ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design of a Approximate Compressor For DADDA Multiplier

B. Jhansi Reddy
M.Tech Student,
TKR College of Engineering and Technology.

Mrs. J.Sunitha Kumari
Associate Professor,
Department Of ECE,
TKR College of Engineering and Technology.

Abstract:

Multiplication is a fundamental operation in most of the signal processing algorithms. Multipliers have large area, long latency and consume considerable power and the design of good multipliers is always a challenge for VLSI system designers. For this inconvenience compressor for low latency, low power consumption and reduced stages of product is designed. In this paper approximate compressor design for reduction of multiplier stages in the Dadda multiplier is proposed. These results are carried out using Tanner EDA tool.

Keywords:

VLSI, Compressors,Tanner EDA.

I. Introduction:

Many scientific and engineering problems are computed using accurate, precise and deterministic algorithms. However, in many applications involving signal/image processing and multimedia, exact and accurate computations are not always necessary, because these applications are error tolerant and produce results that are good enough for human perception. In these error resilient applications, a reduction in circuit complexity, and thus, area, power and delay is very important for the operation of a circuit. Hence, approximate computing can be used in error tolerant applications by reducing accuracy, but still providing meaningful results faster and/or with lower power consumption. Multipliers form an important hardware block in the DSP and embedded applications.Multiplication speed determines processor speed. So high speed multipliers are needed in the processors for many applications. For increasing the speed of multiplication different algorithms are used Multiplication is a most commonly used operation in many computing systems.

A number (multiplicand) is added to itself a number of times as specified by another number (multiplier) to form a result (product). But the implementation of multiplier takes huge hardware resources and the circuit operates at low speed. Multiplication is one of the fundamental components in DSP and Embedded system. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the real time system. Multiplier requires more hardware resources than the adder and subtractors. Improving the performance and reducing the power dissipation of the systems are the most important design challenges for Embedded and DSP applications. Increasing the word length results in hardware complexity and also increases the multiplication time. Many algorithm have been developed in order to realize high speed multipliers such as Booths algorithm, Wallace tree algorithm etc. Multipliers based on Booth algorithm and Wallace tree addition is one of the fast and low power multiplier. Multiplication consists of three major steps: 1) recoding and generating partial products 2) reducing the partial products by partial product reduction schemes to two rows and 3) adding the remaining two rows of partial products by using a carry-propagate adder (e.g. Carry look ahead adder) to obtain the final product.

II. Overview of Multiplier:

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has an important part in low-power VLSI system design. A system is generally determined by the performance of the multiplier because the multiplier is generally the slowest element and more area consuming in the system. Hence optimizing the speed and area of the multiplier is one of the major design issues.

However, area and speed are usually conflicting constraints so that improvements in speed results in larger areas. Multiplication is a mathematical operation that include process of adding an integer to itself a specified number of times. A number (multiplicand) is added itself a number of times as specified by another number (multiplier) to form a result(product). Multipliers play an important role in today's digital signal processing and various other applications. Multiplier design should offer high speed, low power consumption. Multiplication involves mainly 3 steps

1) Partial product generation
2) Partial product reduction
3) Final addition

Dadda Multiplier:

The Dadda multiplier was designed by the scientist Luigi Dadda in 1965. It looks similar to Wallace multiplier but slightly faster and require less gates.

Dadda Multiplier was defined in three 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.

In this paper, a 8*8 multiplier using dada multiplier design is designed. Instead of using conventional full adders and half adder for designing the multiplier, compressors which reduces the complexity of the multiplier is introduced.

4:2 Compressor design:

The 4-2 Compressor has 5 inputs A, B, C, D and Cin to generate 3 outputs Sum, Carry and Cout as shown in figure 1(a). 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. The conventional approach to implement 4-2 compressors is with 2 full adders connected serially as shown in figure 1(b).

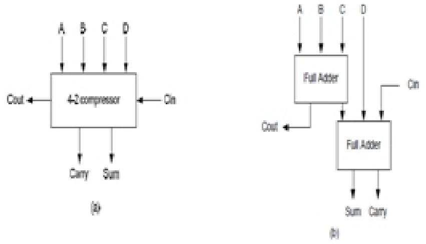

Fig:1(a) 4-2 adder compressor. (b) 4-2 adder compressor implemented with full adders.

$$A+B+C+D+Cin=Sum+2(Carry+C_{out})$$

III. Approximate Compressor Design:

The exact compressor was reduced by proposing two approximate compressors. The two approximate compressors are shown below

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.

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

The change of the value of Cout in some states, may reduce the error distance provided by approximate carry and sum and also more simplification in the proposed design.

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

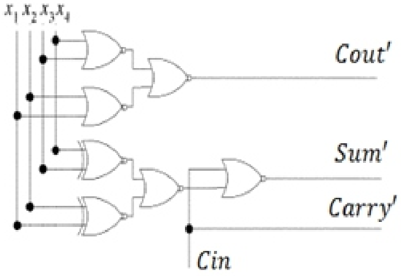

Fig 2: Gate Level Design of design1

Design2:

A design2 was designed for more approximation than the design1 with further increased performance by reducing the error rate. In the proposed design, the carry' and Cin are approximated since they are having the same weight. In this design we take Cin as 0 so that we can remove the carry'.

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

$$\text{C}_{\text{out}}' = \overline{(\overline{x1x2} + \overline{x3x4})}$$

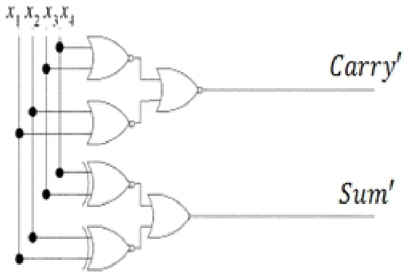

Fig 3: Gate Level Design of Design2

Dadda Multiplier using Design1:

•A 8×8 unsigned Dadda tree multip-lier is considered to access the imp-act of using the proposed compressors 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.

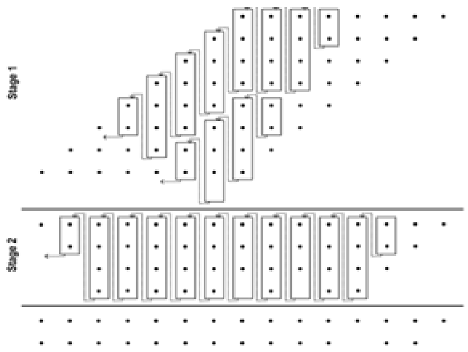

Fig 4: Dadda Multiplier using Design1

Dadda Multiplier using Design2:

•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 (Multiplier 3), 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.

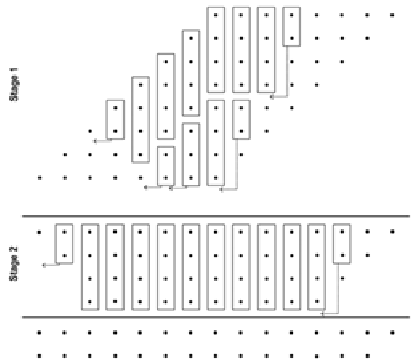

Fig 5: Dadda Multiplier using Design2

IV. Simulation and Results:

These circuits are designed and simulated using Tanner EDA tool.

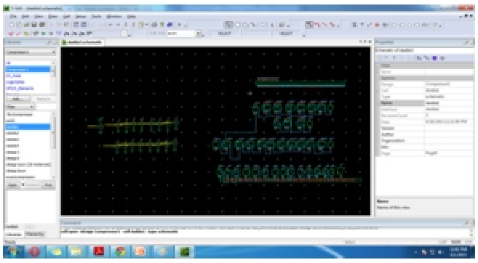

Fig 6: S edit design of Dadda1 Multiplier

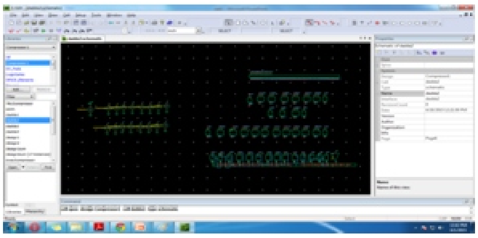

Fig 7: S edit design of Dadda2 Multiplier

The simulation results are carried out using T-spice of Tanner EDA and the power consumed by the multipliers are shown in the table below.

Circuit	Power Consumption	Delay
Dadda1 Multiplier	1.399W	0.2 ns
Dadda2 Multiplier	1.348W	0.1ns

V. Conclusion:

In this paper, four 8x8 bit approximate multipliers are designed using 4-2 compressors. Simulation results have been reported for this design. These approximate compressors are designed using design1 and design2.

References:

[1] 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.

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

[3] J. Liang, J. Han, F. Lombardi, “New metrics for the reliability of approximate and probabilistic adders,” IEEE Trans. on Computers, vol. 63, no. 9, pp. 1760 - 1771, 2013.

[4] K.Y. Kyaw, W.L. Goh, K.S. Yeo, “Low-power high-speed multiplier for error-tolerant application,” IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC), 2010.

[5] P. Kulkarni, P. Gupta, M. Ercegovac, “Trading accuracy for power with an Underdesigned Multiplier architecture,” 24th International Conference on VLSI Design, 2011.

[6] H.R. Mahdiani, A. Ahmadi, S.M. Fakhraie, C. Lucas, “Bio-Inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” IEEE Transactions on Circuits and Systems, vol. 57 no. 4, 2010.

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

[8] 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.

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

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

[11] C.H. Chang, J. Gu, M. Zhang, “Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits,” IEEE Transactions on Circuits and Systems, vol.51, no.10, pp.1985-1997, 2004.

[12] 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