

# An Optimized Implementation of CSLA and CLLA for 32-bit Unsigned Multiplier Using Verilog

<sup>1</sup>P.Sanjeeva Krishna Reddy, PG Scholar in VLSI Design,

<sup>2</sup>A.M.Guna Sekhar Assoc.Professor

<sup>1</sup>appireddigarichaitanya@gmail.com,

<sup>2</sup>guna.421@gmail.com

## Abstract:

In this paper deals with the comparison of the VLSI design of the carry look-ahead adder (CLAA) based 32-bit signed and unsigned integer multiplier and the VLSI design of the carry select adder (CSLA) based 32-bit signed and unsigned integer multiplier. Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important part in low-power VLSI system design. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. Carry select adder is one of the fastest adders used in many applications to perform fast arithmetic functions. This work evaluates the performance of the proposed designs in terms of delay, speed(frequency)and memory. The CLAA based multiplier uses the delay time of 99ns for performing multiplication operation where as in CSLA based multiplier also uses nearly the same delay time for multiplication operation. But the area needed for CLAA multiplier is reduced to 31 % by the CSLA based multiplier to complete the multiplication operation.

Keywords-CLAA; CSLA; Delay; Area; Array Multiplier:VHDL Modeling & Simulation.

## I.Introduction

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. 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. 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. A new concept of hybrid adders is presented to speed up addition process[10]. 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  $C_{in} = 0$  and  $C_{in} = 1$ , then the final sum and carry are selected by the multiplexers (mux).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  $2n$ (64) bit output.

Multiplication is a mathematical operation that at its simplest is an abbreviated process of adding an integer a specified number of times. Multiplication is the fundamental arithmetic operation important in several processors and digital signal processing systems. Multiplication of two  $k$  bit number needed multi operand addition process that can be realized in  $k$  cycles of shifting and addition with hardware, firmware or

software. Multiplication based operations such as multiply and accumulate (MAC) and inner product are among some of the frequently used intensive arithmetic functions currently implemented in many digital signal processing (DSP) applications such as convolution, fast fourier transform(FFT), filtering and in microprocessors in its arithmetic and logic unit. Portable multimedia and digital signal processing (DSP) systems, which typically require low power consumption, short design cycle, and flexible processing ability, have become increasingly popular over the past few years. As many multimedia and DSP applications are highly multiplication intensive so that the performance and power consumption of these systems are dominated by multipliers. Unfortunately, portable devices mostly operate with stand-alone batteries, but multipliers consumes large amount of power. Digital signal processing systems need multiplication algorithms to implement DSP algorithms such as filtering where the multiplication algorithm is directly within the critical path. Along with signal processing applications, multimedia, and 3D graphics, performance, in most cases, strongly depends on the effectiveness of the hardware used for computing multiplications, since multiplication is, besides addition, massively used in these environments. Consequently, it's greatly imperative to develop power-efficient multipliers to compose a high-performance and low-power portable multi-media and DSP system. 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 speed and area remain to be the two major design tools. The higher speed results to enlarged power consumption, thus, low power2 architectures will be the choice of the future. 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. This has given way to the growth of new circuit algorithms, with the plan of reducing the power consumption of multiplication algorithms with having high-speed structures and appropriate performance. The multiplier is fairly large block of a computing system. The size of multiplier is directly proportional to the square of its resolution i.e. size of multiplier.

### 1.1 Classification of multiplier

There are two kinds of multiplier as shown in fig. 1.1

- a) Serial multiplication algorithms
- b) Parallel multiplication algorithms

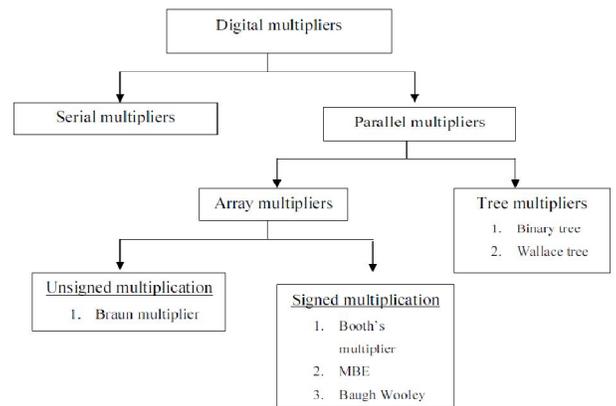


Fig. 1.1: Classification of multipliers

### 1.2 Multiplication of signed and unsigned number

Multiplication of unsigned numbers can be done by simple multiplication algorithm. If the sign of the multiplier and multiplicand are different, sign of the product is negative. The magnitude is multiplied in the same way as the unsigned numbers. If both are negative, the sign of the result is positive. The sign requirements are met by the general rule: the sign of the product is exclusive or of the sign of the number.

ADDDERS:

Addition is the most common and often used arithmetic operation on microprocessor, digital signal processor, especially digital computers. Also, it serves as a building block for synthesis all other arithmetic operations. Therefore, regarding the efficient implementation of an arithmetic unit,

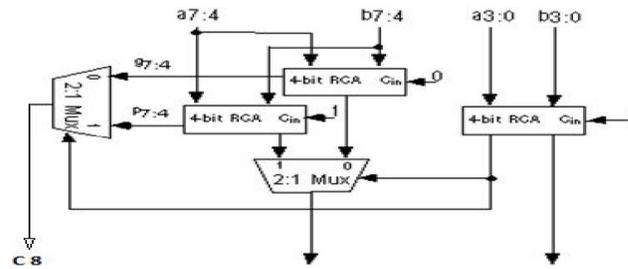
the binary adder structures become a very critical hardware unit.

The first class consists of the very slow ripple-carry adder with the smallest area. In the second class, the carry-skip, carry-select adders with multiple levels have small area requirements and shortened computation times. From the third class, the carry-look ahead adder and from the fourth class, the parallel prefix adder represents the fastest addition schemes with the largest area complexities.

Time delay: time to compute first section + time to select sum from subsequent sections.

### 1.2 Ripple Carry Adders (RCA)

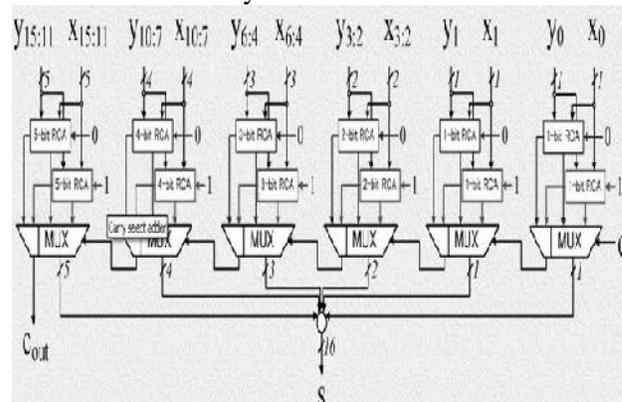
The well-known adder architecture, ripple carry adder is composed of cascaded full adders for n-bit adder, as shown in figure 4.1. It is constructed by cascading full adder blocks in series. The carry out of one stage is fed directly to the carry-in of the next stage. For an n-bit parallel adder it requires n full adders.



Carry Select Adder

### 1.3 Carry Select Adders (CSLA)

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  $k/2$  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. It is composed of two four-bit ripple carry adders per section. Both sum and carry bits are calculated for the two alternatives of the input carry, '0' and '1'. The carry out of each section determines the carry in of the next section, which then selects the appropriate ripple carry adder. The very first section has a carry in of zero.



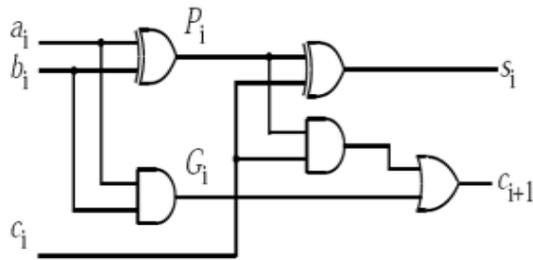
Block Diagram of Regular 16b Carry Select Adder

### 1.4 CARRY LOOK AHEAD ADDER

The carry look ahead adder (CLA) solves the carry delay problem by calculating the carry signals in advance, based on the input signals. It is based on the fact that a carry signal will be generated in two cases:

- 1) when both bits  $a_i$  and  $b_i$  are 1,
- 2) when one of the two bits is 1 and the carry-in is 1

Thus we can write The above two equations can be written in terms of two new signals  $P_i$  and  $G_i$ , which are shown in Figure1.



**Full Adder Stage At Stage I With Pi And Gi**

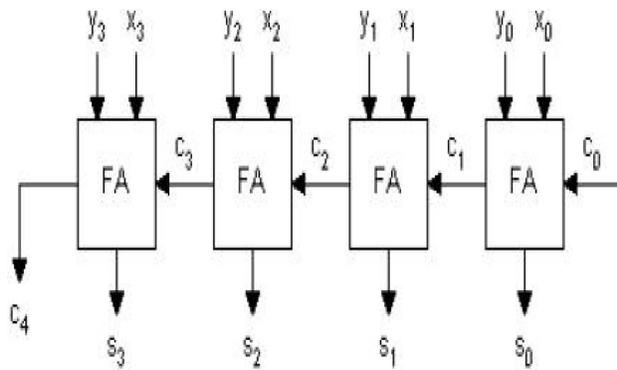
Let  $G_i$  is the carry generate function and  $P_i$  be the carry propagate function, Then we can rewrite the carry function as follows:

$$G_i = A_i \cdot B_i$$

$$P_i = (A_i \text{ xor } B_i)$$

$$S_i = P_i \text{ xor } C_i$$

$$C_{i+1} = G_i + P_i \cdot C_i$$



Carry Look-Ahead Adder

## 2.0 MULTIPLICATION ALGORITHM

There are three representations we consider:

**Signed Magnitude:** Simply multiply the magnitudes as unsigned integers. Compute the sign via XORing the signs of the numbers.

- **One's complement:** First complement the negative operands. Multiply and determine the sign. Complement the result if negative.

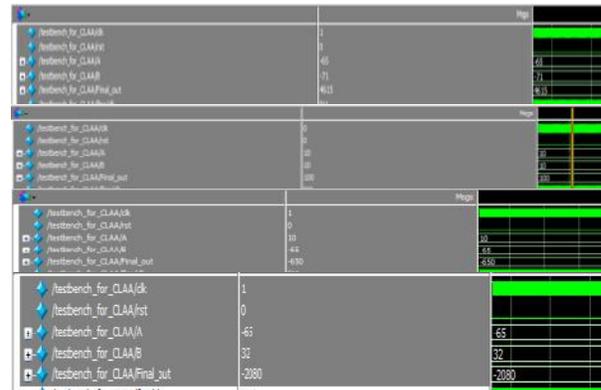
**Two's complement:** There is too much overhead in computing complements. Need an algorithm to multiply signed numbers directly.

When the multiplicand is negative and the multiplier is positive we may simply use the

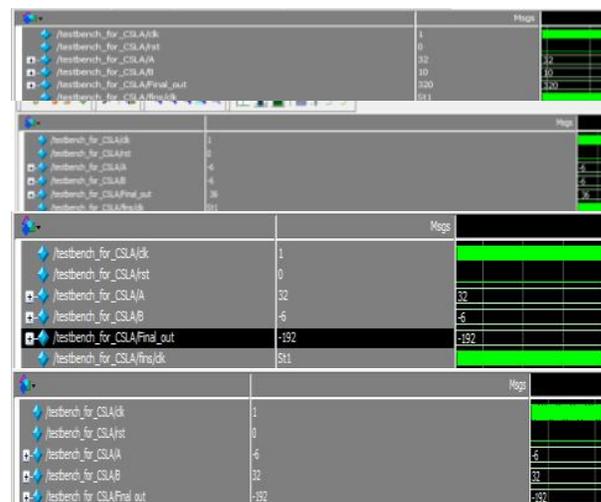
unsigned right shift algorithm. We would have to perform signed additions and carefully sign extend partial products.

## 3.0 SIMULATION RESULTS:

The HDL simulation of the two multipliers is presented in this section. In this, waveforms, timing diagrams, the design summary and the power analysis for both the CLAA and CSLA based multipliers are shown in the figures. The HDL code for both multipliers, using CLAA and CSLA, are generated. The HDL model has been developed using Modelsim6.4b. The multipliers use two 32-bit values.



Various Quadrants CLAA Multiplier Output



the proposed method for multiplication operation can be done as future work.

### Various Quadrants CSLA Multiplier

#### Performance Analysis of Adders

In this analysis table shown in figure, the delay time is nearly same, the area and the area delay product of CSLA based multiplier is reduced to 06 % when compared to CLAA based multiplier.

ADDERS	DELAY (ns)	FREQUENCY (MHz)	MEMORY (MB)
Carry Save Adder	21.666	46.155	146.896
Carry Skip Adder	19.668	50.844	151.688
Carry Select Adder	19.061	52.4631	151.156

The power performance analysis for the CLAA and CSLA based multipliers are represented in the form of the diagram shown in figure and the table above. Here the power dissipation are approximately same for both CLAA & CSLA.

	Total power	Dynami c power	Static power	Thermal power
CLAA	72.52	4.82	46.36	21.33
CSLA	72.53	4.83	46.35	21.36

#### 4.0 CONCLUSION

Performance analysis of various adders are analyzed in terms of delay, frequency and memory from these carry select adder is better parameter values than other adders. and the regular carry select is further modified for speed and area efficiency. A design and implementation of a HDL-based 32-bit Signed and unsigned multiplier with CLAA and CSLA was presented. The power analysis approximately same for both CLAA & CSLA. Thus a 06 % area delay product reduction is possible with the use of the CSLA based 32 bit signed Array multiplier than CLAA based 32 bit signed Array multiplier.

#### 5.0 FUTURE WORK

This 32 bit multiplier can be further extended to 64 bit multiplier and 128 bit multiplier using

#### 6.0 REFERENCES

- [1] Ramkumar, B. and Harish M Kittur,( 2012) "Low Power and Area Efficient Carry Select Adder", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp.1-5.
- [2] V.Vijayalakshmil, R.Seshadd, Dr.S.Ramakrishnan,(2013) Design and Implementation of 32 Bit Unsigned Multiplier Using CLAA and CSLA 978-1-4673-5301-IEEE.
- [3] P. Asadi and K. Navi, "A novel high-speed 54-54 bit multiplier", Am. J Applied Sci., vol. 4 (9), pp. 666-672. 2007.
- [4] W. Stallings, Computer Organization and Architecture Designing for Peljormance, 71h ed., Prentice Hall, Pearson Education International, USA, 2006, ISBN: 0-13-185644-8.
- [5] I. F. Wakerly, Digital Design-Principles and Practices, 4th ed., Pearson Prentice Hall, USA, 2006. ISBN: 0131733494.

