International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 6 75 - 78

Analysis and Comparison of Different Multiplier

Jasbir Kaur
Assistant Professor, E & Ce Department
Pec University Of Technology
Chandigarh
jasbirkaur70@yahoo.co.in

Sumit Kumar
P.G Scholar, E & Ce Department
Pec University Of Technology
Chandigarh
sumitpareek2602@gmail.com

***Abstract*—** Multiplication is one of the important parameter in various digital applications such as in digital signal processor, microprocessor so in this paper firstly we analyse various 4*4 multiplier circuit and then analyses various 12*12 bit multiplier circuit. After then their parameters i.e area, power and delay are analyzed. All these multiplier are designed in Verilog language and synthesized on Xilinx ISE simulator and using cadence RTL schematic respectively. Multipliers included in this paper are Array multiplier, Radix-4 multiplier, Radix-8 multiplier, Wallace Multiplier and Conventional multiplier. On comparison it is found that for 4*4 and 12*12 multiplier, array multiplier have highest delay but have less power consumption while Booth multiplier(Radix-4) is having high speed with moderate power consumption.

Keywords- *Radix-2, Radix-4, Array, Wallace, Conventional Multipliers*

*****

I. INTRODUCTION

In today's world Multipliers are widely used in various high performance systems such as Microprocessor, FIR filters, digital signal processors etc. As the multiplier is the slowest element of a system So Systems performance is generally determined by the performance of multiplier. Furthermore, the most area consuming device in a circuit is also multiplier. In any system the main issue of multipliers are optimization of their speed, area and power. But generally in any system speed and area are usually conflicting constraints because improving area of the system results in low speed and vice versa. Many Researchers have developed their multiplier and optimized the system parameter according to application. Multiplication process is used in nearly all the applications which require mathematical calculation like communications, graphics, image enhancement, instrumentation and measurement, audio and video processing, animations, special effect, navigation, GPS, RADAR, and in control applications like robotics, machine vision etc. By adding partial products in circuit, we get the multiplication of two numbers. Different partial products are generated by different multiplication algorithm and addition of these product produce the final result. Multipliers like Array multipliers, Wallace multiplier, Radix 2, Radix 4 booth multiplier are explained below.

II. ARRAY MULTIPLIER

Array multiplier is a simple multiplier well known due to its regular structure. Figure 1 shows multiplication of two 4*4 bit numbers.

				1	0	1	1	——	11
		×		1	1	0	1	——	13
				1	0	1	1		
			0	0	0	0			
		1	0	1	1				
	1	0	1	1					
1	0	0	0	1	1	1	1	——	143

Figure 1 Example of 4*4 Array multipliers

To multiply two numbers in Array multiplier we firstly generate partial product by bitwise anding of multiplier and multiplicand, each of these partial products is either the multiplicand shifted by a particular amount when multiplier bit is one or 0 when multiplier bit is 0. Ripple adders is used to accumulate these partial products and to provide final output result of array multiplication. In ripple carry adder the carry out from the least significant bit ripples to the most significant bit in same row, and then down the "left side" of the structure. Half and full adders adds all partial product in a ripple form. Half adder have two input and it provide sum and carry as output while a full adder's require two inputs and one carry in from the adjacent full adder in the same row. The disadvantage of ripple adder is that in an array it requires a lot of time to optimize the adders because of carry in from previous stage, so timing delay is linearly proportional to the sum of the sizes of the input operands. To decrease the delay, generally row of array multipliers are designed with carry-save adders (CSA). Array multiplier is used in various VLSI math co-processors and special purpose DSP chips.

The algorithm used in Array multiplier circuit is add and shift algorithm. Figure 2 shows a 12*12 unsigned array multiplier synthesized in cadence RTL schematic using Verilog language.

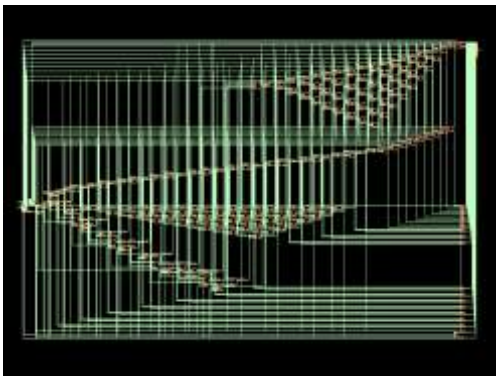

Figure 2. 12*12 Array Multiplier

III. RADIX 2 BOOTH MULTIPLIER

Radix 2 booth multiplier was proposed and developed by O. L. Macsorley in 1961 for speeding Up multiplication in Early

computers. In booth multiplier a binary number is recoded according to booth algorithm which results into the number of partial product by half. The booth recoding method is used for high bit parallel multiplier for generation of partial products [6]. Booth algorithm is generated by looking at two bits of the multiplier at a time, and then determines what partial product to generate according to the booth table. The drawbacks of booth multipliers is that number of add subtract operations and the number of shift operations becomes variable and becomes inconvenient in designing parallel multipliers and algorithm becomes inefficient when there are isolated 1's [6].

IV. RADIX 4 BOOTH MULTIPLIER

Problems of Radix-2 booth multiplier can be overcome by radix-4 multiplier. Radix-4 booth multipliers recode three bits at a time which results into very less number of partial products. Hence recoding the multiplier in radix 4 is a powerful way to speed up the multiplication. In each cycle a greater number of bits can be inspected and eliminated therefore total number of cycles required to obtain products get reduced [11]. Procedure for implementing radix-4 algorithm is as follows:

Firstly in each cycle of radix-4 algorithm inspected 3 bits of binary number

• Place a 0 to the right of LSB

• Extend the sign bit 1 position if necessary to ensure that n is even.

• Determine partial product scale factor from radix-4 encoding table

•Compute the multiplicand multiples and sum up all the partial product.

Radix-4 multiplier groups the multiplier into groups of three consecutive digits where the outermost digit in each group is shared with the outermost digit of the adjacent group [7]. Each of these groups of three binary digits then corresponds to one of the numbers from the set {2, 1, 0,-1, -2}. Number of bits inspected in radix r is given by n = 1 + log2r. Let the two numbers to be multiplied are X = 21 and Y = -15.

TABLE 1. Radix-4 Booth Multiplier Example

								0	0	0	1	0	1	0	1	21
								1	1	1	1	0	0	0	1	-15
						0	0	0	0	0	1	0	1	0	1	PP1
				0	0	0	0	0	0	0	1	0	1	0	1	PP2
		1	1	1	1	1	0	1	1	0	0	0	1	0	1	PP3
1	1	1	1	0	1	1	1	1	0	0	0	0	1	0	1	PP4
1	1	1	1	1	1	0	1	1	1	0	0	0	1	0	1	-315

Multiplicand X = 21 = 00010101

Multiplicand Y= -15=111110001(2's Complement)

X*Y = -315

Logic circuit that carries out recoding of three multiplier bits at a time and generates necessary control bits is shown in figure below:-

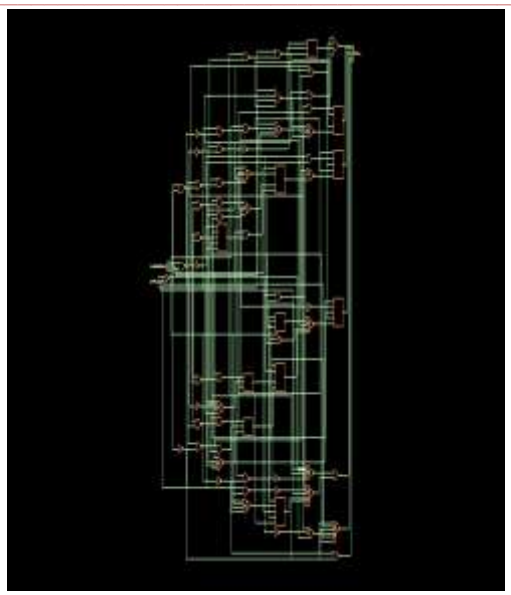

Figure 3: Block diagram of 4*4 radix 4 booth multiplier

V. WALLACE TREE MULTIPLIER

Various algorithm and techniques have been designed in past with objective of improving the speed of the parallel multiplier. Wallace Tree multiplier is a very good technique to improve the speed of parallel multiplier [5]. In Wallace tree architecture, instead of adding the partial product in ripple fashion all the bits of all of the partial products in each column are added together by a set of counters in parallel without propagating any carries. At the last stage of Wallace multiplier a fast adder is used at the end to produce the final result. Since the addition of partial products now becomes 0 (IogN) the speed of Wallace multiplier greatly improves. Figure 3 shows a 4*4 Wallace tree multiplier synthesized in cadence RTL schematic using Verilog language.

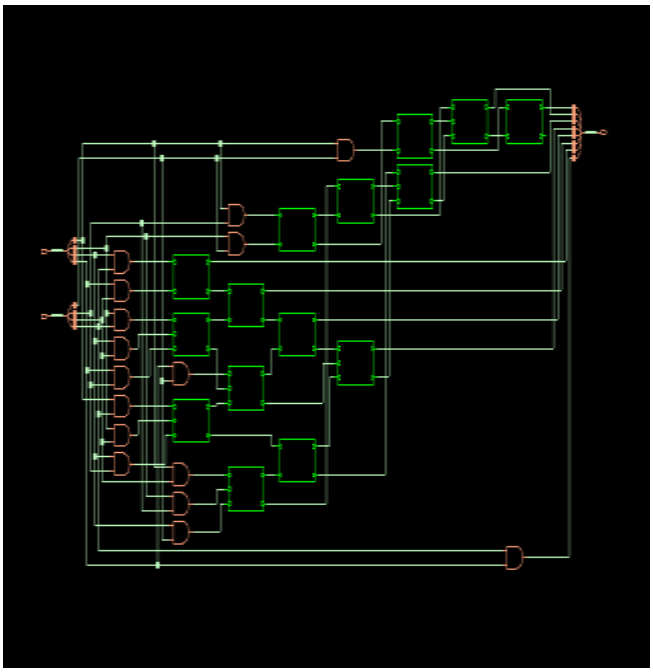

Figure 4: 4*4Wallace Tree Multiplier

VI. RESULTS AND DISCUSSION

- SIMULATION TOOL USED

Multiplier discussed above in this paper is firstly designed using Xilinx FPGA tool in Verilog HDL programming language. After the analysis on Xilinx tool these program are synthesized and there result are obtained on Cadence RTL schematic tool. Different parameter i.e area, power and delay of different multiplier are then analyzed.

- AREA ANALYSIS

From the result obtained from various multiplier, it is observed that for 4*4 multiplier array multiplier have very less area but as we go to higher bit of multiplier and multiplicand Radix 4 multiplier provides very less area. So no. of bits of multiplier decides the multiplier selection based on area.

TABLE 2 COMPARISON OF DIFFERENT PARAMETER OF 4*4 MULTIPLIER

S.no	Multiplier	Delay(ns)	AREA	Power Consumption(uW)	Area-Delay Product(AD)
1	Array Multiplier	1.791	78	1.369	139.698
2	Radix-2 booth multiplier	1.744	158	2.555	275.552
2	Radix-4 booth multiplier	.779	96	1.18	74.78
4	Wallace Multiplier	1.517	87	1.573	131.97
5	Conventional Multiplier	0.992	85	1.453	84.32

TABLE 3 COMPARISON OF DIFFERENT PARAMETER OF 12*12 MULTIPLIER

S.no	Multiplier	Delay(ns)	AREA	Power Consumption(uW)	Area-Delay Product(AD)
1	Array Multiplier	7.106	807	23.98	5.734×10^{3}
2	Radix-2 Booth Multiplier	4.89	1175	38.264	$5.745\text{x}10^{3}$
3	Radix-4 Booth multiplier	4.669	772	26.22	3.604×10^{3}
4	Wallace Multiplier	4.860	956	29.209	4.646×10^{3}
5	Conventional Multiplier	4.665	859	28.830	4.007×10^{3}

- DELAY ANALYSIS

From the result, it is observed that array multiplier is having highest delay despite of having moderate area while radix-4 booth multiplier have very less delay so we use radix-4 booth multiplier in the application where high speed required.

- POWER CONSUMPTION

Since power consumption is an important parameter in portable and other devices so the devices which consume less power are the need of current days. From the result, radix-4 multiplier has very less consumption of power because of very less no. of cell used in both 4*4 and 12*12 multiplier.

VII. CONCLUSION

In this paper, different types of multiplier algorithm such as array multiplier, radix-2, radix-4, wallace and conventional multiplier were designed. They are designed firstly in Xilinx FPGA tool using Verilog HDL language. Then they are synthesized on cadence RTL schematic.

From the result it has been found that if less area is required, then for low bit of multiplier array multiplier and for higher bit Radix 4 multiplier is the best choice to use in circuit . Since Multiplier is a slower unit and speed of circuit depends upon speed of the multiplier so speed is a major parameter and Radix-4 multiplier has very high speed along with moderate area and power consumption. So radix-4 multiplier is widely used in multiplication circuit.

REFERENCES

[1] N. Jiang and D. Harris, "Parallelized Radix-2 Scalable Montgomery Multiplier," submitted to IFIP Intl. Conf. on VLSI, (2007).
[2] D. Kudeeth., "Implementation of low-power multipliers", Journal of low-power electronics, vol. 2, 5-11, (2006).
[3] P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu, vol. 8, no. 2, pp. 153–158, (2004).
[4] M. Sheplie, "High performance array multiplier", IEEE transactions on very large scale integration systems, vol. 12, no. 3, pp. 320-325, (2004).
[5] C.S.Wallace, "A suggestion for a fast multiplier", *IEEE Trans. Elechon. Con@.*, vol. EC-13, pp. 14-17, Feb. 1964.
[6] Kaur.J, Kumar S," Performance Comparison of higher radix booth multiplier using 45nm tech.", IJIRET, Vol-5 ,issue1, pp. 171-177 Jan 2016
[7] 0. Salomon, J.-M. Green, and H. Mar, "General Algorithms for a Simplified Addition of 2's Complement Numbers", IEEE *Journal* of *Solid-state Circuits*, V01.30, No.7, pp.839-844, July 1995
[8] *S.Shah*, A. J. A[-Khabb, D. AI-Khabb , "Comparison of 32-bit Multipliers for Various Performance Measures**"** Microelectronics, 2000. ICM 2000. Proceedings of the 12th International Conference_ pp.75-80,Oct 2000
[9] M. Själander and P. Larsson-Edefors, "*The Case for HPM-Based Baugh-Wooley Multipliers*", Department of Computer Science and Engineering, Chalmers University of Technology, Tech. Rep. 08-8,March 2008.
[10] Sang-In Cho; Kyu-Min Kang; Sang-Sung Choi, "Implemention of 128-Point Fast Fourier Transform Processor for UWB Systems," Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International , vol., no., pp.210-213, 6-8 Aug.2008
[11] A. Efthymiou, W. Suntiamorntut, J. Garside, and L. Brackenbury. An asynchronous, iterative implementation of the original Booth multiplication algorithm. In Proc. Int. Symp. On Advanced Research in Asynchronous Circuits and Systems, pages 207–215. IEEE Computer Society Press, Apr. (2004).