Design of an optimized multiplication technique by using ColumnBypass Multiplier

[1]Thara Rajendra Rao, [2]Dipal Patel

Abstract— **Multiplication is one of the essential operations in Digital Signal Processing (DSP) like Fast Fourier Transform(FFT), Digital filters etc. Our goal is to analyze and compare various adders and multiplication schemes for high speed and low power operations. Since the various filter designs found in DSP applications require computationally efficient multiply and accumulate operations so the blocks with the desired characteristics have to be chosen carefully. Various techniques have been proposed to design multipliers which are efficient in terms of performance, low power consumption area. There have been many algorithms proposals in literature to perform multiplication, each offering different advantages and having trade-offs in terms of speed, circuit complexity, area and power consumption. In this paper we have presented the analysis of various adders and multipliers and by doing comparative study we found that the column bypassing multiplier is an improvement over various multipliers. The dynamic power and delay of bypassing multiplier can be reduced using different adders**.

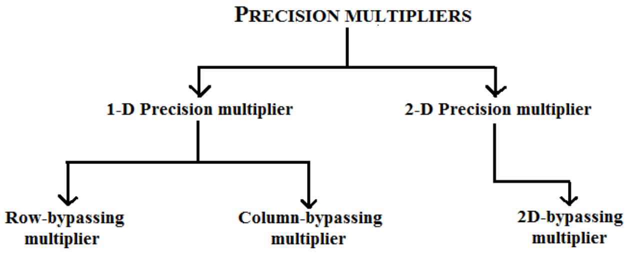
Index Terms— **Area efficient, column bypass multiplier, reduced power consumption,high speed.**

I. INTRODUCTION

The multiplier is fairly large block of a computing system. The amount of circuitry involved is directly proportional to the square of its resolution i.e a multiplier of size n bits has n^2 gates. For multiplication algorithms performed in DSP applications latency and throughput are the two major concerns from delay perspective. Multiplier is not only a high delay block but also a major source of power dissipation. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important part in low-power VLSI system design In recent years, the impact of pervasive computing and the internet have accelerated this trend. The applications for these domains are typically run on battery-powered embedded systems. The resultant constraints on the energy budget require design for power as well as design for performance at all layers.[2]

II. PROPOSED METHODOLOGY

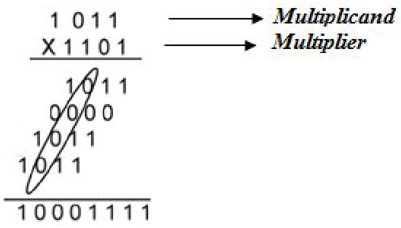
Precision multipliers can be broadly classified into two types, as one dimensional and two dimensional precision multipliers. These multipliers bypass the redundant rows and columns or both in order to save power. Let us have a look on figure , for better understanding of the basic classification of precision multipliers.



These precision multipliers allow transition activity between the successive adders in the array, only if necessary. If predicted unnecessary,[3] these precision multipliers with the help of tri-state buffers restrict the entry of input signal into the adder circuit, it bypasses the result of the of the former adder circuit to the next adder circuits input, present in the array of adders with the help of 2:1 multiplexers.

A) Column-bypassing multiplier :

This approach reduces power dissipation, by bypassing the complete column in the hardware of the multiplier which are redundant. In this approach, we find that the whole circuitry of the multiplier is much simpler and also occupies lesser area when compared to that of the Row bypassing multipliers. This by-passing scheme also reduces switching activity, resulting in less power consumption.[6]This is an arbitrary 4x4 multiplication indicating the redundant column which could be by-passed. As the 3rd LSB of the multiplicand bit is a zero, it results in making the 3rd column of partial products to be zero. Hence,this column is bypassed.



Consider the multiplication shown in Figure 3.6, which executes 1010×1111. It is to be noted that the Y0 and Y2 bits of the multiplicand are zero. In the first and third diagonals (enclosed by dashed lines), two out of the three input bits are 0 in each adder circuit i.e. the "carry" bit from its upper right FA, and the partial product Yi.Xj . As a result, the output carry bit of the FA is 0, and the output sum bit is simply equal to the third bit, which is the "sum" output of its upper FA. In general we can say that when Yi is 0, the operations in the corresponding column can be disabled since all the outputs are known.

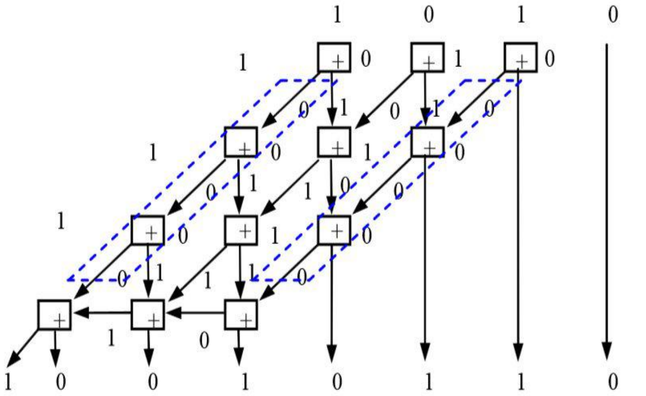


Fig. 4x4column bypass multiplier

Column-bypassing multiplier being a parallel array multiplier, its adder circuits are represented in array form. In the (n–1)×(n–1) array, let FAi,j be the full adders located in row i and column j, where 0 ≤ i, j ≤ n–2. FA0,0 is the adder at the upper-right corner

The reason for column bypassing can be established by the following Theorem.

Theorem 1: When Yj = 0, the output of a column j adder cell FAi,j can be specified as follows

(1) The output carry bit is 0.

(2) The sum bit is equal to the sum bit of FAi–1,j+1.

Proof : This theorem is proved by induction.

I. Consider row 0. Note that, in row 0, there are only two bits to be added. Adder FA0,j

carries out Yj X1 + Yj+1 b0. If Yj = 0, then the output carry bit must be zero, and the out

sum bit is equal to Yj+1 X0

II. In row i+1, the inputs of FAi+1,j are carry bit from FAi,j, sum bit from FAi,j+1, and the

partial product Yj Xi+1. Since Yj = 0, two out of the three inputs are 0, and the output

sum bit is equal to the input sum bit sent by FAi,j+1.

III. Assume that the theorem holds for row i.

According to Theorem 1, when Yj = 0, the operations in column j can be ignored and thus the

full adders can be disabled since the outputs are known.

While designing the circuit for column-bypassing multiplier, it is essential to set the carry

outputs to 0 in the bottom of the CSA array, otherwise, the corresponding FA's may not produce the correct outputs since their inputs are disabled. This is done by adding an AND gate at the outputs of the last-row CSA adders as shown in the following circuit diagram of Column-bypassing multiplier.

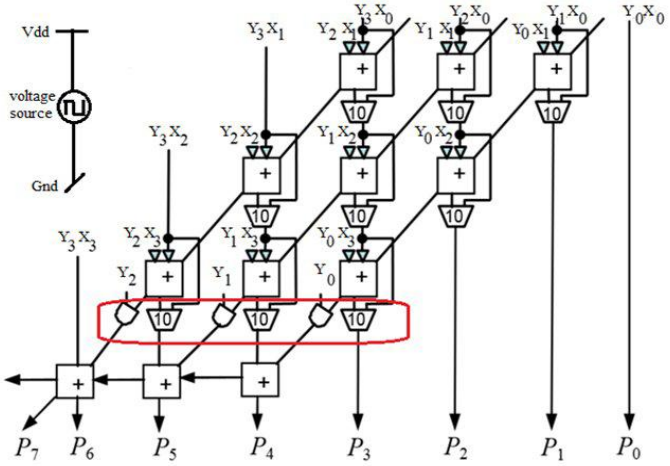


Fig. Generic architecture

The design of any multiplier, completely depends on the internal structure of the adder cell. In column-bypassing multiplier the internal structure of the adder cell is designed in the manner that it is capable of by-passing certain columns, whenever it detects the corresponding bits of multiplicand to be zero. It uses additional two Tri-state buffers and one multiplexer, compared to Standard Braun's multiplier, in its internal adder structure to bypass a column.

III. SIMULATION RESULTS

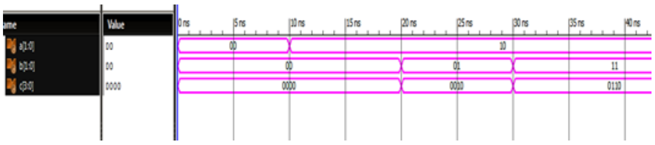

```
Name    Value
a[1:0]  00      00 | 10
b[1:0]  00      00 | 01 | 11
c[3:0]  0000    0000 | 0000 | 0110
```

Fig. 2x2 multiplier

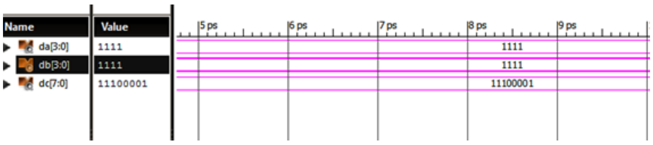

```
Name      Value
da[3:0]   1111       1111
db[3:0]   1111       1111
dc[7:0]   11100001   11100001
```

Fig. 4x4 multiplier

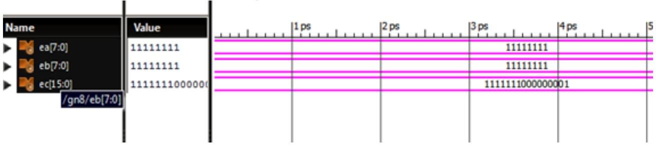

```
Name      Value
ea[7:0]   11111111         11111111
eb[7:0]   11111111         11111111
ec[15:0]  111111100000(    1111111000000001
/gn8/eb[7:0]
```

Fig. 8x8 multiplier

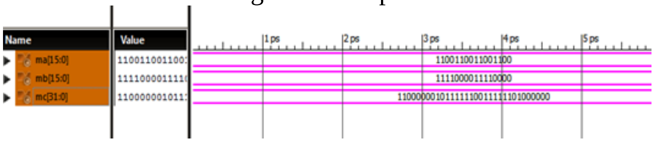

```
Name      Value
ma[15:0]  110011001100:    1100110011001100
mb[15:0]  111100001111(    1111000011110000
mc[31:0]  110000001011:    11000000101111110011111101000000
```

Fig. 16x16 multiplier

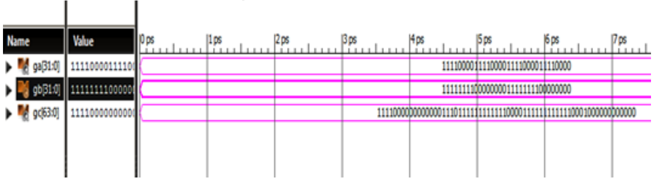

```
Name      Value
ga[31:0]  111100001111(    11110000111100001111000011110000
gb[31:0]  111111110000(    11111111000000001111111100000000
gc[63:0]  111100000000(    1111000000000000111011111111111100001111111111111000100000000000
```

Fig. 32x32 multiplier

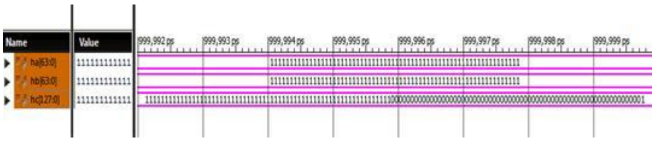
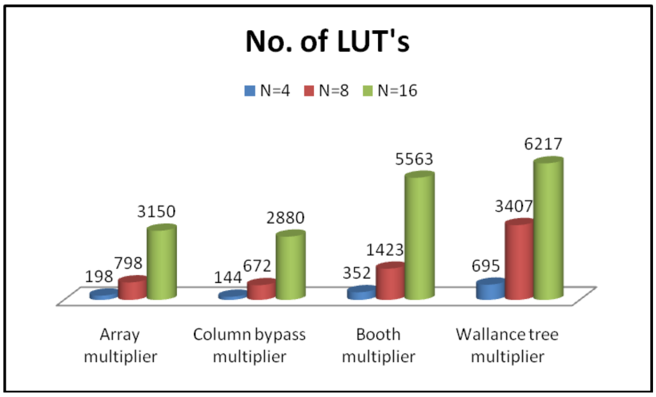
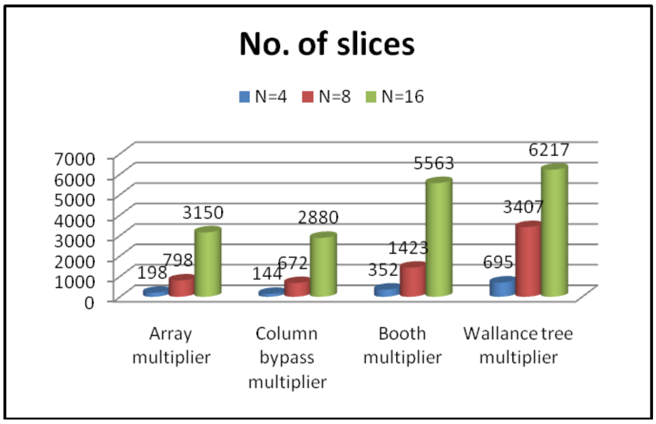
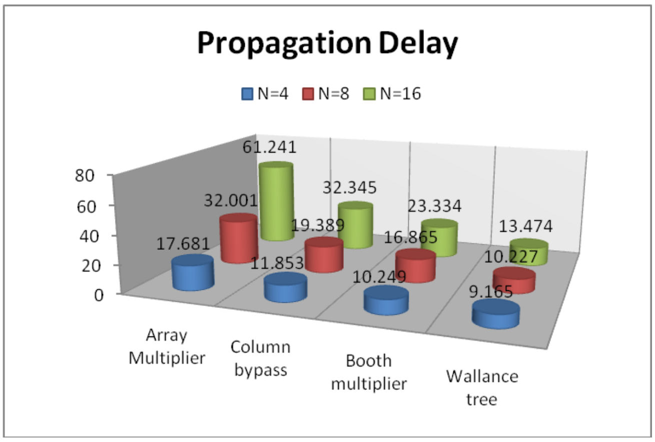


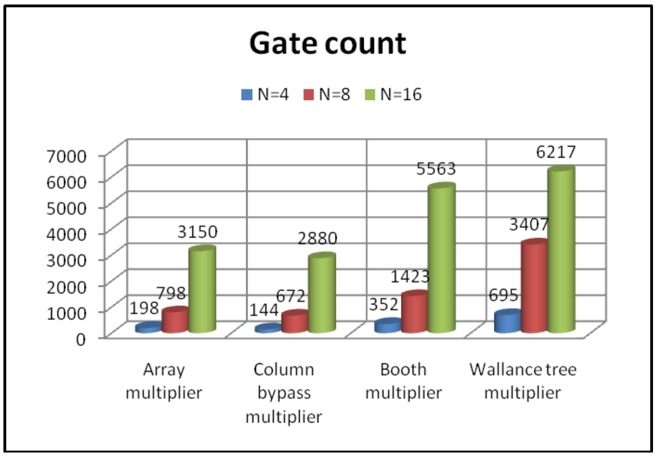
Fig. 64x64 multiplier

IV. ANALYSIS









V. CONCLUSION

The multiplier is the most important block in processors. Thus this project Design of bypassing multiplier , helps to improve the efficiency of the multiplier through bypassing techniques. This technique uses less area and power, as the MFA of the column bypassing multiplier uses only one multiplexer

VI. REFERENCES

[1] C. R. Baugh and B. A. Wooley, “A Two’s Complement Parallel Array Multiplication Algorithm,” IEEE Transactions on Computers, vol. 22,pp. 1045–1047, December 1973.

[2] M. Själander, “HMS Multiplier Generator,” http://www.sjalander.com/ research/multiplier, February 2008.

[3] 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.

[4] Wallace.C.S, “A suggestion for a fast multiplier,” IEEE Trans. Electron.Comput., vol. 13, pp. 14–17, Feb. 1964

[5] Benini.L, Micheli.G.D, Maci.Ai, E.Macii, Poncino.M, and Scars.Ri, “Glitching power minimization by selective gate freezes,” IEEE.

[6] H. Eriksson, P. Larsson-Edefors, M. Sheeran, M. Själander, D. Johansson, and M. Schölin, “Multiplier Reduction Tree with Logarithmic Logic Depth and Regular Connectivity,” in IEEE International Symposium on Circuits and Systems, May 2006.

[7] K.H. Tsoi, P.H.W. Leong, "Mullet - a parallel multiplier generator," fpl, pp.691-694, International Conference on Field Programmable Logic and Applications, 2005., 2005

[8] S. Tahmasbi Oskuii, P. G. Kjeldsberg, and O. Gustafsson, “Transition activity aware design of reduction-stages for parallel multipliers,” in Proc. 17th Great Lakes Symp. On VLSI, March 2007, pp. 120–125.