

Research Article

DOI: <http://dx.doi.org/10.22192/ijamr.2018.05.10.007>

A Design and implementation of compressor for utilizing in Dadda multiplier

Dr. S. Tamilarasan, Dr. A. V. Prathapkumar, V. Vijayakumar

Dhanalakshmi Srinivasan Engineering College - Perambalur

Keywords

compressor,
accuracy,
Delay,
power,
approximate
computing

Abstract

We proposed four 4:2 compressors, which flexibility of switching between the exact and approximate operating modes. In the approximate mode, these dual-quality compressors provide higher speeds and lower power Consumptions at the cost of lower accuracy. Each of these compressors has its own level of accuracy in The efficiencies of these compressors in a 32-bit Dadda multiplier are evaluated in a 45-nm standard CMOS technology by comparing their parameters with those of the state-of-the-art approximate multipliers. The results of comparison indicate, on average, 37.38% and 35% power consumption and lower delay in the approximate mode. Also, the effectiveness of these compressors is assessed in some image processing .

1. Introduction

Real-world signals can be mathematically represented as analog signals. Since the basic radio communication and transceivers were invented at the beginning of the 20th century, a set of problems analog communication systems has arisen. The primary disadvantage of Analog Signal Processing (ASP) is that any communication system has random unwanted variation. Armed with high reliability, accuracy, flexibility, and increased immunity-to-noise, Digital Signal Processing (DSP) has applications. compressors in the structures of parallel multipliers provides configurable accuracies (as well as their powers and speeds) may change dynamically during the runtime. The efficiencies of these compressors in a 32-bit Dadda multiplier are evaluated in a 45-nm standard CMOS technology by comparing their parameters with those of the state-of-the-art approximate multipliers. The results of comparison indicate, on average, and 68% lower delay and power consumption in the approximate mode. Also, the effectiveness of these compressors is assessed in some

image processing applications. one of the most attractive topics in semiconductor industry in the past 30 years. According to [1], the global market share of DSP processors and microcontrollers is more than 95% of the total volume of processors sold huge delay. Most of computer arithmetic applications are implemented using digital logic circuits, thus operating with a high degree of reliability and precision. However, many applications such as in multimedia and image processing can tolerate errors and imprecision in computation and still produce meaningful and useful results. Accurate and precise models and algorithms are not always suitable or efficient for use in these applications. The paradigm of inexact computation relies on relaxing fully precise and completely deterministic building modules when for example, designing energy-efficient systems. This allows imprecise computation to redirect the existing design process of digital circuits and systems by taking advantage of a decrease in complexity and cost with possibly a potential increase in performance and power efficiency.

II. Related work

Multipliers play an important role in today’s digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation. In the first large-scale digital systems, multiplication was performed as a series of additions and shifts. The requisite hardware consisted only of a parallel adder and a few registers. In the early 1950’s, multiplier performance was significantly improved with the introduction of Booth’s method, the modified Booth multiplier, and the development of faster adders and memory components. Booth’s method and the modified Booth method do not require a correction of the product when either (or both) of the operands is negative for two’s complement numbers. They have a delay that is generally proportional to the word length of the multiplier input. Due to the regularity of their structures, array multipliers are easy to layout and have been implemented frequently. The second class of parallel multiplier reduces a matrix of partial product bits to two words through the strategic application of counters or compressors. These two words are then summed using a fast carry-propagate adder to generate the product. This class of parallel multiplier is sometimes termed a *column compression multiplier*. Since the delay is proportional to the logarithm of the multiplier word length, these are also the fastest multipliers.

III. Proposed 4:2 compressor adder

A. Exact compressor

The main goal of either multi-operand carry-save addition or parallel multiplication gate simultaneously generates the XOR and XNOR output signals. Optimized design of an Exact4-2 compressor [8].The design of [8] consists of three XOR-XNOR (denoted by XOR*) gates, one XOR and two 2-1 MUXs. The critical path of this design has a delay of 3τ , where τ is the unitary delay through any gate in the design. The RTL View of

B) Proposed compressor:

is to reduce n numbers to two numbers; therefore, $n-2$ compressors (or $n-2$ counters) have been widely used in computer arithmetic. An $n-2$ compressor (Figure 1) is usually a slice of a circuit that reduces n numbers to two numbers when properly replicated. In slice i of the circuit, the $n-2$ compressor receives n bits in position i and one or more carry bits from the positions to the right, such as $i - 1$ or $i - 2$. It produces two output bits in positions i and $i + 1$ and one or more carry bits into the higher positions, such as $i + 1$ For the correct operation of the circuit shown in Figure 4.1, the following in equality must be satisfied,

$$n + \psi_1 + \psi_2 + \psi_3 + \dots \leq 3 + 2\psi_1 + 4\psi_2 + 8\psi_3$$

Where, ψ_i denotes the number of carry bits from slice i to slice $i+j$

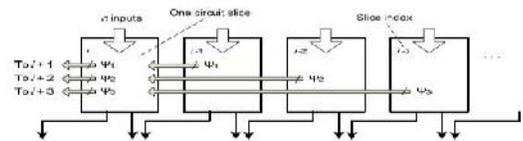


Figure 1.3 Schematic diagram of $n-2$ compressors

A multi operand addition circuit. A widely used structure for compression is the 4-2 compressor a 4:2compressor (Figure In this section, 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 in Figure3 by an approximate full-adder cell (such as the first design proposed in [2]). However, this is not very efficient, because it produces at least 17 incorrect results out of 32 possible outputs, i.e. the error rate of this inexact compressor is more than 53% erroneous outputs over the total number of outputs). Two different designs are proposed next to reduce the error rate; these designs offer significant performance improvement compared to exact Compressor with respect to delay, number of transistors and power consumption.

Design 1

The carry output in an exact compressor has the same value of the input *cin* in 24 out of 32 states. Therefore, an approximate design must consider this feature. In Design 1, the carry is simplified to *cin* by changing the value of the other 8 outputs.

$$Carry' = Cin \tag{5}$$

Since the Carry output has the higher weight of a binary bit, an erroneous value of this signal will produce a difference value of two in the output. For example, if the input pattern is “01001” (row 10 of Table II), the correct output is “010” that is equal to 2. By simplifying the carry output to *cin*, the approximate compressor will generate the “000” pattern at the output (i.e. a value of 0). This substantial difference may not be acceptable; however, it can be compensated or reduced by simplifying the *cout* and *sum* signals

Figure a). Approximate compressor Design 1

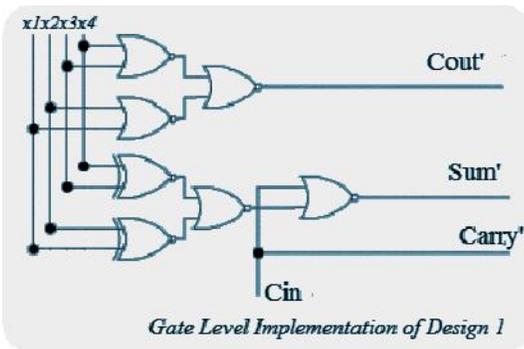


Table 2) truth table of the first Approximate 4-2 compressor

<i>c_{in}</i>	<i>x₁</i>	<i>x₂</i>	<i>x₃</i>	<i>x₄</i>	<i>c_{out}'</i>	<i>carry'</i>	<i>sum'</i>	<i>Difference</i>
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0
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	0	0
0	0	1	0	0	0	0	1	0
0	0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0
0	1	0	0	0	1	1	0	0
0	1	0	0	1	1	1	0	0
0	1	0	1	0	1	1	0	0
0	1	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0	0
0	1	1	0	1	1	1	0	0
0	1	1	1	0	1	1	0	0
0	1	1	1	1	1	1	0	0
1	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0
1	0	0	1	1	0	0	0	0
1	0	1	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0
1	0	1	1	0	0	0	0	0
1	0	1	1	1	0	0	0	0
1	1	0	0	0	1	1	0	0
1	1	0	0	1	1	1	0	0
1	1	0	1	0	1	1	0	0
1	1	0	1	1	1	1	0	0
1	1	1	0	0	1	1	0	0
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	0

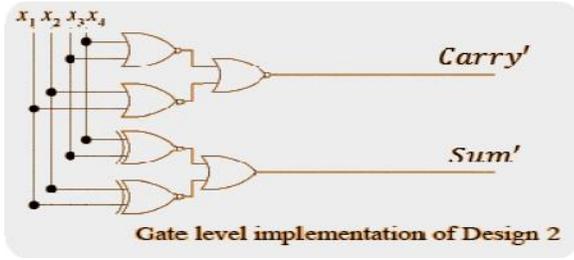
In particular, the simplification of *sum* to a value of 0 (second half of Table II) reduces the difference between the approximate and the exact outputs as well as the complexity of its design. Also, the presence of some errors in the *sum* signal will result in a reduction of the delay of producing the approximate *sum* and the overall delay of the design (because it is on the critical path).

$$Sum' = \overline{Cin}(x1 \oplus x2 + x3 \oplus x4) \tag{6}$$

In the last step, the change of the value of *c_{out}* in some states may reduce the error distance provided by approximate carry and *sum* and also more simplification in the proposed design.

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. This can be realized by comparing (2)-(4) and (5)-(7). Table II shows the truth table of the first proposed approximate compressor. It also shows the difference between the inexact output of the proposed approximate compressor and the output of the exact compressor. As shown in Table II, the proposed design has 12 incorrect outputs out of 32 outputs (thus yielding an error rate of 37.5%). This is less than the error rate using the best approximate full-adder cell of [2]. (5)-(7) are the logic expressions for the outputs of the first design of the approximate 4-2 compressor proposed in this manuscript. The gate level structure of the first proposed design (Figure 4.8) shows that the critical path of this compressor has still a delay of 3, so it is the same as for the exact compressor of Figure 5. However, the propagation delay through the gates of this design is lower than the one for the exact compressor. For example, the propagation delay in the XOR* gate that generates both the XOR and XNOR signals in [8], is higher than the delay through a XNOR gate of the proposed design. Therefore, the critical path delay in the proposed design is lower than in the exact design and moreover, the total number of gates in the proposed design is significantly less than that in the optimized exact compressor of [8].

The RTL View of the Proposed Approximate Compressor Design 1 is shown in Figure 4.9



Design 2

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. In this new design, carry uses the right hand side of (7) and cout is always equal to cin; since cin is zero in the first stage, cout and cin will be zero in all stages. So, cin and cout can be ignored in the hardware design. Figure 4.7 shows the block diagram of this approximate 4-2 compressor expressions shown below describe its outputs.

$$Sum' = \overline{(x_1 \oplus x_2 + x_3 \oplus x_4)}$$

$$Carry' = \overline{(x_1x_2 + x_3x_4)}$$

Figure a) Approximate 4-2 compressor, Design 2 Note that (9) is the same as (7) and (8) is the same as (6) for cin= 0. Figure 4.10 shows the gate level implementation of the second proposed design. The delay of the critical path of this approximate design is 2 , so it is 1 less than the previous designs; moreover, a further reduction in the number of gates is accomplished.

Table 3 Truth table of second proposed 4-2 compressor

X_4	X_3	X_2	X_1	$carry'$	s_1
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

Table 3 shows the truth table of the second approximate design for a 4-2 compressor; this Table also shows the difference between the exact decimal value of the addition of the inputs and the decimal value of the outputs produced by the approximate compressor. For example when all inputs are 1, the decimal value of the addition of the inputs is 4. However, the approximate compressor produces a 1 for the carry and sum. The decimal value of the outputs in this case is 3; Table II shows that the difference is -1. This design has therefore 4 incorrect outputs out of 16 outputs, so its error rate is now reduced to 25%. This is a very positive feature, because it shows that on a probabilistic basis, the imprecision of the proposed design is smaller than the other available schemes.

IV. Implementation of approximate multiplication

A multiplier can be divided into further three stages: - Partial products generation (PPG) stage is the first stage in which the multiplicand and the multiplier are multiplied bit by bit to generate the partial products. Partial products addition stage or reduction of partial products (PPR) is the second stage which is the most important as it is the most complicated and that determines the speed of the overall multiplier and the final addition stage or carry-propagate addition (CPA) using different compressors have been widely employed in the high speed multipliers to lower the latency of the partial product accumulation stage. In order to employ the processor for digital signal processing applications, a modified Wallace tree multiplier this uses compressors circuits to obtain low power and high speed operation in the Arithmetic Logic Unit (ALU) . In digital CMOS design, the well-

known power-delay product is commonly used to evaluate the value the merits of designs. The multiplier architecture comprises of a partial product generation stage, partial product reduction stage and the final addition stage. In the proposed architecture, multi bit compressors are used for realizing the reduction in the number of partial product addition stages.

a) Impact of using the proposed compressors for 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]. a) Partial product generation. b) A Carry Save Adder (CSA) tree to reduce the partial products' matrix to an addition of only two operands C) A Carry Propagation Adder (CPA) for the final computation of the binary result. 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 5.1(a) shows the reduction circuitry of an exact multiplier for $n=8$. In this figure, the reduction part uses half-adders, full-adders and 4-2 compressors; each partial product bit is represented by a dot.

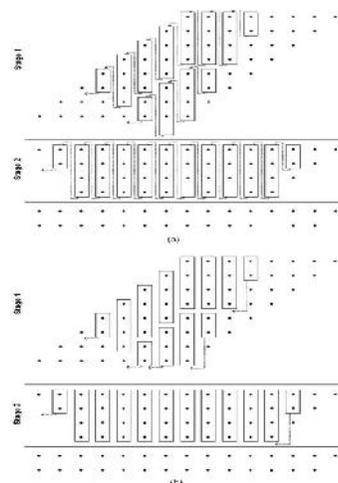


Figure 9) Reduction circuitry of an 8×8 Dadda multiplier, (a) using Design 1 compressors, (b) using Design 2 compressors.

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 then-1 least significant columns. The other n most significant columns in the reduction circuitry use exact 4-2 compressors.

V. Result

In this chapter, the designs of the two approximate compressors (Chapter 4) and the four approximate multipliers (Chapter 5) are simulated using Quartus II 9.1web edition. The Performance of compressor design proposed is compared with that of Exact Design in [8].

6.1 Performance Comparison of Compressor

TABLE 6.1 Performance Comparison of Various Compressor designs

Design	Area(LE)	Delay(ns)	Total Power(mW)
Exact Design [8]	4	11.603	99.09
Proposed Design 1	3	11.653	62.05
Proposed Design 2	2	11.575	47.12

As expected, the second proposed design (Design 2) has the best delay, power consumption; these improvements are irrespective of feature size.

6.2 Performance Comparison of Multiplier Schemes

The four proposed approximate multipliers are simulated for $n=8$. The delay, power consumption and number of Logical Elements are investigated for these approximate designs as well as the exact multiplier.

Delay

The delay of the reduction circuitry (second module) of a Dadda multiplier is dependent on the number of reduction stages and the delay of each stage. In Multipliers 1 and 2, the approximate compressors are used in all columns; therefore, the delay of the stages is equal to the delay of the approximate compressors. However, in Multipliers 3 and 4, the delay of the stages is equal to the delay of the exact compressors. So, the use of these approximate compressors in the $n/2$ LSBs cause no improvement in terms of delay compared to an exact multiplier. The delay reported by Quartus II for the proposed multiplier schemes and exact multiplier is shown in Table 6.2.

Table 6.2. Delay Comparison for Various Multiplier designs

Multiplier Scheme	Delay(ns)
Exact Multiplier	30.860
Multiplier 1	29.270
Multiplier 2	25.233
Multiplier 3	29.010
Multiplier 4	24.783

Power Consumption

The power consumption of each multiplier is determined by the number and type of compressors used. The total power dissipation reported by Quartus II for the proposed multiplier schemes and exact multiplier is shown in Table 6.3.

Table 6.3. Power dissipation Comparison for Various Multiplier designs

Multiplier Scheme	Total Power Dissipation(in mW)
Exact Multiplier	106.97
Multiplier 1	99.93
Multiplier 2	102.97
Multiplier 3	95.25
Multiplier 4	103.45

Logical Element

The Logic Element count is used in this paper as metric of circuit complexity. The first two approximate multipliers have a lower LE count compared with Multipliers 3 and 4. Table 6.4 shows the Total LE reported by Quartus II for the proposed multiplier schemes and exact multiplier.

Table 6.4. LE Comparison for Various Multiplier designs

Multiplier Scheme	Total Logical Elements(LE)
Exact Multiplier	159
Multiplier 1	128
Multiplier 2	114
Multiplier 3	137
Multiplier 4	130

VI. Conclusion

Inexact computing is an emerging paradigm for computation at Nano-scale. Computer arithmetic offers significant operational advantages for inexact computing; an extensive literature exists on approximate adders. However, this paper has initially focused on compression as used in a multiplier; to the best knowledge of the authors, no work has been reported on this topic.

This project has presented the novel designs of two approximate 4-2 compressors. These approximate compressors are utilized in the reduction module of

four approximate multipliers. The approximate compressors show a significant reduction in LE count, power consumption and delay compared with an exact design

In terms of LE count, the first design has a 25% improvement, while the second design has a 50% improvement.

In terms of power consumption, the first design has a 37.38% improvement and the second design has a 52.45% improvement

In terms of delay, the second design has a 44% improvement compared to the exact compressor and 35% improvement compared to the first design .

References

- [1] Liu, D., Embedded DSP Processor Design, Morgan Kaufmann Publishing, 2008. 808 p.
- [2] J. Liang, J. Han, F. Lombardi, “New Metrics for the Reliability of Approximate and Probabilistic Adders,” *IEEE Transactions on Computers*, vol. 63, no. 9, pp. 1760 - 1771, 2013.
- [3] Habibi, A. ; Wintz, P.A. , “Fast Multipliers” *IEEE Transactions on Computers*, Vol. C, No. 19, pp.153-157.
- [4] S.F. Hsiao, M.R. Jiang, J.S. Yeh, (1998) “Design of high low power 3-2 counter and 4-2 compressor for fast multipliers”, *Electronic Letters*, Vol. 34, No. 4, pp. 341-343
- [5] Huddar, S.R. ; Rupanagudi, S.R. ; Kalpana, M. ; Mohan, S. , “Novel High Speed Vedic Mathematics Multiplier using Compressors”, International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013.
- [6] K. Prasad, K.K. Parhi (2001), “Low power 4-2 and 5-2 compressors”, Proceedings of 35th Asilomar Conference on Signals, Systems and Computers, Vol. 1, pp. 129-133.
- [7] J. Gu, C. Chang (2003), “Ultra low voltage low power 4-2 compressor for high speed multiplications”, Proceedings of IEEE International Symposium on Circuits and Systems”, Vol. 5, pp. 321-324.
- [8] 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 I: Regular Papers*, vol. 57, no. 4, pp. 850-862.
- [9] M. J. Schulte and E. E. Swartzlander, Jr., “Truncated multiplication with correction constant”, *VLSI Signal Processing VI*, pp. 388–396.
- [10] E. J. King and E. E. Swartzlander, Jr., “Data dependent truncated scheme for parallel multiplication,” in *Proceedings of the Thirty First Asilomar Conference on Signals, Circuits and Systems*, pp. 1178–1182.
- [11] P. Kulkarni, P. Gupta, and MD Ercegovic, “Trading accuracy for power in a multiplier architecture”, *Journal of Low Power Electronics*, vol. 7, no. 4, pp. 490–501.
- [12] H.-J. Ko and S.-F. Hsiao, “Design and application of faithfully rounded and truncated multipliers with combined deletion, reduction, truncation, and rounding,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 5, pp. 304–308.
- [13] C. 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 & Systems*, Vol. 51, No. 10, pp. 1985-1997.

How to cite this article:

S. Tamilarasan, A. V. Prathap Kumar, V. Vijayakumar. (2018). A Design and implementation of compressor for utilizing in Dadda multiplier. *Int. J. Adv. Multidiscip. Res.* 5(10): 39-45.

DOI: <http://dx.doi.org/10.22192/ijamr.2018.05.10.007>

Access this Article in Online



Website:
www.ijarm.com

Subject:
Engineering

Quick Response
Code

DOI:10.22192/ijamr.2018.05.10.007