Design and Implementation of 64-Bit Multiplier Using CLAA and CSLA

Shaik Meerabi¹, Krishna Prasad Satamraju²

Department of Electronics and communication Engineering, VVIT, Nambur, Guntur, A.P, India

Abstract: This paper deals with comparative study of the Carry Look-Ahead Adder (CLAA) based 64-bit unsigned integer multiplier and Carry Select Adder (CSLA) based 64-bit unsigned integer multiplier. Multiplication is a fundamental operation in most of the signal processing algorithms. Multipliers occupy large area, long latency and consume considerable power. Therefore there is a need for designing a multiplier that consumes less power. Moreover the digital system's efficiency is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system, and consumes more area. Hence, optimizing the speed, area and delay of the multiplier is a major design issue. Carry Select Adder (CSLA) is one of the fastest adders used in many applications to perform fast arithmetic functions. From the structure CSLA there is a scope for reducing area and delay by using Common Boolean Logic (CBL). This work evaluates the performance of the proposed designs in terms of area, delay, and power. The power dissipation is same for both CLAA based multiplier and CSLA based multiplier. But the area delay product of modified CSLA based multiplier is reduced to 6% when compared to CLAA based multiplier. These multipliers are simulated and synthesized using Modelsim6.4b and Xilinx 10.1.

Keywords: CLAA, CSLA, CBL, Delay, Area

1. Introduction

Digital computer arithmetic is an aspect of logic design with the objective of developing appropriate algorithms in order to achieve efficient utilization of the available hardware. The basic operations are addition, subtraction, multiplication and division. Addition operation plays an important role in the design of multiplier as repeated form of addition operations and shifting results in the multiplication operation.

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 numbers need 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 of these systems are dominated by multipliers. Unfortunately, portable devices mostly operate with standalone 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. 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 size of multiplier is directly proportional to the square of its resolution.

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 design, speed, power and chip area are the most often used measures for determining the performance and efficiency of the VLSI architecture.

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 Common Boolean Logic is presented to speed up addition process. 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 paper we are going to compare the performance of different adders which are used to design the multipliers, based on area and time needed for performing multiplication.

2. Carry Look Ahead Adder

The Carry Look Ahead Adder (CLAA) 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 and 64-bit Carry Look Ahead Adder is shown in Figure2.



Figure 1: Full Adder Stage at Stage I with P_i and G_i

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:

$$\mathbf{G}_{\mathbf{i}} = \mathbf{A}_{\mathbf{i}} \cdot \mathbf{B}_{\mathbf{i}}.$$
 (1)

$$P_i = (A_i \operatorname{xor} B_i).$$
 (2)

$$S_i = P_i \text{ xor } C_i. \tag{3}$$

$$C_{i+1} = G_i + P_i \cdot C_i.$$
(4)

3. Carry Select Adders

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 to 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 contains one k/2 bit adder for the lower half of the bits and two k/2 bit adder for upper half of the bits, where k is the length of the input. 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.



Figure 2: 64 bit Carry Look-Ahead Adder

This technique of dividing adder into stages increases the area utilization but addition operation fastens. Time delay= time to compute first section + time to select sum from subsequent sections. Figure4 shows a block diagram of 64-bit carry select adder.



Figure 3: Basic Carry Select adder circuit



Figure 4: Block Diagram of Regular 64 bit Carry Select Adder

4. Proposed System: Modified Carry Select Adder

For area minimization, the Number of slices should be minimized. It can be achieved by altering the Full Adder. The logic design of Full Adder is shown in figure5.Its truth table is shown in Table1. The proposed System is designed with the help of NOT, OR, XOR and AND gates. Here the size of the adder is reduced by Boolean algebra so that the number of slices is reduced. When the number of slices is reduced automatically the area and delay are also reduced



Figure 5: Logic Diagram of Full Adder

Volume 4 Issue 2, February 2015 <u>www.ijsr.net</u> Licensed Under Creative Commons Attribution CC BY

International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

(5)

(6)

Table 1: I ruth Table for full adder						
C _{IN}	Α	В	SUM	CARRY		
0	0	0	0	0		
0	0	1	1	0		
0	1	0	1	0		
0	1	1	0	1		
1	0	0	1	0		
1	0	1	0	1		
1	1	0	0	1		
1	1	1	1	0		

T 1 1 C C 11 1 1

Logical equations for full adder are as below:

Sum = A XOR B XOR
$$C_{IN}$$

 $Carry = (A.B) + C_{IN} (A + B)$ Minimized Logical equations for full adder:

FOR C_{IN}=0

$$S_{v} = A XOR B$$
(7)
$$C_{v} = A.B$$
(8)

$$S_{k} = NOT S_{v}$$
(9)
$$C_{k} = A + B$$
(10)

The logic for the required carry is chosen using 2:1 mux circuits. This results in reduced gate numbers than existing carry select adders. Hence it has the area minimized and optimal computational speed.



Figure 6: modified carry select adder using Common Boolean Logic

5. Multiplication Algorithm

Let the product register size be 128 bits. Let the multiplicand registers size be 64 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 64 times:

- 1) If the least significant bit of the product register is "1" then add the multiplicand to the most significant half of the product register.
- 2) 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. Figure7 shows a block diagram of two n-bit values.



Figure 7: Multiplier of two n-bit values

6. 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. 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 64-bit values. Here we are dealing with the comparison in the bit range of n*n (64*64) as input and 2n (128) bit output.

The performance analysis of the area and delay is shown in the table2. The area delay product of modified CSLA based multiplier is reduced to 6% when compared to CLAA based multiplier.







Figure 9: 64 bit multiplier using CSLA

The power performance analysis for the CLAA and CSLA based multipliers are shown in table3. Here the power dissipation is approximately same for both CLAA & CSLA.

Table 2: Performance analysis of area and delay

Multiplier type	Delay(ns)	Area	Area Delay Product
CLAA based 64-bit Multiplier	224.370	4787 Number of slices	1074059.19
Regular CSLA based 64-bit Multiplier	260.910	10718 Number of slices	2796433.38
Modified CSLA based 64-bit Multiplier	210.881	4709 Number of slices	993038.629

Table 3: power analysis				
Multiplier type	Total Power			
	(1111)			
CLAA based	203			
64-bit Multiplier				
Regular CSLA				
based 64-bit	203			
Multiplier	203			
Modified CSLA				
based 64-bit	203			
Multiplier	203			

7. Conclusion

We present a design and implementation of 64-bit unsigned multiplier with CLAA and CSLA. The design is tested with the existed adders. Analysis shown that there is a reduction of about 6% in the area delay product. The multiplier is proved to be efficient in terms of power utilization as well.

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 BitUnsigned Multiplier Using CLAA and CSLA 978-1-4673-5301-IEEE.
- [3] P. Asadi and K. Navi, "A novel highs-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] 1. F. Wakerly, Digital Design-Principles and Practices, 4th ed.,Pearson Prentice Hall, USA, 2006. ISBN: 0131733494.
- [6] B. Ramkumar and Harish M Kittur," Low-Power and Area-Efficient Carry Select Adder", IEEE Transactions on Very Large Scale Integration(VLSI) Systems, VOL. 20, NO. 2, February 2012.
- [7] Mariano Aguirre-Hernandez and Monico Linares-Aranda, (2011) "CMOS Full-Adders for Energy-Efficient Arithmetic Applications".IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 19, NO. 4.
- [8] J. M. Rabaey, Digital Integrated Circuits," IEEE Trans. on VLSI Systems, 2003.
- [9] O. Bedrij, "Carry Select Adder," IRE Trans. on Electronic Computers, Vol. EC-11, pp. 340-346, 1962

Author Profile

Shaik.Meerabi is pursuing M. Tech with specialization in VLSI and Embedded Systems from Vasireddy Venkatadri Institute of Technology (VVIT) Nambur. She completed her B. Tech in Electronics and Communication Engineering from Acharya Nagarjuna University.

Krishna Prasad Satamraju obtained in Masters Degree in Technology from SRM University, with specialization in Digital Communications and Networking. He is presently working as Assistant Professor in the Department of ECE, VVIT, Nambur. He has over Eight years of teaching experience. His research areas include Embedded Systems, RTOS and Linux based Device driver development.