IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

## IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 10, October 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

9940 572 462 | 6381 907 438 | ijircce@gmail.com | www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110013 |

# Design and Implementation of Decoder Based Approximate Multiplier for Low Power and Low Area Applications

[1]Mrs.M.SUGANYA, [2] Ms.P.MENAGA,

Assistant Professor, Department of ECE, Gnanamani College of Technology, Namakkal(D.T), Tamilnadu, India[1]

PG Student, Department of ECE, Gnanamani College of Technology, Namakkal(D.T), Tamilnadu, India [2]

**ABSTRACT:** Approximate computing is a promising method for designing power-efficient computing systems. Many image and compression algorithms are inherently error-tolerant and can allow errors up to a specific limit. In such algorithms, savings in power can be achieved by approximating the data path units, such as a multiplier. A novel decoder logic- based multiplier design with the intent to reduce the partial products generated. Thus, leading to a reduction in the hardware complexity and power consumption while maintaining a low error rate. Our proposed design which achieves power reduction compared to the accurate and approximate multipliers. It occupies lower die-area and consumes less power while resulting in fewer erroneous results in comparison to the existing approximate multipliers.

## I. INTRODUCTION

Multipliers form an important hardware block in the DSP and Embedded applications. Multiplication determines processor speed. So high speed multipliers are needed in the processors for many applications. For increase the speed of multiplication different algorithms are used Multiplication is a most commonly used operation in many computing systems. A number (multiplicand) is added to itself a number of times as specified by another number (multiplier) to form a result (product). But the implementation of multiplier takes huge hardware resources and the circuit operates at low speed. Multiplication is one of the fundamental components in DSP and Embedded system. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the real time system. Multiplier requires more hardware resources than the adder and subtractors. For Improving the performance and reducing the power dissipation of the systems are the most important design challenges for Embedded and DSP applications. Increasing the word length results in hardware complexity and also increases the multiplication time. Many algorithm have been developed in order to realize high speed multipliers such as Booths algorithm, Wallace tree algorithm etc. Multipliers based on Booth algorithm and Wallace tree addition is one of the fast and low power multiplier. Multiplication consists of three major steps: 1) re-coding and generating partial products 2) reducing the partial products by partial product reduction schemes to two rows and 3) adding the remaining two rows of partial products by using a carry-propagate adder (e.g. Carry look ahead adder) to obtain the final product. In applications like multimedia signal processing and data mining which can tolerate error, exact computing units are not always necessary. They can be replaced with their approximate counterparts.

Then a constant or variable correction term is added to compensate for the quantization error introduced by the truncated part. Approximation techniques in multipliers focus on accumulation of partial products, which is crucial in terms of power consumption. Broken array multiplier, where the least significant bits of inputs are truncated, while forming partial products to reduce hardware complexity. The proposed multiplier saves few adder circuits in partial product accumulation.

The Key design objective of any modern-day digital system is to remain power-efficient with an improvement in performance. Approximate computing has emerged as a promising technique capitalizing on the inherently error resilient image and video processing applications to tolerate some loss of accuracy. The fundamental arithmetic modules in these applications are multipliers; numerous techniques to analyze and evaluate approximate multipliers have been extensively studied.

Approximate multipliers have been primarily developed using two methods: 1) approximation at partial product

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110013 |

reduction (PPR) stage and 2) approximation at partial product generation (PPG) stage. The least significant portion of the partial product (PP) rows are truncated and the error is compensated using a correction function or larger input approximate compressors are used to accumulate the PP rows. This method results in a minor reduction of the resource utilization because the number of PP rows remains unchanged. On the other hand, in, the number of PP rows generated could be drastically reduced by employing an approximate circuit. However, existing designs based on aim to improve latency and reduce error in the MSB section of the output by splitting the PPG into smaller inaccurate sub-blocks, utilizing an accumulation stage within each sub-block.
Unlike some existing designs based on, in this letter, we aim to reduce the number of PP rows generated by designing an approximate decoder logic block without utilizing an internal accumulation stage. Thus, our proposed design occupies lower die-area and consumes less power while resulting in fewer erroneous results in comparison to the existing approximate multipliers.

**OVERVIEW OF 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. Multiplier design. should offer high speed, low power consumption.
Multiplication involves mainly 3 steps

1) Partial product generation

2) Partial product reduction

3) Final addition

**PARTIAL PRODUCT GENERATORS (PPG)**

A binary multiplier is an electronic circuit used in digital electronics, such as a computer, to multiply two binary numbers. It is built using adders.
A variety of computer arithmetic techniques can be used to implement a digital multiplier. Most techniques involve computing a set of partial products, and then summing the partial products together. This process is similar to the method taught to primary schoolchildren for conducting long multiplication on base-10 integers, but has been modified here for application to a base-2 (binary) numeral system**.**

**Multiplication Algorithm**

1) If the LSB of Multiplier is 1, then add the multiplicand into an accumulator.

2) Shift the multiplier one bit to the right and multiplicand one bit to the left.

3) Stop when all bits of the multiplier are zero.

For designing a multiplier circuit we should have circuitry to provide or do the following three things:

1) It should be capable of identifying whether a bit 0 or 1 is.

2) It should be capable of shifting left partial products.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110013 |

**Array multiplier:**

Array multiplier is an efficient layout of a combinational multiplier. Multiplication of two binary number can be obtained with one micro-operation by using a combinational circuit that forms the product bit all at once thus making it a fast way of multiplying two numbers since only delay is the time for the signals to propagate through the gates that forms the multiplication array. In array multiplier, consider two binary numbers A and B, of m and n bits.
There are mn summands that are produced in parallel by a set of mn AND gates. n x n multiplier requires n (n-2) full adders, n half-adders and $n2$ AND gates. Also, in array multiplier worst case delay would be (2n+1) td. Array Multiplier gives more power consumption as well as optimum number of components required, but delay for this multiplier is larger. It also requires larger number of gates because of which area is also increased; due to this array multiplier is less economical. Thus, it is a fast multiplier but hardware complexity is high.

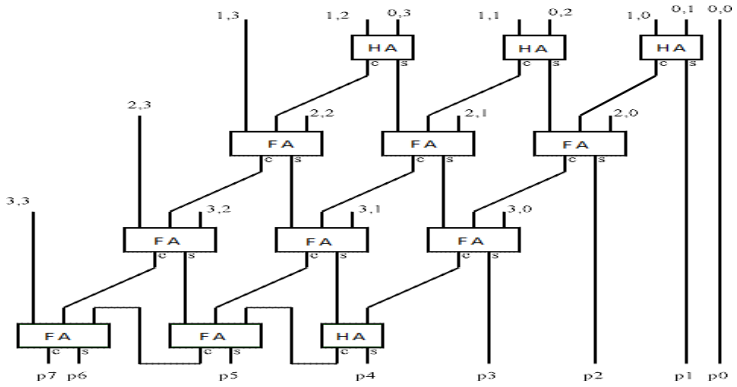


**FIG: ARRAY MULTIPLIER USING HALF ADDER AND FULL ADDER**

In this technique, a range of implementations of common arithmetic units. In the case of adders, which are the most fundamental building block, we evaluate the behavior of three different conventional architectures under voltage over-scaling
- ripple carry adder, carry look-ahead adder, and Han-Carlson adder. We also evaluate three architectures that are specifically designed for approximate computing, based on dynamic segmentation with error compensation, reverse carry propagation, and truth table based approximation. Our results reveal interesting insights into the behavior of these approximate arithmetic units and the trade-offs between them. For example, our analysis identifies that functional correlation between the output bits reduces the maximum error introduced by over-scaling, explaining the empirical observation that the average case error in practice is much smaller than the worst-case. For some designs, we also observe that voltage over-scaling introduces errors of certain specific values much more frequently than others, leading to the possibility of low-complexity error detection and correction. We believe that systematic analysis frameworks such as the one proposed in this paper can significantly facilitate the adoption of approximate computing techniques.

## II. EXISTING SYSTEM

To reduce the computation complexity, various approximate multipliers have been proposed based on methods 1) and 2). Based on 1), a broken-array multiplier was suggested by pruning the carry-save adders that were used at accumulation stage. This method achieved saving in area by eliminating either horizontal partial-product rows or vertical partial-product columns. An approximate 4-2 compressor with intent to achieve a low error rate was proposed. This method tries to exploit the probabilistic characteristic of AND gate output which is used in the PPG stage at the PPR stage. Ha and Lee modified the compressor proposed and improves the accuracy by adding a simple error correction circuit in the PPR stage. To achieve low area and low power, proposed an efficient compressor.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110013 |

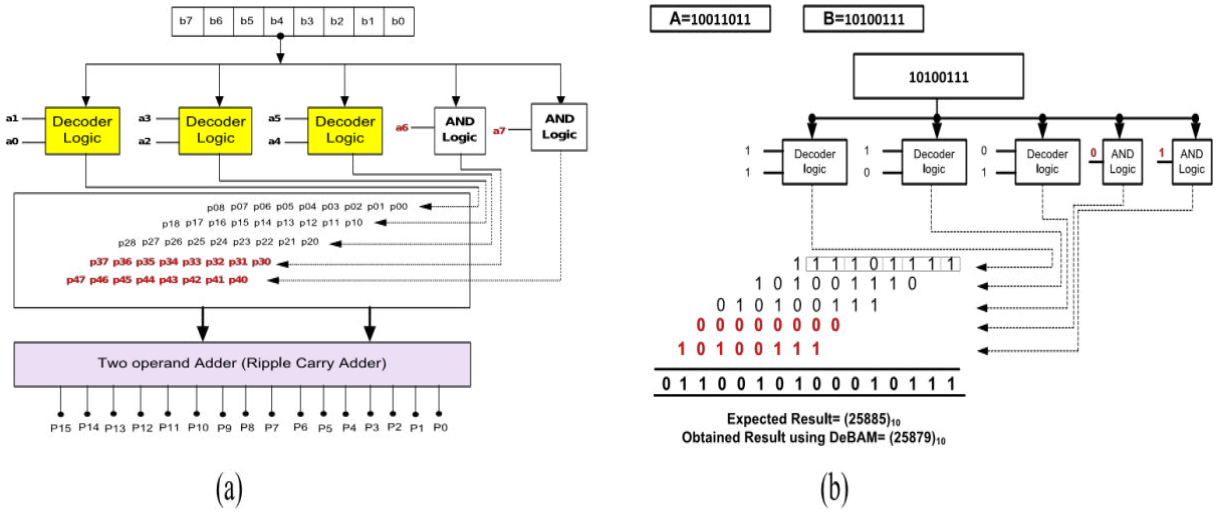


**FIG:EXISTING SYSTEM OF ARRAY MULTIPLIER**

Proposed the compressor-based approximate multiplier with intent to reduce the error rate. Based on 2), Bhardwaj presented an approximate Wallace tree multiplier in mode 4 (AWTM-4) with a carry-in prediction for building recursive blocks. Accurate multiplier modules are used for MSB computation while in-exact blocks were used for LSB portion. Approximate inputs were generated using a fast bit selection scheme for error resilient applications to achieve low-power

## III. PROPOSED WORK

The proposed DeBAM aims to utilize a block configuration that encapsulates a unique 2-bit decoder logic circuit, used to simplify the generation of PPs. The novelty of this letter is the method employed and the circuit used to generate PPs. The PPs generated by the decoder logic are compressed to two rows using accumulator and final reduction is achieved using two-operand adder. For the sake of simplicity, the proposed unsigned multiplier is explained using two 8-bit operands and the same can be extended to a higher number of bits as illustrated later. Consider the multiplication of two numbers A and B each of 8-bit. All the bits of A (multiplier) can be multiplied with B (multiplicand) in a conventional manner using AND gate to generate PPs. However, it will result in 8 PP rows with each consisting of 8 bits. Reducing the number ofPP rows will result in simple hardware at the PPR stage. To accomplish this, we propose DeBAM architecture illustrated, which uses novel decoder logic blocks to generate approximate PP for the lower significant multiplier bits (a0a1, a2a3,a4a5) while the most significant a7&a6 bits generate exact PPs. The multiplier A, that forms the selection input to decoder logic, is divided into three groups (a0a1, a2a3, a4a5), each of 2-bit to generate approximate PP while corresponding to a7&a6 exact PP are generated using conventional AND gate logic, thereby limiting the error to the least significant portion.
Further, choosing two bits for select logic restricts the possible bit combinations of each pair to four unique cases.

The functionality of the decoder logic block is based on the select inputs (ax ax + 1) as illustrated in Table I. Since the select input to the decoder is 2-bits there can be four possible combinations as shown in the table. For example, if the multiplier bits are “01” then the decoder block passes the multiplicand (B) as output. The only case when approximate PP are generated by respective decoder blocks is when the select inputs ax ax+1 are “11.” To obtain PP for this case the multiplicand B and left- shifted B (B<<1) has to be added. This addition is carried out using OR logic. It can be noted that the output of each decoder

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110013 |

logic block is arranged based on the positional weight of the multiplier bits. The accuracy of the proposed multiplier is enhanced by generating the exact PP using MSB bits (a7a6) and multiplicand (B) using traditional AND logic.
Based on the select inputs to the proposed decoder logic blocks (multiplicand as other input) approximate PP is generated while the most significant (a7a6) bits generate exact PPs. The PPG stage is followed by an accumulation stage, carried out using 3:2 carry-save adders, that reduces the 5 PP rows into two. The generation of minimal PP leads to minimization in the number of adders in the accumulation stage, leading to a significant reduction in area and power. The two rows obtained after accumulation stage are subsequently reduced to the final result (p15-p0) using a ripple carry adder.

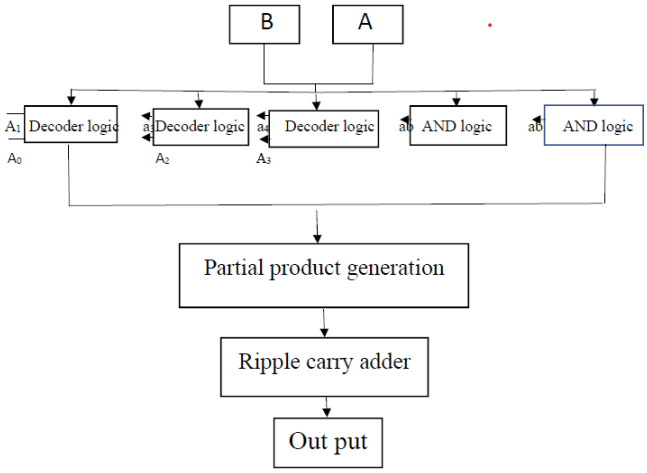


**FIG:BLOCK DIAGRAM OF PROPOSED DE-BAM**

The DeBAM with the help of a numerical example. Based on the input select bits, the three decoder blocks generate three rows of PP while the remaining two rows are generated using AND gate logic. These rows are reduced to the final product using a reduction structure. It can be observed that the result obtained using DeBAM is almost equal to an accurate result.

| ax+1 | ax | Decoder output |
|---|---|---|
| 0 | 0 | Zero |
| 0 | 1 | b7-b0 |
| 1 | 0 | (b7-b0) <<1 |
| 1 | 1 | (b7-b0) OR ((b7-b0) <<1) |

**Gate-Level Implementation of Decoder Logic Block**
This section presents the circuit-level implementation of the proposed decoder based logic for an 8 ∗ 8 multiplier. When ax ax + 1 are "00," the decoder logic block generates zero as the output. For the case "01," the value of b7−b0 form the PP while for "10" the resulting PP is (b7−b0) left-shifted by "1." It is easy to see that the combinational circuit for implementing these three cases requires minimal number of gates as it involves either conditional transfer of input bits or zero. The final case where ax ax + 1 are "11" can be evaluated as the sum of output PPG for the case ax ax+1 is 01 and 10. The adder logic, in this case, has been implemented using a simple OR gate .

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110013 |

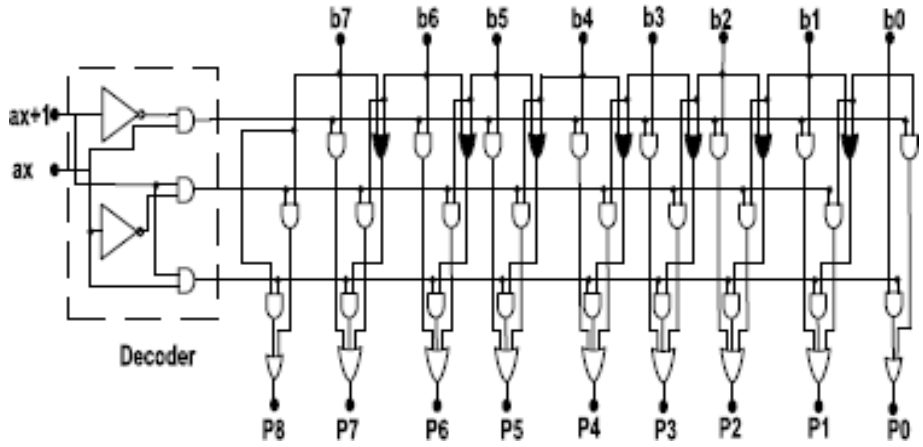


**FIG:CIRCUIT IMPLEMENTATION OF PROPOSED DECODER**

Then select the finish option, choose the bit file program. (That bit file adds to the device). Finally right click the device, select program -> click ok (it'll fetch to the FPGA kit through the port.

## IV. SOFTWARE SYNTHESIS RESULTS AND DISCUSSIONS

Synthesis results of various approximate multipliers. Software results prove the power and area efficacy of the proposed multiplier compared to the existing designs .Finally ,the proposed method achieves a better computation quality effort tradeoff and the same is demonstrated on the image sharpening and comparison algorithms.

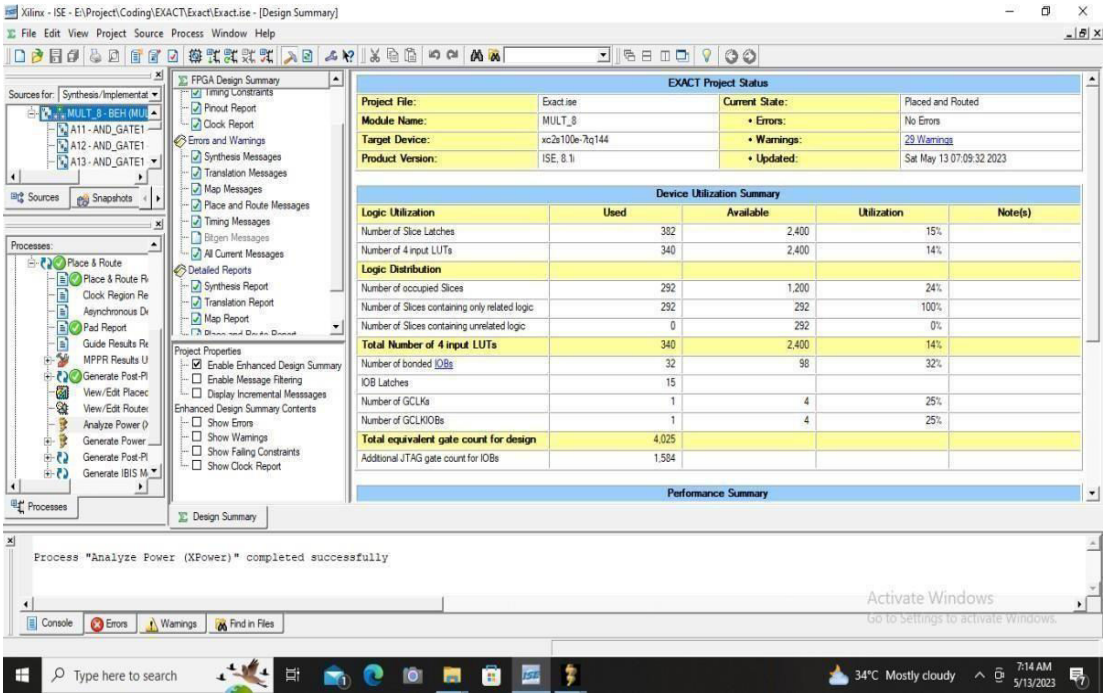


**FIG:SIMULATION RESULT FOR AREA CONSUMPTION**

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110013 |

**DEALING WITH COMPLEXITY**

Divide-and-conquer: limit the number of components you deal with at any one time. Group several components into larger components:

- transistors form gates;
- gates form functional units;
- Functional units form processing elements, etc.

**TOP-DOWN VS. BOTTOM-UP DESIGN**

- Top-down design adds functional detail.

Create lower levels of abstraction from upper levels.

- Bottom-up design creates abstractions from low-level behavior.
- Good design needs both top-down and bottom-up efforts.

**DESIGN STRATEGIES**

IC design productivity depends on the efficiency with which the design may be converted from concept to architecture, to logic and memory, to circuit and hence to a physical layout.

A good design strategy with a good design system should provide for consistent descriptions in various abstraction levels.

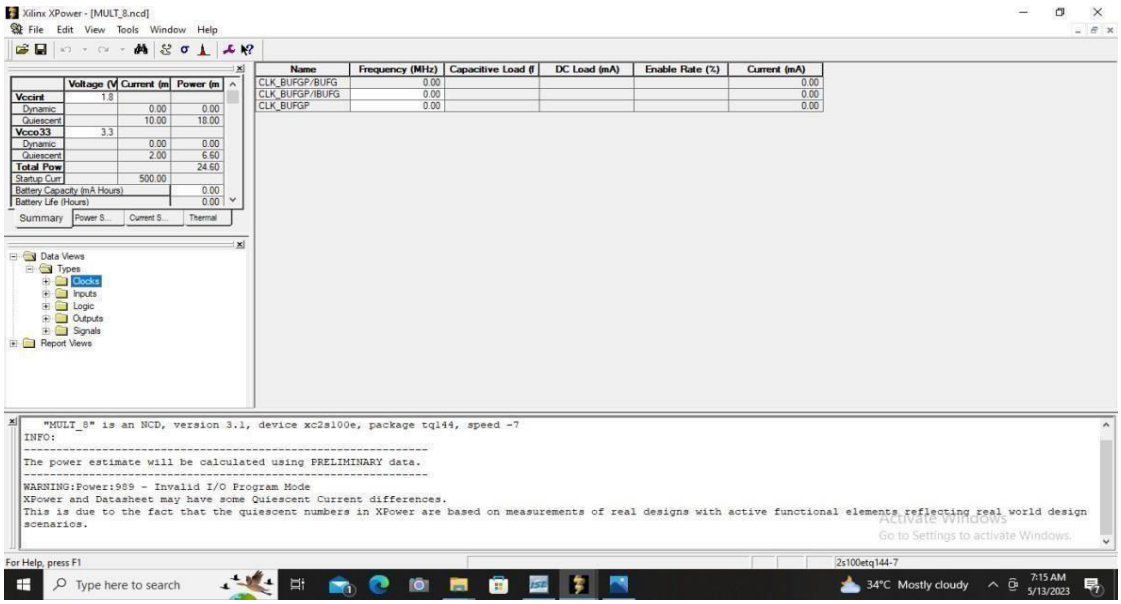


**FIG:SIMULATION RESULT FOR POWER CONSUMPTION**

**WORKING PRODUCT**

Design is a continuous trade-off to achieve adequate results for:

- Performance - speed, power, function, flexibility
- Size of die (hence cost of die)
- Time to design
- Ease of test generation and testability

. Three approximate multipliers were proposed. The proposed approximate multiplier design uses first approximate compressor in all four partial products. It requires less area and power than the existing multipliers. Its accuracy is less compared to proposed approximate multiplier design. The proposed approximatemultiplier design uses accurate compressors in three most significant partial products and first approximate compressor in least significant product. It requires less power compared to existing multipliers. Finally the area and power is obtained as 4.025 and 24.60

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110013 |

respectively Its accuracy is higher than the existing multipliers.
Depending on the application requirements one can select the proposed approximate multiplier design or multiplier design . The proposed approximate multiplier design is very much suitable for high accuracy applications. The proposed multiplier design uses second approximate compressor in all the four partial products. Its accuracy is high compared to the proposed multiplier approximate multiplier design.

## V. CONCLUSION

Approximate computing offers potential benefits in terms of area, delay, and power. This letter attempts to leverage the inherent error-resilient characteristic in the image processing applications. The proposed DeBAM (N,M) architecture reduces the number of PPs generated, as a result, also simplifies the PP accumulation stage. Error and hardware results prove the energy and area efficacy of the proposed multiplier compared to the existing designs. Finally, the proposed method achieves a better computation quality-effort tradeoff and the same is demonstrated on the image sharpening and compression algorithms.

## VI. FUTURE ENHANCEMENT

It can be applicable to implement in hardware and also it can be used for image processing and biomedical applications. Similarly, 32 bit multiplier can also be implemented and the power will be reduced more compared to 16-bit multiplier,8 bit multiplier and 4 bit mutlipliers. And these will be implemented in hardware SPARTAN 3E kit.

## REFERENCES

[1] S. Mittal, “A survey of techniques for approximate computing,” ACM Comput.Surveys, vol. 48, no. 4, p. 62, 2016.
[2] 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, 2014.
[3] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, “A review, classification,and comparative evaluation of approximate arithmetic circuits,”ACM J. Emerg. Technol. Comput. Syst., vol. 13, no. 4, p. 60, 2017.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110013 |

[4] B. Parhami, Computer Arithmetic, vol. 20. Oxford, U.K.: Oxford Univ.Press, 2010.
[5] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol. 62, no. 9, pp. 1760–1771, Sep. 2013.
[6] X. Yi, H. Pei, Z. Zhang, H. Zhou, and Y. He, "Design of an energy efficient approximate compressor for error-resilient multiplications," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), 2019, pp. 1–5.
[7] 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. Qual. Electron. Design, 2014, pp. 263–269.
[8] M. Ha and S. Lee, "Multipliers with approximate 4–2 compressors and error recovery modules," IEEE Embedded Syst. Lett., vol. 10, 2020, no. 1,pp. 6–9.
[9] Z. Yang, J. Han, and F. Lombardi, "Approximate compressors for error resilient multiplier design," in Proc. IEEE Int. Symp. DFTS, 2015, pp. 183–186.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 || A Monthly Peer Reviewed & Referred Journal |

|| Volume 11, Issue 10, October 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1110013 |

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.379

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

## IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details