IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 3 (Nov. - Dec. 2013), PP 64-70
www.iosrjournals.org

VLSI Implementation of High Speed & Low Power Multiplier in FPGA

Prashant Kumar Sahu [1], Asst. Prof. Nitin Meena ,[2] Prof. Shweta Singh[3]
[1,2,3](EC, IES College, Bhopal / RGPV, India)

Abstract : *We known that different multipliers consume most of the power in DSP computations, FIR filters. Hence, it is very important factor for modern DSP systems to built low-power multipliers to minimize the power dissipation. In this paper, we presents high speed & low power Row Column bypass multiplier design methodology that inserts more number of zeros in the multiplicand thereby bypass the number of zero in row & Column as well as reduce power consumption. The bypassing of zero activity of the component used in the process of multiplication, depends on the input bit data. This means if the input bit data is zero, corresponding row and column of adders need not be addition & transfer bit in next row and column adder circuit. If multiplicand having more zeros, higher power reduction can be achieved. At last stage of Row & column bypass multiplier having ripple carry adder which are increase time to generate carry bit to transfer next adder circuit. To reduce this problem by using Carry bypass adder in place of ripple carry adder, then new modification of Row &column multiplier having high speed in comparison to simple row & column bypass multiplier, , the experimental results show that our proposed multiplier reduces power dissipation & High speed overhead on the average for 4x4, 8x8 and 16x16 multiplier.*

Keywords: *Low Power, Row & Column bypass Multiplier, Carry bypassing techniques, FPGA, Xilinx .*

I. INTRODUCTION

Multipliers are key components of many high performance systems such as filters, microprocessors, digital signal processors, etc. A system's capability is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Multiplier required more hardware resources and processing time then addition and subtraction. In fact 8.74% of all instructions in a typical processing unit is a multiplier. In computer a typical central processing unit devotes a considerable amount of processing time in arithmetic operation, particularly multiplication operation. Multiplication is an important fundamental arithmetic operation. Multiplication based operation such as multiply and accumulate (MAC) are currently implemented in many digital signal processing application such as convolution, Fast Fourier Transform (FFT), Filtering and in microprocessor in its arithmetic and logic unit. Since multiplication dominants the execution time of most DSP application, there is a need to high speed multiplier. Currently multiplication time is still that dominant factor in determining the instruction cycle time of DSP chip. The multiplier is fairly large block in of computing system.

In the past many novel ideas for multiplier have been proposed to achieve high performance. Higher throughput arithmetic operations are important to achieve the desire performance in many real time signal and image processing application. On of the key arithmetic operation in such application is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. Reducing the delay and power consumption is very essential requirement for many applications.

When we multiply binary numbers there are lots of zeros, whenever any of the multiplicand bit is zero and it gets multiplied by any bit it produces zero . In Braun's multiplier the resultant zeros are added and the corresponding full adders work and consume power. So Bypass techniques are employed to deactivate the full adders when multiplying with zeros. This will reduce the power consumption.

II. Previous Work And Related Research

The architecture view of a 4*4 Standard Braun multiplier is as shown in Fig.:1.Braun Multiplier is a simple parallel multiplier generally called as CSM (carry save multiplier). This parallel multiplier is used to perform the unsigned bit multiplication. The Braun multiplier structure makes of the array of array of AND gates and full adders. To implement the n*n multiplier n(n-1) full adders and n2AND gate are required[2]. The delay introduced by the Braun's multiplier depends on the delay of the full adders and also on the delay of the final adder in the last stage which is a ripple carry adder. The dynamic power dissipation of the multiplier resulting from the switching activities can be reduced via bypassing techniques like Row bypassing, column bypassing and Reversible Logic techniques [3].

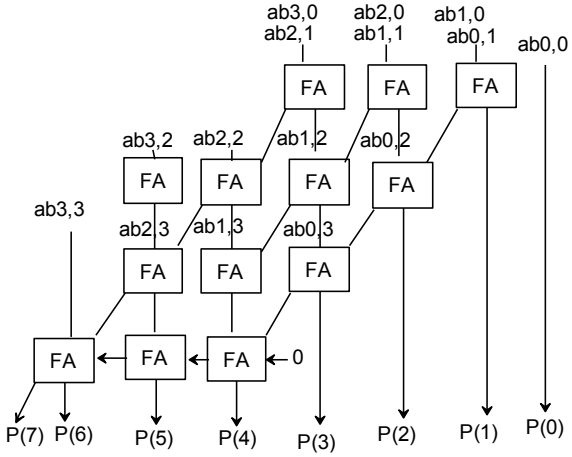


Fig. 1. Array multiplier

In Column Bypassing multiplier, if the multiplier bit ai is zero, then the addition operations in the i-th Column can be bypassed, thus providing directly (i-1)-th Column outputs directly to out at last satge. Thus, the switching activities in the i-th Column reduced and hence the power dissipation. The 4 bit Braun Multiplier with Column Bypassing technique is illustrated in Fig 2.

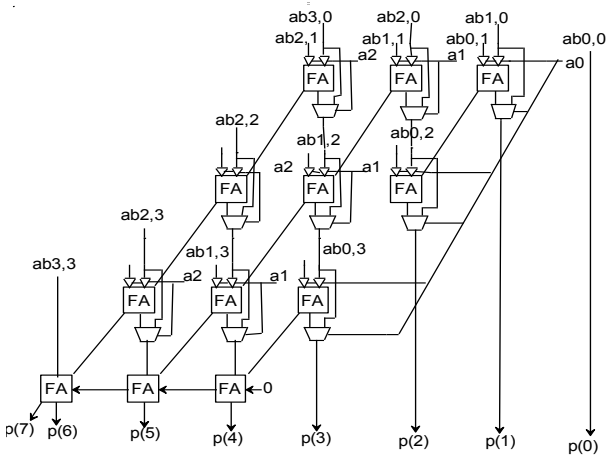


Fig. 2. Column bypass multiplier

In design of Reversible logic, has used as one of the most application approaches for the power optimization with its application in low power requirement of VLSI circuit design. Modification of Reversible logic circuits design have theoretically zero internal power dissipation because they do not lose information, the classical set of gates such as AND, OR, and EXOR are not reversible. TSG gate is used in place of Full Adder , capable of implementing all Boolean functions and can also work singly as a reversible Full Adder . The 4 bit Reversible logic Multiplier is illustrated in Fig 3.

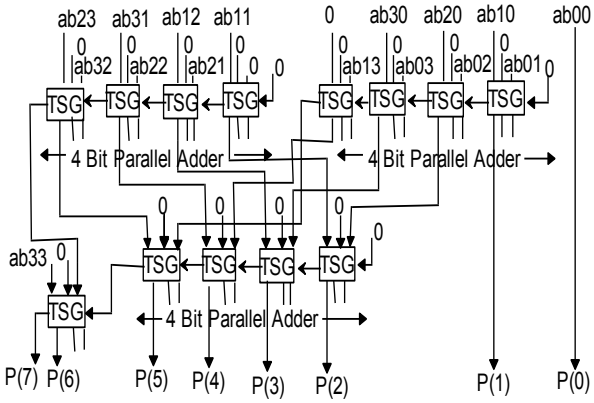


Fig. 3. Reversible Multiplier

III. PROPOSED WORK

In all the multipliers discussed above, According to the bypassing features in the previous low-power multipliers, the addition operations in the (i+1)-th column or the j-th row can be bypassed for the power reduction if the bit, ai, in the multiplicand is 0 or the bit, bj, in the multiplier is 0. On the other hand, the extra correcting circuits in the row-bypassing multiplier are applied to add the bypassed carry results into the multiplication result. Therefore, the addition operation in the (i+1, j) FA can be bypassed in our proposed bypassing multiplier if the product, aibj, is 0 and the carry bit, ci,j-1, is 0, that is, as the product, aibj, is 1 or the bit, ci,j-1, is 1, the addition operation in the (i+1, j) FA can be executed. It is known that the (i+1, j) FA only executes the A+1 addition as the product, aibj, is 1 and the bit, ci,j-1, is 0, or the product, aibj, is 0 and the bit, ci,j-1, is 1. On the other hand, the (i+1, j) FA only executes the A+2 addition as the product, aibj, is 1 and the bit, ci,j-1, is 1. Hence, the carry bit in the (i+1, j) FA can be replaced by the AND result of the product, aibj, and the bit, ci,j-1, and the (i+1, j) FA, n > j > 1, can be replaced with the half adder, A+B+1, in the proposed low-power multiplier. For the HAs in the first row of CSAs, the HAs are also replaced with the incremental adder, A+1, because of the 2-dimensional bypassing process. Besides that, each simplified adder, A+1, in the CSA array is only attached by one tri-state buffer and two 2-to-1 multiplexers and each simplified adder, A+B+1, in the CSA array is only attached by two tri-state buffers and two 2-to-1 multiplexers. In Fig. 5, the logical circuits of the different adders, A+1, A+B and A+B+1 are shown. A 8X8 Braun multiplier with row and column bypassing can be illustrated in Fig. 5.

Example- 11101011 8-bits

10101111 8-bits

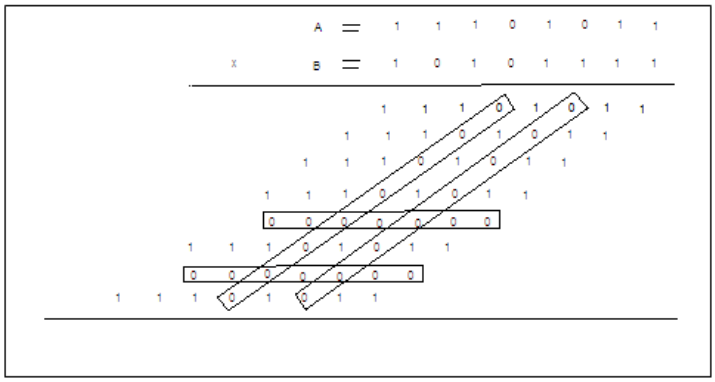


Fig. 4. A 8×8 multiplication of bit ai =bj= 0 corresponds to Row & column i = 0,j=0

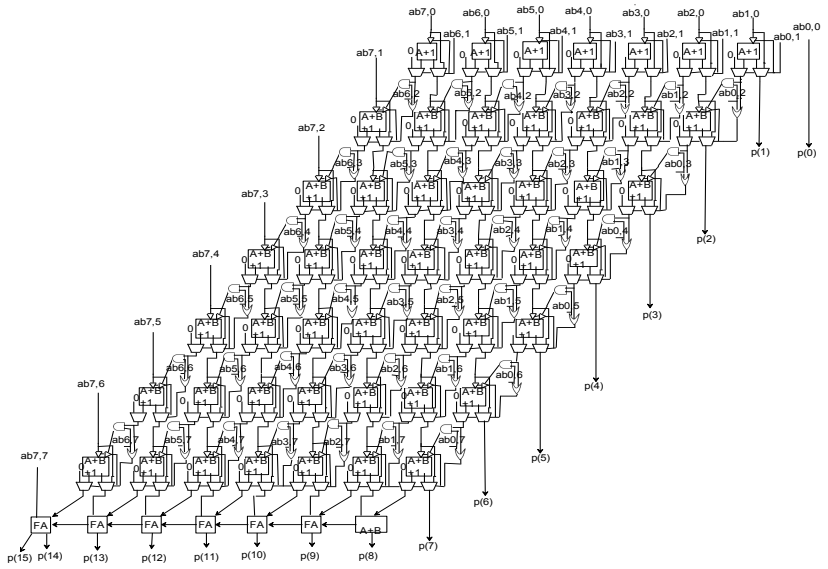


Fig. 5. Row & Column Bypass Multiplier

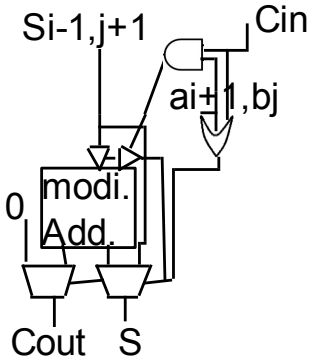


Fig. 6. Adder cell (CA)

But the last stage of Row & Column Bypass Multiplier has been used a ripple carry adder. The main drawback of this multiplier is that because of the ripple carry adder in the last stage glitching problem occurs and also the delay of the multiplier will be high. The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage. Even though this is a simple adder and can be used to add unrestricted bit length numbers, it is however not very efficient when large bit numbers are used. One of the most serious drawbacks of this adder is that the delay increases linearly with the bit length. The worst-case delay of the RCA is when a carry signal transition ripples through all stages of adder chain from the least significant bit to the most significant bit, which is approximated by:

$$t = (n-1)\,tc + ts \qquad \text{Eq (1)}$$

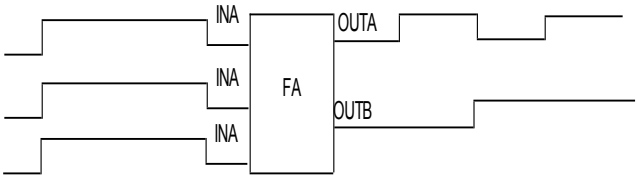


Fig. 7: Glitching on an adder

where tc is the delay through the carry stage of a full adder, and ts is the delay to compute the sum of the last stage. The delay of ripple carry adder is linearly proportional to n, the number of bits, therefore the performance of the RCA is limited when n grows bigger. The advantages of the RCA are lower power consumption as well as compact layout giving smaller chip area. The design schematic of RCA is shown in Figure (2).

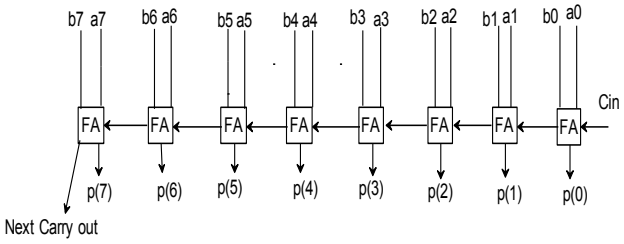


Fig. 8. Ripple Carry adder

Ripple Carry adder is a combination of several full adders. The carry input of full adder is dependent on the carry output of the previous full adder, and the present full adder should wait until the previous full adder has completed producing the outputs. Hence, the delay is more for the ripple carry adder. If the number of bits increases, then the delay also increases more for a ripple carry adder.

The delay and power of the multiplier can be reduced by replacing the ripple carry adder with fast adders like Carry bypass adder.

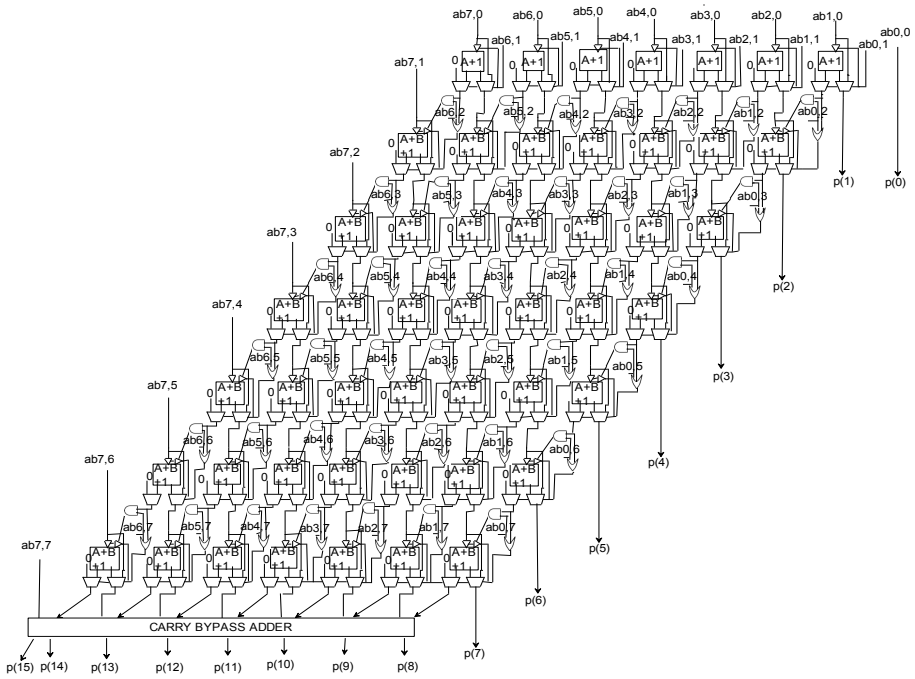


Fig. 9. Modify Row & Column Bypass Multiplier

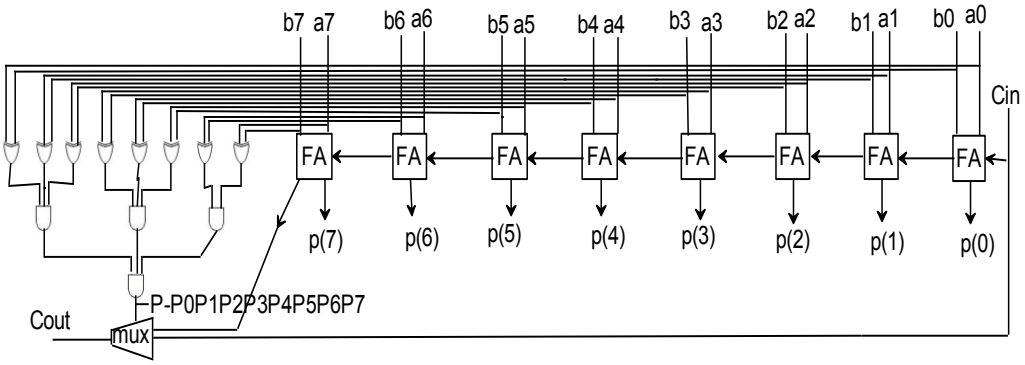


Fig .10. Carry bypass Adder

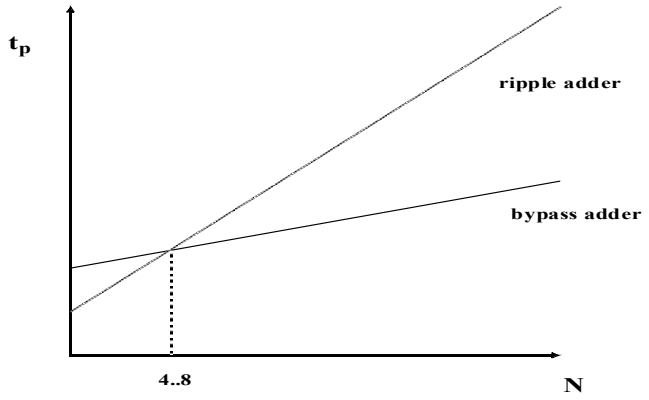


Fig. 11. Carry Ripple versus Carry Bypass

IV. Experimental Results

A. Analysis On Area Overhead of Slices :-

In all the multiplier designs, to give us an idea of the area that will be used, the number of transistors used in the circuit is counted since this directly affects the area. The results are shown in Table 1.

S.N.	Multiplier	No. Of Slice Uses
1	Simple Multiplier	76
2	Bypass Row Multiplier	86
3	Bypass Column Multiplier	84
4	Bypass Row & Column Multiplier	85
5	Modify Bypass Row & Column Multiplier	87

Table 1. No of Slices use in different multiplier (8x8)

B. Analysis Power Dissipation :-

The average dynamic power of each test case were measured and tabulated in Table 2(mw).

S.N.	Multiplier	Dynamic	Quiescent	Total(mw)
1	Simple Multiplier	136.85	15.13	151.99
2	Bypass Row Multiplier	137.46	15.14	152.60
3	Bypass Column Multiplier	133.86	15.10	148.96
4	Bypass Row & Column Multiplier	133.32	11.90	145.22
5	Modify Bypass Row & Column Multiplier	133.88	11.90	145.79

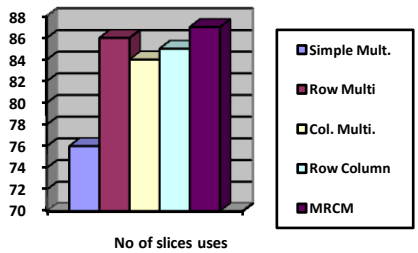
Table 2. Power dissipation in different multiplier (8x8)

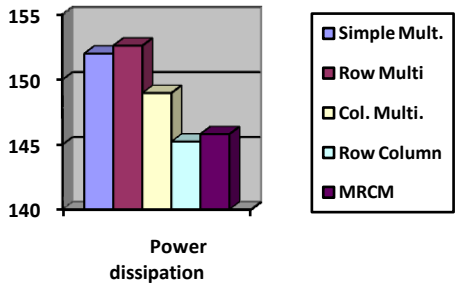
C. Propagation Delay

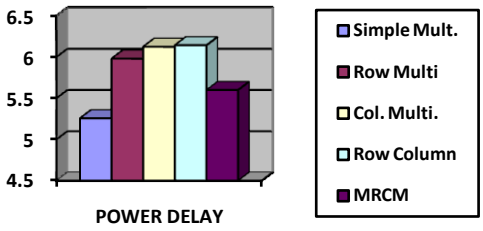
For the propagation delay, the critical path delay is measured which occurs in the middle of the array. The test cases used travel along this path. The propagation delay measured for each of the multiplier architecture is shown in Table 3.

S.N.	Multiplier	Power Delay
1	Simple Multiplier	5.256
2	Bypass Row Multiplier	5.981
3	Bypass Column Multiplier	6.125
4	Bypass Row & Column Multiplier	6.143
5	Modify Bypass Row & Column Multiplier	5.60

Table 3. Delay in different multiplier 8x8 (ns)







V. Conclusion

We have introduced low-power multiplier design and High speed bypass method in this paper. Our proposed modify Row & Column Bypass multipliers are useful for all multipliers. To reduce the power and increase speed, From the obtained results in Xilinx and Cadence, it can be concluded that if the multiplier is to be used for high – speed applications, then a Carry Bypass adder can be used with the multiplier design but the area as well as the dynamic power increases.

Vi. Future Work

In this paper, the proposed work has been done for 8*8 and 16*16 bit unsigned multipliers. The bypassing techniques with the architectural modifications can also be applied to signed array multiplier architectures. Proposed multiplier using conventional Adder (Full Adder) can be replaced by new Fast Adder & change of Multiplication algorithm. In this paper, the proposed work has been done for using tri state buffer which having floating point problem ,can be modify by using transistors.

References

[1] Prashant kumar Sahu and Asst. Prof Nitin Meena, "Comparative Study Of Different Multiplier Architecture", IJETT, Volume 4 Issue ,pp. 4293,10- Oct 2013.
[2] M. C. Wen, S. J. Wang and Y. M. Lin, "Low power parallel multiplier with column bypassing," IEEE International Symposium on Circuits and Systems, 2005.
[3] Ming-Chen Wen, Sying-Jyan Wang, and Yen-Nan Lin, "Low power parallel multiplier with column-bypassing," IEEE International Symposium on Circuits and Systems, ISCAS, vol. 2, pp.1638-1641, May 2005.
[4] J. Ohban, V. G. Moshnyaga, K. Inoue, "Multiplier energy reduction through Bypassing of partial products", IEEE Asia-Pacific Conference on Circuits and Systems, pp.13-17, 2002.
[5] Chang-Young Han, Hyoung-Joon Park, and Lee-Sup Kim, "A low-power array multiplier using separated multiplication technique," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Volume 48, Issue 9, pp.866-871, Sept. 2001.
[6] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs, Oxford University Press, 2000.
[7] C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," IEEE Trans. Comput., vol. C-22, pp. 1045-1047, Dec. 1973.

.