International Journal of Future Generation Communication and Networking
Vol. 13, No. 2, 2020 pp.1191-1205

A Novel FPGA Implementation Of Error Reduction In 8,16 And 32-Bit Scalable Approximate Rounding Based TOSAM (3,7) And (4,8) Multiplier

G. Erna[#1] and S. Tamilselvan[*2]

[1]*Research Scholar, Department of ECE, Pondicherry Engineering College, Puducherry, India, ernamist@gmail.com*

[2]*Associate Professor, Department of ECE, Pondicherry Engineering College, Puducherry, India, tamilselvan@pec.edu*

Abstract

In recent technology of Arithmetic application will have number of approximate multipliers, approximate adders, this work will reduced the complexity in those approximate multiplier and adder by way of uniqueness approach to compact area, delay and power. In the Scalable method of approximate signed and unsigned multiplier in truncated rounding technique will present to reduced number of logic gates in partial products with help of leading one bit architecture. In the approximate signed and unsigned multiplication design be perform with using arithmetic operation, truncation operation, absolute operation for shift with add accumulation. In this process of TOSAM will have number of modes it will differentiate based upon height (h) and truncated (t) such as (h,t) it will described in the architecture TOSAM(0,2), TOSAM(0,3), TOSAM(1,5), TOSAM(2,6), TOSAM(3,7), TOSAM(4,8), TOSAM(5,9). Here this TOSAM Operation include more absolute error in the LSB data shift Unit, thus proposed line of attack will customized in all the arithmetic operations of shift and add unit with using XOR-MUX Full adder to find a improved solution and reduced the absolute error and it will proved higher improvements of area and energy utilizations. In this proposed novelty work will modified approximate signed multiplier architecture as per absolute error reduction in TOSAM (3,7), TOSAM (4,8), TOSAM (5,9) and consequently prove the compared terms of area, delay and power. To conclude this work will designed in Verilog HDL and simulated in Modelsim, Synthesized in Xilinx 14.2.

Keywords *:—TOSAM (Truncation Rounding based Scalable approximate Multiplier), HDL(Hardware Description Language), FPGA (Field Programmable Gate Array).*

I. INTRODUCTION

In a recent technology of digital signal processing application, a multiplier is a priority one with low area and low power utilizations. Now a day's a approximate multiplier will functioning good manner to reduced area, delay and power, but it will have more error rate and more energy utilizations. These approximate computing based multiplier will not computing proper results in error-resilient applications example in such as audio processing and video processing, with signal processing applications. More in particular in gadgets based signal processing applications will take more energy utilization with high latency in arithmetic operations. To overcome this problem of high latency and energy utilization with error-resilient here work will introduced a scalable approximate multiplier with using truncated rounding based technique which present, to minimized a number of partial products which based on leading one bit position. In this multiplier functionality will have three steps, a first steps to generate a partial products based upon input operands, a second sets to accumulated the partial products until two rows remain, a third steps to remained two rows are added and get the final outputs. In the existing method of approximate signed multiplier will have a six number of blocks is such as 1)

ISSN: 2233-7857 IJFGCN

Approximate Absolute Unit, 2) Leading One Detector Unit, 3) Truncation Unit, 4) Arithmetic Unit, 5) Shift Unit and 6) Sign and Zero Detector Unit. Here, Arithmetic Unit will functioning with two conventional full adder, one n x n bit multiplier and Shift Unit. Fig.1 will exposed the Architecture of Existing Approximate Signed and Unsigned Multiplier [1].

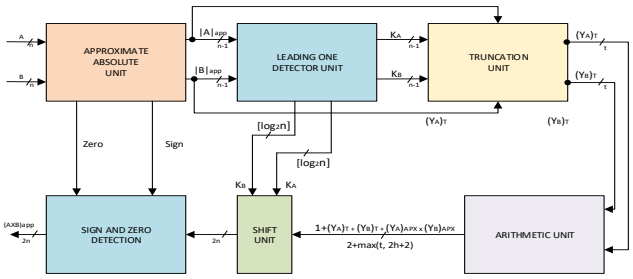

Figure 1 : Block diagram of the Existing approximate signed Multiplier

A Conventional adder is one of the most fundamental and most important arithmetic operations which adds the two binary (1,0) digits. For one bit operation of (a, b) addition it will take one half adder and (a, b, c) addition it will take two half adder instead of one full adder. Here, one half adder will take two logic gates therefore one full adder will take five logic gates with carry operations, thus conventional full adders will performing more logic size. Here, the proposed work will introduced a level synchronous XOR MUX full adder design regarding to reduce a number of logic gates with high speed and minimum area. Here a Approximate Rounding based method of Multiplier will not used in fixed size it's a variable size its depends upon TOSAM mode operations example for TOSAM (3,7) will have to used 4x4 bit multiplier, TOSAM (4,8) will have to used 5x5 bit multiplier, TOSAM (5,9) will have to used 6x6 bit multiplier, therefore this multiplier will take number of partial products and number of conventional full adders with more logic size, therefore this proposed work will replace the full adder instead of XOR-MUX full adder to reduced the logic size in Multiplier design with good performance of area reduction. Next Shift Unit will functioning shifted with number of binary zeros in LSB (Lest Significant bit), for this zero shifting operation a digit value will have loss SNR (Signal to Noise Ratio) level, thus proposed work will introduced one's shifting operations, for example "0100_1111" is 79, "0100_0000" is 64, here difference is "15" thus it will concern the absolute error diversity in approximate rounding based multiplier [10].

In this paper, we elaborated an approximate procedure for decreasing the number of logic size of all the arithmetic operations and reduced the absolute error in TOSAM (3,7) and TOSAM (4,8). A Leading one bit possible also be changes depends upon the bit size and it help to find a position to truncated bits, here height will mention 'h' and truncated will bring up in 't', example TOSAM (h,t). For this proposed operations to reduced the error resulting from the rounding based technique it will find approximate amount of truncated values by rounding then to improve higher accuracy and performance compared to those of the state of the art approximate multipliers of Wallace Tree Multiplier, DRUM, LETAM, U-ROBA, DSM, DQ4:2C4 [11].

In this paper, our aim to provide the consequences in less area and less power utilization using XOR-MUX full adder in Approximate Rounding Based TOSAM Multiplier with proved the effectiveness in terms of area, power and delay comparison. Section II present particulars of XOR-MUX Full adder design with Conventional Full Adder. Section III presents details of Operations of Scalable Approximate Rounding based TOSAM Multiplier. Section IV presents proposed design of Rounding Based TOSAM Multiplier. Section V presents a FPGA Implementation of Proposed TOSAM architecture with result, implementation and comparisons. Section VI drawn a conclusion of this paper with future enhancements.

II. OPERATION OF XOR MUX & CONVENTIONAL FULL ADDER

In a arithmetic addition procedure of full adders will enclose a more critical paths and data paths on digital signal processing applications, thus its core representation will used for many arithmetic operations such as address computation, division, multiplication, cache with memory access in floating point unit (FPU) and arithmetic logic units (ALU). Here, this paper introduced a two consecutive stage of XOR and Multiplexer based single bit full adder design with less area and power optimization. While this XOR and Multiplexer based Full adder design will compare to conventional Full adder design with area, delay and power [2].

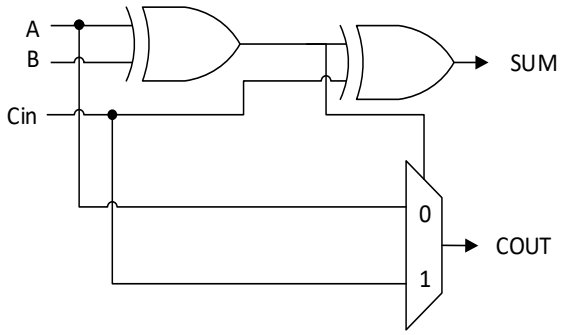

Figure 2 : Proposed XOR-MUX Full Adder

Fig.2 shows the proposed architecture of XOR with MUX Full adder design while employ two consecutive stages of XOR gate for Sum operation and 2:1 non-inverting multiplexer at the same time as using for Carry operations, it will take totally 2 logic gates and 1 multiplexer. The truth table of XOR-MUX Full adder design will shown in Table.1.

Table 1 : Truth Table of XOR-MUX & Conventional Full Adder

C_{IN}	A	B	SUM	COUT
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

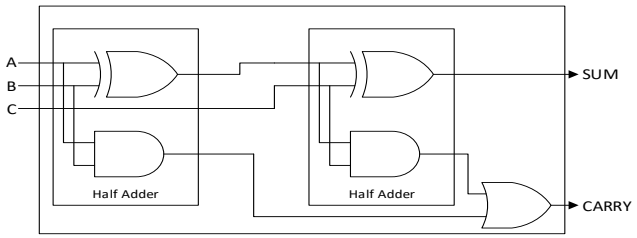

Figure 3 : Conventional Full Adder

A Gate level structure of Conventional Full adder design will shown in Fig.3. In this Fig shown 2 XOR Gate, 2 AND Gate and 1 OR Gate based structure, it will take totally 5 logic gate based structure, compared to XOR MUX Full adder design this Conventional Full Adder will take additionally 3 logic gates [12]. Table.2 and Fig.4 will shown the comparisons and analysis for Single Bit XOR MUX and

ISSN: 2233-7857 IJFGCN

Conventional Full Adder Design.

Table 2 : Comparisons of XOR MUX and Conventional Full adder Design

	XOR MUX Full Adder	Conventional Full Adder
Slice Registers	0	0
LUT	1	2
Occupied Slice	1	1
IOB	5	5
Delay(ns)	6.110	8.025
Power(mW)	14	14

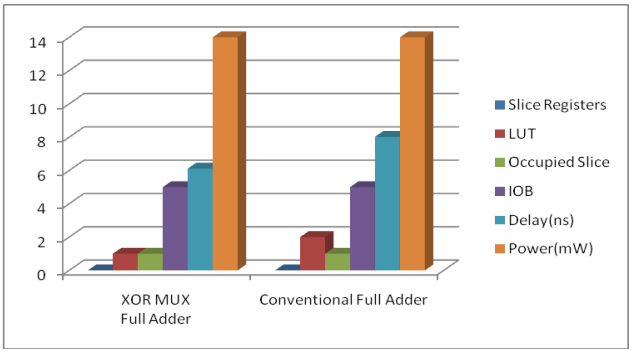

Figure 4 : Comparison and analysis chart for Single Bit XOR MUX and Conventional Full Adder Design

III. OPERATION OF SCALABLE APPROXIMATE ROUNDING BASED TOSAM MULTIPLIER

Approximate Rounding Based Multiplier will need to almost ordinary error distribution with zero mean value, then only its consume less energy compared to that of exact approximate multiplier of Wallace Tree, DRUM, LETAM, U-ROBA, DSM, DQ4:2C4. This proposed work of approximation signed multiplier will have some key contributions as follow 1) A leading one bit position of scalable approximate multiplier will find a positions of highest digit logic one and it will be rounded and truncated to improve the accuracy of approximate multiplier operations. The Schematic diagram of Leading One Detector Unit for 8-Bit input operands is depict in Fig.6, this circuit diagram will performance only with AND, NOR gate operations, In[3:0] will functioning as LSB Operations and In[7:4] will functioning with MSB Operations. 2) Examination of this Rounding based Multiplier will have two thing is priority such as t (truncation) and h (height) it's supportive to tradeoff accuracy, energy utilization with delay. This TOSAM Operation of result by multiplying A and B might be Calculated as in equation (1)

$$A \times B = 2^{kA+kB} \times X_A \times X_B \quad (1)$$

Where, X_A and X_B will be calculated from Approximation Rounding of Truncation Unit it's also called Y_A and Y_B, and this leading one bit output will called K_A and K_B it will denotes the positions with $\log_2 n$. For examples, assume that $|A|_{app} = (00011001)_2$, in this case $K_A = (00010000)_2$ and $k_A = (0100)_2 = 4$. These signals of $(Y_A)_t$ and $(Y_B)_t$ are exert to arithmetic unit to calculate the expression of equation (2)

$$= 1 + (Y_A)_t + (Y_B)_t + (Y_A)_{APX} \times (Y_B)_{APX} \quad (2)$$

As an example, the steps of multiplying A by B for the case of t=7 and h=3 are depicted in Fig.5, where TOSAM(3,7). As per the architecture of Fig.1 Existing Approximate signed Multiplier Block diagram with Numeric Example will present, here the initial case of A and B inputs will take 16 Bit Binary values A = 11761 = (00101101), B = 2482, the approximate results [$(A \times B)_{APX}$] is equal to 28 901 376 while the exact results [$(A \times B)_{EXACT}$] is equal to 29 190 802. In this case the absolute error is 298 426. which is about 0.99% of the exact output (the error is less than 1% in this case).

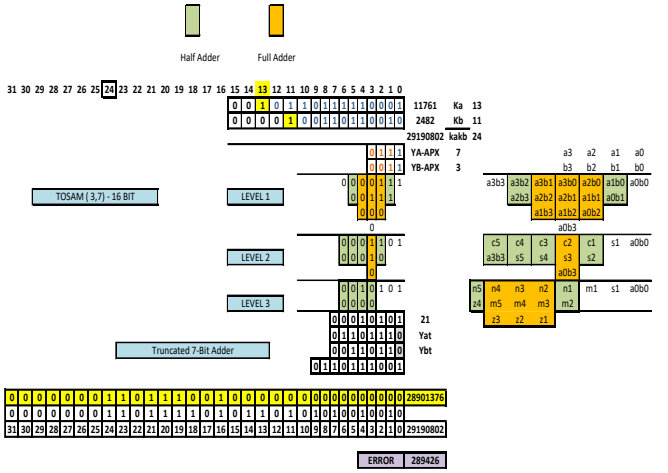

Figure 5 : Numeric Example of Existing 16-Bit TOSAM(3,7) Multiplier

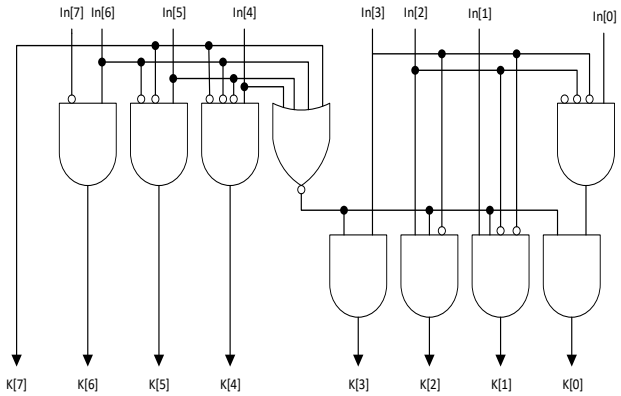

Figure 6: Schematic of the Leading One Detector Unit for 8-bit input operands.

In the Proposed Method of Approximate Signed Multiplier Block diagram with Numeric example of 16-Bit TOSAM(3,7) Multiplier will present in Fig.7, here the initial case of A and B input will take 16 Bit Binary value A = 11761 = (00101101), B = 2482, the proposed operation will be modified in the shift unit operations in such the output value shifted with '1' logic, as per that approximate results [$(A \times B)_{APX}$] is equal to 28 966 911 while the exact results $(A \times B)_{EXACT}$] is equal to 29 190 802. In this case the absolute error is 223 891 which is about 0.76% of the exact output (the error is less than 0.99% in this case). These Error analysis of percentage calculation of TOSAM Multiplier to calculate the term of equation (3)

$$\text{Error analysis (\%)} = [[(A \times B)_{APX}] - [(A \times B)_{EXACT}]] \times 100 / [(A \times B)_{EXACT}] \quad (3)$$

ISSN: 2233-7857 IJFGCN

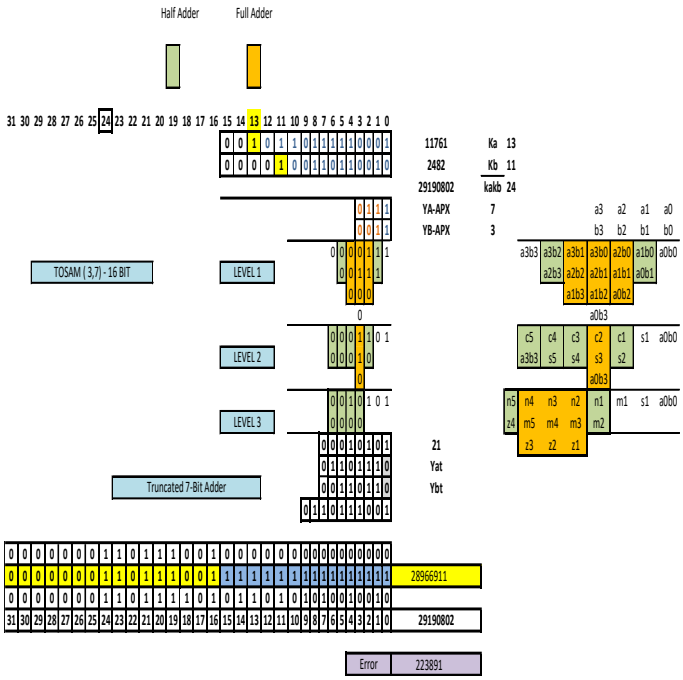

Figure 7 : Numeric Example of Proposed 16-Bit TOSAM(3,7) Multiplier

As per this Numeric Example will be differ in Error Analysis while configure different bit size, the Numeric Example of 8-Bit, 16-Bit and 32-Bit of Existing and Proposed TOSAM(3,7) and TOSAM(4,8) Multiplier will shown in Fig.8 to Fig.17, it will shown Different level and different Multiplier size with Error Analysis (%). Error Analysis results of 8-Bit, 16-Bit, 32-Bit TOSAM (3,7) and TOSAM(4,8) will shown in Fig.18 and Fig.19.

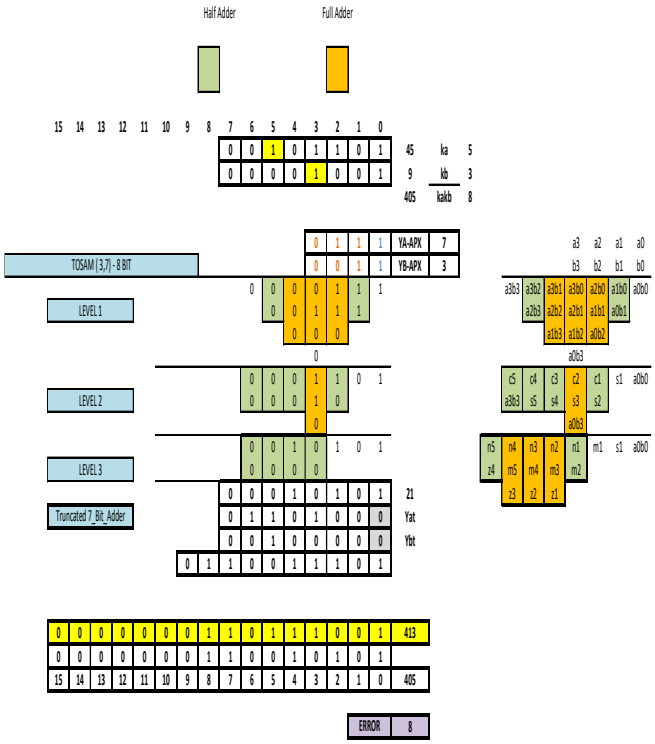

Figure 8: Numeric Example of Existing 8-Bit TOSAM(3,7) Multiplier

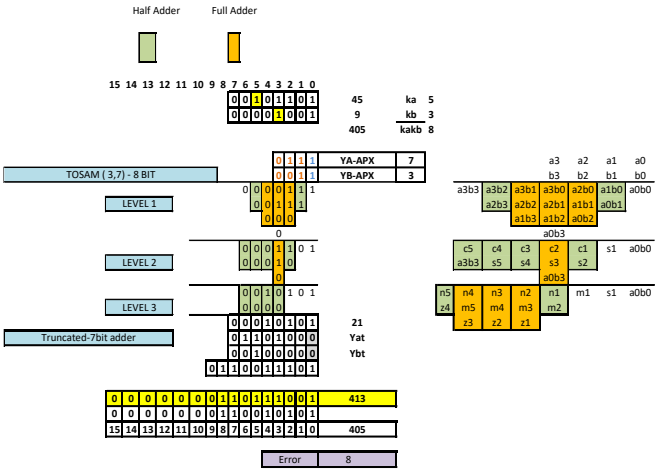

Figure 9: Numeric Example of Proposed 8-Bit TOSAM(3,7) Multiplier

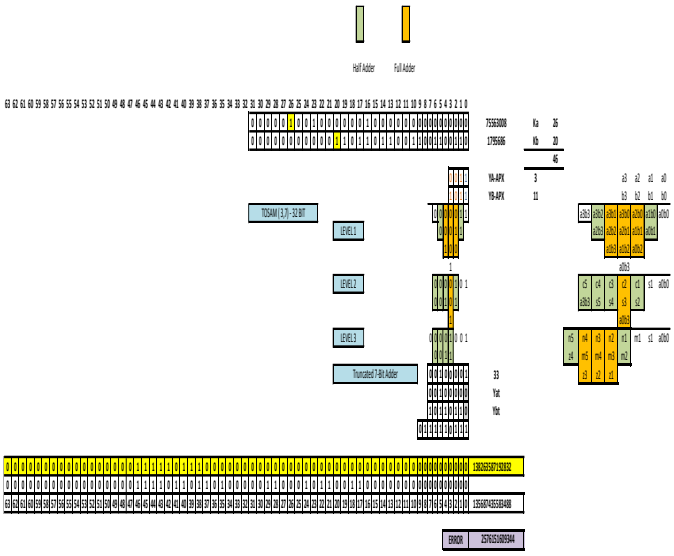

Figure 10: Numeric Example of Existing 32-Bit TOSAM(3,7) Multiplier

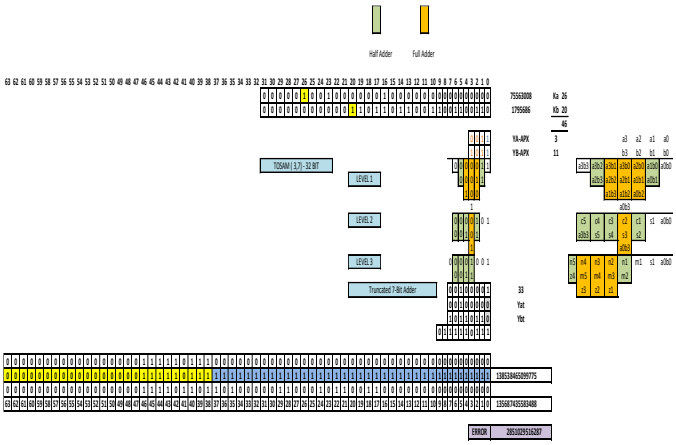

Figure 11 : Numeric Example of Proposed 32-Bit TOSAM(3,7) Multiplier

ISSN: 2233-7857 IJFGCN

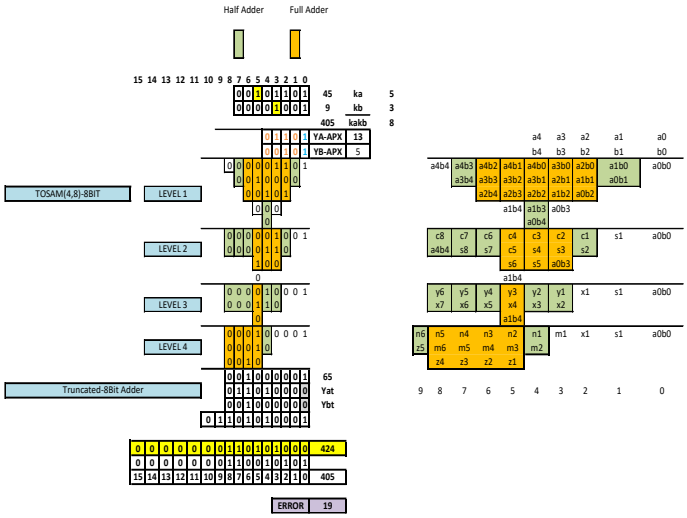

Figure 12 : Numeric Example of Existing 8-Bit TOSAM(4,8) Multiplier

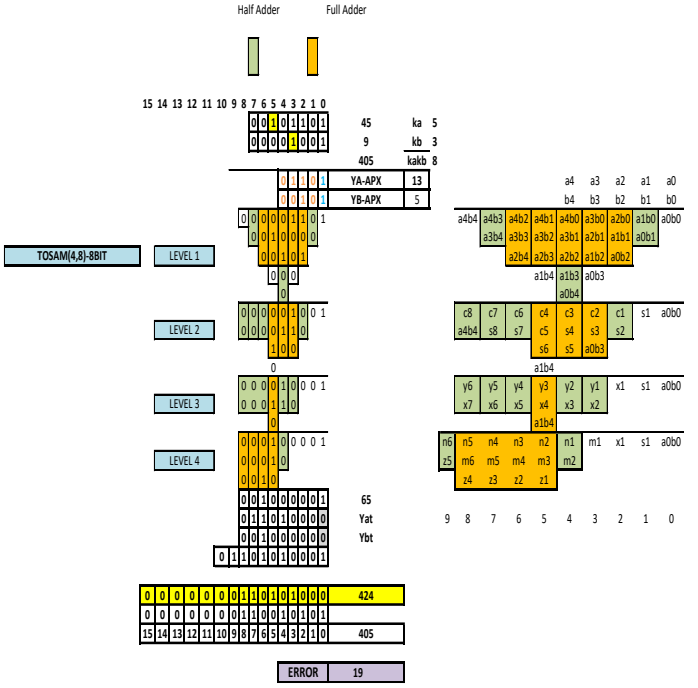

Figure 13 : Numeric Example of Proposed 8-Bit TOSAM(4,8) Multiplier

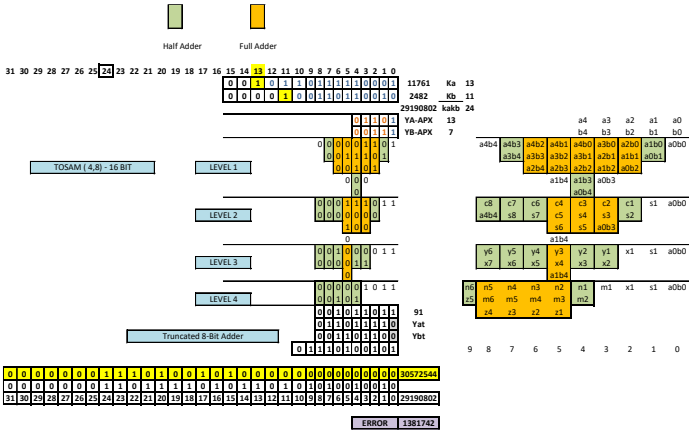

Figure 14 : Numeric Example of Existing 16-Bit TOSAM(4,8) Multiplier

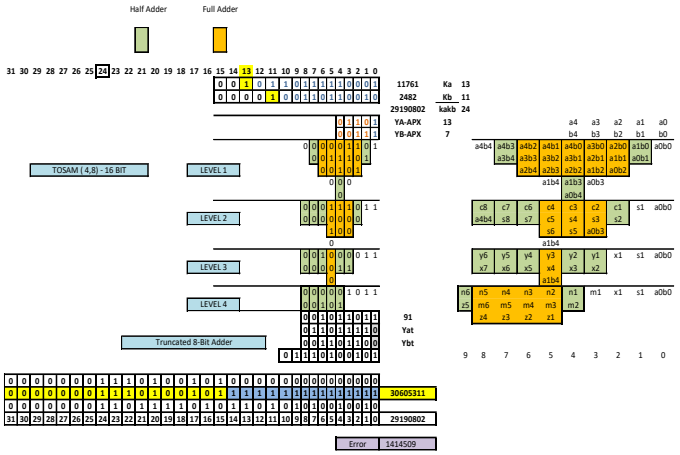

Figure 15 : Numeric Example of Proposed 16-Bit TOSAM(4,8) Multiplier

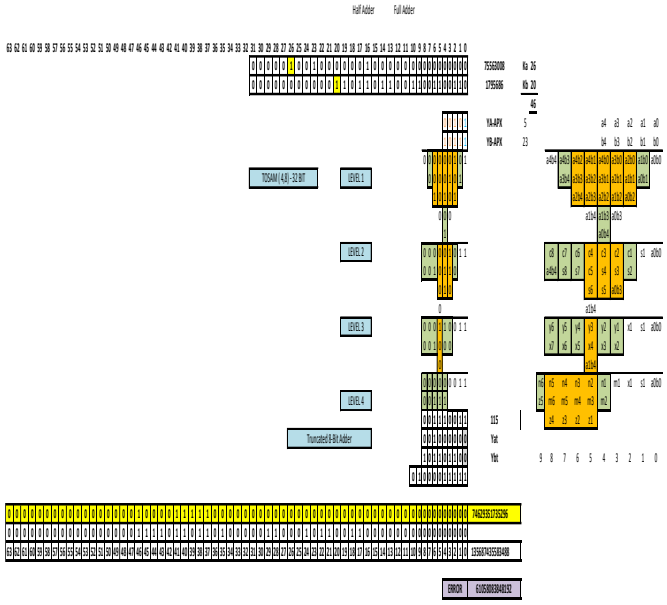

Figure 16: Numeric Example of Existing 32-Bit TOSAM(4,8) Multiplier

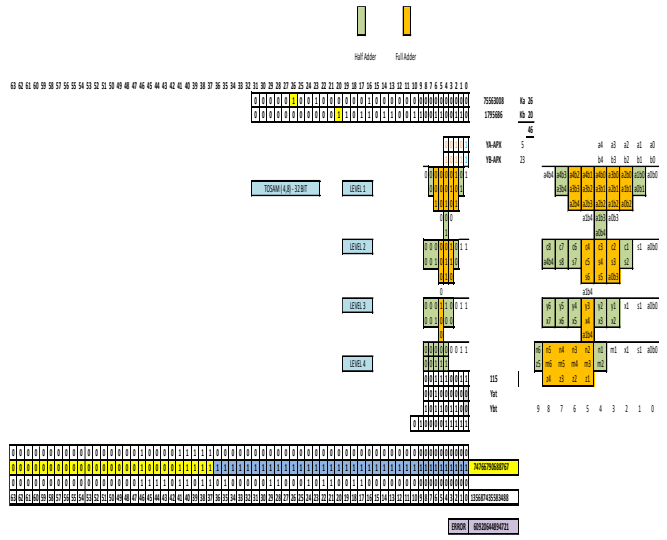

Figure 17 : Numeric Example of Proposed 32-Bit TOSAM(4,8) Multiplier

ISSN: 2233-7857 IJFGCN

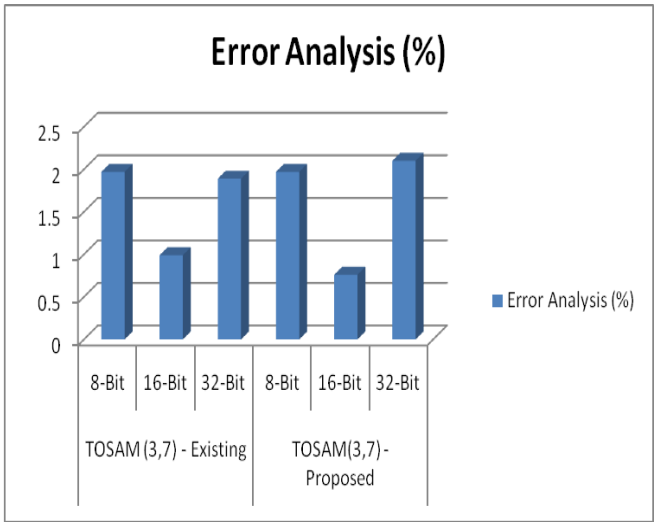

Figure 18 : Error Analysis of TOSAM(3,7) Multiplier

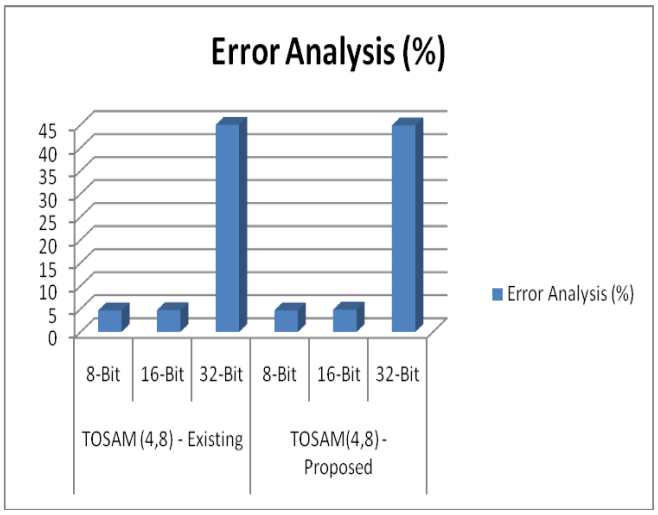

Figure 19 : Error Analysis of TOSAM(4,8) Multiplier

IV. PROPOSED DESGIN OF ROUNDING BASED TOSAM MULTIPLIER

The Block diagram of the proposed approximate singed multiplier is depicted in Fig.20. First, the input of approximate absolute value of the input operands ($|A|_{app}$, $|B|_{app}$) is determined using the Approximate Absolute Unit Block its similar to the one's complement method [3], here the input sign bit will be detected to check its positive value or negative value, if the input is negative the output value will be one's complemented, if the input is positive the output value will be not changed. After the Approximate absolute unit $|A|_{app}$, $|B|_{app}$ value will given to Leading One Detector Unit [4], and the position of leading one bit are found using (4).

$$K[i] = \left(\bigwedge_{j=i+1}^{n-2} \overline{I[j]} \right) \wedge I[i] \text{ for } 0 \le i \le n-2 \qquad (4)$$

where I can be either $|A|_{app}$ or $|B|_{app}$. Only one bit of the signal K is "1" revealing the position of the input leading one bit and assume that the input and output of this unit are I and Y_t. In this case, the *i*th bit of the output can be generated using (5).

$$(Y)_t[i] = \bigvee_{j=0}^{n-2} (K[j] \wedge I[j+i-t]) \text{ for } i < t. \qquad (5)$$

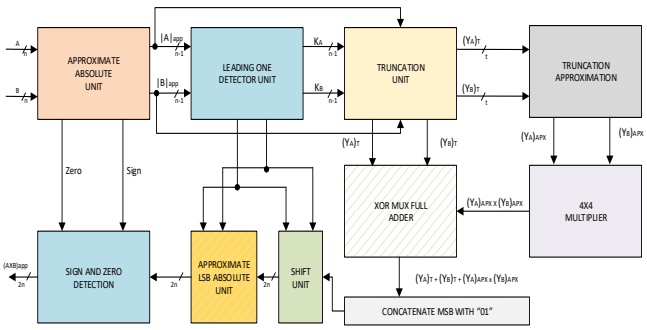

Figure 20 : Block diagram of the Proposed Approximate Signed Multiplier

These Leading one detector unit will generate a output K_A, K_B with $\log_2^n$. For example, assume that $|A|_{app} = (0010\_1101\_1111\_0001)_2$, in this case $K_A = (0010\_0000\_0000\_0000)_2$ and $K_A \log^2_n = (1101)_2 = 13$. Therefore those values will given to Approximate LSB Absolute unit, Shift Unit and Truncation Unit. Once received the value of K_A, K_B into the Truncation Unit, it will generate the value of $(Y_A)_T$ and $(Y_B)_T$ and its given the value to Truncation Approximation block it will work depends upon Mode operation of TOSAM(h,t) here ' h ' is height and ' t ' is truncation. For example, assume that $K_A \log^2_n = (1101)_2 = 13$, $(Y_A)_T = (0\_1101\_11)_2$, and $(Y_A)_{APX} = (0110\_1111)_2$, where t = 7. After generation of $(Y_A)_{APX}$ and $(Y_B)_{APX}$ the value will given to the Multiplier with XOR MUX Full Adder design, and also given to Concatenate MSB with "01" block regarding to calculate the term (6)

$$1+ (Y_A)_t + (Y_B)_t + (Y_A)_{APX} \text{ x } (Y_B)_{APX} \qquad (6)$$

Here, the Multiplier size of operation will be modified depends upon mode of operations eg: TOSAM(3,7) will need 4x4 Multiplier, TOSAM(4,8) will need 5x5 Multiplier and TOSAM(5,9) will need 6x6 Multiplier, here this proposed will design this Multiplier using XOR-MUX Full adder to reduced the number of logic gates in Multiplier design also. Once concatenate with "01" the data will given to Shift Unit. In this block of proposed Shift Unit will shifted the bit with operation of $K_B \log^2_n$ and $K_A \log^2_n$ values, these log value will be added and get the value of $K_{AB} \log^2_n$ value, then the value will be shifted as per the log. For example $K_A = 13$, $K_B = 11$, $K_{AB} = 24$, so the value will be shifted and started with 24^{th} bit, Shift Unit output $(0000\_0001\_1011\_1001\_0000\_0000\_0000\_0000)_2$. Then the Approximate LSB Absolute Unit will receive the data from Shift Unit, this block will K_{AB} value and it will replace the LSB from '0' logic to '1' logic to increases the accuracy in output bits, then the data will given into Sign and Zero Detection Unit to switch the One's complement operations its depends upon zero and sign operations. Finally the $(AxB)_{App}$ data will presented in the Output on 2n Size.

V. FPGA IMPLEMENTATION OF PROPOSED TOSAM ARCHITECTURE WITH RESULTS, IMPLEMENTATION AND COMPARISONS

In this section, will analysis a existing and proposed design of FPGA implementation of TOSAM architecture with all the modules such as,

* TOSAM (3,7) - Existing System (8-Bit, 16-Bit, 32-Bit)
* TOSAM (3,7) - Proposed System (8-Bit, 16-Bit, 32-Bit)
* TOSAM (4,8) - Exiting System (8-Bit, 16-Bit, 32-Bit)
* TOSAM (4,8) - Proposed System (8-Bit, 16-Bit, 32-Bit)

These all the method will be developed in Verilog HDL and simulated in Modelsim and Synthesized in Xilinx FPGA S6LX75-2CSG484 and the design parameter of the proposed configuration are compared with some state-of the art approximate multipliers. In this paper, we have well thought-out the prior

ISSN: 2233-7857 IJFGCN

mechanism that have proved best presentation with LETAM [5], DRUM [6], DSM [7], RoBA [8] and DQ4:2C4 [9] architectures. We have implement both signed and unsigned 8-bit, 16-bit and 32-bit multiplier to show how the delay, power and area are improved by increasing the width of the multiplier operands. Fig.21 and Fig.22 will shown the Existing and proposed architecture of TOSAM(3,7) Multiplier with comparison analysis of Slice register, LUT, Occupied Slice, Delay, Power and Error Analysis of Existing and Proposed will shown in Table 4 and Table 5, and analysis chart will shown in Fig.23 and Fig.24. Simulation result analysis of existing and proposed TOSAM (3,7) and TOSAM(4,8) will shown in Fig.25, Fig.26, Fig.27 and Fig.28.

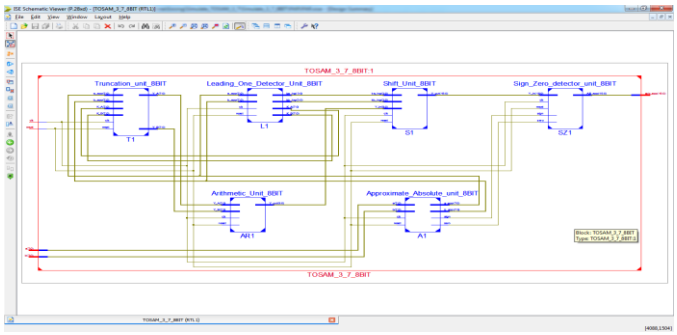

Figure 21 : Existing System RTL Schematic of TOSAM (3,7) Multiplier

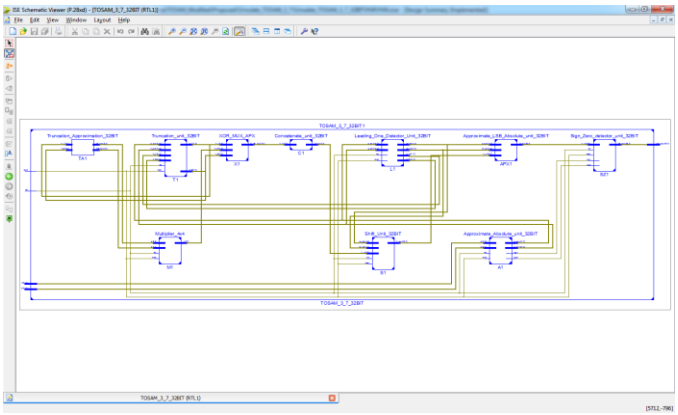

Figure 22 : Proposed System RTL Schematic of TOSAM Multiplier

Table 3: Comparisons Parameter Table for TOSAM (3,7) Approximate Multipliers with 8-Bit, 16-Bit and 32-Bit

	TOSAM (3,7) - Existing			TOSAM(3,7) - Proposed		
	8-Bit	**16-Bit**	**32-Bit**	**8-Bit**	**16-Bit**	**32-Bit**
Slice Register	15	17	19	6	8	10
LUT	195	359	765	187	257	713
Occupied Slice	73	150	383	71	90	332
IOB	34	66	130	34	66	130
Delay (ns)	2.301	2.101	2.121	1.724	1.871	2.088
Power (W)	3.325	3.324	3.343	0.182	0.236	3.360
Error Analysis	1.97	0.99	1.89	1.97	0.76	2.10

(%)						

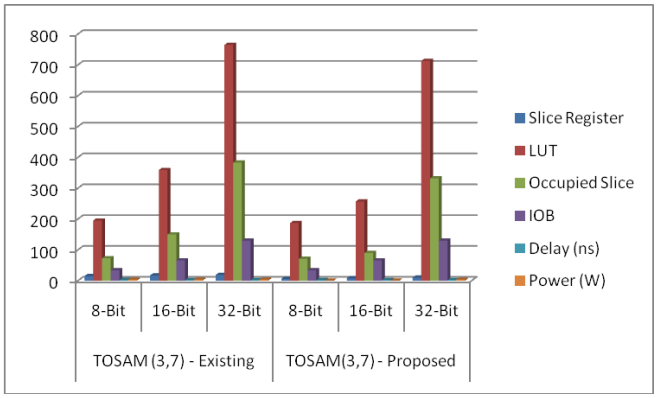

Figure 23 : Comparison analysis results of TOSAM (3,7) Approximate Multipliers with 8-Bit, 16-Bit and 32-Bit

Table 4: Comparisons Parameter Table for TOSAM (4,8) Approximate Multipliers with 8-Bit, 16-Bit and 32-Bit

	TOSAM (4,8) - Existing			TOSAM(4,8) - Proposed		
	8-Bit	**16-Bit**	**32-Bit**	**8-Bit**	**16-Bit**	**32-Bit**
Slice Register	16	18	19	6	8	10
LUT	228	397	784	178	367	931
Occupied Slice	97	158	350	83	169	423
IOB	34	66	130	34	66	130
Delay (ns)	2.299	2.277	2.118	2.221	2.108	2.113
Power (W)	3.325	3.339	3.560	3.410	3.471	3.366
Error Analysis (%)	4.69	4.73	44.99	4.69	4.84	44.89

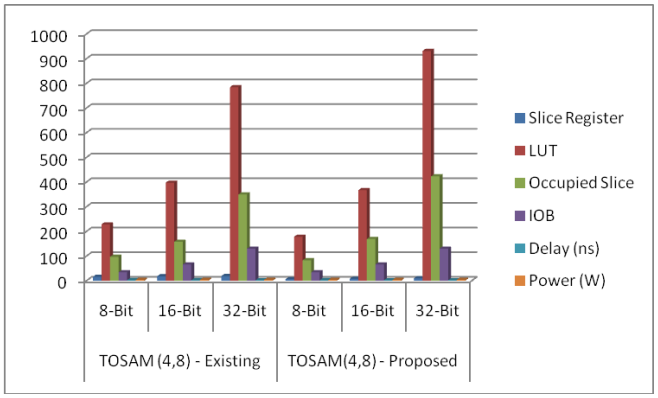

Figure 24 : Comparison analysis results of TOSAM (4,8) Approximate Multipliers with 8-Bit, 16-Bit and 32-Bit

ISSN: 2233-7857 IJFGCN

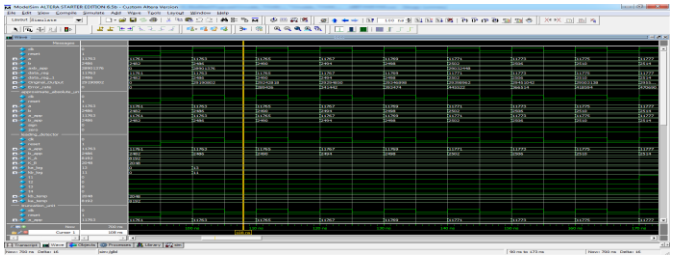

Figure 25 : Simulation results of Existing TOSAM(3,7) - 16-Bit Approximate Multiplier

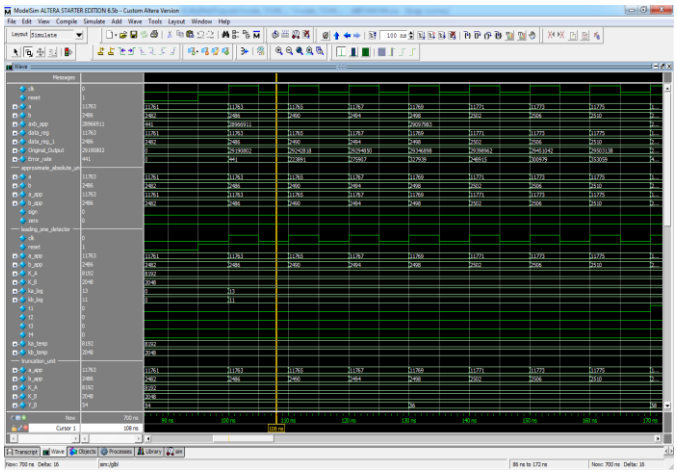

Figure 26 : Simulation results of Proposed TOSAM(3,7) - 16-Bit Approximate Multiplier

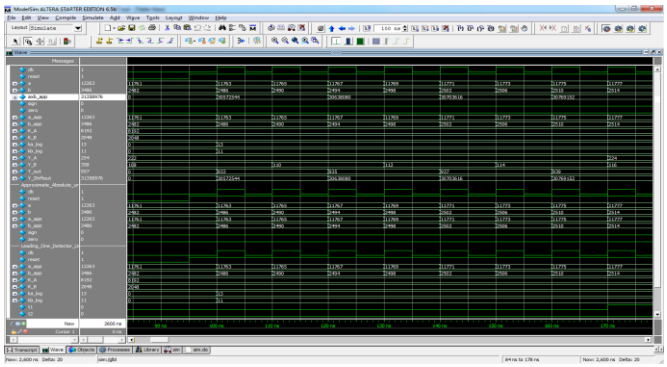

Figure 27 : Simulation results of Existing TOSAM(4,8) - 16-Bit Approximate Multiplier

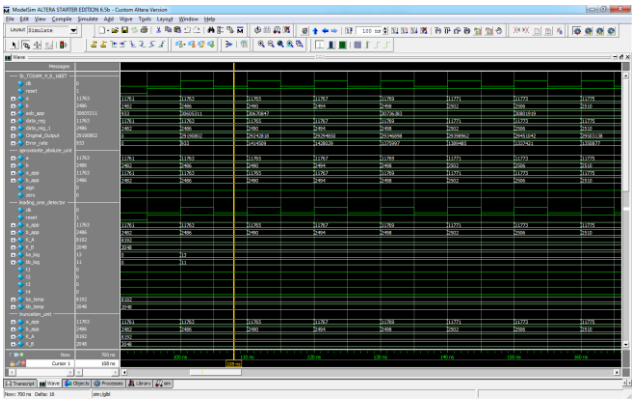

Figure 28 : Simulation results of Proposed TOSAM(4,8) - 16-Bit Approximate Multiplier

VI. CONCLUSION

In this paper, we suggested a novel approach to reduced the number of logic gates and reduced the error in approximate truncation multiplier in which the mode operation using two parameters, t (truncation) and h(height). This proposed multiplier was scalable and outperformed other approximate multiplier in terms of area, delay and power with energy utilizations, here the 32-bit of TOSAM multiplier on average, improved the energy utilization 99% compared to the exact ROBA, Wallace Tree, LETAM, DRUM, DSM and DQ4:2C4 multipliers, and also reduced the area and power utilizations. This proposed approximate Multiplier of TOSAM(3,7), TOSAM(4,8) will proved the efficiency in all the case's and its suitable to all application domains such as image processing, digital signal processing, and gadgets and classification based applications.

VII. REFERENCES

[1] "TOSAM: An Energy Efficient Truncation and Rounding Based Scalable Approximate Multiplier", Shaghayegh Vahdat, Mehdi Kamal, Ali Afzali-Kusha, Massoud Pedram, 2019, IEEE Transactions on Very Large Scale Integration.
[2] "High Speed Gate Level Synchronous Full Adder Design", Padmanabhan Balasubramanian, Nikos E, Mastorakis, School of Computer Science, University of Manchester, United Kingdom, WSEAS TRANSACTIONS on CIRCUITS and SYSTES.
[3] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, and M. Pedram, "RoBa multiplier: A rounding-based approximate multiplier for high-speed yet energy-efficient digital signal processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 2, pp. 393–401, Feb. 2017.
[4] S. Vahdat, M. Kamal, A. Afzali-Kusha, M. Pedram, and Z. Navabi, "TruncApp: A truncation-based approximate divider for energy efficient DSP applications," in Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE), Lausanne, Switzerland, Mar. 2017, pp. 1635–1638.
[5] S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "LETAM: A low energy truncation-based approximate multiplier," Comput. Elect. Eng., vol. 63, pp. 1–17, Oct. 2017.
[6] S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Austin, TX, USA, Nov. 2015, pp. 418–425.
[7] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processing and classification applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.
[8] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, and M. Pedram, "RoBa multiplier: A rounding-based approximate multiplier for high-speed yet energy-efficient digital signal processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 2, pp. 393–401, Feb. 2017.
[9] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 4, pp. 1352–1361, Apr. 2017.
[10] M. Ha and S. Lee, "Multipliers with approximate 4–2 compressors and error recovery modules," IEEE Embedded Syst. Lett., vol. 10, no. 1, pp. 6–9, Mar. 2018.
[11] R. Marimuthu, Y. E. Rezinold, and P. Mallick, "Design and analysis of multiplier using approximate 15–4 compressor," IEEE Access, vol. 5, pp. 1027–1036, 2017.
[12] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 65, no. 12, pp. 4169–4182, Dec. 2018.

ISSN: 2233-7857 IJFGCN