International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

High Performance Multiplier using Booth Algorithm

Snehal R Deshmukh[#1]
Dept of E&TC
SSGMCOE
Shegaon, India (MS)

Prof. Dinkar L Bhombe[#2]
Dept of E&TC
SSGMCOE
Shegaon, India (MS)

Abstract - This paper presents an implementation of a high performance parallel multiplier which is area efficient. Radix-8 Booth multiplier with 3:2 compressors and with 4:2 compressors are presented here. The design is structured for m × n multiplication where m and n can reach up to 126 bits. Carry Look ahead Adder is used as the final adder to accelerate the speed of operation. Finally the performance improvement of the proposed multipliers is validated by implementing a higher order FIR filter.

Keywords – FPGA, HDL, Carry Look ahead Adder, Carry Save Adder, Wallace Tree, Booth Encoding.

I. INTRODUCTION

With the great advancement in multimedia and communication systems, the demand of real-time signal processing and large capacity data processing is increasing. The multiplier is an crucial 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). As they include repetitive application of multiplication and addition, their speed becomes a crucial factor which defines the performance of the total calculation. Since the multiplier requires the largest delay among the basic operational blocks in digital system, the critical path is determined more by the multiplier [1]. Furthermore, multiplier consumes much area and dissipates more power. Hence designing multipliers which offer either high speed, low power consumption [2] or less area or even a combination of them is of great research interest.

Multiplication operation includes generation of partial products and accumulation of all the partial products. The speed of multiplication can be increased by deducting the number of partial products and/or speeding up the accumulation of partial products. Among the various implementation methods high speed parallel multipliers, there are two basic approaches namely Booth algorithm and Wallace Tree compressors. The paper determines an efficient implementation of a high speed parallel multiplier using Booth algorithm and Wallace Tree compressors. The multiplier uses the Radix-8 Booth algorithm with 4:2 compressors. The design is structured for m x n multiplication where m and n can reach up to 126 bits. The Wallace tree uses Carry Save Adders (CSA) to accumulate the partial products. This reduces the time as well as the chip area. To further enhance the speed of operation, carry-look-ahead (CLA) adder is used as the final adder [3].

II. ARCHITECTURE

The architecture of the proposed multiplier is shown in Fig.1. It contains four major modules: Booth encoder, partial product generator, Wallace tree and carry look-ahead adder [4]. The Booth encoder performs Radix-8 encoding of the multiplier bits. Partial products are generated by the generator depending on the multiplicand and the encoded multiplier. For large multipliers of 32 bits, the performance of the modified Booth algorithm is limited. So Booth recoding together with Wallace tree structures have been used in the proposed fast multiplier. The partial products are applied to Wallace Tree and added correctly. A Carry Look-ahead Adder (CLA) to used to add the results from Wallace Tree to get the final product.

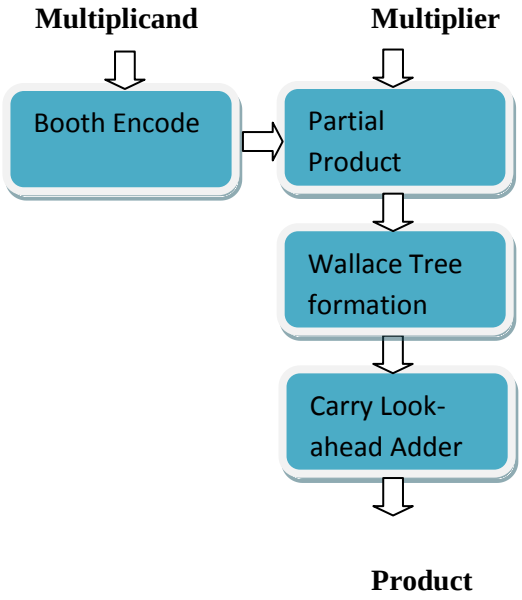


Figure1. Block Diagram of Wallace Booth Multiplier

A. Radix-8 Booth Algorithm

Booth algorithm is a powerful algorithm [5] for signed number multiplication, which treats both positive and negative numbers uniformly. Since a k-bit binary number can be interpreted as a k/3-digit Radix-8 number and so on, it can deal with more than one bit of the multiplier in every cycle by using high radix multiplication [6].

Here we consider the multiplier bits in blocks of four such that each block overlaps the previous block by one bit. Block of four bits is called quartet. Each quartet is codified as a signed digit using Table II. It is advantageous to begin the examination of the multiplier with the least significant bit. The overlap is essential so that we know what happened in the last block, as the most significant bit of the block acts like a sign bit. Radix-8 algorithm reduces the number of partial products to n/3, where n is the number of multiplier bits. Thus it allows a time gain in the partial products summation.

B. Wallace Tree

The Wallace tree method [7] is used in high speed designs to generate two rows of partial products that can be added in the last stage. Also critical path and the number of adders get reduced when compared to the conventional parallel adders. Here the Wallace tree accelerates the accumulation of the partial products. Its advantage seems to be more effective for multipliers of greater than 16 bits .The speed, area and power consumption of the multipliers will be directly proportion to the efficiency of the compressors. The Wallace tree structure with 4:2

compressors is shown in Fig.2. Thus we can expect a significant reduction in computing multiplications.

Table I. Radix-8 Booth Recoding

Multiplier Bits				Recoded Operation on multiplicand, X
Y_{i+2}	Y_{i+1}	Y_i	Y_{i-1}	
0	0	0	0	0X
0	0	0	1	+X
0	0	1	0	+X
0	0	1	1	+2X
0	1	0	0	+2X
0	1	0	1	+3X
0	1	1	0	+3X
0	1	1	1	+4X
1	0	0	0	-4X
1	0	0	1	-3X
1	0	1	0	-3X
1	0	1	1	-2X
1	1	0	0	-2X
1	1	0	1	-1X
1	1	1	0	-1X
1	1	1	1	0X

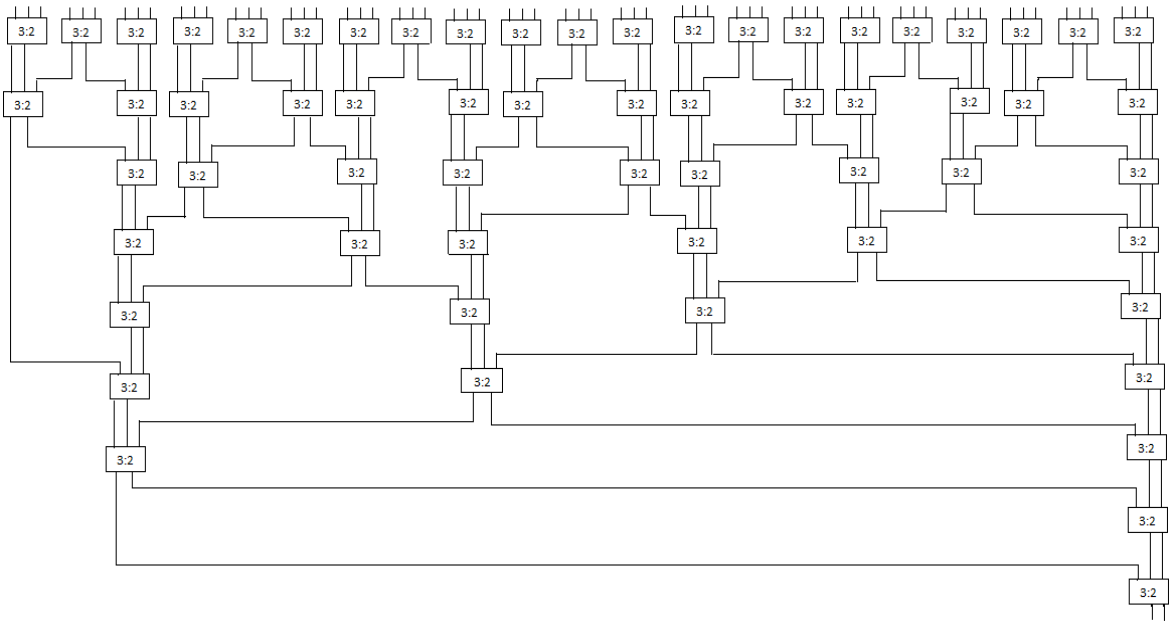


Figure 2.Wallace Tree using 3:2 compressors

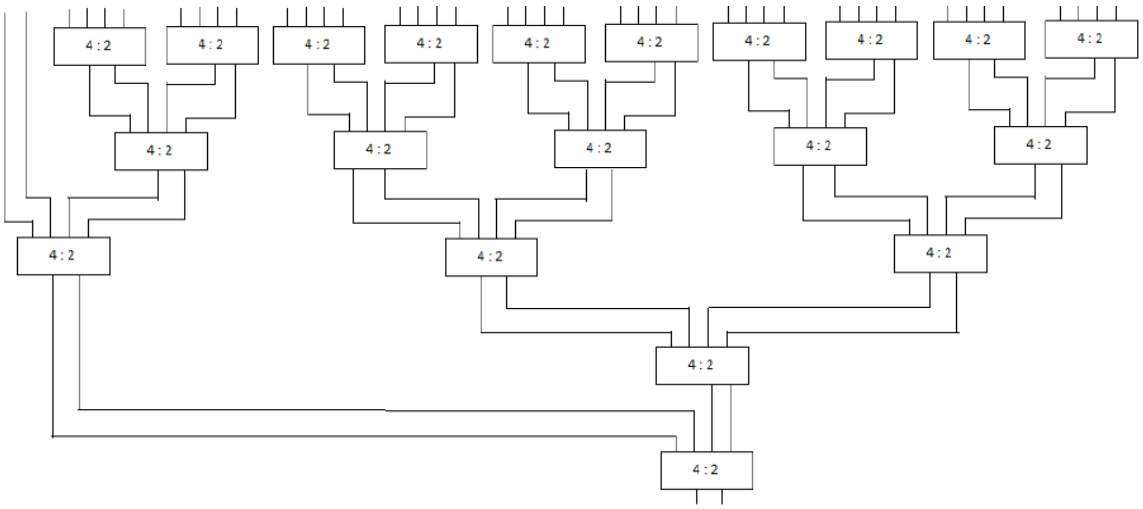


Figure 3.Wallace Tree using 4:2 compressors

A 4:2compressor can be built using two 3:2 compressors. The 3:2 compressors make use of a carry save adder .The carry save adder outputs two numbers of the same dimensions as the inputs, one is a sequence of partial sum bits and other is a sequence of carry bits. In carry save adder, the carry digit is taken from the right and passed on to the left, similar to as it is in conventional addition; but the carry digit passed to the left is the outcome of the previous calculation and not the current one. So in each clock cycle, carries only have to move one step along and the clock can run much faster. Also the carry-save adder generates all of its output values in parallel, and hence has the same delay as a single full-adder. The 4:2 compressors [8] have been widely used in the high speed multipliers to lower the latency of the partial product accumulation stage. The 4:2 compressors are ideally suited for the construction of regularly structured Wallace Tree with low complexity due to its regular interconnection. The number of levels in the Wallace tree using 3:2 compressors can be approximately given as

Number of levels = $\frac{\log (k/2)}{\log (3/2)}$

where k is the number of partial products.

The final results generated at the output of the Wallace tree are added using a Carry Look-ahead Adder (CLA) which is independent of the number of bits of the two operands. In Carry Look-ahead Adder,

for every bit the carry and sum outputs are independent of the previous bits and thus the rippling effect has completely been removed. It operates by creating two signals, propagate and generate for each bit position, based on whether a carry is propagated through from a less significant bit position, a carry is generated in that bit position, or if a carry is killed in that bit position.

III. IMPLEMENTATION RESULTS

The design entry multipliers using Radix-8 Booth algorithm with 3:2 compressors and with 4:2 compressors are done using VHDL and simulated using ISE simulator .It is then synthesized and implemented in a Xilinx XC2S100 FPGA using the Xilinx ISE 9.1i design suite. Table II and Table III summarizes the FPGA resource utilization of these multipliers using 3:2 compressor and using 4:2 compressor.

Table II : Radix 8 Booth Multiplier with 3:2 Compressor

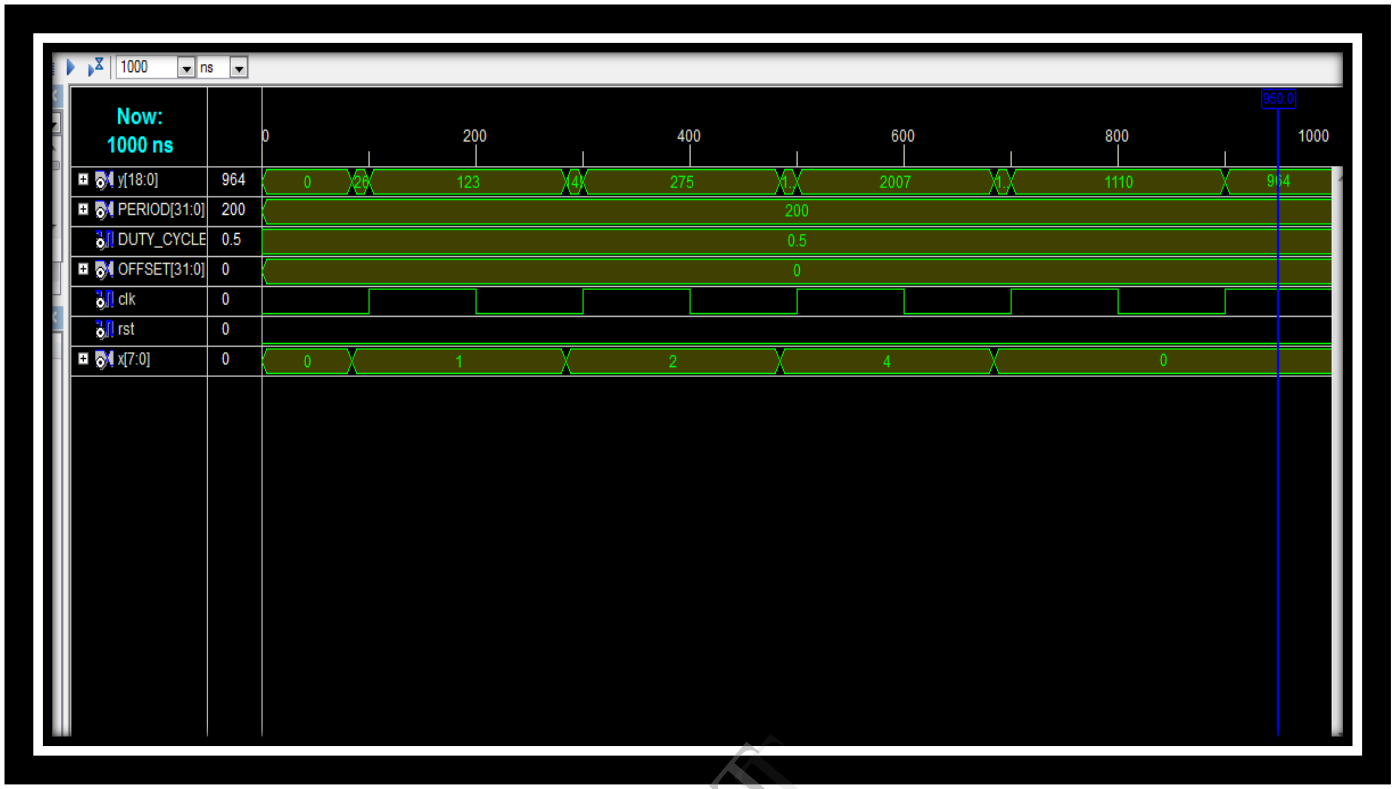
Number of slices	87
Number of 4 input LUTs	170
Number of bounded Inputs	31
Number of bounded Outputs	31

Table III : Radix 8 Booth Multiplier with 4 :2 Compressor

Number of slices	77
Number of 4 input LUTs	14931
Number of bounded Inputs	31
Number of bounded Outputs	31

Finally the performance improvement is validated by implementing a FIR filter using these multipliers. This shows that the multiplier using Radix-8 Booth multiplier with 4:2 compressors gives better speed and the number of occupied slices is lower for the multiplier using Radix-8 Booth algorithm with 3:2 compressors. The FIR filters are implemented in Xilinx XC2S100FPGA.

MULTIPLIER OTPUT



IV. CONCLUSION

In this paper, the design and implementation of two high performance parallel multipliers is proposed. The multiplier makes use of the Radix-8 Booth Algorithm with 3:2 compressors and with 4:2 compressors. Both the designs were implemented on Spartan 2 FPGA. The multiplier using Radix-8 Booth algorithm with 4:2 compressors are found to be faster than the other. Also the use of Radix- 8 Booth multiplier with 4:2 compressors for a higher order FIR filter showed a dramatic speed improvement .

V. 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, Feb. 1964
8. Hussin R et al , “An Efficient Modified Booth Multiplier Architecture” IEEE International Conference 2008.