YIDDISH Vol 10 Issue 03, July 2020
ISSN NO: 0364-4308

DESIGN AND ANALYSIS OF DADDA MULTIPLIER USING APPROXIMATE COMPRESSORS

M. Pavitra [1], N. Praveena[2], M. Chenna Kumari[2], S.V.Sunayana[2], M. Tejaswani[2], M.Navven[2]

[1] *Associate Professor, ECE department, PBR Visvodaya Institute Of Technology & Science, Kavali, Andhra Pradesh, India.*
[2] *UG students, ECE department, PBR Visvodaya Institute Of Technology & Science, Kavali, Andhra Pradesh, India.*

ABSTRACT:

In exact, computing is an attractive paradigm for digital processing at nanometric and micrometric scales. Addition and multiplication are widely used operations in computer arithmetic, digital signal and image processing. This project deals with the analysis and design of two new approximate 4-2 compressors for utilization in a multiplier. The approximate design is to reduce the power, area and delay. These designs rely on different features of compressors, such that imprecision in computation (error rate). Two different schemes for utilizing the proposed approximate compressors are proposed and analyzed for Dadda multiplier. A widely used structure for compression is the 4-2 compressor and it can implement with carry bit between adjacent slices. This method is not accurate because it produces more incorrect results out of possible outputs where error rate is more. Two different designs are proposed next to reduce the error rate, these designs offer significant performance improvement. In approximate compressor, the carry value does not affect the output. So in the design of multipliers use approximate compressors to speed up the partial product reduction tree and reduce the delay.

Key words: Compressor, Dadda multiplier, Approximate circuits.

1. Introduction:

Multiplication is the basic arithmetic operation for several microprocessors and digital signal processing applications. Multipliers are utilized within the arithmetic logic units of microprocessors and to implement DSP algorithms such as convolution and filtering, multipliers are required in the digital signal processing systems. Multipliers lie directly within the critical path in most systems, due to the demand for high speed multiplication. For higher speeds, column compression

multipliers are most popular.

Wallace introduced the first column compression multiplier. L. Dadda modified the Wallace multiplier as Dadda multiplier in 1965 and it is slightly faster than the Wallace multiplier. Also, the hardware requirement is less. Both of these multipliers consist of three stages. In the first stage, the partial product matrix is formed. This partial product matrix is reduced by half in the second stage. In the final stage, by using a carry propagating adder, these two rows are combined. In the first stage, the multiplicand and the multiplier are multiplied bit by bit to generate partial product. To reduce the partial product to half and to generate all partial products in parallel. The most important stage is the second stage as it is complicated and determines the overall speed of the multiplier. Here in place of full adders, 3:2 compressors are used, because when compared to full adders 3:2 compressors have less delay.

For high speed design, Dadda multiplier is used to add the partial products in order to produce two rows of partial products that can be added in the last stage. The addition of the partial products can be done using 4:2 compressors. At nanometric scales, approximate computing is an attractive option for digital processing. It is interesting for computer arithmetic designs. Mainly this project is used for reducing the delay by using approximate compressors

2. Existing Methods:

2.1 Multiplication:

To reduce significant power consumption it is good to reduce the number of operation thereby reducing dynamic power which is a major part of total power consumption so the need of high speed a low power multiplier has increased. Designer mainly concentrates on high speed and low power efficient circuit design. The objective of a good multiplier is to provide a physically packed together, high speed and low power consumption unit. In this first we describe different types of multipliers: Booth multiplier, Sequential multiplier, combinational multiplier, Wallace tree multiplier.

- An efficient multiplier should have following characteristics,

  Accuracy: - A good multiplier should give correct result.

  Speed: - Multiplier should perform operation at high speed.

  Area: - A multiplier should occupy less number of slices and LUTs.

  Power:- Multiplier should consume less power.

- Multiplication process has three main steps

  1. Partial product generation.
  2. Partial product reduction.
  3. Final addition.

  For the multiplication of an n-bit multiplicand with an m-bit multiplier, m partial products are generated and product formed is n + m bits long.

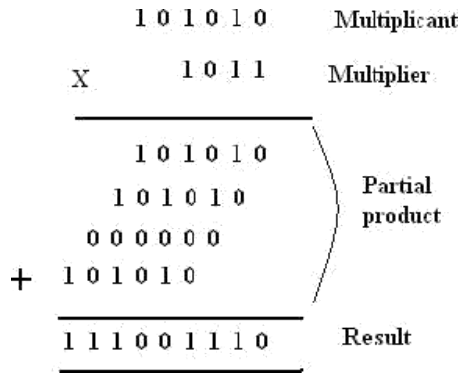

Figure 2.1: General multiplication

2.2 Different Types of Multiplications:-

1. Booth multiplication
2. Wallace tree multiplication
3. Array multiplication
4. Sequential multiplication
5. Serial multiplication
6. Parallel multiplication

3. Proposed methods:

3.1 Compressors:

Compressor is a single bit adder circuit that has more than three inputs. A compressor is circuit which is used to reduce the operands while adding terms of partial products in multipliers. Compressors are the key building blocks used for reducing the partial production stages during multiplication processes.

In the multiplication process, compressors are normally used to serve two major purposes:

1. To speed up the Operation.
2. To reduce the number of stages in the partial product generation

Compressor types:

- 4:2 Compressor
- 3:2 Compressor

Basically 4:2 compressors came into existence to replace the usage of more number of full adders. The general block diagram of 4:2 compressor is shown in Fig.1. It consists of X1, X2, X3, and X4 as 4 inputs with a carry in (Cin) and sum, carry as outputs with a carry out (C_{out}) for propagation.

3.1.1 4:2 Compressor:

An exact 4-2 compressor has five inputs and three outputs as shown in Figure 1.1. It produces a sum for the same order of the next stage and a carry for one order higher in the next stage. Also, a carry out (C_{out}) becomes the carry in (C_{in}) for the next higher order compressor.

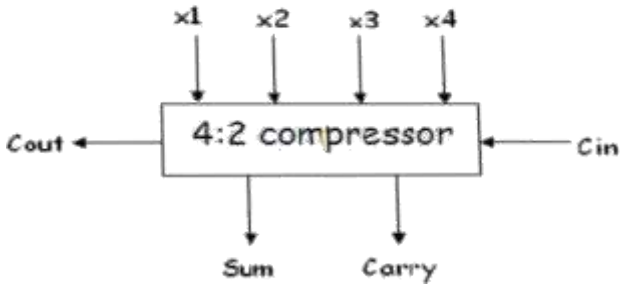

Figure3.1: Logic symbol of 4:2 Compressor:

The carry out (Cout) is a one bit binary number with higher significance whereas the four inputs and sum output carry the same weight. Since the carry in (Cin) bit acts as one of the inputs to the compressor, it will have lower significance while the output Carry out (Cout) comes with higher significance.

3.1 3:2 Compressors:

The full adders are replaced by 3:2 compressors for high speed and low power consumption. It consists of XOR gates and Multiplexer so as to minimize the

number of logic gates, power and delay. The functionality remains the same as the full adder but structurally different.

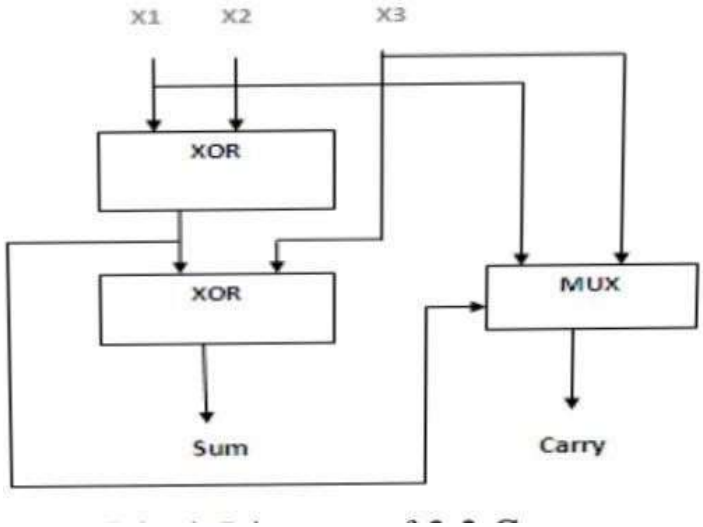

Figure 3.6: Logic diagram of 3:2 compressors

3.2: Truth table for 3:2 Compressor

X_1	X_2	X_3	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$Sum = (X_1 \oplus X_2) \bullet \overline{X_3} + (\overline{X_1 \oplus X_2})\bullet X_3 \quad (3.7)$$

$$Carry = (X_1 \oplus X_2) \bullet X_3 + (\underline{\quad})\bullet X_1 \quad (3.8)$$

The first two stages of multiplier use 3:2 compressors for effective performance. The multiplexer selects the input that has to be delivered as final carry. The sum is calculated using 2 XOR gates similar to a full adder.

The 3:2 compressors has 3 inputs and 2 out puts as like full adder but the structure is different. By using 3:2 compressors in place of full adders the delay is reduced.

3.4 Multiplier:

Multiplier is an electronic circuit which is used to multiply two binary numbers to generate the final product. Multipliers play an important role in today's digital signal processing and various other applications.

Generally, in the design of multipliers full adders are used to reduce the partial products but it requires more gates and the delay is also high. In the design of Dadda multipliers, full adders are replaced by the approximate compressors for high speed

An effective multiplier should have the following characteristics:

Accuracy*:* a good multiplier should give correct results.

Speed: multiplier should perform operation at high speed.

Area: multiplier should occupy less number of splices and LUT's.

Basic multipliers have 3 main steps.

They are

Product generation.

Partial product reduction.

Final addition

3.4.1 Dadda Multiplier:

The dadda multiplier is a hard ware multiplier design invented by a computer scientist Luigi Dadda in 1965. It is similar to the Wallace multiplier, but it is slightly faster (for all operand sizes) and requires fewer gates.

A modified architecture of Dadda Multiplier is presented for the effective utilizationof the proposed compressor and to reduce the error at the output. Through

extensive experimental evaluation, the efficiency of the proposed compressor..Multiplication is an elementary arithmetic operation and crucial in applications like digital signal processing.

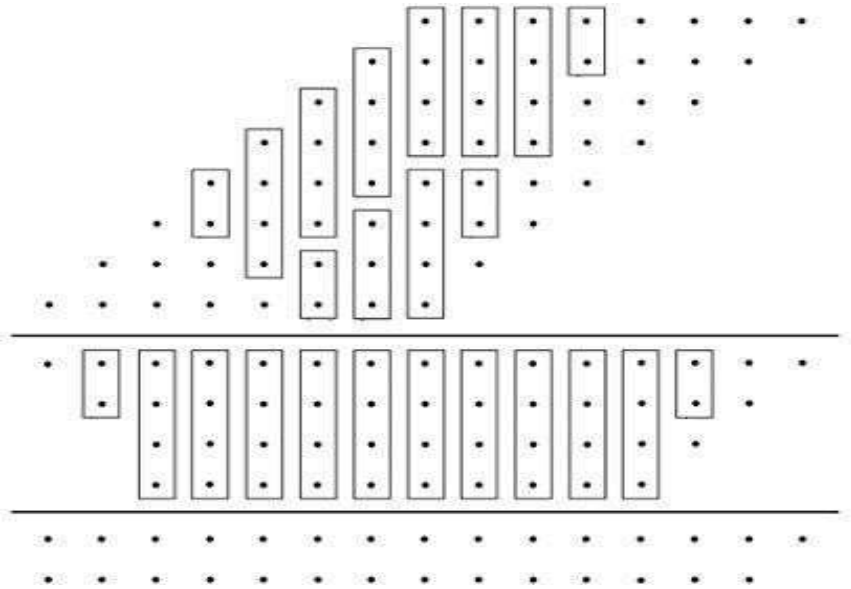

Figure3.7: General structure of Dadda multiplier

A modified architecture of Dadda Multiplier is presented for the effective utilization of the proposed compressor and to reduce the error at the output. Through extensive experimental evaluation, the efficiency of the proposed compressor..Multiplication is an elementary arithmetic operation and crucial in applications like digital signal processing.

Dadda and Wallace multipliers have the same three steps for two bit strings and of lengths and respectively:

1. Multiply each bit of , by each bit of , yielding results, grouped by weight in columns
2. Reduce the number of partial products by stages of full and half adders until we are left with at most two bits of each weight.
3. Add the final result with a conventional adder.

As with the Wallace multiplier, the multiplication products of the first step carry different weights reflecting the magnitude of the original bit values in the multiplication. For example, the product of bits has weight. Unlike Wallace

multipliers that reduce as much as possible on each layer, Dadda multipliers attempt to minimize the number of gates used, as well as input/output delay. Because of this, Dadda multipliers have a less expensive reduction phase, but the final numbers may be a few bits longer, thus requiring slightly bigger adders.

In this section, the implementation of proposed compressors on multiplier is analyzed. An exact multiplier with high speed is composed of three modules.

I. Partial Product Generation.

II. A Carry Save Adder (CSA) tree for the reduction of partial products.

III. A Carry Propagation Adder(CPA) for final computation of result.

In a multiplier design, the second module plays a major role with respect to delay, power consumption and circuit complexity. Compressors are usually used to increase the speed of the CSA tree and decrease its power dissipation in order to achieve low-power operation. These approximate compressors lead to the design of approximate multipliers.

An unsigned 8*8 Dadda tree multiplier is chosen for the usage of proposed compressors in multipliers. The reduction circuit of an exact multiplier where two half-adders, two 3:2 compressors (instead of full adders) and eight 4:2 compressors are utilized in the first stage to reduce the partial products to at most four rows. In the very next stage, one half-adder, one 3:2 compressor and ten compressors are used to calculate the 2 final rows of partial products. Hence, 2 stages of reduction and 3 half adders, three 3:2 compressors and 18 compressors are needed in this reduction of 8x8 Dadda tree multiplier. In this work, we considered four cases to design an approximate multiplier.

Dadda multiplier using design 1 approximate compressor:

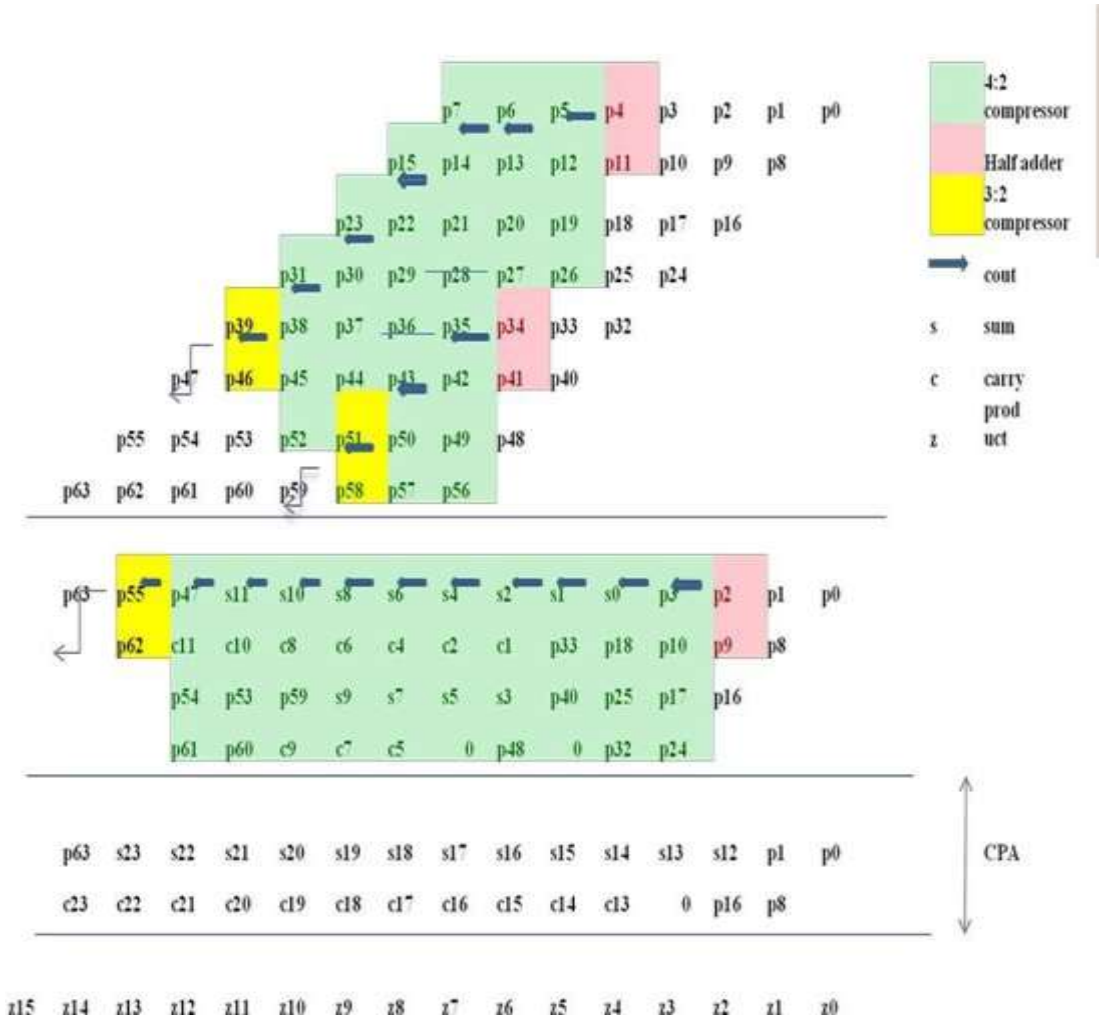

Figure 3.8 : Structure of Dadda multiplier using approximate compressor

•A 8×8 unsigned Dadda tree multiplier is considered to access the imp-act 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

Theoretical Calculations:-

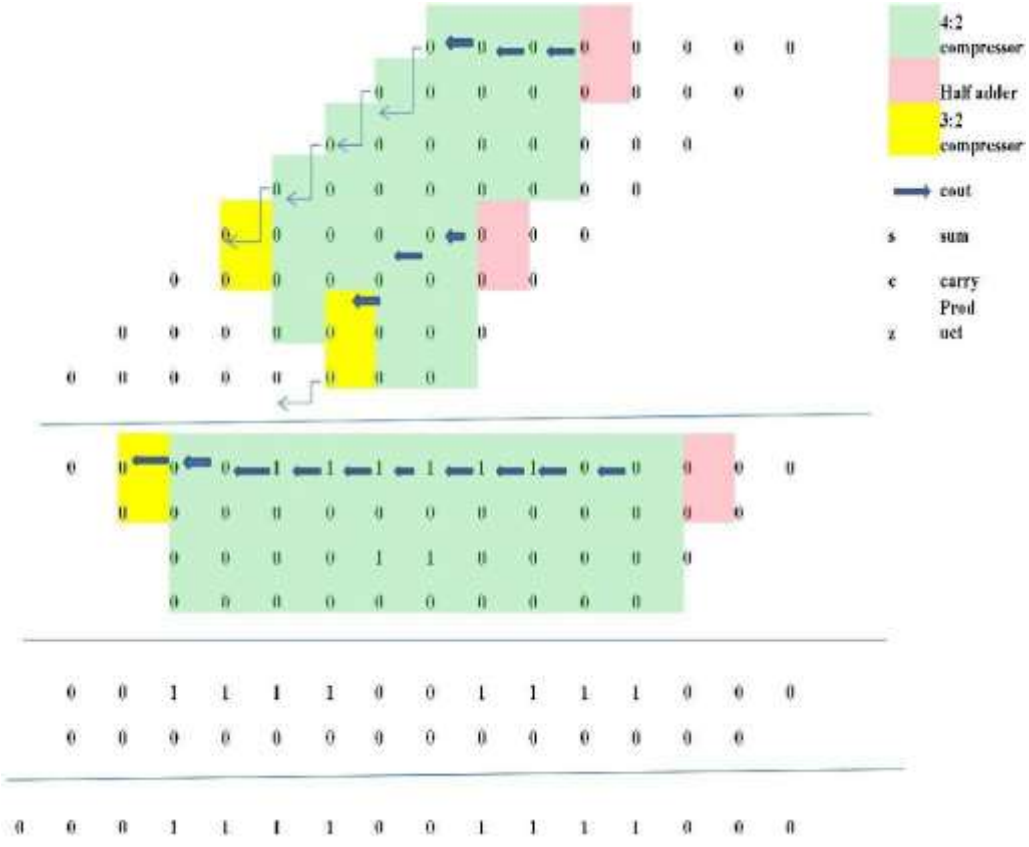

Figure 3.9 : Theoretical calculation of Dadda multiplier using design 1 approximate compressor

Limitation:

This multiplier produces more delay. Hence , design 2 approximate compressor is used in the design of Dadda multiplier.

Dadda multiplier using design 2 approximate compressor:

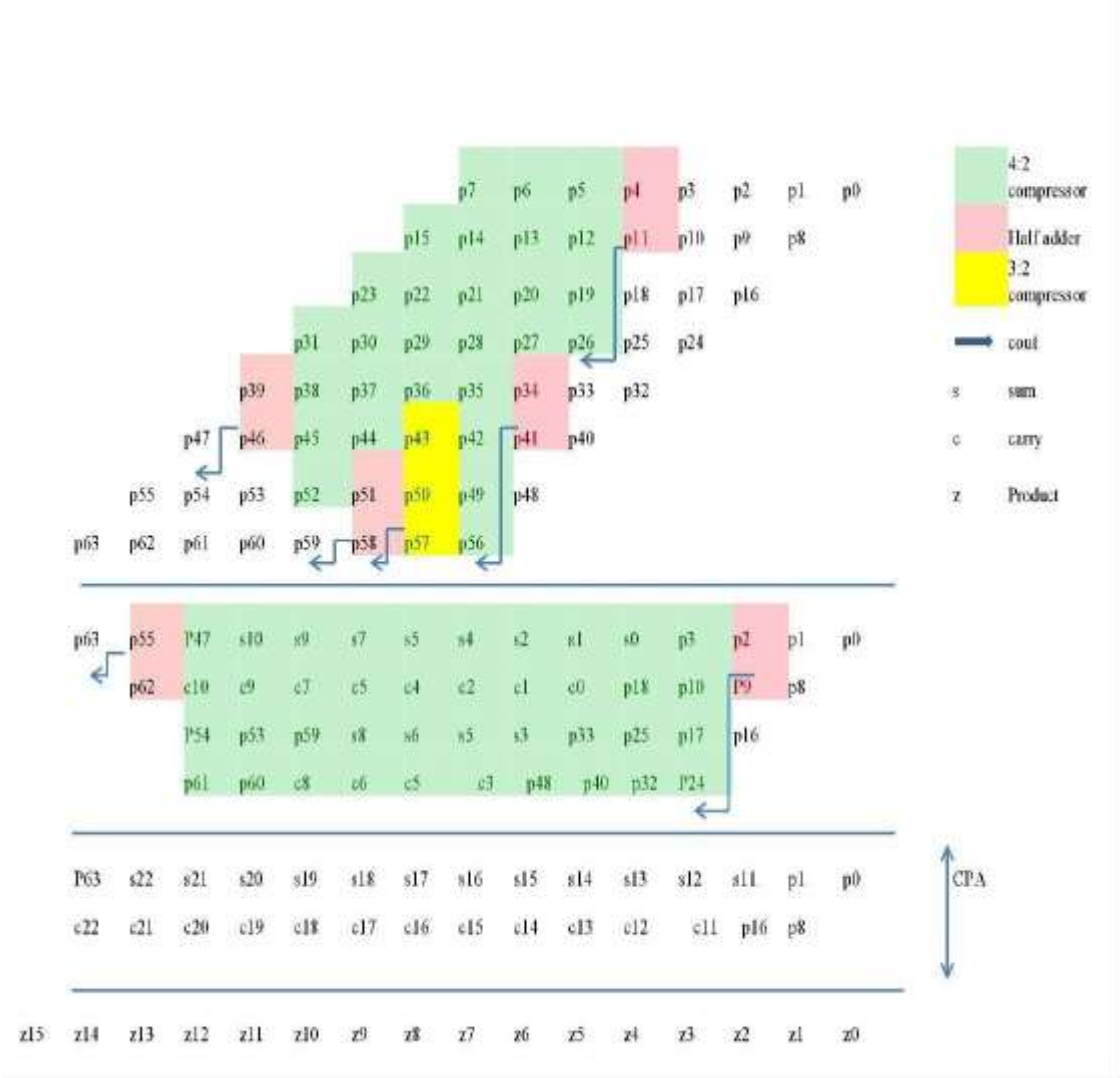

Figure 3.10: Structure of dadda multiplier using approximate compressor

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

In the third case (Multiplier 3), Design1 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.

Theoretical Calculations:

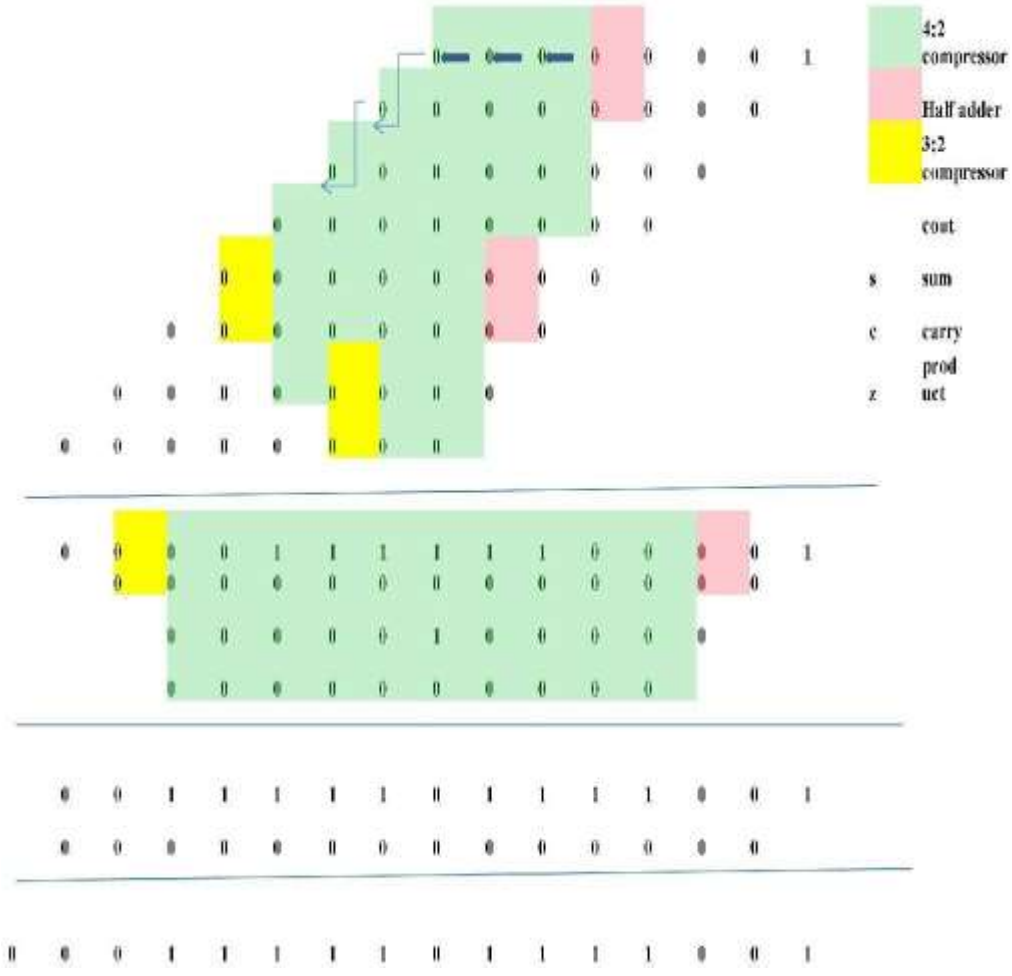

Figure 3.11: Theoretical calculation of dadda multiplier using design 2 approximate compressor

4 Simulation Results:

4.1 Simulation result of AND Gate:

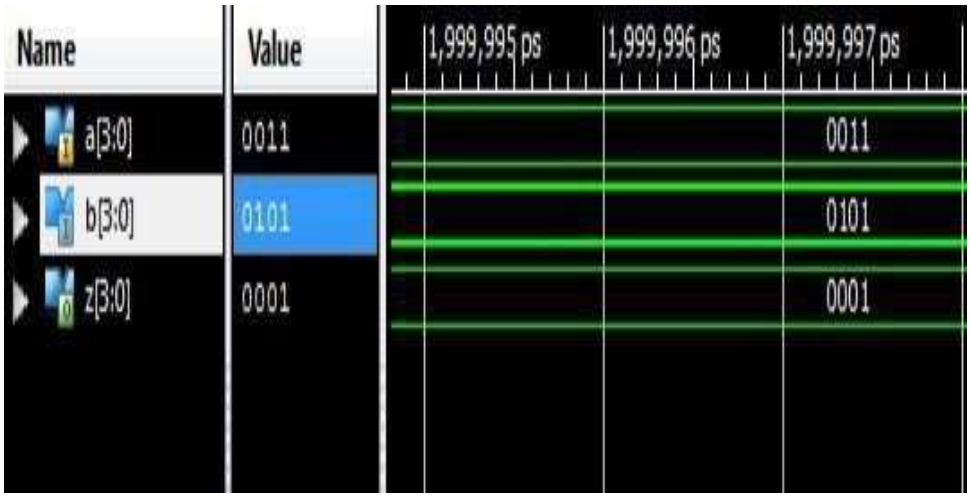

Figure 4.1:-Simulation result of AND gate

For AND gate, if one of inputs is logic 0 then the output is also logic 0 otherwise the output is logic 1 or the output is logic 1, when both the inputs are logic1. Otherwise logic 0.

4.2 Simulation result of OR Gate:

For OR gate, if the one of the inputs is logic 1 then the output is logic 1otherwise the output is logic zero or the output is logic zero when the both inputs are logic 0 otherwise output is logic 1.

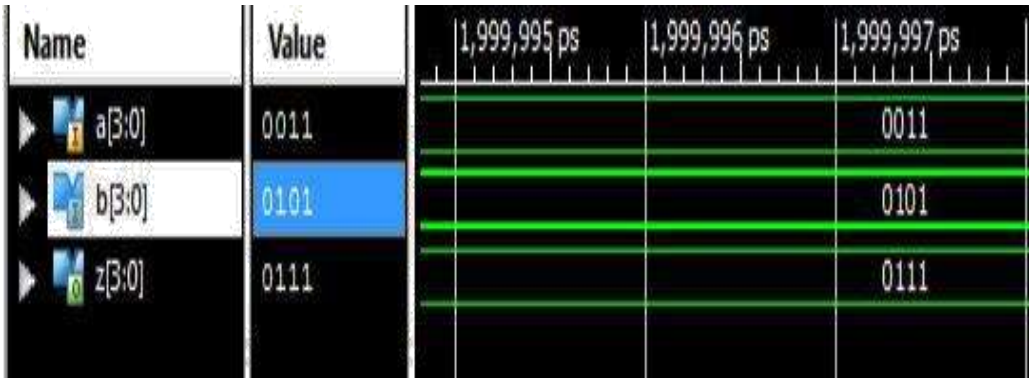

Figure 4.2:-Simulation result of OR gate

4.3 Simulation result of Xor Gate:

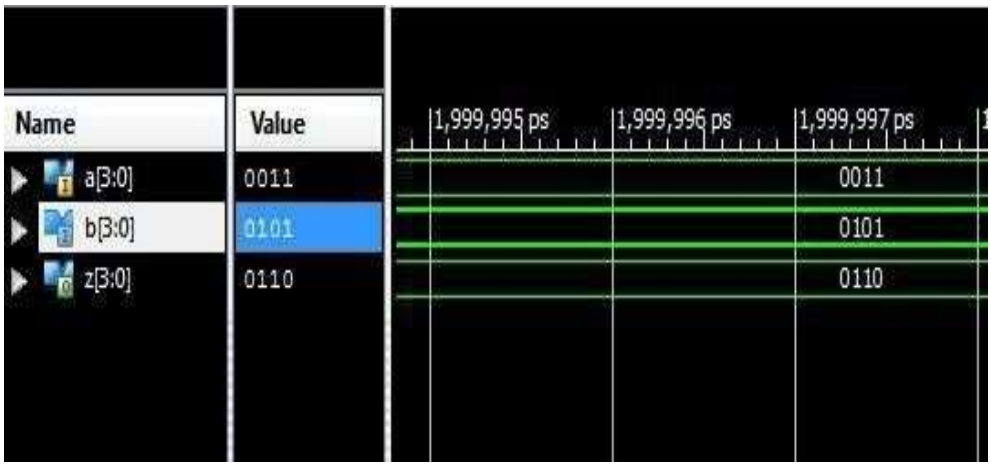

Figure 4.3:-Simulation result of Xor gate

For XOR gate, the output is logic 1 when the both inputs are different and the output is logic 0 when the both inputs are same

4.4 Simulation result of Half Adder:

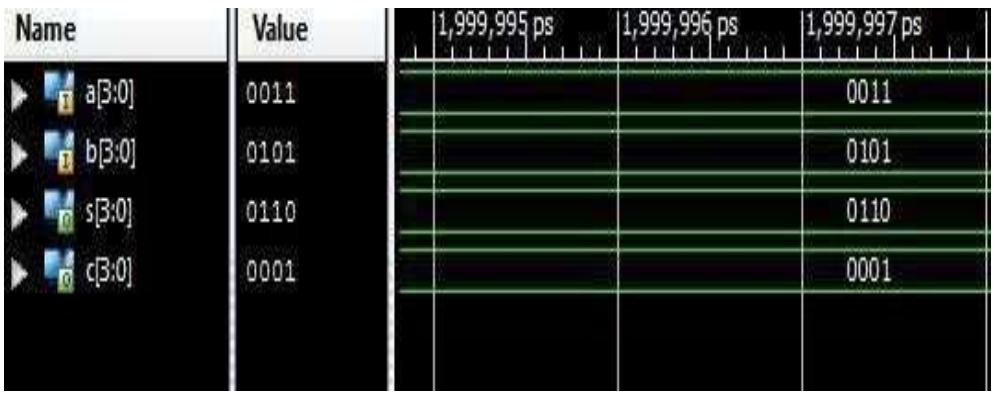

Figure 4.4:-Simulation result of Half Adder

For half adder, if the two inputs are logic 0 then both sum and carry also logic 0 and if one the inputs is logic 1 then sum is logic1and carry is logic 0. Otherwise both sum and carry are logic 1.

4.5 Simulation result of Full Adder:

The full adder circuit can be implemented with the help of two half adder circuits. At first, half adder will be used to add a and b inputs to produce a partial sum and a second half adder logic can be used to add c_{in} in to the sum produced by first half adder to get the finals output.

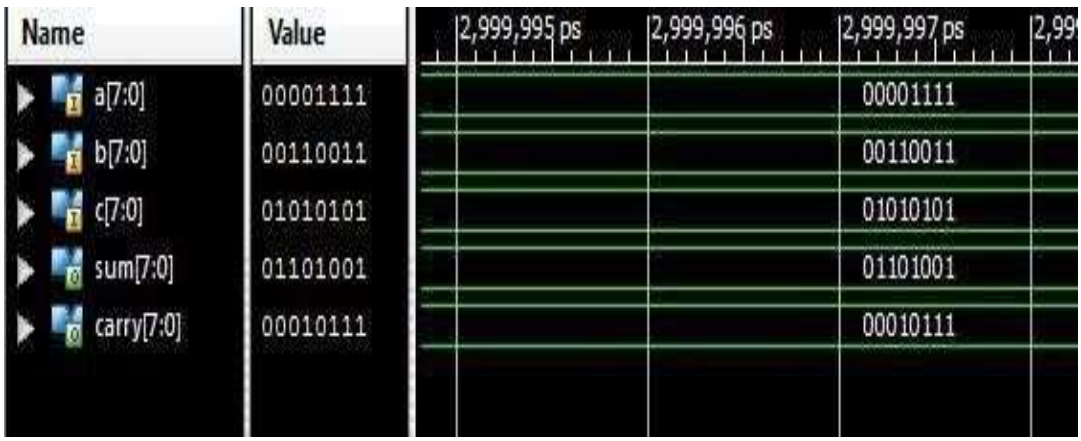

Figure 4.5:-Simulation result of Full Adder

4.6 Simulation results of Multiplexer:

Figure 4.6:-Simulation result of Multiplexer

Figure 4.7:-Simulation result of multiplexer

Depending on the selector switching the inputs are produced at outputs , i.e., I0 , I1 and are switched to the output for S=0 and S=1 respectively . Thus, the Boolean expression for the output becomes I0 when S=0 and output is I1 when S=1.

4.7 Simulation result of Design-1 Approximate Compressor:

Figure 4.8:-Simulation result of Design-1 Approximate Compressor

4.8 Simulation result of Design-2 Approximate Compressor:

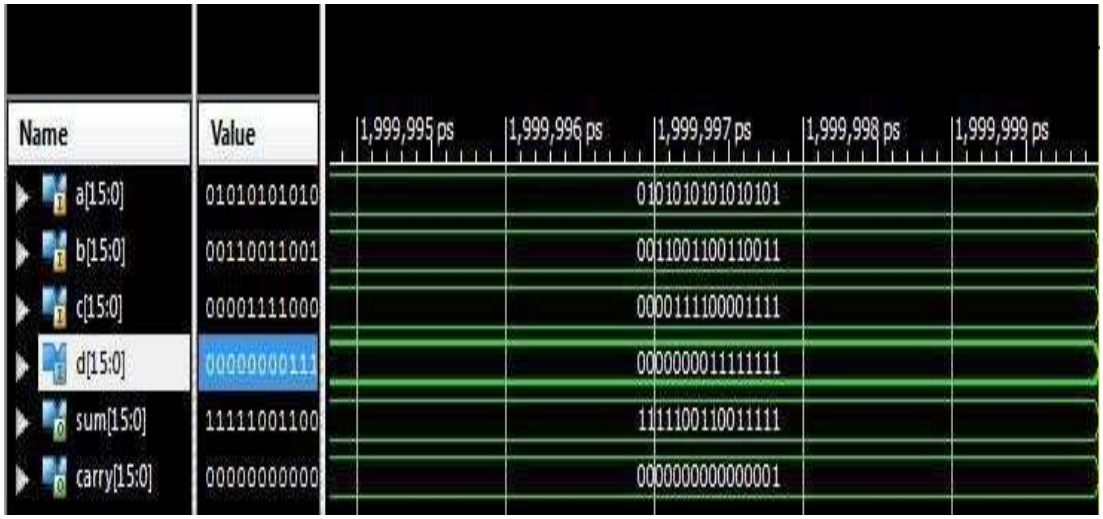

Figure 4.9:-Simulation result of Design-2 Approximate Compressor

4.9 Simulation result of 3:2 Approximate Compressors:

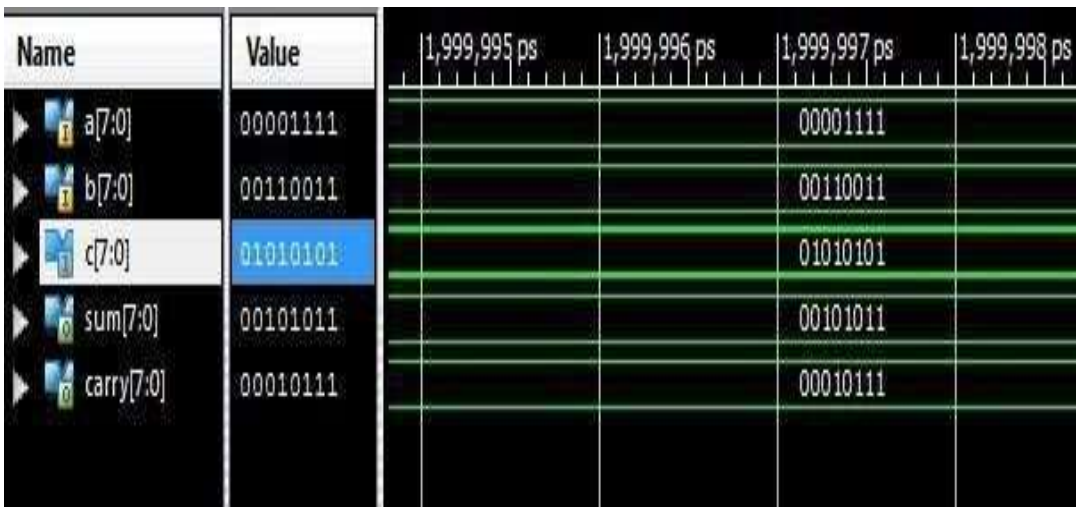

Figure 4.10:-Simulation result of 3:2 Approximate Compressor

4.10 Simulation result of Dadda Multiplier using Design -1 Approximate Compressor:

Figure 4.11:-Simulation result of Dadda Multiplier using Design-1 Approximate Compressor

4.11 Simulation result of Dadda Multiplier using Design-2 Approximate Compressor:

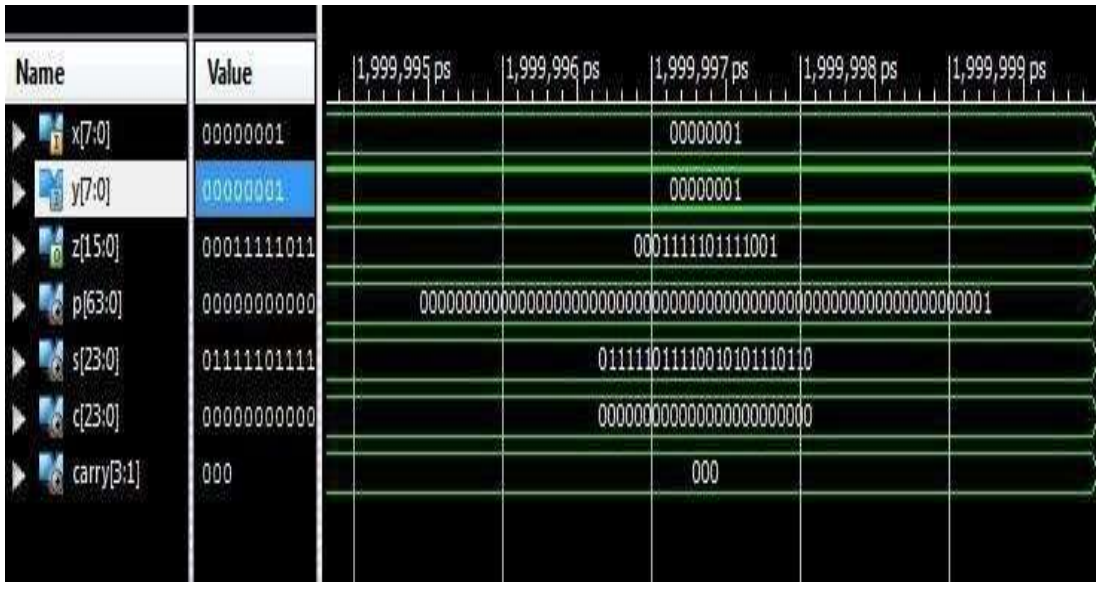

Figure 4.12:-Simulation result of Dadda Multipplier using Design-2 Approximate Compressor

5. Conclusion:

This paper has shown that by an appropriate design of approximate compressors, multipliers can be designed for inexact computing; these multipliers offer significant advantages in terms of both circuit-level and delay. Although not discussed and beyond the scope of this manuscript, the proposed designs may also be useful in other arithmetic circuits for applications in which in exact computing can be used.

Parameter	Dadda Multiplier using Design 1approximate Compressor	Dadda multiplierusing design 2 approximate compressor
Delay	27.663 ns	23.141 ns
LUT's	128 out of 1536	123 out of 1920
Slices	74 out of 768	69 out of 960

References:

1. Ahmadi. A,Fakhraie .S.M, Lucas .C and Mahdiani. H.R (2010), "Bioinspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," IEEE Trans. Circuits Syst. I: Reg. Papers, vol. 57, no. 4, pp. 850–862.

2. Akgul .B.E.S , Chakrapani .L.N , Cheemalavagu .S , Korkmaz .P , Palem .P.K.V (2005) , "A probabilistic CMOS switch and its realization by exploiting noise," presented at the IFIP Int. Conf. Very Large Scale Integ., Perth, Australia.

3. Al-Sarawi .S, Kelly .D and Phillips .B (2009), "Approximate signed binaryinteger multipliers for arithmetic data value speculation," in Proc.Conf. Design Architect. Signal Image Process, pp. 97–104.

4. Jullien .G.A, Miller .W.C and Wang .Z (1995), "A new design technique forcolumn compression multipliers," IEEE Trans. Comput.,vol. 44, no. 8, pp. 962–970.