

# Design and Simulation of 8 Bit x 8 Bit Approximate Dadda Multiplier

1.Pothuraju Sindhu Bharathi Student of M.Tech VLSI, Department of ECE, Research Center for VLSI and Embedded Systems, Sree Vidyanikethan Engineering college Tirupati, India

2.Kurakula Ramesh Assistant professor, Department of ECE, Research Center for VLSI and Embedded Systems, Sree Vidyanikethan Engineering college Tirupati, India

3.Neelima K Assistant professor, Department of ECE, Research Center for VLSI and Embedded Systems, Sree Vidyanikethan Engineering college Tirupati, India

**Abstract**— Error Tolerance formally captures applications such as audio, video, graphics and wireless communications. Erroneous values are produced as outputs by a defective chip [called as approximate value]. To reduce these errors, mostly filters come into existence. Filters are a part of the analog and digital technologies which are used to suppress or attenuate the data or the range of frequencies of a signal. Predominantly multiplier plays a vital role in the digital applications. In digital signal processing, multipliers are key arithmetic circuits. In this paper, instead of traditional multipliers, an approximate multiplier is used because it consists of low power consumption and short critical path. It is also used in DSP applications due to high performance. There are certain limitations for the design of multipliers mainly area, delay, power consumption, power dissipation. Even though there are changes in the area, they can be accepted. Approximate multiplier has most of the errors in their magnitude. The simulation of the designs is performed in Xilinx ISE 14.5 Tool and their functionality is verified by using ISIM Simulator.

**Keywords**—Approximate computing, Error Tolerant, Dadda Multiplier

## 1. INTRODUCTION

A high degree of realization precision is required which is used in digital logic circuits. The major requirement of conventional arithmetic circuits is functional accuracy and which can be implemented “inexact” or “approximate”. In multimedia and image processing applications, errors and imprecision even though they provide good results. For the computation of digital circuits, there is a need to have a decrease in cost and complexity with a possible increase in performance and power efficiency [1]. Approximate computing depends on the usage of properties to design but approximate circuits operate at higher and lower power consumption.

By this significance, there are a wide range of applications such as multimedia, data recognition and data mining also considered for error resilient applications mainly hearing or vision. Multiplication and Addition are used in computer arithmetic for adding the full adders. In this paper, approximate computing can be done by reducing the area of the multiplier of full adders and the Dadda tree multiplier and changes in the power consumption, shortest critical path as well as reduced circuit complexity. And also there are changes in the loss of accuracy which can achieve better performance in energy efficiency.

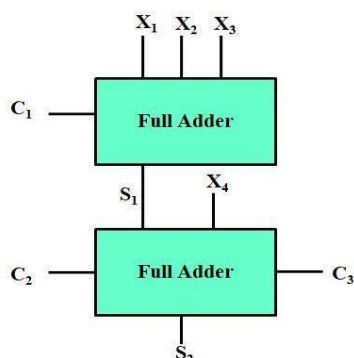
## 2. DESIGN ASPECTS

Binary multipliers comprise of three stages:  
AND gates are used to generate Partial Products.

- These Products are reduced by using an adder tree.

- CARRY PROPAGATION ADDER (CPA) USED FOR ADDING THE RESULTS.

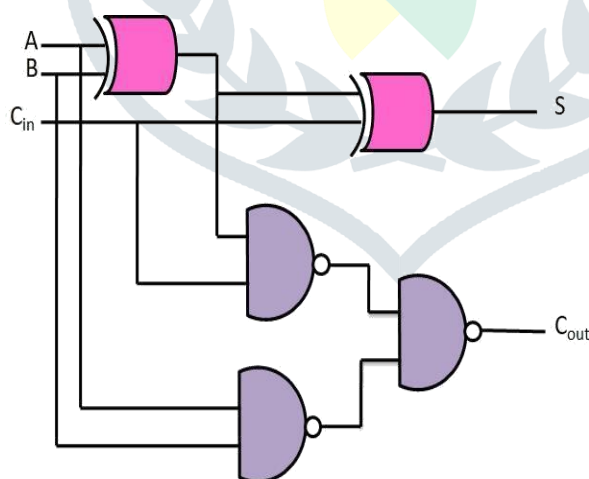
• The parameters like delay, power consumption, circuit complexity of a multiplier are play a dominant role for the generation of partial product. To achieve power reducing and the delay estimation, compressors are used. Circuit complexity is degraded by reducing the power dissipation are obtained by using approximate multiplier instead of exact compressors [2]. Multipliers can be formed by the set of full and half adders. In this paper, the basic building block at which the change of full adders with providing sum and carry at the output end.



**Figure 1.** Block Diagram of the Multiplier

From Figure 1, the input bits namely X1, X2, X3 are passed through a full adder. Full adder consists of a half

adders providing the sum and carry at the output level. That sum at the output level acts an input to the next full adder providing the sum and carry vice versa. In the Figure 2 and XXX-X-XXXX-XXXX-X/XX/\$XX.00 ©20XX IEEE Figure 3, describes about an exact algorithm of two different Full Adders [3] [4]. By considering above block diagram of multiplier,



**Figure2.** Design of Full Adder 1 [3]

$$\text{Sum} = (X_1 \oplus X_2) \oplus (X_3 \oplus X_4) + X_1 + X_2 + X_3 + X_4 \quad (1)$$

From equation (1) the first term is obtained as

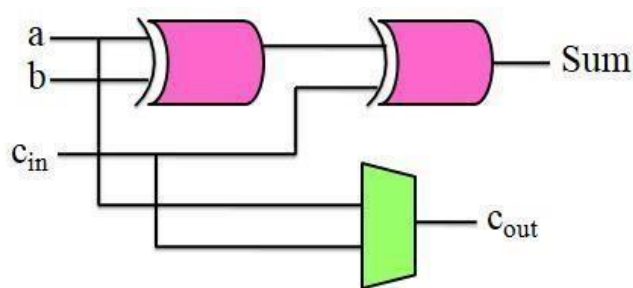
$$(X_1 \oplus X_2) \oplus (X_3 \oplus X_4) \quad (2)$$

$$\text{Sum} = (X_1 \oplus X_2) \oplus (X_3 \oplus X_4) + X_3 X_4 \quad (3)$$

and the next term  $X_3 X_4$  at the end then the output is

**Table 1.** Truth Table of ACCI1

Cs	X1 X2 00	01	11	10
X3 X4 00	00	01	10	01
01	01	10	11	10
11	10	11	11	11
10	01	10	11	10

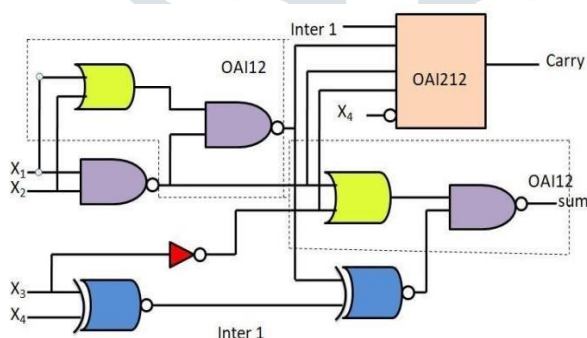


**Figure 3.** Design of Full Adder 2

### 3.COMPRESSOR

Compressors are designed to restrict the low probability of error occurrence. The approximation and truncation are employed for the partial product reduction at which only approximate compressors are used at high accuracy.

Compressors are used to decrease the power dissipation and to speed up the carry save adder tree to achieve fast and low power operation [5]. In this paper, there are three compressors namely Approximate Compressors with  $C_{in}$  and  $C_{out}$  are ignored [ACCI1, ACCI2, ACCI3]. In the Figure 4, ACCI1 is an accurate compressor. If all the inputs are '1', then the output is '100' at which the output can be formed as to 11 with an ER of 1/256. To calculate the accuracy, carry is most important than the Sum. When all the inputs are 1's then the Carry is '1' in the proposed designs [6]. The need of ER is to constraint the metric for the approximatedesigning. From the Table 1, Truth table of ACCI1, the modified results are used to minimize the complexity the sum is changed as '1' to '0011' input. Similarly, ACCI2 is used for sum generation and the complexity is reduced by removing the last term.

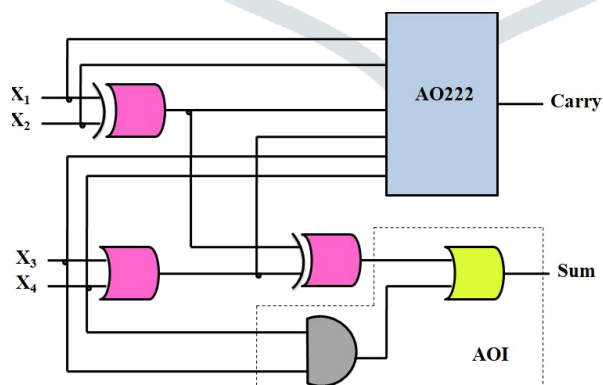


**Figure 4.** ACCI1 Architecture

**Table 2.** Truth Table of ACCI2

<b>Cs</b>	<b>X1 X2 00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>X3 X4 00</b>	00	01	10	01
<b>01</b>	01	10	11	10
<b>11</b>	<b>11</b>	<b>11</b>	<b>11</b>	11
<b>10</b>	01	10	11	10

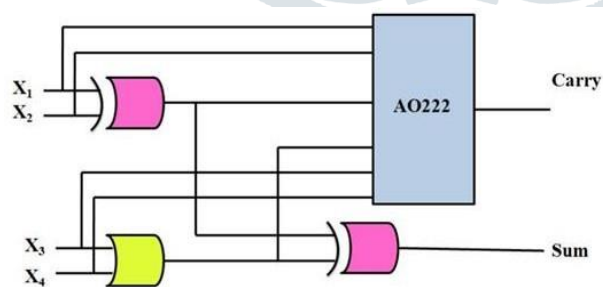
Similarly, in the Figure 5, ACCI2 is further used to generate the sum, by removing the last term and the operation is done with the help of Table 2. Hence the complexity of the logic is reduced [5][7].

**Figure 5.** ACCI2 Architecture

In the Figure 6, indicates the ACCI3 with having ER of 1/256, 10/256, 1/16 and 25/64 so that to acquire better accuracy and its truth table is shown in Table 3 [5]. Since the ACCIs have better accuracy and the partial product of low columns are truncation which can be used as design multiplier with more significant bits.

**Table 3.** Truth Table of ACCI3

Cs	X1 X2 00	01	11	10
X3 X4 00	00	01	10	01
01	01	10	11	10
11	11	10	11	10
10	01	10	11	10

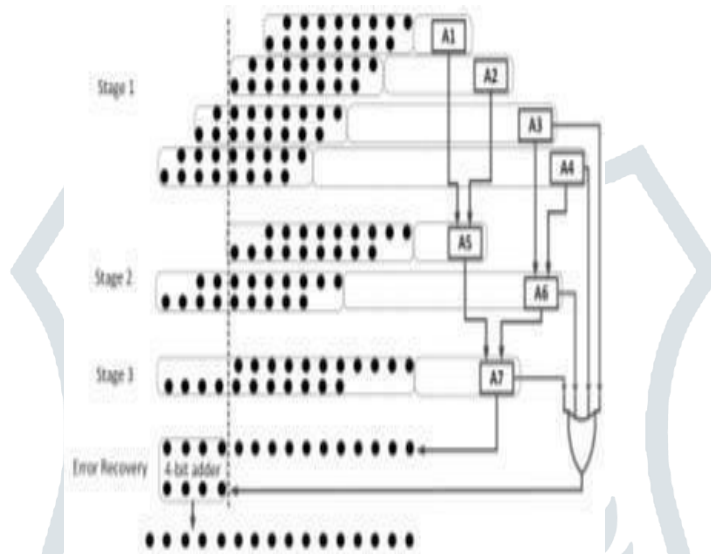


**Figure 6.** ACCI3 Architecture

## 4.8 BY 8 DADDA MULTIPLIER

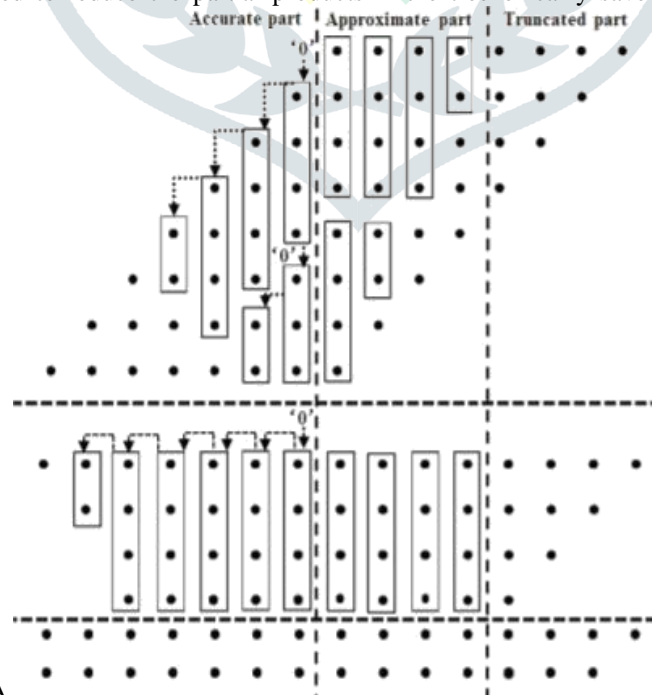
Figure 7,[3] shows approximate multiplier. It is the simple multiplier using adders in the partial product accumulation. The error is reduced, by using two techniques like error accumulation and error recovery. Error Accumulation: Accurate adders are used to sum the error signals. The accumulated error can fully compensate the inaccurate product and complexity is reduced

Error Recovery: Using a conventional adder, an error vector is added to adder tree output to reduce the error. To reduce the overall complexity, several MSBs of the error signals are used to compensate the output.



**Figure 7.** Design of an Approximate Multiplier with OR gate based partial error recoveries using four MSDs of the Error Vector [3]

Dadda Algorithm is one of the best multiplier when compared with other multipliers. From the Figure 8,[6] shows an approximate compressor for an 8x8 bit Dadda multiplier [8] [9]. Here the AND gates are used to generate the partial products and then the approximate compressors are used to reduce the partial products in the tree of carry save adder. Finally, the binary results are



computed by using an exact CPA.

**Figure 8.** Partial Product Reduction Design using Truncation for an 8 by 8 Dadda Multiplier[6]

In Figure 8, Partial Product Reduction Design using Truncation for an 8 by 8 Multiplier [6]. In this the number of half-adders, full-adders and compressors are reduced into 4 rows and then into 2 rows. Dadda multiplier minimizes the number of HAs and FAs. Reduction considers each column separately. Reduces the number of dots in each column to the maximum number of layers in the next stage. This multiplier reduces the partial product reduction by shifting the carry to the next adder [10] [11].

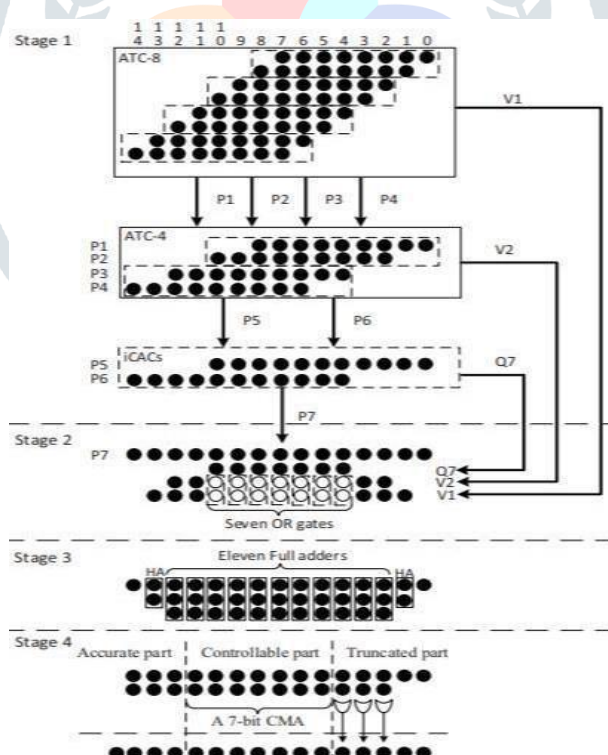
### 5.8 BY 8 PARTIAL PRODUCTS MULTIPLIER

In this Figure 9, [12] an 8-bit multiplier with 8x8 partial products (PPs), it has four stages. The first three stages perform partial product reduction (PPR) and stage 4 perform carry propagation adder (CPA) [13].

Stage 1: Eight rows of PPs are reduced to form P1, P2, P3 and P4 rows and it has accuracy compensation vector (V1), these rows are reduced to two rows (P5 and P6) and accuracy compensation vector (V2). It performs the partial products eight rows are finally reduced to 4 rows P7, Q7, V1 and V2.

Stage 2: Here four partial products of bits 4 to 10 will perform OR gates to sum V1 and V2. OR gate are used to achieve delay. Using OR gates, four rows reduced to three rows. Stage 3: In this stage the first two row bits 1 and 13 perform half adder and the three row bits 2 to 12 perform full adder.

Stage 4: In this stage carry propagation adder (CPA) is performed. It has 3 parts, first part is truncated, in this two, three and four bits are generated by 3 OR gates. The second part is the controllable part, in this 7 bit carry maskable adder (CMA) is placed in

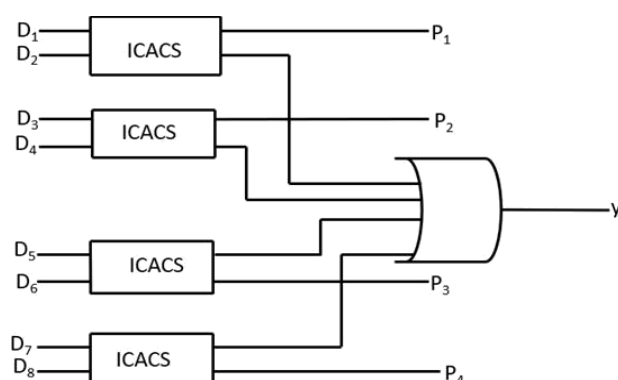


from five to eleven bit. The third part is the accurate part bits 12 to 14 are the most significant for accuracy [14].

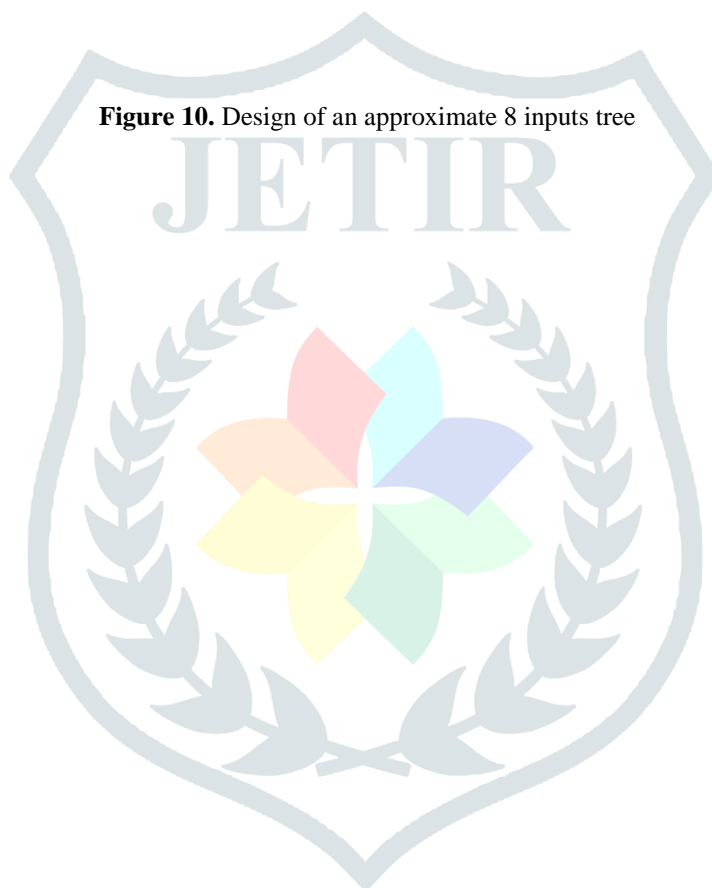
**Figure 9.** Structure of 8x8 Partial Product [12]



Figure 10, is a design of an approximate 8 inputs tree compressor [12]. This consists of eight inputs from D1 to D8. It consist of four Incomplete Adder Cells [ICAC], four output as P1, P2, P3, P4 as sum, and carries from ICACS undergo OR operation and produces a common output as Y [15].



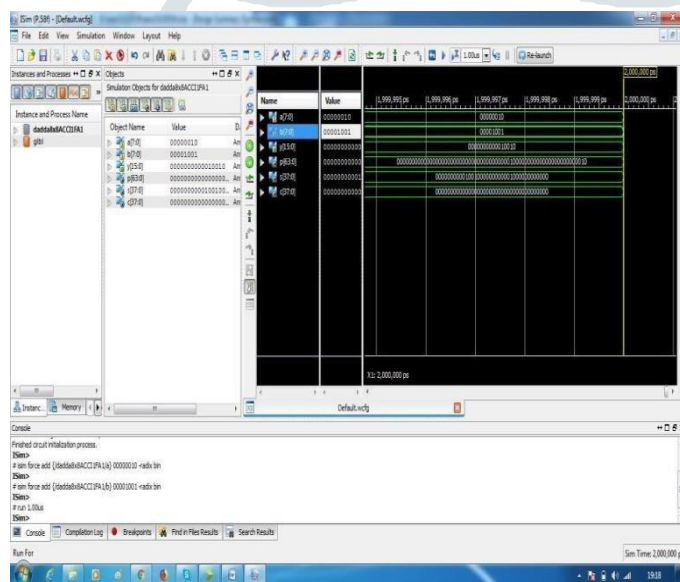
**Figure 10.** Design of an approximate 8 inputs tree



## 6.RESULTS

The simulation of the designs is performed in Xilinx ISE Tool and their functionality is verified by using ISIM Simulator. The simulated waveform is shown in figure below. Where  $a=8'b00000010$ ,  $b=8'b00001001$  when multiplied the result obtained was

$$y=16'b00000000000010010.$$



**Figure 11.** Simulated Waveform

From Table 4, Dadda algorithm is compared with different ACCIs, full adders and also 8x8 partial product multiplier. When compared with Dadda 8x8 ACCI1 FA1, the Dadda 8x8 ACCI1 FA2 reduces the delay by 8.04%. When compared with Dadda 8x8 ACCI2 FA1, the Dadda 8x8 ACCI1 FA2 reduces the delay by 8.052%. When compared with Dadda 8x8 ACCI2 FA2, the Dadda 8x8 ACCI1 FA2 reduces the delay by 10.19%. When compared with Dadda 8x8 ACCI3 FA1, the Dadda 8x8 ACCI1 FA2 reduces the delay by 8.6%. When compared with Dadda 8x8 ACCI3 FA2, the Dadda 8x8 ACCI1 FA2 reduces the delay by 25.5%. When compared with 8x8 Partial Product, the Dadda 8x8 ACCI1 FA2 reduces the delay by 40.36%. Dadda 8x8 ACCI1 FA2 is the best multiplier by comparing delay and area parameters with different multipliers.

**Table 4.** Comparison Table for Different Multipliers

Multipliers	Delay	LUT	Slices	IOBS
Dadda 8x8 ACCI1 FA1	20.856ns	125	72	32
Dadda 8x8 ACCI1 FA2	19.18ns	123	70	32
Dadda 8x8 ACCI2 FA1	20.967ns	145	83	32
Dadda 8x8 ACCI2 FA2	21.358ns	144	82	32

Dadda 8x8 ACCI3 FA1	21.00ns	141	81	32
Dadda 8x8 ACCI3 FA2	14.287ns	137	78	32
Partial product 8x8	32.165ns	198	101	32

## 7.CONCLUSION

For Applications like audio, video, graphics and wireless communications, error due to approximation is allowed with a limited tolerance. For error reduction, filters can be used to suppress or attenuate the data or the range of frequencies of a signal. Predominantly multiplier plays a vital role in the digital applications especially in DSP related applications like FIR, IIR Filters, MAC Unit, etc. In this paper, an approximate multiplier with a low power consumption and short critical path than the traditional multipliers is proposed.

The key design aspect focus on the parameters like area, delay, power consumption, power dissipation. The modeling of the 8x8 Dadda Multiplier design is performed in Xilinx ISE 14.5 Tool using Verilog HDL and their functionality is verified by using ISIM Simulator. Among the designs developed Dadda 8x8 ACCI1 FA2 reduces the delay by nearly Minimum of 8% and Maximum of 40% when compared to other designs and proves to be the fastest multiplier.

## 8.REFERENCES

- [1] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Trans. Image Processing, vol. 13, no.4, pp.600-612, April, 2014.
- [2] NamanS Kumar, Shreyas Hande V, Sudhanva N. G., Shravan S.D., Kariyappa B. s., "Design of Area-Efficient Multiplier", International Conference on Recent Advances in Electronics and Communication, IEEE, October 2017.
- [3] Cong Liu, Jie Han, Fabrizio Lombardi, "A Low-Power, High Performance Approximate Multiplier with Configurable Partial Error Recovery", EDAA, 2014.
- [4] J. Huang, J. Lach, and G.Robins, "A methodology for energy-quality tradeoff using imprecise hardware," pp. 504-509, in DAC, 2012.
- [5] Zhixi Yang, Jie Han, Fabrizio Lombardi, "Approximate Compressors for Error- Resilient Multiplier Design", IEEE, 2015.
- [6] A. Momeni, J. Han, P. Montuschi and F. Lombardi, "Design and analysis of Approximate Compressors for Multiplication," IEEE Trans. Computers, vol.64, no.4, pp.984-994, April, 2015.
- [7] C. Bickerstaff, E. E. Swatzlander and M .J. Schulte, "Analysis of column compression multipliers," IEEE Symp. On Computer Arithmetic, June, 2001.
- [8] Muhammad Hussnain Riaz, Syed Adrees Ahmed, Qasim Javaid, Tariqq Kamal, "Low Power 4x4 Bit Multiplier Design using Dadda Algorithm and Optimized Full Adder," International Bhurban Conference on Applied Science & Technology (IBCAST), IEEE, Islamabad, Pakistan, January, 2018.
- [9] V. M. Dhivya, A. Sridevi, A. Ahilan, "A High Speed Area Efficient FIR Filter using Floating Point Dadda Algorithm," International Conference on Communication and Signal Processing, April, 2014.
- [10] J. Han and M. Orshansky, "Approximate Computing: An Emerging Paradigm For
- [11] Y. Kyaw, W. L. Goh, and K. S. Yeo, "Low-Power High Speed Multiplier for Error Tolerant Architecture," International

Conference Electron Devices and Solid State Circuits (EDSSC), 2010.

[12] Tongxin Yang , Tomoaki Ukezono, Toshonori Sato, “A Low-Power High Speed Accuracy-Controllable Approximate Multiplier Design”, IEEE 2018.

[13] K. Bhardwaj , P. S. Manenad J.Henkel , “ Power and area efficient Approximate Wallace Tree Multiplier for error resilient systems.” ISQED, March, 2014.

[14] C. H. Lin and I.C. Lin, “ High Accuracy Approximate Multiplier with Error Correction,” In Proc. ICCD,13: In the 2013 IEEE 31<sup>st</sup> International Conference on Computer Design (ICCD), Asheville, NC,USA, pp.33-38, October, 2013.

[15] S. Hashemi ,R. I. Bahar, and S. Reda, “DRUM: A Dynamic range unbiased Multiplier for approximate applications ,” IEEE, November, 2015.

