32 Bit Parallel Multiplier Using VHDL

Vrushali Gaikwad[#1], Rajeshree Brahmankar[#2], Amiruna Warambhe[#3], Yugandhara Kute[#4], Nishant Pandey[*5]

[#]*Student, electronics engineering, G. H. Raisoni Institute of Engineering and Technology for women's, Nagpur (Maharashtra), India*

[*]*Assistant Professor, electronics engineering, G. H. Raisoni Institute of Engineering and Technology for women's, Nagpur (Maharashtra), India*

***Abstract*—In this paper, design of 32-bit parallel multiplier is presented, by introducing Carry Save Adder (CSA) in partial product lines. The multiplier given in this paper is modeled using VHDL (Very High Speed Integration Hardware Description Language) for 32-bit unsigned data. Here comparison is done between Carry Save Adder (CSA) and Carry Look Ahead Adder (CLA). The comparison is done on the basis of two performance parameters i.e. Speed and Power consumption. To design an efficient integrated circuit in terms of power and speed, has become a challenging task VLSI design field**

***Keywords* — Multiplier, Carry Save adder(CSA), Carry Look Ahead adder(CLA), Ripple Carry Adder(RCA), VHDL simulation.**

I. INTRODUCTION

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. Low power consumption and smaller area are some of the most important criteria for the fabrication of DSP systems and high performance systems. 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. In our project we try to determine the best solution to this problem by comparing a few multipliers. And preparing a 32 bit multiplier by using CSA.

Arithmetic operations such as addition, subtraction, multiplication and division are widely used and play an important role in various digital systems such as digital signal processor (DSP) architecture, microprocessor and microcontroller and data process unit [4]. As we are designing multiplier we need multiplication process. The basic principle used for multiplication is to evaluate partial products and accumulation of shifted partial products. In order to perform this operation number of successive addition operation is required. Therefore one of the major components required to design a multiplier is Adder. Adders can be Ripple Carry, Carry Look Ahead, Carry Select, Carry Skip and Carry Save [1,2,3].

Several researchers had addressed the adder performance issues and others did the same with regard to the multiplier performance. Sertbas and Özbey worked on the performance analysis of classified binary adder architectures. They compared the ripple adder, carry-look-ahead adder, carry select adder and the conditional sum adder. They used VHDL implementation for their designs and comparison studies. Their work included the unit-gate models for area and delay [5].

Performance analysis of multipliers is also carried out by number of researchers. Basic architecture of multiplier uses Ripple Carry Adder in the partial product lines. In 2005, Fonseca, M.; da Costa, E. et al presented a design of a Radix 2m hybrid Array multiplier to handle operands in 2's-complement form by using Carry Save Adder in each partial product lines. The results showed that the multiplier architecture with CSA gives better performance in terms of area, speed and power consumption as compared to the architecture with RCA [10].The multiplier presented in this paper are all done by using VHDL for 32-bit unsigned data. XILINX ISE v 9.1i is used as synthesis tool and FPGA-Spartan III (XC3S250E) device is selected to get area and power reports.

II. CARRY SAVE ADDER

Carry Save Adder is used to do the summation of three or more n-bit numbers. It is same as Full Adder. As shown in Fig.1, we are performing sum of two 32-bits binary numbers, so we are using 32 Full Adders in first stage. Carry save unit consists of 32 full adders, each of which performs single sum and carry bit based only on the corresponding bits of the two input numbers. Let A and B are two 32-bit numbers and produces partial sum and carry as S and C as shown in the Table1:

$$Si = Ai \text{ xor } Bi \quad (1)$$
$$Ci = Ai \text{ and } Bi \quad (2)$$

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.

Table 1. Carry save adder computation

A:	1 0 0 1 1
B:	1 1 0 0 1
C: +	0 1 0 1 1
S:	0 0 0 0 1
C: +	1 1 0 1 1
Sum:	1 1 0 1 1 1

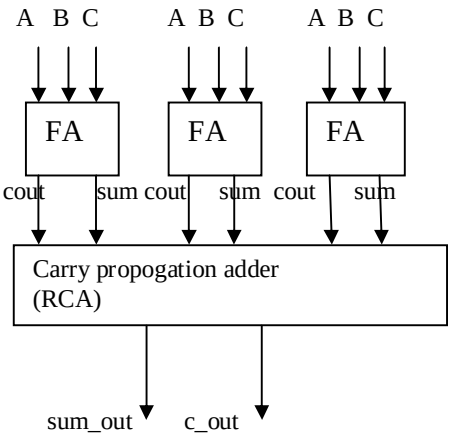


Fig 1: Computation flow of Carry Save Adder.

III. CARRY LOOK AHEAD ADDER

Lookahead carry algorithm is used to perform the addition operation speedily, because in this algorithm carry for the next stages is calculated in advance based on input signals. In CLA, the carry propagation time is reduced to O(log2(Wd)) by using a tree like circuit to compute the carry rapidly. The CLA exploits the fact that the carry generated by a bit-position depends on the three inputs to that position [7]. If 'R' and 'S' are two inputs then if R=V=1, a carry is generated independently of the carry from the previous bit position and if R=V= 0, no carry is generated. Similarly if R ≠V, a carry is generated if and only if the previous bit-position generates a carry. 'C' is initial carry, "S" and "Cout" are output sum and carry respectively, then Boolean expression for calculating next carry and addition is:

Pi = Ri xor Vi -- Carry Propagation (1)
Gi = Ri and Vi -- Carry Generation (2)
Ci+1 = Gi or (Pi and Ci) -- Next Carry (3)
Si = Xi xor Yi xor Ci -- Sum Generation (4)

Thus, for 4-bit adder, we can extend the carry, as shown below:

$C1 = G0 + P0 \cdot C0$ (5)
$C2 = G1 + P1 \cdot C1 = G1 + P1 \cdot G0 + P1 \cdot P0 \cdot C0$ (6)
$C3 = G2 + P2 \cdot G1 + P2 \cdot P1 \cdot G0 + P2 \cdot P1 \cdot P0 \cdot C0$ (7)
$C4 = G3 + P3 \cdot G2 + P3 \cdot P2 \cdot G1 + P3 \cdot P2 \cdot P1 \cdot G0 + P3 \cdot P2 \cdot P1 \cdot P0 \cdot C0$ (8)

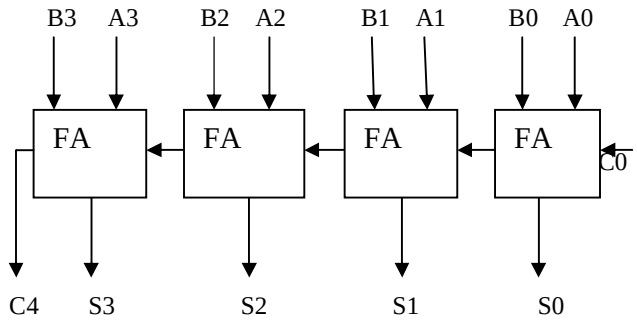


Fig 2: Computation flow of Carry look Ahead Adder.

IV. MULTIPLIER USING CSA

Instead of using other Adders, here we are using Carry save adder for adding each group of partial product terms. Because as compared to CSA some other adders are slow and as CSA is more accessible for understanding, therefore we are using CSA in our project. Figure 2 & 3 shows architecture of 32-bit multiplier using CSA respectively to add each group of partial products in parallel.

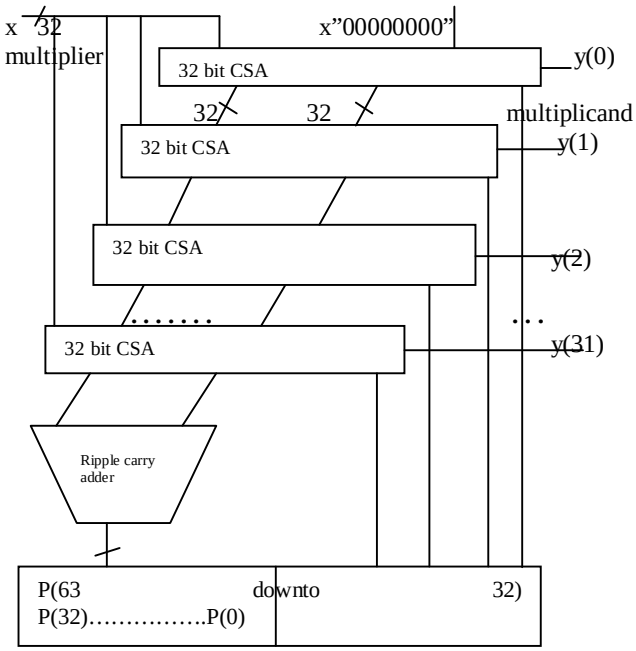


Fig 3: 32-Bit Array Multiplier Using Carry save Logic

V. SIMULATION RESULT

The VHDL simulation of multiplier is shown in this section. For simulation process XILINX 9.1i software is used and hardware used is universal VLSI (FPGA) trainer kit. Here, we have performed the simulation process for half adder, full adder, 4-bit multiplier, 8-bit multiplier and 32-bit multiplier .

Here, simulation process of half adder is shown. In half adder there are two inputs a and b and the outputs as sum and carry. Thus, simulation waveform for half adder is given in fig. 4.

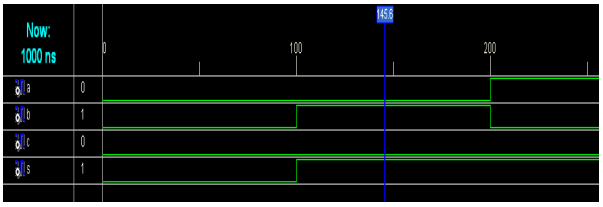


Figure 4: Simulation Waveform for half adder.

Here, simulation process of full adder is shown. In full adder there are three inputs and the outputs as sum and carry. Thus, simulation waveform for full adder is given in fig. 5.

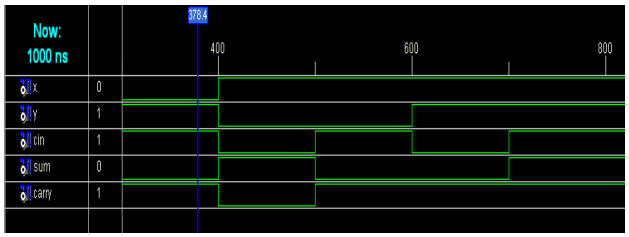


Fig 5: Simulation Waveform for full adder.

Here, simulation process of 4-bit multiplier is performed. Here,there are two 4 bit inputs resulting in 8 bit output. Thus, simulation waveform for 4-bit multiplier is given in fig. 6.

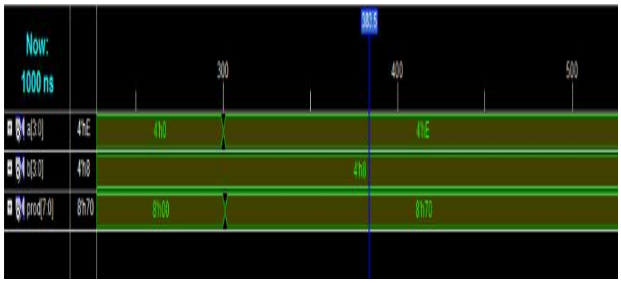


Fig 6: Simulation Waveform for 4bit-multiplier.

Here, simulation process of 8-bit multiplier is shown. Here, there are two 8 bit inputs which results in 16 bit output. Thus, simulation waveform for 8-bit multiplier is given in fig. 7.

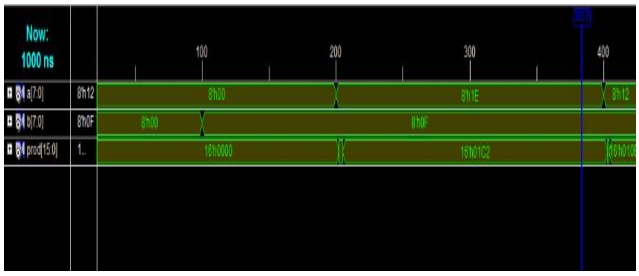


Fig 7: Simulation Waveform for 8-bit multiplier.

Here, simulation process of 32-bit multiplier is performed. Here, there are two 32 bit inputs resulting in 64 bit output. Thus, simulation waveform for 32-bit multiplier is given in fig. 8

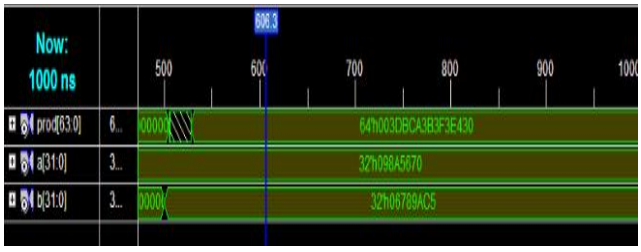


Fig 8: Simulation Waveform for 32-bit multiplier.

VI. OUTPUT ON FPGA KIT

Simulation process of full adder and half adder on FPGA kit is shown below:

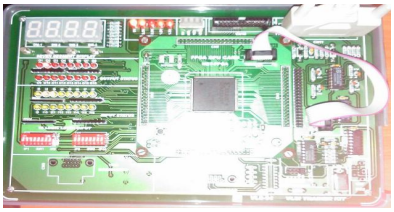
for half adder:

i/p:11 ; o/p:01

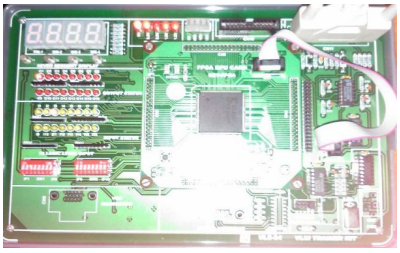
i/p:10 ; o/p:10

Fig 9: Simulation results of half adder.

for full adder:



i/p:011; o/p:01



i/p:001; o/p:10

Fig 10: Simulation results of half adder.

VI. CONCLUSION

This paper presents a highly efficient method of multiplication using VHDL. The multiplication using carry save adder is more faster and efficient than any other adder. According to the results, implementation of CSA logic in each partial product lines improves overall performance of multiplier unit. The delay is reduced and hence, this multiplier is very much useful in large multiplications.This work is

performed on 32-bit unsigned data .Therefore, it can be extended for signed multiplication as well as for designing of 64-bit multiplier.

REFERENCES

[1] B. Parhami, Computer Arithmetic, Algorithm and Hardware Design, Oxford University Press, New York, pp. 91-119, 2000.

[2] Stephen Brown and Zvonko Vranesic, Fundamentals of Digital Logic with VHDL Design.2nd Edn. McGraw-Hill Higher Education, USA.ISBN: 0072499389, 2005.

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

[4] Rajender Kumar, Sandeep Dahiya, "Performance Analysis of Different Bit Carry Look Ahead Adder Using VHDL Environment, International Journal of Engineering Science and Innovative Technology (IJESIT)", Volume 2, Issue 4, July 2013. ISSN: 2319-5967.

[5] 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.

[6}http://www.istanbul.edu.tr/eng/ee/jeee/main/pa ges/issues/is41/41005.pdf

[7] 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.

[8] Hasan Krad and Aws Yousif, "Design and Implementation of a Fast Unsigned 32-bit Multiplier Using VHDL"

[9] Hasan Krad and Aws Yousif Al-Taie, "Performance Analysis of a 32-Bit Multiplier with a Carry-Look-Ahead Adder and a 32-bit Multiplier with a Ripple Adder using VHDL", Journal of Computer Science 4 (4): 305-308, 2008

[10] Fonseca, M.; da Costa, E. et al, "Design of a Radix-2m Hybrid Array Multiplier Using Carry Save Adder" integrated Circuits and System Design, 18th Symposium on Volume, Issue, 4-7 Sept. 2005 Page(s): 172-177.