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

Design and Analysis of Approximate Compressors for Multiplication

Ms.Vinutha H
M.Tech ES & VLSI,
Malla Reddy Collage of Engineering.

Mr.M Chandra Mohan, B.E, M.Tech
Assistant Professor,
Malla Reddy Collage of Engineering.

ABSTRACT:
Inexact (or approximate) computing is an attractive paradigm for digital processing at nanometric 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 (as measured by the error rate and the so-called normalized error distance) can meet with respect to circuit-based figures of merit of a design (number of transistors, delay and power consumption). Four different schemes for utilizing the proposed approximate compressors are proposed and analyzed for a Dadda multiplier. Extensive simulation results are provided using Xilinx14.4 ISE. The results show that the proposed designs accomplish significant reductions in delay and area compared to an exact design.

Index Terms:
Compressor, Dadda multiplier, inexact computing, approximate circuits

I.INTRODUCTION:

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

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

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. For an nXn multiplier, this design calculates the sum of the n+k most significant columns of the partial products and truncates the other n-k columns. Then n+k bit result is then rounded to n bits.

The reduction error (i.e., the error generated by truncating the n-k least significant bits) and rounding error (i.e., the error generated by rounding the result to n bits) are found in the next step. The correction constant (n+k bits) is selected to be as close as possible to the estimated value of the sum of these errors to reduce the error distance. Initially in this paper, two novel approximate 4-2 compressors are proposed and analyzed. It is shown that these simplified compressors have better delay and power consumption than the optimized (exact) 4-2 compressor designs. These approximate compressors are then used in the restoration module of a DADDA multiplier; four different schemes are proposed for inexact multiplication. Extensive simulation results are provided at circuit-level for figures of merit, such as delay, transistor count, power dissipation, error rate and normalized error distance under CMOS feature sizes of 32, 22 and 16 nm. The application of these multipliers to image processing is then presented.

The analysis and simulation results show that the proposed approximate designs for both the compressor and the multiplier are viable candidates for inexact computing. A multitude of various multiplier architectures have been published in the literature, during the past few decades. The multiplier is one of the key hardware blocks in most of the digital and high performance systems such as digital signal processors and microprocessors. With the recent advances in technology, many researchers have worked on the design of increasingly more efficient multipliers. They aim at offering higher speed and lower power consumption even while Occupying reduced silicon area. This makes them compatible for various complex and portable VLSI circuit implementations [2]. However, the fact remains that the area and speed are two conflicting performance constraints. Hence, innovating increased speed always results in larger area. The proposed architecture enhances the speed performance of the widely acknowledged Wallace tree multiplier when implemented on a FPGA.

The structural optimization is performed on the conventional Wallace multiplier, in such a way that the latency of the total circuit reduces considerably.A truncated multiplier with constant correction has the maximum error if the partial products in the n-k least significant columns are all ones or all zeros. A variable correction truncated multiplier has been proposed. This method changes the correction term based on column n-k-1. If all partial products in column n-k-1 are one, then the correction term is increased. Similarly, if all partial products in this column are zero, the correction term is decreased. In a simplified 22 multiplier block is proposed for building larger multiplier arrays. In the design of a fast multiplier, compressors have been widely used to speed up the partial product reduction tree and decrease power dissipation. Kelly et al. and Ma et al. have also considered compression for approximate multiplication. An approximate signed multiplier has been proposed for use in arithmetic data value speculation (AVDS); multiplication is performed using the BaughWooley algorithm.

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

However no new design is proposed for the compressors for the inexact computation. Designs of approximate compressors have been proposed; however, these designs do not target multiplication. It should be noted that the approach of improves over by utilizing a simplified multiplier block that is amenable to approximate multiplication.

II.EXACT COMPRESSORS:

The main goal of either multi-operand carry-save addition or parallel multiplication is to reduce n numbers to two numbers; therefore, n-2 compressors (or n-2 counters) have been widely used in computer arithmetic. A n-2 compressor (Fig. 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 or i+2. For the correct operation of the circuit shown in Fig.1, the following inequality must be satisfied

$n+\varphi_1+\varphi_2+\varphi_3+\ldots.<=3+2\varphi_1+4\varphi_2+8\varphi_3+\ldots..$ (1)

Where φ_j denotes the number of carry bits from slice i to slice i+j.

A widely used structure for compression is the 4-2 compressor; a 4-2 compressor (Fig. 2) can be implemented with a carry bit between adjacent slices($\varphi_1 = 1$). The carry bit from the position to the right is denoted as Cin while the carry bit into the higher position is denoted as Cout. The two output bits in positions i and i+1 are also referred to as the sum and carry respectively.

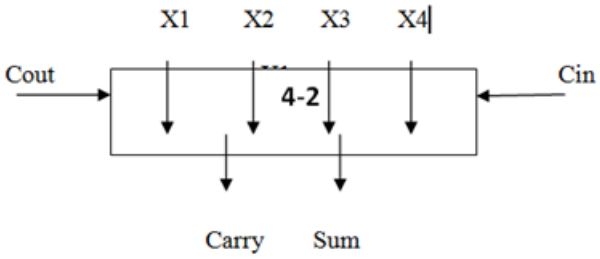

Fig.1 4-2 compressor

The following equations give the outputs of the 4-2 compressor, while Table 1 shows its truth table

$$Sum = x1 \oplus x2 \oplus x3 \oplus x4 \oplus Cin \quad \ldots. (2)$$

$$Cout = (x1 \oplus x2)x3 + \overline{(x1 \oplus x2)}x1 \quad \ldots.(3)$$

$$Carry = (x1 \oplus x2 \oplus x3 \oplus x4)Cin + \overline{(x1 \oplus x2 \oplus x3 \oplus x4)}x4 \quad \ldots.(4)$$

The common implementation of a 4-2 compressor is accomplished by utilizing two full-adder (FA) cells. Fig. 4 shows the optimized design of an exact 4-2 compressor based on the XOR-XNOR gates; A XOR-XNOR gate simultaneously generates the XOR and XNOR output signals. The design consists of three XOR-XNOR (denoted by XOR) gates, one XOR and two 2-1 MUXes. The critical path of this design has a delay of 3D, where D is the unitary delay through any gate in the design.

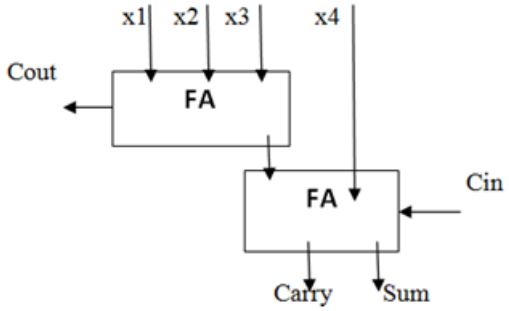

Fig.2 Implementation of 4-2 compressor

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

Table I: Truth Table of 4-2 Compressor

Cin	X4	X3	X2	X1	Cout	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

III.PROPOSED COMPRESSORS:

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 Fig. 3 by an approximate full-adder cell (such as the first design proposed). 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 percent (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 proposed next 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 exact compressor was reduced by proposing two approximate compressors. The two approximate compressors are shown below

A. Design 1:

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. As shown in Table 2.5, 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 eight outputs.

$$Carry' = Cin \quad \ldots(5)$$

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 2), 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. In particular, the simplification of sum to a value of 0 (second half of Table 2) 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 (because it is on the critical path).

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

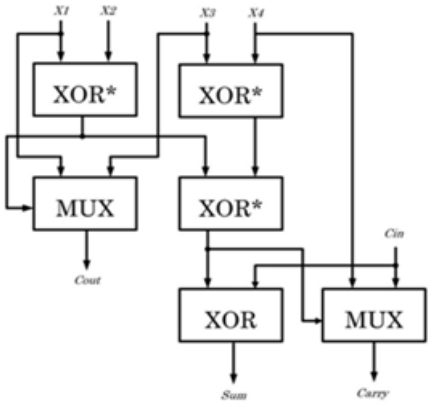

Fig.3 Optimized 4-2 compressor of design 1

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

In the last step, 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.

$$Cout' = \overline{(\overline{x1x2} + \overline{x3x4})} \quad \text{....(7)}$$

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. Table 2 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 2, the proposed design has 12 incorrect outputs out of 32 outputs (thus yielding an error rate of 37.5 percent).

This is less than the error rate using the best approximate full-adder cell.

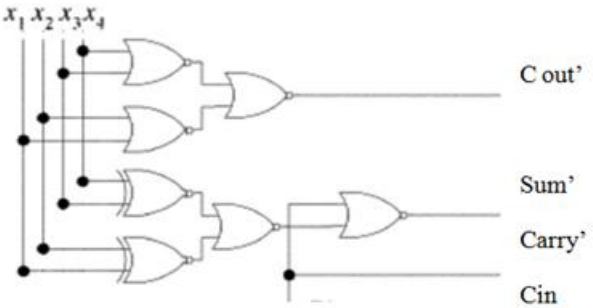

Fig.4 Gate level implementation of design 1

Equations 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 (Fig. 2.2(b)) shows that the critical path of this compressor has still a delay of 3D, so it is the same as for the exact compressor. 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 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.

Table II: Truth table of Approximate Compressor

Cin	X4	X3	X2	X1	Cout'	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. 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 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. Fig. 2.5(c) shows the block diagram of this approximate 4-2 compressor and the expressions below describe its outputs.

$$Sum' = \overline{(x1 \oplus x2 + \overline{x3 \oplus x4})} \quad \text{....(8)}$$

$$Carry' = \overline{(\overline{x1x2} + \overline{x3x4})} \quad \text{....(9)}$$

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

Fig. 2.5(d) shows the gate level implementation of the second proposed design. The delay of the critical path of this approximate design is 2Δ. So it is 1Δ less than the previous designs; moreover, a further reduction in the number of gates is accomplished.

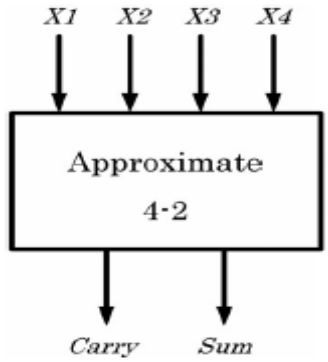

Fig.5 Optimized 4-2 compressor of design 2

Table III 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.

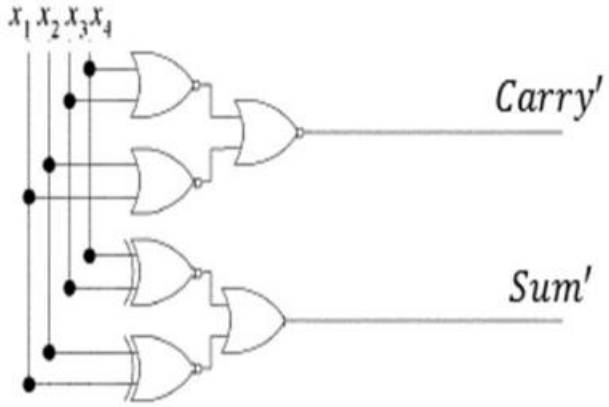

Fig.6 Gate level implementation of design 2

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 2 shows that the difference is –1. This design has therefore four incorrect outputs out of 16 outputs, so its error rate is now reduced to 25 percent.

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.

Table III: Truth table of Approximate Compressor Design 2

X4	X3	X2	X1	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

4. PROPOSED 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. 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).

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

- 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.

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 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 8x8 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. Fig. 9a shows the reduction circuitry of an exact multiplier for n ¼ 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, two half-adders, two full-adders and eight 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 three half-adders, three full-adders and 18 compressors are needed in the reduction circuitry of an 8x8 DADDA multiplier. 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 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. Multiplier 2 uses six half-adders, one 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. The DADDA multiplier was designed by the scientist Luigi Dadda in 1965. It looks similar to Wallace multiplier but slightly faster and requires 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.

A 8x8 multiplier using dada multiplier design is designed. Instead of using conventional full adders and

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

half adder for designing the multiplier, compressors which reduce the complexity of the multiplier is introduced.

A.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 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.

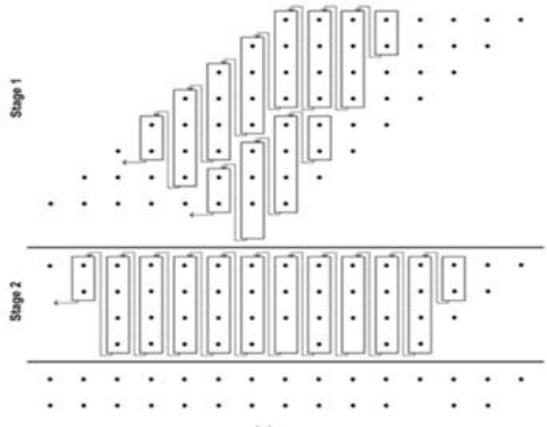

Fig.7.a Reduction circuitry of 8x8 Dadda Multiplier using design 1 Compressor

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

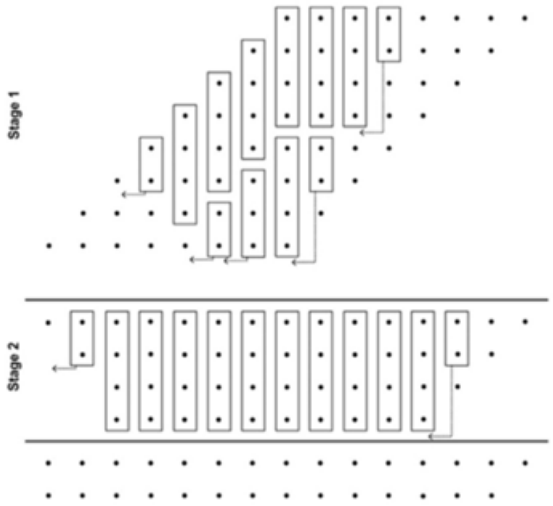

Fig.7.b Reduction circuitry of 8x8 Dadda Multiplier using design 2 Compressor

V.SIMULATION RESULTS:

All the synthesis and simulation results are performed using Verilog HDL. The synthesis and simulation are performed on Xilinx ISE 14.4. The simulation results are shown below figures. The corresponding simulation results of the Approximate Muliplier4 are shown below.

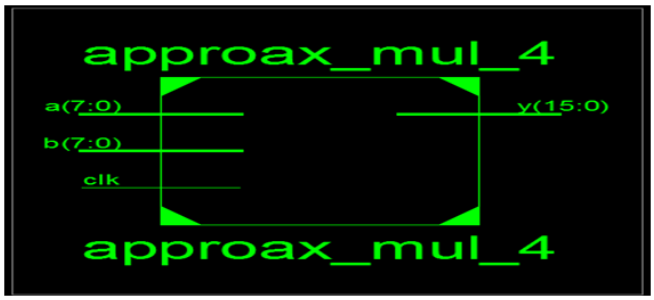

Figure 8: RTL schematic of Approximate Multiplier4

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

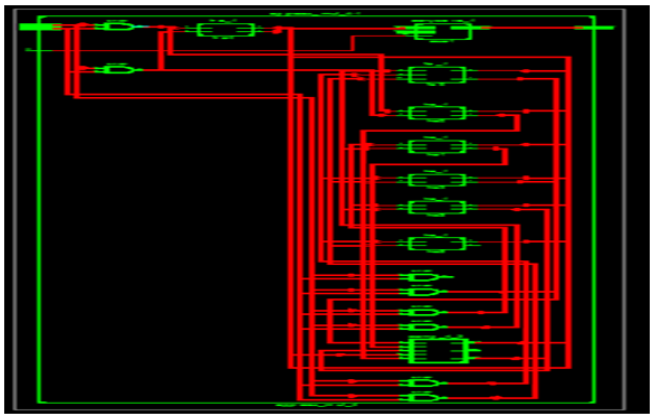

Figure 9: RTL schematic of Internal Approximate Multiplier4

FINAL_ISE Project Status			
Project File:	FINAL_ISE.ise	Current State:	Synthesized
Module Name:	approax_mul_4	• Errors:	No Errors
Target Device:	xc3s500e-5fg320	• Warnings:	6 Warnings
Product Version:	ISE 9.2i	• Updated:	Thu Feb 2 21:56:19 2017

FINAL_ISE Partition Summary
No partition information was found.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	84	4656	1%
Number of Slice Flip Flops	14	9312	0%
Number of 4 input LUTs	151	9312	1%
Number of bonded IOBs	33	232	14%
Number of GCLKs	1	24	4%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Thu Feb 2 21:56:14 2017	0	6 Warnings	0
Translation Report					
Map Report					

Figure 11: Synthesis report of Approximate Multiplier4

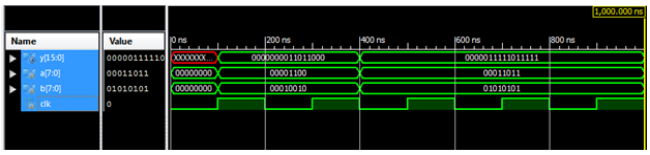

Figure 12: Simulated output for Approximate Multiplier4

Figure 13: Simulated output for Approximate Multiplier3

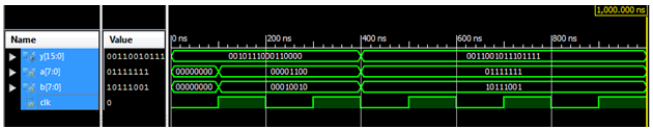

Figure 14: Simulated output for Approximate Multiplier2

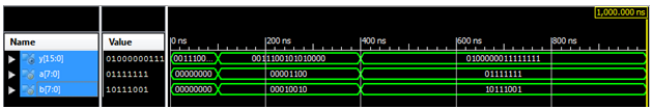

Figure 15: Simulated output for Approximate Multiplier1

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 paper 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 Area, power consumption and delay compared with an exact design. Four different approximate schemes have been proposed in this paper to investigate the performance of the approximate compressors for the aforementioned metrics for inexact multiplication. The approximate compressors have been utilized in the reduction module of a DADDA multiplier.

REFERENCES:

[1]J. Liang, J. Han, and F. Lombardi, “New metrics for the reliability of approximate and Probabilistic Adders” IEEE Trans Computers vol. 63, no. 9, pp. 1760–1771, Sep 2013.

[2]V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, “IMPACT: IMPrecise adders for low-power approximate computing,” in

Proc. Int. Symp. Low Power Electron Design, Aug. 2011, pp. 409–414.

[3] M. J. Schulte and E. E. Swartzlander Jr., "Truncated multiplication with correction constant" in Proc. Workshop VLSI Signal Process VI, 1993, pp. 388–396.

[4] C. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low- power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," IEEE Trans. Circuits Syst., vol. 51, no. 10, pp. 1985–1997, Oct. 2004.

[5] D. Radhakrishnan and A. P. Preethy, "Low-Power CMOS pass logic 4-2 compressor for high-speed multiplication," in Proc. IEEE 43rd Midwest Symp Circuits Syst., 2000, vol.3, pp 1296–1298.

[6] J. Gu and C. H. Chang, "Ultra low-voltage, low-power 4-2 compressor for high speed multiplications," in Proc. 36th IEEE Int. Symp. Circuits Syst., Bangkok, Thailand, May 2003, pp. v-321–v-324.

[7] M. Margala and N. G. Durdle, "Low-power low-voltage 4-2 compressors for VLSI Applications," in Proc. IEEE Alessandro Volta Memorial Workshop Low-Power Design, 1999, pp. 84–90.

[8] D. Baran, M. Aktan, and V. G. Oklobdzija, "Energy efficient implementation of parallel CMOS multipliers with improved compressors," in Proc. ACM/IEEE 16th Int. Symp. Low Power Electron. Design, 2010, pp. 147–152.

[9] J. Ma, K. Man, T. Krilavicius, S. Guan, and T. Jeong, "Implementation of high performance multipliers based on approximate compressor design," presented at the Int. Conf. Electrical and Control Technologies, Kaunas, Lithuania, 2011.