International Journal for Modern Trends in Science and Technology
Volume 10, Issue 04, pages 41-44.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol10issue04.html
DOI: https://doi.org/10.46501/IJMTST1004007

Design and Comparison of High Speed Multipliers

Dr A V S Swathi[1], K Amrutha[2], P RVS Pavan kumar[2], K Hemanth Varaha Rajesh[2], K Jaya Sheel[2]

[1]Associate Professor, Raghu Engineering college (A), Visakhapatnam, India.
[2]Raghu Engineering college (A), Visakhapatnam. India.

To Cite this Article
Dr A V S Swathi, K Amrutha, P RVS Pavan kumar, K Hemanth Varaha Rajesh, K Jaya Sheel, Design and Comparison of High Speed Multipliers, International Journal for Modern Trends in Science and Technology, 2024, 10(04), pages. 41-44.https://doi.org/10.46501/IJMTST1004007

Article Info
Received: 18 March 2024; Accepted: 03 April 2024; Published: 04 April 2024.

ABSTRACT

A Binary multiplier is an integral part of the arithmetic logic unit (ALU) subsystem found in many processors whose binary multiplication is efficient in cost, time, and hardware. Integer multiplication can be inefficient and costly, in time and hardware, depending on the representation of signed numbers.

Here we design different multipliers whose performance is compared in reference with different parameters like speed, number of gates, time delays. The Multipliers which are compared are Basic Multiplier, Booth Multiplier, Radix 4 Booth's Multiplier, Wallace tree multiplier and booth encoded Wallace tree multiplier of 16 bit length. Here to compare various parameters we use Verilog as HDL in XILINX ISE tool

KEYWORDS: Wallace tree multiplier, ALU, Radix 4 Booth's multiplier

1. INTRODUCTION

Multipliers are the basic part of an ALU (Arithmetic and logical unit). The performance of many computational problems are often dominated by the speed at which a multiplication operation can be executed. To perform multiplication there are two terms known as multiplicand and multiplier. Each partial product is generated by multiplying the multiplicand with a bit of the multiplier – which, essentially, is an AND operation – and by shifting the result in the basis of the multiplier bit's position.

The 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. To achieve speed improvements Wallace Tree algorithm can be used to reduce the number of sequential adding stages. Further by combining both Modified Booth algorithm and Wallace Tree technique we can see advantage of both algorithms in one multiplier.

Literature Survey:

There are number of techniques to perform binary multiplication. In general, the choice is based upon factors such as latency, throughput, area, and design complexity. More efficient parallel approach uses some sort of array or tree of full adders to sum partial products. Array multiplier, Booth Multiplier and

Wallace Tree multipliers are some of the standard approaches to have hardware implementation of binary multiplier which are suitable for VLSI implementation at CMOS level.

A multiplier is one of the key hardware blocks in most digital signal processing (DSP) systems and ALU. Typical DSP applications where a multiplier plays an important role include digital filtering, digital communications and spectral analysis.

There are many types of multipliers which are used according to its own advantages and disadvantages.

The multipliers which we are going to discuss here are

i. Binary Multiplier
ii. Booth's Multiplier
iii. Radix 4 Booth's Multiplier
iv. Wallace tree Multiplier
v. Booth encoded Wallace tree multiplier

The booth algorithm is an effective technique for multiplying operands that are in 2s complement representation, including the case where the multiplier is negative. In the basic multiplier algorithm, each multiplier bit generates a version of the multiplicand that is added to the partial product. For large operands, the delay to obtain the product can be significant. The Booth algorithm reduces the number of partial products by shifting over strings of zeros in a recoded version of a multiplier.This depicts the different combinations for 2 bits in the multiplier. There are 4 different combinations i.e., 00,01,10,11. According to this algorithm, the multiplicand is either added or subtracted to the multiplier. The multiplicand and multiplier consists N+M+1 bits where N indicates the total number of bits in the multiplier and M indicates the total number of bits in the multiplicand. The addition or subtraction of multiplicand is done in the MSB of the multiplier, where the M+1 zeroes are appended in the multiplier as MSB bits. While N+1 zeroes are appended in the lsb bits of the multiplicand.

Existing Method:

Basic Binary Multiplier

AND gates are used to generate the Partial Products, PP, If the multiplicand is N-bits and the Multiplier is M-bits then there is N* M partial product. The way that the partial products are generated or summed up is the difference between the different architectures of various multipliers Multiplication of binary numbers can be decomposed into additions. Consider the multiplication of two 8-bit numbers A and B to generate the 16 bit product P.

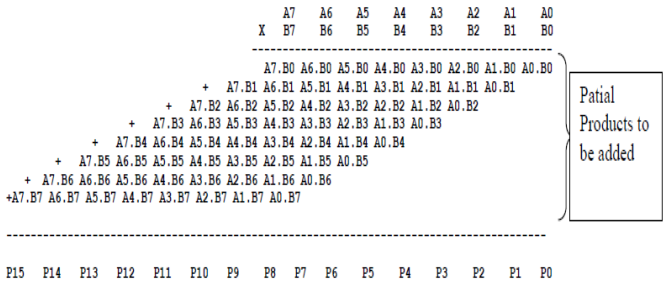


Fig: Basic Multiplier Partial Products formation

Steps involved in Binary Multiplication Algorithm

1. If the LSB of Multiplier is '1', then add the multiplicand into an accumulator.
2. Shift the multiplier one bit to the right and multiplicand one bit to the left.
3. Stop when all bits of the multiplier are zero.

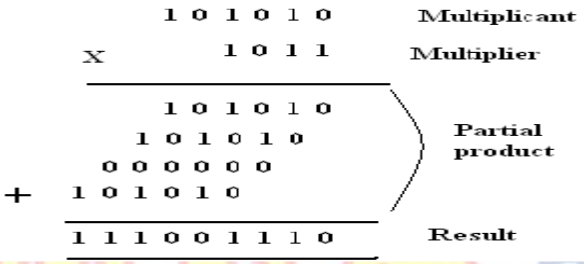


Fig: Basic Binary Multiplier

Proposed Design:

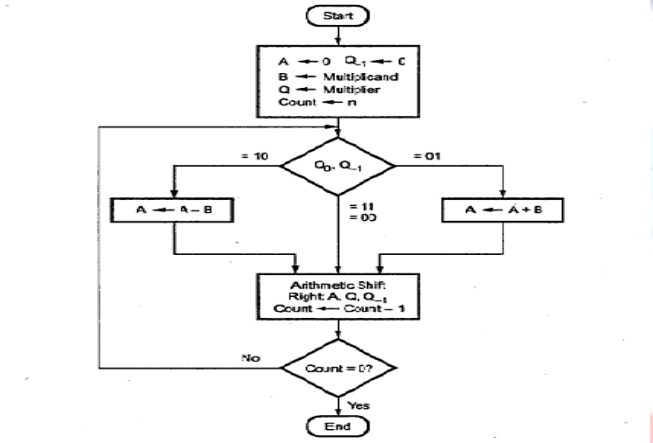
The booth algorithm is an effective technique for multiplying operands that are in 2s complement representation, including the case where the multiplier is negative. In the basic multiplier algorithm, each multiplier bit generates a version of the multiplicand that is added to the partial product. For large operands, the delay to obtain the product can be significant. The Booth algorithm reduces the number of partial products by shifting over strings of zeros in a recoded version of a multiplier.

Multiplier		Version of Multiplicand
Bit i	Bit i-1	
0	0	0 * Multiplicand
0	1	+1* Multiplicand
1	0	-1 * Multiplicand
1	1	0 * Multiplicand

Radix-2 Booth multiplier Recoding Table

The below flow chart indicates the booth's algorithm for radix2 multiplier. This depicts the different combinations for 2 bits in the multiplier. There are 4 different combinations i.e., 00,01,10,11. According to this

algorithm, the multiplicand is either added or subtracted to the multiplier. The multiplicand and multiplier consists N+M+1 bits where N indicates the total number of bits in the multiplier and M indicates the total number of bits in the multiplicand. The addition or subtraction of multiplicand is done in the MSB of the multiplier, where the M+1 zeroes are appended in the multiplier as MSB bits. While N+1 zeroes are appended in the lsb bits of the multiplicand.



For 00 combination, multiplier is shifted left.

01 combination, multiplicand is added and shifted left.

10 combination, multiplicand is subtracted i.e., the 2's compliment of multiplicand is added and shifted left.

11 combination, multiplier is shifted left.

SIMULATION RESULTS:

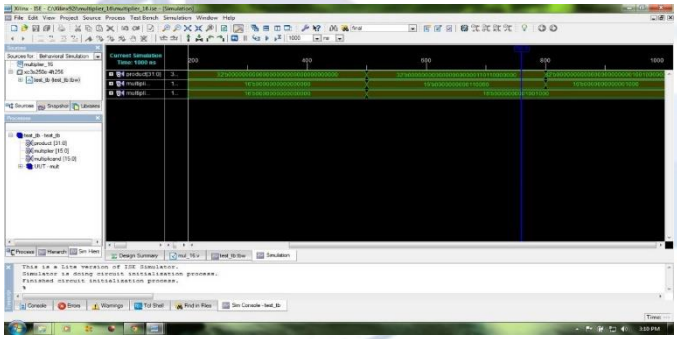


Fig: 16-bit basic binary multiplier result from Xilinx ISE

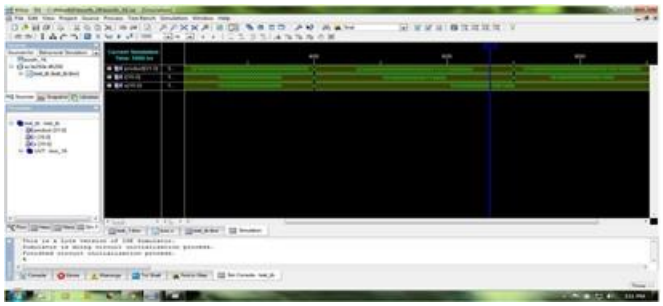


Fig: 16-bit Booth Multiplier result from Xilinx ISE

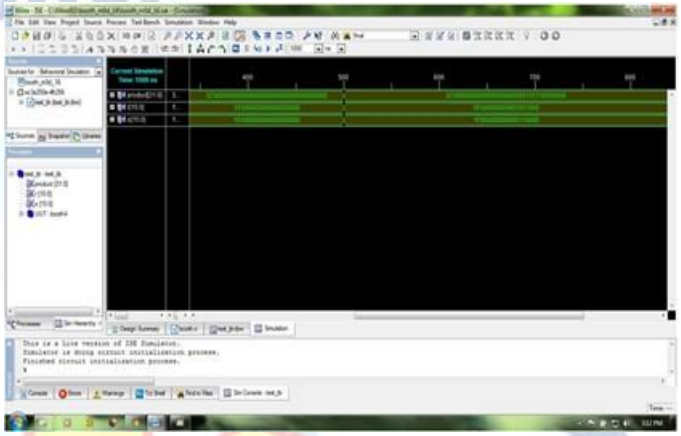


Fig: 16 bit Radix-4 Booth Multiplier using Xilinx

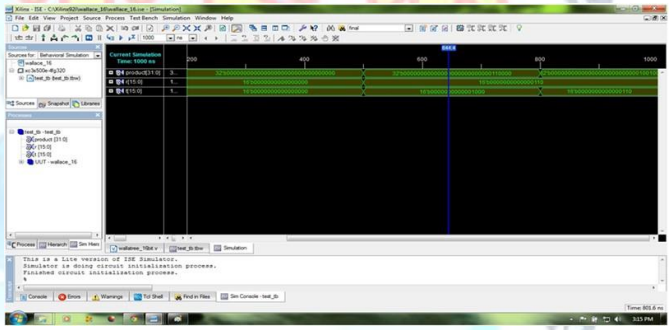


Fig: 16 bit Wallace multiplier result using Xilinx ISE

The discussed multipliers are compared in terms of area and speed as shown in the following table:

S.NO	Type Of Multiplier	Slices	Look Up Tables
1	Basic Multiplier	270	507
2	Radix 2 Booth encoded	162	290
3	Radix 4 Booth encoded	128	239
4	Wallace tree	212	379
5	Booth Encoded Wallace	111	210

CONCLUSION:

Now-a-days the application of multipliers became the necessity and decreasing the area usage and delay became the major task in VLSI designing. So keeping this in mind, various multipliers are being developed to

overcome these conflicts and bring out the most efficient multiplier. In this project we have designed various multipliers (like Basic multiplier, Radix 2 Booth Multiplier, Radix 4 Booth Multiplier, Wallace tree and booth encoded Wallace tree multiplier) having width of 16 bit. These Multipliers are simulated and synthesized using Xilinx ISE tool. While comparing the radix 2 and the radix 4 booth multipliers we found that radix 4 consumes lesser power than that of radix 2. This is because it uses almost half number of iteration and adders when compared to radix 2. Our Project gives a clear concept of different multipliers in terms of area and speed. We found that the proposed architecture booth encoded Wallace tree architecture gives very less delay and occupies less space. This is clearly depicted in our results. This multiplier speeds up the calculation and makes the system faster.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] V.S Kumar "Analysis of energy efficient PTL based full adders using different nanometer technologies", Electronics and Communication Systems (ICECS) , 2nd International Conference on, IEEE (2015), pp. 310-315 .2015.

[2] S Srikanth, I.T. Banu, G.V. Priya, G. Usha "Low power array multiplier using modified full adder",2016 IEEE International Conference on Engineering and Technology (ICETECH), IEEE (2016), pp. 1041-1044

[3] M. Moris Mano, "Computer Systems Architecture", 3rd Edition, Pearson Education, 2006.

[4] Wayne Wolf, "Modern VLSI Design", Pearson Education, 3rd Edition, 1997.

[5] John. P. Uyemura, "Introduction to VLSI Circuits and Systems", 1st Edn., 2003. John Wiley.

[6] Joseph Cavanaugh, "Computer Arithmetic and Verilog HDL fundamentals", 1st Edn., 2005.

[7] Samir Palnitkar, "Verilog HDL", Pearson education, 2nd edition, 2003.

[8] Ravi Nirlakalla, Thota Subba Rao, Talari Jayachandra Prasad, "Performance Evalution of High Speed Compressor for High Speed Multipliers,"

[9] P. E. Madrid, B. Millar, and E. E. Swartzlander, "Modified Booth algorithm for high radix fixed-point multiplication," IEEE Trans. VLSI Syst., vol. 1, no. 2, pp. 164-167, June 1993.