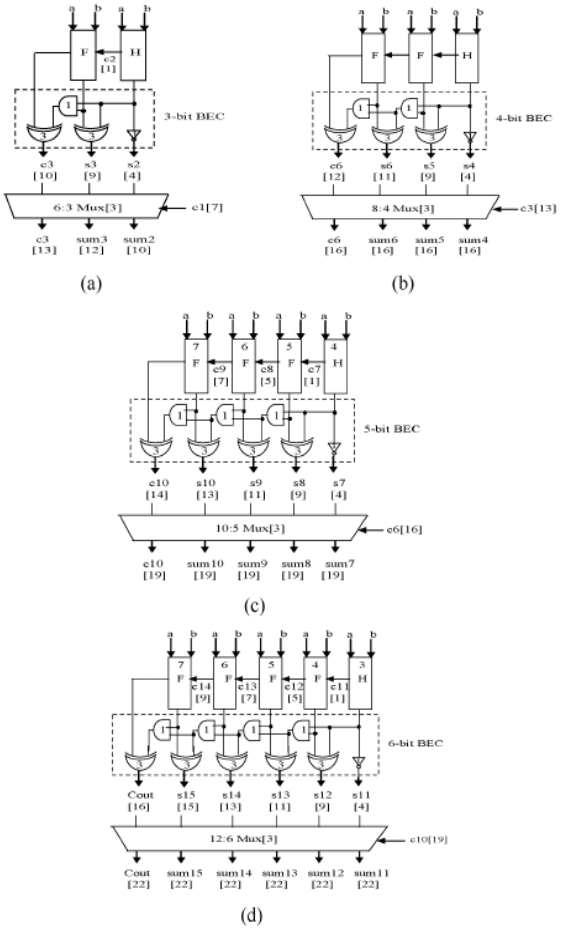


Figure:6. Delay and area evaluation of modified SQRT CSLA: (a) group2, (b) group3, (c) group4, and (d) group5. H is a Half Adder.

The group2 has one 2-b RCA which has 1 FA and 1 HA for Cin=0 Instead of another 2-b RCA with Cin=1 a 3-b BEC is used which adds one to the output from 2-b RCA. Based on the consideration of delay values of Table I, the arrival time of selection input C1 [t=7] of 6:3 mux is earlier than the s3 [t=9] and C3[t=10] and later than theS2[t=4]. Thus, the sum3 and finalC3 (output from mux) are depending on S3 and mux and partial C3 (input to mux) and mux, respectively. The sum2 depends on C1 and mux.

1) The area count of group2 is determined as follows:

$$\text{Gate count} = 43\ (\text{FA} + \text{HA} + \text{Mux} + \text{BEC})$$
$$\text{FA} = 13(1 * 13)$$
$$\text{HA} = 6(1 * 6)$$
$$\text{AND} = 1$$
$$\text{NOT} = 1$$
$$\text{XOR} = 10(2 * 5)$$
$$\text{Mux} = 12(3 * 4).$$

3) Similarly, the estimated maximum delay and area of the other groups of the modified SQRT CSLA are evaluated and listed in Table IV.

TABLE IV
DELAY AND AREA COUNT OF MODIFIED SQRT CSLA

Group	Delay	Area
Group2	13	43
Group3	16	61
Group4	19	84
Group5	22	107

PROPOSED SQRT CSLA USING COMMON BOOLEAN LOGIC:

To remove the duplicate adder cells in the conventional CSLA, an area efficient SQRT CSLA is proposed by sharing Common Boolean Logic (CBL) term. While analyzing the truth table of single bit full adder, results show that the output of summation signal as carry-in signal is logic "0" is inverse signal of itself as carry-in signal is logic "1". It is illustrated by red circles in Table II. To share the Common Boolean Logic term, we only need to implement a XOR gate and one INV gate to generate the summation pair. And to generate

the carry pair, we need to implement one OR gate and one AND gate. In this way, the summation and carry circuits can be kept parallel.

TABLE III

TRUTH TABLE OF SINGLE BIT FULL ADDER, WHERE THE UPPER HALF PART IS THE CASE OF CIN=0 AND THE LOWER HALF PART IS THE CASE OF CIN=1

Cin	A	B	S0	C0
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	0	0	1
1	1	1	1	1

This method replaces the Binary to Excess-1 converter add one circuit by common Boolean logic. As compared with modified SQRT CSLA, the proposed structure is little bit faster. Internal structure of proposed CSLA is shown in Fig. 8.

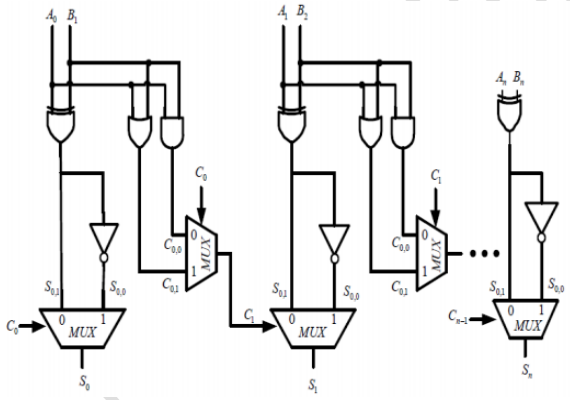


Fig.7. Internal structure of the proposed area-efficient carry select adder is constructed by sharing the common Boolean logic term.

In the proposed SQRT CSLA, the transistor count is trade-off with the speed in order to achieve lower power delay product. Thus the proposed SQRT CSLA using CBL is better than all the other designed adders. Fig. 9 shows the Block diagram of Proposed SQRT CSLA.

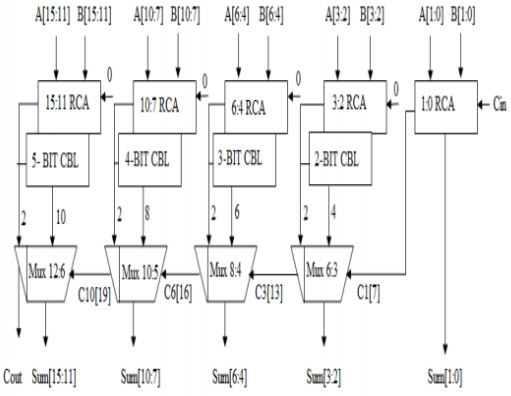


Fig. 8 16-Bit Proposed SQRT CSLA using Common Boolean Logic.

PROPOSED METHOD

In digital circuits, the arithmetic circuits are of high significance. Adders [1],[2],[3],[4] play a vital role in the efficient implementation of the arithmetic circuit. All arithmetic operations can be implemented by using an adder circuit by following a suitable logic. An adder is a circuit that performs addition of two numbers and gives sum and carryout as results [4].The basic adders cells used are the full adder and the half adder which add three and two inputs respectively [4]. There are many ways of implementation of full adder cell which are efficient than conventional full adder cell. In most of the digital systems like computers or processors, there exists 32-bit architecture[5] such as 32-bit integers, registers, memory addresses etc. So the optimization of 32-bit adders reduces the total delay and hence speed increases[5]. In this paper, an attempt is made to design and compare the performance of different 32-bit adders using different full adder cells in the terms of slices, look up tables, delay and fan-out

using Verilog language Xilinx ISE 14.5 Spartan 3E family device with speed grade -5. This paper has 5 sections that is --- section I contains about brief introduction on the adder, section II deals with literature survey on full adder cells, section III contains the design of 32-bit adders, section IV deals with the simulation results, section V has performance analysis of RCA,CINA and CBYA for various full adder cells, section VI has the conclusion about this work and section VII has references of this work.

V. LITERATURE SURVEY

A. **Conventional Full Adder (FA):** The traditional full adder circuit is obtained using

B. 2 XOR gates, 3AND gates and an OR gate[4]. The Boolean expressions used are z sum = a⊕ b⊕ cin (1) z carry=(a · b)+(b· cin)+(cin · a) (2)

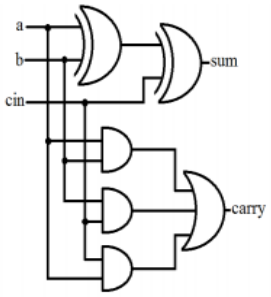
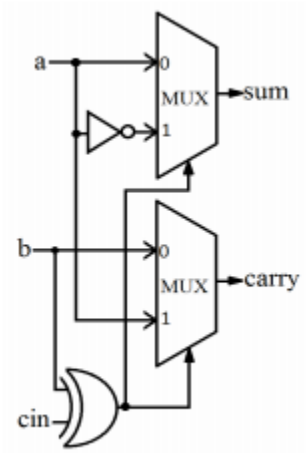


Fig.1 Conventional full adder cell[4] (FA)

C. Modified full adder- 1 (FA1):

Prof. S.Murugeswari and Dr. S.Kaja Mohideen have proposed this model. In this work, the detailed study about Wallace tree multiplier and truncated multipliers is done with both normal full adder and modified full adder-1 and it is shown that modified full adder-1 implementation occupies the less area[6].



ig.2 Modified full adder-1 cell[6] (FA1)

D. Modified full adder- 2 (FA2):

Kokila Bharti Jaiswal et al., proposed this model[7]. In this work, the detailed study about Wallace tree multiplier with conventional full adder, modified full adder-1 and modified full adder- 2 and it is shown that modified full adder3 implementation occupies the less area and less power[7].

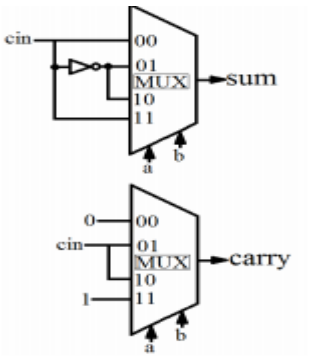


Fig.3 Modified full adder-2 cell[7] (FA2)

D. Modified full adder- 3(FA3): Sreehari Veeramachaneni et al., have proposed this model[8]. It is also shown that the use of modified full adder-3 performs better in terms of delay and area [8].

Journal of Engineering Sciences
Vol 10, Issue 12, DEC/2019
ISSN NO:0377-9254

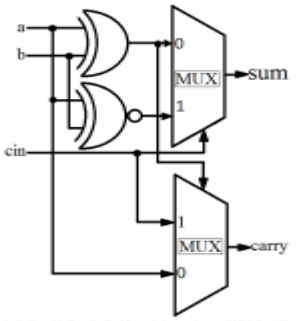


Fig.4 Modified full adder-3 cell[8] (FA3)

DESIGN OF 32-BIT ADDERS

A. Ripple Carry Adder (RCA):

Ripple Carry Adder is a basic adder circuit which contains individual full adder cells and the carry generated upon addition is propagated between the respective adder cells[4]. The computation of result takes place only after the carry from the previous stage is applied as an input to present stage[4]. Due to these propagation delays, the delay is bound to occur which is a major disadvantage. The architecture of the 32-bit RCA is shown in Fig.5 .

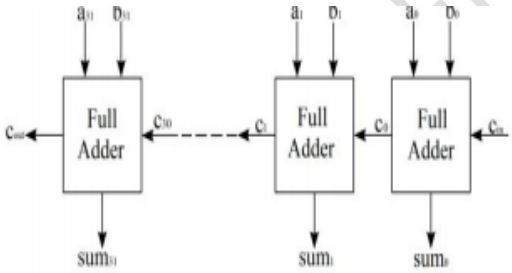


Fig.5 32-bit ripple carry adder

B. Carry Increment Adder (CINA):

In carry increment adder, various RCA blocks are used to compute the results. The first RCA block is given carry-in as input along with addends to get carry(c1) and sum[9],[10]. For the rest RCA blocks, the carry-in is given as logic '0' to get temporary sum (sum1) and temporary carry which are given to increment circuit. Increment circuit consists of half adders which add temporary sum and carries to get the actual sum(sum) and carry (cy). The carry-out of increment circuit is obtained by performing OR operation between carry (cy) and carry-out of the previous stage. As the carry-in for the RCA stages is logic '0' and hence the carry propagation delay decreases[10],[11],[12]. The architecture for the CINA 32bit is given in Fig.6 .

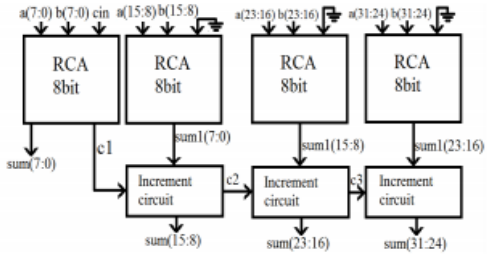


Fig.6 32-bit carry increment adder

C. Carry Bypass Adder (CBYA):

In CBYA, the carry skip logic block is used to compute the carry of next stage. The Boolean expression implemented in carry skip logic block[10],[11],[12] is z p[n+3:n] = a[n+3:n]⊕ b[n+3:n] (3) z sel[n] = p[n+3] • p[n+2]• p[n+1]• p[n] (4) The multiplexer is used to select either carry-in or carry generated in RCA block depending on the select line (sel[n]). In this adder, the carry-out is equal to input carry if and only if p[0] • p[1]• ....p[31] is '1' that is input carry is bypassed to next stage[13] and carry-out is equal to carry from RCA if p[0] • p[1]• ....p[31] is '0'[10] . The architecture of 32 bit CBYA is given in Fig.7. When the carry is bypassed between successive stages, net propagation delay decreases[13].

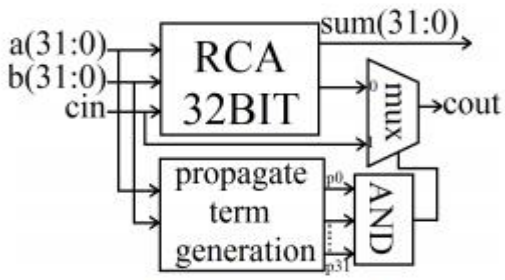
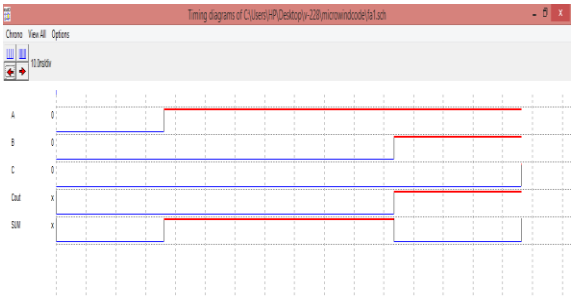
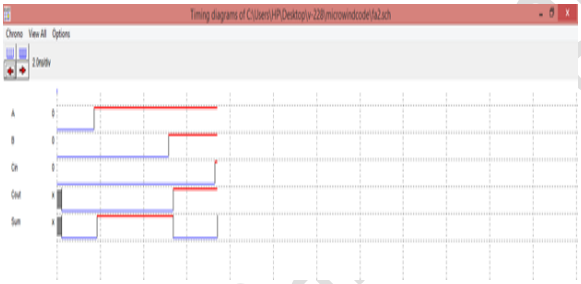


Fig.7 32-bit carry bypass adder

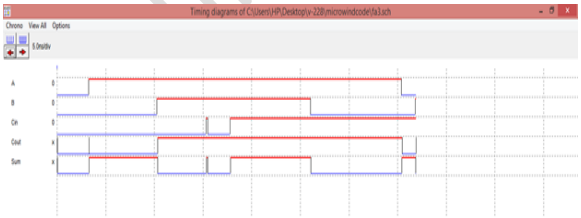
VI. SIMULATION RESULT



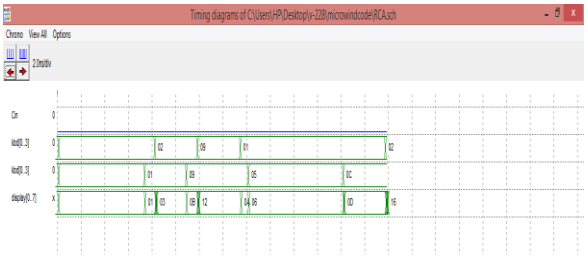
Fa1 output



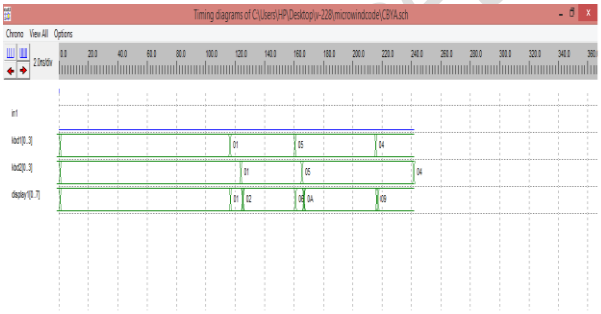
Fa2 output



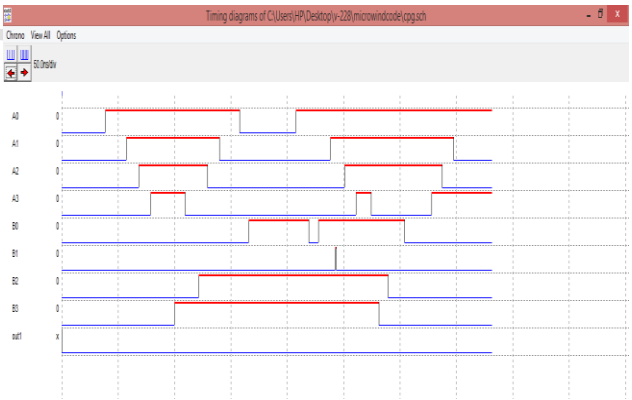
Fa3 output

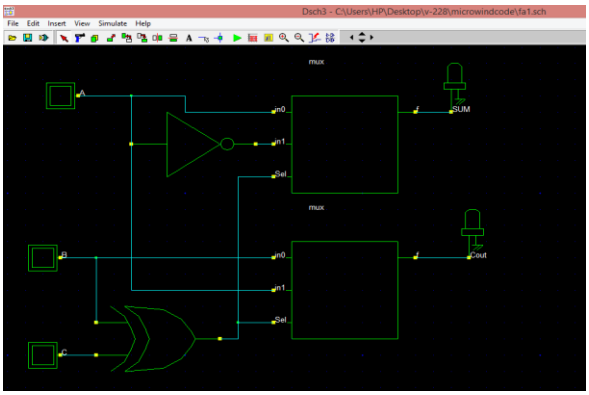


RCA output

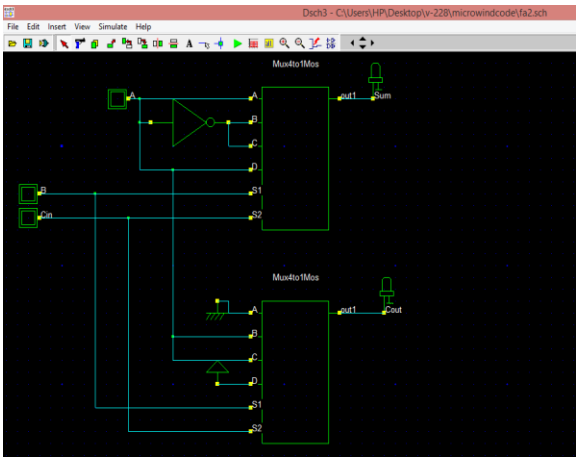


CBYA Output

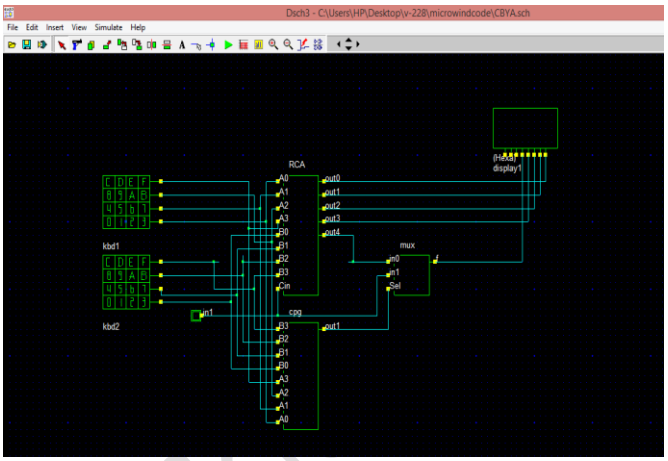




FA1 RTL SCHEMATIC

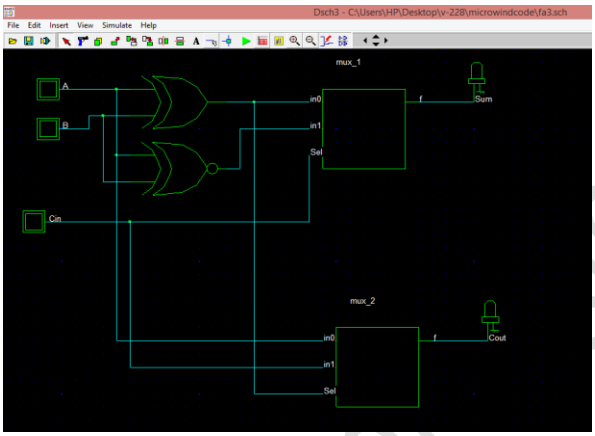


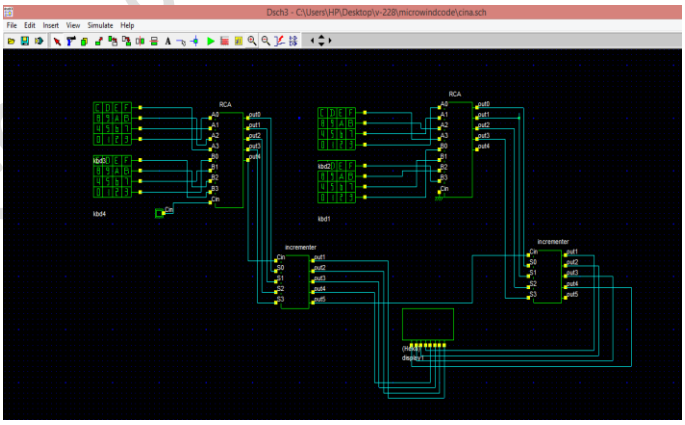
RCA



FA2 RTL

CBYA





FA3

CINA

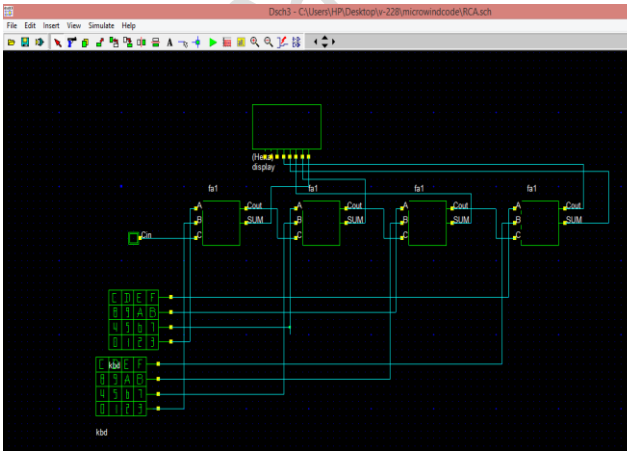


Table1 : Comparison for various 32-bit adders with FA

	LUTs	Slices	Delay(ns)	Fan-out
RCA (FA)	64	37	38.665	1.74
CINA (FA)	88	50	26.57	2.15
CBYA (FA)	101	56	25.514	2.53

Table2 : Comparison for various32-bit adders with FA1

	LUTs	Slices	Delay(ns)	Fan-out
RCA (FA1)	64	37	37.456	1.74
CINA (FA1)	88	51	26.336	2.15
CBYA (FA1)	102	57	24.63	2.62

Table3 : Comparison for various32-bit adders with FA2

	LUTs	Slices	Delay(ns)	Fan-out
RCA (FA2)	64	37	38.665	1.74
CINA (FA2)	88	50	26.57	2.15
CBYA (FA2)	102	57	26.489	2.55

Table4 : Comparison for various32-bit adders with FA3

	LUTs	Slices	Delay(ns)	Fan-out
RCA (FA3)	64	37	38.665	1.74
CINA (FA3)	92	52	26.354	2.16
CBYA (FA3)	102	57	26.465	2.5

VII. CONCLUSION

From above results, it is evident that the LUTs are less for RCA. The LUTs increased by 38% and 59% for CINA and CBYA respectively. The slices (area) are increased by 37% and 54% for CINA and CBYA respectively when compared with the RCA. The more the fan-out the more the load driving capacity[14]. The fan-out is increased by 23% and 46% for CINA and CBYA respectively when compared with the RCA. However, CBYA with FA1 has 36% less delay than the RCA with FA and hence provides the least delay. It is also evident that the delay produced by opting to modified full adder 2 results in decrease of the delay for RCA, CINA and CBYA . Hence for implementation of RCA, CINA and CBYA, the use of modified full adder as basic cell will improve the performance in terms of the delay provided.

FUTURE SCOPE

This work has been designed for 8-bit, 16-bit, 32-bit and 64-bit word size and results are evaluated for parameters like area, delay and power. This work can be further extended for higher number of bits. New architectures can be designed in order to reduce the power, area and delay of the circuits. Steps may be taken to optimize the other parameters like frequency, number of gate clocks, length etc.

REFERENCES

[1] Kuldeep Rawat, Tarek Darwish and Magdy Bayoumi, “A low power and reduced area Carry Select Adder”, 45th Midwest Symposium on Circuits and Systems, vol.1, pp. 467-470, March 2002.

[2] Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area, " Electron. Lett. vol. 37, no. 10, pp. 614- 615, May 2001.

[3] J. M. Rabaey, Digtal Integrated Circuits-A Design Perspective.Upper Saddle River, NJ: Prentice-Hall,2001.

[4] Cadence, "Encounter user guide, " Version 6.2.4, March 2008.

[5] R. Priya and J. Senthil Kumar, “ Enhanced area efficient architecture for 128 bit Modified CSLA”, International Conference on Circuits, Power and Computing Technologies,2013.

[6] Shivani Parmar and Kirat pal Singh,"Design of high speed hybrid carry select adder",IEEE ,2012.

[7] I-Chyn Wey, Cheng-Chen Ho, Yi-Sheng Lin, and Chien-Chang Peng," An Area-Efficient Carry Select Adder Design by Sharing the Common Boolean Logic Term", Proceedings of the International MultiConference of Engineers and Computer Scientist 2012 Vol II,IMCES 2012,HongKong,March 14-16 2012.

[8] B. Ramkumar and Harish M Kittur," Low-Power and Area-Efficient Carry Select Adder", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, VOL. 20, NO. 2, February 2012.

[8] Ms. S.Manjui, Mr. V. Sornagopae," An Efficient SQRT Architecture of Carry Select Adder Design by Common Boolean Logic",IEEE, 2013.

[9] Youngjoon Kim and Lee-Sup Kim, "64-bit carry-select adder with reduced area", Electronics Letters, vol.37, issue 10, pp.614-615, May 2001.

[10] Yajuan He, Chip-Hong Chang and Jiangmin Gu, "An area efficient 64-bit square root Carry-Select Adder for low power applications", IEEE International Symposium on Circuits and Systems,vol.4, pp.4082-4085, May 2005.

[11] Youngjoon Kim and Lee-Sup Kim, "A low power carry select adder with reduced area", IEEE International Symposium on Circuits and Systems, vol.4, pp.218-221, May 2001.

[12] Hiroyuki Morinaka, Hiroshi Makino, Yasunobu Nakase, Hiroaki Suzuki and Koichiro Mashiko, "A 64bit Carry Look-ahead CMOS Adder using Modified Carry Select",Proceeding of IEEE on Custom Integrated Circuits Conference, pp.585-588, May 1995.