International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Asic Implementation of 12-Bit Radix-8 Booth Multiplier

U.Geetalakshmi, M.V.Nageswara Rao

***Abstract*: Multipliers are playing a vital role in DSP and Neural Networks applications. Many methods have been introduced to work on multipliers that offer high speed, less power consumption and reduced area. Booth Algorithm demonstrates an efficient way of signed binary multiplication. In this paper, physical design of 12-bit radix-8 booth multiplier for signed multiplication is presented with an aim to improve the performance metrics such as power, area and delay. The performance of 12-bit radix-8 booth multiplier is compared with the 64-bit radix-16 booth multiplier.**

***Index Terms*: modified booth re-coding, partial products, radix-8.**

I. INTRODUCTION

Binary multiplier is a combinational logic circuit to perform multiplication of two binary numbers. The two binary numbers are more explicitly known as multiplicand and multiplier and the outcome is known as a product. The multiplicand and multiplier can be available in variety of bit sizes. The number of bits required to represent product can be estimated depending on the bit sizes required for the multiplicand and multiplier. The addition of the bit sizes of multiplicand and multiplier is equal to the total bit size of the product. Binary multiplication involves following steps:

1. Re-coding of the bits in a multiplier within a particular number configuration.
2. The multiplication of every bit in a multiplier can be multiplied with the multiplicand, producing no. of products
3. Reduce the partial product range using multi operand accumulation technique.
4. The two operands can be added by using carry propagate adder to get the ultimate result.

Elisardo Antelo et.al suggested the methods to improve the above and to achieve reduction in area, delay, power consumption and MBE multipliers. Recoding method is used to determine the no. of partial products. The re-coding process recodes binary information into signed bit information. The partial products generated for convention modified booth encoding is n/2+1 instead of n/2. Addition of extra 1bit at the LSB of partial product. This leads to unequal partial product array. In radix-q (q=2m) the binary operand is collected of unnecessary radix-q digits.

Revised Manuscript Received on 30 July 2019.

* Correspondence Author

Uriti Geetalakshmi*, ECE,GMR Institute of Technology, Rajam, AP, India

M.V.Nageswara Rao, ECE,GMR Institute of Technology, Rajam, AP, India.

Radix-4 is a commonly used modified booth re-coding method that recodes operands into set {-2,-1, 0, 1, 2}. This is a widely used re-coding because it requires simple shifts and complementation to get partial products.

Higher radix booth re-coding is not widely used. Since in higher radix, multiplicand needs odd multiples which can't be obtained by easy shifts and complementation and thus it requires carry propagate additions [1].

II. MODIFIED BOOTH MULTIPLIER

The modified booth encoding is a method used for decreasing the no. of partial products. Consider the multiplication of 2 two's complement representation of N-bit numbers of A and B

$$A = -a_{N-1} + \sum_{j=1}^{N-1} a_{N-1-j} 2^{-j}$$

$$B = -b_{N-1} + \sum_{j=1}^{N-1} b_{N-1-j} 2^{-j} \quad (1)$$

For modified booth encoding zero must be padded at the LSB side of B and N must be even. Kyung-Ju Cho et.al suggested the modified booth encoding, B can be defined as [2, 3].

$$B = \sum_{j=0}^{N/2-1} b'_{N/2-1-j} 2^{-(2j+1)} \quad (2)$$

$$b'_j = -2b_{2j+1} + b_{2j} + b_{2j-1} \quad (3)$$

By modified booth encoding those re-coded bits are grouped into triplets. Table 1 shows the Modified Booth Encoding operation refers to (3).

F. Lamberti et al. suggested the below Table 1 shows the Modified Booth Encoding for commonly used radix-4 booth multiplier [4]. The 2A in top table represents left shift of A by one bit. For negative function every A bit is complemented and add additional "1" bit binary value to the LSB bit of subsequent partial product row. Jiun-Ping Wang et.al suggested an additional "1" bit can be taken as correction bit cor_j, it describes pp_i partial product row is positive when cor_j = zero and is negative when cor_j=one. Each partial product row pp_i can be represented in its two's complement form so that the sign bit of pp_i have to be extensive to (2n-1) th bit location [5].

Table 1: Modified Booth Encoding Table for radix-4[3]

b_{2j+1}	b_{2j}	b_{2j-1}	b_i'	A_{sel}	$2A_{sel}$	NEG	cor_j
0	0	0	+0	0	0	0	0
0	0	1	+A	1	0	0	0
0	1	0	+A	1	0	0	0

0	**1**	**1**	**+2A**	**0**	**1**	**0**	**0**
1	**0**	**0**	**+2A**	**0**	**1**	**1**	**1**
1	**0**	**1**	**-A**	**1**	**0**	**1**	**1**
1	**1**	**0**	**-A**	**1**	**0**	**1**	**1**
1	**1**	**1**	**-0**	**0**	**0**	**1**	**0**

Fig.1 below represents the grouping of bits for commonly used radix-4 for N=8. Zero must be padded at the right side of

the multiplier operand. The 1st group of recoding digit is formed by padded bit and the remaining 2 bits are the 2 Least Significant Bits of the multiplier. The second group of recoding digit is formed by taking the starting bit as the MSB of 1st group and remaining 2 bits in LSB of multiplier.

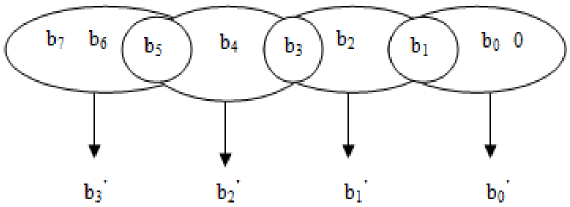

Fig.1 Grouping of multiplier bits for N=8

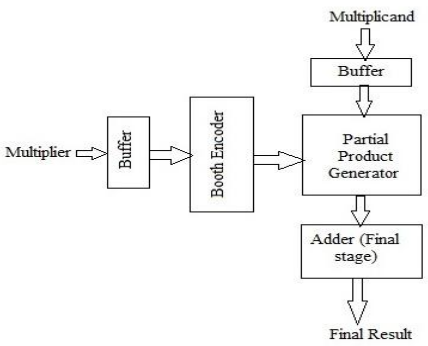

Fig.2 Block diagram for Modified Booth Multiplier [6]

There are several multiplier architectures available in the literature which focuses on the better power, speed and area. These multipliers are the used in Digital Signal Processors (DSP) and Microprocessors. Fig.2 above shows the block diagram of Modified Booth Multiplier and it is partitioned in two stages. A.Cilardo et.al suggested multiplier architecture consists of multiplier and multiplicand blocks produces partial products using booth encoder, partial product generator in first stage. These are added by using adder block in 2nd stage [6].

Buffer is used to store the 2 operands of multiplier and multiplicand. Zero must be padded at right side of multiplier and the multiplier bits can be grouped. It will generate the recoded digits, depending on the booth encoding table. The partial products are obtained by using Partial Product Generator (PPG) in the first stage. The obtained partial products are added by using an adder in the second stage.

Fixed width multipliers reduce the no. of partial products by using truncation techniques. Davide De Caro et.al suggested a method to separate partial products in to MP and LP [7].

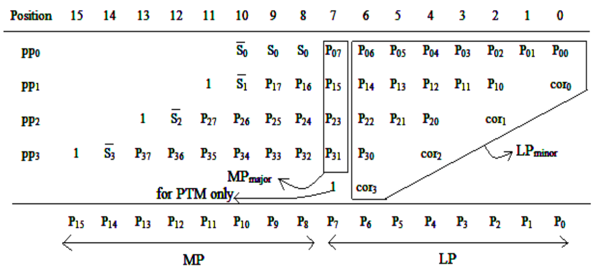

Fig.3 Partial products for modified 8×8 booth multiplier[7]

For generation of error compensation, LP can be further separated into LPmajor, LPminor [8,9]. Fig.3 shows the partial products of modified 8×8 booth multiplier with sign extension methods. $P_{i,j}$ denotes the jth bit of the partial product pp_i and s_j denotes the sign bit of partial product pp_i.

III. RADIX-8 BOOTH ALGORITHM

In this Radix-8, the multiplier(y) bits can be grouped into 4-bits in use at a time. Zero must be padded at right side of the multiplier operand. The 1st group of recoding digit is formed by padded bit and the remaining 3 bits are the 3 Least Significant bits of multiplier. The second group of recoding digit is formed by taking the starting bit as the MSB of 1st group and remaining 3 bits of Least Significant bits of multiplier. Below Fig.4 shows the grouping of bits for N=12 in Radix-8 booth multiplier. The Radix-8 algorithm generates the partial products are n/8. Consider multiplicand as x. Below Table 2 shows Modified booth encoding table for radix-8.

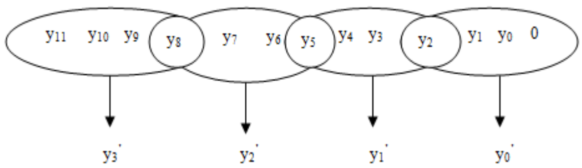

Fig.4 Grouping of multiplier bits for N=12

Table 2: Modified booth encoding table for radix-8

POSITIVE INPUT					NEGATIVE INPUT				
0	0	0	0	0	1	0	0	0	-4X
0	0	0	1	+1X	1	0	0	1	-3X
0	0	1	0	+1X	1	0	1	0	-3X
0	0	1	1	+2X	1	0	1	1	-2X
0	1	0	0	+2X	1	1	0	0	-2X
0	1	0	1	+3X	1	1	0	1	-1X
0	1	1	0	+3X	1	1	1	0	-1X
0	1	1	1	+4X	1	1	1	1	0

IV. FULL-WIDTH MULTIPLIER

Full width multiplier multiplies 2 N-bits of operand and gives N-bits of partial product. These multipliers are used to remove the one fraction of partial product array, to decrease the economy and make use of rectification functions. Here fixed width multipliers were taken to decrease the errors during multiplication.

The Fig.5 shows the Partial product for modified 12×12 booth multiplication. Which implies, the operation can be performed with 12 bits of multiplicand and 12 bits of multiplier which produces 12 bit partial product. The result of this multiplication depends on the fixed width multiplier so that it gives 12-bit output. Truncation takes place at the MSB side produces 12-bit product. Fig.6 below shows the addition of partial products in 12-bit Radix-8 Booth Multiplication. Multiplier bit can be grouped into four digits and it generates one recoded bits.

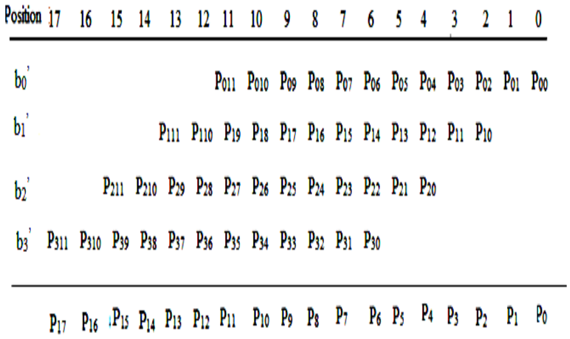

Fig.5 Partial product matrix for modified 12×12 booth multiplication

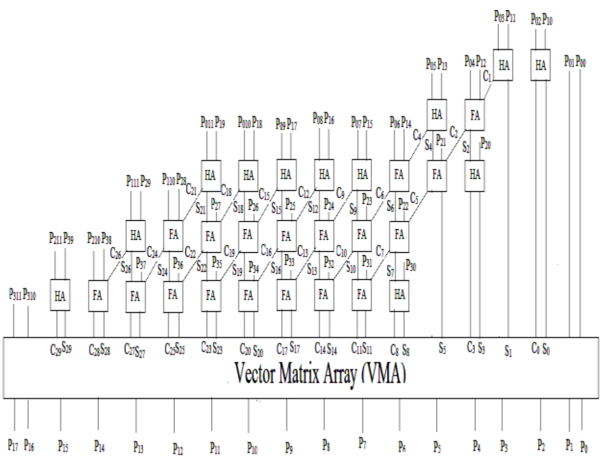

Fig.6 Addition of Partial products

This process continues all the multiplier bits can be grouped. In radix-8 for N=12 produces four recoded bits which are b0', b1', b2', b3'. Multiply the multiplicand with these four recoded bits gives products. Wen-Quan He et.al suggested those partial products are added with the help of half adders (HA) and full adders (FA) and the carry of HA and FA will be propagated to next levels. All the sum and carry terms are added by the Vector Matrix Array (VMA). VMA is used to add all the values which are in the vector notation. The VMA generates partial products from P0 to P17. Final result will come after the truncation [10].

V. Results and Discussion

The 12-bit Radix-8 Booth Multiplier is simulated and verified the results. The code for modified 12-bit Radix-8 booth multiplier is simulated using Cadence NC Launch tool. NC launch is a tool which helps to simulate the verilog and VHDL codes. Fig.7 refers to the simulation result for modified 12-bit Radix-8 Booth Multiplier. A radix-8 booth multiplier is elaborated, synthesized and verified in cadence tool. The technology library used is gpdk090 (90nm) synthesis of RTL design gives report on area, power and timing analysis. Physical verification gives the floor plan, placement and routing details.

A. Simulation Results

For checking the operation of booth multiplier, we implement a test bench. Here we gave different series of inputs to the Booth Multiplier which produces the output.

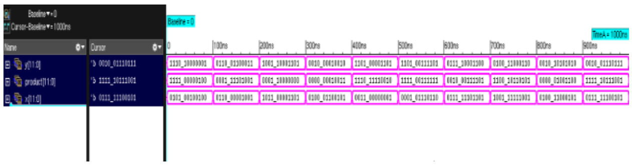

Fig.7 Simulation waveforms of the 12-bit Radix-8 Booth Multiplier

Along with the simulation, the schematic also generated. Below Fig.8 represents the schematic of the 12-bit Radix-8 booth multiplier.

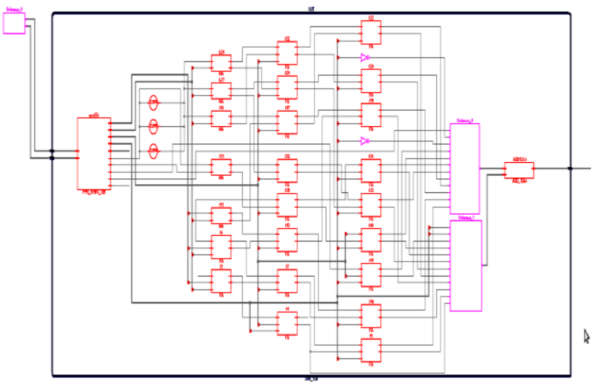

Fig.8 Schematic of 12-bit Radix-8 Booth Multiplier

B. Cadence RTL compiler

RTL Compiler tool helps to convert the Register Transfer Level which contains verilog code into the gate level netlist. The 12-bit Radix-8 Booth Multiplier, the netlist will be generated. This RTL compiler tool gives the area, power and timing reports of the entire 12-bit Radix-8 Booth Multiplier design. Below Fig.9 refers the RC schematic of 12-bit Radix-8 Booth Multiplier.

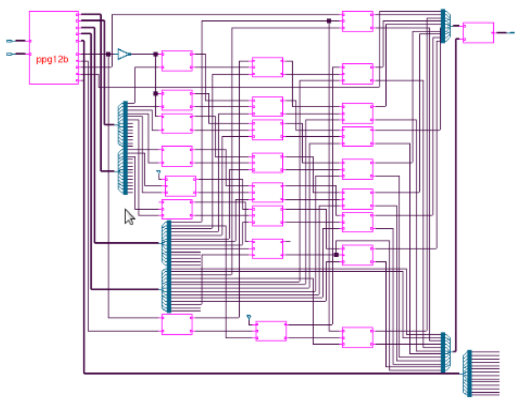

Fig.9 RC schematic of 12-bit Radix-8 Booth Multiplier

Retrieval Number: B3181078219/19©BEIESP
DOI: 10.35940/ijrte.B3181.078219
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Reports for Area, Power and Delay Analysis In the power report there will be few parameters like Dynamic Power (nW), Leakage Power (nW) and Total Power (nW) of the multiplier that is written in verilog code. The power report was generated according to the hierarchy given in the code. The verilog code for the multiplier consists many cells like Adder 12-bit, Partial Product Generator (PPG), full adders (FA), half adders (HA) and carry look ahead adder (CLA) etc, Vector Matrix Array (VMA) etc. and also the power consumption by every cell in the multiplier.

The area report consists of some parameters like Cells, Cell Area, Net Area and Total Area. Area report will give the summary of the area of the each cell in the current multiplier. And also this area report gives no. of gates and area sizes based on the specified technology library.

Timing analysis or Delay report consists of Fanout, load (fF), Slew rate(ps), Delay time (ps), arrival time (ps) as per the hierarchy given in the code. Arrival time contains rise time data. This report will give the slew rate and, delay time and arrival time of each cell of the multiplier. Below Table 3 gives the comparison table of the power, area and delay report of Radix-8 booth multiplier with Radix-16 booth multiplier in 90nm technology.

Table 3: Comparison table for Power, Area and Delay in 90nm technology

S. No.	Parameters	64-bit Radix-16 Booth multiplier	12-bit Radix-8 Booth multiplier
1.	No. of bits	64-bits	12-bits
2.	Power	672000nW	124924nW
3.	Area	42000 mm^2	2500 mm^2
4.	Delay	967.5 ps	690.2 ps

C. Layout

The automated layout is generated by the Cadence SoC Encounter which contains the constraints like import design, Floor plan, placement and routing and netlist of the multiplier is generated in pre-layout steps. The final layout is generated after performing the routing is shown in below Fig.10.

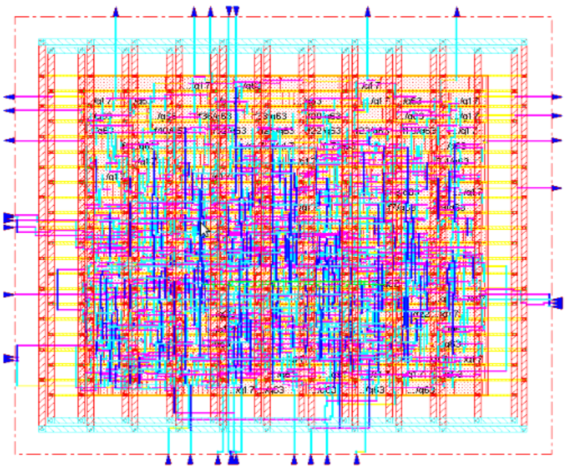

Fig.10 Layout of the Radix-8 Booth Multiplier using Cadence SoC Encounter

VI. CONCLUSION

The 12-bit Radix-8 Booth Multiplier is implemented and its architecture is verified. Synthesis is performed using RTL compiler tool using slow and fast libraries. Results shows that 12-bit radix-8 booth multiplier needs less power, less area and less delay when compared with 64-bit radix-16 booth multiplier. Power consumed, area required and delay occurred by 12-bit radix-8 booth multiplier is 124924nW, 2500mm^2 and 690.2ps respectively. When compared with the 64-bit radix-16 booth multiplier the 12-bit radix-8 booth multiplier consumes less power by 5times , reduction in area by 18times and reduced delay by 0.7 times. Hence, 12-bit radix-8booth multiplier is suitable for applications such as DSP and Neural Networks.

REFERENCES

1. Elisardo Antelo, Paolo Montuschi, Fellow, IEEE, and Alberto Nannarelli, Senior Member, IEEE," Improved 64-bit Radix-16 Booth Multiplier Based on Partial Product Array Height Reduction," IEEE Trans. Circuits Syst. I, vol.64, no.2, February 2017
2. Kyung-Ju Cho, Kwang-Chul Lee, Jin-Gyun Chung, Member, IEEE, and Keshab K. Parhi, Fellow, IEEE," Design of Low-Error Fixed Width Modified Booth Multiplier," IEEE Trans.Circuits Systs, vol. 12, no. 5, May 2004.
3. S. Kuang, J. Wang, and C. Guo, "Modified booth multipliers with a regular partial product array," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 56, no. 5, pp. 404–408, May 2009.
4. F. Lamberti et al., "Reducing the computation time in (short bit-width) twos complement multipliers," IEEE Trans. Comput., vol. 60, no. 2, pp. 148–156, Feb. 2011.
5. Jiun-Ping Wang, Shiann-Rong Kuang and Shish-Chang Liang," High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications," IEEE Trans. Circuits Syst. vol. 19, no. 1, January 2011.
6. A.Cilardo etal.,"High speed speculative multipliers based on speculative carry-save tree," IEEE Trans.Circuits Syst. I, vol. 61, no. 12, pp. 3426–3435, Dec. 2014.
7. Davide De Caro, Nicola Petra, , Antonio Giuseppe Maria Strollo, Fabio Tessitore, and Ettore Napoli," Fixed-Width Multipliers and Multipliers Accumulators With Min-Max Approximation Error," IEEE Trans. Circuits Syst. I, vol. 60, no. 9, September 2013.
8. N. Petra et al., "Design of fixed-width multipliers with linear compensation function," IEEE Trans.Circuits Syst. I, Reg. Papers, vol. 58, no. 5, pp. 947–960, May 2011.
9. Shyh-Jye Jou, Meng-Hung Tsai, and Ya-Lan Tsao," Low-Error Reduced-Width Booth Multipliers for DSP Applications", Fundamental Theory and Applications, IEEE Trans. Circuits Syst. I, vol. 50, no. 11, November 2003.
10. Wen-Quan He, Yuan-Ho Chen, and Shyh-Jye Jou," High-Accuracy Fixed-Width Booth Multipliers Based on Probability and Simulation", IEEE Trans. Circuits Syst. I, vol.62, no.8, August 2015

AUTHORS PROFILE

U.Geetalakshmi received her B.Tech degree in Electronics and Communication Engineering from Sri Sivani College of Engineering which is affiliated under JNTUK, Andhra Pradesh, India in 2016.She is pursuing her M.Tech degree in VLSI and Embedded Systems Design specialization in GMRIT which is affiliated under JNTUK, Andhra Pradesh, India. Her areas of interest are ASIC design implementations and low power VLSI.

Dr.M.V.Nageswara Rao received his B.Tech, ME and PhD degree from Nagarjuna University, Osmania University and Andhra University respectively. Presently working as a professor in the Department of ECE , GMR Institute of Technology, Rajam, AP, India. His areas of interests are Signal Processing and VLSI.