ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 11, November 2015

Design of Low Power Approximate Compressor For Dadda Multiplier

S.Sudha[1], M.Revathy[2]

[1]PG Scholar, M.E VLSI Design, Department of ECE, PSNA College Of Engineering and Technology, Dindigul

[2]Assistant Professor, Department of ECE, PSNA College Of Engineering and Technology, Dindigul

***Abstract*- Multipliers have large area, long latency and consume considerable power and the design of good multipliers is always a challenge for VLSI system designers. Approximate computing is one of the attractive parameter for digital processor and also for computer arithmetic operations and designs. In this paper analysis and design of two new approximate 4-2 compressors are discussed and they addressing the three main important parameters such as area, delay and power consumption. The proposed approximate compressors are proposed and analyzed for dadda multiplier. The simulation results are implemented for the application of 8x8 binary multiplication by using Xilinx ISE design suite 13.2.**

Index terms - VLSI, Approximate compressor, 4-2 compressor, dadda multiplier.

I. Introduction

In recent years, the continues integration of more and more components on minimum silicon chip is very challenging to the scientists to produce the new trends and techniques. According to the Moore's law area, speed and power are the three major areas these must be the challenging factors in the designing of VLSI circuits. Multipliers are the primary block and multiplication is important for many operations such as image processing, DSPs, digital filters etc.., these multipliers consumes more operating time and power. Multiplication consists of three stages as follows, 1. Partial product generation, 2.Partial product reductions, 3.Final carry propagating addition. Using compressor in partial products reduction step is completed. Compressors are basic circuits which counts the number of ones in the given input. Compressors being one of the PEs (processing element) are the fundamental building block for accumulation of partial product in multiplication. Compressors dictate the overall critical path of the circuit owing which high speed and reduced power demand is progressively increasing.

II. Dadda Multiplier

In a popular multiplication scheme the array, the summation proceeds in a more regular, but slower manner, to obtaining the summation of the partial products. The Dadda multiplier is a hardware multiplier design, invented by computer scientist Luigi Dadda in 1965. It is slightly faster (for all operand sizes) and requires fewer gates (for all but the smallest operand sizes) than array multiplier. Dadda multipliers have the same 3 steps:

• Multiply each bit of one argument with the each and every bit of other argument and continue until all arguments are multiplied.
• Reduce the number of partial products to two layers of full and half adders.
• Group the wires in two numbers, and add them with a conventional adder.

Dadda multipliers perform few reductions only when compared to Wallace multiplier. Because of this, Dadda multipliers have less expensive reduction phase, but the numbers may be a few bits longer, thus requiring slightly bigger adders. Compared to a Wallace tree, which requires ten full adders and half adders, the reduction phase of the Dadda multiplier has the same delay, but requires only six. On the other hand, the final adder has 6-bit inputs (weights 2 to 64), rather than 5-bit (weights 8 to 128) as in a Wallace tree.

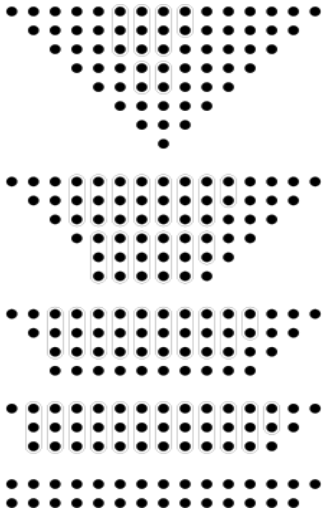

Figure.2.1 Dot diagram of 8x8 dadda multiplier

III. Exact 4-2 Compressor Design

The main goal of either multi-operand carry-save addition or parallel multiplication is to reduce n numbers to two numbers; therefore, n-2 compressors (or n-2 counters) have been widely used in computer arithmetic. An-2 compressor is usually a slice of a circuit that reduces n numbers to two numbers when properly replicated.

A widely used structure for compression is the 4-2 compressor. The 4-2 Compressor has 5 inputs A, B, C, D and Cin to generate 3 outputs Sum, Carry and Cout. The 4 inputs A, B, C and D and the output Sum have the same weight. The input Cin is output from a previous lower significant compressor and the Cout output is for the compressor in the next significant stage.

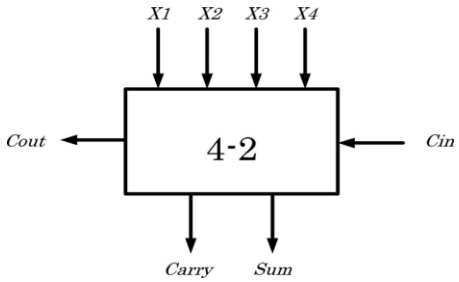

Fig. 3.1 4-2 compressor.

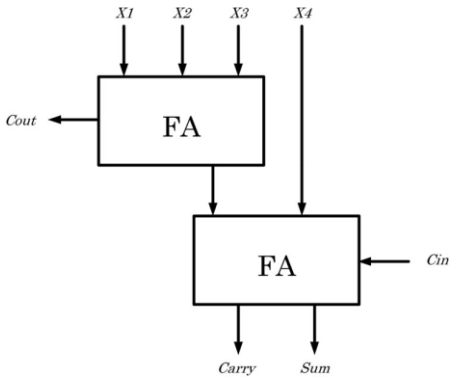

Fig.3.2 Implementation of 4-2 compressor.

The following equations give the outputs of the 4-2 compressor,

$$Sum = x1 \oplus x2 \oplus x3 \oplus x4 \oplus Cin$$

$$Cout = (x1 \oplus x2)x3 + \overline{(x1 \oplus x2)}x1$$

$$Carry = (x1 \oplus x2 \oplus x3 \oplus x4)Cin + \overline{(x1 \oplus x2 \oplus x3 \oplus x4)}x4.$$

The common implementation of a 4-2 compressor is accomplished by utilizing two full-adder (FA) cells.

Optimized 4-2 compressor

The optimized design of an exact4-2 compressor based on the so-called XOR-XNOR gates a XOR-XNOR gate simultaneously generates the XOR and XNOR output signals. The design consists of three XOR-XNOR (denoted by XOR*) gates, one XOR and two 2-1 MUXes. The critical path of this design has a delay of 3Δ, where Δ is the unitary delay through any gate in the design.

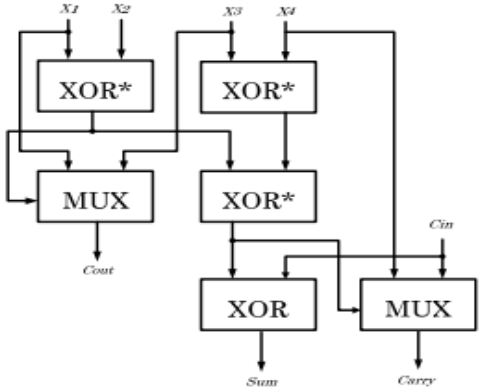

Fig.3.3 Optimized 4-2 compressor

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 11, November 2015

IV. Approximate Compressor Design

Two designs of an approximate compressor are proposed. Intuitively to design an approximate 4-2 compressor, it is possible to substitute the exact full-adder cells by an approximate full-adder cell.

Design1

In the design1 approximation, we approximate the result by making Carry'=Cin with this approximation the carry output in an exact compressor has the same value of input Cin. In particular, the simplification of sum to a value of 0 reduces the difference between the approximate and the exact outputs as well as the complexity of its design. Also, the presence of some errors in the sum signal will results in a reductions of the delay of producing the approximate sum and the overall delay of the design.

$$Sum' = C_{in}'(\overline{x1 \oplus x2} + \overline{x3 \oplus x4})$$

Although the above mentioned simplifications of carry and sum increase the error rate in the proposed approximate compressor, its design complexity and therefore the power consumption are considerably decreased.

$$C_{out}' = \overline{(\overline{x1x2} + \overline{x3x4})}$$

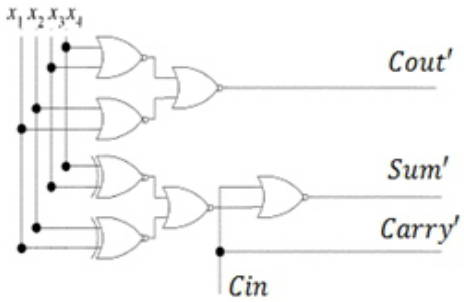

Fig 4.1 Gate Level Design of design1

The gate level structure of the first proposed design shows that the critical path of this compressor has still a delay of 3D, so it is the same as for the exact compressor. However, the propagation delay through the gates of this design is lower than the one for the exact compressor.

Design2:

A second design of an approximate compressor is proposed to further increase performance as well as reducing the error rate. Since the carry and cout outputs have the same weight, the proposed equations for the approximate carry and cout in the previous part can be interchanged.

$$Sum' = (\overline{x1 \oplus x2} + \overline{x3 \oplus x4})$$

$$C_{out}' = \overline{(\overline{x1x2} + \overline{x3x4})}$$

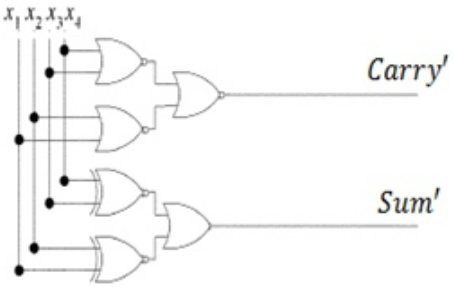

Fig.4 2 Gate Level Design of design2

The delay of the critical path of this approximate design is 2D, so it is 1D less than the previous designs; moreover, a further reduction in the number of gates is accomplished.

Dadda Multiplier Using Design1:

An 8×8 unsigned Dadda tree multiplier is considered to access the imp-act of using the proposed compress-ors in approximate multipliers.

- The proposed multiplier uses in the first part, the AND gates to generate all partial products.
- The reduction part uses half-adders, full-adders and 4-2 compressors; each partial product bit is represented by a dot. In the first stage, 2 half-adders, 2 full-adders and 8 compressors are utilized to reduce the partial products into at- most four rows.

- In the second or final stage, 1 half-adder, 1 full-adder and 10 compress-ors are used to compute the two final rows of partial products.
- Therefore, two stages of reduction and 3 half-adders, 3 full-adders and 18 compressors are needed in the reduction circuitry of an 8×8 Dadda multiplier.

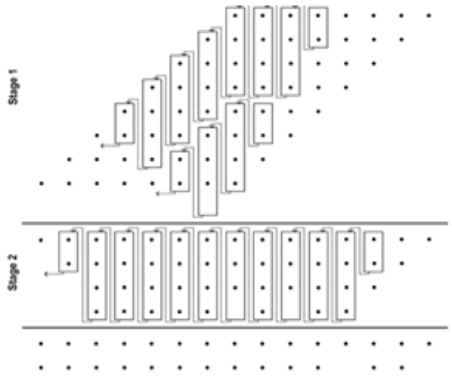

Fig 4.3 Dadda Multiplier using Design1

Dadda Multiplier Using Design 2:

- In the first case (Multiplier1), Desi-gn1 is used for all 4-2 compressors.
- In the second case (Multiplier2), Design2 is used for the 4-2 compressors. Since Design2 does not have Cin and Cout, the reduced circuitry of this multiplier requires a lower number of compressors. Multiplier2 uses 6 half-adders, 1 full-adder and 17 compressors.
- In the third case, Design1 is used for the compressors in then 1-least significant columns. The other n most significant columns in the reduction circuitry use exact 4-2 compressors.

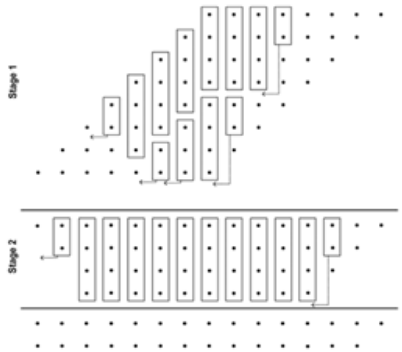

Fig.4.4 Dadda Multiplier using Design2

The objectives of the first two approximate designs are to reduce the delay and power consumption compared with an exact multiplier.

V. Simulation and Results

Simulation results are carried out by Xilinx ISE design 13.2. Results are taken individually for all the tasks in the process of binary multiplication.

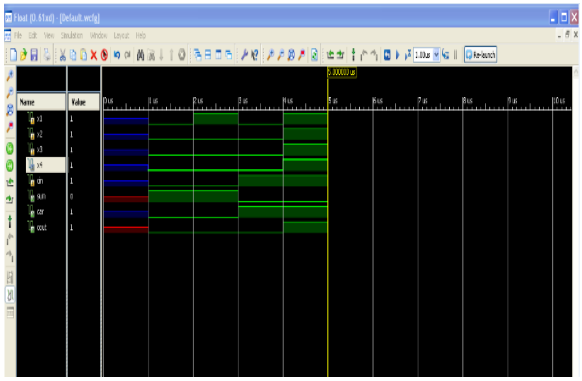

Fig.5.1Waveform of design 1

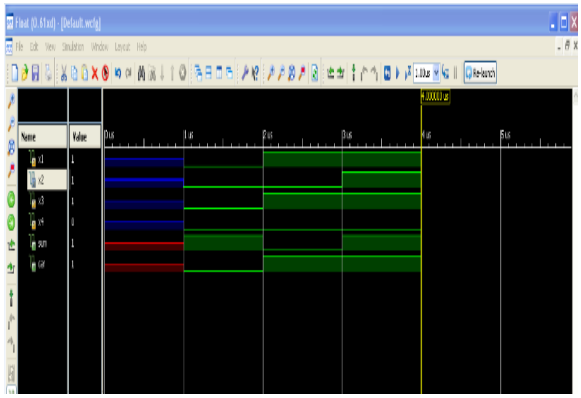

Fig.5.2 Waveform of design 1

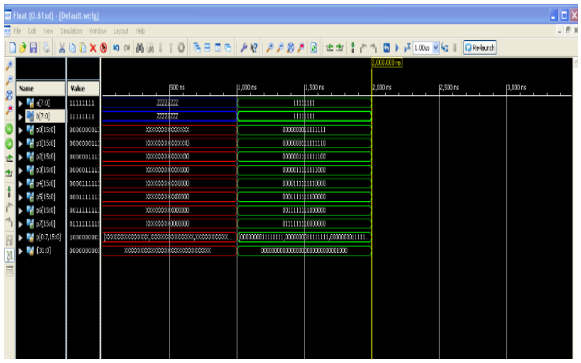

Fig.5.3 Wave form of partial product generation

Delay:

The delay of the reduction circuitry of a Dadda multiplier is dependent on the number of reduction stages and the delay of each stage. In Multipliers design1 the delay is 3Dand design2 the delay is considerably reduced to 2D and 1D.

Transistor Count:

The transistor count is used in this paper as metric of circuit complexity. In this paper transistor count is measured in terms of 4 input LUTs.

Power consumption:

The power consumption of each multiplier is determined by the number and type of compressors used.

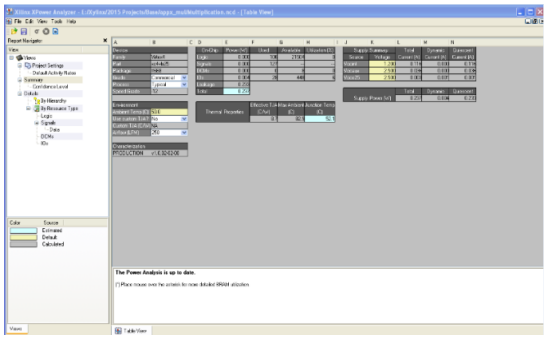

Fig.5.4 Simulation of design1 power result

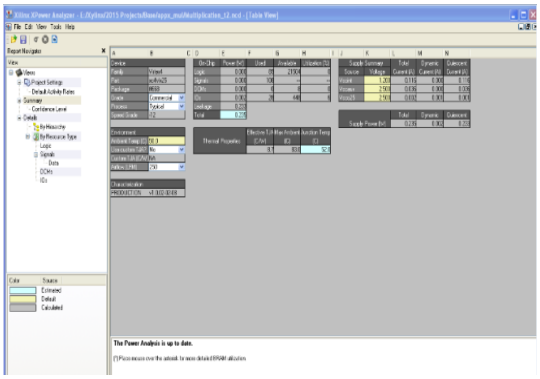

Fig.5.5 Simulation of design1 power result

	Power(w)	Delay	No.of LUTs(area)
Multiplier 1	0.237	2D	108
Multiplier 2	0.235	1D	89

Comparison of multiplier 1and 2

Parameters	Design 1	Design 2
Number of 4 input LUTS	108	89
Number of occupied slices	57	49
Number of bonded IOBs	28	28
Average fanouts of nonclock nets	3.07	3.13

Comparison of design1 and design2

VI. Conclusion

In this paper, four 8x8 bit approximate multipliers are designed using 4-2 compressors. Simulation results have been implemented reported. These approximate compressors are designed using design1 and design2. And finally concluded that the design 2 multiplier is suitable for implement the multiplication with low power.

References

[1] A. Momeni, J. Han, P. Montuschi, F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Transactions on Computers, in press, 2014.

[2] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bioinspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," IEEE Trans. Circuits Syst. I: Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[3] C. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low- power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," IEEE Trans. Circuits Syst., vol. 51, no. 10, pp. 1985–1997, Oct. 2004.

[4] D. Radhakrishnan and A. P. Preethy, "Low-Power CMOS pass logic 4-2 compressor for high-speed multiplication," in Proc. IEEE 43rd Midwest Symp. Circuits Syst., 2000, vol. 3, pp. 1296–1298.

[5] Z. Wang, G. A. Jullien, and W. C. Miller, "A new design technique for column compression multipliers," IEEE Trans. Comput., vol. 44, no. 8, pp. 962–970, Aug. 1995.

[6] J. Gu and C. H. Chang, "Ultra low-voltage, low-power 4-2 compressor for high speed multiplications," in Proc. 36th IEEE Int. Symp. Circuits Syst., Bangkok, Thailand, May 2003, pp. v-321–v-324.

[7] M. Margala and N. G. Durdle, "Low-power low-voltage 4-2 compressors for VLSI Applications," in Proc. IEEE Alessandro Volta Memorial Workshop Low-Power Design, 1999, pp. 84–90.

[8] D. Baran, M. Aktan, and V. G. Oklobdzija, "Energy efficient implementation of parallel CMOS multipliers with improved compressors," in Proc. ACM/IEEE 16th Int. Symp. Low Power Electron. Design, 2010, pp. 147–152.

[9] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and Probabilistic Adders," IEEE Trans. Computers, vol. 63, no. 9, pp. 1760–1771, Sep. 2013.

[10] J. Han and M. Orshansky, "Approximate Computing: An Emerging Paradigm for Energy-Efficient Design," in ETS'13, Avignon, France, May 27-31, 2013, pp.1-6.

[11] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in ICCAD 2011, pp. 667 – 673.

[12] C.-H. Lin, I.-C. Lin, "High accuracy approximate multiplier with error correction," IEEE 31st International Conference on Computer Design (ICCD), 2013.

[13] K. Bhardwaj, P.S. Mane, J. Henkel, "Power- and area-efficient Approximate Wallace Tree Multiplier for error-resilient systems," 15th International Symposium on Quality Electronic Design (ISQED), 2014.

[14] C. Liu, J. Han and F. Lombardi, "A Low-Power, High-Performance Approximate Multiplier with Configurable Partial Error Recovery," DATE 2014, Dresten, Germany, 2014.

[15] M.S.K Lau, K.V. Ling, Y.C. Chu. "Energy-aware probabilistic multiplier: design and analysis." In Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems, Grenoble, France, pp. 281-290, 2009.