Research Inventy: International Journal Of Engineering And Science
Vol.4, Issue 12 (December 2014), PP 12-16
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

COSA and CSA based 32 -bit unsigned multipler

Lekshmi Suresh
Electronics and Communication MCET,Pathanamthitta
Pathanamthitta,India

ABSTRACT: *In this paper, design of two different arraymultipliers are presented, one by using conditional sum (COSA) logic for addition of partial product terms and another by introducing Carry Save Adder (CSA) in partial product lines. The multipliers presented in this paper were all modeled using VHDL (Very High Speed Integration Hardware Description Language) for 32-bit unsigned data. The comparison is done on the basis of three performanceparameters i.e. Area, Speed and Power consumption. To design an efficient integrated circuit in terms of area, power and speed, has become a challenging task in modern VLSI design field. Previously in the literature, performance analysis was carried out between multiplier using Ripple carry adder (RCA) and by using CLA. In this work, same multiplier is designed by using CSA logic and compare it's performance with the multiplier designed by using CSLA logic. Multiplier with CSA gives better result in terms of speed (78.3% improvement), area (reduced by 4.2%) and power consumption (decreased by 1.4%).*

INDEX TERMS: *Multiplier, Carry Save Adder, Conditional Sum Adder, VHDL Simulation*

I. INTRODUCTION

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. Multipliers are most commonly used in various electronic applications e.g. Digital signal processing in which multipliers are used to perform various algorithms like FIR, IIR etc. Earlier, the major challenge for VLSI designer was to reduce area of chip by using efficient optimization techniques to satisfy MOORE'S law. Then the next phase is to increase the speed of operation to achieve fast calculations like, in today's microprocessors millions of instructions are performed per second. Speed of operation is one of the major constraints in designing DSP processors and today's general-purpose processors. However area and speed are two conflicting constraints. So improving speed results always in larger areas. Now, as most of today's commercial electronic products areportable like Mobile, Laptops etc. that require more battery back up. Therefore, lot of research is going on to reduce power consumption. So, in this paper it is tried to find out the best solution to achieve low power consumption, less area required and high speed for multiplier operation. 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 conditional sum (COSA) based multiplier the area of calculation of the carry save adder (CSA) 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.

II. CONDITIONAL SUM ADDER

A conditional-sum adder is actually a (log2 k)-level carry-select adder. Consider that we need to add ai and bi which are the ith bits of operands A and B,respectively.In the conditional sum algorithm ,instead of waiting for the arrival of carry value,instead of waiting for the arrival of carry, conditional carry and conditional sum are generated by considering both possible values of the carry-in bit.The result of adding ai,bi, and either a 0 or 1 carry-in bit is a two bit number.

In general, the algorithm is given by:

If Carry in =1, then the sum and carry out are given by,

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 Carry in =0, then the sum and carry out are given by,

The sum function:

Sum (i) = a (i) xor b (i).

Carry (i+ 1) = (a (i) and b (i)).

Consider the following example,

	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$A =$	1	1	0	1	1	0	1
$B =$	0	1	1	0	1	1	0
S_i^0	1	0	1	1	0	1	1
C_i^0	0	1	0	0	1	0	0
S_i^1	0	1	0	0	1	0	0
C_i^1	1	1	1	1	1	1	1

Fig.1.Conditional sum and Conditional carries

The conditional sum adder of 8-bit is given by,

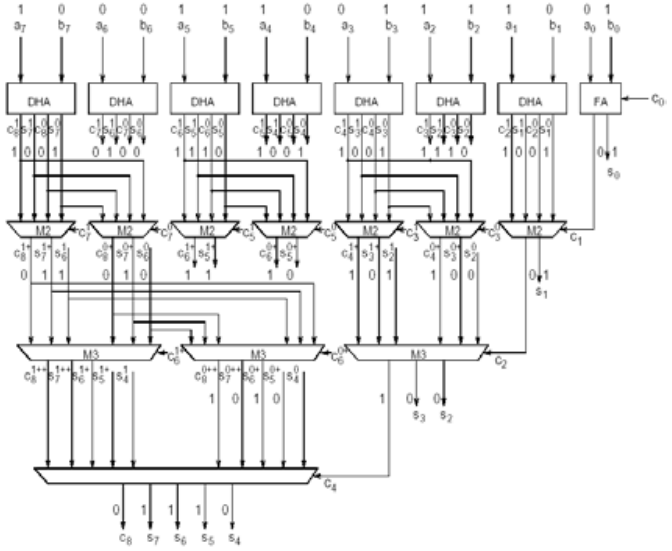

Fig.2. Carry Select Adder with 8-bit

III. CARRY SAVE ADDER

Basically, carry save adder is used to compute sum of three or more n-bit binary numbers. Carry save adder is same as a full adder. The full adder is usually implemented with a reduced delay from Cin to Cout because the carry chain is the critical delay path in adders. Unfortunately, there is no single carry chain in the carry save adder trees in multipliers. Here we are computing sum of two 32-bit binary numbers, so we take 32 full adders at first stage. Carry save unit consists of 32 full adders, each of which computes single sum and carry bit based only on the corresponding bits of the two input numbers. Let X and Y are two 32-bit numbers and produces partial sum and carry as S and C :

Si = Xi xor Yi (3)

Ci = Xi and Yi (4)

The final addition is then computed as:

1. Shifting the carry sequence C left by one place.
2. Placing a 0 to the front (MSB) of the partial sum sequence S.
3. Finally, a ripple carry adder is used to add these two together and computing the resulting sum.

This can be represented in figure 3.

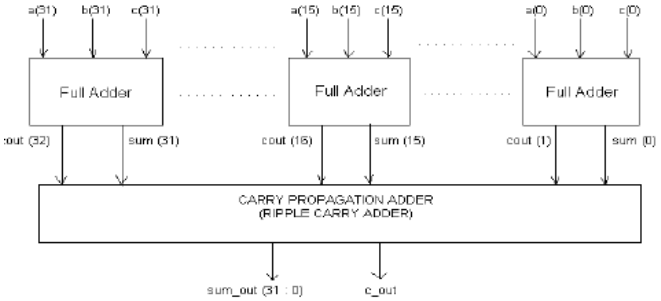

Fig.3. Computational flow of Carry Save Adder

IV. IMULTIPLTER FOR UNSIGNED DATA

In Array multiplier, almost identical calls array is used for generation of the bit-products and accumulation. All bit-products are generated in parallel and collected through an array of full adders or any other type of adders and final adder. Array multiplier has a regular structure that simplifies the wiring and the layout. Therefore, among other multiplier structures, array multiplier takes up the least amount of area but it is also the slowest with the latency proportional to *O* (Wd) where Wd is the word length of the operand. Instead of using Ripple carry adder (RCA), here we use Carry lookahead logic and Carry save adder for adding each group of partial product terms because RCA is slowest adder among all other fast adders available.Figure 3& 4 shows architecture of 32-bit Array multiplier using CSLA and CSA respectively to add each group of partial products in parallel.

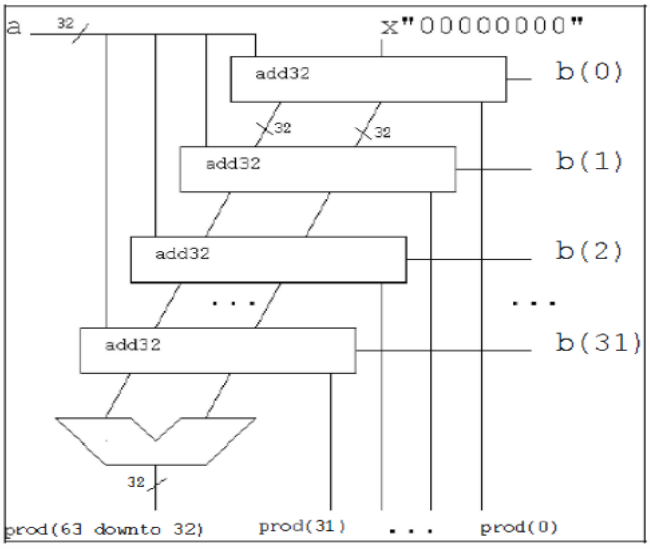

Fig.4. Multiplier logic using COSA

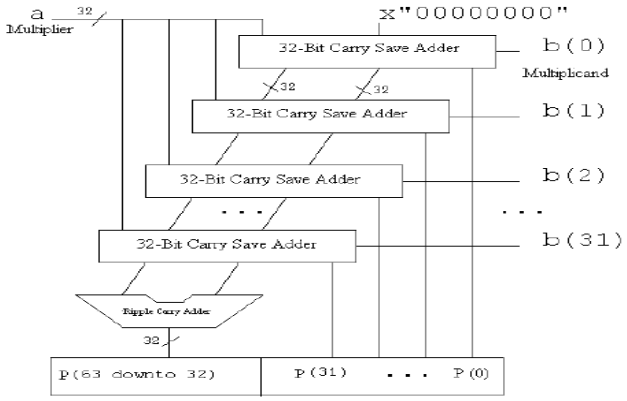

Fig.5. Multiplier implemented using CSA logic

V. MULTIPLICATION ALGORITHM

There are product registers , multiplier and multiplicand registers. They are of 64, 32 and 32-bits respectively .The algorithm used for the multiplication is given as follows,

[1] Clear the product register and the half MSB of the product register
[2] Check the last bit of the multiplier, if it is '1' then add with the product register
[3] Shift the product register once
[4] If the lsb of multiplier is '0',then only shift the product register only once
[5] Add all the partial products

This algorithm can be represented below:

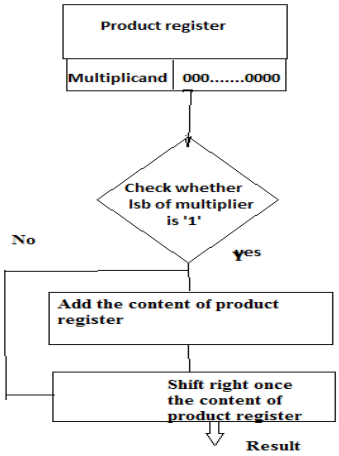

Fig 6. Algorithm

VI. SIMULATION RESULTS

The VHDL simulation of the two multiplier is presented in this section. For simulation Modelsim SE 6.2c tool is used. The multipliers use 32-bit values as shown in simulation waveforms.

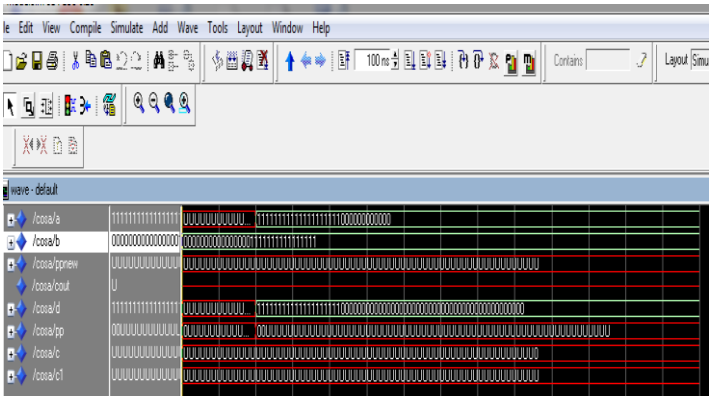

Fig.7 Simulation of COSA based multiplier

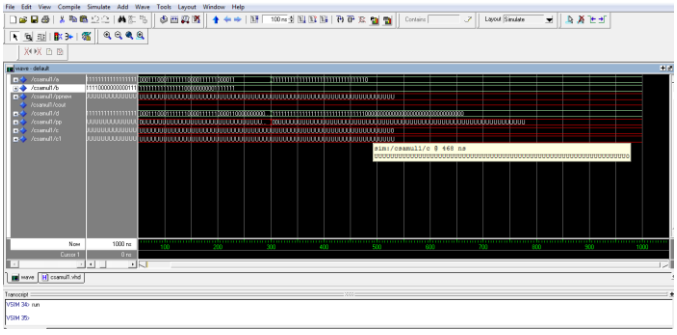

Fig.8 Simulation of CSA based multiplier

VII. CONCLUSION

From the above analysis,the 32-bit unsigned multiplier implemented using COSA and CSA provide great result as compared to the other high speed adder based multipler like ripple cary based multiplier etc.By analysis, the area consumed by carry save adder based multiplier is less than that of conditional sum adder based multiplier.

REFERENNCES

[1] V. Vijayalakshmi , R.Seshadri, Dr.S.Ramakrishnan, "Design and Implementation of 32 Bit Unsigned Multiplier Using CLAA and CSLA",*IEEEProceedings on Circuits,Devices and Systems*,2013

[2] Raminder Preet Pal Singh, Parveen Kumar, Balwinder Singh, "Performance Analysis of 32-Bit Array Multiplier with a Carry Save Adder and with a Carry-Look-Ahead Adder", *International Journal of Recent Trends in Engineering*, Vol 2, No. 6, November 2009

[3] P. S. Mohanty, "Design and Implementation of Faster and Low PowernMultipliers", *Bachelor Thesis*. National Institute of Technology, Rourkela, 2009

[4] Ahmet SERTBAS, R.Selami OZBEY," A performance analysis of classified binary adder architectures and the vhdl simulations",*Journal of electrical & electronics engineering***,**Vol.4,2004'