A New Approach for Faster and Size Efficient Signed Multiplier of 8bits

Monika Raghav[1], Rakesh Jain[2]

[1]Department of Electronics and Communication, Gyan Vihar University, Jaipur, Rajasthan, India

[2]Assistant Professor, Department of Electronics and Communication, Gyan Vihar University, Jaipur, Rajasthan, India

Abstract: ***Multiplier is one of the essential element for microprocessors, digital signal processors etc. In this paper, we designed and implemented a new high speed 8 bit signed multiplier. This paper presents an efficient implementation of high speed multiplier using the shift and adds method, using Booth algorithm. This proposed architecture is implemented by using the radix-4 booth recoding process. The proposed multiplier reduces the partial product array by almost 1/2th the size of the bits. This reduction in partial product increases the speed of the multiplier. The proposed multiplier is compared with the former method of multipliers, it demonstrated that it is more efficient in the term of delay and area .The proposed multiplier is designed and implemented using Verilog HDL in XILINX 9.2 version. The result of conventional multiplier is compared with proposed multiplier. Experimental results demonstrate that the proposed multiplier has 26.13% faster than conventional signed multiplier and size also reduces 33%. The proposed method can be extended to any higher radix encodings, as well as to any size square and rectangular multipliers.***

Keywords: signed multiplier, Radix-4 multiplier, Radix-8 multiplier, Baugh-Wooley (BW) algorithm, Verilog , FPGA .

1. Introduction

Multiplier is one of the basic hardware block used for many digital and high performance systems such as FIR filters, digital signal processors and microprocessors etc. Many high speed low power multiplication algorithms and architectures have been proposed. Advances in technology have permitted many researchers to design multipliers which offer both high-speed and regularity of layout, thereby making them suitable for VLSI implementation. Digital signal processing requires efficient multiplication operations with the highest possible speed without compromising the power budget. In general, basic multiplication algorithm can be divided into three following steps.1) partial product (pp) generation, 2) partial product reduction and 3) final carry propagated addition [1-2]. In the first step, a set of partial product rows is generated where each one is the result of the product of one bit of the multiplier by multiplicand. For example, if we consider the multiplication X x Y with both X and Y on n bits and of the form xn_1 . . . x0 and yn_1 . . . y0, then the ith row is, in general, a proper left shifting of yi x X, i.e., either a string of all zeros when yi = 0, or the multiplicand X itself when yi = 1. In this case, the number of PP rows generated during the first phase is clearly n [1-4]. Recoding of binary numbers was first hinted at by Booth [5] four decades ago. MacSorley [6] proposed a modification of Booth's algorithm a decade after. Modified booth encoding (MBE) [6] is a technique that has been introduced to reduce the no of pp rows with a maximum height of [n/2] +1 rows. More specifically, Two's complement multiplier [7] using radix-4 MBE generates a pp array with a maximum height of [n/2] rows without any increase of delay, each row of the pp array follows the one of the following possible values: all zeros, +X, +2X [8].This pp reduction may increases the speed of the multiplier. During pp reduction phase, all pp rows are reduced by using compression tree [9-10].Since the intermediate addition values is not important, the outcome of this phase is a result represented in redundant carry save form, i.e., as two rows, which allows for much faster implementations. The final (carry-propagated) addition has the task of adding these two rows and of presenting the final result in a non-redundant form, i.e., as a single row. In this work, we introduce an idea to reduce the pp array with a maximum height of [n/3] rows by using radix-8 booth recoding process. A similar study aimed at the reduction of the maximum height to [n/3] but using a different approach has recently presented interesting results in [11]. Thus, in the following, we will evaluate and compare the proposed approach with the technique in [7].

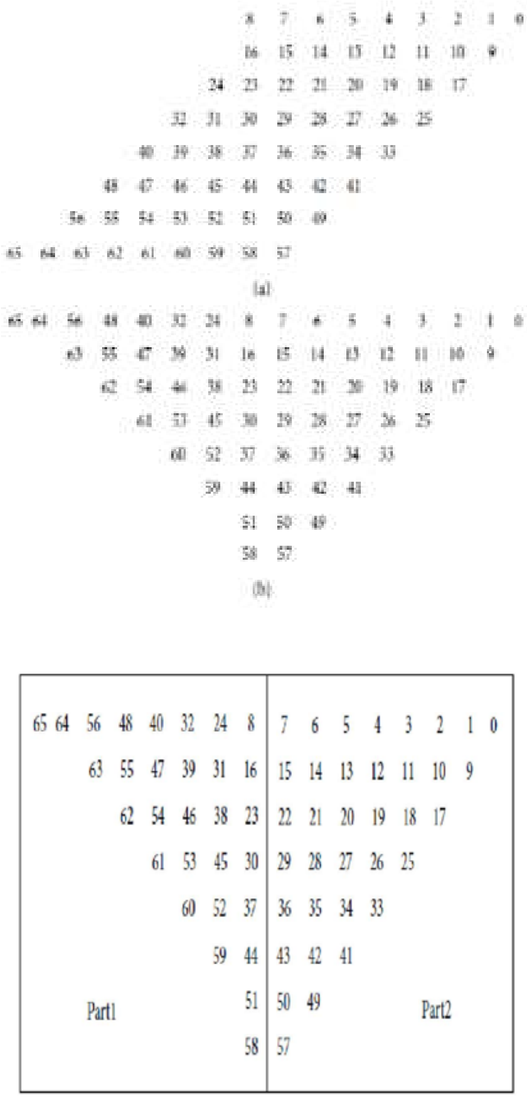
This paper is structured as follows. Sections II describe HPM signed multiplier and Section III describe new signed multiplier. Section IV Summarize the result analyse. Finally section V describe about conclusion of new signed multiplier and section VI future scope. Throughout the paper, it is assumed that the number of bits in the multiplier and multiplicand is equal.

2. HPM Signed Multiplier

The multiplication process begins with the generation of all partial products in parallel using an array of AND gates .The next major steps in the design process are partitioning of the partial products and their reduction process. Each of these steps is elaborated in the following subsections.

2.1 Partitioning the Partial Products

It is consider two n-bit (8-bit) operands of Baugh-Wooley multiplier partial products which form a matrix of n rows and $2n$ columns .Initially for the partial product delay in the PPST. Therefore, in this work, it split up the PPST into two parts as shown in the Figure 1(c), in which both parts share equal number of columns. That is, part0 consists of n columns of Baugh-Wooley multiplication, it assign an integer as shown in Figure 1(a); for example, p00 is given an index 0, p10 an index 1, and so on. For convenience, it rearrange the partial products as shown in Figure 1(b).The two longest columns in the middle of the partial products contribute to the maximum and part1 also consists of n columns then proceed to sum up each column of the two parts in parallel.



(c)

Figure 1: Partitioning the partial products: (a) partial-product array diagram for 8 multiplier, (b) an alternative representation, and (c) partitioned structure of multiplier showing part0 and part1

The two parallel structures based on the HPM method are shown in Figure 2, where HA, FA, p0, p1, and p denote half adder ((2,2) counter)), full adder ((3,2) counter), partial final product frompart0, partial final product from part1, and final product, respectively .The numerals residing on the HA and FA indicate the position of partial products .The outputs of part0 and part1 are computed independently in parallel, and those values are added using a high-speed hybrid final adder to get the final product. However, before we proceed to carry out the final addition with the proposed hybrid adder, it first carry out the final addition with the faster adder of CLA for both the unpartitioned W, D, H Baugh-Wooley multiplier and the partitioned W, D, H Baugh-Wooley multiplier and analyze the effect of partitioning the PPST into two parts. In this it is clear that it give improvement and overhead in delay, area, and power of the partitioned multipliers with respect to the unpartitioned multiplier. It can be seen that there is 4.3%improvement in the speed for 16-b and 11.2% for 64-b size.The speed limitation in lower bit size multipliers is due to the greater difference between input arrival profile to the final CPA from part0 and part1. But with the increase in the word size, this difference becomes lesser and the improvement in the speed of the partitioned multipliers increases. There is maximum of 11%, 8%, and 6% speed improvement for 64-b W, D, H Baugh-Wooley multipliers with 1% area overhead. Having clearly demonstrated the reduction in the delay of the multipliers due to the partitioning of the partial products, we now proceed to further enhance the speed of the proposed multiplier. There is maximum of 6% to 7% power overhead in W, D, H based Baugh- Wooley multiplier, and this is due to the use of CLA as CPA in each part. But this power overhead is interestingly reduced by proposed hybrid CPA which is elaborated in the next section.

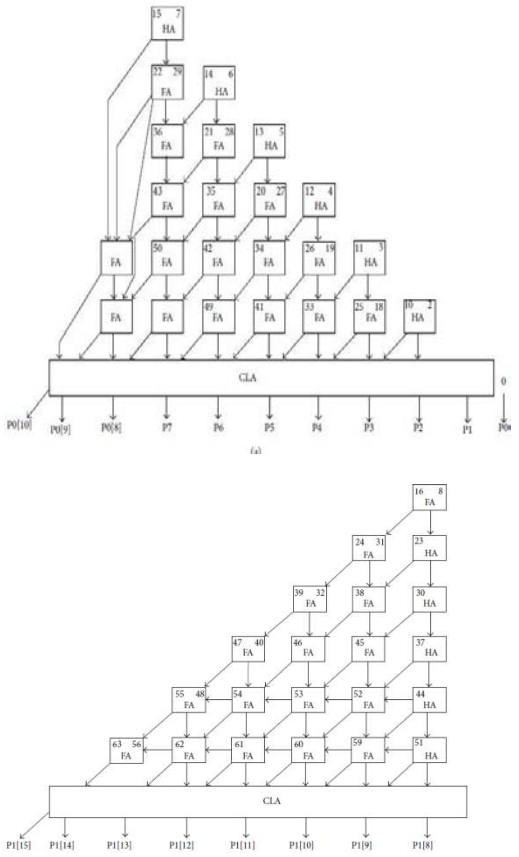


Figure 2: The HPM-based implementation: (a) implementation of part0; (b) implementation of part1.

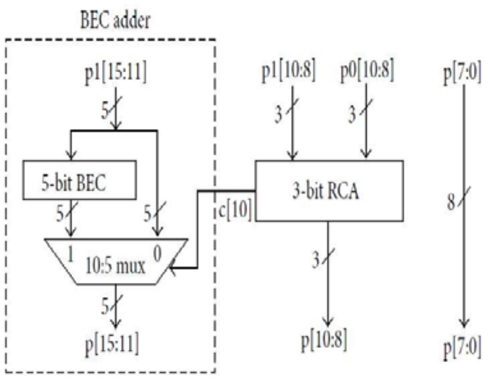


Figure 3: Hybrid final adder of 8-b multiplier

In this work it uses BEC logic for fast summation of uneven input arrival time of the signals originating from the PPST. The BEC adder provides faster performance than carry save adder (CSA) and it consumes less area, low power than the carry select adder (CSLA) [14, 15]. The HPM signed multiplier is outperformed all other previous design in term of delay and area. The HPM signed multiplier also has some limitation because it has more partial product by which it increases the propagation delay and area.

3. Proposed New Signed Multiplier

We have seen in HPM signed multiplier that it has many partial product which is divided into two parts which increases the size of multiplier and reduces the speed. To reduce this limitation we proposed a new architecture of signed multiplier. The basic architecture design of proposed signed multiplier is given in fig 4. The basic idea of new signed multiplier is based on radix 4 booth algorithm.

From the architecture design of proposed signed multiplier it is clear that we have a multiplicand A and a multiplier B of 8 bit on which we apply booth algorithm.
Now we are going to describe the architecture design of proposed signed multiplier.

2's complement: In this block the multiplicand A first perform the operation of 2's complement. 2's complement is a mathematical operation on binary number as well as signed number representation based on this operation. It is the operation in which first we perform 1's complement operation which says that invert all the binary bits means 1 to 0 and 0 to1 and then add 1 then the resultant is known as 2's complement.

bit grouping sequencer: In this operation we add 0 at the right LSB of multiplier and then grouping them in a sequence, here its grouping is done on 3 bits and the last bit of these three bit also includes in the next grouping of 3 bit , By which we Find vector of three bits of multiplier ({ Bn+1, Bn, Bn-1}.

Encoder and Extender:
In this block we have to do two important works which are encoded and extend the bits. So for the encoding we uses following table 1.

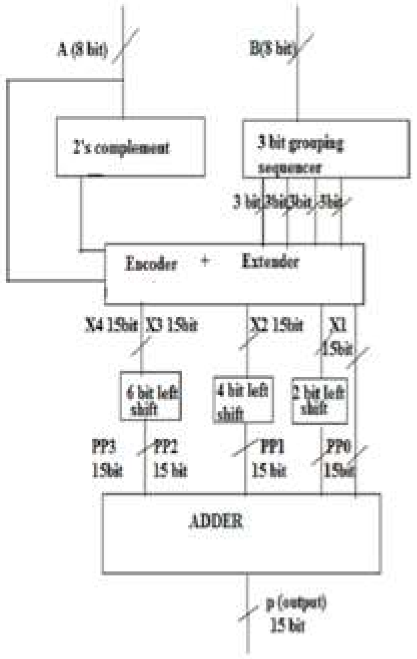


Figure 4: Proposed signed multiplier

Table 1: Radix 4 Encoding Table

Multiplier bit (Bn+1)	*Multiplier bit (Bn)*	*Multiplier bit (Bn-1)*	*Encoded bits (z)*
0	0	0	0*A
0	0	1	1*A
0	1	0	1*A
0	1	1	2*A
1	0	0	-2*A
1	0	1	-1*A
1	1	0	-1*A
1	1	1	0*A

By using that 3 bit vector and given table we find encoded bits z. From an operational point of view, multiply by zero means the multiplicand (A) is multiplied by "0".Multiply by "1" means the product still remains the same as the multiplicand value. Multiply by "-1" means that two's complement of the multiplicand value. Multiply by "2" means just left shift. The multiplicand by one place Multiply by "-2" is to shift left one bit the two's complement of the multiplicand value. So performing these operation we can encoded these bits. Bits which is produces from this operation is known as partial product.

Extender: In the extender operation we have to extend that last bit to 15 bits by it reduces the complexity during the operation..

Adder: In digital electronics adder is a digital circuit that performs addition of numbers, these can be classified in bit adder or multi bit adders. So from architecture design of booth algorithm it is clear that the partial product which is generate after the multiply of bits get add by using adder so from it is clear that the output of this adder is the output of new signed multiplier.

After completing the whole operation we have observe that it has very less partial product so it reduces the size of signed multiplier and increases the speed. In comparison to previous signed multiplier the speed of proposed signed multiplier is more and its area is also decreases. The overall improved performance is elaborated in result summary.

4. Result and Comparison

In this paper we evaluate the performance of base paper HPM multiplier and proposed signed multiplier and implemented them. We synthesize these multiplier using Xilinx ISE 9.2 and use verilog as hardware description language. We design proposed signed multiplier by which we analyze that delay of 8 bit proposed signed multiplier as comparison to the delay of 8 bit HPM multiplier is less and the size is also reduce in proposed signed multiplier. As we can see that the partial product in HPM multiplier is more and proposed signed multiplier have less partial product which results , speed is increased by 26.13% because we have very less partial product in comparison to the HPM multiplier and is also gave area efficient result by reduces the size of proposed sign multiplier by 33% . So it is clear from here that proposed signed multiplier is given area efficient and faster result. The technology schematic of proposed multiplier is shown in figure 6 and the simulation waveform is also shown in figure 5.

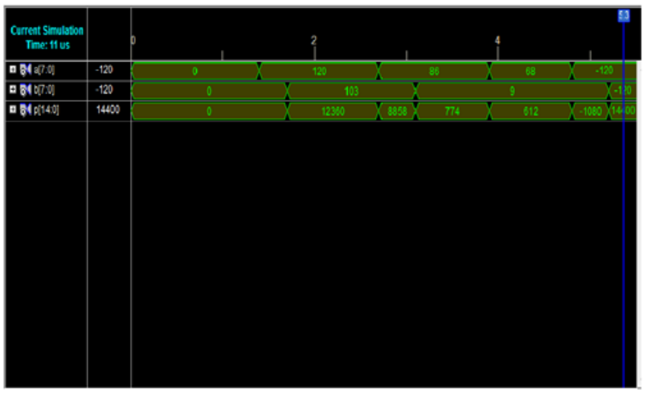


Figure 5: Simulation of proposed multiplier

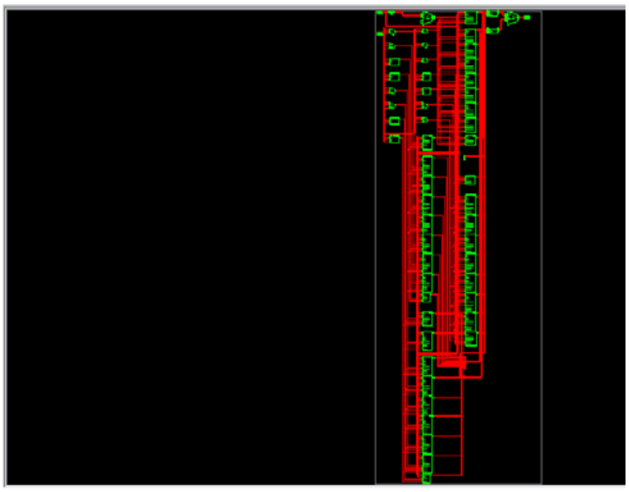


Figure 6: RTL diagram of proposed signed multiplier

Table

Comparison Table

Serial No.	Multipliers	No. of slice	Delay
1	HPM Signed Multiplier	127	10.708 ns
2	Radix 4 signed Multiplier	84	7.909 ns

From above comparison table it is clear that speed is increased by 26.13% and area reduced by 33%.

5. Conclusion

We have successfully achieved faster and size efficient multiplier by using radix 4 booth multiplier with increasing the speed and reducing the area. The proposed 64 bit signed multiplier have delay lower as comparison to the previous HPM signed multiplier and there it is area efficient. We have good reasons to believe that size greater than or equal to 64bit , significant speeds and area can be achieved. Also proved that the proposed techniques improve the performance of signed multiplier. .

6. Future Scope

In future we can use 16 radix booth algorithm for signed multiplier in which the size will reduce and delay will reduce. In 4 radix booth algorithm if size reduces 33% then if we use 16 radix booth algorithm then size will reduce approx 50% and delay will reduce. So it is clear from here that it will give very efficient result in size and speed. In this paper we have shown signed multiplier of 8 bit which is more efficient than previous one but in my paper I will work on radix 8 signed multiplier of 64 bit which is also efficient than HPM signed multiplier.

Reference

[1] K. C. Bickerstaff, *Optimization of Column Compression Multipliers [Ph.D. thesis]*, Department of Electrical and Computer

[2] Engineering, University of Texas at Austin, Austin, Tex, USA, 2007.

[3] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press, New York, NY, USA, 2nd edition, 2010.

[4] C. S.Wallace, "A suggestion for a fast multiplier," *IEEE Transactions on Electronic Computers*, vol. EC-13, pp. 14–17, 1964.

[5] L. Dadda, "Some schemes for parallel multipliers," *Alta Frequenza*, vol. 34, pp. 349–356, 1965.

[6] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach," *IEEE Transactions on Computers*, vol. 45, no. 3, pp. 294–306, 1996.

[7] K. C. Bickerstaff, E. E. Swartzlander, and M. J. Schulte, "Analysis of column compression multipliers," in *Proceedings of the 15th IEEE Symposium on Computer Arithmetic*, pp. 33–39, June 2001.

[8] W. J. Townsend, E. E. Swartzlander, and J. A. Abraham, "A comparison of Dadda and Wallace multiplier delays," in *Advanced Signal Processing Algorithms, Architectures, and Implementations XIII*, vol. 5205 of *Proceedings of the SPIE*, pp. 552–560, August 2003.

[9] H. Eriksson, P. Larsson-Edefors, M. Sheeran, M. Sj¨alander, D. Johansson, andM. Sch¨olin, "Multiplier reduction tree with logarithmic logic depth and regular

connectivity," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 4–8, May 2006.

[10] M. Sj¨alander and P. Larsson-Edefors, "The case for HPM based Baugh-Wooley multipliers," Tech. Rep. 08-8, Chalmers University of Technology, Goteborg, Sweden, 2008.

[11] C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," *IEEE Transactions on Computers*, vol. 22, pp. 1045–1047, 1973.

[12] M. Sj¨alander and P. Larsson-Edefors, "High-speed and low power multipliers using the Baugh-Wooley algorithm and HPM reduction tree," in *Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems (ICECS '08)*, pp. 33–36, September 2008.

[13] V. G. Oklobdzija and D. Villeger, "Improving multiplier design by using improved column compression tree and optimized final adder in CMOS technology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, no. 2, pp. 292–301, 1995.

[14] P. F. Stelling and V. G. Oklobdzija, "Design strategies for optimal hybrid final adders in a parallel multiplier," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 14, no. 3, pp. 321–331, 1996.

[15] B. Ramkumar, H. M. Kittur, and P. M. Kannan, "ASIC implementation of modified faster carry save adder," *European Journal of Scientific Research*, vol. 42, no. 1, pp. 53–58, 2010.

[16] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry select adder," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 2, pp. 371–375, 2012.

[17] B. Ramkumar and H. M. Kittur, "Optimal final carry propagate adder design for parallel multipliers," http://arxiv.org/abs/1110.3584.

[18] Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for low power applications," in *Proceedings of the IEEE International Symposiumon Circuits and Systems (ISCAS '05)*, vol. 4, pp. 4082–4085,May 2005.

[19] M. Sj¨alander and P. Larsson-Edefors, "High-speed and low power multipliers using the Baugh-Wooley algorithm and HPM reduction tree," in *Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems (ICECS '08)*, pp. 33–36, September 2008.

[20] S. R. Kuang, J. P. Wang, and C. Y. Guo, "Modified booth multipliers with a regular partial product array," *IEEE Transactions on Circuits and Systems II*, vol. 56, no. 5, pp. 404–408, 2009.

[21] B. Ramkumar and Harish M. Kittur "*Faster and energy efficient signed multiplier*," Hindawi Publishing Corporation VLSI Design Volume 2013, Article ID 495354, 12 pages .