International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 4, Issue 1, pp: (169-173), Month: January - March 2016, Available at: www.researchpublish.com

DESIGN OF BINARY MULTIPLIER USING ADDERS

Sudhir Bussa[1], Ajaykumar Rao[2], Aayush Rastogi[3]

[1]Assist. Prof Electronics and Telecommunication Department, Bharatividyapeeth Deemed University College of Engineering, Pune
[2,3]Electronics and Telecommunication Department, Bharatividyapeeth Deemed University College of Engineering, Pune.

Abstract: **The most important feature of any electronic device like a mobile phone is its processor and the speed of operation. Every person who uses electronic devices like laptops, mobile phones wants the work to be done in a split of second. This can only happen if the device has a good and fast processor. The basic building block of a processor is a multiplier. A multiplier is made up of many adders. It is the speed of multiplication that we are concerned about. The processing speed of a multiplier can be improved by using various adders. This paper showcases various types of adders and also how a multiplier can be made using adders.**

Keywords: **Multiplier, adder, Xilinx, simulation.**

1. 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. 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. Since ultimately, speed, power and chip area are the most often used measures of the efficiency of an algorithm, there is a strong link between the algorithms and technology used for its implementation. Our research aims at improving the speed of the binary multiplication by using various adders like carry look ahead adder and carry save adder.

The objective of our project is to use various types of adders and compare the results in terms of delay and the power consumed in the multiplier. We are aiming at using different algorithms for multiplier such as Booths algorithm, Vedic multiplier, and so on. In order to build a multiplier which consumes less power and delivers results faster, we will be using ripple carry adder, carry look ahead adder, and carry save adder.

2. BASIC BINARY MULTIPLICATION

A binary computer multiplies the binary numbers in the same manner as decimal multiplication is done. In binary encoding each long number is multiplied by one digit (either 0 or 1), and that is much easier than in decimal, as the product by 0 or 1 is just 0 or the same number. Therefore, the multiplication of two binary numbers comes down to calculating partial products (which are 0 or the first number), shifting them left, and then adding them together (a binary addition, of course).

```
       1011   (this is 11 in decimal)
     X 1110   (this is 14 in decimal)
     ======
       0000   (this is 1011 x 0)
      1011    (this is 1011 x 1, shifted one position to the left)
     1011     (this is 1011 x 1, shifted two positions to the left)
  + 1011      (this is 1011 x 1, shifted three positions to the left)
  =========
   10011010   (this is 154 in decimal)
```

Figure 1:- Basic binary multiplication.

This is much simpler than in the decimal system, as there is no table of multiplication to remember: just shifts and ads. In recent years, power consumption, as well as area and speed, are the most important issues in VLSI design. Specifically, the design of multipliers is critical in digital signal processing applications, where a high number of multiplications are required. Multipliers require high amount of power and delay during the partial products addition. At this stage, most of the multipliers are designed with different kind of adders that are capable to add two/three or at most 4 bits.

3. DIFFERENT TYPES OF ADDERS

3.1 RIPPLE CARRY ADDER:

A ripple carry adder is a logic circuit in which the carry-out of each full adder is the carry in of the succeeding next most significant full adder. It is called a ripple carry adder because each carry bit gets rippled into the next stage. In a ripple carry adder the sum and carry out bits of any half adder stage is not valid until the carry in of that stage occurs.

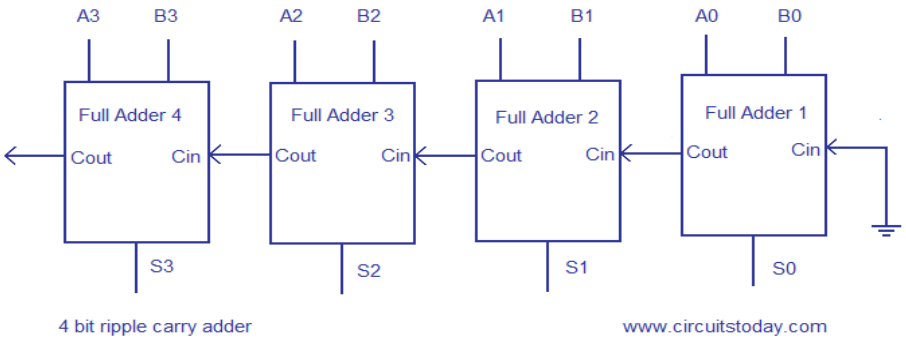

Figure 2:-Ripple carry adder

Sum out S0 and carry out Cout of the Full Adder 1 is valid only after the propagation delay of Full Adder 1. In the same way, Sum out S3 of the Full Adder 4 is valid only after the joint propagation delays of Full Adder 1 to Full Adder 4. In simple words, the final result of the ripple carry adder is valid only after the joint propagation delays of all full adder circuits inside it.

3.2 CARRY LOOK AHEAD ADDER:

A carry-look ahead adder (CLA) is a type of adder used in digital logic. A carry-look ahead adder improves speed by reducing the amount of time required to determine carry bits. It can be contrasted with the simpler, but usually slower, ripple carry adder for which the carry bit is calculated alongside the sum bit, and each bit must wait until the previous carry has been calculated to begin calculating its own result and carry bits. The carry-look ahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits.

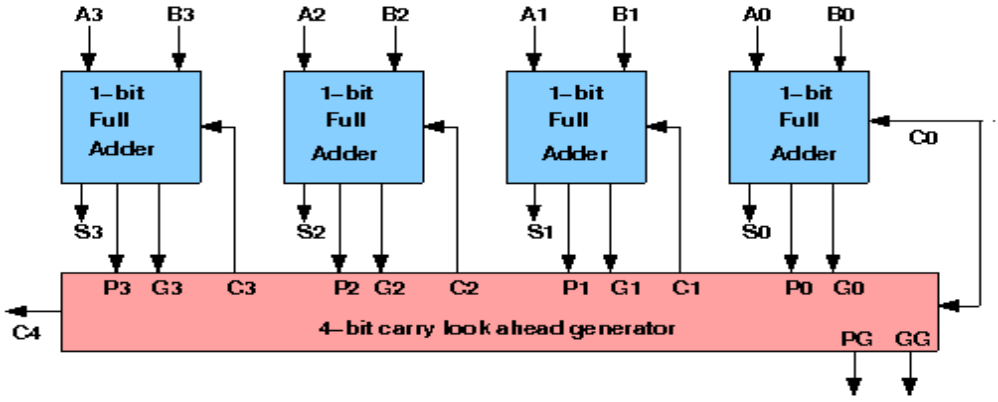

Figure 3:-Carry look ahead adder

Carry look ahead logic uses the concepts of generating and propagating carries.

G (A,B)= A.B

Ci+1= Gi+ (Pi.Ci)

3.3 CARRY SAVE ADDER:

A carry-save adder is a type of digital adder, used in computer micro-architecture to compute the sum of three or more *n*-bit numbers in binary. It differs from other digital adders in that it outputs two numbers of the same dimensions as the inputs, one which is a sequence of partial sum bits and another which is a sequence of carry bits.

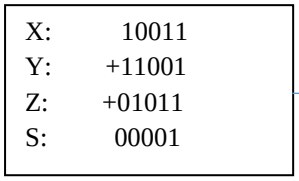

Figure 4:- Computation of only sum

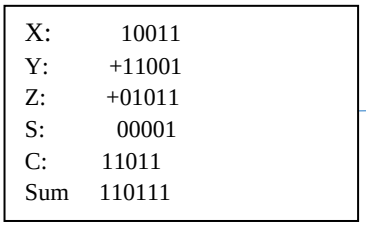

Figure 5:- Computation of only carry

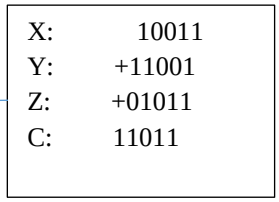

Figure 6:- Computation of sum and carry

The first is to compute the sum ignoring any carries as shown in figure 4. Each si is equal to the sum of xi + yi + zi . Now, separately, we can compute the carry on a column by column basis. In this case, each ci is the sum of the bits from the previous column divided by 10 (ignoring any remainder). Another way to look at it is that any carry over from one column gets put into the next column. Now, we can add together c and s, and we'll verify that it indeed is equal to x + y + z.

3.4 COMPARISON OF THE ABOVE ADDERS:

We have taken the following results of adders from the research paper Design and performance analysis of various adders using VERILOG.

Table 1:- Performance analysis of various adders

S.NO	Design	Area (LUTs)	Delay (ns)
1	Ripple carry adder	8	2.191
2	Carry look ahead adder	10	2.266
3	Carry save adder	13	1.433

4. A 4X4 MULTIPLIER USING CARRY LOOK ADDER

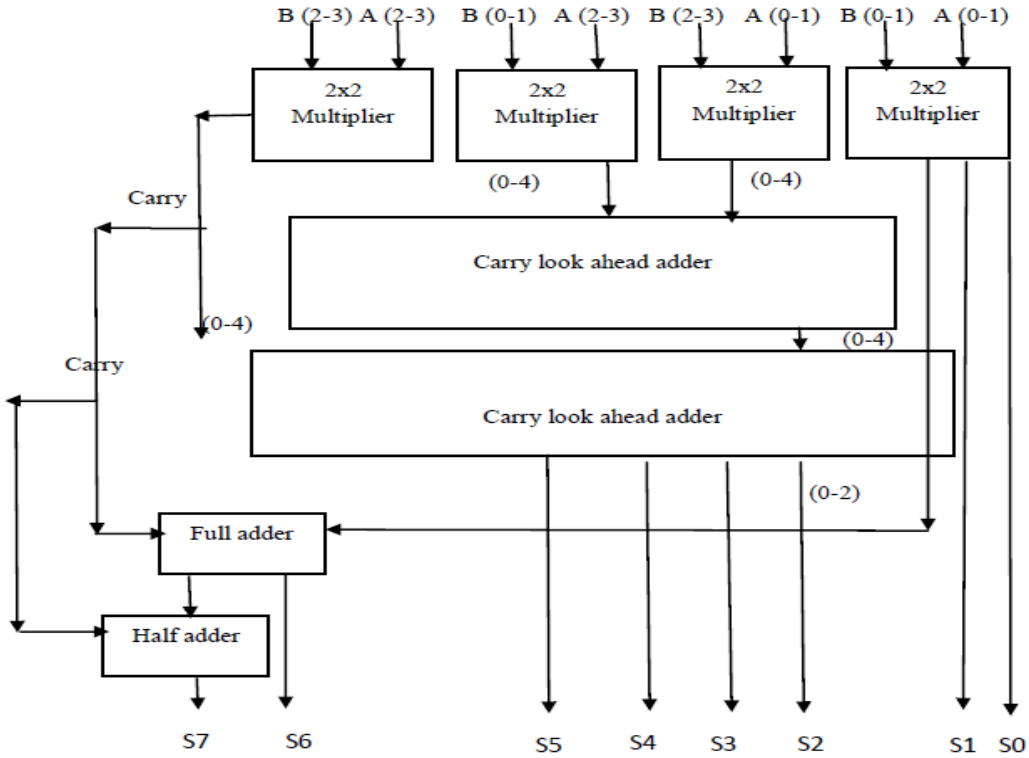

Figure 7:- Block diagram of 4x4 multiplier using carry look ahead adder

We have used four 2x2 multipliers and two carry look ahead adder and one full adder and a half adder. The purpose of using carry look ahead adder is to reduce the delay in the generation of carry bits. We have written the code in VHDL format in Xilinx software and have also obtained simulation results.

4.1 Mapping of 4x4 multiplier:

$A_3A_2A_1A_0$ X $B_3B_2B_1B_0$

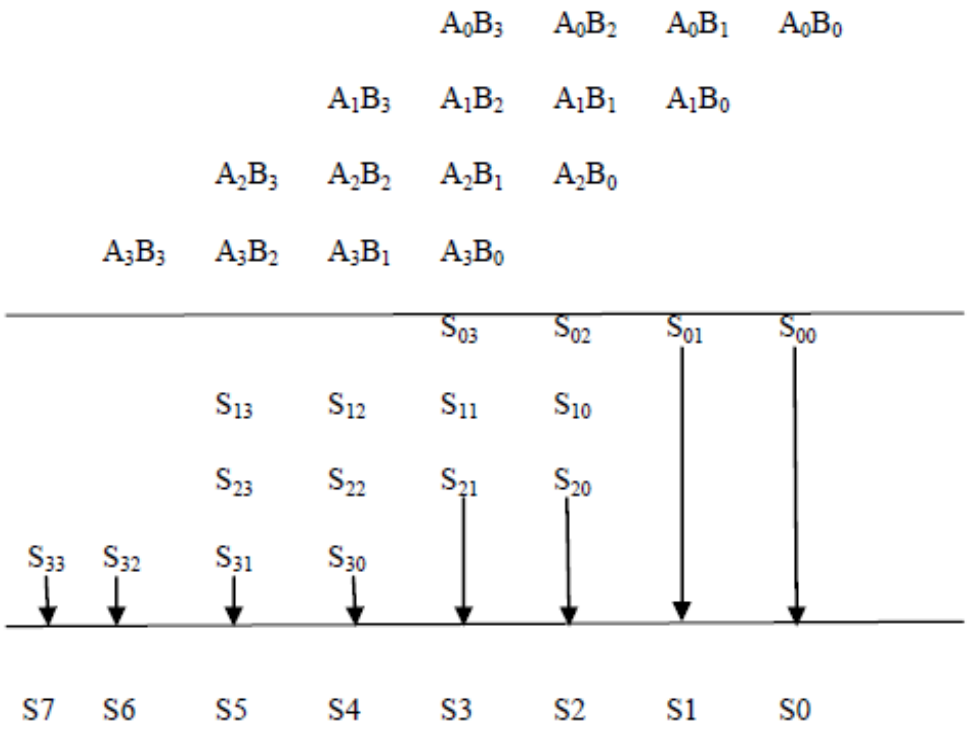

Figure 8:- Mapping for 4x4 multiplier

4.2 Simulation results:

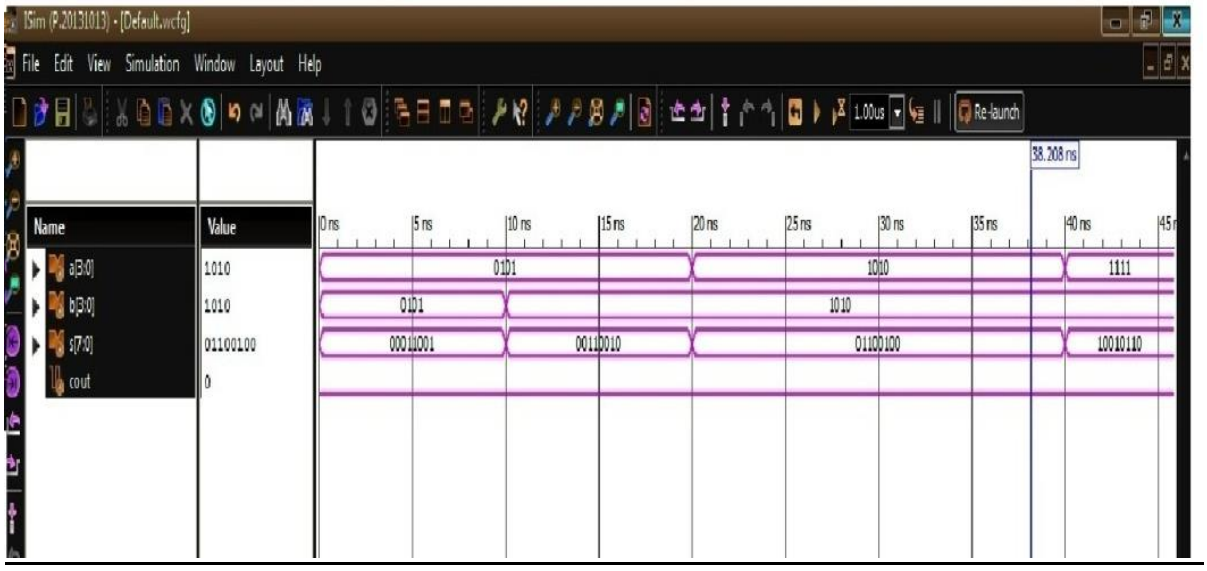

Figure 9:- Simulation result of 4x4 multiplier in binary

Figure 10:- Simulation result of 4x4 multiplier in decimal

5. COCLUSION AND FUTURE SCOPE

Thus we have successfully made a 4x4 multiplier using Xilinx platform. As mentioned above, the main purpose of using a carry look ahead adder was to reduce the time to calculate the carry bits. We have used a mapping technique to calculate the partial products of the multiplication and then giving these partial products to the carry look ahead adders we have obtained the final product. In future we will implement carry save adder for making a multiplier and then compare the two. Also, we look forward to increase the range of multiplier like 8x8, 16x16 and so on.

REFERENCES

[1] Maroju SaiKumar, Dr. P. Samundiswary, "Design and Performance Analysis of Various Adders using Verilog", IJCSMC, Vol. 2, Issue. 9, September 2013, pg.128 – 138.

[2] Prof. Loh CS3220 - Processor Design - Spring 2005, " Carry save adition".

[3] http://www.circuitstoday.com/ripple-carry-adder.

[4] http://cse10-iitkgp.virtual-labs.ac.in/cla_design.html.