International Journal of Current Trends in Engineering & Technology
ISSN: 2395-3152
Volume: 02, Issue: 01 (JAN-FAB, 2016)

Low Area 8 Bit Multiplier using Hardware Reuse Technique

Rakesh Patel, Prof. Himanshu Nautiyal, Prof. Sangeeta Shukla
Department of Electronics & Communication Engineering
Sagar Institute of Science & Technology, Bhopal

Abstract - This paper presents a hardware and power efficient binary multiplier using resource reuse technique. The proposed design uses efficient design of half adder and full adder circuits which uses less number of logic gates. An array multiplier of M xM M needs M rows and n columns of full adder circuits to generate the product term. The proposed architecture requires only 3 rows of adders to produce the product term. Intermediate product terms are stored in the memory elements (flip flop). As FF (Flip-flop) takes less area and consume less power as compared to adder circuit (combinational circuit), this improves the hardware and power efficiency of the design. This technique is used to implement a 8 X 8 multiplier whose the results are compared with other 8 X 8 array multipliers. Spartan 3 FPGA is used to implement the design. The design is very linear and it can be easily increased to implement large multiplier.

Keywords – power efficient, multiplier, switching delay, hardware efficient and resource reuse

INTRODUCTION

Multipliers play very critical role in today's digital signal processing and many other applications. With advances in technology, many researchers have and are practicing to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and therefore less area or even combination of them in one multiplier thus making them suitable for various low power and compact VLSI implementation. It is well known that Multipliers consume maximum power in DSP computations [1]. Hence, it is very important for modern DSP systems to design low-power multipliers to reduce the power dissipation. For low-power multiplier architecture design, many researcher experiments & find out results on the reduction of the activity factor (switching activities) [2] have been published. Besides that, a simple and straightforward approach [3] for low-power multiplier is to design a low-power Full Adder to reduce the power losses in an array multiplier. The other designs are proposed for reduce the power wastage (dissipation) in a multiplication operation, interchanging dynamic operands [4] or using partially guarded computation [5]. Furthermore, to minimize power dissipation architectural modification can be used via row bypassing [6] or column bypassing [7] technique. Based on the concept of theory row and column bypassing techniques for the reduction of the power loss/dissipation, a low-power 2 - dimensional bypassing based multiplier [8] and a low-power row and column bypassing-based multiplier [9] are further proposed. However, the introduction of the extra bypassing circuit decreases the ability of minimize the power dissipation, and it also induces extra delay in the circuit. In array multiplier n number of full adder layers are required, where n is the size of the architecture. In this work we have used only three layers of full adders to implement the complete design, this will reduce the area requirement and it will also reduce the power consumption of the design. The paper is organized as follows in section II related work is given, in section III array multiplier is discussed; Section IV discusses the proposed hardware efficient multiplier. Section VI discusses the synthesis report of hardware efficient multiplier and comparison with other designs is also

International Journal of Current Trends in Engineering & Technology
ISSN: 2395-3152
Volume: 02, Issue: 01 (JAN-FAB, 2016)

discussed. Section VI contains concluding remarks for proposed design.

II RELATED WORK

The multiplication of two 4 bit numbers is shown in the figure 1.

			Y=	Y3	Y2	Y1	Y0
			X=	X0	X0	X0	X0
				Y3X0	Y2X0	Y1X0	Y0X0
			Y3X1	Y2X1	Y1X1	Y0X1	
		Y3X2	Y2X2	Y1X2	Y0X2		
	Y3X3	Y2X3	Y1X3	Y0X3			
P7	P6	P5	P4	P3	P2	P1	P0

Figure 1: 4 X 4 Array Multiplications

			Y=	1	0	0	1
			X=	1	1	1	0
				0	0	0	0
			1	0	0	1	
		1	0	0	1		
	1	0	0	1			
0	1	1	1	1	1	1	0

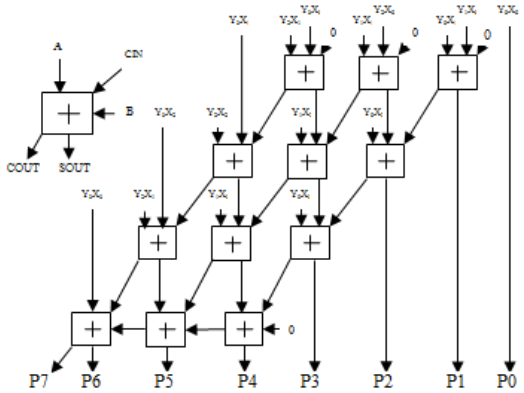
Figure 2: 4 X 4 example of array multiplication

An example of above multiplication process is shown in figure 2: AND & OR gates are used to generate the Partial Products, P_n, If the multiplicand is N-bits & Multiplier is M-bits then there will be N x M partial product. The way that the partial products are generated / summed up is the difference between the different architectures of various multipliers. For CMOS circuits design, the power dissipation can be divided in two categories as static power dissipation and dynamic power dissipation. In general, static power consumption result of leakage current and dynamic consumption is due to the switching transient current. For static power dissipation, the loss of power is directly proportional to the number of transistors used. For dynamic power dissipation, the consumption is due to charging and discharging action of load capacitance. The average dynamic dissipation of a CMOS gate is P avg = ½ Cf V_{dd} N

Where C is the load capacitance, f is the clock frequency, VDD is the power supply voltage and N is the no. of switching activity in a clock signal .Hence, it is very important for modern DSP circuit application to develop low-power multipliers to reduce the power losses. In this paper we present a novel technique of multiplication that will serve our two important needs i.e. low power consumption and low area to make our design greener and compact.

III ARRAY MULTIPLICATION

In array multiplier, each partial product is generated by taking into account the multiplicand and one bit of multiplier every time. The Impending addition is carried out by high-speed carry-save algorithm and the final product is obtained by employing fast adder – the number of partial products depends upon the number of multiplier bits. A 4x4 array multiplier is shown in Fig. 3. The structure of the full adder can be realized on FPGA. Each products can be generated in parallel with the AND function. Each partial product can be added with the sum of partial product which was previously generated by using the row of adders. The carry out will be shifted one bit to the left and then it will be added to the sum generated by the first adder and the newly generated partial product. The shifting would carry out with the help of Carry Save Adder and the Ripple carry adder or any fast adder can be used for the final stage. [10].



International Journal of Current Trends in Engineering & Technology
ISSN: 2395-3152
Volume: 02, Issue: 01 (JAN-FAB, 2016)

Figure 3: Array Multiplier

IV HARDWARE EFFICIENT MULTIPLIER

In section III we have discussed simple array multiplier, we need to cut down the area requirement of the multiplier, there are many possibilities one of them is using low power adder. In this design we have reduced the number the layers in array multiplication method by reusing the middle layer again and again. Also we have used low power adders in these layers to further reduce the power consumption. First we will discuss the different adders used in our design. In total 6 different adders are used to implement the design.

a) Simple full adder: The first adder we are using is the full adder. The logic diagram is shown in figure.

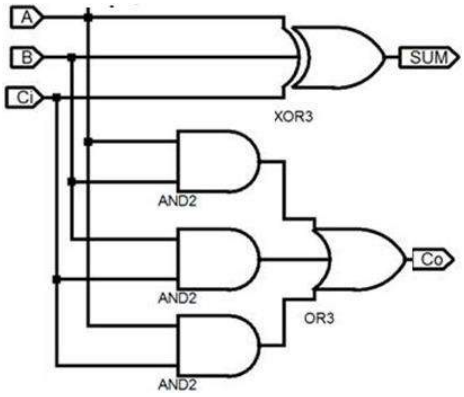


Figure 4: Simple Full Adder

b) *Half Adder:* The second adder is the simple half adder with two inputs and two outputs.

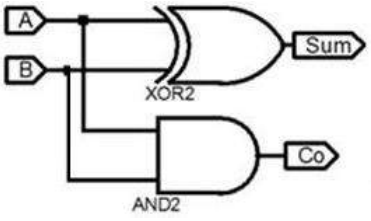


Figure 5: Half Adder

c) Row Column Adder Type – 1: This is the third adder of this design. This is the custom adder with four inputs and two outputs. The row column type one adder uses bypassing technique to reduce the dynamic power consumption. Whenever input r or input s becomes zero the control input of the tri-state buffer "BUFGCE" becomes zero and it tri-states its output which tri-states channel 1 of both the multiplexer and hence reduce dynamic power consumption. The output of multiplexer 1 mux is equal to the AND of p and q input, and the output of the second multiplexer count becomes 0. Figure 6 depicts the row column adder type1.

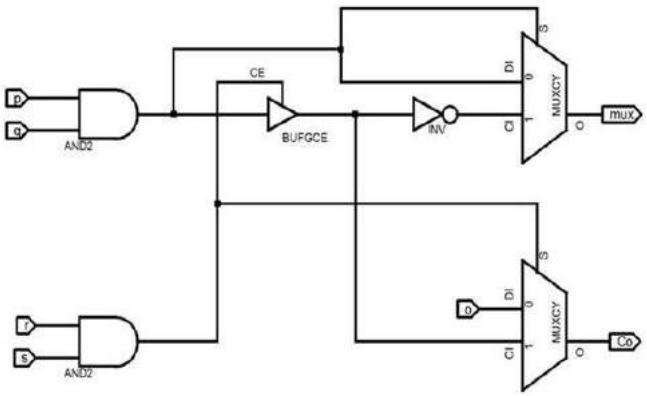


Figure 6: Row Column Adder Type – 1

d) Row Column adder type – 2: This is the fourth adder we have used in our design; this is again a custom adder for this design only. The adder has four inputs and two outputs. This is also a low power adder which tri-states the XNOR gate and OR and channel 1 of both the multiplexer the CE input of the tri-state buffer "BUFGCE" becomes 0. Figure 7 depicts the row column adder type 2.

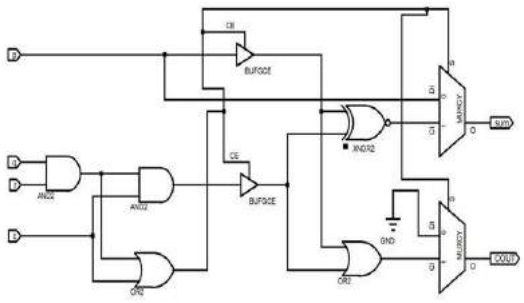


Figure 7: Row Column Adder Type – 2

International Journal of Current Trends in Engineering & Technology
ISSN: 2395-3152
Volume: 02, Issue: 01 (JAN-FAB, 2016)

e)Row Column Adder Type – 3: This is again a custom adder we have used in this design. The adder has five inputs and three outputs in it. Multiplexers are employed to implement this adder. Again bypassing technique is used to implement the design and to reduce dynamic power consumption. In this design first we have reduced the number of layers in array multiplier to only three. To implement this we have reused the middle layer of adders as shown in figure 9. The partial products are generated and stored in flip flops and then stored results are used again for next level of partial products. The process is repeated till the end of multiplication. Since the number of adders in the design is reduced by a large amount the area requirement also reduces. The number of storage elements basically flips flops increases, but these elements are available in abundance in any FPGA and require less area.

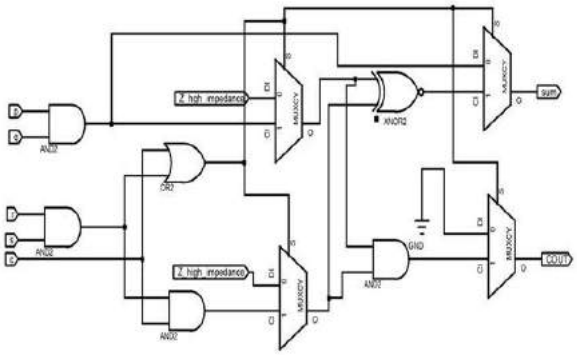


Figure 8: Row Column Adder Type – 3

So the area requirement reduces. The second level of optimization is the use of optimized adders in the design. These adders type 1 through type 3 are special adders with less power consumption. So our proposed design reduces the FPGA resource usage and power consumption of the design. Figure 9 shows the symbols used to represent different adders used in proposed hardware & power efficient multiplier. And figure 10 depicts the proposed hardware and power efficient multiplier with only three rows of full adders.

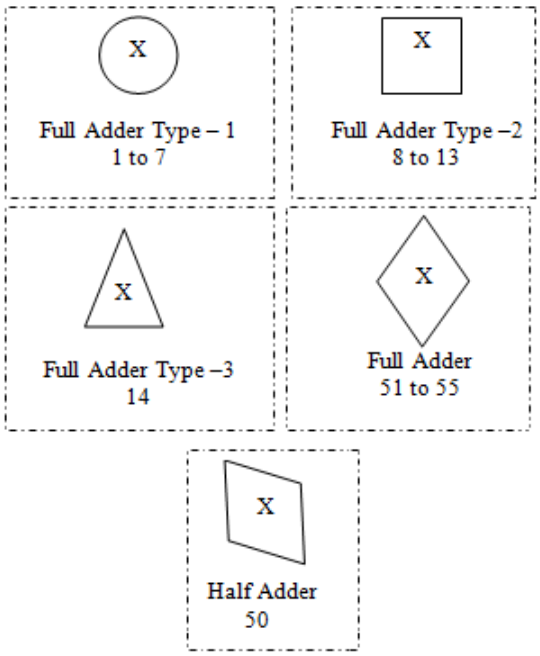


Figure 9: Representation of different adders

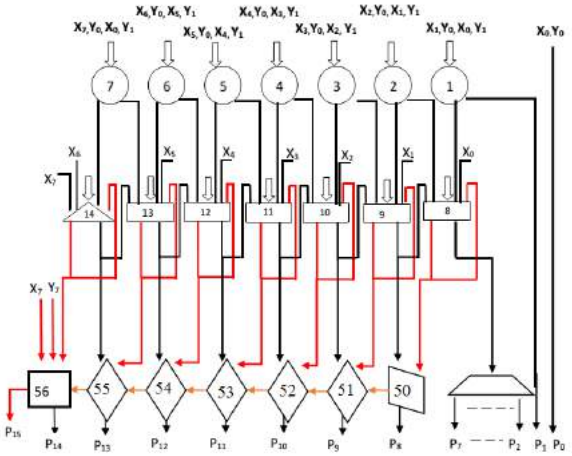


Figure 10: 8 x 8 Multiplier using hardware reuse method

V. RESULTS

In order to evaluate the performance of low power parallel multiplier, we implement all these designs on Spartan FPGA. Table 1 shows the Cell Usage summary. And table 2 shows the device utilization summary. The 8 x 8 multiplier shown in figure 10 is regular in shape so, we have designed a 8 x 8 multiplier using same technique and here we have presented synthesis results.

International Journal of Current Trends in Engineering & Technology
ISSN: 2395-3152
Volume: 02, Issue: 01 (JAN-FAB, 2016)

Table 1A: Device utilization Summary (8X 8 Multiplier)

Design(8 x 8)	*Number of LUT's*
Conventional Wallace multiplier	*165*
Confined Wallace multiplier	*157*
Proposed Hardware & Power efficient Multiplier	*80*

Device Utilization Summary (estimated values)		
Logic Utilization	**Used**	**Available**
Number of Slices	47	768
Number of Slice Flip Flops	49	1536
Number of 4 input LUTs	80	1536
Number of bonded IOBs	35	124
Number of GCLKs	3	8

Figure 12: Device utilization summary 8 x 8 multiplier - snapshot

Table 1B: Device utilization Summary (16 X 16 Multiplier)

Design(16 x 16)	*Number of LUT's*
Array Multiplier	*590*
Row Bypassed Multiplier	*921*
Column Bypassed	*941*
Proposed Hardware & Power efficient Multiplier	*179*

Device Utilization Summary		
Logic Utilization	**Used**	**Available**
Total Number Slice Registers	105	1,536
Number used as Flip Flops	16	
Number used as Latches	89	
Number of 4 input LUTs	179	1,536
Number of occupied Slices	117	768
Number of Slices containing only related logic	117	117
Number of Slices containing unrelated logic	0	117
Total Number of 4 input LUTs	179	1,536
Number of bonded IOBs	67	124

Figure 13: Device utilization summary 8 x 8 multiplier - snapshot

Table 1A and 1B depicts that the hardware requirement of different integer multiplier, the hardware requirements for the proposed hardware efficient multiplier is least compared to other designs. Table 2 shows the power consumption of our design and we have also compared it with available multipliers in literature. All the multipliers mentioned in table 3 are 16 x 16 multiplier, so we have implemented 16 x 16 hardware and power efficient multiplier and compared it with different array multipliers available in literature.

Table 2: Power Consumption report

Design	*Power Consumption*
Without Bypassing (16 x 16)[11]	*44mW*
Row Bypassed(16 x 16) [11]	*39mW*
Column Bypassed(16 x 16) [11]	*35mW*
Proposed hardware efficient multiplier(16 x 16)	*29mW*
Proposed hardware efficient multiplier(8 x 8)	*28mW*

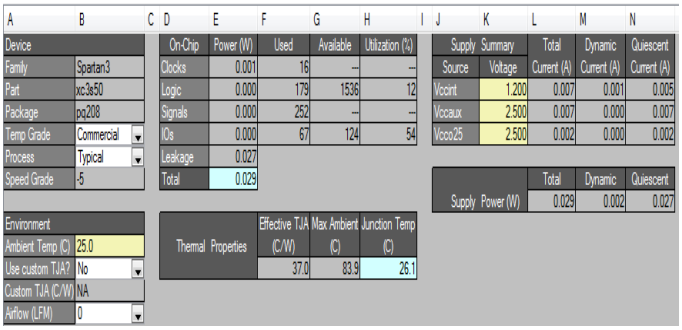


Figure 14: Power consumption summary (16 x 16) - snapshot

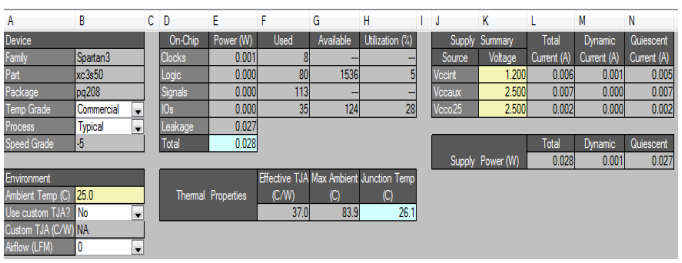


Figure 15: Power consumption summary (8 x 8) - snapshot

VI CONCLUSION

We have implemented a hardware efficient multiplier on Spartan FPGA using only three layers of adders and few storage elements. We succeeded in curtailing the area requirement and the power consumption of the multiplier compared to previously available designs as seen from table 2 and table 3 respectively. As we are using many storage elements (flip flops), clock gating can be used in future to further reduce the power consumption of the design.

REFERENCES

[1] T. Nishitani, "Micro-programmable DSP chip," 14th Workshop on Circuits and Systems, pp.279-280, 2001.

[2] V. G. Moshnyaga and K. Tamaru, "A comparative study of switching activity reduction techniques for design of low power multipliers," IEEE International Symposium on Circuits and Systems, pp.1560-1563, 1995.

[3] Wu, "High performance adder cell for low power pipelined multiplier," IEEE International Symposium on Circuits and Systems, pp.57–60, 1996.

[4] T. Ahn and K. Choi, "dynamic operand interchange for low power," Electronics Letters, Vol. 33, no. 25, pp.2118- 2120, 1997.

[5] J. Choi, J. Jeon and K. Choi, "Power minimization of functional units by partially guarded computation," International Symposium on Low-power Electronics and Design, pp.131-136, 2000.

[6] J. Ohban, V. G. Moshnyaga, and K. Inoue, "Multiplier energy reduction through bypassing of partial products," IEEE Asia-Pacific Conference on Circuits and Systems, pp.13–17, 2002.

[7] M. C. Wen, S. J. Wang and Y. M. Lin, "Low power parallel multiplier with column bypassing," IEEE International Symposium on Circuits and Systems, pp.1638- 1641, 2005.

[8] G. N. Sung, Y. J. Ciou and C. C. Wang, "A power-aware 2- dimensional bypassing multiplier using cell-based design flow," IEEE International Symposium on Circuits and Systems, pp.3338-3341, 2008.

[9] J. T. Yan and Z. W. Chen, "Low-power multiplier design with row and column bypassing," IEEE International SOC Conference, pp.227-230, 2009.

[10] Jin-Tai Yan and Zhi-Wei Chen, "low-power multiplier design with row and column bypassing", department of computer science and information engineering, chung-hua University.

[11] Tushar V. More and Dr. R.V. Ksirsagar "Design of low power column bypass multiplier using FPGA" 2011 IEEE.