

# Survey on Compressor & Dadda Multiplier Inexact Computing Approximate Circuits

Dr. V. PADMANABHA REDDY<sup>1</sup> & Mr. M. MAHABOOB BASHA<sup>2</sup> & BALA GOPAL  
PULLAGOILA<sup>3</sup>

<sup>1</sup>Professor Dept. of ECE, Svr Engineering College Nandyal Mail: - [vpr2008@gmail.com](mailto:vpr2008@gmail.com)

<sup>2</sup>Associate Professor Dept. of ECE, Svr Engineering College Nandyal Mail: - [mmbfasi@gmail.com](mailto:mmbfasi@gmail.com)

<sup>3</sup>PG-Scholar Dept. of ECE, Svr Engineering College Nandyal Mail: - [balagopal410@gmail.com](mailto:balagopal410@gmail.com)

## Abstract

Multipliers are usually deemed as a critical component in digital signal processor design since a large number of multiplications are required in DSP applications. Inexact (or approximate) computing is an attractive paradigm for digital processing at nano metric scales. Inexact computing is particularly interesting for computer arithmetic designs. This paper deals with the analysis and design of two new approximate 4-2 compressors for utilization in a multiplier. These designs rely on different features of compression, such that imprecision in computation can meet with respect to circuit-based figures of merit of a design. Four different schemes for utilizing the proposed approximate

compressors are proposed and analyzed for a Dadda multiplier. Binary logarithms can be used to perform computer multiplication through simple addition. Exact logarithmic conversion is prohibitively expensive for use in general multipliers; however, inexpensive estimate conversions can be used to perform approximate multiplication. Such approximate multipliers have been used in domain-specific applications, but existing designs either offer superior efficiency or flexibility. The results show that the proposed designs accomplish significant reductions in power dissipation, delay and transistor count compared to an exact design.

**Keywords:** Compressor, Dadda Multiplier, Inexact Computing, Approximate Circuits.

## 1. Introduction

With the rapid advances in multimedia and communication systems, real-time signal processing and large capacity data processing are increasingly being demanded. The multiplier is an essential element of the digital signal processing such as filtering and convolution.

Most digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) or discrete wavelet transform (DWT). As they are basically accomplished by repetitive application of multiplication and addition, their speed becomes a major factor which determines the performance of the entire

calculation. Since the multiplier requires the longest delay among the basic operational blocks in digital system, the critical path is determined more by the multiplier. Furthermore, multiplier consumes much area and dissipates more power. Hence designing multipliers which offer either of the following design targets high speed, low power consumption [2], less area or even a combination of them is of substantial research interest. Multiplication operation involves generation of partial products and their accumulation.

The speed of multiplication can be increased by reducing the number of partial products and/or accelerating the accumulation of partial products. Among the many methods of implementing high speed parallel multipliers, there are two basic approaches namely Booth algorithm and Wallace Tree compressors. This paper describes an efficient implementation of a high speed parallel multiplier using both these approaches. Here two multipliers are proposed. The first multiplier makes use of the Radix-4Booth Algorithm with 3:2 compressors while the second multiplier uses the Radix-8 Booth algorithm with 4:2compressors. The design is structured for  $m \times n$  multiplication where  $m$  and  $n$  can reach up to 126 bits. The number of partial products is  $n/2$  in Radix-4 Booth algorithm while it gets reduced to  $n/3$  in Radix-8 Booth algorithm. The Wallace tree uses Carry

Save Adders (CSA) to accumulate the partial products. This reduces the time as well as the chip area. To further enhance the speed of operation, carry-look-ahead(CLA) adder is used as the final adder.

## 2. Related Work

### 2.1 OVERVIEW OF MULTIPLIER:

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore lowpower 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

## 2.2 DIFFERENT TYPES OF COMPRESSOR ARCHITECTURE:

### A. Compressor design

The 4-2 Compressor has 5 inputs  $x_1, x_2, x_3, x_4$  and  $C_{in}$  to generate 3 outputs Sum, Carry and  $C_{out}$  as shown in figure 1(a). The 4 inputs A, B, C and D and the output Sum have the same weight. The input  $C_{in}$  is output from a previous lower significant compressor and the  $C_{out}$  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 3. Different Compressor logic based upon the concept of counter of full adder. Compressor is defined as single bit adder circuit that has more than three inputs as in full adder and less number of outputs. In the proposed architecture which is shown in Fig. 6, the fact that both the XOR and XNOR values are computed is efficiently used to reduce the delay by replacing the second XOR with a MUX. This is due to the possibility of the select bit at the MUX block before the inputs are applied. Thus the time

taken for switching of the transistors in the critical path is highly reduced.

### 1. 4:2 Compressor

The 4-2 compressor which has 4 inputs ( $x_1, x_2, x_3$  and  $x_4$ ) and 2 outputs (Sum & Carry) along with a Carry-in ( $C_{in}$ ) and a Carry-out ( $C_{out}$ ) as shown in Fig 3. The input  $C_{in}$  is the output from the neighboring lower significant compressor.

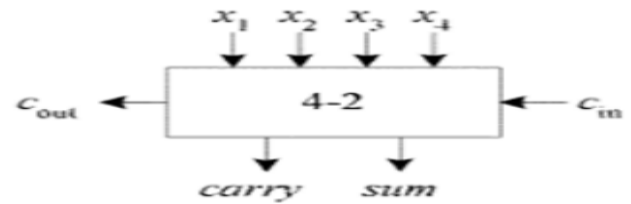


Fig.1. 4:2 Compressor.

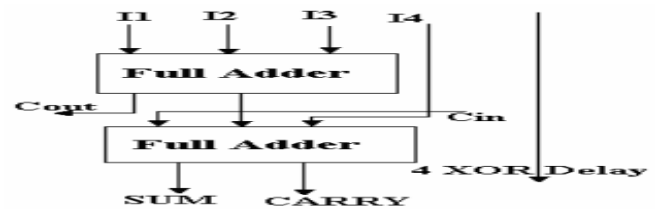


Fig.2. 4:2 Compressor using full adder.

The  $C_{out}$  is the output to the next significant stage compressor. It consists of two 3-2 compressors (full adders) in series and involves a critical path of 4 XOR delays which is shown in fig.4. An alternative implementation is shown in Fig.5. This implementation is better and involves a critical path delay of three XOR's, hence reducing the critical path by 1 XOR delay.

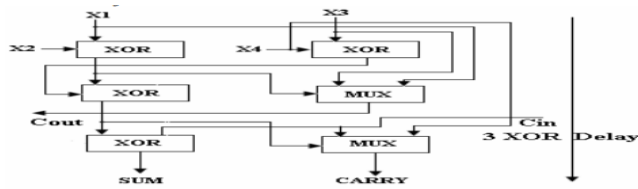


Fig.3. Alternative implementation of 4:2 compressors.

### B. 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, a compressor which reduces the complexity of the multiplier is introduced.

### 1. Dadda Multiplier using Design1

A 8×8 unsigned Dadda tree multiplier is considered to access the impact 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 halfadder, 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.

## 3. Implementation

In this section, the impact of using the proposed compressors for multiplication is investigated. A fast (exact) multiplier is usually composed of three parts.

### A. Partial product generation.

1. A Carry Save Adder (CSA) tree to reduce the partial products' matrix to an addition of only two operands
2. 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  $8 \times 8$  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  $n=8$ . In this figure, the reduction part uses halfadders, 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  $8 \times 8$  Dadda multiplier.



Fig4. Reduction circuitry of an  $8 \times 8$  Dadda multiplier, (a) using Design 1 compressors, (b) using Design 2 compressors.

In this paper, four cases are considered for designing an approximate multiplier. 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. In the fourth case (Multiplier 4), Design 2 and exact 4-2 compressors are used in then-1 least significant columns and then 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

### 4. Experimental Result

Compressor written in verilog, compiled and simulation using modelsim. The circuit simulated and synthesized. The simulated result for Multipliers using compressor.

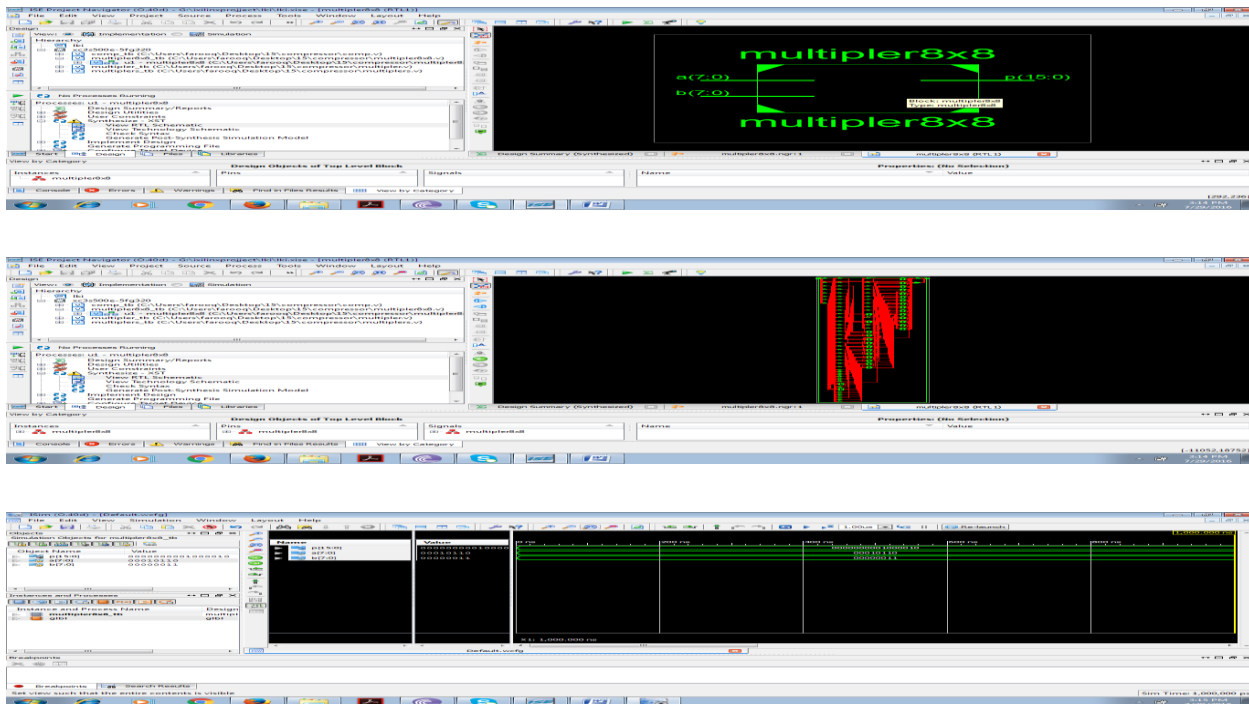


Fig.5 Simulation Result.

### 5. Conclusion

In this paper, four designs of 8x8 bit approximate multipliers have been proposed. Simulation results have been reported for design and error metrics. Also, an image processing application has been presented in detail. The proposed designs show significant

improvements in accuracy, power and latency at a cost of a slightly larger area.

### 6. References

[1] 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.



- [2] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," *Low Power Electronics and Design (ISLPED) 2011 International Symposium on*, 1-3 Aug. 2011.
- [3] S. Cheemalavagu, P. Korkmaz, K.V. Palem, B.E.S. Akgul, and L.N. Chakrapani, "A probabilistic CMOS switch and its realization by exploiting noise," in *Proc. IFIP-VLSI SoC, Perth, Western Australia, Oct. 2005*.
- [4] 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, April 2010.
- [6] M. J. Schulte and E. E. Swartzlander, Jr., "Truncated multiplication with correction constant," *VLSI Signal Processing VI*, pp. 388–396, 1993.
- [7] 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, 1998.
- [8] P. Kulkarni, P. Gupta, and MD Ercegovac, "Trading accuracy for power in a multiplier architecture," *Journal of Low Power Electronics*, vol. 7, no. 4, pp. 490--501, 2011.
- [9] C. Chang, J. Gu, M. Zhang, "Ultra Low-Voltage LowPower CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits," *IEEE Transactions on Circuits & Systems*, Vol. 51, No. 10, pp. 1985-1997, Oct. 2004.
- [10] D. Radhakrishnan and A. P. Preethy, "Low-power CMOS pass logic 4-2 compressor for high-speed multiplication," in *Proc. 43rd IEEE Midwest Symp. Circuits Syst.*, vol. 3, 2000, pp. 1296–1298.
- [11] ] Z. Wang, G. A. Jullien, and W. C. Miller, "A new design technique for column compression multipliers," *IEEE Trans. Compu*
- [12] J. Ma, K. Man, T. Krilavicius, S. Guan, and T. Jeong, "Implementation of High Performance Multipliers Based on Approximate Compressor Design" in *international Conference on Electrical and Control Technologies (ECT)*, 2011.