

# Review on Implementation of a High Performance Multiplier Using HDL

Prof. G. D. Dalvi, Miss. Prajakta P. Chaure

**Abstract**— Paper presents implementation of high performance multiplier using HDL. In that paper for implementing the multiplier uses both encoder and carry look-ahead adder. This paper use VHDL language. The paper implementing for simulation used two types of software: Xilinx ISE 9.2i target towards Spartan 3 FPGA, modelsim SE6.4. Paper is based on FPGA and VLSI. Carry look ahead are used for enhance the speed of operation. Radix-4 Booth multiplier with 3:2 compressors and Radix-8 Booth multiplier with 4:2 compressors are presented here. The design is structured for  $m \times n$  multiplication where  $m$  and  $n$  can reach up to 126 bits. Finally the performance implementing multiplier is validated when implementing higher order FIR filter. Area and speed are usually conflicting constraints so that for improving the speed of the system in large area. The multiplier is an essential element of the digital signal processing such as filtering and convolution. Most digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) or discrete Wavelet transform (DWT). In this paper, we will use essential technique for improving the performance of the multiplier.

**Index Terms** — FPGA, HDL, Carry look-ahead adder, Booth encoding, Wallace Tree.

## I. INTRODUCTION

The multiplier is an essential element of the digital signal processing such as Filtering and convolution. Multiplier consumes much area and dissipates more power. In electronics and telecommunication system and multimedia, real time signal processing and large capacity data processing is increasing being demanded. Most of the DSP computation involves the use of multiply accumulated operations. Therefore the design of fast and efficient multiplier is imperative. In this project we have used VHDL as HDL and XILLINX ISE for decreasing and verifying a hardware design based on booths and some other efficient algorithm timing and correctness properties are verified. We try to present an efficient multiplier is produce fast, accurate and required minimum area. Multiplication operation involves generation of partial products and their accumulation. The speed of multiplication can be increased by reducing the number of partial products. Among the many method of implementing high speed parallel multiplier, there are two basic approaches namely booth algorithm and Wallace tree compressors. This paper describes an efficient implementation of high speed parallel multiplier using both these approaches. Here two multipliers are proposed. The fist multiplier make use of the radix 4 booth algorithm with 3:2 compressors while the second multiplier use radix 8 booth algorithm with 4:2 compressors. The design is structured for  $m \times n$  multiplication where  $m$  and  $n$  can reach up 126 bits. The number of partial product Is  $n/2$  radix 4 booth algorithm while it gets reduce to  $n/3$  radix 8 booth algorithm. The Wallace trees use carry save adders (CSA) to accumulate the partial products. This

reduces time as well as chip area. Most digital signal processing method used non liner function such as discrete cosine transform (DCT) and Discrete Wavelet Transform (DWT). As they are basically accomplished by repetitive application of multiplication and addition their speed become a major factor which is determine the performance.

## II. LITERATURE SURVEY

Dong-Wook Kim, Young-Ho Seo, had published a paper on “A New VLSI Architecture of Parallel Multiplier-Accumulator based on Radix-2 Modified Booth Algorithm”, in 04 Feb. 2010. In this paper, they proposed a new architecture of multiplier-and-accumulator (MAC) for high-speed arithmetic. By combining multiplication with accumulation and devising a hybrid type of carry save adder (CSA), the performance was improved. Since the accumulator that has the largest delay in MAC was merged into CSA, the overall performance was elevated. The proposed CSA tree uses 1's-complement-based radix-2 modified Booth's algorithm (MBA) and has the modified array for the sign extension in order to increase the bit density of the operands. The CSA propagates the carries to the least significant bits of the partial products and generates the least significant bits in advance to decrease the number of the input bits of the final adder. Also, the proposed MAC accumulates the intermediate results in the type of sum and carry bits instead of the output of the final adder, which made it possible to optimize the pipeline scheme to improve the performance. The proposed architecture was synthesized with 250, 180 and 130m, and 90 nm standard CMOS library. Based on the theoretical and experimental estimation, we analysed the results such as the amount of hardware resources, delay, and pipelining scheme. We used Sakurai's alpha power law for the delay modelling. The proposed MAC showed the superior properties to the standard design in many ways and performance twice as much as the previous research in the similar clock frequency. We expect that the proposed MAC can be adapted to various fields requiring high performance such as the signal processing areas [1].

Prasanna Raj P, Rao, Ravi had published a paper on “VLSI Design and Analysis of Multipliers for Low Power”, in Sept. 2009. In this paper they said that Low power multipliers with high clock frequencies play an important role in today's digital signal processing. In this work, the performance analysis of Wallace-tree, Array and Baugh-Wooley multiplier architectures is carried out. Physical verification of all the sub-blocks is performed using HSpice to check their functionality and to optimize for low power by using transistor sizing. The layouts of the sub-blocks are drawn using Cadence Virtuoso to form the multipliers macros. DRC and LVS checks are performed using HerculesI and fed to RC-XT for parasitic extraction and to carry out post layout simulation and the power

analysis using Astro rail. Delay and power dissipation of Wallace Tree multiplier is least whereas Array multiplier is a best for reduced area applications but not speed. In this work, the area of 5times5 Array multiplier is 67.73 times 7  $\mu\text{m}^2$  is the least compared to others. Each multiplier has to be selected depending on performance measures and nature of applications [2].

### III. PROPOSED WORK

The architecture of the proposed multiplier is shown in fig.1. It consists of four major modules: booth encoder, partial product generator, Wallace tree, carry look-ahead adder, booth encoder performs Radix2 and radix 4 encoding of the multiplier of the bits. Based on the multiplicand and the Encoded multiplier, partial products are generated by the generator. For large multiplier of 32 bits, the performance of the modified booth algorithm is limited. So booth recording together with Wallace tree structures have been used in the fast multiplier. The partial products are supply to Wallace tree and added appropriately. The result are finally Added using a carry look-ahead adder (CLA) to get the final products.

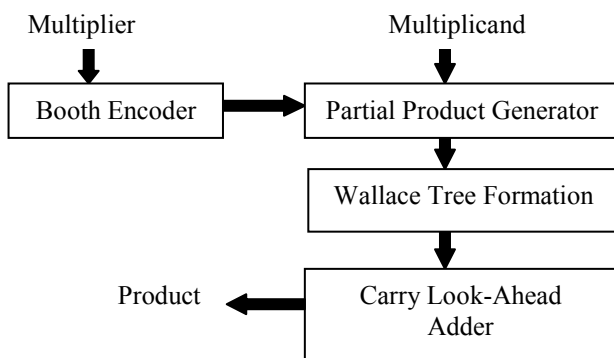


Fig 1: Block Diagram Of Wallace Booth Multiplier

Radix 4 booth algorithm is powerful algorithm for sign number multiplication, which treats of both positive and negative numbers uniformly .since k bit binary number can be interpreted as k/2 digit radix 4 number, as k/3 digit radix 8 number as so on, it can be deal with more than one bit of the multiplier in each cycle by using high radix. Radix 8 booth recording applies same algorithm as that of radix 4. Radix 8 booth reduce the number of partial products to n/3, where n is the number of multiplier bit. The Wallace tree method is used high speed design in order to produce two row of partial product that can be added in last stage. Wallace tree has taken the role of accelerating the accumulation of last products. The speed, area and power consumption of the multiplier will be in direct proportion to the efficiency of compressor. On this review paper we are using the essential factor that is booth algorithm Wallace tree formation partial product generator that is generating by generator. Wallace factor reducing the excess of factor that is low power consumption and improving the high performance of the multiplier. We are using high speed multiplier for implementation which is critical requirement for a processor with a high performance. We are using some additional operation for all the partial products. We are using fast additional adder like carry look ahead adder for multiplier for the improving the high performance of the

multiplication operations. In future we will use that additional structure for industrial work.

FPGA or Field Programmable Gate Arrays can be programmed or configured by the user or designer after manufacturing and during implementation. Hence they are otherwise known as On-Site programmable. Unlike a Programmable Array Logic (PAL) or other programmable device, their structure is similar to that of a gate-array or an ASIC. Thus, they are used to rapidly prototype ASICs, or as a substitute for places where an ASIC will eventually be used. This is done when it is important to get the design to the market first. Later on, when the ASIC is produced in bulk to reduce the NRE cost, it can replace the FPGA. The programming of the FPGA is done using a logic circuit diagram or a source code using a Hardware Description Language (HDL) to specify how the chip should work. .

### IV. CONCLUSION

In this paper we will implement the high performance multiplier using some techniques which is very essential. That will to reduce complexity of the system. Using are essential techniques so we will consume much more area and dissipate more power that increases the reliability and the performance of the multiplier. In this paper we are using additional operation that will help us to increasing overall performance and increasing the reliability.

### ACKNOWLEDGEMENT

I am very thankful to prof. G. D. dalvi for his valuable comments on this work.

### REFERENCES

- [1] Dong-Wook Kim, Young-Ho Seo, "A New VLSI Architecture of Parallel Multiplier-Accumulator based on Radix-2 Modified Booth Algorithm", Very Large Scale Integration (VLSI) Systems, IEEE Transactions, vol.18, pp.: 201-208, 04 Feb. 2010
- [2] Prasanna Raj P, Rao, Ravi, "VLSI Design and Analysis of Multipliers for Low Power", Intelligent Information Hiding and Multimedia Signal Processing, Fifth International Conference, pp.: 1354-1357, Sept. 2009
- [3] Lakshmanan, Masuri Othman and Mohamad Alauddin Mohd.Ali, "High Performance Parallel Multiplier using Wallace -Booth Algorithm", Semiconductor Electronics, IEEE International Conference, pp.: 433-436, Dec. 2002.
- [4] Jan M Rabaey, "Digital Integrated Circuits, A Design Perspective", Prentice Hall, Dec.1995
- [5] Louis P. Rubinfield, "A Proof of the Modified Booth's Algorithm for Multiplication", Computers, IEEE Transactions, vol.24, pp.: 1014-1015, Oct. 1975
- [6] Rajendra Katti, "A Modified Booth Algorithm for High Radix Fixedpoint Multiplication", Very Large Scale Integration (VLSI) Systems, IEEE Transactions, vol. 2, pp.: 522-524, Dec. 1994.
- [7] C. S. Wallace, "A Suggestion for a Fast Multiplier", Electronic Computers, IEEE Transactions, vol.13, Page(s): 14-17, Feb. 1964
- [8] Hussin Retal, "An Efficient Modified Booth Multiplier Architecture", IEEE International Conference, pp.:1-4, 2008