A Low Power Row and Column Compression for High-Performance Multiplication on FPGAs using Fast Adder

Jaswant singh[1] Gaurav Shamra[2]
[1,2]Department of Electronics and Communication Engineering
[1,2]Mewar University

Abstract— Digital system design in a remarkable and emerging field now days. Users are using digital devices for almost each and everything in daily life like calculator for calculation, Digital cameras for photo shot and video recording, Mobiles for communication, Computer to connect all over the world etc. In fact, increasing demand for manageable digital electronics products for computing and communication, as well as for other applications, has necessitated longer battery life, lower weight, high speed and lower power consumption. However, the two design criteria are often in conflict by improving one particular aspect of the design constrains the other. The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. In majority of digital signal processing (DSP) applications, multiplication and accumulation are the most critical operations.

Key words: Low Power Row, FPGA, CLA

I. INTRODUCTION

Computation operations like fast parallel multiplication using adder trees are present in many parts of a digital system or digital computer, especially in signal processing, high-speed circuits, graphics and scientific computation. Examples of such are graphic processor, digital signal processors, communication or code compression. To speed up addition is a very important part for computation.

Parallel computation is always providing the fast results and thus for fast multiplication, parallel multiplier is preferred such as Array multiplier, Booth multiplier using recoding bits, Modified Booth algorithm (MBE). The computation time taken by the array multiplier is comparatively less because the partial products are calculated independently in parallel. The delay associated with the array multiplier is the time taken by the signals to propagate through the gates that form the multiplication array

There are many tree structure like Wallace adder tree [1], CSA tree, over turn stair tree [2] and some other kinds of adder trees are mentioned in [3]-[7] for fast parallel multiplication. In this thesis, Wallace tree is used as the tree structure because it is suitable for implementation.

II. METHODOLOGY

The Wallace tree multiplier is considerably faster than a simple array multiplier because its height is logarithmic in word size, not linear. However, in addition to the large number of adders required, the Wallace tree's wiring is much less regular and more complicated. As a result, Wallace trees are often avoided by designers, while design complexity is a concern to them. The Wallace tree multiplier is a high speed multiplier. The summing of the partial product bits in parallel using a tree of carry-save adders became generally known as the "Wallace Tree".

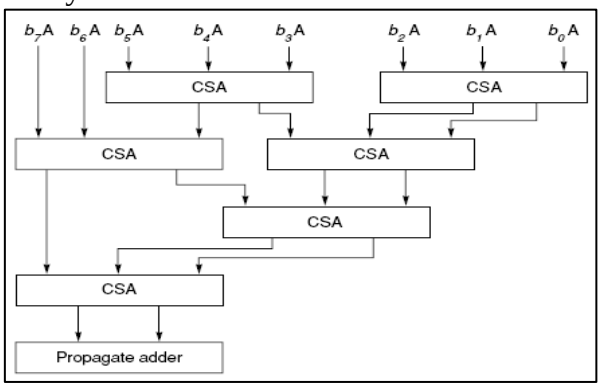


Fig. 1: Wallace tree adder

Three step processes are used to multiply two numbers.

- Formation of bit products.
- Reduction of the bit product matrix into a two row matrix by means of a carry save adder.
- Summation of remaining two rows using a faster Carry Look Ahead Adder

A. (CLA)

Wallace tree has been used in this thesis in order to accelerate multiplication by compressing the number of partial products. This design is done using half adders; Carry save adders and the Carry Look Ahead adders to speed up the multiplication. In order to design an n-bit Wallace tree Multiplier an algorithm was derived from the flow diagram developed below. The flow diagram below shows the intermediate state reductions of the multipliers are being done by Carry save adders and half adders while the final step additions being done by a Carry Look Ahead Adder. After generating the flow diagram for 8-bit × 8-bit and hence we designed a Modified WALLACE TREE which is proposed in this thesis.

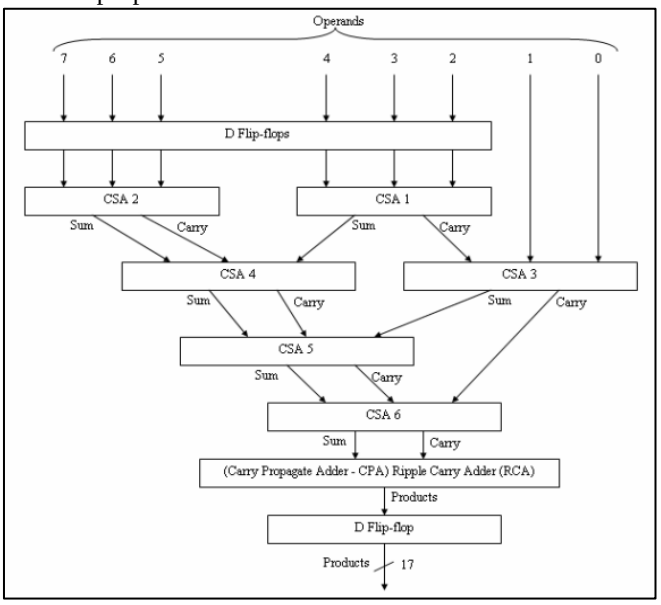


Fig. 2: Modified Wallace tree adder

III. RESULTS AND DISCUSSION

Delay(ns)	Ripple Carry Adder	Modified Ripple Carry Adder	Carry Look ahead Adder	Modified Carry Look ahead Adder	Carry Select Adder	Modified Carry Select Adder
Row Bypassing	28.369	23.252	24.365	22.618	21.369	18.374
Column Bypassing	23.345	18.267	21.365	18.267	16.354	13.951
Row& Column Bypassing	30.253	25.646	26.358	23.268	23.645	20.467

Table 1 Comparison of Delay of bypassing 8x8 multiplier on Spartan - 3E (xc3s500e-4fg320).

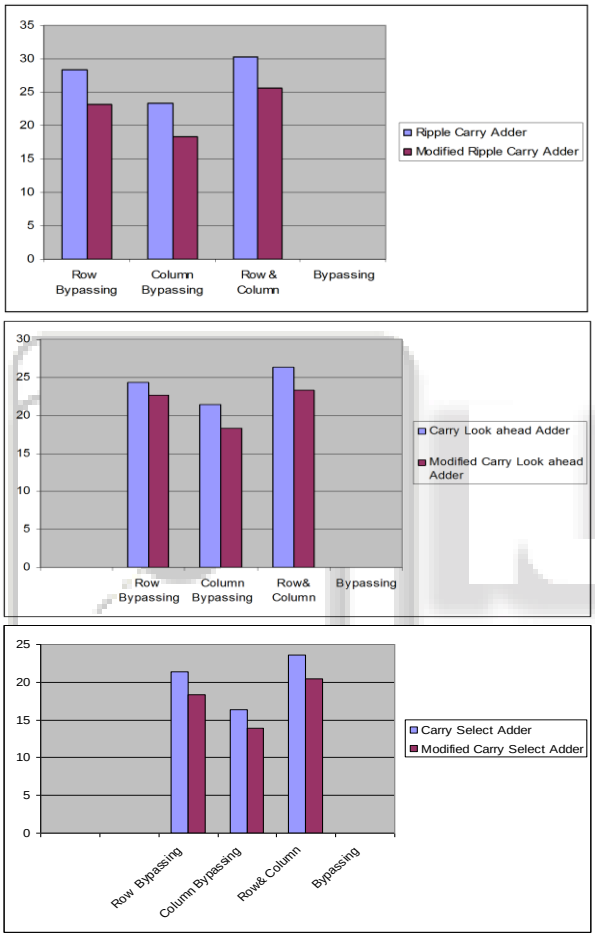
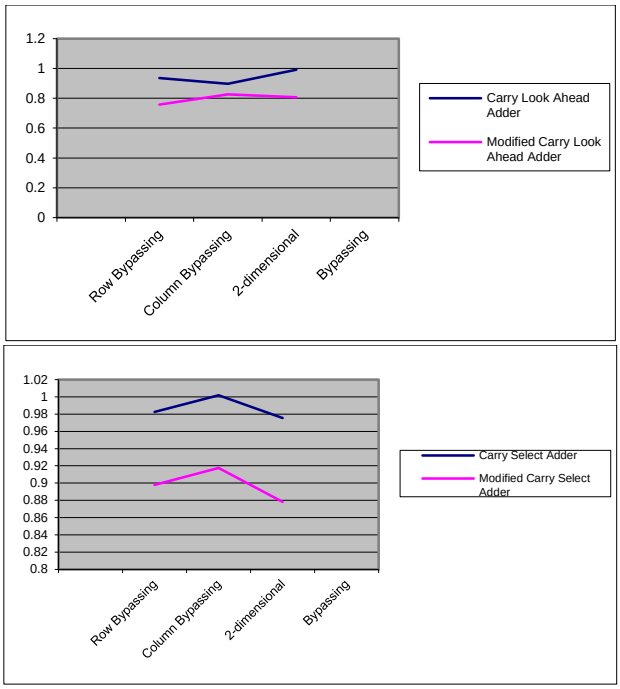


Fig.: The delay of8x8 row bypassing, column bypassing and row and column bypassing, for different adders

Power(mW)	Carry Look Ahead Adder	Modified Carry Look Ahead Adder	Carry Select Adder	Modified Carry Select Adder	Ripple Carry Adder	Modified Ripple Carry Adder
Row Bypassing	0.9364	0.7565	.9824	0.8977	.8235	0.775
Column Bypassing	0.9835	0.8258	1.002	0.9174	.8356	0.8065
2-dimensional Bypassing	.9925	0.8060	.9753	0.8781	.7526	0.7312

Table 2: Comparison of dynamic Power a of different 8x8 multipliers on Spartan 3E (xc3s500e-4fg320)



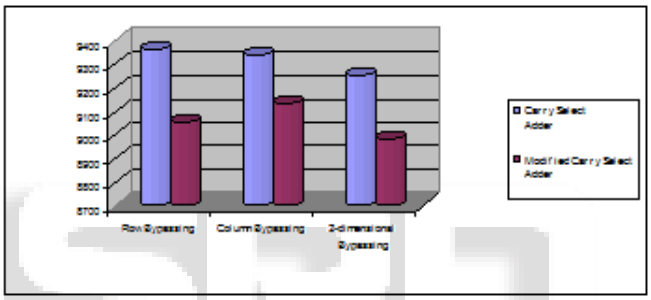
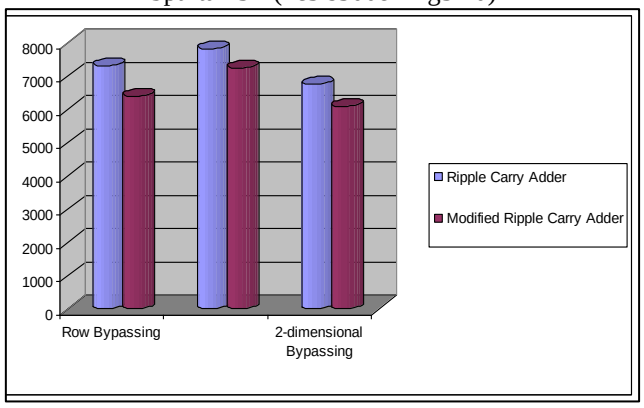


Fig.: The dynamic power of 8x8 row bypassing, column bypassing and row and column bypassing, for different adders

Area (N and gate)	Ripple Carry Adder	Modified Ripple Carry Adder	Carry Look Ahead Adder	Modified Carry Look Ahead Adder	Carry Select Adder	Modified Carry Select Adder
Row Bypassing	7325.69	6405.84	6532.32	6246.347	9366.95	9056.834
Column Bypassing	7825.66	7252.66	8626.65	8196.35	9342.60	9140.04
2-dimensional Bypassing	6753.22	6089.809	7252.23	6880.81	9256.65	8986.925

Table 3 Comparison of Area of different 8x8 multipliers on Spartan 3E (xc3 s500e-4 fg3 20)



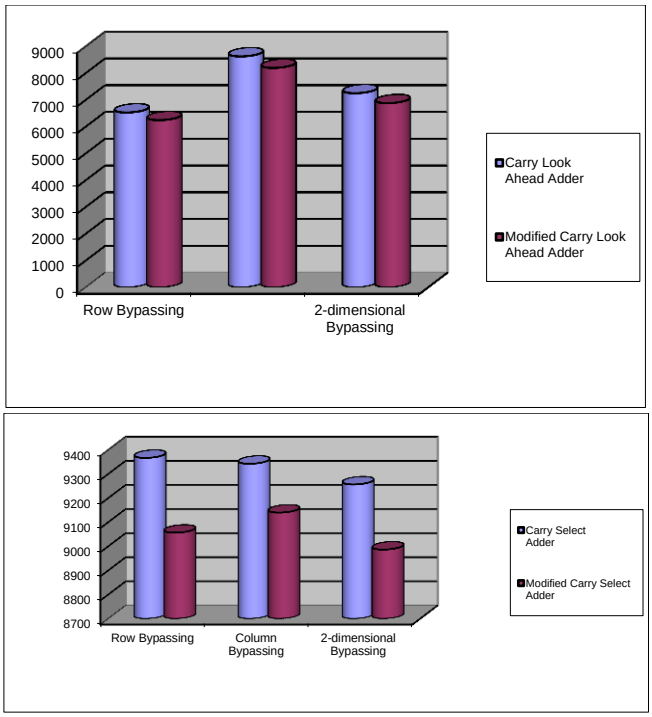


Fig.: The area of Ripple Carry, Carry look ahead, and Carry Select adder for 8x8 row bypassing.

IV. Conclusion and Future Work

Power dissipation in multiplier designs has been much researched in recent years, due to the importance of the multiplier circuit in a wide variety of microelectronic systems. The focus of multiplier design has traditionally been delay optimization, although this design goal has recently been supplemented by power consumption considerations. Our goal has been first to understand how power is dissipated in multipliers, and secondly to devise ways to reduce this power consumption. In this paper, we described previous work which has been done in the area of multiplier delay and power optimization. We identified methods by which multiplier delay has been reduced, and we concentrated on understanding how these various speedup techniques impact the power dissipation of the multiplier as a whole. Power savings of up to 25% were achieved, along with reductions in die area and interconnect. We have presented an investigation of multiplier power dissipation, along with some techniques which allow reductions in power consumption for this circuit. Given the importance of multipliers, it is likely that further research efforts will be directed at optimizing this block for delay and power efficiency.

After putting lot of hard efforts, we learnt importance of the multiplier and the usage of bypassing techniques in the multiplier.

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. There are three bypassing techniques:

- Row Bypassing
- Column Bypassing
- Row and Column Bypassing

The Row bypassing technique implements bypassing if the multiplicand bit makes the partial product zero. It is better in terms of power as compared to the Braun multiplier. The row bypassing technique results in usage of the extra circuitry, to avoid the problem of carry propagation. Thus, an improvement is Column bypassing multiplier. This technique uses less area and power, as the MFA of the column bypassing multiplier uses only one multiplexer, whereas the row bypassing technique uses two multiplexer. To further improve the efficiency of the multiplier, and reduce the switching activity, Row and Column bypassing technique is implemented. If the multiplicand or multiplier bit is zero, full adder is not implementing. Thus, this results in less power consumption with respect to one dimensional bypassing. On comparing, the results show highest speed for column bypassing when compared to Braun multiplier and bypassing multipliers. For area and dynamic power, two dimensional bypassing consumes least power and area as compared to row bypassing and column bypassing. Thus, we can conclude column bypassing gives least delay and two dimensional bypassing techniques has lowest dynamic power consumption. The use of different adders like carry look ahead adder (CLA), carry select adder apart from ripple carry adder (RCA) in the last stage helped to improve in terms of delay

References

[1] C.S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron. Comput., pp. 14-17,Feb. 1964.
[2] Z.-J. Mou, "'Overturned-Stairs' Adder Trees and Multiplier Design," F IEEE Computer Society, http://csdl.computer.org/comp/trans/tcI1992/08/t0940abs.htm
[3] A. Weinberger and J. L. Smith, "A logic for high-speed addition," National Bureau of Standards Circular591, pp. 3-12,1958.
[4] T. Lynch and E. E. Swartzlander, Jr., "The redundant cell adder," in Proc. 10th Symp. Comput. Arithmetic, 1991, pp. 165-170.
[5] Kantabutra, "Designing One-Level Carry-Skip Adders," Ieee Trans. Comput., Vol. 42, No. 6, Pp. 759-764, June 1993.
[6] A. Weinberger and J. L. Smith, "A logic for high-speed addition," National Bureau of Standards Circular591, pp. 3-12,1958.
[7] Y. Harata et al., "A high-speed multiplier using a redundant binary adder tree," IEEE J. Solid-State Circuits, vol. SC-22, pp. 28-34, Feb. 1987.
[8] http://www.mathworks.comlaccess/helpdesklhelp/techdoc/matlab ~rog/
[9] E. Herniter, "Programming in MATLAB," Northern Arizona University, 2001.
[10] R. Lipsatt, "VHDL: Hardware Description and design," Intermetrics Inc., 1993.
[11] Peter Kornerup, Southern Danish University IEEE Computer Society http://csdl.computer.orglcomp/proceedings/asap/200211712/00117120218abs.htm
[12] F. Ancona, S. Rovetta, R. Zumino , "High performance in tree-based parallel architectures," Genoa Univ IEEE Computer Society, HUNGARYp.474.

[13] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," IRE Transactions on Electronic Computers, vol. EC-10, pp. 389-400, Sep 1961.
[14] 1.Koren, Computer Arithmetic Algorithms, Englewood Cliffs, N.J.: Prentice Hall, 1993.
[15] Kiamal Z. Pekmestzi, "Multiplexer-Based Array Multipliers", IEEE Transactions on Computers, Vol. 48, No.1, pp.15-23, Jan. 1999.
[16] N.Ravi, Dr.T.S.Rao, Dr.T.J.Prasad, "Performance Evaluation of Bypassing Array Multiplier with Optimized Design", International Journal of Computer Applications(0975 – 8887), Vol. 28, No.5, pp.1-5, Aug. 2011.
[17] Jalil Fadavi-Ardekani, "Ad x N Booth Encoded Multiplier Generator Using Optimizedn Wallace Trees", lEEE Transactions on Very- Large Scale Integration (Msl) Systems,
[18] A. R. Fridi, "Partial multiplication: A low-power approach for parallel multiplier", ECE729 Course Project, Dept. Electrical and Computer Eng., Univ. Waterloo, Apr.1994. Vol. 1, No. 2, pp. 120-125, Jun. 1993.
[19] Sumit Vaidya, Deepak Dandekar, "Delay-Power performance comparison of multipliers in VLSI circuit design", International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, pp. 47-56, Jul. 2010.
[20] Harpreet Singh Dhillon, abhijit Mitra, "A Reduced –Bit Multiplication Algorithm for Digital Arithmetic", International journal of Computational and Mathematical Sciences, spring, pp. 64-69, 2008.

