Volume-4, Issue-5, May Special Issue-2015 • ISSN No 2277 - 8160

Research Paper Engineering

# DESIGN AND IMPLEMENTATION OF AREA EFFICIENT 32-BIT UNSIGNED MULTIPLIER USING CSLA

**Smt. D. LALITHA KUMARI** Department of ECE, JNTUACEA

**P.NAGENDRABABU** Department of ECE, JNTUACEA

ABSTRACT

*This work deals with the VLSI designfor area efficient CSLA (carry select adder)based 32-bit unsigned integer multiplier by sharing the common Boolean logic term . The VLSI design for the multiplier involves multiplying two 32-bit unsigned integer values which gives a product term which is of of 64-bit. The CSLA based multiplier uses more area for performing multiplication. On the other hand , the modified CSLA based multiplier takes almost the same delay time for multiplication. The area required for modified CSLA multiplier is minimized by comparing previous methods in order to complete the multiplication. These multipliers are implemented on Altera Quartus II and timing waveforms are obtained through avan waves.*

**KEYWORDS : CSLA, MODIFIED CSLA, Array Multiplier, VHDL Modelling&Simulation.**

## I. INTRODUCTION

Digital computer arithmetic is a logic design with the aim of developing appropriate algorithms for an effective use of the hardware available. The basic operations includeaddition, subtraction, multiplication, division. This work deals with the design of multiplication operation with the help of adders. Shift and Add method is used for multiplication operation.The hardware performs relatively a simple and primitive set of Boolean operations, arithmetic operations based on a hierarchy of operations which are built with the help of simple ones.

In VLSI designs, area, speed and power are the key parameters used to determine the functioning of VLSI architecture. Additions and Multiplications are the mostly used arithmetic computations which are performed in all DSP (digital signal processing) applications. Addition is basic and fundamental operation of any digital multiplication.

This project mainly deals with reduction of Area and Time delay of multiplier design. Multiplier using CSLA [1] requires more area to design.In order to overcome this problem we are using Area efficient CSLA instead of CSLA. In this, AOI logic is used instead of lower 4-bit RCA CSLA architecture. Approximately 10% of area reduction is possible in compare to previous methods by using Area Efficient CSLA.

## II. CARRY SELECT ADDER

The architecture of CSLA computes alternative results in parallel with simultaneously selecting the correct result with either single or multiple stage hierarchical methods. In this CSLA architectureboth sum and carry bits are calculated in two alternatives $C_{in}$=O and $C_{in}$=1 . After delivering $C_{in}$, the correct computation is selected by usinga mux and to produce the desired output. In previous methods we are waiting for $C_{in}$ to calculate the sum,but here there is no waiting for$C_{in}$ to calculate the sum because here two alternatives are there. This process is used to reduce the time taken to compute the sum. This result is good improvement in speed.

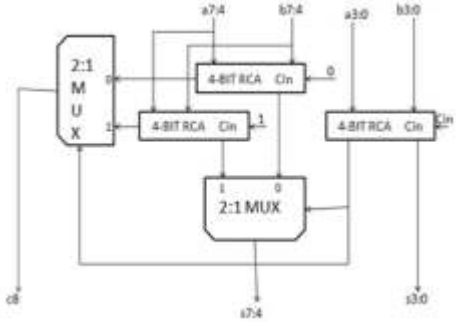



**Fig.1. Carry Select Adder**

If $C_i$=1, then, sum and $C_{out}$ are given as,

Sum[i]= a[i] Xor b[i] Xor '1' (1)

Carry [i+1]= [a(i) AND b(i)] OR [b(i) OR a(i)] (2)

If $C_{in}$ =0, then sum and $C_{out}$ are given as,

Sum[i]= Sa[i] Xor b[i] (3)

Carry[i+1]= a[i] AND b[i] (4)

The sum function is S:

$S_i=CiS_i^0+C_iS_i^1$ (5)

The carry function is :

$C_{i+1}=Ci\,C_{i+1}^0+C_iC_i+1^1$ (6)

## III .AREA-EFFICIENT CARRY SELECTS ADDER

We can design multiplier [2] using AOI logic instead of lower 4-bit RCA in CSLA architecture. This design [3] was used to reduce the area in comparision to multiplier using CSLA and CLAA.

The common Boolean logic (AOI) term was implemented in sum generation, we illustrate our proposed area-efficient carry select adder design in Fig. 2. In common Boolean logic [4] term, we only need to implement one XOR gate with one INV gate to generate the summation signal pair. As the carry-in signal is ready, we can select the correct summation output in accordance to the logicstate of carry-in signal. As for the carry propagation path, we construct one OR gate and one AND gate to anticipate possible carry input values in advance. Once the carry-in signal is ready, we can select the correct carry-out output in accordance to the logic state of carry-in signal. In this way, we can keep both the summation generation circuit of XOR gate and INV gate and the carry-out generation circuit of OR gate and AND gate in parallel. Since we still retain part of parallel architecture of conventional Carry Select Adder [5] , we can still maintain some competitiveness in speed. On the other hand, we need not prepare the duplicated adder cells in the conventional carry select adder, which can greatly reduce the transistor count and lower the power consumption.

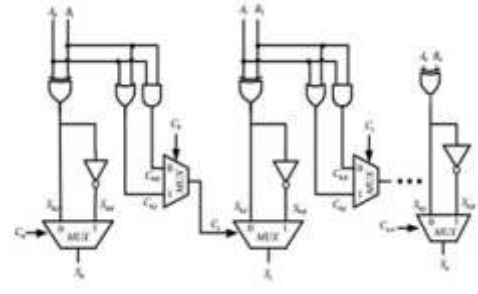


**Fig.2. Area Efficient Carry Select Adder using Boolean logic**

AOI Means AND OR INVERETER Logic

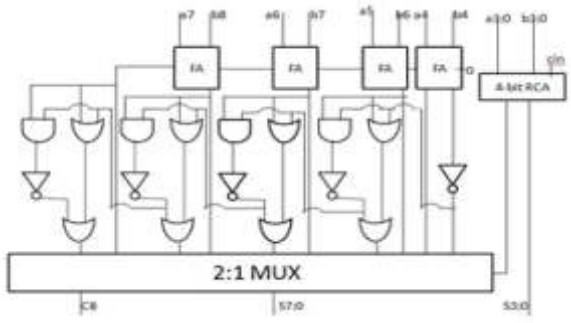



**Fig.3. Area efficient carry select adder**

Carry in =1, then Half of the Least Significant Bits of the Sum and Cout are given as,
Sum [i]= a [i] xor b [i] xor '1'
Carry [i+ 1] = [a (i) and b (I)] or [b (i) or a (i)]
If $C_{in}$ =0, then half of the LSBs of the Sum and $C_{out}$ are given as,
Sum [I]= a [i] xor b [i]
Carry [I+ 1]= [a (i) and b (i)]
Remaining half of MSB of the Sum and Carry out are given by Fig3. In CSLA Architecture total number of transistors used by Lower 4-bit RCA is 192. Those transistors are reduced to 82 by using AOI logic instead of lower 4-bit RCA,. this is called Area Efficient CSLA

## IV.MULTIPLIER FOR UNSIGNED DATA

Multiplication involves the generation of partial products(PPs), with one for each digit, as shown in Fig4 . These PPs are added to yield the final product. The multiplication of two n-bit binary integers gives a product of length up to 2n bits.

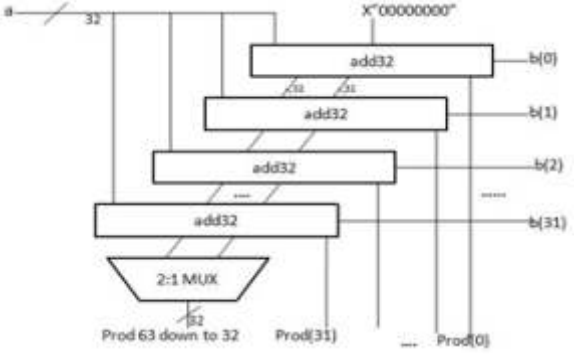



**Fig. 4. A partial schematic of the multiplier**

We use the algorithm shown below to implement the multiplication [6] for unsigned data.

## V. MULTIPLICATION ALGORITHM

Let us consider the product register size be 64 bits, multiplicand registers size be 32 bits and multiplier register size be 32 bits. The value of 'multiplier'is stored in the least significant half of the product register and 'Clear'is stored in the most significant half of the product register.
The following steps are repeated for 32 times:
1. If the LSB of the product registers is '1'
1.1 The multiplicand is added to the most significant half of the product register and
1.2Shift the content of the product register to the right by one bit .
2. If the LSB of the product registers is '0'
2.2Shift the content of the product register to the right by one bit.

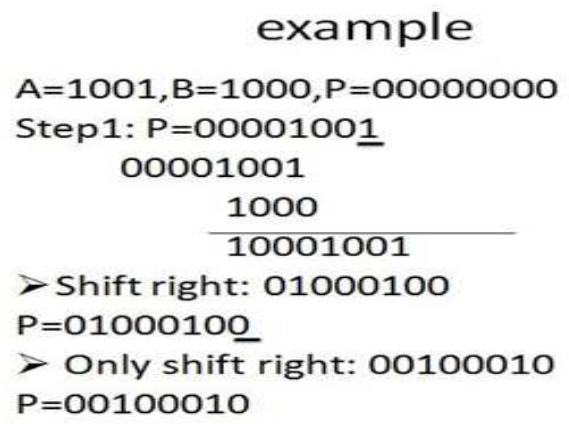
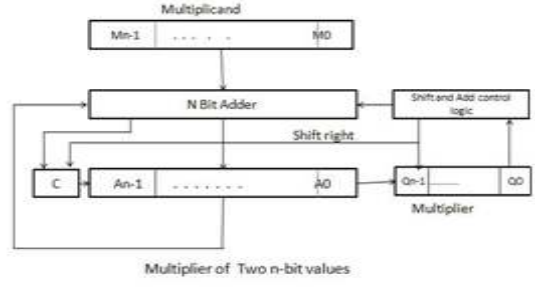



**Fig. 5. Multiplier of two n-bit values.**




## VI. VHDL SIMULATIONS

The verilog simulation for the two multipliers is shown here. The timing waveforms and the design summary for both the CSLA and Area Efficient CSLA based multipliers areas shown in the figures. The verilog code for both multipliers, using CSLA and Area Efficient CSLA are generated. The VHDL [7], [8], [9] and Verilog model is developed on Altera Quartus II and the timing waveformsare observed with the help of avan waves. These multipliers [10] makes use of 32 bit values.

| multiplierlaa Project Status (02/11/2015 - 11:25:59) | | | |
|---|---|---|---|
| Project File: | A/N..ise | Parser Errors: | No Errors |
| Module Name: | multiplierlaa | Implementation State: | Synthesized |
| Target Device: | xc7k70t-3fbg676 | •Errors: | No Errors |
| Product Version: | ISE 13.2 | •Warnings: | 3 Warnings (1 new) |
| Design Goal: | Balanced | •Routing Results: | |
| Design Strategy: | Xilinx Default (unlocked) | •Timing Constraints: | |
| Environment: | System Settings | •Final Timing Score: | |

| Device Utilization Summary (estimated values) | | | |
|---|---|---|---|
| Logic Utilization | Used | Available | Utilization |
| Number of Slice LUTs | 1701 | 41000 | 4% |
| Number of fully used LUT-FF pairs | 0 | 1701 | 0% |
| Number of bonded IOBs | 128 | 300 | 42% |

**Fig. 6.Design summary of CSLA multiplier**

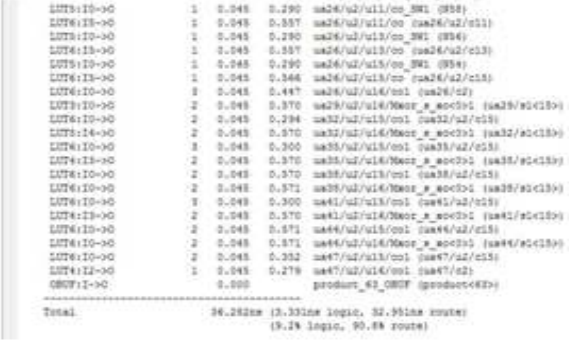



**Fig.7.Timing Analysis for CSLA based multiplier**



**Fig.8.Output of 32-bit CSLA multiplier**

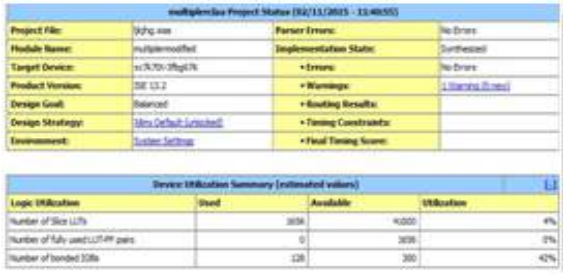


**Fig.9. Design summary of area efficient CSLA multiplier**

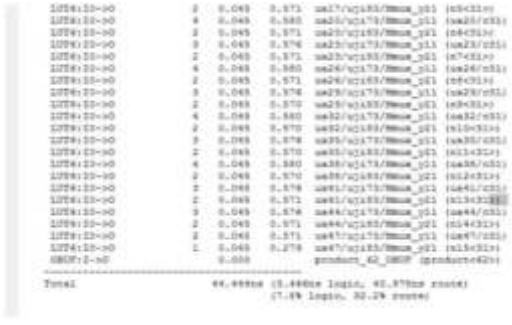


**Fig.10.Timing Analysis for area efficient CSLA based multiplier**



**Fig.11.Output of 32-bit area efficient CSLA multiplier.**

## VII. PERFORMANCE ANALYSIS

| | CSLA based multiplier | Area Efficient CSLA based multiplier |
|---|---|---|
| Area | 1705LUTs | 1696LUTs |

## VIII. CONCLUSION

The gate count required for Implementation of Area Efficient CSLA based multiplier (1696) is less than the gate count required for Implementation of CSLA based multiplier (1705)). Hence the amount of area required for implementation of Area Efficient CSLA based multiplier was reduced when compared with implementation of CSLA based multiplier.

The approach employed in this work has reduced the area and improves the speed of operation of Area Efficient CSLA based multiplier. The reduced number of gates results in area optimization.

## IX. FUTURE WORK

This work has been designed for 8-bit, 16-bit, 32-bit word size and results are allocated for parameters like area and delay. This work can be further extended for higher number of bits. New architectures can be designed in order to reduce the area and delay of the circuits. Steps may be taken to optimize the other parameters like power, frequency, number of gate clocks, length etc.

## REFERENCES

[1]V.Vijayalakshmi,"Design and implemetation of 32 bit unsigned multiplier using CLAA and CSLA ", IEEE 2013.|| [2] P. Asadi and K. Navi, "A novel highs-speed 54-54 bit multiplier",Am. J Applied Sci., vol. 4 (9), pp. 666-672. 2007.|| [3]W.Stallings,Computer Organization and Architecture Designing for Peljormance, 71h ed., Prentice Hall, Pearson Education International, USA, 2006, ISBN: 0-13-185644-8.|| [4] 1. F. Wakerly, Digital Design-Principles and Practices, 4th ed.,Pearson Prentice Hall, USA, 2006. ISBN: 0131733494|| [5] A. Sertbas and R.S. Ozbey, "A performance analysis of classifiedbinary adder architectures and the VHDL simulations", J Elect.Electron. Eng., Istanbul, Turkey, vol. 4, pp. 1025-1030,2004.|| [6] P. S. Mohanty, "Design and Implementation of Faster and Low PowerMultipliers", Bachelor Thesis. National Institute of Technology,Rourkela, 2009.|| [7] S. Brown and Z. Vranesic, Fundamentals of Digital Logic with VHDLDesign, 2nd ed., McGraw-Hill Higher Education, USA, 2005. ISBN:0072499389.|| [8] J. R. Armstrong and F.G. Gray, VHDL Design Representation andSynthesis, 2nd ed. Prentice Hall, USA, 2000. ISBN: 0-13-021670-4.|| [9] Z. Navabi, VHDL Modular Design and Synthesis of Cores andSystems, 3rd ed., McGraw-Hill Professional, USA, 2007. ISBN:9780071508926.|| [10] P. C. H. Meier, R. A. Rutenbar and L. R. Carley, "ExploringMultiplier Architecture and Layout for low Power", CIC'96, 1996.