Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7778383

DESIGN AND IMPLEMENTATION OF AN ENERGY EFFICIENT MULTIPLIER FOR APPROXIMATION USING DRUM

P.Bharathi Anantha Lakshmi[1], Y.Arpitha[2], Swathi Kambhampati[3],Prathyusha Kuncha[4]
PG Scholor, Department of ECE, NRI Institute of technology, Pothavarappadu,
Corresponding author email: bharathi78414@gmail.com
Associate Professor, Department of ECE, NRI Institute of technology, Pothavarappadu, email: arpitha@nriit.edu.in
Associate Professor, Department of ECE, NRI Institute of technology, Pothavarappadu , e-mail: swathi@nriit.edu.in
Associate Professor, Department of ECE, NRI Institute of technology, Pothavarappadu,e-mail: prathyushakuncha@gmail.com

ABSTRACT:
Approximate computing is one in all fine suited green information processing for mistakes resilient applications, such as sign and photograph processing, pc vision, system learning, information mining etc. Approximate computing reduces accuracy which is appropriate as the price of increasing the circuit traits relies upon at the application. Desirable accuracy is the threshold factor for controlling the change off, among accuracy and circuit traits beneath the manipulate of the circuit designer. In this work, the rounding approach is brought as a green technique for controlling this change off. In this regard multiplier circuits as a vital constructing block for computing in maximum of the processors have been taken into consideration for the assessment of the rounding approach efficiency. The effect of the rounding technique is investigated by assessment of circuit traits for 3 multipliers. These 3 multipliers are the traditional Wallace tree accurate multiplier, DRUM [4] the currently proposed approximate multiplier and the rounded primarily based totally approximate multiplier proposed in this work. Simulation outcomes for three selected technologies display good sized development at the circuit traits in phrases of power, area, speed, and electricity for proposed multiplier in assessment with their counterparts. Input information rounding sample and the chance of the repetition for rounded values has been brought as critical gadgets to manipulate the extent of the accuracy for every variety of the information with minimal price at the hardware.
Index Terms -Approximate computing, error metrics, Dynamic Range, RTL schematic.

INTRODUCTION

Some applications like wireless communication, data mining can tolerate errors to some extent. The error tolerance level varies based on various parameters like noisy input, data redundancy. The main demand for many designs turns out to be a higher speed. Another important parameter that plays a vital role in the design is power consumption. Approximate computing provides the option for the users to trade-off among all these parameters. A simple approach for analysing approximate computing is to design the approximate circuits and the error percentage is controlled based on the design of the modules in the circuit. These approximate blocks are then introduced in the main computational unit of the system. Approximate arithmetic conveys

us that any basic building block in a system can be made approximate which in turn reduces the power by a huge amount and the propagation delay. In this paper the design of the approximate multiplier is done. Since the multiplier is the basic unit of many processors and it occupies more silica area and consumes more power. Thus, it turns out to be an impressive idea to approximate the multiplier. A Dynamic Range Unbiased Multiplier (DRUM) is designed to meet the approximate applications. The unbiased design of the multiplier gives the advantage of error cancelation instead of error accumulation in the output. Another important advantage is that trade-off can be made between accuracy and power loss. An analytical formula is used to calculate the average error and maximum error. The formula is based on the parameters which determine the configuration of the multiplier. The multiplier inaccuracy, power consumption, timing analysis is done as a function of the number of bits truncated from the inputs of the multiplier. After all the analysis, even more, modification is done in the design of the existing DRUM multiplier to improve the accuracy which is termed ad Optimised DRUM multiplier. The error matrix calculation, power analysis, timing analysis is performed for this design as well. The results are compared with the DRUM multiplier. The works done in the paper are as follows. A substantial amount of papers has been referred on the topics of approximate multiplier, approximate computing, types of multiplier, and discussed in the related work section. In the section of proposed work, the proposed multiplier is described elaborately. To increase the accuracy further few changes have been made to the multiplier and the new design is termed as Optimized DRUM multiplier. The brief working of this multiplier is discussed in the section of the optimized DRUM multiplier.

RELATED WORKS

In this section some researches in the field of approximate multiplier and approximate computing is done. Gupta et al. [1] introduced various approximate adders, multipliers by removing some logical blocks which introduces some error but the area and power consumption are reduced drastically. Cong Liu et al. [2] discussed various types of multipliers and their performances. Array multiplier is discussed followed by carry save multiplier which has lesser propagation delay. To decrease the area Wallace tree multiplier and its approximation is discussed. For multiplication of negative number booth multiplier and its approximation is discussed. Finally, for multiplication of decimal numbers floating point multiplier and logarithmic multiplier is explained. Mahadiani et al. [3] suggested an approximate adder by introducing OR gates for computation of sum for the lower bits of the operands, which results in a significant reduction in area. New methodologies for approximations have been discussed. Multiplier with error tolerance has been discussed in where the multiplier has two modules. One module computes the exact multipoint and the other the approximate value. To increase the accuracy an iterative approach is followed in. Mitchel multiplier et al. suggest a multiplier that calculates the partial product with the help of fast adders. Kulkarni et.al suggest the multiplier which computes the exact multiplication value for the MSB and the approximate results for the LSB. Narayanamoorthy et.al introduced the multiplier based on truncation the upper middle and lower parts are truncated and the results are calculated separately which is then added together to form the output. One of the disadvantages in is the number of bits to be truncated is not mentioned which will be corrected in the design of the DRUM multiplier. A recent approach for approximation is on the generation

of approximate logics which is discussed. In this way approximate circuits are synthesized automatically and they are developed as hardware.

EXISTED SYSTEM

In this paper, three factors are being focussed that are suitable for selecting the approximate multiplier. (1) The type of approximate full adder (FA) used to construct the multiplier, (2) the architecture, i.e., array or tree, of the multiplier and (3) the placement of sub-modules of approximate and exact multipliers in the main multiplier module. Based on these factors, the design space is explored for circuit level implementations of approximate multipliers. This paper includes used circuit level implementations of some of the most widely used approximate full adders, i.e., approximate mirror adders, XOR/XNOR based approximate full adders and exact adder cell. These FA cells are then used to develop circuits for the approximate high order compressors as building blocks for 8x8 array and tree multipliers. Then various implementations of higher bit multipliers are developed by using a combination of exact and inaccurate 8x8 multiplier cells. The design space of these multipliers is explored based on their power, area, delay and error and the best approximate multiplier designs are identified.

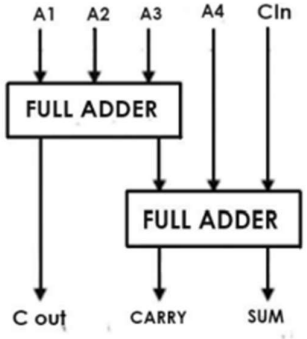


Figure 1. Exact 4:2 compressor

The general block diagram of an exact 4:2 compressor is shown in Figure 1. It consists of five inputs, three outputs and two cascaded full adders. A1, A2, A3, A4 and C_{in} are the inputs and C_{out}, CARRY and SUM are the outputs of the exact 4:2 compressor. C_{out}, CARRY and SUM are given as

$$COUT = A3(A1 \oplus A2) + A1(A1 \oplus A2) \quad (1)$$

$$CARRY = CIN\,(A1 \oplus A2 \oplus A3 \oplus A4) + A4(A1 \oplus A2 \oplus A3 \oplus A4) \quad (2)$$

$$SUM = CIN \oplus A1 \oplus A2 \oplus A3 \oplus A4 \quad (3)$$

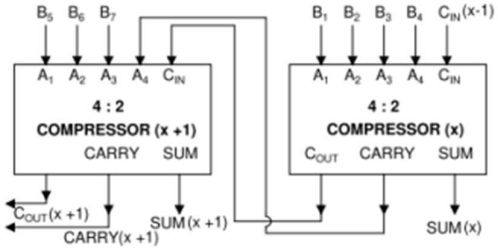


Figure 2. Compressor chain.

Figure 2 represents the compressor chain. Where C_{in} is the input carry from the previous 4:2 compressor that moved to lower significant bits CARRY and C_{out}.

PROPOSED METHODOLOGY:

The proposed system used the Drum 16 algorithm which will take 16 bit value (any format either binary, octal, decimal or hexadecimal) as input and produces 32 bit hex value as output. The drum 16 algorithm consists of a detector, a priority encoder, multiplexer and a Barrel shifter.

The functionality of the drum algorithm includes the first block that detects the input format weather the input format is in binary format or hex format or decimal format or octal format and then it will convert the input number to hexadecimal format.

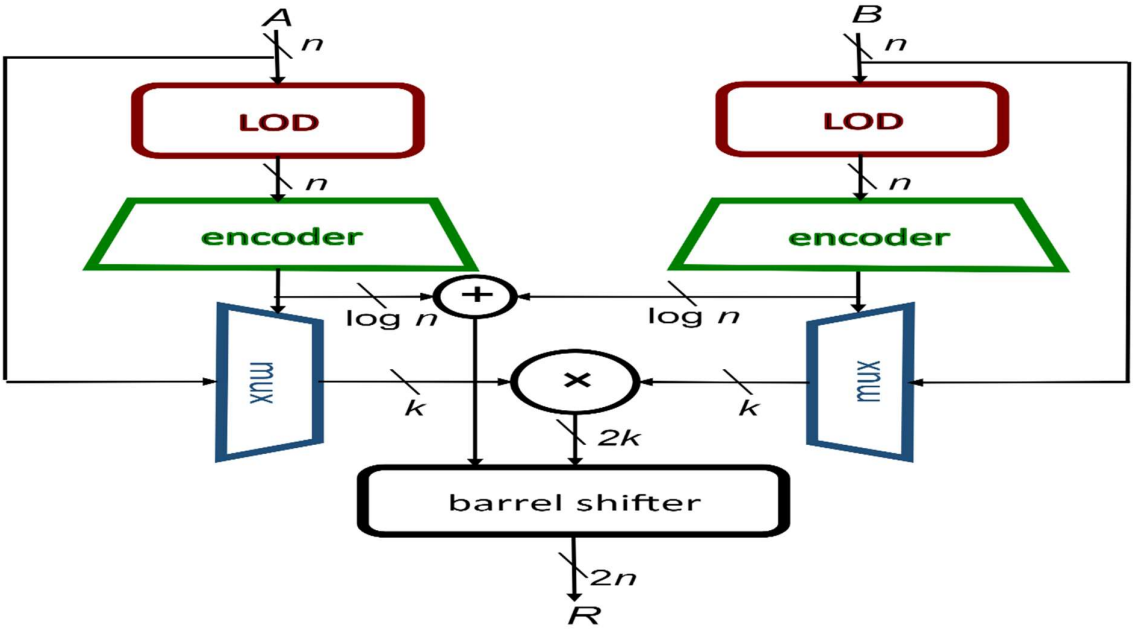


Fig: 3 DRUM based approximate multiplier

Priority encoder:

The priority encoder solves the problems according to the priority allocated to that priority encoder. The priority encoder output corresponds to the currently active input which has the highest priority. So when an input with a higher priority is present, all other inputs with a lower priority will be ignored. Based on the priority given to the encoder then the priority data can be passed first through the encoder.

The outputs of the priority encoder are then applied to the multiplexer selection inputs. Based on the selection inputs the multiplexer allows the data through it there are two 16 to 1 multiplexers are in the proposed system for a and b inputs respectively.

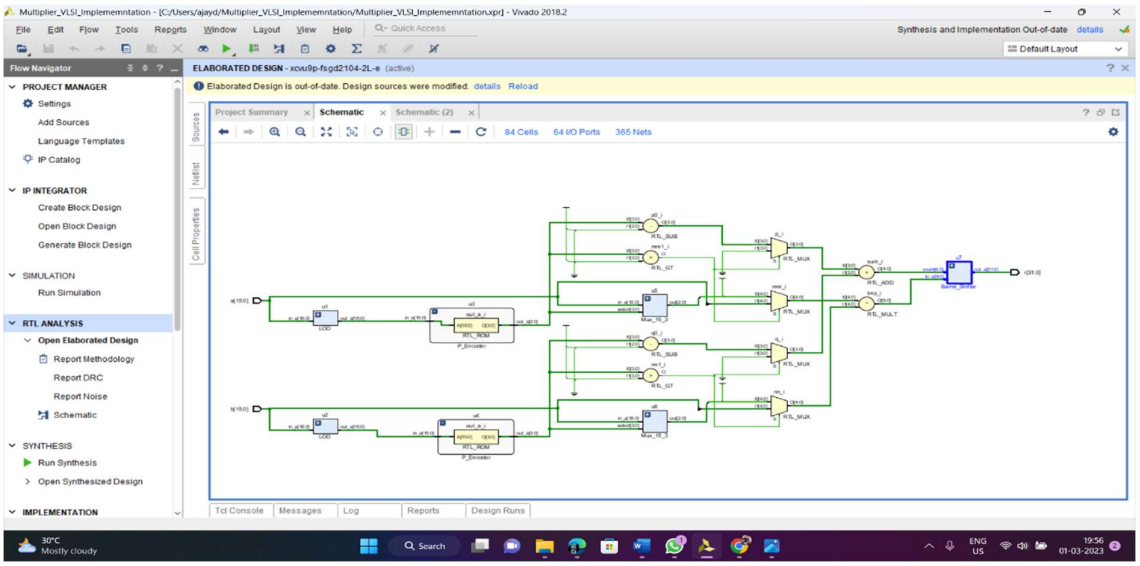
Then the multiplexed output is given to the barrel shifter barrel shifter that shifts the bits according to the multiplication process and priority bits this shifted bits are the finally combined and forms a 32 bit hexadecimal value by the hex – converter . This output is taken as the final output from the design.

The reason for the DRUM multiplier to be successful in comparison with other approximate multipliers in terms of error percentage which is decided by the truncation part design. In many

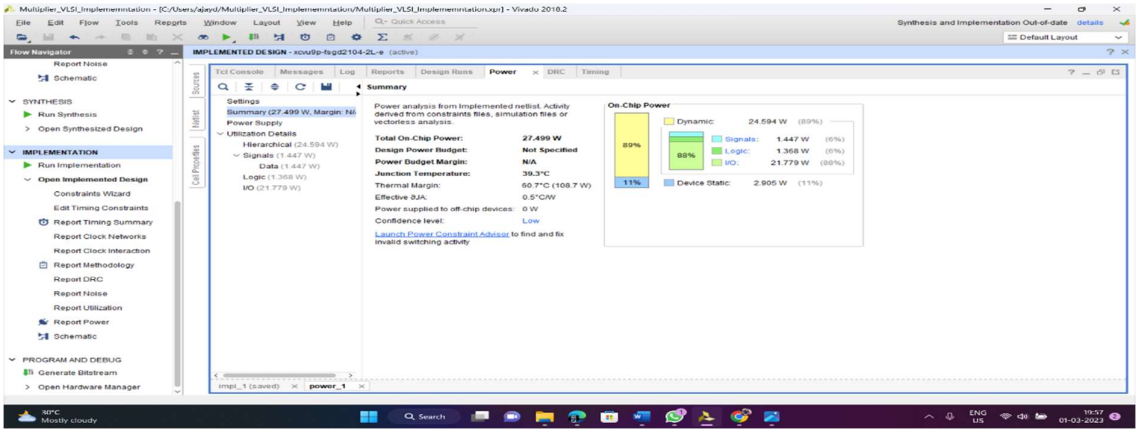
approximate multipliers after truncation, the last bit of the biased operand is retained as its original value but in the design of the DRUM multiplier the last bit is changed to the value of '1'.This technique reduces the error significantly from 1.86 % to 0.26%. The truncation technique is explained in Fig 1. After truncation of both the operands are multiplied using a small exact core multiplier and the result is obtained. This exact result is shifted and the shift is decided by the LOD block and the no of bits truncated in both the operands.

RESULTS

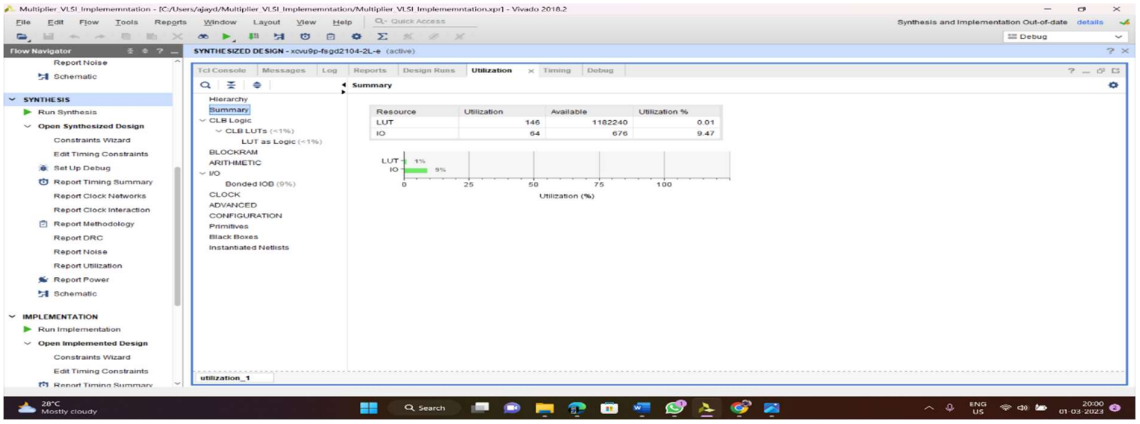
RTL schematic of the energy efficient approximate multiplier



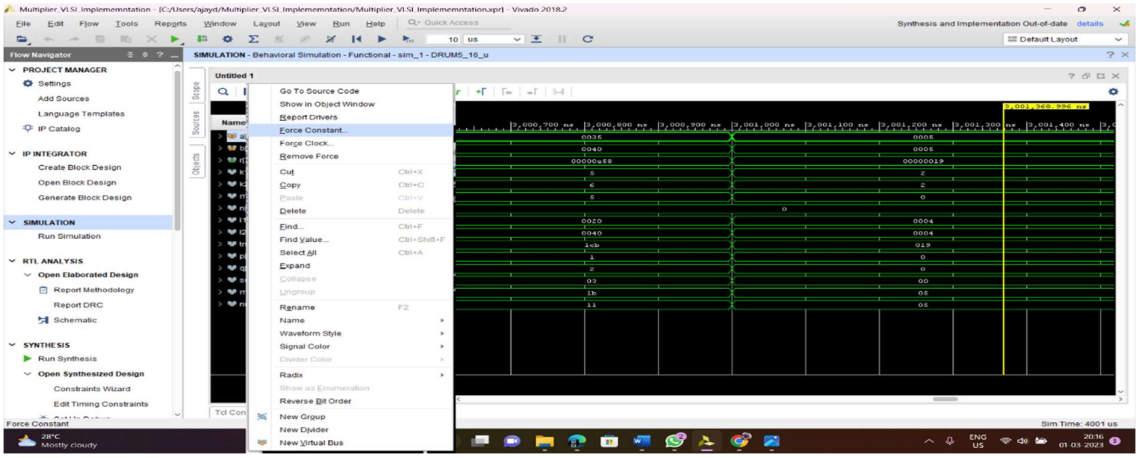
The power report shows that the power consumption of the proposed multiplier is 24.57W and total on chip power is 27.499W.



Lookup table (LUT) utilization obtained shows that 146 are utilized out of available 1182240 and I/O ports utilized are 64 out of 676.



Simulation Result



CONCLUSION

This paper presents approximate 4:2 compressor architectures. A high speed area efficient DRUM compressor architecture is proposed, which achieved a considerable reduction in area, delay and power when compared to other state-of-the-art compressor designs. The proposed design has comparable accuracy with 25% error rate and equal positive and negative absolute error deviation of 1. As a result, the proposed design reduces MED and MRED considerably without reducing the error rate. A novel approximate 4 -2 compressor design is presented in this paper. Firstly, a high speed area efficient compressor design is proposed, which attained a considerable reduction in area, delay and power when compared to other state-of-the-art approximate compressor designs. The proposed approximate multiplier design has accuracy with 25% error rate and equal positive and negative absolute error deviation of 1.The proposed approximate multiplier shows a significant improvement in terms of area and power consumption as compared to the existing approximate multiplier. This work concludes that this multiplier can be implemented for approximate computing by an approximate design of a compressor; this proposed multiplier offers advantages in terms of design parameters compared to existing approximate multipliers, and in terms of accuracy metrics, area and power consumption.

REFERENCES

[1] M. Alioto, "Ultra-low power VLSI circuit design demystified and explained: A tutorial," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 1, pp. 3–29, Jan. 2012.

[2] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[3] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of softcomputing applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[4] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in Proc. Int. Conf. Comput.-Aided Design, Nov. 2011, pp. 667–673.

[5] F. Farshchi, M. S. Abrishami, and S. M. Fakhraie, "New approximate multiplier for low power digital signal processing," in Proc. 17th Int. Symp. Comput. Archit. Digit. Syst. (CADS), Oct. 2013, pp. 25–30.

[6] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an underdesigned multiplier architecture," in Proc. 24th Int. Conf. VLSI Design, Jan. 2011, pp. 346–351.

[7] D. R. Kelly, B. J. Phillips, and S. AlSarawi, "Approximate signed binary integer multipliers for arithmetic data value speculation," in Proc. Conf. Design Archit. Signal Image Process., 2009, pp. 97–104.

[8] K. Y. Kyaw, W. L. Goh, and K. S. Yeo, "Low-power high-speed multiplier for error-tolerant application," in Proc. IEEE Int. Conf. Electron Devices SolidState Circuits (EDSSC), Dec. 2010, pp. 1–4.

[9] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.

[10] K. Bhardwaj and P. S. Mane, "ACMA: Accuracy-configurable multiplier architecture for error-resilient system-on-chip," in Proc. 8th Int. Workshop Reconfigurable Commun.- Centric Syst.-Chip, 2013, pp. 1–6.

[11] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power- and area-efficient approximate wallace tree multiplier for error-resilient systems," in Proc. 15th Int. Symp. Quality Electron. Design (ISQED), 2014, pp. 263–269.

[12] J. N. Mitchell, "Computer multiplication and division using binary logarithms," IRE Trans. Electron. Comput., vol. EC-11, no. 4, pp. 512–517, Aug. 1962.

[13] V. Mahalingam and N. Ranganathan, "Improving accuracy in Mitchell's logarithmic multiplication using operand decomposition," IEEE Trans. Comput., vol. 55, no. 12, pp. 1523–1535, Dec. 2006.

[14] Nangate 45nm Open Cell Library, accessed on 2010. [Online]. Available: http://www.nangate.com/

[15] H. R. Myler and A. R. Weeks, The Pocket Handbook of Image Processing Algorithms in C. Englewood Cliffs, NJ, USA: Prentice-Hall, 2009.