IMPLEMENTATION OF 16- BIT BINARY MULTIPLIER BY USING FULL ADDERS

Project Report

Submitted in partial fulfillment of the requirements for the award of

BACHELOR OF SCIENCE

In

ELECTRICAL AND COMPUTER ENGINEERING

Submitted by

Joshua Ball (R-11330455) & Dipendra Yadav (R-11538518)

Under the guidance of Dr. Tooraj Nikoubin

ECE Department

Texas Tech University

TEXAS TECH UNIVERSITY

LUBBOCK, TEXAS, 79415

SUMMER 2019

ACKNOWLEDGEMENT

First and foremost, we would like to express our immense gratitude towards our institution **TEXAS TECH UNIVERSITY,** which is helping us to attain profound technical skills in the field of Electrical and Computer Engineering, thereby fulfilling our most cherished goal.

Our sincere thanks to Dr. Tooraj Nikoubin, ECE Professor, for providing this opportunity to carry out the present project work and for his encouragement and advice during the course of this work. We are also deeply indebted to him, for his constant encouragement and valuable guidance during the period of the project work.

ABSTRACT

A binary multiplier is an electronic circuit used in digital electronics, such as a computer, to multiply two binary numbers. It is designed using binary adders. It plays a very important role in today's digital signal processing and various other applications. Researchers in the field of computer architecture are working constantly to design computer processors and systems that has high speed, low power consumption, regularity of layout, and takes less area.

This project deals with the construction of a binary multiplier which multiplies two 16-bit binary numbers. A common multiplication method is "add and shift" algorithm. In parallel multipliers, number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, modified Booth algorithm is one of the most popular algorithms widely used. However, with increasing parallelism, the number of shifts between the partial products and intermediate sums to be added will increase which results in the increased power consumption and reduced speed. On the other hand, serial-parallel multipliers compromise speed to achieve better performance area and power consumption. Thus, the selection of a parallel or serial multiplier depends on the nature of application. In this project, we introduce the 16*16-bit binary multiplication algorithm that uses Full Adders, represent its pictorial design, and talk about its speed, area, power, and efficiency.

CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

This project deals with the implementation of 16x16 bit binary multiplier using full adders. There are plenty of various adders and all these adders differ in various aspects such as speed, complexity and structure. In this project the math is done utilizing two 24bit full adders and one 16-bit full adder. This report will detail what a full adder is at the design level as well as at the gate level. There also will be discussion on the speed and timing of the overall system.

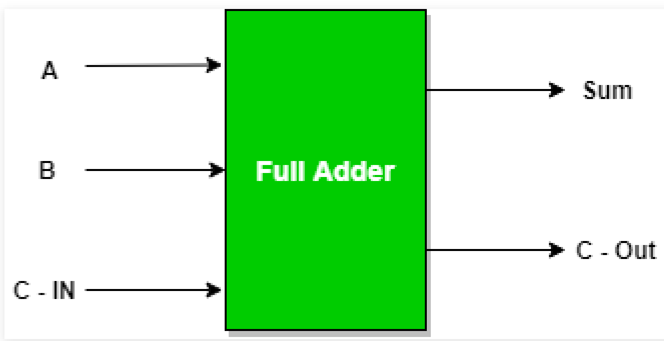
As part of the implementation, which was created in Verilog utilizing Xilinx ISE design suite, there were several modules designed such as one for the full adder, 24bit full adder and 16-bit full adder. With the combination of all of these modules, the solution is implemented in the main module to successfully get the proper product from the multiplication of two 16-bit binary numbers.

CHAPTER 2

FULL ADDER

This project deals with the implementation of 16-bit binary multiplier using Full Adders. Full Adders are building block for various components in an electronic system. There are so many types of adders that are different in various aspects like their structure and complexity. In most cases, Full Adders are the adders that takes three different inputs and produces two outputs. The first two inputs are A and B and the third is an input carry as C-In. The outputs are the Sum and the C-Out i.e. the output carry.

A full adder logic is usually designed in such a way that it can take 8, 16, 32, and so on number of bits binary inputs at a time and cascade the carry bit from one end to the other.



Inputs			Outputs	
A	B	C – IN	Sum	C - Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 1: Full Adder Schematic & Truth Table

A basic Full Adder is central to most digital circuits that perform addition or subtraction. The Full Adder logic circuit is presented below.

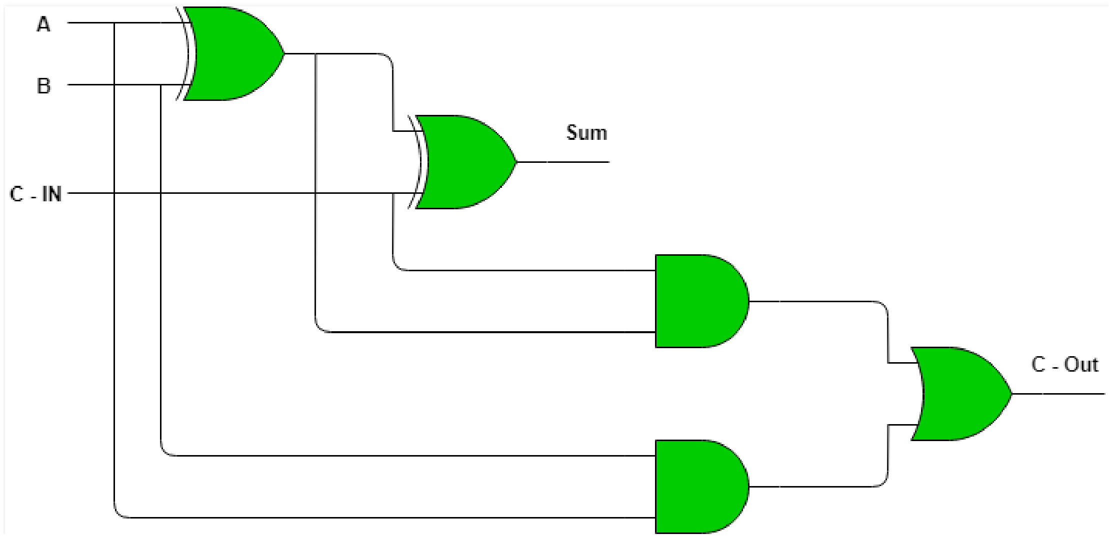


Figure 2: Full Adder Circuit

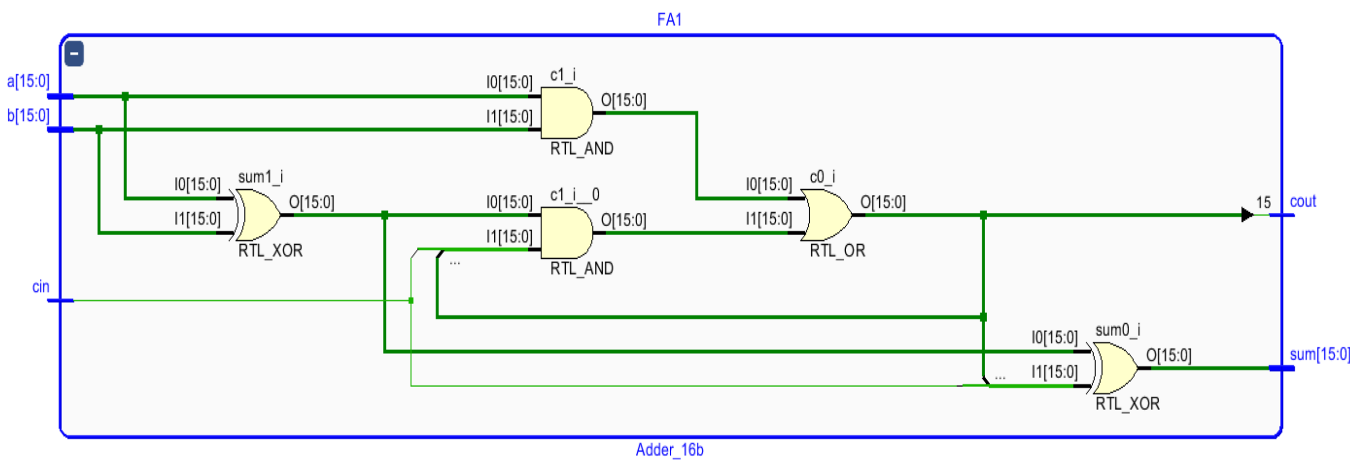


Figure 3 Full Adder Used in the Multiplier

Also, Two Half Adders and a OR gate can be used to implement a Full Adder.

CHAPTER 3

BINARY MULTIPLIER

A Binary Multiplier is an electronic circuit used in digital electronics to multiply two binary numbers. There are number of methods or computer arithmetic techniques that can be implemented to perform a digital multiplication. Most techniques involve computing a set of partial products, and then summing the partial products together. The same technique has been modified for the application to a binary numeral system.

The two numbers specifically known as multiplicand and multiplier can be of various size and the output result of them is termed product. The size of the product depends on the bit size of the multiplier and the multiplicand. The partial product of the LSBs of the inputs almost always stays the LSB of the product. The other terms of each partial product are considered and added using Full Adders.

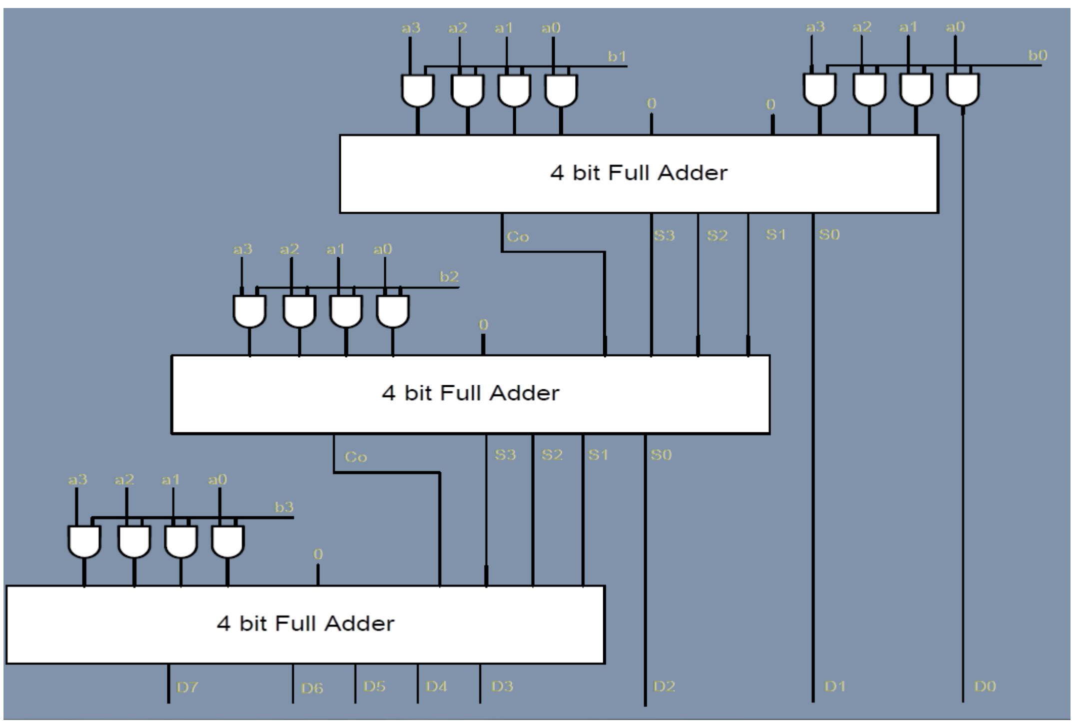


Figure 4: 4 Bit Binary Multiplier using Full Adders

CHAPTER 4

16 BIT MULTIPLIER MODULE

This is the module that we designed and implemented using 3 Full Adders: FA1 (16 Bit), FA2 (24 Bit), and FA3 (24 Bit). This model is very basic, and the code was written to implement this model was done in Verilog. Two inputs ‘a’ and ‘b’, 16 bit each are passed through as inputs into the 16-bit Full Adder and all of these bits gets propagated along with the carry bits to the other two 24-bit Full Adders and the result is a 33-bit output.

The 16-bit Full Adder named ‘FA 1’ in the circuit is shown in Figure 4. It is a combination of two AND gates, two XOR gates and one OR gate. The other Full Adder used in the circuit is a 24-bit binary adder. The figures below show the circuit diagram of the all the Full Adders used in the procedure of building the 16*16-bit binary multiplier. Also, the complete circuit of the implementation is also depicted in figure 7.

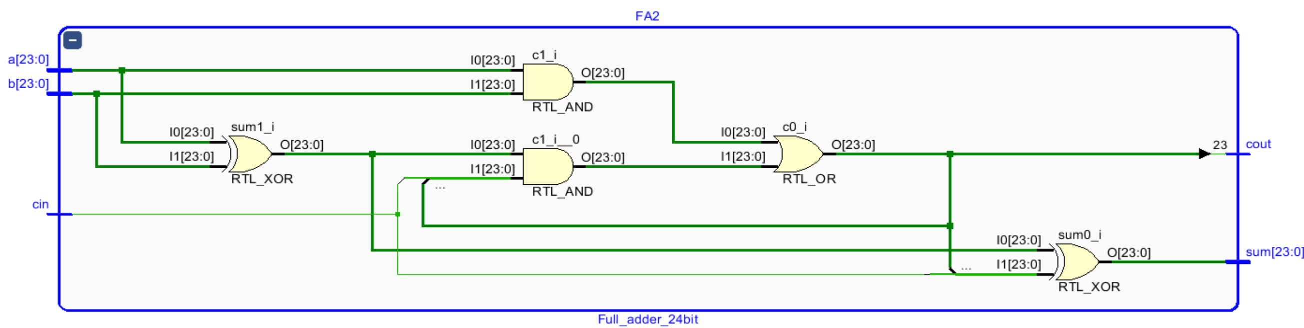


Figure 5: 24-bit Full Adder

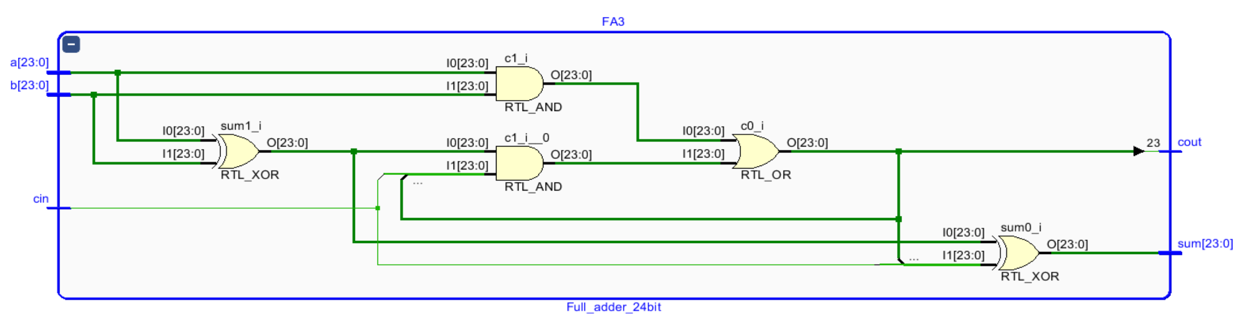


Figure 6: 24-bit Final Full Adder in the circuit

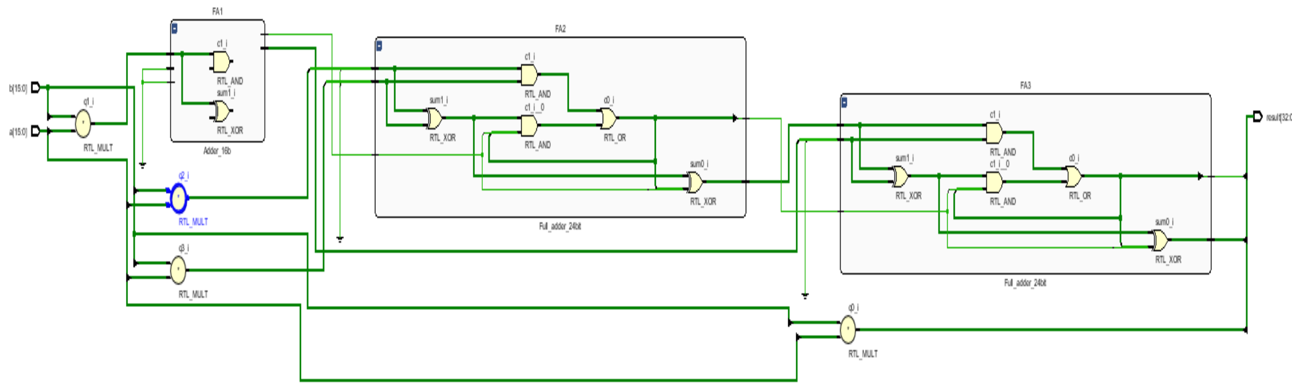


Figure 7: Full Circuitry of the Implementation

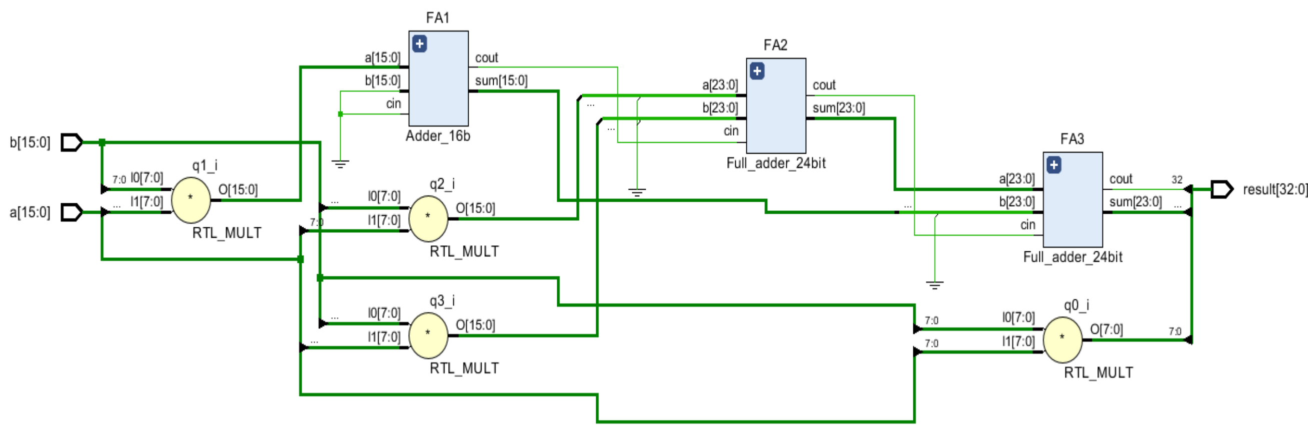


Figure 8: 16 Bit Binary Multiplier

CHAPTER 5

IMPLEMENTATION, TESTING & RESULT

The implementation of the 16-bit binary multiplier was performed using Verilog. The code was written in XILINX ISE and the simulation was performed onto 16 different multiplicand and multiplier. The functionality of the multiplier was verified more than 20 times and the accuracy of multiplication is 100%. The below shown table shows the data that was used for verification.

	A	B	C	D
1	**Input 1 (A)**	**Input 2 (B)**	**Result**	**Relevent Image**
2	010111000001111	010001100001111	000001100100111010111111111100001	Image 1
3	000000000001111	000000000001111	000000000000000000000000011100001	Image 2
4	010111011001111	000001101001111	000000001001101011011110111100001	Image 3
5	010100011011010	010001100001111	000001011001100000110010110000110	Image 4
6	010111011001111	100000100010011	000010111110011000001000010111101	Image 5
7	010111011001111	000000110001111	000000000100100011110100101000001	Image 6
8	010111011001101	000000110001000	000000000100011110101001111010000	Image 7
9	010111010000111	011110110001110	000010110010111111111001111100010	Image 8
10	010111011000001	011111110111000	000010111010001100011001101110000	Image 9
11	011111011001101	110110110100101	000110101110010111000011001000001	Image 10
12	010111010110000	111100110100111	000101100010111110100100110100000	Image 11
13	011000011001110	000000110001001	000000000100101011101100001111110	Image 12
14	010111011001001	010110110001010	000010000101001010001101010110100	Image 13
15	000111100001111	010110110001001	000000101010110110110010000001111	Image 14
16	000111111101010	001010100010011	000000010100111101100000010111110	Image 15

Figure 9: Table with Inputs and Outputs

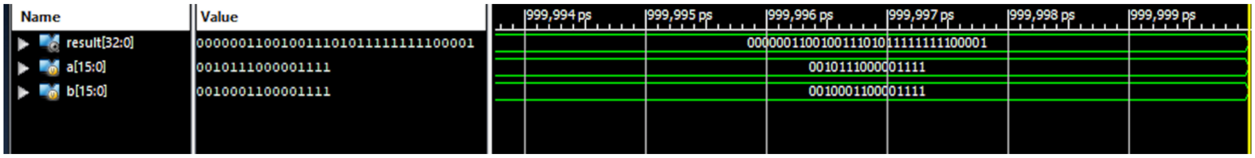


Figure 10: Image 1

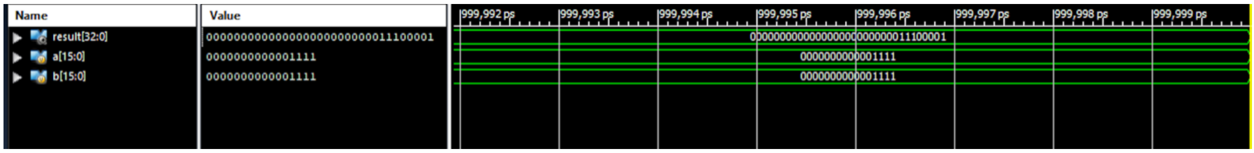


Figure 11: Image 2

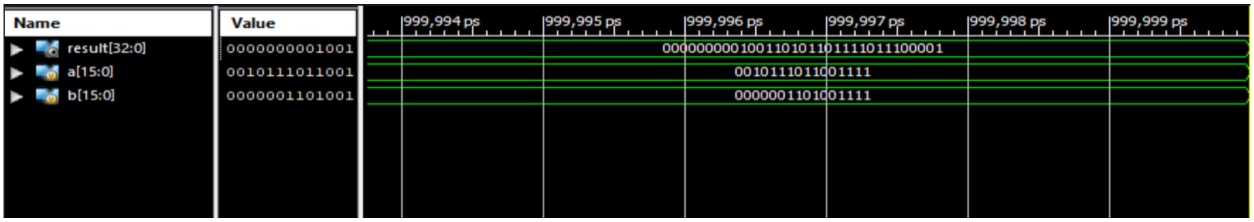


Figure 12: Image 3

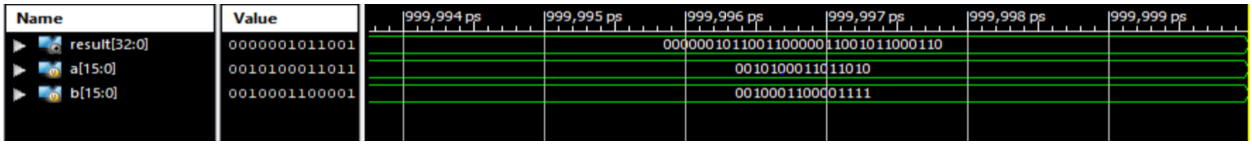


Figure 13: Image 4

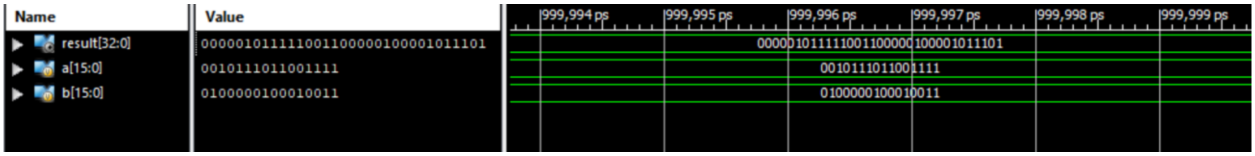


Figure 14: Image 5

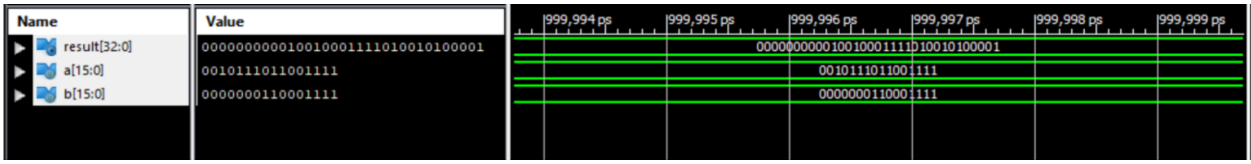


Figure 15: Image 6

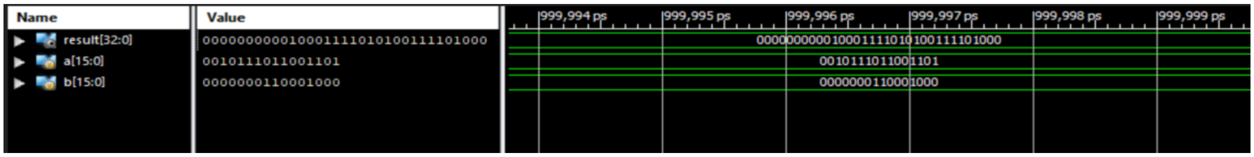


Figure 16: Image 7

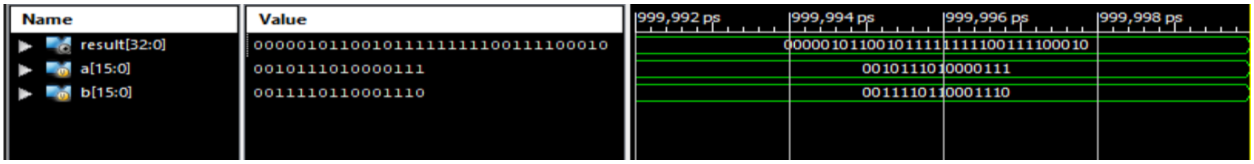


Figure 17: Image 8

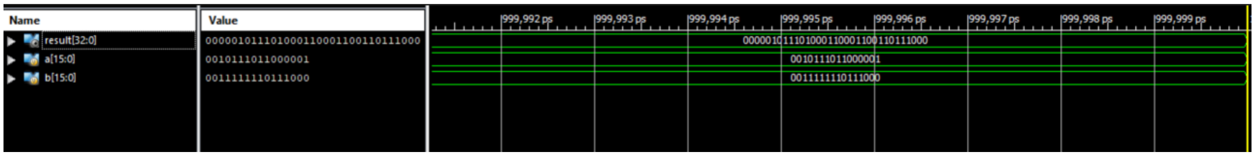


Figure 18: Image 9

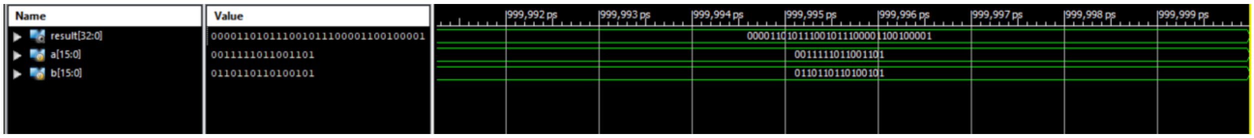


Figure 19: Image 10

CHAPTER 6

CONCLUSION

The 16*16-bit binary multiplier uses shift and add operators as a multiply routine. Due to large scale integration, it is possible to put or assemble adequate adders on a single chip to add all the partial products. However, this technique comes with some disadvantages along side with the super beneficial advantages. The advantages and disadvantages are discussed below.

Advantages:

- Regular structure of the system
- Easy to layout
- Ease of design for a pipeline structure
- Small size
- Design time is less

Disadvantages:

- As operand size increases, arrays grow to the squared size of the operand
- Underutilized hardware

There are better methods of doing binary multiplication in the world we live right now. Some famous techniques include Booth Multiplier Algorithm, Wallace Tree Multiplier, Modified Booth Multiplier, and so on. The speed, circuit complexity, layout, and area of all these multipliers are compared in the table below.

Multiplier Type	Speed	Circuit Complexity	Layout	Area
Array	Low/Medium	Simple	Regular	Smallest
Booth	High	Complex	Irregular	Medium
Wallace	Higher	Medium	More irregular	Large
Booth Wallace	Highest	More complex	Medium regularity	Largest

From looking at the table, it is concluded that the binary multiplier we have designed and implemented in this project consume less power and exhibits good performance. However, it has a limited number of bits for multiplier and multiplicand. For operands of 16 bit or higher, Booth algorithm or modified Booth algorithm reduce the partial product number by half and is comparatively more efficient then the array multiplication itself.

CHAPTER 7

REFERENCES

- https://en.wikipedia.org/wiki/Binary_multiplier
- Files from Dr. Nikoubin (Classroom Slides)
- https://www.electricaltechnology.org/2018/05/binary-multiplier-types-binary-multiplication-calculator.html#what-is-digital-binary-multiplier
- http://www.ijemr.net/DOC/DigitalMultipliers-AReview(220-223)8aa5905d-5da5-4b50-8b2e-f0bd850becb7.pdf
- https://www.electronics-tutorials.ws/combination/comb_7.html