International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 02 | Feb 2020 www.irjet.net p-ISSN: 2395-0072

Implementation of Low Area and Less Delay of Various Multipliers using Verilog

K. Srilatha[1], B. Reeshma[2], K. Pratyusha[3], D. Padmaja[4]

[1]Assistant Professor[1], Department of Electronics and Communication Engineering, Bapatla Women's Engineering College, Bapatla

[2,3,4]U.G Scholars, Department of Electronics and Communication Engineering, Bapatla Women's Engineering College, Bapatla

---

Abstract -*Multiplier plays an important role in today's digital image processing and various other applications. To improve the performance of multipliers, there are mainly three accepts. They are Delay, Area, and Power. To improve the performance of multipliers is to decrease the area and delay. Here, we are increasing the performance in terms of area and delay. In this paper, we explain three multipliers. One of the three multipliers is the Array multiplier, it is the simplest method and high performance, but it suffers from high propagation delay because of the large number of partial products. Wallace tree multiplier technique is used to overcome the problem of Array multiplier. It has less delay and high performance because it decreases the number of partial products compare to the Array multiplier but it requires a large area. Dadda tree multiplier is the fastest multiplier and is used to overcome the problem of Wallace tree multiplier, its performance is also high, and it reduces the area partial products in early stages.*

Key words: Array multiplier, Wallace tree multiplier, Dadda tree multiplier, Delay, Area, Verilog, Xilinx-14-7 Version.

1. INTRODUCTION

Most of the digital circuits and digital signal processing systems are depends on the execution of the multipliers and they are important to increase the speed of any digital systems. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of digital signal processing.

A basic multiplier can be divided into three parts: I. Partial product generation ii. Partial product addition. iii. Final addition.

Here we are using Verilog to build logic. We are using Verilog to design the code very easy and effective way. To design the Multipliers we are using Gates, Half Adders and Full Adders. Gates like XOR, AND and OR Gates are using to design in Half Adder and Full Adder.

For Example, A and B are the Multiplicand and Multiplier respectively. In every Multiplication, we Perform multiplication for each bit with shifting and adding operations.

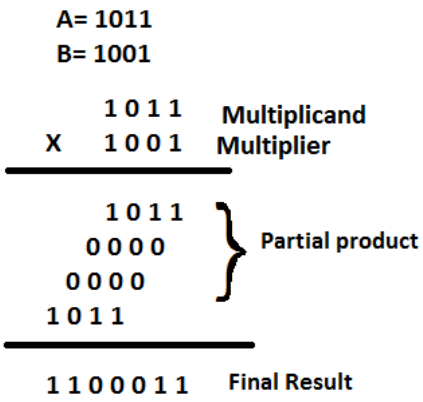


Fig-1: Normal Multiplication

Here A is the Multiplicand and B is the Multiplier.

Multiplication is mostly used in the performance of some instructions in a fast manner or in an effective way to decrease the delay time.

2. ARRAY MULTIPLIER

Array Multipliers is well known due to its regular structure. These multipliers are fully based on adding and shifting operations. In every multiplier, partial product is the main thing that arranged in a specific manner by using adder and shifter.

Partial product is nothing but multiplying multiplier to the each of the multiplicands. The number to be multiplied is the "Multiplicand", and the number by which it is multiplied is the "multiplier". Usually, the multiplier is placed first and the multiplicand is placed second, however, the first factor is the multiplicand and the second is the multiplier.

In Array Multiplication, we are using both half adders and full adders to reduce the delay in both 4-Bit and 8-Bit Multiplication.

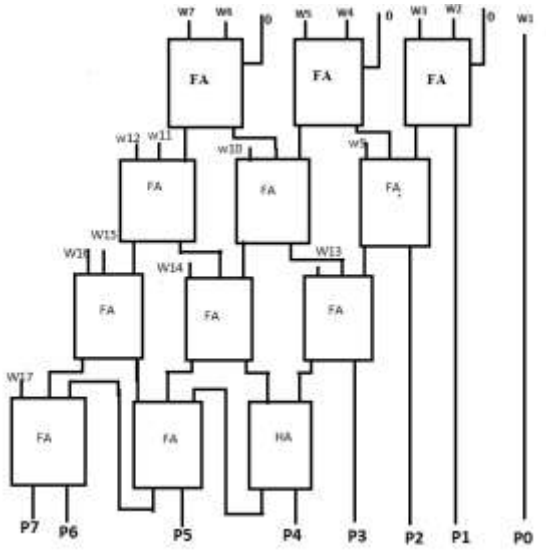


Fig-2: 4-Bit Array Multiplier

If we are using full of Full Adders in the 4-Bit multiplication the delay is 2.63nsec.

Compared to 4-bit multipliers, in 8-bit multipliers number of Gates, Half adders and full adders are more. The structure of the 8-bit partial product is depicting below.

Fig-3: Partial Product of Multiplication for any 8-Bit Multiplications

Here w1, w2, w64 are the Partial Products of the Multiplication. We are the multiplication (AND Gate) of each multiplier with each multiplicand, here I am from one to 64.

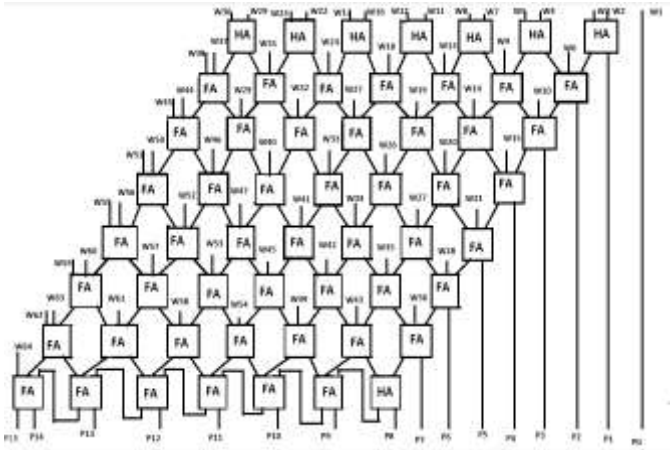


Fig-4: 8-Bit Array Multiplier

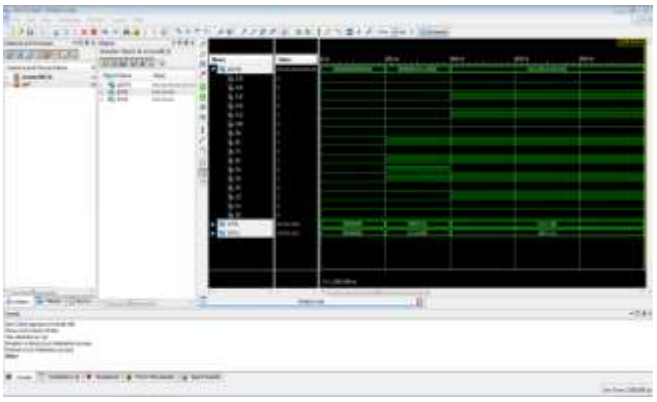
Here w1, w2, soon are the Partial Product of the multiplication.p0, p2, p3, p15are the Final Result of the multiplier.

For all full adders, the delay of the 8-bit multiplier is 10.786nsec. For both full adders and half adder's circuits, we have a delay of 8.760nsec.

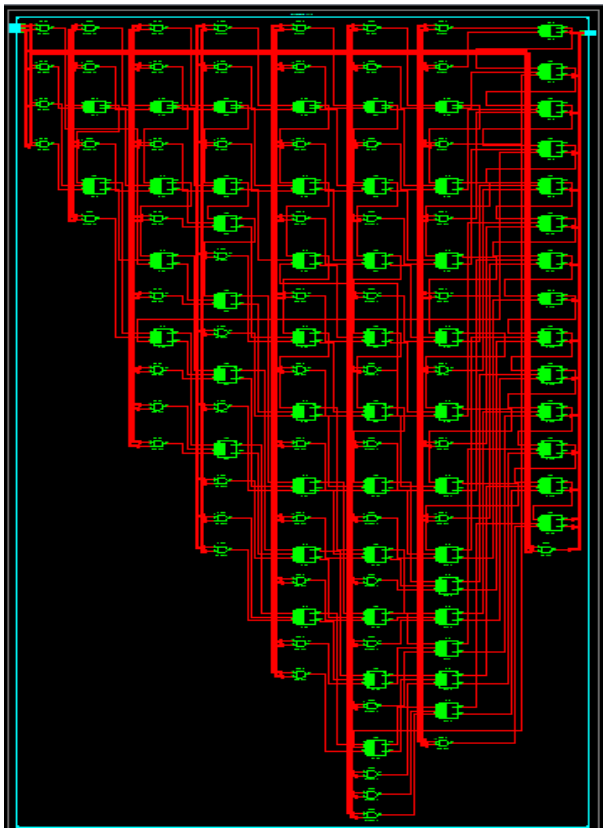
The number of LUTs present in the Array Multiplier is 101.Based on LUTs number the area of the multiplier is more.

Simulation Results for Array Multiplier:

Wave Forms:



RTL Schematic:



Array Multipliers having more delay for 8-Bit multiplication when compared to the 4-Bit multipliers .so we move on to the Wallace tree Multipliers.

3. WALLACE TREE MULTIPLIER

Chris Wallace, an Australian computer Scientist in 1964, proposed the Wallace Multiplier.

To improve the performance that is to decrease the delay time compared to the Array Multiplier we are using row reduction technique. Wallace tree multiplier is a fast multiplier.

To compress the number of rows in a partial product we are using half adders and full adders. There are mainly three stages in a Wallace multiplier. There is the formation of partial products using logic gates, reducing the number of partial products by using half adders and Full adders and finally merging two rows partial products with carrying propagate adder.

FLOW CHART:

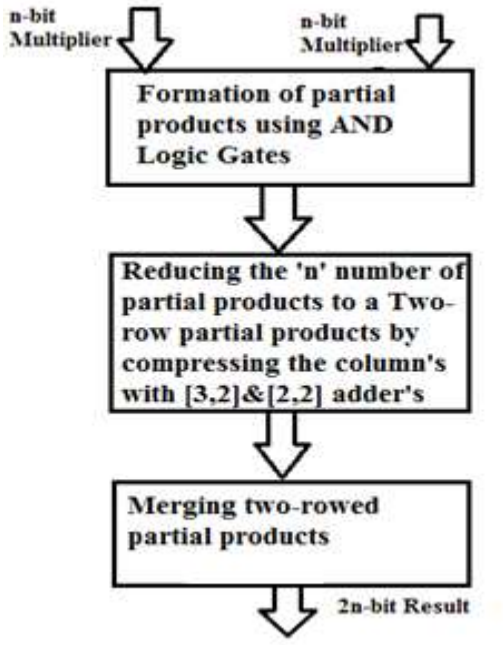


Fig-5: Illustration of three stages in Wallace Multiplier

Illustrating the Wallace Tree Multipliers:

The Wallace tree multiplier has three steps:

1. Multiply (that is-AND) each bit of one of the arguments, by each bit of the other, yielding n^2 results. Depending on the position of the multiplied bits, the wires carry different weights.

2. Reduce the number of partial products to two by layers of full and half adders.

3. Group the wires in two numbers, and add them with a conventional adder.

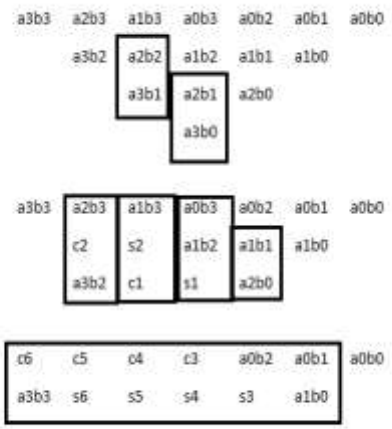


Fig-6: Structure of 4X4 multiplication

The delay of the 4-Bit multiplier is 2.25nsec.

The Structure of the 8-bit Wallace tree Multiplier is also followed the three stages of the Wallace tree multiplier(Fig-5).

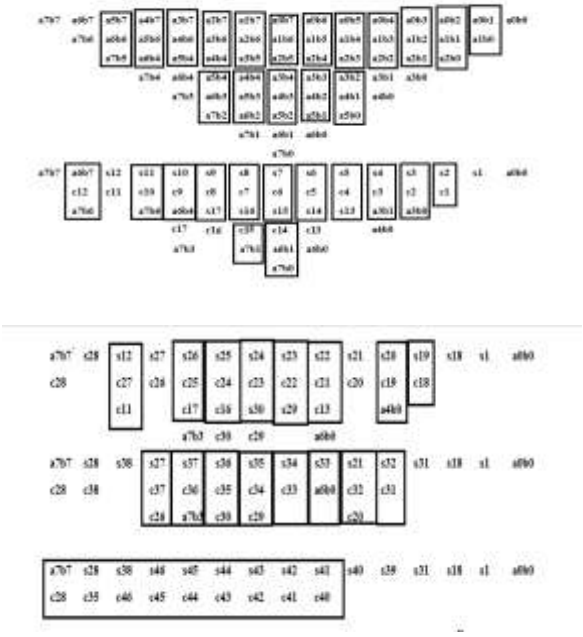


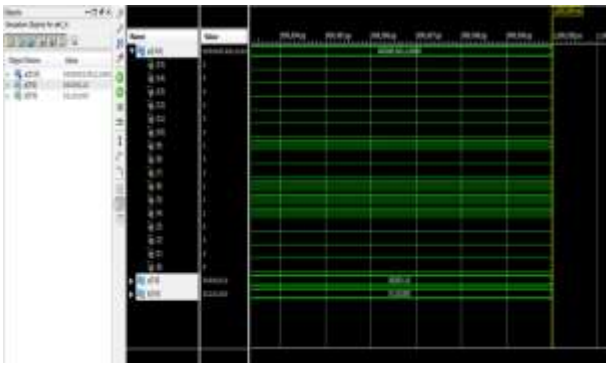
Fig-7: Structure of 8-bit Multiplication

By using this logic, we get the delay as 2.97nsec. Compared to Array multiplier delay is reduced to more than 60%. The number of LUTs present in Wallace is 80. So, the area also reduces when compared to Array.

To reduce the delay of Wallace tree multiplier we go with Dadda tree multiplier.

Simulation Results:

WaveForms:



RTL Schematic:

4. DADDA TREE MULTIPLIER

The Dadda Multiplier is a hardware multiplier design invented by computer scientist Luigi Dadda in 1965. Dadda Multiplier is in the tree-like structure, which is similar to that of the Wallace Tree Multiplier.

Dadda Tree Multiplier is also having three major parts.i.Partial Product Generation. ii. Partial Product Addition. iii. Final Addition.

The Major Part changed in these Multipliers is Partial Product. Here the shape of the Multiplier is tree Shape.

Row compression Multipliers have more popular because of their high computational speed. Wallace and Dadda both multipliers are having column compression technique. Compared to Wallace Tree multipliers data multipliers have less delay.

The reduction is achieved by reducing the number of rows by using counters that are [3:2] counters (Full Adders) and [2:2] counters (Half Adders). Here we perform this operation in three stages.

The Three stages having Formation of partial products using AND logic Gates, reducing the number of partial products by using Half adders and full adders and Merging two rows Partial products with carrying propagation adder.

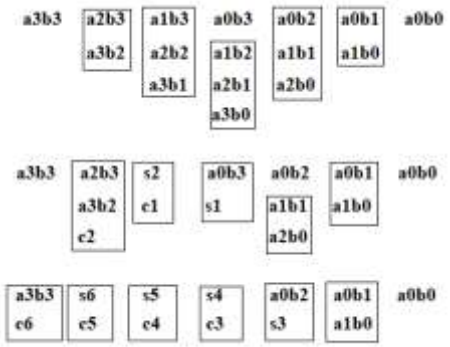


Fig-8: Structure of 4-bit multiplier.

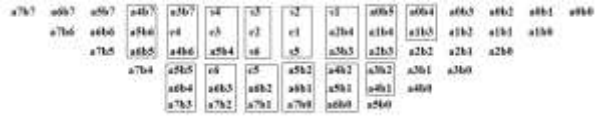
In dadda multiplier, the way we perform partial products is different from the Wallace tree multiplier.so, the delay is reduced in Dadda tree multiplication. The delay of the multiplier is 1.30nsec.

Illustrating the Dadda 8x8 Multiplier:

1. In Dadda Multiplier we are arranging the Partial Product into the Tree-like structure.

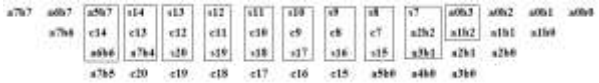
2. Here 'a' and 'b' is the multiplicand and Multiplier. We are reducing these rows by using 3 [3, 2] counters and 3 [2, 2] counters

3. We reduce to six rows from step one.



Here s1,s2,... and c1,c2,.... are the results of [3,2]counter and [2,2] counter that is half adders and full adders respectively.

4. Here we compress the rows to four rows by using half adders and full adders.



5. Form the fourth step we reduce the row to three by using nine [3:2] counters and one [two, 2] counter.

6. This is the result of the two-row partial products.

After this, we are reducing step 6 by using carry propagation.

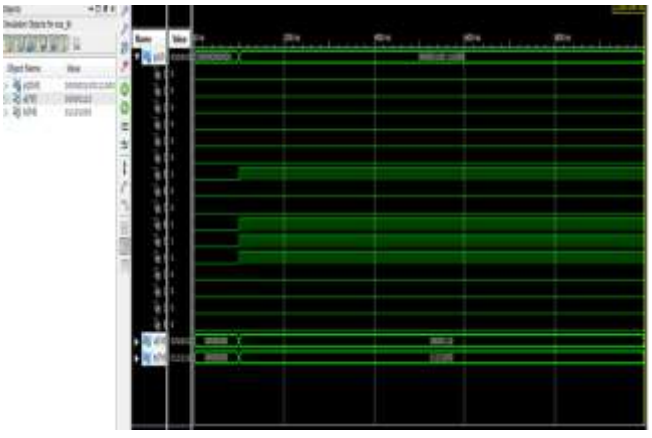
By applying this logic, we get a 2.70nsec delay.

In dadda, tree multiplier the number of LUTs is 77. When compared to both Array and Wallace Multipliers the delay of the Dadda tree multiplier is less in terms of Area.

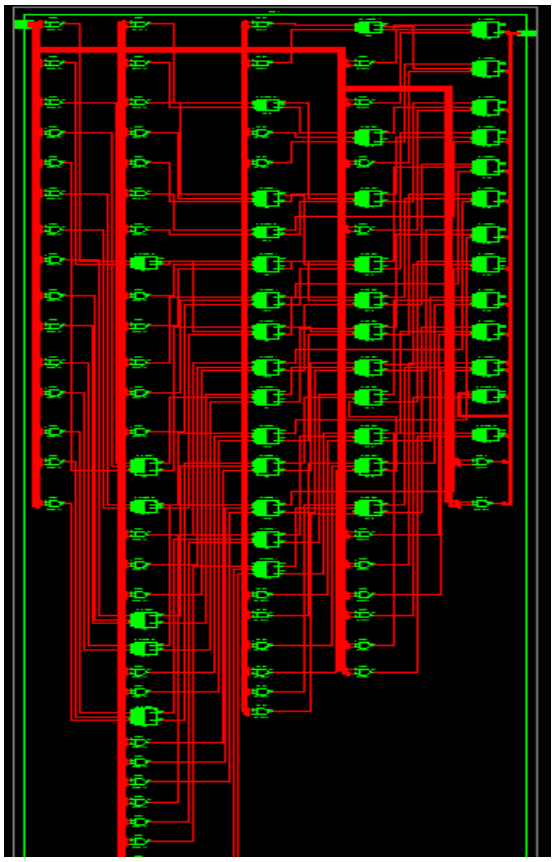
Dadda tree Multiplier is more effective in terms of Delay and Area.

Simulation Results:

Wave Forms:



RTL Schematic:



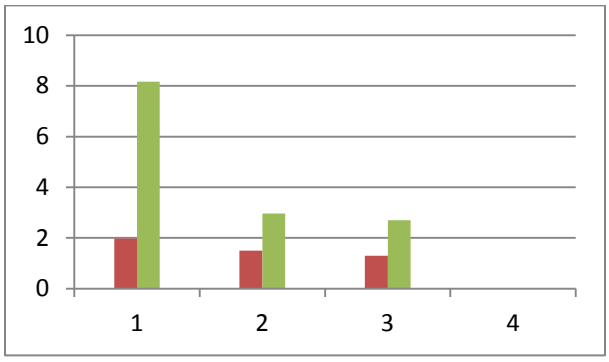


Fig: Delay comparison

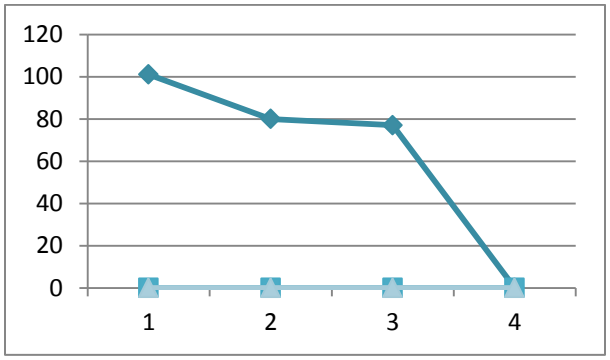


Fig: LUTs variations

CONCLUSION:

In this paper, three multipliers discussed above are compare based on there delay and area. Array is a simple method and it is very complex. Wallace tree multiplier is less complex and high area. Therefore, to improve the performance of multipliers dadda tree multiplier is more effective. This Paper proves that the Dadda tree multiplier is the faster multiplier than array and Wallace tree multipliers.

References:

[1].T.Sharma prof.B.P.sing, K.G.Sharma, N.Arora, "High-speed array multipliers based on 1-bit full adders ", in Int.J.of recent trends in Engineering and technologies vol.4, no.4, pp.26-28, November 2010.

[2].”Dadda Lugi” (May 1965) "Some schemes for parallel multipliers". Alta frequenza.

[3]. Townsend Whitney J.Swartzlander, Jr.Earl E.Abraham, Jacob A.(Dec 2003).”A Comparison of Dadda and Wallace Multiplier Delays”.

[4]. J.-Y.Kang, J.-L.Gaudiot, “A fast and well-structured multiplier”, in proceedings of the EUROMICRO systems on digital system design (DSI’04), pp.508-515, September 2004.

[5].C.R.Daugh and D.A.wooley, "A two's complement parallel array multiplication algorithm," IEEE Transactions on computers, vol.C-22, no.12, pp.1045-1047, 1973.

[6]. C.W.Wallace, "Suggestion for the fast multiplier," IEEE transactions of electronic computers, vol.EC-13, no.1, pp. 14-17, 1964.

BIOGRAPHYIES:

K. SRILATHA [M.TECH] is presently working as an Asst. Professor in the department of ELECTRONICS AND COMMUNICATION ENGINEERING in BAPATLA WOMEN'S ENGINEERING COLLEGE ,BAPATLA.

B.REESHMA pursing B.TECH in ELECTRONICS AND COMMUNICATION ENGINEERING at BAPATLA WOMEN'S ENGINEERING COLLEGE, BAPATLA.

K.PRATYUSHA pursing B.TECH in ELECTRONICS AND COMMUNICATION ENGINEERING at BAPATLA WOMEN'S ENGINEERING COLLEGE, BAPATLA

D.PADMAJA pursing B.TECH in ELECTRONICS AND COMMUNICATION ENGINEERING at BAPATLA WOMEN'S ENGINEERING COLLEGE, BAPATLA.