



## DESIGN OF EFFICIENT APPROXIMATE MULTIPLICATION CIRCUITS USING APPROXIMATE 4:2 COMPRESSORS

AMMINENI LAVANYA<sup>1</sup>, M.AMARANATHREDDY<sup>2</sup>

<sup>1</sup>PG Scholar, Dept of ECE, SIR C.V. RAMAN Institute of Technology & Science, AP, India

<sup>2</sup> Assistant Professor, Dept of ECE, SIR C.V. RAMAN Institute of Technology & Science, AP, India

**Abstract:** Compressors have become essential components for partial product reduction stage of multiplier architectures. Formerly, different adders like carry save adders were used for partial product reduction but with the requirement of low power and smaller area, adders were replaced with different order compressors. Machine Learning (ML) has been one of the applications of approximate circuits. These circuits, part of approximate computing, can be implemented using either probabilistic pruning or inexact logic minimization. Since low power consumption and smaller silicon area are the critical parameters in portable devices, approximate circuits have been the current topic for discussion. This project presents a 4:2 compressors with inexact logic minimization by flipping some of the output bits considering efficiency/accuracy into account. The proposed 4:2 compressor has been utilized in an  $8 \times 8$  Dadda multiplier and average power, area and propagation delay of the architectures have been computed.

### 1.INTRODUCTION

Compressors have become essential components for partial product reduction stage of multiplier architectures. Formerly, different adders like carry save adders were used for partial product reduction but with the requirement of low power and smaller area, adders were replaced with different order compressors such as 3:2, 4:2, 5:2.

MOST 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. Approximate (or inexact) computing relies on using this property to design simplified, yet approximate circuits operating at higher performance and/or lower power consumption compared with precise (exact) logic circuits. Addition and multiplication are widely used operations in computer arithmetic;



for addition full-adder cells have been extensively analyzed for approximate computing.

Liang et al. has compared these adders and proposed several new metrics for evaluating approximate and probabilistic adders with respect to unified figures of merit for design assessment for inexact computing applications. For each input to a circuit, the error distance (ED) is Defined as the arithmetic distance between an erroneous output and the correct one. The mean error distance (MED) and normalized error distance (NED) are proposed by considering the averaging effect of multiple inputs and the normalization of multiple-bit adders. The NED is nearly invariant with the size of an implementation and

is therefore useful in the reliability assessment of a Specific design. The tradeoff between precision and power has also been quantitatively evaluated. However, the design of approximate multipliers has received less attention. Multiplication can be thought as the repeated sum of partial products; however, the straightforward application of approximate adders when designing an approximate multiplier is not viable, because it would be very inefficient in terms of precision, hardware complexity and other performance metrics. Most of these designs use a truncated multiplication method; they estimate the least significant columns of the partial products as a constant. In an imprecise array multiplier is used for neural network applications by omitting some of the least significant bits in the partial products (and thus removing some adders in the array). A truncated

multiplier with a correction constant is proposed.

## 2.LITERATURE SURVEY

### **Novel architectures for high-speed and lowpower 3-2, 4-2 and 5-2 compressors by Veeramachaneni et.al.,**

The 3-2, 4-2 and 5-2 compressors are the basic components in many applications, in particular partial product summation in multipliers. In this paper novel architectures and designs of high speed, low power 3-2, 4-2 and 5-2 compressors capable of operating at ultra-low voltages are presented. The power consumption, delay and area of these new compressor architectures are compared with existing and recently proposed compressor architectures and are shown to perform better. The proposed architecture lays emphasis on the use of multiplexers in arithmetic circuits that result in high speed and efficient design. Also in all existing implementations of XOR gate and multiplexers, both output and its complement are available but current designs of compressors do not use these outputs efficiently. In the proposed architecture these outputs are efficiently utilized to improve the performance of compressors. The combination of low power, low transistor count and lesser delay makes the new compressors a viable option for efficient design

### **Ultra low voltage, low power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits, by Chang et.al**

This paper presents several architectures and designs of low-power 4-2 and 5-2 compressors capable of operating at ultra low supply voltages. These compressor architectures are anatomized into their



constituent modules and different static logic styles based on the same deep submicrometer CMOS process model are used to realize them. Different configurations of each architecture, which include a number of novel 4-2 and 5-2 compressor designs, are prototyped and simulated to evaluate their performance in speed, power dissipation and power-delay product. The newly developed circuits are based on various configurations of the novel 5-2 compressor architecture with the new carry generator circuit, or existing architectures configured with the proposed circuit for the exclusive OR (XOR) and exclusive NOR (XNOR) [XOR-XNOR] module. The proposed new circuit for the XOR-XNOR module eliminates the weak logic on the internal nodes of pass transistors with a pair of feedback PMOS-NMOS transistors. Driving capability has been considered in the design as well as in the simulation setup so that these 4-2 and 5-2 compressor cells can operate reliably in any tree structured parallel multiplier at very low supply voltages. Two new simulation environments are created to ensure that the performances reflect the realistic circuit operation in the system to which these cells are integrated. Simulation results show that the 4-2 compressor with the proposed XOR-XNOR module and the new fast 5-2 compressor architecture are able to function at supply voltage as low as 0.6 V, and outperform many other architectures including the classical CMOS logic compressors and variants of compressors constructed with various combinations of recently reported superior low-power logic cells.

## **A Power-Efficient 4-2 Adder Compressor Topology by Raphael, D et.al**

The addition is the most used arithmetic operation in Digital Signal Processing (DSP) algorithms, such as filters, transforms and predictions. These algorithms are increasingly present in audio and video processing of battery-powered mobile devices having, therefore, energy constraints. In the context of addition operation, the efficient 4-2 adder compressor is capable to performs four additions simultaneously. This higher order of parallelism reduces the critical path and the internal glitching, thus reducing mainly the dynamic power dissipation. This work proposes two CMOS+ gate-based topologies to further reduce the power, area and delay of the 4-2 adder compressor. The proposed CMOS+ 4-2 adder compressor circuits topologies were implemented with Cadence Virtuoso tool at 45 nm technology and simulated in both electric and layout levels. The results show that a proper choice of gates in 4-2 adder compressor realization can reduce the power, delay and area about 22.41%, 32.45% and 7.4%, respectively, when compared with the literature.

## **Approximate Computing and the Quest for Computing Efficiency by Swagath, V et.al**

Diminishing benefits from technology scaling have pushed designers to look for new sources of computing efficiency. Multicores and heterogeneous accelerator-based architectures are a by-product of this quest to obtain improvements in the performance of computing platforms at similar or lower



power budgets. In light of the need for new innovations to sustain these improvements, we discuss approximate computing, a field that has attracted considerable interest over the last decade. While the core principles of approximate computing - computing efficiently by producing results that are good enough or of sufficient quality - are not new and are shared by many fields from algorithm design to networks and distributed systems, recent efforts have seen a percolation of these principles to all layers of the computing stack, including circuits, architecture, and software. Approximate computing techniques have also evolved from ad hoc and application-specific to more broadly applicable, supported by systematic design methodologies. Finally, the emergence of workloads such as recognition, mining, search, data analytics, inference and vision are greatly increasing the opportunities for approximate computing. We describe the vision and key principles that have guided our work in this area, and outline a holistic cross-layer framework for approximate computing.

### **Parsimonious Circuits for Error-Tolerant Applications through Probabilistic Logic Minimization by Avinash, L et.al**

Contrary to the existing techniques to realize *inexact* circuits that relied mostly on scaling of supply voltage or pruning of “least-significant” components in conventional correct circuits to achieve cost (energy, delay and/or area) and accuracy tradeoffs, we propose a novel technique called *Probabilistic Logic Minimization* which relies on synthesizing an inexact circuit in the first

place resulting in *zero hardware overhead*. Extensive simulations of the datapath elements designed using the proposed technique demonstrate that normalized gains as high as 2X-9.5X in the Energy-Delay-Area product can be obtained when compared to the corresponding correct designs, with a relative error magnitude percentage as low as 0.001% upto 1%.

### **3.EXISTING SYSTEM**

Luigi WALLACE, the computer scientist has invented the WALLACE hardware multiplier during 1965. WALLACE multiplier is extracted form of parallel multiplier [5]. It is slightly faster and requires fewer gates. Different types of schemes are used in parallel multiplier. The WALLACE scheme is one of the parallel multiplier schemes that essentially minimize the number of adder stages required to perform the summation of partial products. This is achieved by using full and half adders to reduce the number of rows in the matrix number of bits at each summation stage. Even though the WALLACE multiplication has regular and less complex structure, the process is slower in manner due to serial multiplication process. Further, WALLACE multiplier is less expensive compared to that of Wallace tree multiplier. Hence, in this paper, WALLACE multiplier is designed and analysed by considering different methods using full adders involving different logic styles.

### **Wallace Tree Multiplier Using Ripple Carry Adder**

Ripple Carry Adder is the method used to add more number of additions to be performed with the carry in and carry outs that is to be chained.



Thus multiple adders are used in ripple carry adder. It is possible to create a logical circuit using several full adders to add multiple-bit numbers. Each full adder inputs a  $C_{in}$ , which is the  $C_{out}$  of the previous adder. This kind of adder is a ripple carry adder, since each carry bit "ripples" to the next full adder. The proposed architecture of WALLACE multiplier algorithm using RCA is shown in Figs.9 to 11 Take any 3 values with the same weights and gives them as input into a full adder. The result will be an output wire of the same weight. Partial product obtained after multiplication is taken at the first stage. The data's are taken with 3 wires and added using adders and the carry of each stage is added with next two data's in the same stage. Partial products reduced to two layers of full adders with same procedure. At the final stage, same method of ripple carry adder method is performed and thus product terms  $p_1$  to  $p_8$  is obtained.

#### 4.PROPOSED SYSTEM

The Dadda multiplier is a hardware multiplier design, invented by computer scientist Luigi Dadda in 1965. It is one type of parallel multiplier. 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 less expensive reduction phase. Compared to a Wallace tree, which requires ten full adders and half adders, the reduction phase of the Dadda multiplier requires only six [2]. The Dadda Multiplier requires less hardware than the Wallace. Compressors are primary component of the multiplier. Large delay was observed in partial

products addition stage that increase the amount of power consumed. Using compressor adders, that add four, five, six or seven bits at a time, the number of full adders and half adders are reduced and hence the power consumed is less. Compressors are building blocks used for accumulating partial products during the multiplication process [3]. The basic idea in an  $n: 2$ compressor is that  $n$  operands can be reduced to two, by doing the addition while keeping the carries and sums separate.

Dadda multipliers do as few reductions as possible. Because of this, dadda multipliers have a less expensive reduction phase, but the numbers may be a few bits longer, thus requiring slightly bigger adders

Two designs of an approximate compressor are presented. Intuitively to design an approximate 4-2 compressor, it is possible to substitute the exact full-adder cells in Figure by an approximate full-adder cell. 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% (where the error rate is given by the ratio of the number of erroneous outputs over the total number of outputs). Two different designs are presented to reduce the error rate; these designs offer significant performance improvement compared to an exact compressor with respect to delay, number of transistors and power consumption. The approximate compressor design drives the inputs as  $x_1, x_2, x_3, x_4, c_{in}$  and the outputs are sum, carry, and  $c_{out}$ . And in the existing system, the implement the multiplier by

the compressor using dadda's tree. The output equations are given below,

$$\text{Carry} = \text{cin}$$

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

$$\text{Cout} = (\overline{x_1}x_2 + x_3x_4)$$

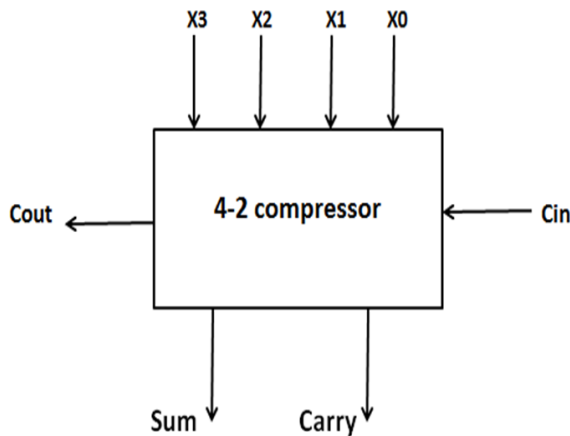


Fig 1 4:2 Compressor

On comparison of exact [1] and approximated 4:2 compressor, the number of components is increased in the form of AND and OR gates in the proposed structure but the XORXNOR and MUX circuits of the exact compressor is using increased number of transistors, which increases the power consumption of this circuit and also the multiplier structure. The power consumption and circuit size of the proposed 4:2 compressor has been reduced due to usage of all pass transistor logic based modules like AND, OR, 2T MUX and 6T XORXNOR and the transistor count of the proposed compressor is reduced to 34 from 52 [2] and 50 [1].

This research work employs probabilistic logic minimization on exact 4:2 compressor, where bit flipping in the minterms of Boolean functions of SUM,

Carry and Cout is done to minimize the number of literals, thereby reducing power consumption, area and delay of the circuit. To identify the favorable bit flips, different combinations are attempted at the expense of error which is proportional to the number of bit flips introduced. After implementing this process on any exact circuit, the size of the circuitry should decrease with less error rate.

Since MSB's plays important role than LSB's in producing the result, lower order bits of 4:2 compressor are flipped from 1's to 0's which produces an error rate (ratio of no. of inexact outputs to correct outputs) of 25% without loss in the count of number of inputs and outputs. The bits can be flipped either from '1 to 0' or '0 to 1', but in this paper '1 to 0' is chosen to reduce the size of the circuitry. Cout, Carry, SUM bits are flipped by 8, 4, 8 in number respectively. The Sum, Carry and Cout

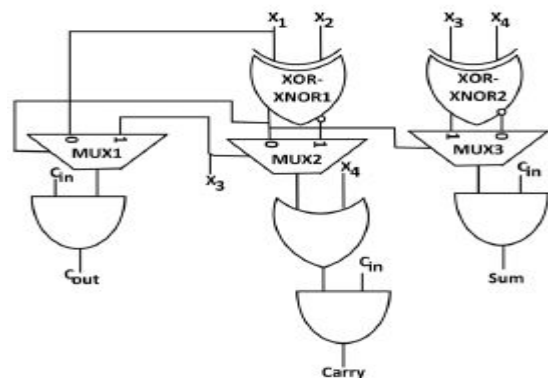


Fig. 2. Proposed Approximate 4:2 Compressor

In the proposed design the maximum number of bit flips is only 8 and hence the error rate is 25%. When compared the proposed design with the error rate of design1 and design2 in it are 37.5%

and 25% respectively. The error rate of the proposed design is same as the design2 of but the number of inputs is only four with the removal of carry  $C_{in}$  in [8]. In the proposed design, the number of inputs is same as the exact compressor but with little increase in the transistor count and power consumption than.

Out of all the parallel multipliers, Dadda multiplier is the fastest one. Hence, in this work, multiplication operation is performed using this technique. For the three phases of multiplication, basic AND gates were employed to generate the partial products, Carry Save and Carry Propagate Adders were used for second and third stages respectively in the literature.

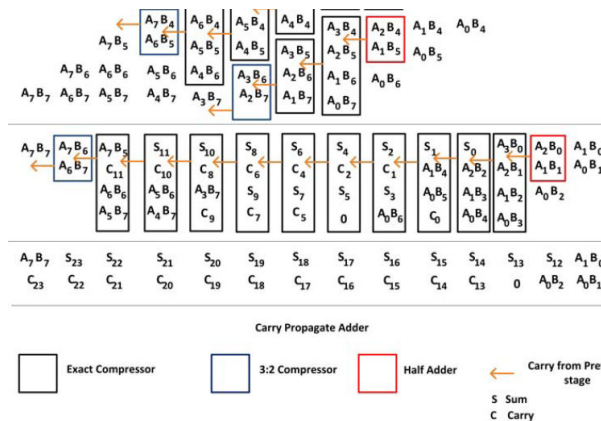
Out of the three, the second stage is the critical one in the design of a multiplier since it slows down the operation and consumes more power. Thus, compressors are being employed in the second stage which reduces the power consumption and speeds up the operation of the entire multiplier circuit. The multiplier structure with exact compressors is shown in Fig.5.2.

In this section, the proposed approximate compressor is placed to make an exact multiplier as approximate one. An  $n=8$  Dadda multiplier is designed in this work employing AND gates, the proposed approximated 4:2 compressor in the second stage and an exact carry propagate adder in the third stage.

In this project, five Dadda multipliers have been simulated.

- The first multiplier employs all exact 4:2 compressors of [2].
- The second multiplier also employs all exact 4:2 compressors of [1].
- The third multiplier uses exact multipliers of [2] in the first stage of Dadda multiplier and proposed approximated 4:2 compressor in the second stage. Exact compressors are utilized in the first stage to enable approximations only from the second stage.
- The fourth multiplier uses exact multipliers of [1] in the first stage of Dadda multiplier and proposed approximated 4:2 compressor in the second stage.
- The fifth multiplier uses all approximate proposed 4:2 compressors in the first and second stages of Dadda multiplier.

The first two are exact multipliers with zero error distance but consumes more power with increased delay. The third and fourth are partially approximate multipliers since only one stage of the multiplier is utilizing approximate compressor. The fifth is completely approximate multiplier since it is employing all approximate multipliers and thus error distance increases while power consumption and area decreases. The architectures of third and fourth



**Fig. 3. Multiplier with Exact Compressors**

multipliers are not shown here as the the structure looks same as Fig. 5.2 but the exact compressors in stage1 is replaced with approximate 4:2 compressor and the fifth multiplier has all proposed compressors. The architecture of the multipliers does not vary since there is no change in number of inputs and outputs

## 5. RESULTS

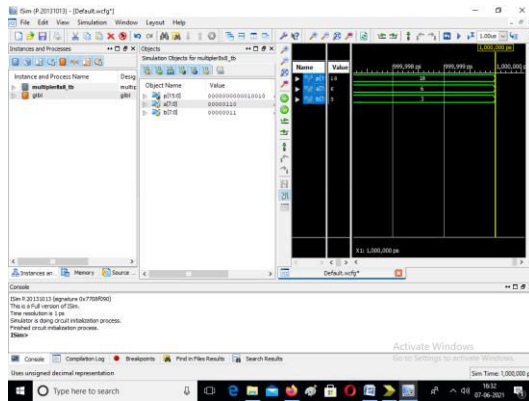


Fig 4 Simulation Result of Multiplier  
Here we can give the inputs as  $A=6$ ,  
 $B=3$ , then the final output is 18.

## 6 CONCLUSION

A demonstration of approximate 4:2 compressors employing inexact logic minimization with error analysis has been presented in this paper and the state-of-art 4:2 compressor has been positioned in  $8 \times 8$  Dadda multiplier. The comparison results shows that the average power has been diminished for the proposed circuits and have Algorithm 1 ML in Approximate Circuits 16 to 31 inputs of 4:2 compressor are remained unchanged. less error rate when considered with the number of transistors. Finally, as an application, the concept of connecting VLSI approximate circuits with machine learning

## 7 FUTURE SCOPE

This multipliers plays a very important role in our day to day life. In future the multipliers are going to play a major role. The speed of the multipliers are increased by using carry save adders, carry look ahead adder, and so on. Rounding patterns will be optimized based on required accuracy and different compression techniques. The area and delay can be reduced in future by using advanced technology.

## BIBLIOGRAPHY

- [1] Veeramachaneni et.al., Novel architectures for high-speed and low power 3-2, 4-2 and 5-2 compressors, Proc. Int. Conf. on VLSI Design (VLSID), 2007, pp. 324-329.
- [2] Chang et.al, Ultra low voltage, low power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits, IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., 2004, 51, (10), pp. 1985-1997.
- [3] Raphael, D et.al, A Power-Efficient 4-2 Adder Compressor Topology, 15th IEEE (NEWCAS), Strasbourg, France, 2017, pp. 281-284.
- [4] Swagath, V et.al, Approximate Computing and the Quest for Computing Efficiency, 52nd (DAC), 2015, San Francisco, CA, USA.
- [5] Shaahin, A et.al, Majority-Based Spin-CMOS Primitives for Approximate Computing, IEEE Trans. on Nanotech., 17(4), 2018, pp. 795-806.
- [6] Avinash, L et.al, Parsimonious Circuits for Error-Tolerant Applications through Probabilistic Logic Minimization, Int. Workshop on PATMOS 2011, pp.204-213.
- [7] Darjn, E et.al, Approximate Multipliers Based on New Approximate





- Compressors, IEEE Trans. on CAS-I: Reg. Pap, PP(99), 2018, pp. 1-14.
- [8] Momeni, A et.al, Design and analysis of approximate compressors for multiplication, IEEE Trans. on Comp., 64 (4), 2015, pp. 984994.
- [9] Liang, J et.al, New Metrics for the Reliability of Approximate and Probabilistic Adders, IEEE Trans. on Comp., 63(9), 2013, p. 1760-1771.
- [10] Zervakis, G et.al, Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation, IEEE Trans. on VLSI Systems, 24(10), 2016, pp. 3105-3117.
- [11] Sina,B et.al, Exploration of approximate multipliers design space using carry propagation free compressors, ASP-DAC, 2018, Jeju, southkorea, pp.611-616.
- [12] Peter, A et.al, "Opportunities for Machine Learning in Electronic Design Automation," in ISCAS, 2018, Italy.
- [13] Kumud,N et.al, ABACUS: A Technique for Automated Behavioral Synthesis of Approximate Computing Circuits, DATE, 2014, Dresden, Germany.
- [14] Schlachter,J et.al, Design and Applications of Approximate Circuits by Gate-Level Pruning, IEEE Trans. on VLSI Systems 25 (5), 2017, pp.1694-1702.