

COMPARISON OF CSLA AND CLAA 32-BIT MULTIPLIER USING VHDL

¹S. Navya, ²P. Sravya, ³N. Sravya Sruthi ⁴G. Jyothi

¹Assistant Professor, Department of ECE, CMR Engineering College, TS, India

²Research Scholar, Department of ECE, Godavari Institute of Engineering and Technology, TS, India

³Research Scholar, Department of ECE, CMR College of Engineering & Technology, TS, India.

⁴Assistant Professor, Department of ECE, Malla Reddy Engineering College (Autonomous), TS, India

ABSTRACT

This project deals with the implementation of the VLSI design of the carry look-ahead adder (CLAA) based 32-bit unsigned integer multiplier and the VLSI design of the carry select adder (CSLA) based 32-bit unsigned integer multiplier. Both the VLSI design of multiplier multiplies two 32-bit unsigned integer values and gives a product term of 64-bit values. 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. These multipliers are implemented using FPGA dumping kit. To achieve increase in speed by using the adders. Carry select adder and Carry look ahead adder.

INTRODUCTION

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 selects 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 $n \times n$ (32×32) as input and $2n$ (64) bit output. 26.93% speed improvement was possible with the use of the carry look-ahead adder.

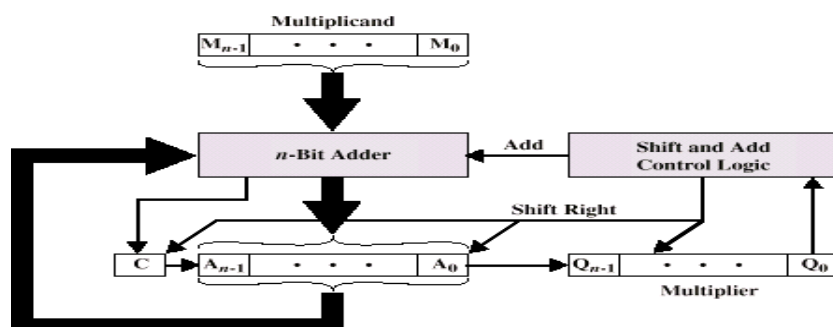


Fig. 1 Multiplication of Two N-Bit Values

SPECIFICATIONS

In this project we design a carry select adder and carry look ahead adder which are the main building blocks of the project.

The specifications all are considered for the design of carry select adder and carry look ahead adder are:

1. Technology: 180nm
2. Supply voltage: 1.8V
3. Technique: 3 different V_{th}
4. Input signal: step signal or quaternary signal
5. Number of inputs: 2
6. Number of outputs: 1

Logic table:

Quaternary logic levels	Reference voltage levels	Voltage standard
Logic 3	Vdd	1.8V
Logic 2	$5/6 \cdot V_{dd}$	1.5V
Logic 1	$3/6 \cdot V_{dd}$	0.9V
Logic 0	$1/6 \cdot V_{dd}$	0.3V

2. DESIGN ASPECTS AND APPROACH

The main aspect of the project is to design a multiplier using carry select adder and carry look ahead adder

Carry Select Adder:

In Carry select adder scheme, blocks of bits are added in two ways: one assuming a carry-in of 0 and the other with a carry-in of 1. This results in two precomputed sum and carry-out signal pairs ($s_{0i-1:k}$, c_{0i} ; $s_{1i-1:k}$, c_{1i}), later as the block's true carry-in (c_k) becomes known, the correct signal pairs are selected. Generally, multiplexers are used to propagate carries.

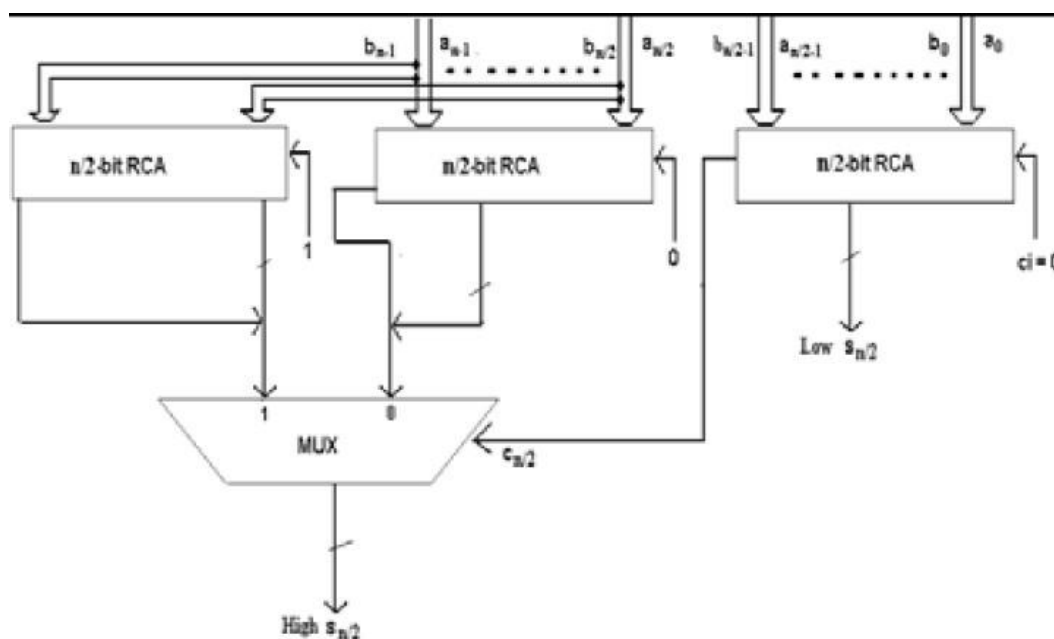


Fig.2 A Carry select Adder with 1 level using $n/2$ -bit RCA

Because of multiplexers larger area is required.

- Have a lesser delay than Ripple Carry Adders (half delay of RCA).
- Hence, we always go for Carry Select Adder while working with smaller no of bits.

Carry Look Ahead Adders:

Carry Look Ahead Adder can produce carries faster due to carry bits generated in parallel by an additional circuitry whenever inputs change. This technique uses carry bypass logic to speed up the carry propagation.

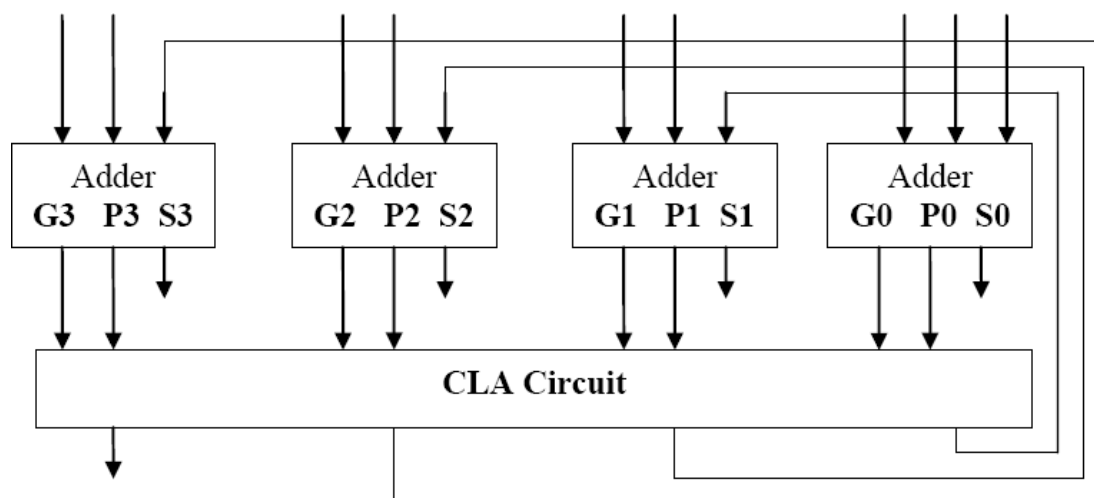


Fig. 3 4-bit CLA logic equations

Let a_i and b_i be the augends and addend inputs, c_i the carry input, s_i and c_{i+1} , the sum and carry-out to the i^{th} bit position. If the auxiliary functions, p_i and g_i called the *propagate* and *generate* signals, the sum output respectively are defined as follows. $p_i = a_i + b_i$ $g_i = a_i b_i$ $s_i = a_i \text{ xor } b_i \text{ xor } c_i$ $c_{i+1} = g_i + p_i c_i$.

- As we increase the no of bits in the Carry Look Ahead adders, the complexity increases because the no. of gates in the expression C_{i+1} increases. So practically it's not desirable to use the traditional CLA shown above because it increases the Space required and the power too.
- Instead, we will use here Carry Look Ahead adder (less bits) in levels to create a larger CLA. Commonly smaller CLA may be taken as a 4-bit CLA. So, we can define carry look ahead over a group of 4 bits. Hence now we redefine terms carry generate as [Group Generated Carry] $g[i, i+3]$ and carry propagate as [Group Propagated Carry] $p[i, i+3]$ which are defined below.

Redefined Equations:

$g[i, i+3] = g_{i+3} + g_{i+2} p_{i+3} + g_{i+1} p_{i+2} p_{i+3} + g_i p_{i+1} p_{i+2} p_{i+3}$ $p[i, i+3] = p_i p_{i+1} p_{i+2} p_{i+3}$
Now the modified block diagram for the Carry Look ahead Adder (8-bit) using levels (of 4-bit CLA) will be as block diagram below.

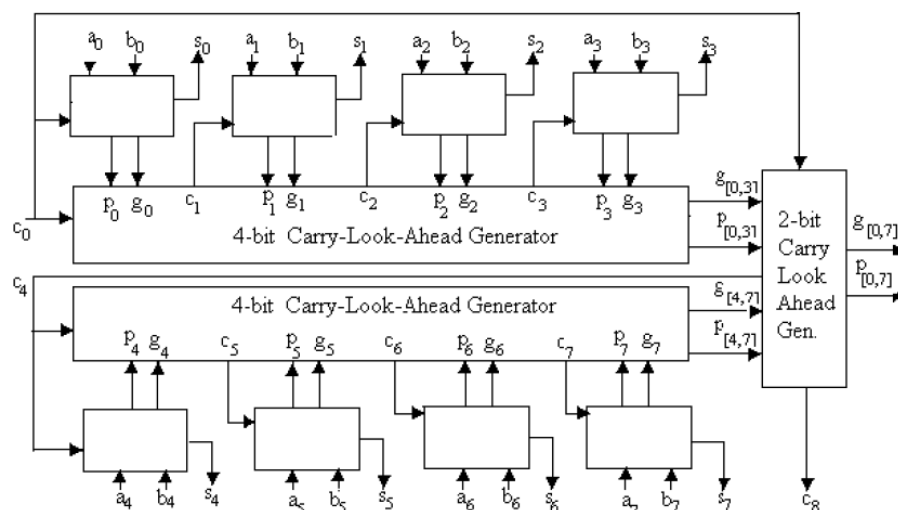


Fig.4 8bit carry look ahead generator using 2-bit CLA

Analysis of Adders:

In our project we compared 3- different adders **Ripple Carry Adders**, **Carry Select Adders** and **the Carry Look Ahead Adders**. The basic purpose of our experiment was to know the time and power trade-offs between different adders' which will give us a clear picture of which adder suits best in which type of situation during design process. Hence below we present both the theoretical and practical comparisons of all the three adders' which were taken into consideration.

3. TEST RESULTS

In this module of project, I generated carry select adder and carry look ahead adder schematic.

The schematic is as shown below:

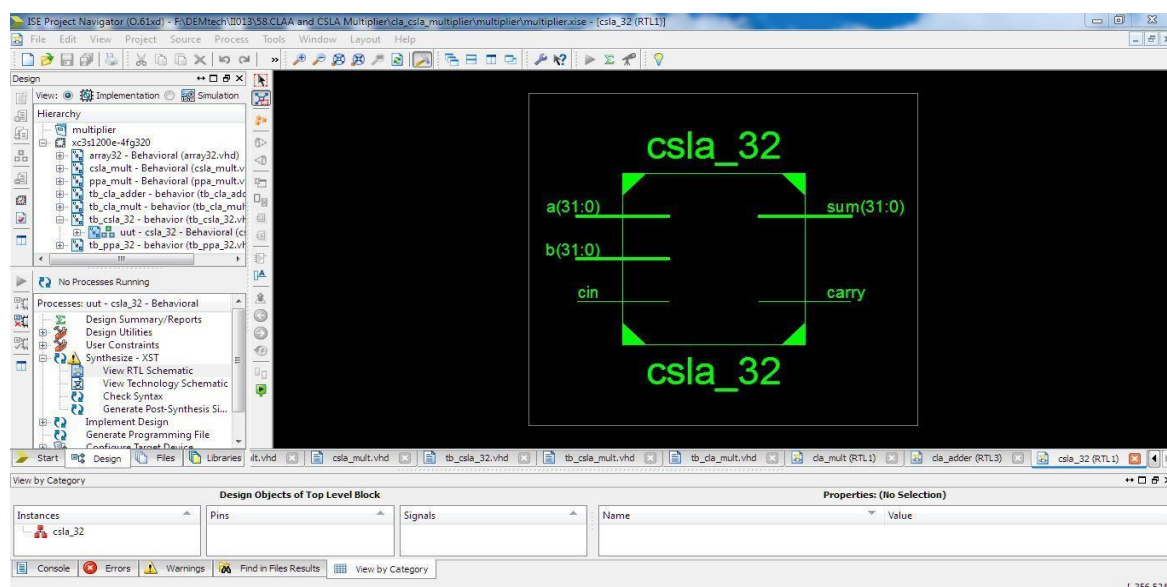


Fig 5: carry select adder schematic

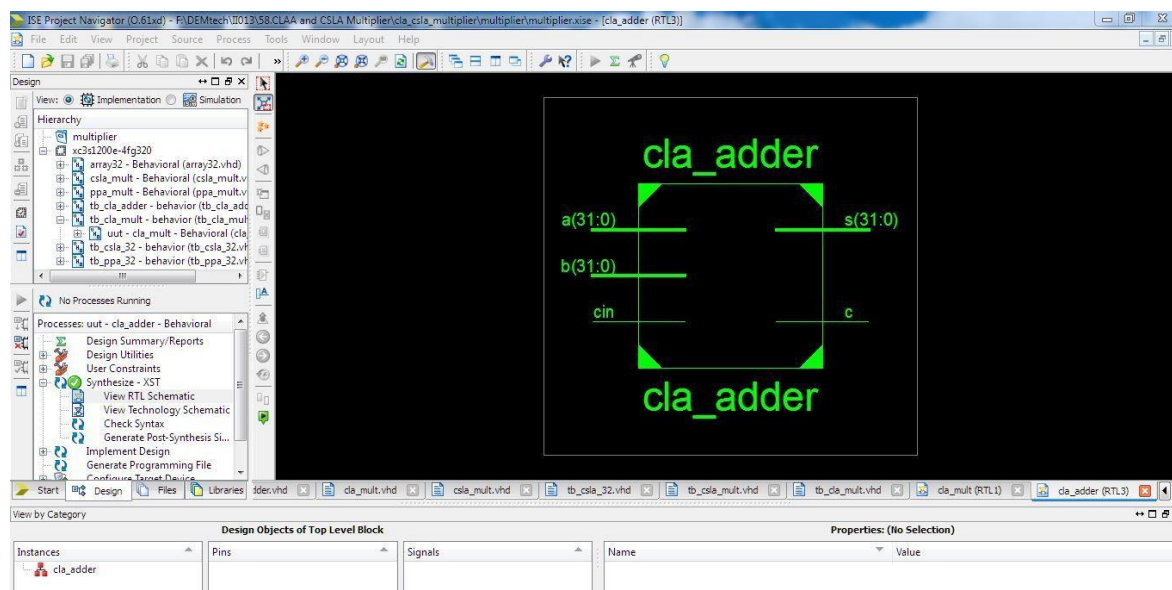


Fig 6: carry look ahead adder schematic

4. SIMULATION RESULTS AND ANALYSIS

The simulation results are going to be observed in

Once after writing the code in VHDL language the following wave forms will be observed. The source VHDL is used to generate waveforms.

The simulation results as shown below:

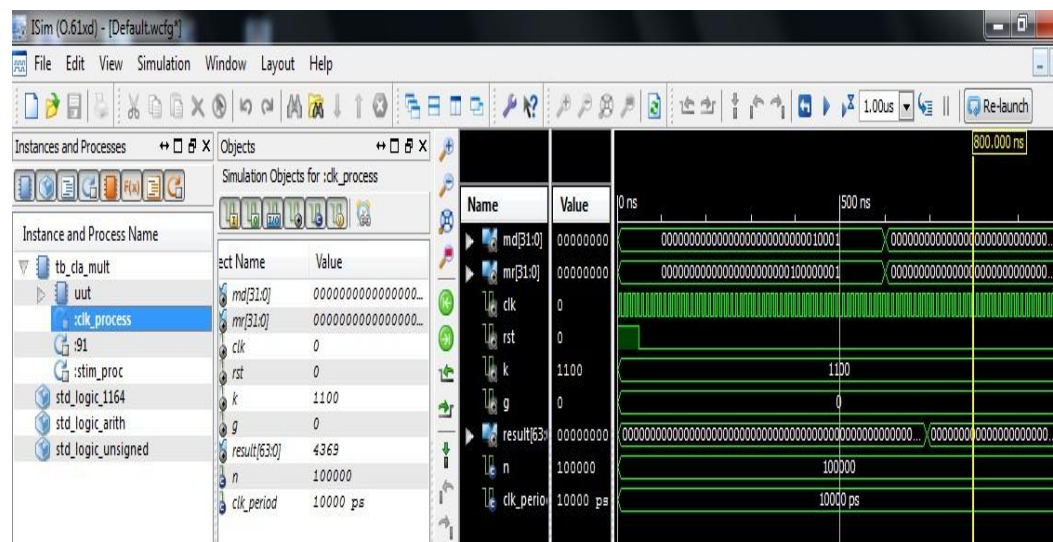


Fig. 7: Carry Look Ahead Multiplier Output Waveform

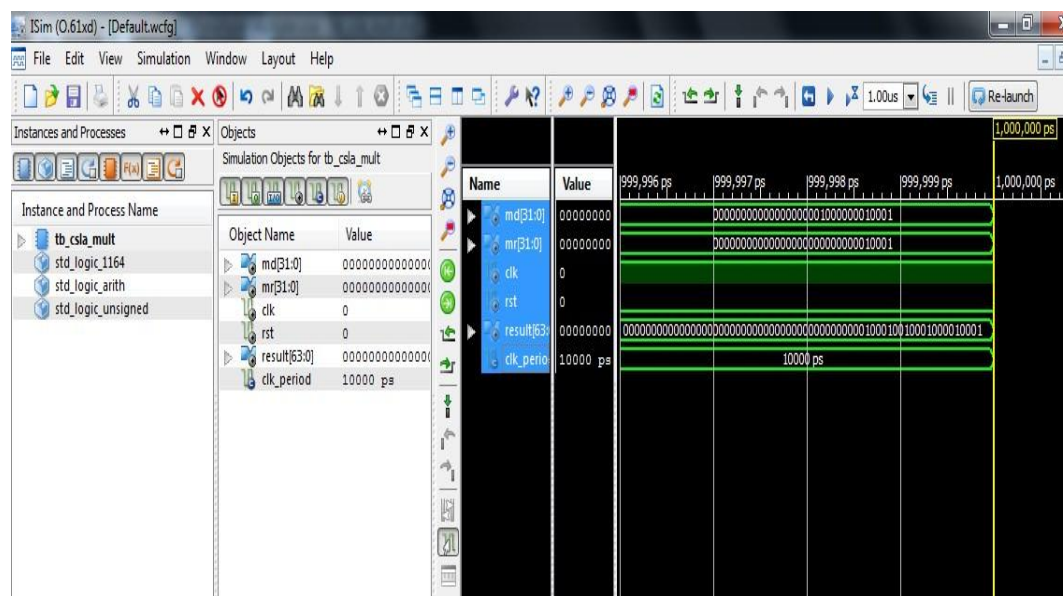


Fig. 8: carry select multiplier output waveform

Table:1 comparison of propagation time between the adders

Adder	Propagation time
Carry look ahead adder	12.377ns
Carry select adder	11.105ns

As the propagation time is reduced by carry select adder, we can say that it has achieved more speed.

5. CONCLUSION

A design and implementation of a VHDL-based 32-bit unsigned multiplier with CLAA, CSLA was presented. VHDL, a Very High-Speed Integrated Circuit Hardware Description Language, was used to model and simulate our multiplier. The carry select Adder improves the overall performance of the multiplier.

REFERENCES

1. 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, 2015.
2. P. Asadi and K. Navi, "A novel high-speed 54-54-bit multiplier", Am. J Applied Sci., vol. 4 (9), pp. 666-672. 2017.
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. <http://www.scipub.org/fulltext/ajas/ajas49666-672.pdf>.

4. W. Stallings, Computer Organization and Architecture Designing for Performance, 7th ed., Prentice Hall, Pearson Education International, USA, 2006, ISBN: 0-13-185644- 8.
5. J. F. Wakerly, Digital Design-Principles and Practices, 4th ed., Pearson Prentice Hall, USA, 2006. ISBN: 0131733494.
6. A. Sertbas and R.S. Özbey, “A performance analysis of classified binary adder architectures and the VHDL simulations”, J. Elect. Electron. Eng., Istanbul, Turkey, vol. 4, pp. 1025-1030, 2004. <http://www.istanbul.edu.tr/eng/ee/jeee/main/pages/issues/is41/41005.pdf>.