

# Modified Booth Multiplier using Wallace Structure and Efficient Carry Select Adder

Neeta Sharma  
ME (DI)

Electronics & Instrumentation Department  
IET DAVV, Indore

Ravi Sindal, PhD.  
Associate Professor

Electronics & Instrumentation Department  
IET DAVV, Indore

## ABSTRACT

The multiplier forms the core of systems such as FIR filters, Digital Signal Processors and Microprocessors etc. This paper presents a model of two different 16X16 bit multipliers. First is Radix-4 Multiplier with SQR Carry Select Adder and Second one is Radix -4 multiplier with Modified SQR Carry Select Adder. Modified Booth Algorithm is used for Partial Products Generation. Wallace Tree Structure is used to accumulate partial Products. SQR Carry Select Adder is used for addition of last two rows. SQR Carry Select Adder uses non uniform block size & Modified SQR Carry Select Adder uses non-uniform block size with Binary to excess one Converter. Both the adders proved to be fast as compared to regular carry select adder. An efficient VHDL code has been written & successfully synthesized & Simulated using Xilinx ISE 12.1. Simulation results shows that the delay of both the multipliers is reduced & the number of logic levels is also reduced with slight increase in number of slices & LUTS as compared to multiplier that uses regular carry select adder.

## Keywords

VHDL, Booth Encoder, Carry Save Adder, Wallace tree, SQR Carry Select Adder, BEC.

## 1. INTRODUCTION

Many application systems based on DSP require extremely fast processing of a huge amount of digital data. The multiplier is an essential element of the digital signal processing such as filtering and convolution. The multiplier is also an important element in microprocessor. The demand of fast processors is increasing for high-speed data processing [6]. Since the multiplier requires the longest delay among the basic operational blocks in digital system, the critical path is determined more by the multiplier [4].

Any multiplier can be divided into three stages: Partial products generation stage, partial products addition stage, and the final addition stage. The speed of multiplication can be increased by reducing the number of partial products. Many high-performance algorithms and architectures have been proposed to accelerate multiplication. Various multiplication algorithms such as Booth, Modified Booth, Braun, and Baugh-Wooley have been proposed [9]. The modified Booth algorithm reduces the number of partial products to be generated and is known as the fastest multiplication algorithm. Wallace Tree Carry Save Adder structures have been used to sum the partial products in reduced time. Our goal is to reduce computation time by using Booth's algorithm for multiplication and to reduce chip area by using Carry Save

Adders arranged in a Wallace tree structure. In the last stage, the two-row outputs of the tree are added using any high-speed adder. Ripple Carry Adder (RCAs) have the most compact design among all types of adders. Carry Look Ahead Adders (CLAs) are the fastest adders, but they are worst from the area point of view. Carry select adders have been considered as a compromise solution between RCAs & CLAs because they offer a good tradeoff between the compact area of RCAs and the short delays of CLAs. CSLAs designs can be further modified to get better performance. CSLA with non uniform block size & CSLA with non uniform block size & Binary to Excess-1 Converter (BEC) will achieve lower delay as compared to regular CSLA.

The paper is organized as follows: section 2 describes Partial Product Generation, section 3 discusses Carry save adders used in Wallace tree structures, section 4 discusses Final Adder that is Efficient Carry Select Adders, section 5 shows simulation results and comparison and section 6 shows the Conclusion, section 7 gives the references.

Multiplier Multiplicand

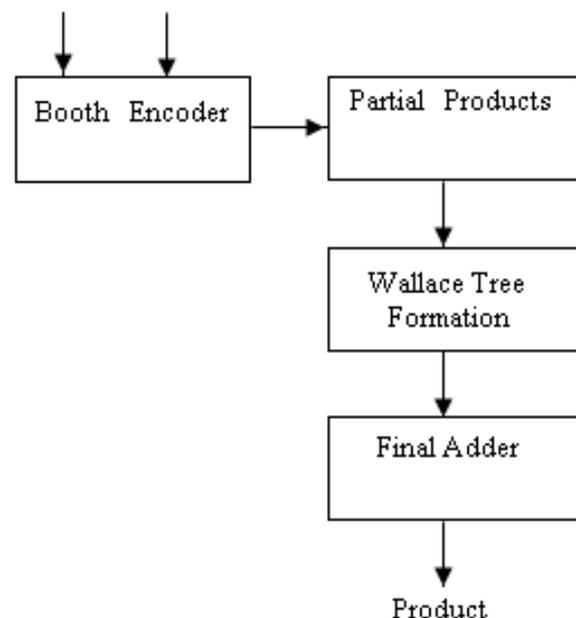


Fig.1. Block Diagram of Wallace Booth Multiplier

## 2. BOOTH ENCODER

Booths algorithm involves encoding of multiplier bits and partial product generation. Different modified booths algorithms have been proposed according to how many number of bits are used to encode multiplier. Modified booth algorithm reduces the number of partial products. Radix 2,

radix 4, radix 8, radix 16, radix 32 are the different modified booth algorithms.  $N \times N$  bit multiplication involves  $N$  partial products but modified Radix  $2^r$  produces  $N/r$  partial products. In the proposed modified Radix -4 Booth Algorithm, multiplier has been divided in groups of 3 bits and each group of 3 bits have been considered according to modified Booth Algorithm for generation of partial products  $0, \pm 1A, \pm 2A$ . In first group, first bit is taken zero and other bits are least significant two bit of multiplier operand. In second group, first bit is most significant bit of first group and other bits are next two bit of multiplier operand. In third group, first bit is most significant bit of second group and other bits are next two bit of multiplier operand. This process is carried on. Partial product is generated using multiplicand operand A. For  $n$  bit multiplier there is  $n/2$  or  $[n/2 + 1]$  groups and partial products. For  $16 \times 16$  bit multiplication 8 partial products are generated. So it reduces the number of partial products in comparison to Booth algorithm (radix-2) improves the computational efficiency of multiplier, reduce the calculation delay.

### 3. WALLACE TREE STRUCTURE

The Wallace tree method [8] is used in high speed designs in order to produce two rows of partial products that can be added in the last stage. Half adder, full adder, unit adder and carry save adders are employed in Wallace tree structure to reduce partial products. Wallace tree accelerate the accumulation of the partial products. The speed, area and power consumption of the multipliers will be in direct proportion to the efficiency of the compressor. The structure is made of carry save adders. The carry save adders reduce the number of partial products and sum rows [7]. Compressors are arithmetic components, similar in principle to parallel adders. The 4:2 compressor [11] shown in Fig.2 has 4 input bits and produces 2 sum output bits (out0 and out1), it also has a carry-in ( $c_{in}$ ) and a carry-out ( $c_{out}$ ) bit (thus, the total number of input/output bits are 5 and 3).

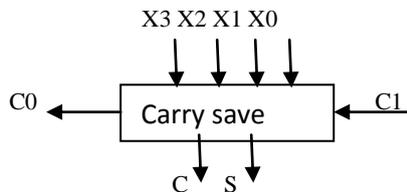


Fig.2. Carry save Adder, 4:2 Compressor

The Carry save adder structure for eight partial products is shown in Fig.3. 16 bit carry save adder blocks is used. The first row adds partial products as  $P_4, P_3, P_2$  and  $P_1$  and the partial products  $P_8, P_7, P_6$  and  $P_5$ . The second row adds the sum & carries outputs from the first row.

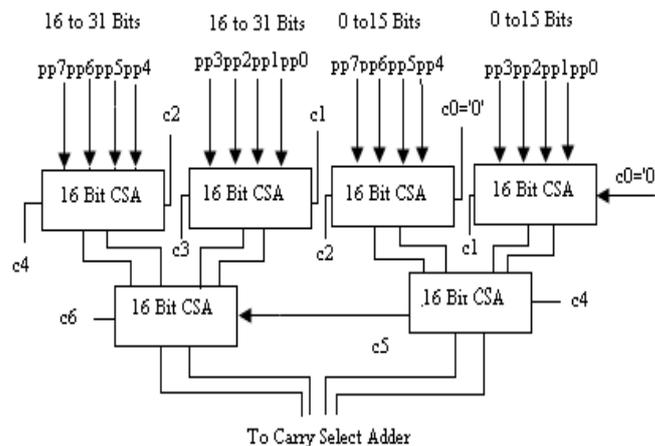


Fig.3. Wallace Tree Structure

### 4.FINAL ADDER

The CSLA is used in many computational systems to moderate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum. However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input and then the final sum and carry are selected by the multiplexers (mux). Regular carry select adder is formed by using uniform block composed of full adders for both the carry inputs viz  $c_{in}=0$  &  $c_{in}=1$  and mux to select correct sum output. A 16-bit carry select adder with a uniform block size has the delay of four full adder delays and three MUX delays. A 16-bit carry select adder shown in fig 4 with variable block size has the delay of two full adder delays, and four mux delays [3]. Therefore we use carry select adder with variable block size. 32 bit SQRT CSLA [5] as final adder enhance speed of multiplier. SQRT CSLA is further modified to get better performance. It is similar to regular 32-bit SQRT CSLA. In Modified SQRT CSLA one RCA fed with constant 1 carry in of Basic block is replaced by Binary to one Converter (BEC) [2]. The BEC is designed by using NOT, XOR and AND gates. Four Bit BEC is shown in Fig.5.

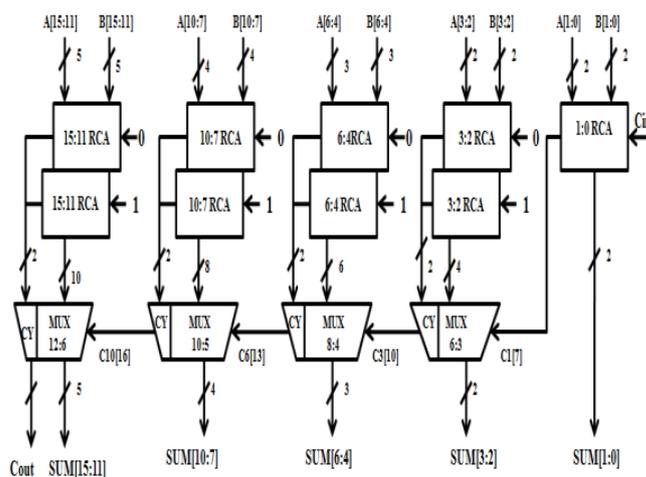


Fig.4. 16 Bit SQRT Carry Select Adder

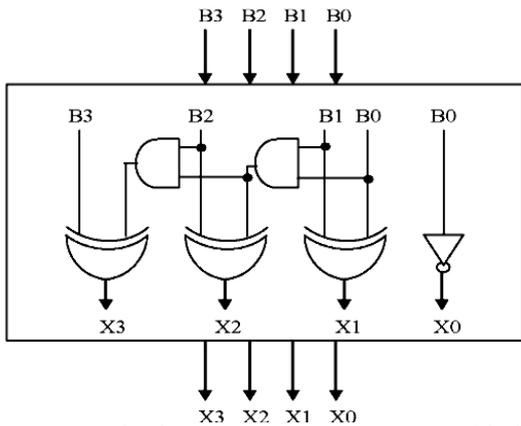


Fig.5. 4 Bit Binary to excess one converter (BEC)

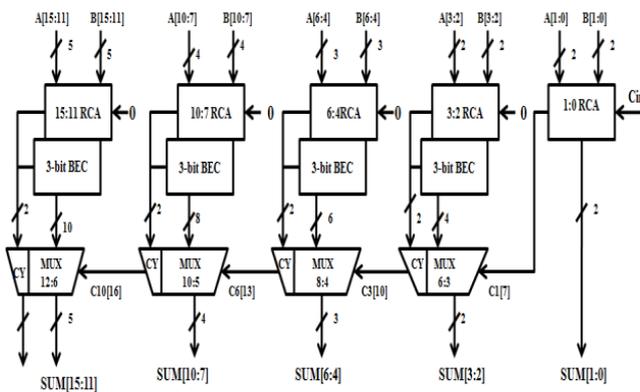


Fig.6. 16 Bit Modified Sqrt Carry Select Adder

## 5. SIMULATION RESULTS

The simulation results of the Modified Booth Multiplier with Sqrt Carry Select Adder are shown in fig. 7, 8 and 9. & simulation results of the Modified Booth Multiplier with modified Sqrt Carry Select Adder are shown in Fig. 10, 11 and 12. Table 1 shows the comparison of Modified Booth Multiplier using Sqrt CSLA & using modified Sqrt CSLA with MBM using regular CSLA in terms of area, speed & levels of logic.

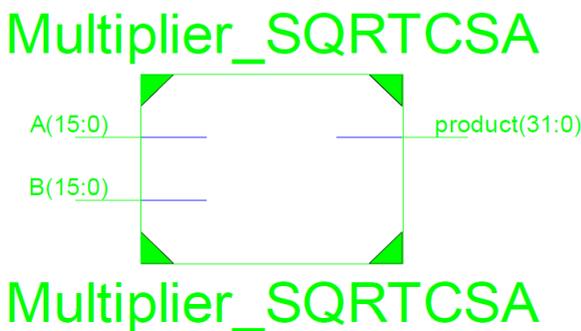


Fig.7. RTL View of multiplier using Sqrt CSLA.

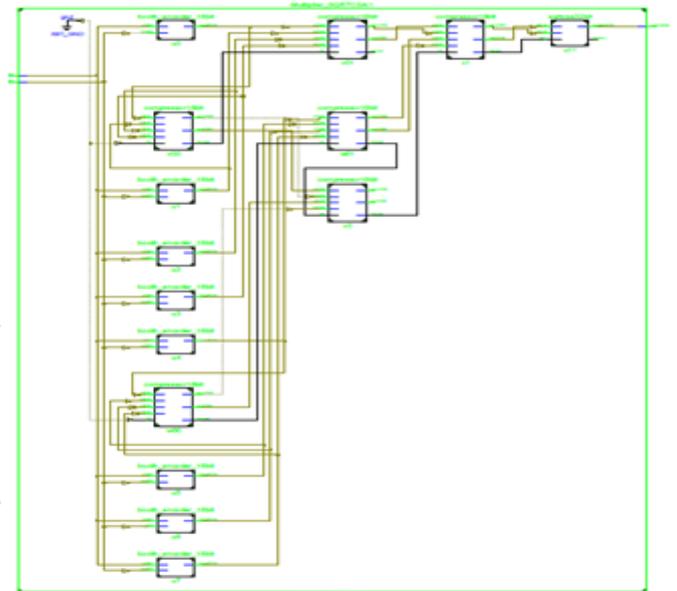


Fig.8 Internal view of multiplier using Sqrt CSLA

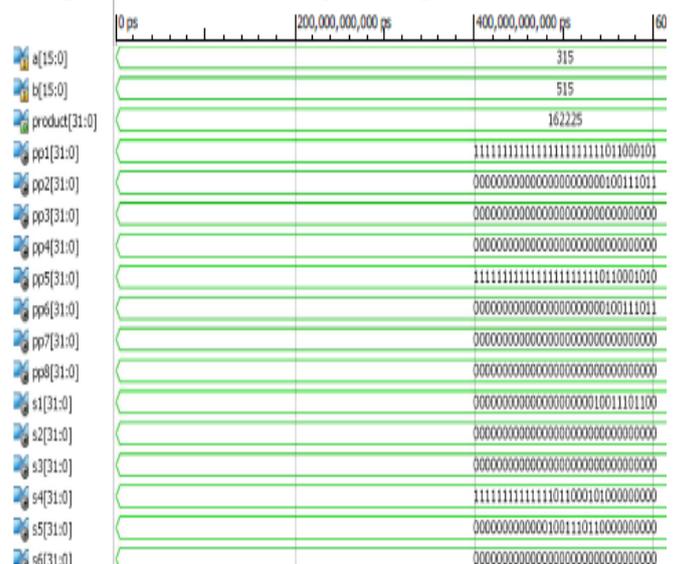


Fig.9. Simulation of multiplier using Sqrt CSLA.

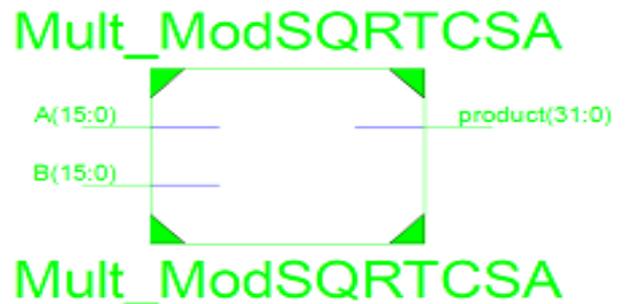


Fig.10. RTL View of multiplier using Modified Sqrt CSLA.

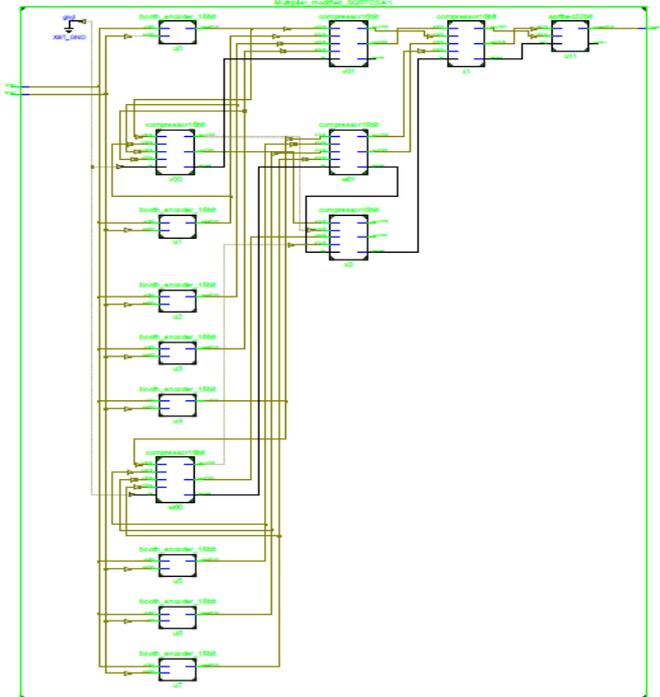


Fig. 11. Internal view of multiplier using Sqrt CSLA

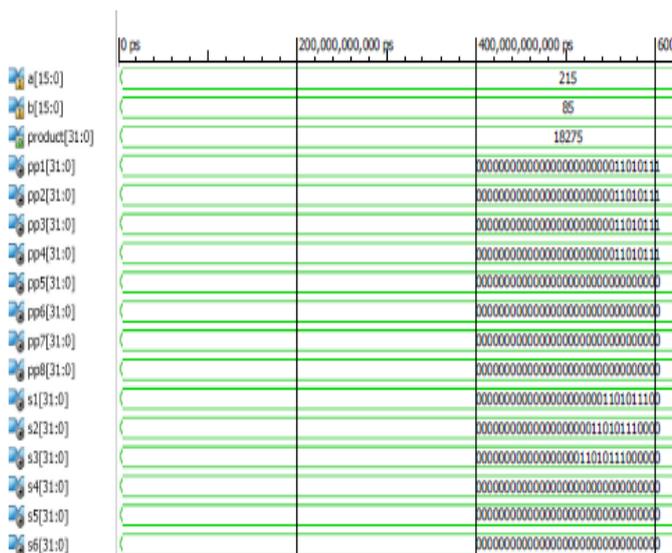


Fig.12. Simulation of multiplier using Modified Sqrt CSLA.

Table1. Comparison of Multipliers

Logic Utilization	Multiplier using CSA	Multiplier using SqrtCSA	Multiplier using Modified SqrtCSA
Number of Slices	669	678	676
Number of LUTS	1272	1287	1281
Levels of Logic	26	23	22
Combinational Path Delay	34.023ns	32.672ns	33.852ns

## 6.CONCLUSION

In this paper, the design & implementation of two high speed multipliers is proposed. The first multiplier makes use of radix-4 booth algorithm with 4:2 compressors & Sqrt CSLA. The second Multiplier uses Radix-4 booth algorithm with 4:2 compressor & modified Sqrt CSLA.

Both the Multipliers show reduction in delay and levels of logic with slight increase in area. The first multiplier shows more reduction in delay. The second multiplier shows more reduction in levels of logic.

We can extend this work by employing Radix-8 Booth algorithm for partial products generation. Expected outcome is less number of partial products, reduced area & reduced delay.

## 7.REFERENCES

- [1] Kulvir Singh , Dilip Kumar “Modified Booth Multiplier with Carry Select Adder using 3-stage Pipelining Technique” International Journal of Computer Applications (0975 – 8887) Volume 44– No14, April 2012
- [2] A. Andamuthu, S. Rithanyaa.” Design Of 128 Bit Low Power and Area Efficient Carry Select Adder” International Journal of Advanced Research in Engineering (IJARE) Vol. 1, Issue 1,2012 Page 31-34
- [3] Aparna P R, Nisha Thomas” Design and Implementation of a High Performance Multiplier using HDL ”
- [4] 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
- [5] Kim, Y. and Kim, L.-S., (May2001), “64-bit carry-select adder with reduced area, “Electron Lett., vol. 37, no. 10, pp. 614–615.
- [6] V. Oklobdzija, “High-Speed VLSI Arithmetic Units: Adders and Multipliers in Design of High-Performance Microprocessor Circuits”, Book Chapter, Book edited by A Chandrakasan, IEEE Press, 2000.
- [7] J. Fadavi-Ardekani, “M × N booth encoded multiplier generator using optimized Wallace trees”, IEEE Transaction on Very Large Scale Integration (VLSI) System, vol. 1, pp. 120–125, 1993.
- [8] C. S. Wallace, “A Suggestion for a Fast Multiplier”, Electronic Computers, IEEE Transactions, vol.13, Page(s): 14-17, Feb. 1964
- [9] Neil H.E Weste, David Harris, Ayan Banerjee, “CMOS VLSI DESIGN”,PEARSON Education, New Delhi,2004
- [10] J. Bhasker, “A VHDL PRIMER” PEARSON Education, New Delhi,2006
- [11] Behrooz Parhami, "Computer Arithmetic", Oxford Press, 2000, pp131