An Efficient Method to Implement Low- Area and Power adder for Fast Computation

Mr. R. Krishnaprasanna, Ms. G. Pavithra, Ms. T. Rekha

Department of Electronics and Communication Engineering,
Christ College of Engineering and Technology,Puducherry,India

***Abstract*— Carry Select adder (CSLA) is one of the fastest adders used in many data processing processors to perform fastest arithmetic functions. From the structure of the CSLA, it is clear that there is scope for reducing the area and power consumption in the CSLA. This work uses a simple and efficient gate-level modification to significantly reduce the area and power of the CSLA. Based on this modification 4-bit, 8-bit and 16bit CSLA architecture have been developed and compared with the conventional CSLA and CSLA with BEC logic architecture. The proposed CSLA has been developed by sharing the common Boolean logic terms and also simulated using VHDL programming and the power dissipation is also calculated for CSLAs using Microwind. This proposed design has reduced area and power as compared with the conventional CSLA and CSLA with BEC logic. This work evaluates the performance of the proposed designs in terms of delay, area, power, and their products by hand with logical effort and through custom design and layout CMOS process technology.**

Keywords— conventional CSLA, BEC Logic, multiplexer, full adders, Power dissipation

I. INTRODUCTION

VLSI is the process of creating integrated circuits 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. This is the field which involves packing more and more logic devices into smaller and smaller areas. VLSI, circuits that would have taken board-full of space can now be put into a small space few millimeters across! This has opened up a big opportunity to do things that were not possible before.

In electronics, an adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic unit(s), but also in other parts of the processor, where they are used to calculate addresses, table indices, and similar operations. Although adders can be constructed for many numerical representations, such as binary-coded decimal or excess-3, the most common adders operate on binary numbers. In cases where two's or one's complement is being used to represent negative numbers, it is trivial to modify an adder into an adder–subtractor. Other signed number representations require a more complex adder. A typical adder circuit produces a sum bit (denoted by S) and a carry bit (denoted by C) as the output. Typically adders are realized for adding binary numbers but they can be also realized for adding other formats like BCD (binary coded decimal, XS-3 etc. Besides addition, adder circuits can be used for a lot of other applications in digital electronics like address decoding, table index calculation etc.

II. EXISTING SYSTEM

A. conventional CSLA

The carry-select adder generally consists of two ripple carry adders and two multiplexer Adding two n-bit numbers with a carry-select adder is done with two adders (therefore 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 number of bits in each carry select block can be uniform, or variable. In the uniform case, the optimal delay occurs for a block. Two 4-bit ripple carry adders are multiplexed together, where the resulting carry and sum bits are selected by the carry-in. Since one ripple carry adder assumes a carry-in of 0, and the other assumes a carry-in of 1, selecting which adder had the correct assumption via the actual carry-in yields the desired result.

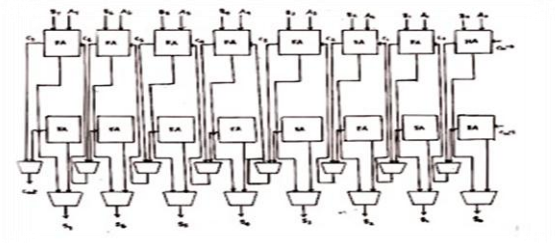

Fig 1: 8- bit SQRT conventional CSLA

The number of bits in each carry select block can be uniform, or variable. In the uniform case, the optimal delay occurs for a block. Two 4-bit ripple carry adders are multiplexed together, where the resulting carry and sum bits are selected by the carry-in. Since one ripple carry adder assumes a carry-in of 0, and the other assumes a carry-in of 1, selecting which adder had the correct assumption via the actual carry-in yields the desired result.

In conventional CSLA the area is calculated by counting the number of gates present in the circuit .It consist of

1 half adder (1*6) =6gates
15 full adders (15*13)=195 gates
16 multiplexers (16*4)=64 gates
Total = 265 gates

- **Area Evaluation Of Components In Conventional Csla**

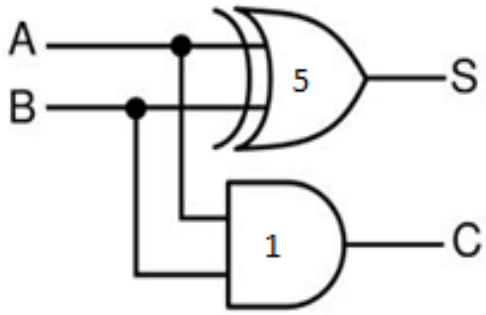

Fig2: half adder

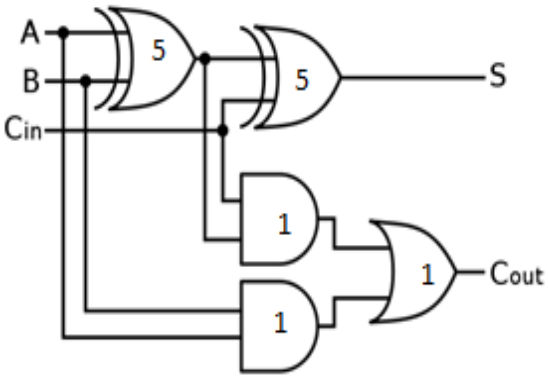

Fig3: full adder

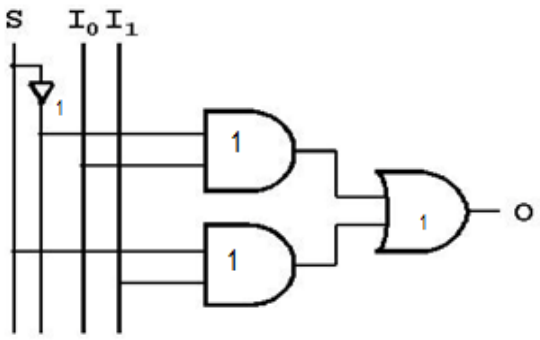

Fig 4: 2:1 multiplexer

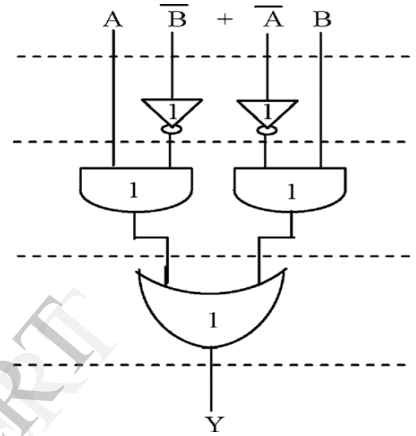

Fig 5: XOR gate

AREA AND DELAY CALCULATIONS

ADDER BLOCKS	DELAY	AREA
Half adder	3	6
Full adder	6	13
2:1 MUX	3	4
XOR	3	5

Table 1:area and delay calculations

The AND,OR, and Inverter (AOI) implementation of an XOR gate is shown in Fig. 1. 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 In-verter, 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

The area evaluation is done by counting the total number of 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 tabulated.

B. MODIFIED CSLA

In modified CSLA the number of gates used is reduced by replacing the full adders by BEC logic. By reducing the number of gates in the circuit the power dissipation is also reduced considerably.The 8-bit conventional CSLA is modified by using the BEC logic is drawn below

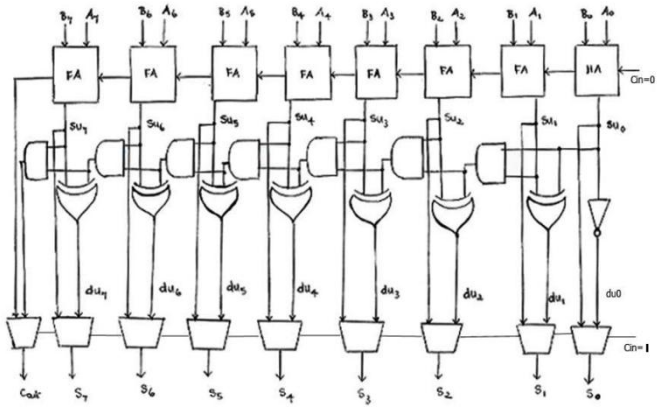

Fig 6: 8-bit modified CSLA

In conventional CSLA there are two ripple carry adders (with $c_{in} = 0$ and$c_{in} = 1$) and multiplexer are used to construct a CSLA structure as the ripple carry adders are nothing but cascading the N full adders .as the number of full adders in the conventional CSLA is more then the power dissipation is considerably high.as the power is directly proportional to the number of gates present in the circuit. Hence to reduce the area of CSLA the ripple carry adder with $c_{in} = 1$ is replaced by a BEC logic

- **BEC LOGIC**

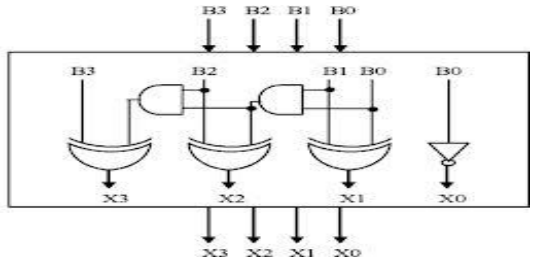

Fig 7: 4-bit BEC logic

The basic function of the CSLA is obtained by using the 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, & AND, ^XOR)

$$x0 = \sim B0$$
$$x1 = B1 \wedge B2$$
$$x2 = B2(B0 \& B1)$$
$$x3 = B3(B1 \& B2)$$

In modified CSLA the full adders area is high hence to reduce the number of gates the full adders are replaced by BEC logic circuit. Thus the 8bit modified CSLA contains

1 half adder (1*6) =6 gates
7 full adder (7*13) =91 gates
9multiplexer (9*4) =36 gates
7-EXOR (7*5) = 35 gates
7-AND (7*1) = 7 gates
1 -NOT (1*1) = 1 gate
Total = 176 gates

While comparing with conventional CSLA 89 gates are reduced in modified CSLA.

III. PROPOSED WORK

The main aim of our project is to reduce the power and area of the CSLA. The modified CSLA power also more hence in order to reduce the power, further we are constructing the CSLA using a EXOR gate, NOT gate, AND gate and a OR gate.

This method reduces the size of the circuit.when compare to the conventional and binary to excess 1 convertor logic this method is much more efficient. According to this concept,

- The sum is obtained by exclusive OR and NOT gate
- The carry is obtained by AND and OR gate

INPUT			OUTPUT	
Cin	A	B	SUM	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	0	0	1
1	1	1	1	1

Table 2: truth table for proposed methodology

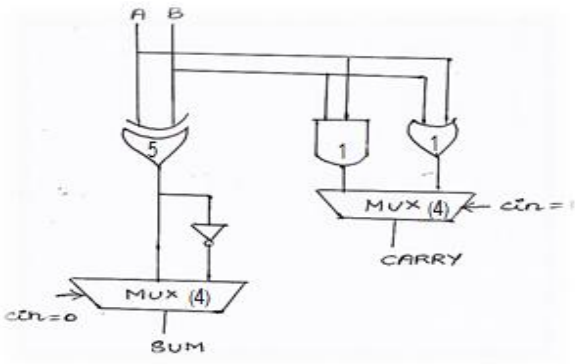

Fig 8: 1-bit proposed methodology

If we give $C_{in} = 0$ it selects the EXOR gates as sum, AND gate in carry. If the$C_{in} = 1$it selects the NOT gate as sum, OR gate in carry.

The sum is obtained by using an EXOR gate and a NOT gate. The carry is obtained by using an AND gate and OR gate. This is the basic structure of one bit CSLA. Hence by cascading the full adders we are able to obtain the 8, 16 and 32 bit CSLA circuits.

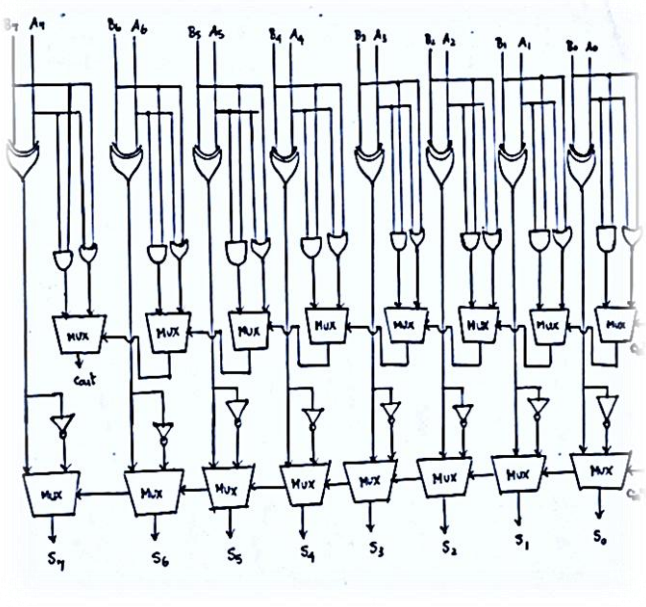

Fig 9: 8-bit proposed work

The further reduce the number of gates used in the CSLA circuit is reconstructed by using a proposed methodology.

8 EXOR gates (8*5)	**=**	**40gates16**
Multiplexer(16*4)	**=**	**64 gates**
8 NOT (8*1)	**=**	**8 gates**
8 AND(8*1)	**=**	**8gates**
8 OR (8*1)	**=**	**8 gates**
Total	**=**	**128 gates**

While comparing with the modified CSLA the gates are reduced by 48 gates in case of proposed methodology.

A. Comparison Of Number Of Gates In Existing And Proposed CSLA

CONVENTIONAL CSLA	CSLA WITH BEC	PROPOSED CSLA
Half adder=1 Full adder=15 Multiplexer=16	Half adder=1 Full adder=7 Multiplexer=9	Half adder=0 Full adder=7 Multiplexer=8
Total area=265	Total area=176	Total area=128
Power dissipation is maximum	Power dissipation is reduced by 34%	Power dissipation is reduced by 28%+34%=62%

IV. SIMULATION AND RESULTS

The Existing and Proposed Design were implemented in VHDL language and simulated using Xilinx ISE Design Suite 12.1.The power dissipation is calculated using Microwind. Initially the layout designs were made using lambda based design rules. The simuation results for both existing and proposed work are obtained.

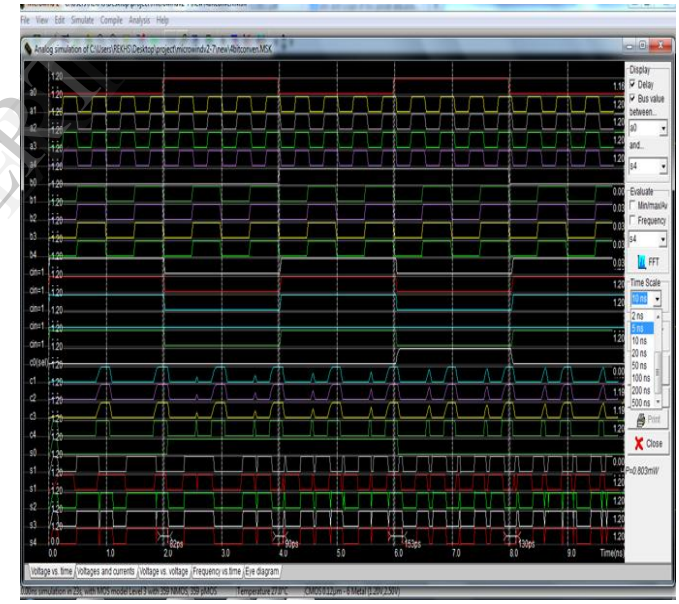

Fig 10: power dissipation of conventional csla

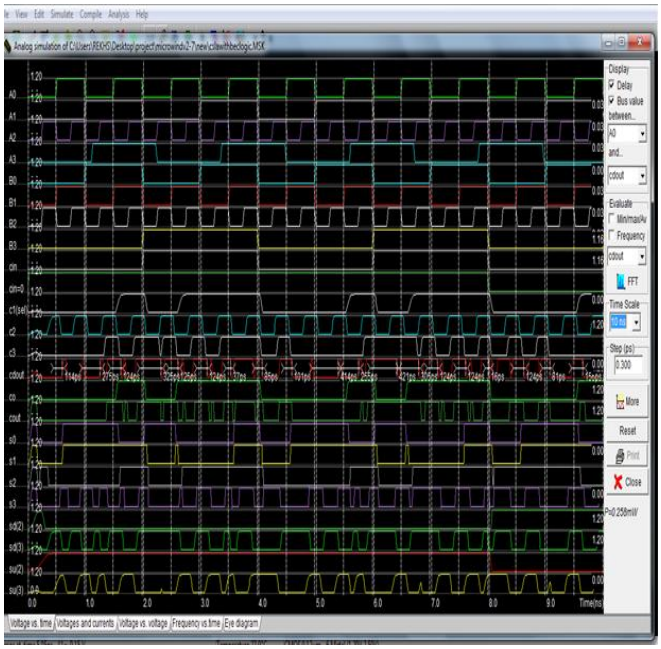

Fig 11:power dissipation of modified csla

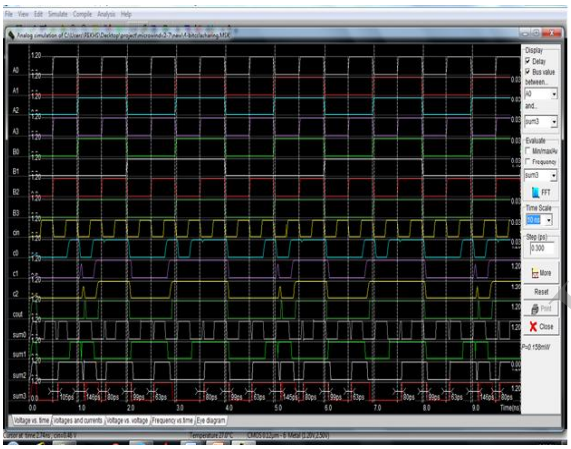

Fig12 : power dissipation of proposed csla

A. *OUTPUT COMPARISON OF POWER DISSIPATION*

CONVENTIONAL CSLA	MODIFIED CSLA	PROPOSED CSLA
Power dissipation =0.328 milli watt	Power dissipation=0.258 milli watt	Power dissipation=0.158 milli watt

Table 3: output comparison

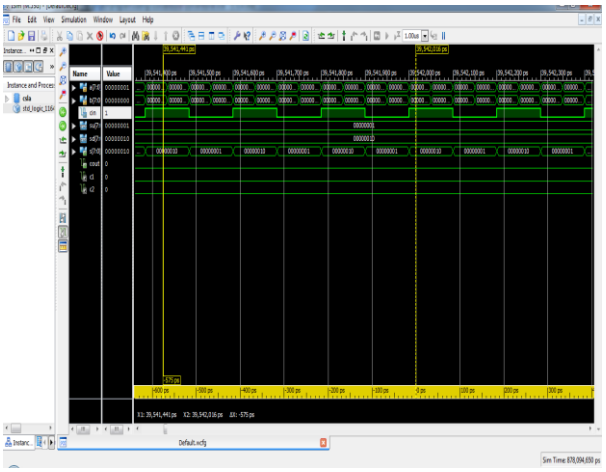

Fig.13:conventionalcsla

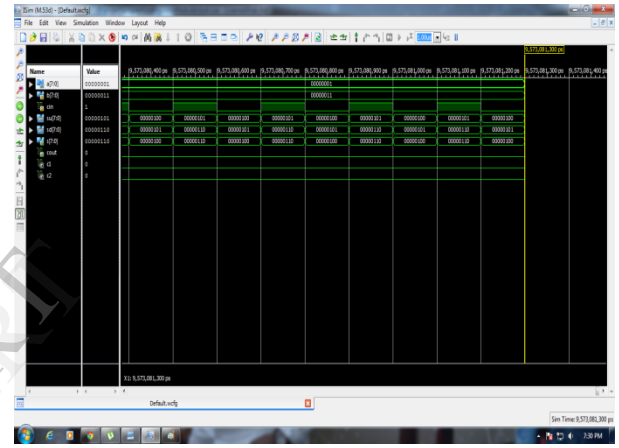

Fig.14:modified csla

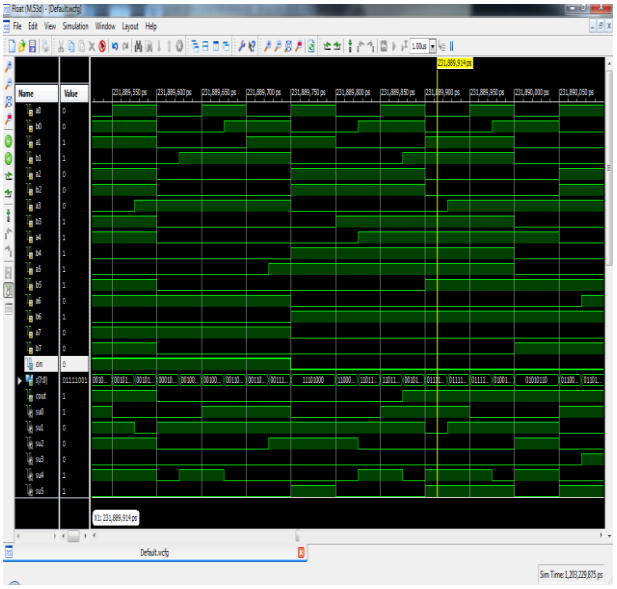

Fig.15:proposedcsla

V. SUMMARY AND CONCLUSION

Evaluating the delay and area count of the basic blocks of CSLA.The 8 bit conventional CSLA can be constructed by cascading the full adders in parallel with cin=0(half adder) and cin=1(full adder)., the problem existing in conventional CSLA is when the number of full adders are increased then the circuit complexity also increases.so the power consumption is also more.

In order to reduce the number of gates we replace ripple carry adder by BEC logic with cin=1 in the conventional CSLA.

In the proposed system, first we construct the 1 bit full adder circuit then we cascade that 1 bit full adder to 8 bit proposed methodology.For the further reduction of gates in modified CSLA, we replaced BEC logic by that full adder circuit, So that we finally achieved the less number of gates and power in proposed methodology.

The power consumption can be achieved by MICROWIND software.In this we have drawn a layout diagram for conventional, modified and proposed CSLA and finally we obtained the low power and reduced in area using XILIX software.

In our project the output of power consumption is obtained as 0.158mw in the proposed system. In the modified CSLA the number of gates in the components has been reduced from 265 gates to 176 gates by using BEC logic with the power consumption of 0.258mw.For the further reduction of gates can be done by using the full adder circuit and we reduce the gates from 176 to 128 gates with the power consumption of 0.158mw. We can reduce the total area by using the XILINX software.

REFERENCE

1. O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput., pp. 340–344, 1962.
2. B. Ramkumar, H.M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," Eur. J. Sci. Res., vol. 42, no. 1, pp. 53–58, 2010.
3. T. Y. Ceiang and M. J. Hsiao, "Carry-select adder using single ripple carry adder," Electron. Lett.,vol. 34, no. 22, pp. 2101–2103, Oct. 1998.
4. 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.
5. J. M. Rabaey, Digtal Integrated Circuits—A Design Perspective. Upper Saddle River, NJ: Prentice-Hall, 2001.
6. Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for lowpower applications," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.
7. Cadence, "Encounter user guide," Version 6.2.4, March 2008.

OTHER REFERENCE

1. *"BASIC VLSI DESIGN"- Douglas A. PucknellkamranEshraghian3rd edition -2012.*
2. *www.microwind.net*
3. *www.scribd.com*
4. *www.cmosvlsi.com*
5. *www.journals.elsevier.com*
6. *www.vlsi-world.com*