ISSN Online: 2319 – 9253
Print: 2319 – 9245

NOVEL HIGH SPEED IMPLEMENTATION OF 32 BIT MULTIPLIER USING CSLA and CLAA

[#1]NANGUNOORI THRIVENI – Pursuing M.Tech,
[#2]P.NARASIMHULU - Associate Professor,
SREE CHAITANYA COLLEGE OF ENGINEERING, KARIMNAGAR, T.S., INDIA.

ABSTRACT: This paper presents a Very high speed integrated circuit – Hardware Description Language (VHDL) based design and implementation of a fast unsigned multiplier. The multiplier uses a carry look-ahead adder, which reduces the delay time caused by the effect of carry propagation through all the stages of a ripple-carry adder. In this paper, design of two different array multipliers are presented, one by using carry-look-ahead (CLA) logic for addition of partial product terms and another by introducing Carry Select Adder (CSLA) in partial product lines. The multipliers presented in this paper were all modeled using VHDL (Very High Speed Integration Hardware Description Language) for 32-bit unsigned data. The comparison is done on the basis of three performance parameters i.e. Area, Speed and Power consumption. To design an efficient integrated circuit in terms of area, power and speed, has become a challenging task in modern VLSI design field. Previously in the literature, performance analysis was carried out between multiplier using Ripple carry adder (RCA) and by using CLA. In this work, same multiplier is designed by using CSLA logic and compare it's performance with the multiplier designed by using CLA logic. Multiplier with CSLA gives better result in terms of speed (78.3% improvement), area (reduced by 4.2%) and power consumption (decreased by 1.4%).However, the carry look-ahead adder requires extra logic circuit to generate the carry, which can reasonably justified by the relatively cheap cost of the contemporary hardware. The VHDL– based model of the fast multiplier was developed using the Direct VHDL simulator software.

Keywords: CLAA, CSLA, Delay, Area, Array multiplier, VHDL modeling, Simulation

I.INTRODUCTION

Digital computer arithmetic is an aspect of logic design with the objective of developing appropriate algorithms in order to achieve an efficient utilization of the available hardware. The basic operations are addition, subtraction, multiplication and division. In this, we are going to deal with the operation of additions implemented to the operation of multiplication. The repeated form of the addition operations and shifting results in the multiplication operations. Given that the hardware can only perform a relatively simple and primitive set of Boolean operations, arithmetic operations are based on a hierarchy of operations that are built upon the simple ones. In VLSI designs, speed, power and chip area are the most often used measures for determining the performance and efficiency of the VLSI architecture.

Multiplications and additions are most widely and more often used arithmetic computations performed in all digital signal processing applications. Addition is afundamental operation foranydigital multiplication. A fast, area efficient and accurate operation of a digital system is greatly influenced by the performance of the resident adders.

Adders are also very important component in digital systems because of their extensive use in these systems. In this project we are going to compare the performance of different adders implemented to the multipliers based on area and time needed for calculation. On comparison with the Carry Look-Ahead Adder (CLAA) based multiplier the area of calculation of the carry select adder (CSLA) based multiplier is smaller and better with nearly same delay time. Here we are dealing with the comparison in the bit range of n*n (32*32) as input and 2 n (64) bit output.

Hence, to design a better architecture the basic adder blocks must have reduced delay time consumption and area efficient architectures. The demand is of DSP style systems for both less delay time and less area requirement for designing the systems. Our interest is in the basic building blocks of arithmetic circuits that dominate in DSP applications, VLSI architectures, computer applications and where ever reduced area computation is needed.

Reduced area and high speed data path logic systems are the major areas of research in VLSI system design. High-speed addition and multiplication has always been a fundamental necessity of high-performance processors and systems. In digital adders, the speed of addition is partial by the time necessary to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated serially only after the previous bit position has been summed and a carry propagated into the next position. There are several types of adder designs available (RCA, CLAA, CSA, CSA) which have its own advantages and disadvantages. The main speed limitation in any adder is in the production of carries and many authors considered the addition problem. To solve the carry propagation delay CSLA is developed which drastically decreases the area and delay to a great extent. The CSLA is used in many computational systems design to moderate the problem of carry propagation delay by separately generating multiple carries and then select a carry to generate the sum. It uses independent ripple carry adders (for Cin=0 and Cin=1) to generate the resultant sum. However, the Regular CSLA is not area and speed efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input. The final sum and carry are selected by the multiplexers (mux). Due to the use of two independent RCA the area will enlarge which leads an increase in delay. To overcome the above problem, the basic idea of the planned work is to use n-bit binary to excess-1 code converters (BEC) to improve the speed of addition. This logic can be replaced in RCA for Cin=1 to further improves the speed and thus decreases the delay. Using Binary to Excess-1 Converter (BEC) instead of RCA in the regular CSLA will achieve lower area, delay which speeds up the addition operation. The major advantage of this BEC logic comes from the lesser number of logic gates than the Full Adder (FA) structure because the number of gates used will be decreased.

II.RELATED WORK

A digital designer has to concentrate on many criteria like Circuit speed, power consumption, area and cost.While designing the digital circuits, the fundamental arithmetic operations like addition and multiplication are the main optimizations. Our design concentrates on multiplication of binary numbers for larger applications. For the multiplication operation, adding as well as shifting of bits is necessary. Considering these two operations, we have designed a 64 bit multiplier by using carry save adder for addition and multi bit flip flop based shift register for shifting of bits. In this paper we have shown the comparison among the adders for multiplication based on the time. On comparison with carry look ahead adder (CLAA) and carry select adder (CSLA), carry save adder (CSA) based multiplier is less complex and results have shown that CSA based multiplier is very faster than the other two multipliers. A shift register isvery important digital building block. It has a large amount of applications. Registers are often used to momentarily store binary information appearing at the output. Shift registers are the logic types which are used basically for the storage and transfer of digital data. The basic storage elements are the flip flops. The most of the registers use D flip flops due to its simplicity.

Basically a flip flop stores a single bit data. In our proposed design the flip flop stores a multi bit data. By using the multi bit flip flops, we have shown a shift registerwhich is serial in serial out (SISO). Finally we have designed a 64 bit multiplier by using carry save adder and multi bit flip flop shift register.

2.1 Carry look ahead adder

In the adders the performance of the adder is mainly based on the carry propagation. The ripple carry adder calculates the carry bits along with the sum so the performance of the ripple carry adder is slow but it takes the low power. To overcome this problem we are designing the carry look ahead adder. The carry look ahead adder first calculates the all carry bits after its calculates the sum bits.The carry propagation (P) and Carry generation (G) is given as

$$P_i = A_i \oplus B_i \qquad \text{Carry propagate}$$
$$G_i = A_iB_i \qquad \text{Carry generate}$$

The outputs sum and carry is given as

$$S_i = P_i \oplus C_{i-1}$$
$$C_{i+1} = G_i + P_iC_i$$

The final carry output is given as

$$C_{i+1} = Gi + PiG_{i-1} + PiP_{i-1}G_{i-2} + \ldots\ldots. P_iP_{i-1}\ldots.P_2P_1G_0 + PiP_{i-1}\ldots.P_1P_0C_0.$$

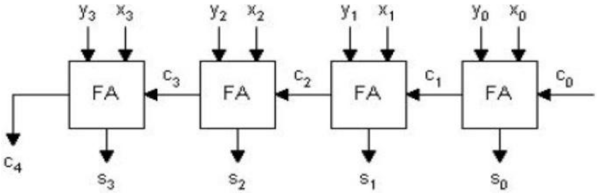


Fig 1: Carry looks ahead adder

2.2 Carry select adder

Generally the CSLA have the two ripple carry adder stages and multiplexer. Carry select adder selects the correct result using multiplexer with single stage or multiple stage. For two stages of ripple carry adders we have the two outputs (2 sums, 2 carrys). The correct result will be selected by the multiplexer and speed will be high when comparing with the different adders.

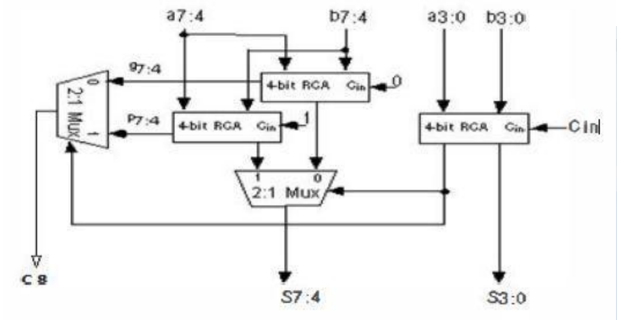


Fig 2 . Carry select adder

III.MULTIPLICATION ALGORITHM

Let the product register size be 64 bits. Let the multiplicand registers size be 32 bits. Store the multiplier in the least significant half of the product register. Clear the most significant half of the product register.

Repeat the following steps for 32 times:

- If the least significant bit of the product register is “1” then add the multiplicand to the most significant half of the product register.
- Shift the content of the product register one bit to the right (ignore the shifted-out bit).
- Shift-in the carry bit into the most significant bit of the product register. Figure 3 shows a block diagram for such a multiplier.

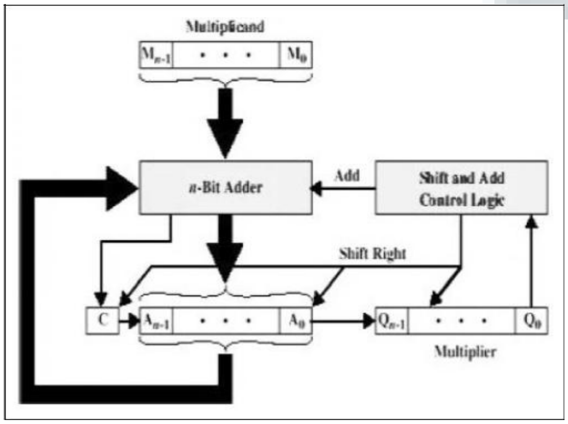


Figure 3: Multiplier of Two n-bit Values

V. VHDL SIMULATIONS

The VHDL simulation of the two multipliers is presented in this section. In this, waveforms, timing diagrams and the design summary for both the CLAA and CSLA based multipliers are shown in the figures. The VHDL code for both multipliers, using CLAA and CSLA, are generated. The VHDL model has been developed using Altera Quartus II and timing diagrams are viewed through avan waves. The multipliers use two 32-bit values.

Figure 4: Waveform for a CLAA Based Multiplier

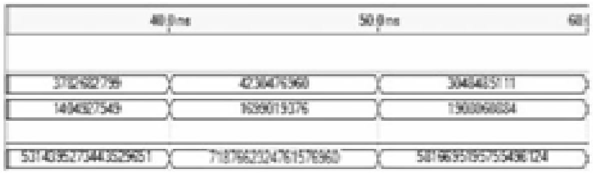


Figure 5: Waveform for CSLA Based Multiplier

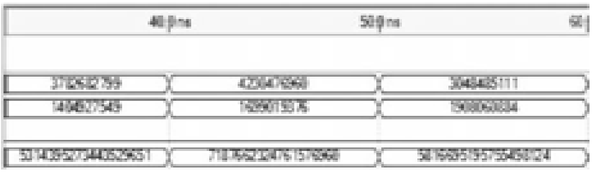
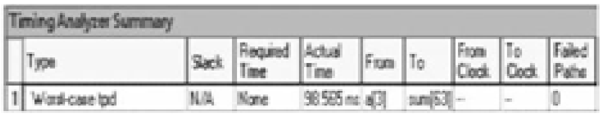


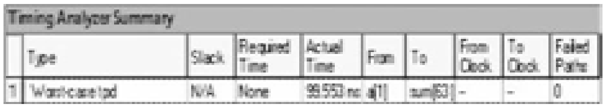
Figure 6: Timing Analysis for CLAA Based Multiplier



Timing Analyzer Summary

	Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths
1	Worst-case tpd	N/A	None	98.585 ns	a[3]	sum[63]	--	--	0

Figure 7: Timing Analysis for CSAA Based Multiplier



Timing Analyzer Summary

	Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths
1	Worst-case tpd	N/A	None	99.553 ns	a[1]	sum[63]	--	--	0

VI.PERFORMANCE ANALYSTS

Area Analysis

The performance analysis for the area of CLAA and CSLA based multipliers are represented in the form of the diagram shown in Figure 8.

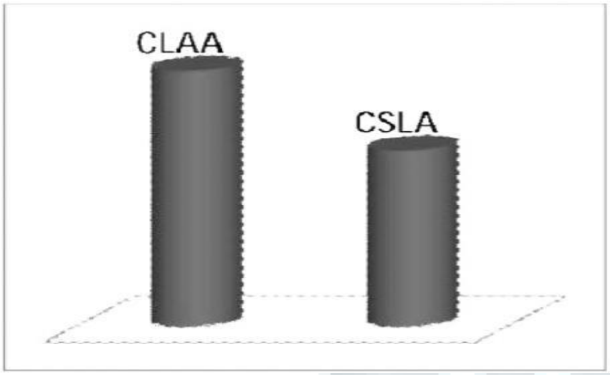


Figure 8: Figure Area Analysis Chart

Delay Analysis

The performance analysis for the delay time of CLAA and CSLA based multipliers are represented in the form of the diagram shown in Figure 9.

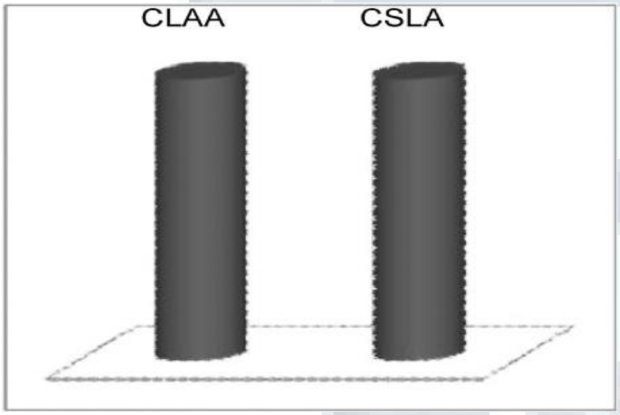


Figure 9: Figure Delay Analysis Chart

Area Delay Product Analysis

The performance analysis for the area delay product of CLAA and CSLA based multipliers are represented in the form of the diagram shown in Figure 10.

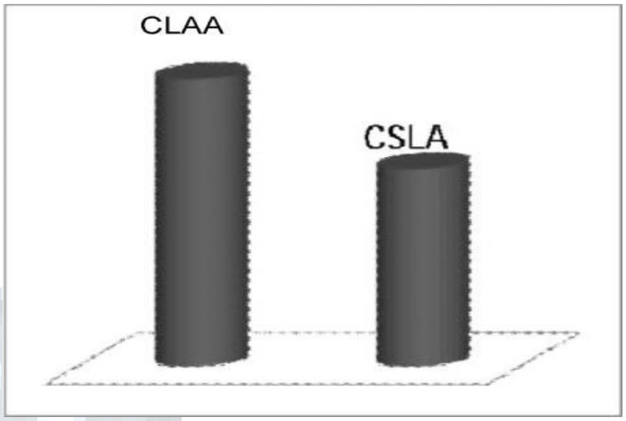


Figure 10: Figure Area Delay Product Analysis Chart

The area needed and delay for both the CLAA and CSLA implemented to the multiplier was analyzed and the comparison was shown in the figure in the form of a table.

Analysis Table

Tn this analysis table shown in Table 1, the delay time is nearly same, the area and the area delay product of CSLA based multiplier is reduced to 31% when compared to CLAA based multiplier.

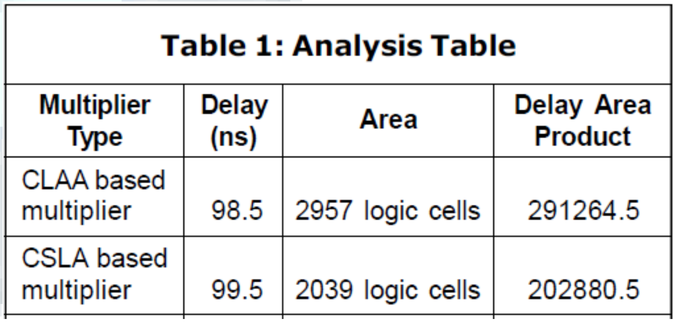


Table 1: Analysis Table

Multiplier Type	Delay (ns)	Area	Delay Area Product
CLAA based multiplier	98.5	2957 logic cells	291264.5
CSLA based multiplier	99.5	2039 logic cells	202880.5

VII.CONCLUSION

A design and implementation of a VHDL based 32-bit unsigned multiplier with CLAA and CSLA was presented. VHDL, a Very High Speed Integrated Circuit Hardware Description Language, was used to model and simulate our multiplier. Using CSLA improves the overall performance of the multiplier. Thus a 31% area delay product reduction is possible with the use of the CSLA based 32-bit unsigned parallel multiplier than CLAA based 32-bit unsigned parallel multiplier.

REFERENCES

[1] J. Bedrij, “Carry-select adder,” IRE Trans. Electron, pp. 340–344, 1962.

[2] B. Rajkumar, .M. Kittur, and P. Kanna, “ASIC implementation of modified faster carry select adder,” Eur. J. Sci.Res., vol. 42, no. 1, pp.53–58, 2010.

[3] Ceiang.T.Y and M. J. Hsiao, “Carry-select adder using ripple Carry adder,” Electron. Lett, vol. 34, no. 22, pp. 2101–2103, Oct. 1998.

[4] Y. Kim and L.-S. Kim, “carry-select adder with reduced area,” Electron. Lett. vol. 37, no. 10, pp. 614–615, May 2001.

[5] J. M. Rabaeay, Integrated Circuits—A Design Perspective.Upper Saddle River, NJ: Prentice-Hall, 2001.

[6] Y. He, C. H. Chang, and J. Gu, “low power & area efficient 64-bit square Root carry-select adder for low power applications,” in Proc. IEEE Int. Symp.Circuits Syst., vol. 4, pp. 4082–4085, 2005.

[7]Rado Zlatanovici, Borivoje Nikolic, “Energy-Delay of Optimization 64-Bit Carry- Lookahead Adders,” *IEEE J Solid State circuits*,vol.44, no. 2, pp. 569-583, Feb. 2009.

[8] Navi.K , Kavehei.O, Rouholamini.M , Sahafi.A, “Low-Power and High-Performance 1-bit CMOS Full.Adder Cell,” *Journal of Computers*, Academy Press, vol. 3, no. 2,Feb. 2008.