International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 27th Oct 2017. Link :http://www.ijiemr.org/downloads.php?vol=Volume-6&issue=ISSUE-9

Title: ADAPTIVE MODEL FOR APPROXIMATE MULTIPLICATION WITH COMPRESSORS

Volume 06, Issue 09, Pages: 297 – 303.

Paper Authors

AKULA MOUNIKA, HARITHA
VAAGDEVI COLLEGE OF ENGINEERING.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

ADAPTIVE MODEL FOR APPROXIMATE MULTIPLICATION WITH COMPRESSORS

[1]AKULA MOUNIKA, [2]HARITHA
M.Tech Scahloar,Dept of E.C.E, VAAGDEVI COLLEGE OF ENGINEERING
Assistant Professor, Dept of E.C.E, VAAGDEVI COLLEGE OF ENGINEERING

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 we are designing compressors for low latency, low power consumption and reduced stages of products. In paper we propose approximate compressor design for reduction of multiplier stages in the dada multiplier. 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 [1]. 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 increase 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 substractors. For 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

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

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) re-coding 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

- Partial product generation
- Partial product reduction
- Final addition

A. Dadda Multiplier The Dadda multiplier was designed by the scientist Luigi Dadda in 1965. Its looks similar to Wallace multiplier but slightly faster and required less gates. Dadda Multiplier was defined in three steps

- Multiply the each bit of on 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 we have designed a 8*8 multiplier using dada multiplier design. Instead of using conventional full adders and half adder for designing the multiplier we introduce compressors which will reduces the complexity of the multiplier.

B. 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 . The 4 inputs A, B, C and D and the output Sum have the same weight. The input Cin is the 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

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

$$Sum = A \oplus B \oplus C \oplus D \oplus Cin \quad (1)$$

$$Cout = (A \oplus B) * C + \overline{(A \oplus B)} * A \quad (2)$$

$$Carry = (A \oplus B \oplus C \oplus D) * Cin + \overline{(A \oplus B \oplus C \oplus D)} * D \quad (3)$$

TABLE I. Truth Table of 4-2 Compressor

c_{in}	X_4	X_3	X_2	X_1	c_{out}	carry	sum
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	1	1	0	0
0	0	1	0	0	0	0	1
0	0	1	0	1	1	0	0
0	0	1	1	0	1	0	0
0	0	1	1	1	1	0	1
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	0	1	1	1	0	1
0	1	1	0	0	0	1	0
0	1	1	0	1	1	0	1
0	1	1	1	0	1	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	0	1	0	0	1	0
1	0	0	1	1	1	0	1
1	0	1	0	0	0	1	0
1	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	0	0	1	0
1	1	0	0	1	0	1	1
1	1	0	1	0	0	1	1
1	1	0	1	1	1	1	0
1	1	1	0	0	0	1	1
1	1	1	0	1	1	1	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1

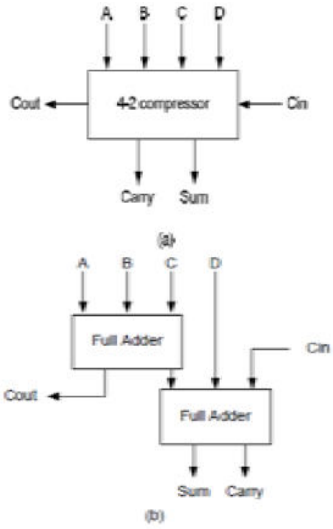

Fig.1. (a) 4-2 Adder Compressor. (b) 4-2 Adder Compressor Implemented with Full Adders.

A+B+C+D+CIN= SUM+2(CARRY+COUT).

III. APPROXIMATE COMPRESSOR DESIGN

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

A. Design1 In the design 1 approximation the we approximate the result by making Caary'=Cin With this approximation the *carry* output in an exact compressor has the same value of the input *cin* in 24 out of 32 states.

Carry'=Cin (4)

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

$$Sum' = \overline{Cin}(\overline{x1 \oplus x2} + \overline{x3 \oplus x4}) \quad (5)$$

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.

$$Cout' = \overline{(x1x2 + x3x4)} \quad (6)$$

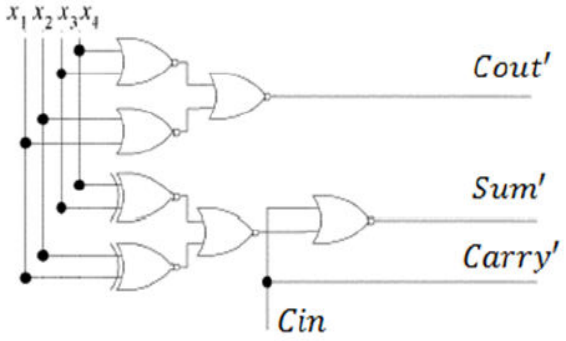

Fig.2. Gate Level Design Approximate Compressor Desin1

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

TABLE II. Truth Table of Design 1

c_{in}	X_4	X_3	X_2	X_1	c_{out}'	carry'	sum'	Difference
0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	1	0
0	0	0	1	1	0	0	1	-1
0	0	1	0	0	0	0	1	0
0	0	1	0	1	1	0	0	0
0	0	1	1	0	1	0	0	0
0	0	1	1	1	1	0	1	0
0	1	0	0	0	0	0	1	0
0	1	0	0	1	1	0	0	0
0	1	0	1	0	1	0	0	0
0	1	0	1	1	1	0	1	0
0	1	1	0	0	0	0	1	-1
0	1	1	0	1	1	0	1	0
0	1	1	1	0	1	0	1	0
0	1	1	1	1	1	0	1	-1
1	0	0	0	0	0	1	0	1
1	0	0	0	1	0	1	0	0
1	0	0	1	0	0	1	0	0
1	0	0	1	1	0	1	0	-1
1	0	1	0	0	0	1	0	0
1	0	1	0	1	1	1	0	1
1	0	1	1	0	1	1	0	1
1	0	1	1	1	1	1	0	0
1	1	0	0	0	0	1	0	0
1	1	0	0	1	1	1	0	1
1	1	0	1	0	1	1	0	1
1	1	0	1	1	1	1	0	0
1	1	1	0	0	0	1	0	-1
1	1	1	0	1	1	1	0	0
1	1	1	1	0	1	1	0	0
1	1	1	1	1	1	1	0	-1

B. Design2 A desgin2 was designed for more approximation than the design1 further increase performance by reducing the error rate. In the proposed design we approximate the carry' and cin since they are having the same weight. In this design we take Cin as o so that we can remove the caary'.

$$Sum' = \overline{(x1 \oplus x2} + \overline{x3 \oplus x4)} \quad (7)$$

$$Cout' = \overline{(x1x2} + \overline{x3x4)} \quad (8)$$

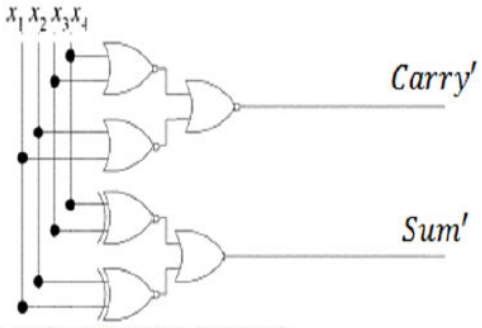

Fig.3. Gate Level Design of Design2.

C. Multiplication In this section, the impact of using the proposed compressors for multiplication is investigated. A fast (exact) multiplier is usually composed of three parts (or *modules*) [8].

- Partial product generation.
- A Carry Save Adder (CSA) tree to reduce the partial products' matrix to an addition of only two operands
- A Carry Propagation Adder(CPA) for the final computation of the binary result

TABLE III. Truth Table of Design 2

X_4	X_3	X_2	X_1	carry'	sum'	difference
0	0	0	0	0	1	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	-1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	1	0
1	0	0	0	0	1	0
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	0	1	-1
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	1	1	-1

In the design of a multiplier, the second module plays a pivotal role in terms of delay, power consumption and circuit complexity. Compressors have been widely used [9, 10] to speed up the CSA tree and decrease its power dissipation, so to achieve fast and low-power operation. The use of approximate compressors in the CSA tree of a multiplier results in an approximate multiplier. A *8×8* unsigned Dadda tree multiplier is considered to assess the impact of using the proposed compressors in approximate multipliers. The proposed multiplier uses in the first part AND gates to generate all partial products. In the second part, the approximate compressors proposed in the previous section are utilized in the CSA tree to reduce the partial products. The last part is an exact CPA to compute the final binary result. Figure 9(a) shows the reduction circuitry of an exact multiplier for

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

n=8. In this figure, 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 compressors 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. In this paper, four cases are considered for designing an approximate multiplier.

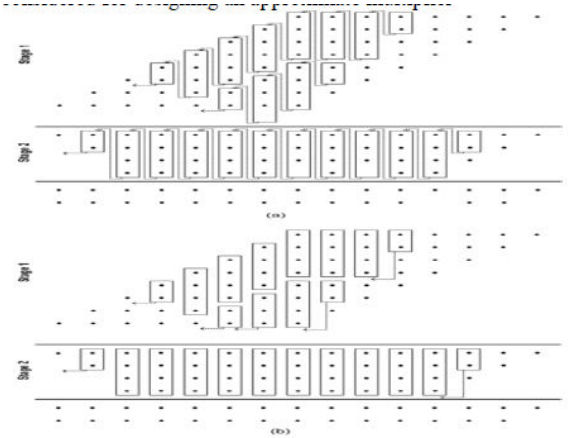

Fig.4. Reduction Circuitry of an *8*×*8*Dadda Multiplier, (a) Using Design1 Compressors,(b) Using Design2 Compressors.

In the first case (Multiplier 1), Design 1 is used for all 4-2 compressors in Figure 9(a).

- In the second case (Multiplier 2), Design 2 is used for the 4-2 compressors. Since Design 2 does not have *cin* and *cout*, the reduction circuitry of this multiplier requires a lower number of compressors (Figure 9(b)). Multiplier 2 uses 6 half-adders, 1 full-adder and 17 compressors.
- In the third case (Multiplier 3), Design 1 is used for the compressors in the*n-1* least significant columns. The other *n* most significant columns in the reduction circuitry use exact 4-2 compressors.
- In the fourth case (Multiplier 4), Design 2 and exact 4-2 compressors are used in the*n-1* least significant columns and the*n* most significant columns in the reduction circuitry respectively.

The objectives of the first two approximate designs are to reduce the delay and power consumption compared with an exact multiplier; however, a high error distance is expected. The next two approximate multipliers (i.e. Multipliers 3 and 4) are proposed to decrease the error distance. The delay in these designs is determined by the exact compressors that are in the critical path; therefore, there is no improvement in delay for these approximate designs compared with an exact multiplier. However, it is expected that the utilization of approximate compressors in the least significant columns will decrease the power consumption and transistor count (as measure of circuit complexity). While the first two proposed multipliers have better performance in terms of delay and power consumption, the error distances in the third and fourth designs are expected to be significantly lower.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

D. 4-2 Compressor

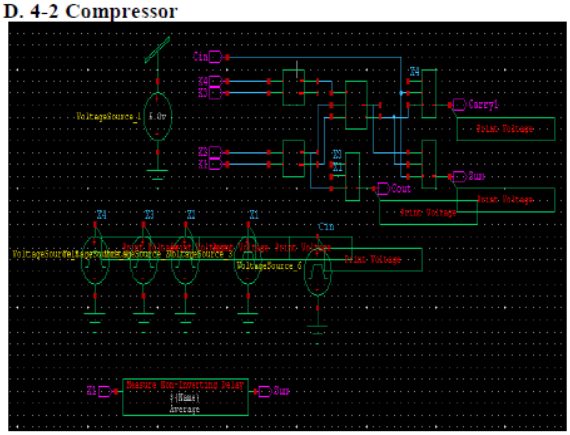

Fig.5. 4-2 Compressor

The 4-2 Compressor has 5 inputs A, B, C, D and Cin to generate 3 outputs Sum, Carry and Cout as shown The 4 inputs A, B, C and D and the output Sum have the same weight. The input Cin is the 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 Fig5.

IV. SIMULATION AND RESULTS

These circuits are designed and simulated using Tanner EDA tool the power consumption results are done using Tanner T-spice and the values are shown in below table.

TABLE IV. @180nm

Circuit	Power Consumption	Delay	PDP
Exact compressor[8]	2.257e-7	4.9459e-5	11.1628963
Design 1	4.016e-9	2.6162e-5	10.50
Design 2	6.4e-10	1.1893e-5	7.61152

TABLE V. @250nm

Circuit	Power consumption	Delay	PDP
Exact compressor[8]	2.5067e-7	8.996950e-5	22.552
Design 1	3.5412e-10	4.931098e-5	17.462
Design 2	3.18881e-10	3.898e-3	12.427

TABLE VI. Comparison of number of Transistor

Design	Number of transistors
Exact design	52
Design 1	28
Design 2	26

TABLE VII. Power Consumption Improvement in Reduction Theory

Design	Power
Multiplier 1	7.769892e2
Multiplier 2	6.991142e2
Multiplier 3	1.930730e2
Multiplier 4	1.874607e2

TABLE VIII. Delay Improvement in Reduction Theory

Design	Delay
Multiplier 1	3.9138e-13
Multiplier 2	3.4126e-13
Multiplier 3	3.1300e-13
Multiplier 4	3.1267e-13

TABLE IX. Transistor Count Improvement in Reduction Theory

Design	Improvement(%)
Multiplier 1	42.11
Multiplier 2	48.15
Multiplier 3	14.03
Multiplier 4	22.42

TABLE X. Approximate Multipliers and Their Features

Design	Feature
Multiplier 1	Design 1 in all coloumns
Multiplier 2	Design 2 in all coloumns
Multiplier 3	Design 1 in LSBs and exact compressor in MSBs
Multiplier 4	Design 2 in LSBs and exact compressor in MSBs

V. CONCLUSION

In this paper, we have designed four designs of 8x8 bit approximate multipliers have been proposed. Simulation results have been reported for design. This approximate compressors are designed using design1 and design was propose was shown **VI.**

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.