International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 7, July 2014)

High Performance Radix-8 Multiplier Using Asynchronous Counters

EASARI. PARUSHA RAMU[1]
Assistant professor
Sri Indu College of Engineering and Technology,
Shariguda(V), Ibrahimpatnam(M) RR Dist. Hyderabad
parushuece@gmail.com

TABASUM.GULEDGUDD [2]
Assistant professor
SECAB Institute of Engineering & Technology. Bagalkota road, Bijapur Dist. Karnataka.
tabuag85@gmail.com

***Abstract* — This paper presents a neighborhood economical implementation of a high performance parallel multiplier factor. Radix-4 Booth multiplier factor with 3:2 compressors, number-8 Booth multiplier factor with 4:2 compressors and Radix -8 Booth multiplier factor with asynchronous counters square measure bestowed here. the look is structured for m × n multiplication wherever m and n will reach up to 126 bits. Carry Look ahead Adder is employed because the final adder to reinforce the speed of operation. Finally the performance improvement of the planned multipliers is valid by implementing the next order FIR filter. the look entry is completed in VHDL and simulated exploitation ISE thirteen.2 style suite from Xilinx. it's then synthesized and enforced exploitation Xilinx ISE thirteen.2 targeted towards Spartan 3E FPGA.**

Keywords- FPGA; HDL; Carry Look ahead Adder; Carry Save Adder; Wallace Tree; Booth Encoding; asynchronous counters.

I. INTRODUCTION

With the fast advances in multimedia system and communication systems, time period signal process and enormous capability processing square measure more and more are being demanded. The multiplier factor is a vital part of the digital signal process like filtering and convolution. Most digital signal process strategies use nonlinear functions like separate cos remodel (DCT) or separate ripple remodel (DWT). As they're essentially accomplished by repetitive application of multiplication and addition, their speed becomes a serious issue that determines the performance of the complete calculation. Since the multiplier factor needs the longest delay among the fundamental operational blocks in digital system, the important path is set a lot of by the multiplier factor. What is more, multiplier factor consumes a lot of space and dissipates a lot of power. Thence coming up with multipliers which provide either of the subsequent style targets – high speed, low power consumption, less space or perhaps a mixture of them is of considerable analysis interest.
Multiplication operation involves generation of partial merchandise and their accumulation. The speeds of multiplication are often exaggerated by reducing the quantity of partial merchandise and/or fast the buildup of partial merchandise.

Among the numerous strategies of implementing high speed parallel multipliers, there square measure 2 basic approaches specifically Booth algorithmic program and Wallace Tree compressors. This paper describes AN economical implementation of a high speed parallel multiplier factor exploitation each these approaches. Here 3 multipliers square measure projected. the primary multiplier factor makes use of the Radix-4 Booth algorithmic program with 3:2 compressors whereas the second multiplier factor uses the Radix-8 Booth algorithmic program with 4:2compressors and third multiplier factor uses the Radix-8 Booth algorithmic program with asynchronous counters. the look is structured for m x n multiplication wherever m and n will reach up to 126 bits. the quantity of partial merchandise is n/2 in Radix-4 Booth algorithmic program whereas it gets reduced to n/3 in Radix-8 Booth algorithmic program. The Wallace tree uses Carry Save Adders (CSA) to accumulate the partial merchandise. This reduces the time furthermore because the chip space. To any enhance the speed of operation, carry-look-ahead (CLA) adder is employed because the final adder.

II. ARCHITECTURE

The design of the planned number is shown in Fig.1. It consists of 4 major modules: Booth encoder, partial product generator, Wallace tree and carry look-ahead adder. The Booth encoder performs Radix-2 or Radix-4 encryption of the number bits. Supported the number and therefore the encoded number, partial merchandise ar generated by the generator. for big multipliers of thirty two bits, the performance of the changed Booth formula is restricted. thus Booth cryptography in conjunction with Wallace tree structures are utilized in the planned quick number. The partial merchandise ar provided to Wallace Tree and accessorial fittingly. The results ar finally accessorial employs a Carry Look-ahead Adder (CLA) to urge the ultimate product.

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 7, July 2014)

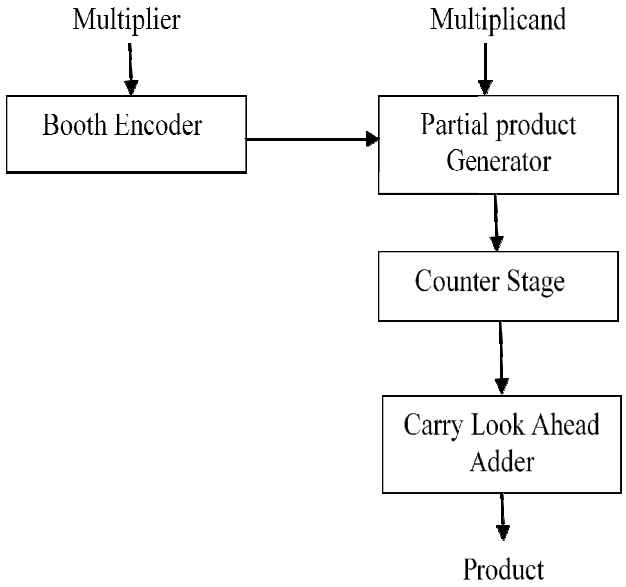

Figure 1. Block diagram of Wallace Booth Multiplier

A. *Radix – 4 Booth Algorithm*

Booth formula may be a powerful formula for signed variety multiplication, that treats each positive and negative numbers uniformly. Since a k-bit binary variety will be understood as k/2-digit Radix-4 variety, a k/3-digit Radix-8 variety so on, it will touch upon over one little bit of the number in every cycle by mistreatment high base multiplication. The key disadvantage of the Radix-2 formula was that the method needed n shifts associate degreed a median of n/2 additions for an n bit number. This variable variety of shift and add operations is inconvenient for planning parallel multipliers. additionally the Radix-2 formula becomes inefficient once there area unit isolated 1's. The Radix-4 changed Booth formula overcomes of these limitations of Radix-2 formula. For operands up to or larger than sixteen bits, the changed Radix-4 Booth formula has been wide used. it's supported encryption the two's complement number so as to cut back the quantity of partial product to be else to n/2. The number, Y in two's complement kind will be written as in

Y = -Yn-1 2n-1 + _i Yi2i ; 0 _ i_ n-2 (1)
It can be written as
Y = _i (- 2 Y2i+1 + + Y2i +Y2i-1) 22i ; 0 _ i _ n-2 (2)

Table I shows the encoding of the signed multiplier Y, using the Radix-4 Booth algorithm. Radix-4 Booth recoding encodes multiplier bits into [-2, 2]. Here we consider the multiplier bits in blocks of three, such that each block overlaps the previous block by one bit. It is advantageous to begin the examination of the multiplier with the least significant bit. The overlap is necessary so that we know what happened in the last block, as the most significant bit of the block acts like a sign bit.

TABLE I. RADIX - 4 BOOTH RECODING

Multiplier Bits			Recoded Operation on multiplicand, X
0	0	0	0X
0	0	1	+X
0	1	0	+X
0	1	1	+2X
1	0	0	-2X
1	0	1	-X
1	1	0	-X
1	1	1	0X

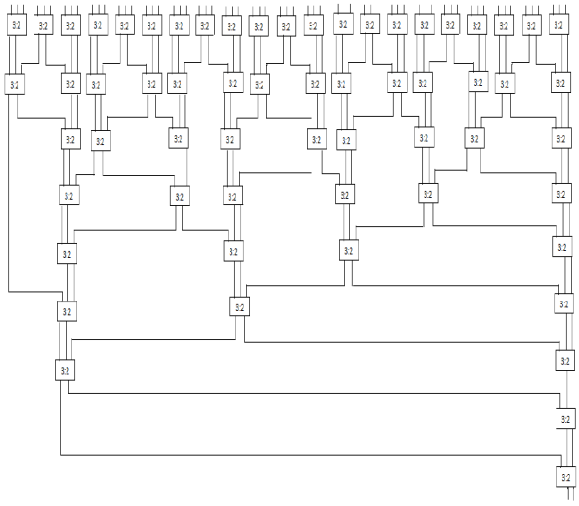

Figure.2: Wallace Tree using 3:2 compressors

B. *Radix 8 Booth Algorithm*

Radix-8 Booth recoding applies the same algorithm as that of Radix-4, but now we take quartets of bits instead of triplets. Each quartet is codified as a signed digit using Table II. 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.

C. *Wallace Tree*

The Wallace tree method is used in high speed designs in order to produce 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 has taken the role of accelerating the accumulation of the partial products. Its advantage becomes more pronounced for multipliers of greater than 16 bits .The speed, area and power consumption of the multipliers will be in direct proportion to the efficiency of the compressors. The Wallace tree structure with 3:2 compressors, 4:2 compressors and asynchronous counters is shown in Fig.2,

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 7, July 2014)

Fig.3 and Fig.4 respectively. In this regard, we can expect a significant reduction in computing multiplications.

TABLE II: 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

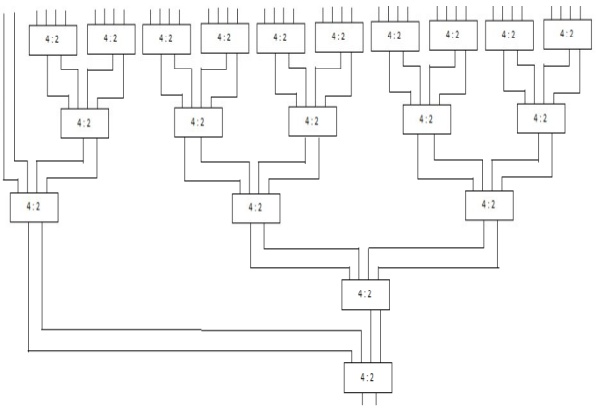

Figure. 3. Wallace Tree using 4:2 compressors

D. proposed counter

In Fig. 4, the counters are used to count the number of 1's in a column. Each of them is a simple DFF-based ripple counter. The clock is provided to the first DFF and all the other DFFs are triggered by the preceding DFF outputs. A typical 7-bit 1's counter is shown in Fig. 4. The clock input is synchronized with the input data rate and thus the operands can be accumulated with a high frequency defined by the setup time and propagation delay of a DFF. Moreover, the counters change states only when the input is "1," which leads to low switching power. This simple and efficient bit accumulation technique is used to design the proposed radix-8 multiplier.

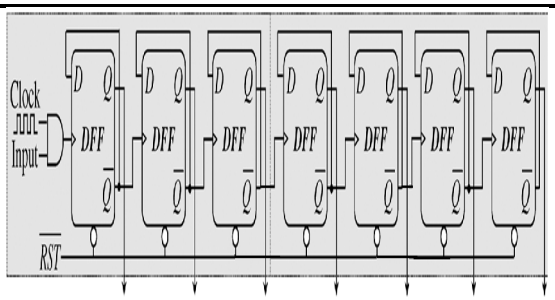

Figure 4.architecture of a 7-bit asynchronous counter.

The final results obtained 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 eliminated. It works 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.

II. IMPLEMENTATION RESULTS

The design entry of 126×126 bit multipliers using Radix-4 Booth algorithm with 3:2 compressors, Radix-8 Booth algorithm with 4:2 compressors and Radix-8 Booth algorithm with asynchronous counter are done using VHDL and simulated using ISE 13.2 design suite from Xilinx. It is then synthesized and implemented in a Xilinx XC3S5000 fg320 -4 FPGA using the Spartan3E kit. Figure 4 presents a snapshot of simulation waveforms for 126×126 bit multiplier. Table IV summarizes the FPGA resource utilization of these three multipliers. Finally the performance improvement is validated by implementing a higher order FIR filter using these multipliers. Table V summarizes the FPGA resource utilization for FIR filters using these multipliers. This shows that the multiplier using Radix-8 Booth multiplier with asynchronous counter gives better speed and the number of occupied slices is lower for the multiplier using Radix-8 Booth algorithm with 4:2 compressors. The FIR filters are implemented in Xilinx XC3S1500fg320-4 FPGA. The specifications of the FIR filter chosen are as follows.

Sampling frequency : 24 KHz
Pass band frequency : 8 KHz
Stop band frequency : 9 KHz
Pass band ripple : 0.1 linear scale
Stop band attenuation : 0.001 linear scale

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 7, July 2014)

TABLE IV: DEVICE UTILIZATION SUMMARY OF MULTIPLIERS

Logic Utilization (XC3S5000 fg1156 -4)	Radix -8 Booth Algorithm with 4:2 compressors	Radix -8 Booth Algorithm with asynchronous counter
Number of four input LUTs	57734	84572
Number of occupied Slices	30779	42408
Number of bonded IOBs	504	506
JTAG Gate Count for IOBs	24,144	24,251
Maximum combinational path delay (ns)	381.069	315.023
Average Connection Delay (ns)	4.781	6.058

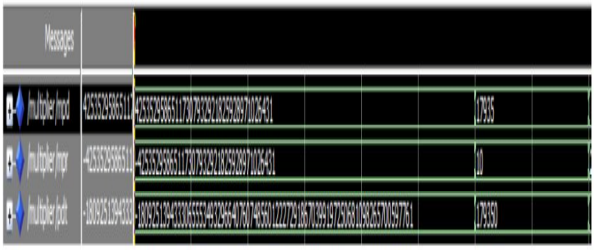

Figure. 5: Simulation wave forms

III. CONCLUSION

In this paper, the design and implementation of two high performance parallel multipliers is proposed. The first multiplier makes use of the Radix-4 Booth Algorithm with 3:2compressors while the second multiplier uses the Radix-8 Booth algorithm with 4:2 compressors, the third multiplier uses the Radix-8 Booth algorithm with asynchronous counter. Three designs were implemented on Spartan3E FPGA. The multiplier using Radix- 4 Booth algorithm with 3:2 compressors shows more reduction in device utilization as compared to the multipliers using Radix-8 Booth algorithm with 4:2 compressors and Radix-8 Booth algorithm with asynchronous counter. Meanwhile the multiplier using Radix-8 Booth algorithm with asynchronous counter are found to be faster than the other. Also the use of Radix- 8 Booth multiplier with asynchronous counter for a higher order FIR filter showed a dramatic speed improvement than that using Radix-4 Booth multiplier with 3:2 compressors and Radix-8 Booth multiplier with 4:2 compressors.

ACKNOWLEDGMENT

We express our sincere gratitude to Mrs. Kumari Roshni V.S, Centre for Development of Advanced Computing, Thiruvananthapuram for incorporating us as part of this prestigious 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 R et al, "An Efficient Modified Booth Multiplier Architecture", IEEE International Conference, pp.:1-4, 2008.

About the Authors:

Easari.Parusha Ramu, Assistant Professor. Currently working in Sri Indu College of Engineering Technology in ECE dept.

Tabasum. Guledgudd, Assistant Professor. currently working in SECAB Institute of Engineering & Technology in ECE Dept.