Survey of High Speed Efficient Modified 64-Bit Booth Multiplier using VHDL

Miss. Sayali Gaidhane[1] Prof. R. D. Kadam[2]
[1]M.Tech Student [2]Assistant Professor
[1,2]Department of Electronics and Telecommunication
[1,2]BDCOE, Sewagram, Maharashtra, India

Abstract— *This paper focuses on detail survey of 64-bit Modified Booth Multiplier using VHDL. Further we will introduce CSA (Carry save Adder) into the proposed design to increase the speed and performance of the proposed design. CSA is one way of improving the overall processing performance of a multiplier. Here, CSA with modified techniques are used to increase the speed of the Booth Multiplier; also it can perform fast operations as compared to normal multiplier. Design, synthesis and simulation of 64-bit Modified Booth Multiplier will be done using XILINX ISE 14.5 software. Coding of the proposed design will be done in VHDL (Very high Speed Integrated Circuit Hardware Description Language).*

Keywords— Booth Multiplier, XILINX ISE, VHDL

I. INTRODUCTION

Multipliers are most commonly used in various electronic applications e.g. Digital signal processing in which multipliers are used to perform various algorithms like FIR, IIR etc. Earlier, the major challenge for VLSI designer was to reduce area of chip by using efficient optimization techniques to satisfy MOORE'S law. Then the next phase is to increase the speed of operation to achieve fast calculations like, in today's microprocessors millions of instructions are performed per second. Speed of operation is one of the major constraints in designing DSP processors and today's general-purpose processors. However area and speed are two conflicting constraints. So improving speed results always in larger areas. Now, as most of today's commercial electronic products are portable like Mobile, Laptops etc. that require more battery back-up. Therefore, lot of research is going on to reduce power consumption. So, in this paper it is tried to find out the best solution to achieve low power consumption, less area required and high speed for multiplier operation [4].

Digital computer arithmetic is an aspect of logic design with the objective of developing appropriate algorithms in order to achieve efficient utilization of the available hardware. The basic operations are addition, subtraction, multiplication and division. Addition operation plays an important role in the design of multiplier as repeated form of addition operations and shifting results in the multiplication operation [2]. Low power consumption in CMOS circuits is the present research as high integration causes increases the power dissipation. Modified Booth Encoders attract much attention because of low power dissipation and high throughput. The advantages of the Modified Booth Algorithm are the operation speed and the power dissipation. The AND array method and the Booth encoding method are algorithms for partial product generation. The reduction of partial products is required for higher-bit multiplication, because the addition of partial products stage occupies a large circuit area and causes huge delay [1].

In general, a multiplier uses Booth encoder and array of full adders (FAs), or CLA tree instead of the array of FAs. This multiplier mainly consists of the three parts: Booth encoder, a tree to compress the partial products such as CLA tree, and final stage adder. CLA tree is to add the partial products from encoder as parallel as possible. In real implementation, many (4:2) compressors are used to reduce the number of outputs in each pipeline step. The most effective way to increase the speed of a multiplier is to reduce the number of the partial products because multiplication precedes a series of additions for the partial products. To reduce the number of calculation steps for the partial products, Modified Booth Encoder has been applied mostly where CLA tree has taken the role of increasing the speed to add the partial products. The process of computation in various digital circuits can be speed up through various arithmetic operations such as addition, subtraction and multiplication [1].

A. Booth Algorithm

Booth algorithm is used for simulation and development of digital multiplier. Booth multiplication algorithm is the multiplication algorithm that multiplies two signed binary numbers in two's complement form. The algorithm was invented by Andrew Donald Booth 1951 while doing research on crystallography in London. Booth algorithm uses a small number of additions and shift operations to do the work of multiplications. It is a powerful algorithm for signed-number multiplication which treats both Positive number as well as Negative number Booth algorithm is a method that will reduce the number of multiplicand multipliers.

B. Booth Algorithm Flowchart

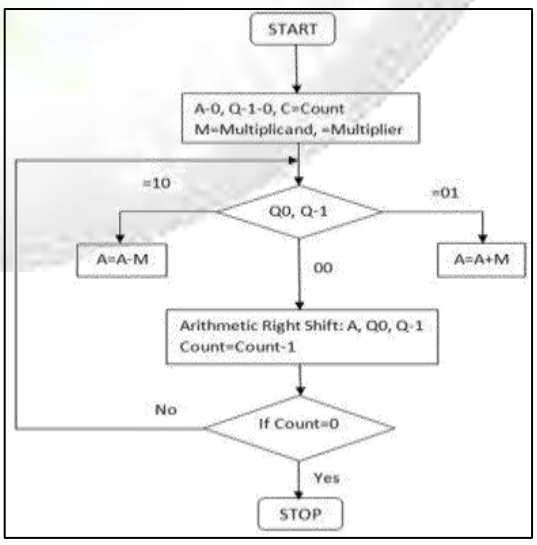

Fig. 1: Booth Algorithm Flowchart

II. LITERATURE REVIEW

Nyamatulla Patel et.al [1] presents a design of High Speed Hardware Efficient Modified Booth Multiplier Using HDL. A 2-bit Booth encoder with Josephson Transmission Lines (JTLs) and Passive Transmission Lines (PTLs) is designed. The Booth encoding method is one of the algorithms to obtain partial products. With this method, the number of partial products decreases down to the half compared to the AND array method. The circuit area of the multiplier designed with the Booth encoder method is compared to that designed with the AND array method. The proposed 64-bit modified booth encoders are designed using Modified Booth Algorithm and Carry Look Ahead Adder. The efficiency of this project is verified by successive execution with different inputs. The 64- bit multiplication is executed successfully and got the correct results. The project has reduced the number of partial products considerably, and increased the speed of execution. The program has given correct results for number of times. The project code is much efficient in terms of number of bits and in reducing the amount complexity accomplished. This project can be further modified to obtain the results for the numbers with the numbers of bits greater than 64. They can also be used for designing SFQ logic circuits [1].

Shaik Meerabi et.al [2] describes a design and Implementation of 64-Bit Multiplier Using CLAA and CSLA. They have shown comparative study of the Carry Look-Ahead Adder (CLAA) based 64-bit unsigned integer multiplier and Carry Select Adder (CSLA) based 64-bit unsigned integer multiplier. Multiplication is a fundamental operation in most of the signal processing algorithms. Multipliers occupy large area, long latency and consume considerable power. Therefore there is a need for designing a multiplier that consumes less power. Moreover the digital system's efficiency is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system, and consumes more area. Hence, optimizing the speed, area and delay of the multiplier is a major design issue. Carry Select Adder (CSLA) is one of the fastest adders used in many applications to perform fast arithmetic functions. From the structure CSLA there is a scope for reducing area and delay by using Common Boolean Logic (CBL). This work evaluates the performance of the proposed designs in terms of area, delay, and power. The power dissipation is same for both CLAA based multiplier and CSLA based multiplier. But the area delay product of modified CSLA based multiplier is reduced to 6% when compared to CLAA based multiplier. These multipliers are simulated and synthesized using Modelsim6.4b and Xilinx 10.1. They present a design and implementation of 64-bit unsigned multiplier with CLAA and CSLA. The design is tested with the existed adders [2].

Sweta Khatri et.al [3] presents the FPGA Implementation of 64-bit fast multiplier using barrel shifter. In this research they have described the implementation of a 64-bit Vedic multiplier which is enhanced in terms of propagation delay when it is compared with conventional multiplier like modified booth multiplier, Wallace tree multiplier, Braun multiplier, array multiplier. They use various Vedic multiplication techniques for arithmetic operation .It has been found that the most efficient of all the sutra is Urdhva-triyagbhyam, which gives minimum delay for multiplication of all types of numbers, either small or large numbers. For 'n' number of shifts only one clock cycle is required in our design, using 64-bit barrel shifter. The design is implemented and verified using ISE simulator and FPGA. Synthesis report and static timing report are used for the comparison of propagation delay. The design uses barrel shifter in base selection module and multiplier which achieves propagation delay of 6.781ns [3].

Deepthi et.al [4] presents a Performance Analysis of a 64-bit signed Multiplier with a Carry Select Adder Using VHDL. They have presented a performance analysis of carry-look-ahead-adder and carry select adder signed data multiplier we are using, one uses a carry-look- ahead adder and the second one uses a carry select adder. The main focus of their research's on the speed of the multiplication operation on these 64-bit multipliers which are modeled using verilog code, a hardware description language. The multiplier with a carry select adder has shown a better performance over the multiplier with a carry select adder in terms of gate delays. In this paper we are going to prove that the area and delay product of carry select adder gives better performance compare with carry-look-ahead adder signed 64 bit multiplier. Use booth's multiplier with CSLA if area is critical. Use booth's multiplier without CSA if area is critical and a bit of compromise on timing can be made. The Design of high speed bit signed multiplier using adders is proposed. Simulation and synthesis of high speed Bit signed multiplier using CLAA and CSLA has been done in Xilinx 10.2 E using verilog Hardware Description Language. The CSLA increases the performance of the multiplier. This radix-4 algorithm can be extended to radix-16 algorithms to get a high speed and efficient multiplication This 64 bit multiplier can be further extended to 128 bit multiplier and 256 bit multiplier using the proposed method for multiplication operation can be done as future work [4].

EPPILI JAYA et.al [5] presents POWER, AREA AND DELAY COMPARISION OF DIFFERENT MULTIPLIERS. In this research, multipliers for low power applications were implemented. Three basic algorithms namely Booth, Vedic and Modified Booth multipliers were implemented on Spartan 3E XLINX 13.1 version. The Power, Delay and Area are calculated for these algorithms. Power measurements were performed using Xilinx power estimator. From the table5, it is evident that Modified Booth multiplier requires less time, low power and less area to implement compared with Booth and Vedic multipliers. This low power, fast and area efficient multiplier can be used for FIR filter design, MAC design as an extension to this paper. Multipliers play an important role in today's digital signal processing and various other applications. High speed and low power multiplier unit is the requirement of today's VLSI systems and Digital Signal Processing applications. Multiplication operation involves generation of partial products and their accumulation. The speed of multiplication can be increased by reducing the number of partial products. So, minimization of partial products is the main requirement. Here the fast multipliers like Booth multiplier, Vedic multiplier and Modified Booth recoded multiplier are designed, analyzed and compared on the basis of Power, Speed and Area. The analysis i.e., simulation and

synthesis of above multipliers were done using XILINX 13.1 [5].

Many of the literature reviewed presents a Design of High Speed Efficient Modified 64-Bit Booth Multiplier Using VHDL to ease the description, verification, simulation and hardware realization. The 64- bit multiplication is executed successfully and got the correct results [1]. They present a design and implementation of 64-bit unsigned multiplier with CLAA and CSLA. The design is tested with the existed adders [2]. The design uses barrel shifter in base selection module and multiplier which achieves propagation delay of 6.781ns [3]. This 64 bit multiplier can be further extended to 128 bit multiplier and 256 bit multiplier using the proposed method for multiplication operation can be done as future work [4].

A. Comparison Table

PARAMETERS	REF [1]	REF [2]			REF [3]		REF [4]		REF [5]		
Method	—	CLAA based 64-bit multiplier	Regular CSLA based 64-bit multiplier	Modified CSLA 64-bit multiplier	64-bit normal	64-bit Vedic	CLAA	CSLA	Booth	Vedic	Modified Booth
Power (mW)	—	203	203	203	—	—	—	—	135	133	95
Delay (ns)	31.18	224.370	280.910	210.881	315.871	315.725	172.4	172.02	52.627	31.929	19.959
No of SLICES	—	—	—	—	—	—	—	—	35%	27%	8%
No of LUTs	—	—	—	—	220893	214645	—	—	31%	20%	8%
Area		4787	10718	4709	—	—	4938	4018	—	—	—

III. PROPOSED WORK

This research work targets the design of design of High Speed Efficient Modified 64-Bit Booth Multiplier Using VHDL. It includes; design of high speed 64-bit multiplier, Multiplier will be design using CSA, modified booth algorithm and pipelining.

Multiplication is an important fundamental function in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform (FFT), filtering and in microprocessors in its arithmetic and logic unit. Since multiplication dominates the execution time of most DSP algorithms, so there is a need of high speed multiplier. The following are the main objective of this project.

- To reduce delay.
- To reduce power.
- To increase the speed

The Proposed work is to design of High Speed Efficient Modified 64-Bit Booth Multiplier Using VHDL. High Speed Efficient Modified 64-Bit Booth Multiplier can be composed with Booth Encoder, Booth Decoder, Booth Multiplier and Carry Save Adder. Following figure shows a proposed block diagram of High Speed Efficient Modified 64-Bit Booth Multiplier Using VHDL.

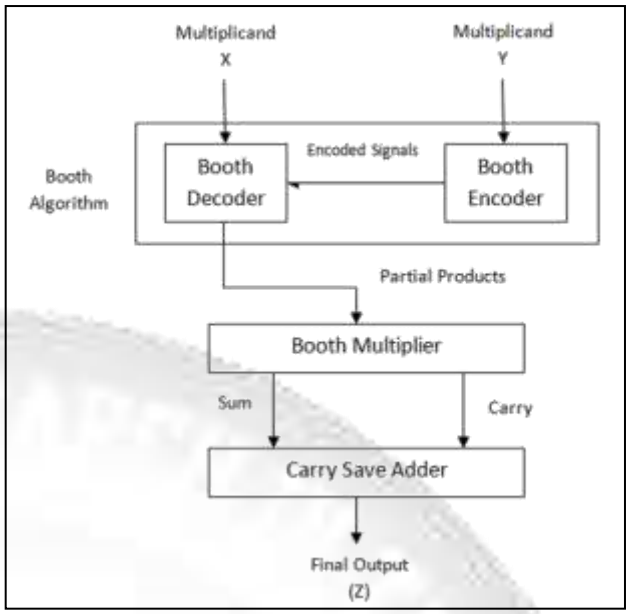

Fig. 2: Diagram of Proposed 64-Bit Booth Multiplier

IV. CONCLUSION

In this paper we have discussed in detail about High Speed Efficient Modified 64-Bit Booth Multiplier Using VHDL. Design, synthesis and simulation of Modified 64-Bit Booth Multiplier will be done using XILINX ISE 14.5. Coding of the proposed design will be done in VHDL. Future work is likely to achieve the design, synthesis and simulation of Booth Multiplier, CSA adder and finally the High Speed Efficient Modified 64-Bit Booth Multiplier Using VHDL. Therefore, Modified 64-Bit Booth Multiplier using VHDL is the aim of this research work.

REFERENCES

[1] Nyamatulla Patel, Vidyashri M. Bastawadi, Suparna R. Daddimani, "Design of High Speed Hardware Efficient Modified Booth Multiplier Using HDL", Journal of Advances in Science and Technology Vol. 14, Issue No. 1, June-2017, ISSN 2230-9659.

[2] Shaik Meerabi, Krishna Prasad Satamraju, "Design and Implementation of 64-Bit Multiplier Using CLAA and CSLA", International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013).

[3] Sweta Khatri, Ghanshyam Jangid, "FPGA Implementation of 64-bit fast multiplier using barrel shifter", INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLO GY (I JRAS ET), Vol. 2 Issue VII, July 2014.

[4] Deepthi, Rani, Manasa, "Performance Analysis of a 64-bit signed Multiplier with a Carry Select Adder Using VHDL", IJCSNS International Journal of Computer Science and Network Security, VOL.15 No.11, November 2015.

[5] EPPILI JAYA, "POWER, AREA AND DELAY COMPARISION OF DIFFERENT MULTIPLIERS", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 5, Issue 6, June 2016.

[6] Bai A. Akahori, M. Tanaka, A. Sekiya, A. Fujimaki, and H.Hayakawa (1996). "Design and demonstration of SFQ pipelined multiplier," IEEE Trans. Appl. Supercond, vol. 13, no. 2, pp. 559–562.
[7] Cherkauer B. and Friedman E. (1996). "A Hybrid Radix-4/Radix-8 Low Power, High Speed Multiplier Architecture for Wide Bit Widths", In IEEE International Symposium on Circuits and Systems, volume 4, pages 53–56.
[8] Da Huang, Afsaneh Nassery (2002). "Encoding Radix 4 8 bit multipliers", final project report, pp. 5- 6.
[9] David Villager, Vojin Goklobdzija (2000). "Analysis of booth encoding in parallel multipliers using compressor for reduction of partial products", University of California, pp. 1-2.
[10] Hsin-Lei Lin, Robert C., Chang, Ming-Tsai Chan (2003), "Design of a Novel Radix-4 Booth Multiplier", pp. 14-17.