

# Multiplier Design Using Carry save Adder

Priyanka.S<sup>1</sup>, Ramya.R<sup>2</sup>, Ramya.S<sup>3</sup>, Roobini Priya.B<sup>4</sup> and Nithya priya.A.P<sup>5</sup>UG Scholar, Department of ECE, Adhiyamaan College of Engineering, Hosur, TN, India<sup>1,2,3,4</sup>Associate Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, TN, India<sup>5</sup>

**ABSTRACT:** Our project is modeled using VHDL Language by using 2, 3, 4 bit signed data. This project sets up the design of 2, 3, 4 bit multiplier which is high speed, has power consumption, delay is less. The speed of the multiplier is increased by reducing the generated partial products. Wallace tree is one of the method to reduce the partial products which are generated. Therefore, minimizing the number of full adders used in the multiplier will reduce the circuit complexity.

**KEYWORDS:** Carry Save Adder [CSA], Wallace Tree, Xilinx ISE 12.3.

## I. INTRODUCTION

Multipliers are the very important part of modern computing systems. Multipliers are mainly used in electronic applications like Digital Signal Processor (DSP), VLSI designs. In computational systems the most important operation is multiplication. The multiplier plays a crucial role in the field of graphics and process control. Depending upon the output requirements of the applications the multiplier structure will vary. The basic principle used for multiplication is to evaluate partial products and accumulation of shifted partial products. Successive addition operations are required in order to perform these operations. Hence one of the major components required to design a multiplier is adder.

## II. CARRY SAVE ADDER

Carry save adder is mainly used in the addition of three or more n-bit numbers. CSA is similar to the full adder. Instead of using any other adder here we use CSA for the addition of the partial product terms of each group. Compared with other adders CSA is fast and it is more easy to understand. We need to add more than two numbers together in many cases, the easiest way of adding together m numbers is by adding the first two, wherein both the numbers are n bits wide. Then their sum is added to the next one and so on.

Consider the sum:

$$\begin{array}{r} 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8 \\ +\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 0 \\ \hline =1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \end{array}$$

Using basic arithmetic, we calculate right to left, "8+2=0, carry 1", "7+2+1=0, carry 1", "6+3+1=0, carry 1", and so on to the end of the sum. Although we know the last digit of the result at once, we cannot know the first digit until we have gone through every digit in the calculation, passing the carry from each digit to the one on its left. Thus adding two n-digit numbers has to take a time proportional to n, even if the machinery we are using would otherwise be capable of performing many calculations simultaneously.

In electronic terms, using bits (binary digits), this means that even if we have n one-bit adders at our disposal, we still have to allow a time proportional to n-digits to allow a possible carry to propagate from one end of the number to the other.

Until this process is done,

1. We do not know the result of the addition.
2. We do not know whether the result of the addition is larger or smaller than a given number (for instance, we do not know whether it is positive or negative).

A carry look-ahead adder can reduce the delay. In this principle the delay can be reduced so that it is proportional to  $\log(n)$ , but for large numbers this is no longer the case, because even when carry look-ahead is implemented, the distance that signals have to travel on the chip increase in proportion to  $n$ , and propagation delays increase at the same rate. Once we get to the 512-bit to 2048-bit number sizes that are required in public-key cryptography, carry look-ahead is not of much help.

### V. BLOCK DIAGRAM

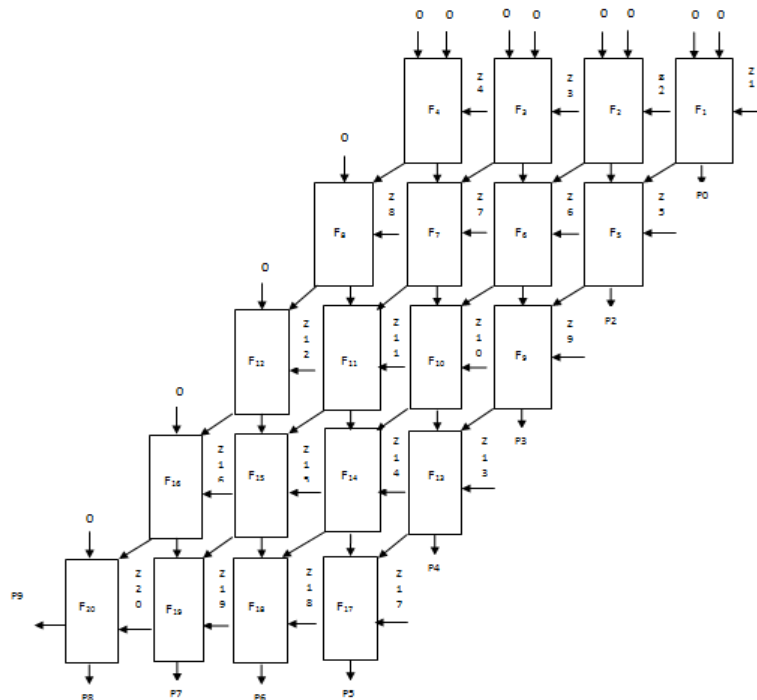


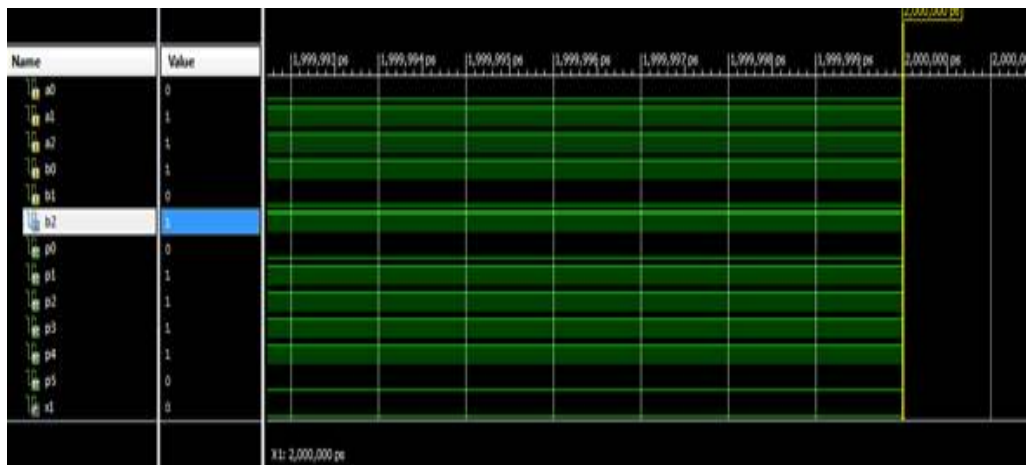
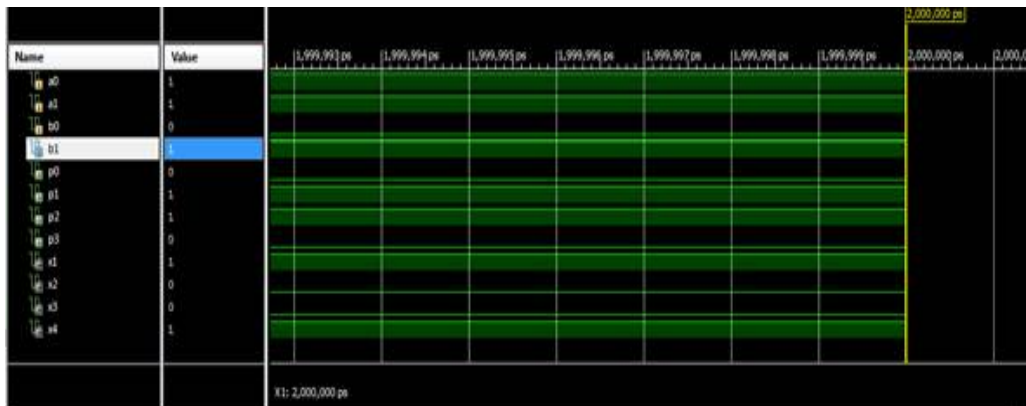
Fig 1: Block Diagram

### VI. EXPLANATION

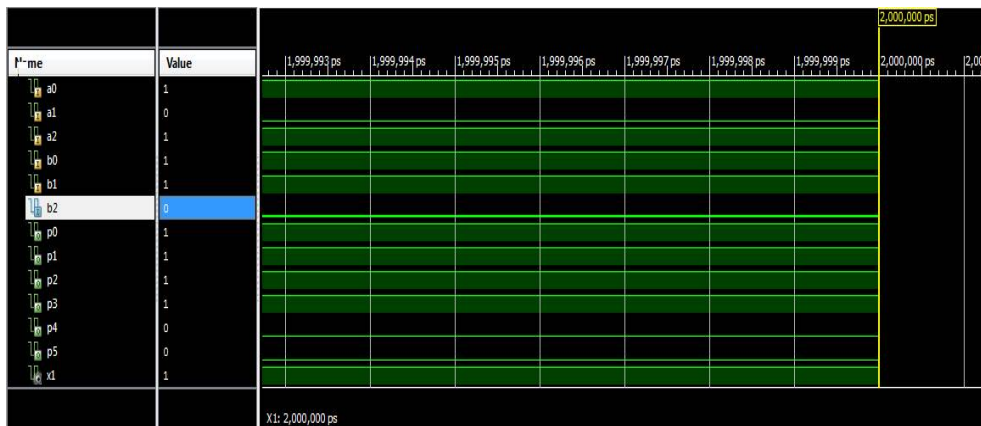
The block diagram consists of “n” number of full adders. It has three inputs like traditional full adder. As traditional adders, when three bits are added sum and carry are generated. The carry generated by first full adder is given to the next full adder as input which is in series, whereas in carry save adder the carry is given to the next set of full adders as shown in the fig 1. So the carry delay is very less when compared to traditional multiplier.

**VII. OUTPUT ANALYSIS**

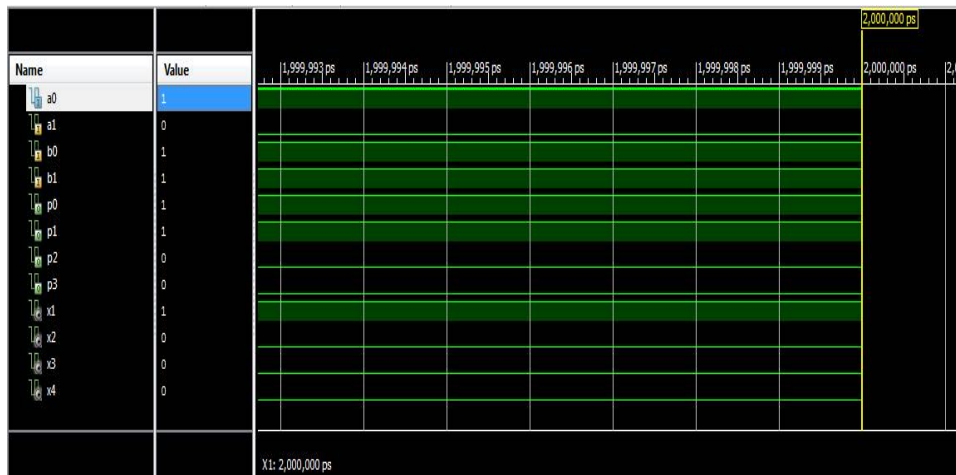
**4-BIT MULTIPLIER**



**3-BIT MULTIPLIER**



**2-BIT MULTIPLIER**



**VIII. CONCLUSION**

Multiplier design is highly efficient and low power consumption. The multiplication operation using carry save adder is faster and efficient than any other adder. From the results, we see how CSA logic is implemented in each partial product line which improves the overall performance of the multiplier unit. This multiplier is very much helpful in larger multiplications as the delay has been reduced. This design is implemented on unsigned data. This design can also be extended over signed multiplication as well as it can be used for designing of higher bit multiplier.

**REFERENCES**

- [1] Mariya Stephen, "Design for Low Power Multiplier Based on Fixed Width Replica Redundancy Block & Compressor Trees", IJSR, Volume 4 Issue 10, October 2015.
- [2] Vrushali gaikwad, Rajeshree Brahmanekar, Amiruna Warambhe, Yugandhara Kute, Nishant Pandey, "32 Bit Parallel Multiplier Using VHDL", IJETT, Volume 9 Number 3, March 2014.
- [3] 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", IJRTE, Volume 2 Number 6, Nov 2009.
- [4] Prof. Loh, "Carry Save Addition", CS3220 – Processor Design, Feb 2005.
- [5] Stephen Brown and Vranesic, "Fundamentals of Digital Logic with VHDL Design", 2nd Edn. McGraw-Hill Higher Education, USA. ISBN: 0072499389, 2005.
- [6] B. Parhami, "Computer Arithmetic, Algorithm and Hardware Design", Oxford University Press, New York, pp. 91-119, 2000.
- [7] Wakerly, J.F. "Digital Design-Principles and Practices", 4th Edn, Pearson Prentice Hall, USA. ISBN: 0131733494.