Ila Chaudhary, Akash Kumar, Deepika Sharma / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 4, Jul-Aug 2013, pp.2090-2093

Design and Optimization of High Speed Multiplier

Ila Chaudhary[1], Akash Kumar[2], Deepika Sharma[3]
[1](Assistant Professor, Department of ECE, Manav Rachna International University, Faridabad)
[2](Department of ECE, Manav Rachna International University, Faridabad)
[3](Department of ECE, Manav Rachna International University, Faridabad)

ABSTRACT

Two's complement multipliers are important for a wide range of applications. Paper describes a technique to reduce by one row the maximum height of the partial product array generated by Radix-4 Booth's multiplier, without any increase in the delay of the partial product generation stage. The design of 8 bit and 16 bit multiplication scheme using different types of multiplier like Array multiplier, Shift and Add multiplier, Radix-2 Booth's multiplier and Radix-4 Booth's multiplier is presented.

Keywords –Array multiplier, Shift and Add multiplier, Radix-2 Booth's multiplier, Radix-4 Booth's multiplier.

I. INTRODUCTION

The need of high speed multiplier is increasing as the need of high speed processors are increasing. Higher throughput arithmetic operations are important to achieve the desired performance in many real time signal and image processing applications [3]. One of the key arithmetic operations in such applications is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. In the past multiplication was implemented generally with a sequence of addition, subtraction and shift operations. A multiplier is one of the key hardware blocks in most digital signal processing (DSP) systems. Typical DSP applications where a multiplier plays an important role include digital filtering, digital communication and spectral analysis.

II. DIFFERENT TYPES OF MULTIPLIER

2.1 Array Multiplier

A combinational multiplier is a good example of how simple logic functions can be combined to construct a much more complex function. The multiplier is constructed from an array of building blocks as shown in fig.1. Each building block consists of an AND gate for computing locally the corresponding partial product, an input passed into the block (Sum In) and a carry (Cin) passed from a block diagonally above. It generates a carry out bit (COUT) and a new sum out bit (Sum Out).

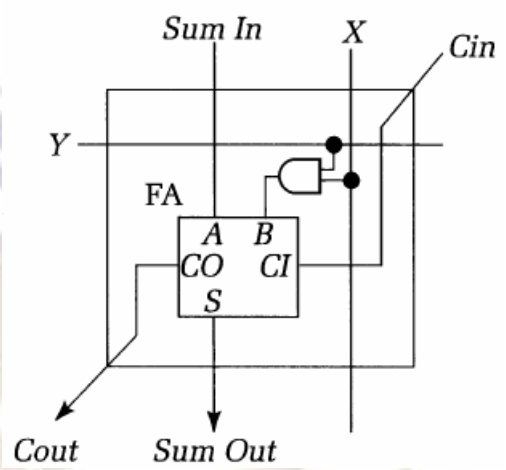

Fig.1 Multiplier with an array of building blocks

2.2 Shift and Add Multiplier

This method adds the multiplicand X to itself Y times, where Y denotes the multiplier. To multiply two numbers, the algorithm is to take the digits of the multiplier one at a time from right to left, multiplying the multiplicand by a single digit of the multiplier and placing the intermediate product in the appropriate positions to the left of the earlier results [4].

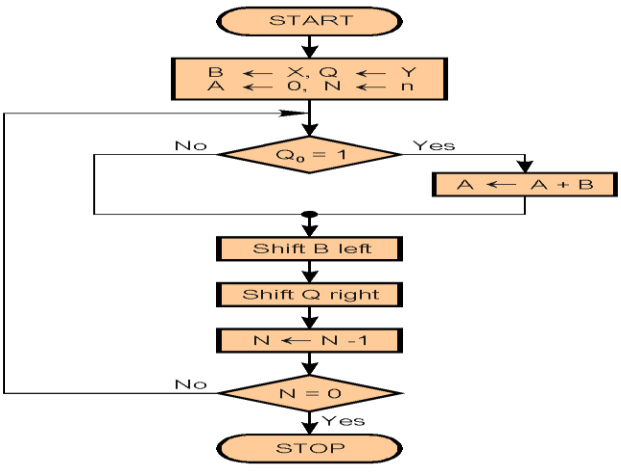

Fig.2 Shift and Add multiplication algorithm

2.3 Radix-2 Booth's Multiplier

One of the ways to multiply signed number was invented by booth. Booth's algorithm examines adjacent pairs of bits of N-bit multiplier.

Ck	S
00	0
01	+1
10	-1
11	0

Table 1 Radix-2 Booth's recoding strategy for each of the possible block

2.4 Radix-4 Booth's Multiplier

One of the solutions realizing high speed multipliers is to enhance parallelism which helps in decreasing the number of subsequent calculation stages. The original version of Booth's multiplier (Radix-2) had two drawbacks [7] [8]. One is the number of add / subtract operations became inconvenient while designing parallel multipliers. Second is the algorithm becomes inefficient when there are isolated 1s. These problems are overcome by using Radix 4 Booth's Algorithm.

Multiplier bits	Signed Multiplier Digit
000	0
001	+1* Multiplier
010	+1* Multiplier
011	+2* Multiplier
100	-2* Multiplier
101	-1* Multiplier
110	-1* Multiplier
111	0

Table 2 Radix-4 Booth's recoding strategy for each of the possible block

III. SIMULATION RESULTS

All the multipliers are coded with VHDL, simulated and synthesized by Xilinx ISE 8.1i and Modelsim SE 6.3f.

3.1 Array Multiplier

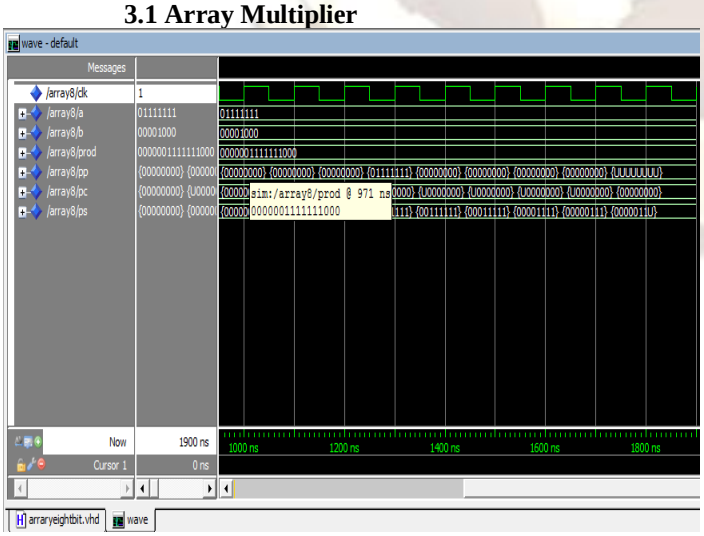

Fig.3 Simulation result of 8-Bit Array multiplier

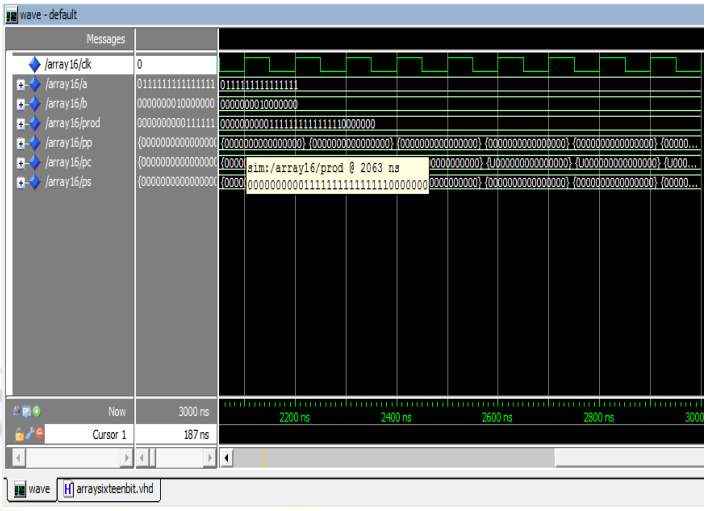

Fig.4 Simulation result of 16 Bit Array multiplier

3.2 Shift and Add Multiplier

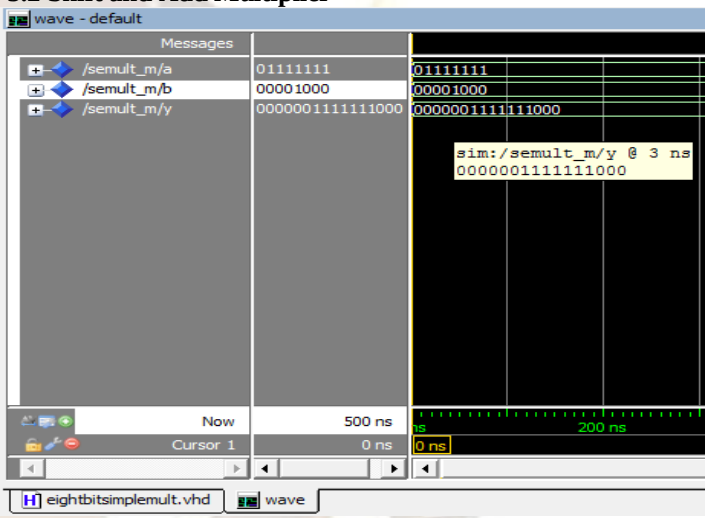

Fig.5 Simulation result of 8-bit Shift and add multiplier

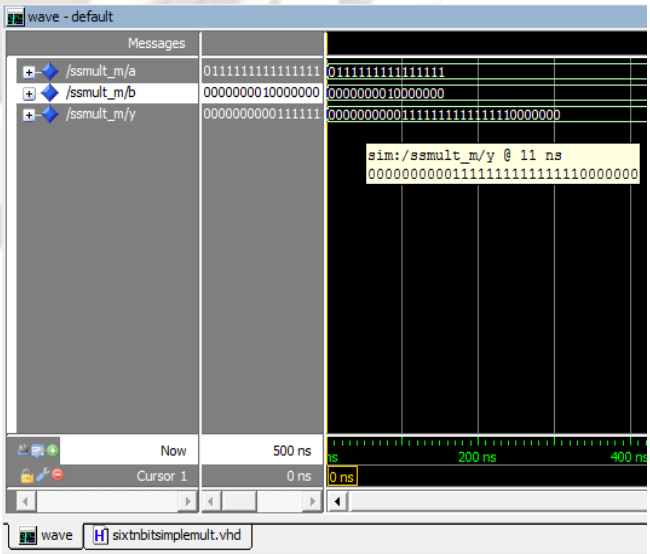

Fig.6 Simulation result of 16-Bit Shift and Add multiplier

3.3 Radix-2 Booth's Multiplier

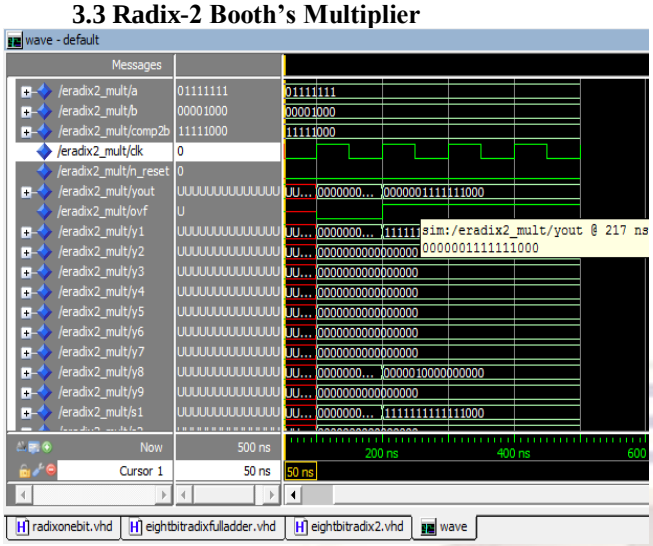

Fig.7 Simulation result of 8-Bit Radix-2 Booth's multiplier

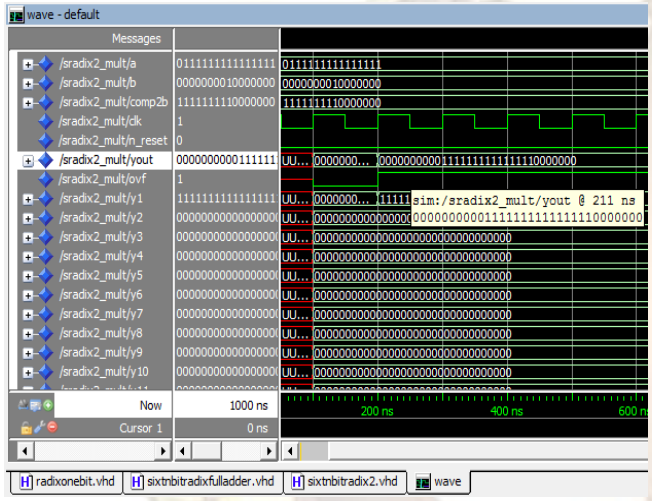

Fig.8 Simulation result of 16-Bit Radix-2 Booth's multiplier

3.4 Radix-4 Booth's Multiplier

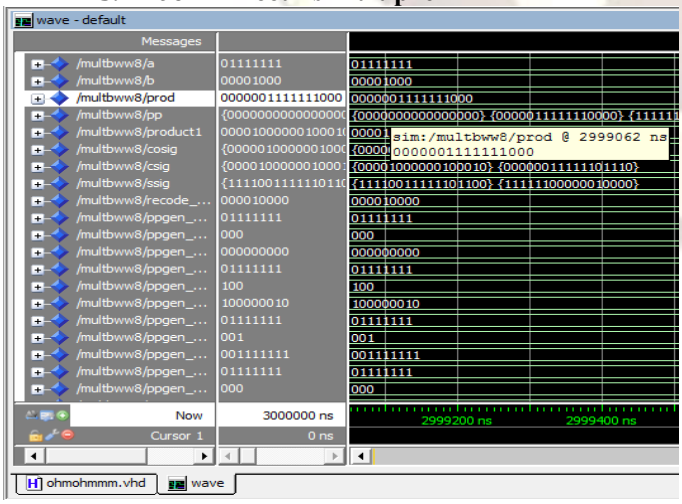

Fig.9 Simulation result of 8-Bit Radix-4 Booth's multiplier

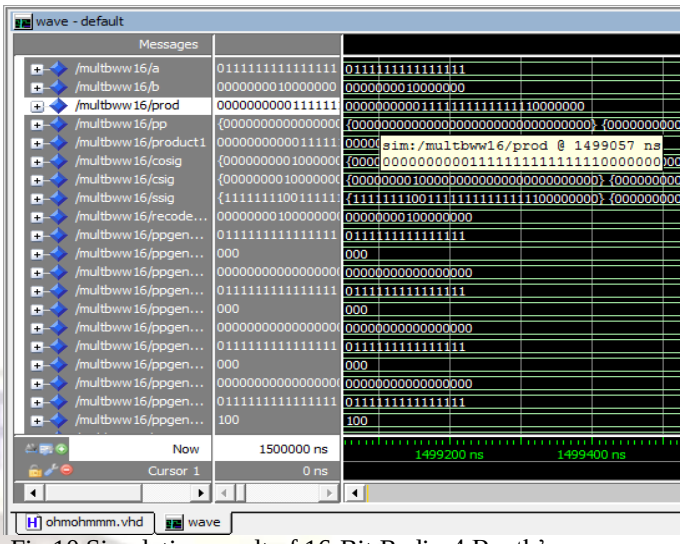

Fig.10 Simulation result of 16-Bit Radix-4 Booth's multiplier

3.5 Analysis

8-BIT MULT -IPL -IER	ARRAY MULT -IPL -IER	SHIFT AND ADD MULT -IPL -IER	RADIX -2 BOOTH 's MULT -IPL -IER	RADIX -4 BOOTH 's MULT -IPL -IER
No. of Slices & Slices of flip flop	70	67	128 & 88	54
4 I/P LUTs	121	125	224	98
Bonded I/O	32	32	43	28
Level of logic	16	30	19	16
Maximum Comb. Delay (ns)	32.261	30.526	36.171	22.490
Logic delay (%)	44 %	70 %	45 %	55.7 %
Route delay (%)	56 %	30 %	55 %	44.3 %
Total Power (watt)	0.280	0.280	0.285	0.279
Quiescent Power (watt)	0.263	0.263	0.263	0.263
Dynamic Power (watt)	0.017	0.017	0.022	0.017

Table 3 Synthesis results of 8-bit multiplier

16-BIT MULT -IPL -IER	ARRAY MULT -IPL -IER	SHIFT AND ADD MULT -IPL -IER	RADIX -2 BOOTH 's MULT -IPL -IER	RADIX-4 BOOTH 's MULT -IPL -IER
No. of Slices & Slices of flip flop	290	274	519 & 333	380
4 I/P LUTs	505	509	912	707
Bonded I/O	64	64	83	64
Level of logic	32	62	37	41
Maximum Comb. Delay (ns)	61.421	59.773	64.207	28.972
Logic delay (%)	37.5	69.0	40.8	55.9
Route delay (%)	62.5	31.0	59.2	44.1
Total Power (watt)	0.282	0.282	0.289	0.282
Quiescent Power (watt)	0.263	0.263	0.263	0.263
Dynamic Power (watt)	0.020	0.020	0.026	0.020

Table 4 Synthesis results of 16-bit multiplier

IV. CONCLUSION

By analyzing the result for large size operand like 16 bit operand Radix-4 has minimum logic delay as compared to other. Hence it is concluded that Radix-4 Booth's multiplier reduces the partial product generated during multiplication and increase the speed of multiplier

REFERENCES

[1] IEEE TRANSACTIONS ON VERY LARGE INTEGRATION VLSI SYSTEMS.VOL2. No.1 DECEMBER 1994.
[2] IEEE TRANSACTIONS ON COMPUTERS, VOL. 47. No.3 MARCH 1998.
[3] Ramya Muralidharm, Student member, IEEE, and Chip-Hong Chang, Senior Member, IEEE, IEEE TRANCTIONS ON CIRCUITS AND SYSTEMS-1: REGULAR PAPERS, VOL. 58, NO.5, MAY 2011.
[4] M. Mottaghi-Dastjerdi, A. Afzali, and M. Pedram, To appear in IEEE Trans. On VLSI Systems, 2008.
[5] Young-Ho Seo and Dong- Wook Kim, IEEE TRANSACTION ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 18, NO.2, FEBRUARY 2010.
[6] Peter-Michael Seidel, Lee D. McFearin and David W. Mathula, IEEE TRANSACTIONS ON COMPUTERS, VOL. 54, NO. 2, FEBRUARY 2005.
[7] K. Sreenivasa Rao, D. Rajesh Seety, International Journal of Scientific & Engineering Research, Volume2, Issue 12, December-2011-ISSN 2229-5518.
[8] Nishat Bano, International Journal of Scientific & Engineering Research, Volume 3, Issue2, February- 2012 1 ISSN 2229-5518.