A Technical Review of Efficient and High Speed Adders for Vedic Multipliers

Vishnu Prasad Patidar[1], Sourabh Sharma[2]

[1]Research scholar, Dept. of Electronics Engineering, Trinity Institute of Tech. & Research, Bhopal, Madhya Pradesh, India
[2]Assistant Professor, Dept. of Electronics Engineering, Trinity Institute of Tech. & Research, Bhopal, Madhya Pradesh, India

Abstract –*In the VLSI system design, the main regions of research are the reduced size & increase speed path logic systems. A fundamental requirement of high speed, addition and multiplication is always needed for the high performance digital processors. In the digital system, the speed of addition depend on the propagation of carry, which is generated successively, after the previous bit has been summed & carry is propagated, into the next location. There are numerous types of adders available likes Ripple Carry Adder, Carry Look Ahead Adder, Carry Save Adder, Carry Avoid Adder, and Carry Select Adder, which have their own benefits and drawbacks. With the advances technology, proposal of Carry select adder (CSA) which deals either of the high speed, low power consumption, regularity of layout a smaller amount area and compact VLSI design implementation. Researchers justify that Ripple Carry Adder had a lesser area but having lesser in speed, in comparing with Carry Select Adders are fastest speed but possess a larger area. The Carry Look Ahead Adder is in between the spectrum having proper trade-offs between time and area complexities.*

Keyword – RCA, CLA, CSA.

I. INTRODUCTION

Adding is the basic fundamental operation of any arithmetic function [1], which is required in any digital systems, digital processing systems [2,5] or control system where multiplication, subtraction or division programming is done; the adder circuit plays a very significant role in the calculations. The most basic arithmetic operation is the addition of two binary digits 0 / 1 and they, are known as bits. A combinational, digital circuit which adds two bits i.e. 0 or 1 is known as half adder [4]. A full adder is one that adds three bits, that consist of two bit and one carry bit which is in fact formed from the addition of preceding two bits. Full adder circuit is easily implemented by combining two half adder circuits in any design.

This paper is organized into three sections. Firstly a brief study of dissimilar types of fast digital adders with their construction and working will be discussed in Section II. In Section III, we will, be discussing about the hardware necessities in terms of LUT memories and cells, speed of switching, benefits and disadvantages by comparing the results obtained by comprehensive simulations of Ripple Carry Adder (RCA),[6] Carry Look Ahead Adder (CLA) and Carry Select Adder (CSA), [2]. Conclusions drawn after comparing results, future work suggested will be discussed in the section IV.

II. ARCHITECTURE OF ADDERS

1. Full Adders

Refining performance of the digital adder would greatly advance the execution of binary operations inside a circuit find the middle ground of such blocks. the basic full adder architecture is depicted in fig.1a and input-output relation is shown in fig.1b. Dissimilar techniques have been developed for addition optimizing the performance in terms of speed, power and area [3]. In this sector we will review the architecture of RCA, CLA and CSA at gate level [4].

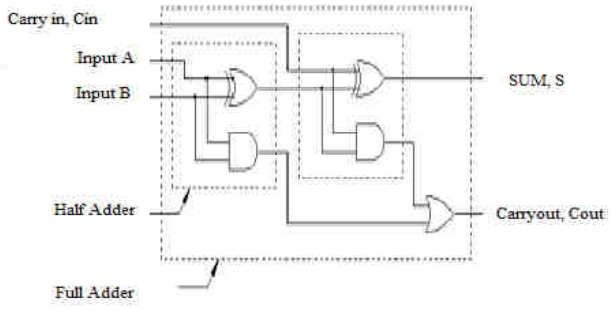

Fig.1(a): Gate level Architecture of 1 Bit Full Adder

INPUT			OUTPUT	
A	**B**	**C_{in}**	**S**	**C_{out}**
0	**0**	**0**	**0**	**0**
0	**0**	**1**	**1**	**0**
0	**1**	**0**	**1**	**0**
0	**1**	**1**	**0**	**1**
1	**0**	**0**	**1**	**1**
1	**0**	**1**	**0**	**1**
1	**1**	**0**	**0**	**1**
1	**1**	**1**	**1**	**1**

Fig.1(b): Truth Table for Full Adder Circuit

2. Ripple Carry Adder

The ripple carry adder is created by cascading full adders (FA) blocks in series. One full adder (FA) adds 2- binary digits at any stage of ripple carry adder and the carry output of one stage, is directly fed to the next stage of the adder circuit.

As number of full adders may be increased or decreased according to the requirement of the design. In case of the increasing the stages of Ripple Carry Adder (RCA), number of full adders (FA) dropped in series depending on the bit requirement and they are different sizes. For designing of n-bit Ripple Carry Adder (RCA) it necessitates N number of Full Adders (FA). Fig. 1c shows an example of a parallel adder: a 4-bit ripple-carry adder includes 4-full adder circuits. The augends bits of x are added to the addend bits of y respectfully of their binary position. Each bit addition generates a sum and a carry out. The carry out is then transmitted to the carry in of the next higher-instruction bit. The final result creates is a sum of four bits plus a carry out (c4).

Though it is a simple adder circuit and can be used to add unrestricted bit length numbers, but not very

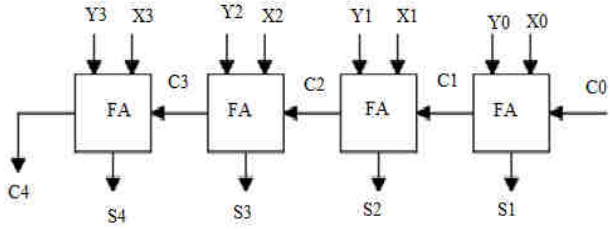

Fig.1(c) Block Diagram of 4-bit Ripple Carry Adder

Capable, when large numbers of bits are used. One of the most serious difficulty of this adder circuit is that the increase in delays which be influenced by on the bit length. In the working of Ripple Carry Adder (RCA) every stage of Full Adder be governed by on the carry of the previous stage. Taking again the example in figure 4, in the 4^{th} stage of the 16-bit Ripple Carry Adder (RCA), calculation of x4 and y4 will not influence their final condition until the carry out c4 is produced. For this final carry out c4 to come depend on carry c3, which carry out of the previous stage i.e. from the 3^{rd} stage of the Ripple Carry Adder (RCA). Similarly, it goes on from higher to lower stage i.e. carry of existing stage to come depend on the carry of previous stage such as c4←c3←c2←c1. Therefore, suppose if one full adder takes T seconds to full its adding process, so for 4 full adders it requires 4T seconds to accomplish this operation. there will be very minor change in the area or size of the circuit. If it is priory known that initially carry bit is zero, at that time we can simply replace full adder (FA) with a half adder circuit and that will know as first stage of the circuit. Let the delay count is given by T_{gate} counted each single gate area, by A gate then by estimating the basic circuit, we found that delay becomes 3n T_{gate} and area 5n T_{gate}, n is the number of bit size [5,6].The poorest-case delay of the RCA is when a carry signal conversion ripples through all stages of adder sequence from the least significant bit to the most significant bit,

This is given by:

$$t = (n - 1)\, t_c + t_s \qquad \text{.....(i)}$$

Where,

t_c is the delay through the carry period of a full adder, and t_s is the delay to compute the sum of the last stage. The delay of ripple carry adder is linearly equivalent to n, the number of bits; therefore the performance of the RCA is inadequate when n grows bigger.

3. Carry Look Ahead Adder

As understood in the ripple-carry adder, its limiting factor is the time it takes to propagate the carry. The carry look-ahead adder [6] solves this problem by calculating the carry signals in advance, based on the input (I/P) signals. The result is a reduced carry propagation delay time. To be able to understand how the carry look-ahead adder works, we have to influence the Boolean expression allocating with the full adder. The Propagate P, and generate G, in a full-adder, is given as:-

$Pi = Ai \oplus Bi$. Carry Propagate.
$Gi = Ai.Bi$ Carry Generate.

Note that both propagate and generate signals be determined by only on the input bits and thus will be valid after one gate delay.

The new expressions, for the output sum, and the carryout, are given by:

$$Si = Pi \oplus Ci_{-1,}$$
$$Ci_{+1} = Gi + PiGi,$$

These equations show, that a carry signal will be produced in two cases:

1) If both bits Ai and Bi are 1,
2) If either Ai or Bi is 1, and the carry-in Ci is 1.

Let's, apply these equalities for a 4-bit adder:

1. $C1 = G0 + P0C0$,
2. $C2 = G1 + P1C1 = G1 + P1\,(G0 + P0C0) = G1 + P1G0 + P1P0C0$,
3. $C3 = G2 + P2C2 = G_2 + P_2\,(G_1 + P_1G_0 + P_1P_0C_0) = G2 + P2G1 + P2P1G0 + P2P1P0C0$,
4. $C4 = G3 + P3C3 = G_3 + P_3\,(G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0) = G3 + P3G2 + P3P2G1 + P3P2P1G0 + P3P2P1P0C0$,

These expressions show that C2, C3 and C4 do not depend on its former carry-in. Therefore C4, does not necessity to wait for C3, to propagate. As soon as C0 is computed, C4 can, reach steady state. The equal is also true, for C2, and C3.

The general expression is

C_{i+1}= Gi + PiGi-1 + PiPi-1Gi-2 + ……. PiPi-1….P2P1G0 + PiPi-1 ….P1P0C0,.

This is a double level Circuit. In CMOS however, the delay of the function is not-linearly dependent, on its fan in. Therefore big fan in gates are not practical.

Carry look-ahead adder's structure can be divided into three parts: the broadcast generator shown in Fig.2a, the sum generator Fig.2b and the carry generator Fig.2c

.

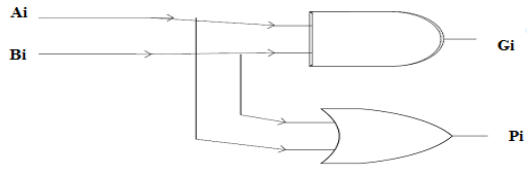

Fig. 2(a) Propagate / Generate Generator

Fig. 2(b) Sum Generator

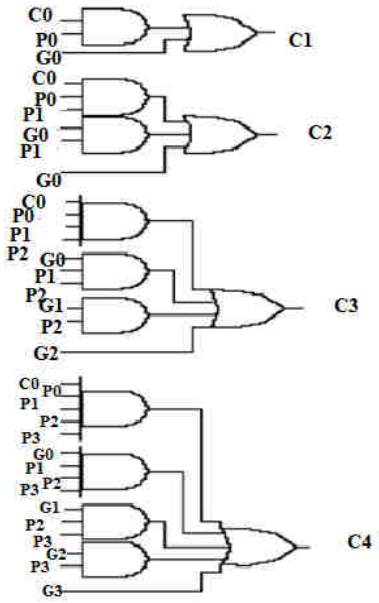

Fig. 2(c): Look Ahead Carry Generator

The size and fan-in of the gates needed to appliance the Carry-Look-ahead adder is usually limited to four, so 4-bit Carry-Look ahead adder is designed as a block.

4. Carry Select Adder

Carry-select adders can be separated into equal or unequal sections. Fig.3a shows the implementation of an 8 bits carry-select adder with 4-bit subdivisions. For each section, shown in Fig.3a, the calculation

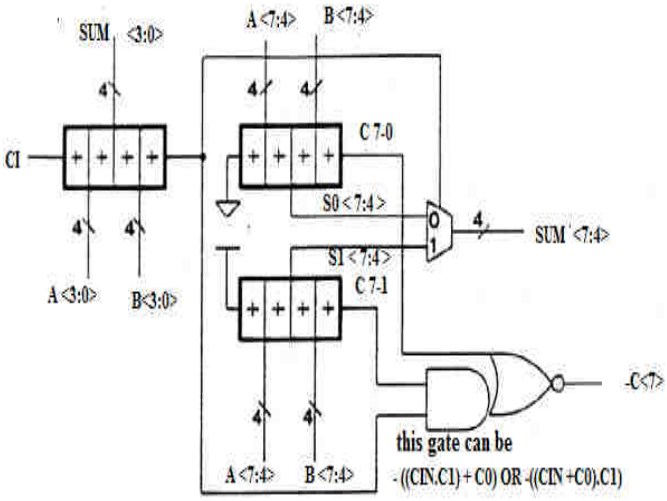

Fig. 3(a): 4-Bit Carry Select Adder

of two sums is proficient using two 4-bit ripple-carry adders. One of these adders is served with a 0 as carry-in however the other is fed a 1. Then using a multiplexer, depending on the real carryout of the foregoing section, the correct sum is chosen. Similarly, the carryout of the section is computed twice and chosen depending of the carry-out of the previous section. The concept can be extended to any length for example a 16-bit carry-select adder can be composed of four subdivisions each section is shown in Fig. 3b. Each of these sections is self-possessed of two 4-bits ripple-carry adders. This is referred as undeviating expansion. The delay of n-bit carry select adder based on an m-bit CLA blocks can be given by the succeeding equality when using constant carry number blocks

$$T = t_{setup} + mt_{carry} (n/m) t_{tmux} + t_{sum}$$

And by the following equality when using sequentially incremented carry number blocks correspondingly.

$$T = t_{setup} + mt_{carry} + (2n)^{1/2} t_{tmux} + t_{sum}$$

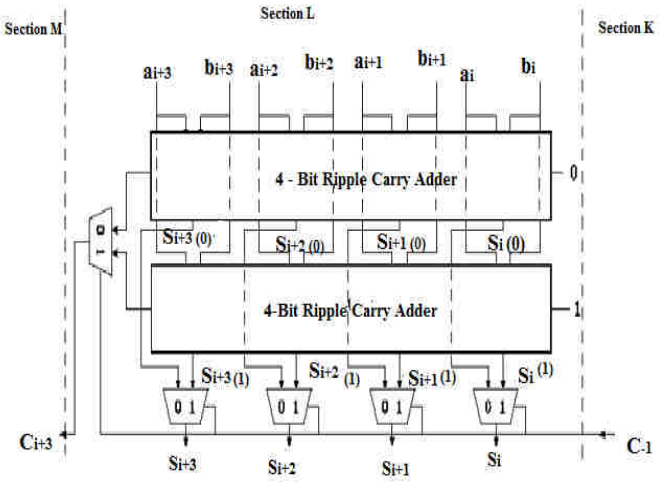

Fig. 3(b) One Section of Larger Carry Select Adder

III. SIMULATIONS AND RESULTS

The implementation of the different is done in the Verilog module using Xilinx 10.2i software platform and verify the simulation results.

Simulation Results:

Simulation results of 16 bit Ripple Carry Adder is shown in Fig. 4a, 16 bit Carry Look Ahead Adder is shown in Fig.4b and 16 bit Carry select adder in Fig. 4c.

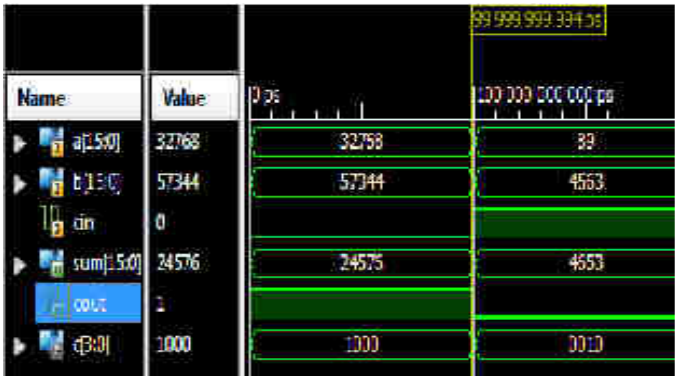

Fig. 4(a): 4-Bit Ripple Carry Adder

Fig. 4(b): Carry Look Ahead Adder

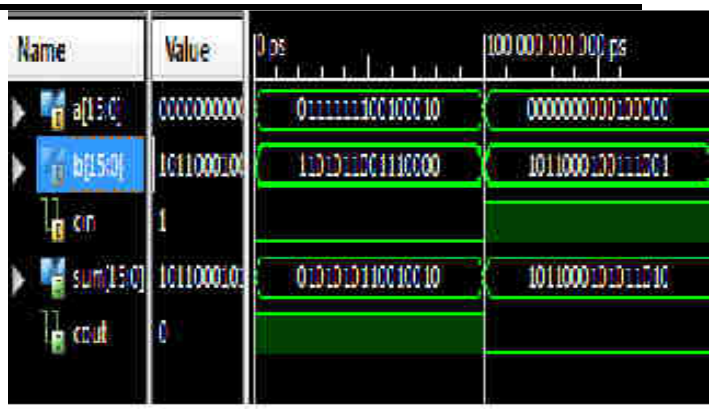

Fig. 4(c): Carry Select Adder

A comparison of three adder table is presented in Table 1 which demonstrates the performance parameters of adders by calculating of area of device in terms of number of slices, LUT computation, and input/output pads, and speed of the adders using gate delay count.

Table 1. Comparison of RCA, CLA, CSA

Logic Utilization	**RCA**	**CLA**	**CSA**
No. of Slices	96	72	108
Number of 4 input LUTs	128	128	192
Number of Bonded IOBs	200	200	200
Delay	96.686ns	96.686ns	88.092ns

IV. CONCLUSION

In this review paper we have simulated parallel adders using Xilinx 10.2i version and targeted device. In terms of area Carry Look Ahead Adder is a better choice than Carry Select and Ripple Carry Adders. But in terms of gate delay count Carry Select Adder offers higher speed than the other two. For optimization between area and speed CSA has an edge of 5% over CLA and RCA. In future one can obtained results for 32 bits or higher number of bits so that the convergence of results can accurately predicted. The CSA can be replaced in Vedic Multipliers and efficiency in terms of area and speed can be measured.

REFERENCES

[1] B.Ramkumar et al. " Low Power and Area Efficient Carry Select Adder" IEEE transactions on Very Large Integration System (VLSI) VOL. 20, No. 2, February 2012.

[2] T Y Chang and M J Hsiao, "Carry Select Adder using single Ripple Carry Adder" Electronics, Letters 29th October 1998,Vol. 34

[3] Shipra Mishra et.al, "Design Low Power 10 Full Adder Using Process and Circuit Techniques", Proceedings of 7th International Conference on Intelligent Systems and Control, IEEE 2012.

[4] Seji,Kahihara and Tsutomu Sasao, "On the adders with minimum tests", IEEE Proceedings of the 5th Asian Test Symposium 1997.

[5] Chetana Nagendra, Mary Jane Irwin and Robert Michael Owens, "Area-Time-Power Trade-offs in Parallel Adders", IEEE Transactions on circuits and systems-II: Analog and Digital signal Processing, Vol. 43, No.10, oct. 1996, pp. 689 – 702.

[6] May Phyo Thwal, Khin Htay Kyi, and Kyaw Swar Soe; "Implementation of Adder-Subtracter Design with Verilog HDL" World Academy, of Science, Engineering and Technology Vol:2 , March 2008.

[7] Y. He, C. H. Chang and J. Gua, "An area efficient 64-bit square rootcarry-select adder for lowpower applications," in *Proc. IEEE Int. Symp.Circuits Syst.*, 2005, vol. 4, pp. 4082–4085.

[8] Digital Design with VHDL, Charles Roth Tata Mcgraw Hill 2nd edition.