RESEARCH ARTICLE OPEN ACCESS

Design and Performance Analysis of Reversible Logic Four Quadrant Multiplier Using CSLA and CLAA

Mr. P. Dileep Kumar Reddy, Mrs. T.V. Ananthalakshmi
M.Tech (pursuing)-VLSI
M.Tech- VLSI & EMBEDDED SYSTEMS

Abstract
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. There has been extensive work on low-power multipliers at technology, physical, circuit and logic levels. 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. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area- speed constraints has been designed with reversible logic gates. The reversible logic has the promising applications in emerging computing paradigm such as quantum computing, quantum dot cellular automata, optical computing, etc. In reversible logic gates there is a unique one-to-one mapping between the inputs and outputs.

Keywords: CLSA, CLAA, Array multiplier, Delay, Reversible logic gates, HDL 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 a fundamental operation for any digital 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 2n (64) bit output.

Hence, to design a better architecture the basic adder blocks must have reduced delay time consumption, area efficient architectures, reduced power and this can be achieved by reversible logic gates. The demand is of DSP style systems for both less delay time and less area, less power requirement for designing the systems. Our interest is in the basic building blocks of arithmetic circuits that dominate in DSP applications, Low power VLSI architectures, computer applications such as Quantum Computing, Nanotechnology, Sprintronics and Optical Computing. Reversibility plays an important role when energy efficient computations are to be designed.

II. Reversible logic gates

A reversible logic gate is an n-input n-output logic device with one-to-one mapping. This helps to determine the outputs from the inputs and also the inputs can be uniquely recovered from the outputs. Also in the synthesis of reversible circuits direct fan-Out is not allowed as one–to-many concept is not reversible. However fan-out in reversible circuits is achieved using additional gates. A reversible circuit

should be designed using minimum number of reversible logic gates. From the point of view of reversible circuit design, there are many parameters for determining the complexity and performance of circuits.

- The number of Reversible gates (N): The number of reversible gates used in circuit.
- The number of constant inputs (CI): This refers to the number of inputs that are to be maintained constant at either 0 or 1 in order to synthesize the given logical function.
- The number of garbage outputs (GO): This refers to the number of unused outputs present in a reversible logic circuit. One cannot avoid the garbage outputs as these are very essential to achieve reversibility.
- Quantum cost (QC): This refers to the cost of the circuit in terms of the cost of a primitive gate. It is calculated knowing the number of primitive reversible logic gates (1*1 or 2*2) required to realize the circuit.

They are no of reversible logic gates are available. Among them are peres gates is one with low quantum cost.

The figure below shows a 3*3 Peres gate. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The output is defined by P = A, Q = A Exor B and R=AB Exor C. Quantum cost of a Peres gate is 4. In the proposed design Peres gate is used because of its lowest quantum cost.

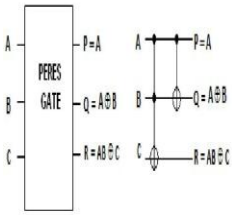

Figure: peres gate

Truth table of peres gate

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

A full- adder using two Peres gates is as shown in fig 6. The quantum realization of this shows that its quantum cost is 8 two Peres gates are used

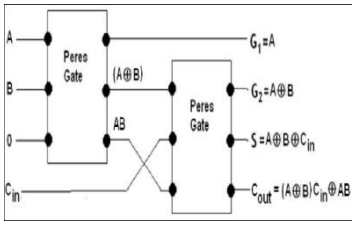

Figure: Full adder using two peres gate

III. 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

$$c_{i+1}=a_i b_1+(a_i \oplus b_i)c_i$$
$$s_i=(a_i \oplus b_i)\oplus c_i$$

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

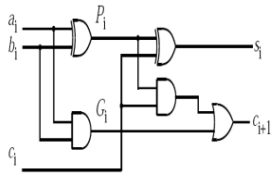

Figure1:Full Adder Stage At Stage I With P_i And G_i

$$C_{i+1}=G_i+P_i.C_i$$
$$S_i=P_i \oplus C_i$$
$$G_i=a_i.b_i$$
$$P_i=a_i \oplus b_i$$

G_i and Pi are called the carry generate and carry propagate terms, respectively. Notice that the generate and propagate terms only depend on the input bits and thus will be valid after one and two gate delay, respectively. If one uses the above expression to calculate the carry signals, one does not need to wait for the carry to ripple through all the previous stages to find its proper value. Let's apply this to a 4-bit adder to make it clear.

Putting i=0,1,2,3 in Equation 5, we get

$$C_1=G_0+P_0C_0$$
$$C_2=G_1+P_1G_0+P_1P_0C_0$$
$$C_3=G_2+P_2G_1+P_2P_1G_0+P_2P_1P_0C_0$$
$$C_4=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1G_0+P_3P_2P_1P_0C_0$$

Notice that the carry-out bit,Ci+1, of the last stage will be available after four delays: two gate delays to calculate the propagate signals and two delays as a result of the gates required to implement Equation 13.Figure below shows that a 4-bit CLA is built using gates to generate the P_i and G_i signals and a logic block to generate the carry out signals according to Equations above.

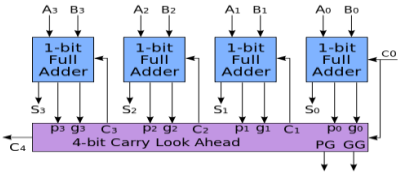

Figure: Carry Look Adder

The disadvantage of CLA is that the carry logic block gets very complicated for more than 4-bits.For that reason, CLAs are usually implemented as 4-bit modules and are used in a hierarchical structure to realize adders that have multiples of 4-bits.

IV. CARRY SELECT LOOK AHEAD ADDER

The concept of CSLA is to compute alternative results in parallel and subsequently selecting the correct result with single or multiple stage hierarchical techniques. In CSLA both sum and carry bits are calculated for two alternatives Cin=O and 1. Once Cin is delivered, the correct computation is chosen using a mux to produce the desired output. Instead of waiting for Cin to calculate the sum, the sum is correctly output as soon as Cin gets there. The time taken to compute the sum is then avoided which results in good improvement in speed.

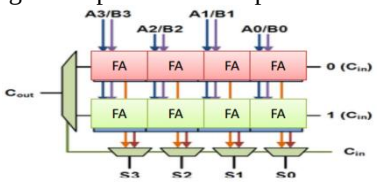

Figure:4-bit carry select adder

If Carry in =1, then the sum and carry out are given by,
Sum (i) =a (i) xor b (i) xor '1'.
Carry (i+ 1) = (a (i) and b (i)) or (b (i) xor a (i))
If Carry in =0, then the sum and carry out are given by,
Sum (i) = a (i) xor b (i).
Carry (i+ 1) = (a (i) and b (i)).
The sum function:

$$S_i = C_i S_i^0 + C_i S_i^1$$

The carry function:

$$C_{i+1} = C_i C_{i+1}^0 + C_i C_{i+1}^1$$

V. 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.

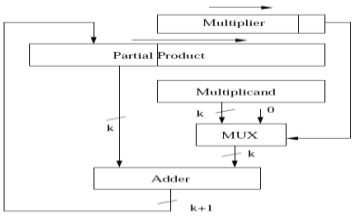

Figure: Architecture for implementing the right shift multiplication algorithm.

What if both numbers are negative? Remember we can view the sign bit as a digit with negative weight, e.g.

$$A = -a_{k-1}r^{k-1} + a_{k-2}r^{k-2} + \ldots + a_1 r + a_0$$

signed weight $-r^{k-1}$ of digit a_{k-1}. We handle all bits of the multiplier as before but handle the last bit (or sign bit)by adding the 2's complement of the last partial product.

VI. VERILOG SIMULATIONS

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 Altera Quartus II . The multipliers use two 32-bit values.

Figure: Various Quadrants CLAA Multiplier Output

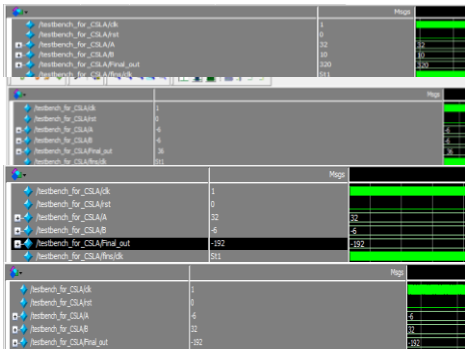

Figure19: Various Quadrants CSLA Multiplier Output

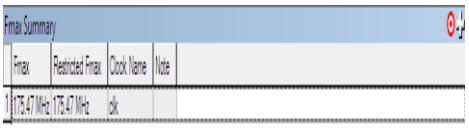

Figure20: CLAA Multiplier Delay

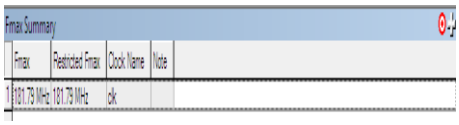

Figure: CSLA Multiplier Delay

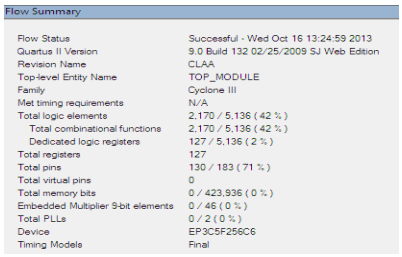

Figure: Design Summary CLAA Multiplier Output

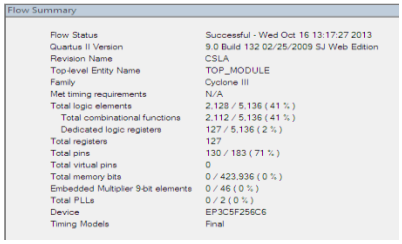

Figure: Design Summary Of CSLA Multiplier Output

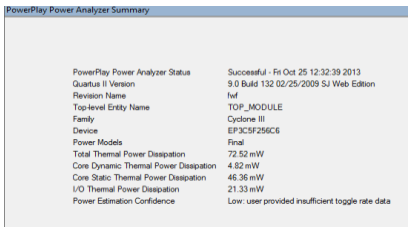

Figure: Power Analysis Of CLAA Multiplier Output

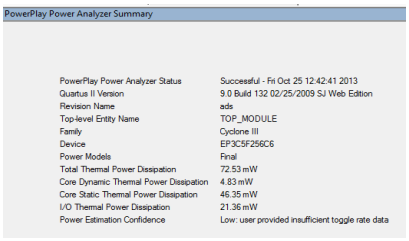

Figure: Power Analysis Of Csla Multiplier Output

VII. PERFORMANCE ANALYSIS TABLE

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.

Table3: Analysis Table

MULTIPLIER TYPE	DELAY	AREA	DELAY AREA PRODUCT
CLAA based multiplier	5.69ns	2170	12347.3
CSLA based multiplier	5.5ns	2128	11704

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	Dynamic power	Static power	Thermal power
CLAA	72.52	4.82	46.36	21.33
CSLA	72.53	4.83	46.35	21.36

VIII. CONCLUSION

The design and implementation of a HDL-based 32- bit signed reversible logic multiplier with CLAA and CSLA was presented. HDL, 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 over CLAA. 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.

REFFERENCES

[1] W. Stallings, Computer Organization and Architecture Designing for Peljormance, 71h ed., Prentice Hall, Pearson Education

International, USA, 2006, ISBN: 0-13-185644-8.

[2] J. F. Wakerly, Digital Design-Principles and Practices, 4th ed., Pearson Prentice Hall, USA, 2006. ISBN: 0131733494.

[3] DMITRI MASLOV " REVERSIBLE LOGIC SYNTHESIS", Research gate MCS University of New Brunswick,2002,2003 https://citeseerx.ist.psu.edu_viewdoc_download_doi=10.1.1.84

[4] Ripple Carry and Carry Look ahead Adders www.ece.uvic.ca_~fayez_courses_ceng465_lab_465_project1_adders

[5] *ISSN: 2278 – 1323* Comparative Performance Analysis of Different CMOS Adders Using 90nm and 180nmTechnology by Jatinder Kumar, Parveen Kaur *(IJARCET) Volume 2 Issue 8, August 2013*

[6] ISSN: 2319 - 1163 T. NagaBabu1, D.Sounder2, B.Subhakara Rao3, P. Bose Babu " A LOW POWER ADDER USING REVERSIBLE LOGIC GATES" Volume: 1 Issue: 3

[7] Vivek V. Shend, Aditya K. Prasad,Igor L. Markov,John P. Hayes" Synthesis of Reversible Logic Circuits" University of Michigan, Ann Arbor, MI 48109-2122

[8] ISSN 2278-9901 "REVIEW OF FULL ADDER USING REVERSIBLE LOGIC" SEEMA RAJPUT, RITA JAIN & AWADHESH K. G. KANDU (IJECE) Vol. 2, Issue 3, July 2013, 97-102

[9] (ISSN 2250-2459, ISO 9001:2008 Certified Journal)"A Review on Reversible Logic Gates and their Implementation" Raghava Garipell, P.Madhu Kiran, A.Santhosh Kumar, Volume 3, Issue 3, March 2013

[10] "A PERFORMANCE ANALYSIS OF CLASSIFIED BINARY ADDER ARCHITECTURES AND THE VHDL SIMULATIONS" Ahmet SERTBAŞ R.Selami ÖZBEY, YEAR VOLUME NUMBER: 2004: 4: 1

[11] ISSN: 2049-3444 IJET Volume 2 No. 1, January, 2012 © 2011 – IJET Publications UK. All rights reserved. 44 Design of Optimal Reversible Carry Look-Ahead Adder with Optimal Garbage and Quantum Cost Lafifa Jamal, Md.Shamsujjoha, Hafiz Md. Hasan Babu