IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 5, May 2024

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

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

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

DeBAM: Decoder-Based Approximate Multiplier For Low Power Application

Suresh G, Prabhakaran G

PG Scholar, Department of Electronics and Communication Engineering, Nandha Engineering College, Erode, Tamil Nadu, India

Assistant Professor, Department of Electronics and Communication Engineering, Nandha Engineering College, Erode,Tamil Nadu, India

ABSTRACT: The development of energy-efficient computer systems utilizes approximate computation, an exceptionally efficient methodology. There are numerous methods for image and compression that can handle faults. These solutions could save power by using approximate data routing units, such as multipliers. To limit the number of incomplete items generated, the multiplier's design includes decode logic. As a result, hardware becomes simpler and requires less power. When we compare our technique to accurate and approximation multipliers, we find that power usage is significantly reduced. Compared to other approximation multipliers, our technique consumes less power and dies area while delivering a lower average error rate.

KEYWORDS: DeBAM, Low Power Consumption,Approximate Computing, Multiplier Design.

I. INTRODUCTION

Digital signal processing and embedded systems rely heavily on the multiplier as a hardware component. High-rate multiplication operations rely on a computer's processing power. Effective multipliers are required for a variety of tasks. Many techniques speed up the multiplication process. Numerous computer systems perform multiplication operations on a regular basis. Assigning the multiplier the initial value, the multiplicand denotes the quantity to be multiplied. The product is the result of multiplying two integers together. A substantial amount of hardware resources are required for the device to operate as a multiplier, however, and it operates slowly. Digital signal processing and integrated devices employ multiplication techniques. Within a real-time system, the multiplier component operates at a glacial pace, so its efficiency directly influences the system's overall operation. Matrices demand a greater allocation of hardware resources in comparison to subtractors and adders.

Concurrently reducing power dissipation and enhancing system efficiency is one of the most difficult challenges in embedded and DSP architecture. As the word length increases, the hardware processing speed decreases, and the multiplication rate decreases. The development of high-speed amplifiers has employed various techniques, including the Wallace tree algorithm and the Booths algorithm. Combining Wallace tree addition with the Booth method produces a low-power, highly efficient multiplier. Creating and recording fragmentary products are the three primary phases that comprise the multiplication procedure. Dividing partial products into two columns is one method for reducing them. Meanwhile, transport the unfinished objects along the remaining two aisles.

There are two situations in which high-precision computers are unnecessary and can function with imperfections. We can use an analogous substitute instead. Programming that can function despite errors is becoming an increasing component of approximate computation. The most critical components of these particular systems are multiplexers and adders. Specifically designed for implementation in digital signal processing (DSP), we design and produce practically entire transistor-level adders. Using the recommended complete adders, we mutually add the partial products of multipliers.

As a means of enhancing the readability of the hardware, fixed-width multiplier designs often incorporate truncation. We insert fixed or variable correction terms to offset the quantization error the truncated segment introduces. When it comes to power consumption, a significant proportion of multiplier approximation methodologies rely on the overall quantity of fragmentary products. It performs partial products and truncates the least significant bits of the inputs. As a result, the hardware becomes less complicated. Implementing the proposed multiplier for partial product accumulation leaves a relatively small number of adder circuits unchanged.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

Contemporary design principles for digital systems prioritize the optimization of performance and power consumption. Approximation computation is a novel method for effectively regulating a degree of precision loss. This is due to the error-resistant nature of algorithms designed specifically for processing images and videos. In numerous situations, multipliers serve as foundational mathematical components. We have examined an extensive range of methodologies to assess approximation factors.

The two phases of approximation, partial product reduction (PPR) and partial product generation (PPG), comprise the two primary methods used to derive the estimated multipliers. We use correction functions or compressors with bigger input approximations to make the decreased partial product (PP) rows less important. The error was subsequently corrected. This strategy marginally reduces resource utilization while maintaining a constant number of PP rows. Thus, implementing an estimating circuit can significantly reduce the number of PP rows generated. Modern methods, on the other hand, use a consolidation mechanism within each sub block of pulse position modulation (PPM) to cut down on latency and improve the accuracy of the most significant bit (MSB) generation, even though they do this by breaking the subblocks up into smaller ones that are less precise. The goal of this correspondence is to make approximate decoder logic block that, unlike other designs, doesn't need an internal collecting phase. This will cut down on the number of PP rows. In comparison to the current estimated multipliers, our approach yields a lower number of false positives while consuming less power and die space.

For the majority of data processing systems, multiplication is a fundamental operation. In addition to being enormous in size and power consumption, multipliers exhibit considerable delay. We must create low-power multipliers to facilitate the development of low-power VLSI systems. Due to its typically sluggish speed and ample space requirements, the multiplier's efficacy is vital for the entire system. Optimization of the multiplier's area and efficiency is a critical design challenge. Area and velocity are generally not synonymous. In general, larger regions correspond to higher velocities. The multiplication operation multiplies an integer by itself a predetermined number of times. The mathematical term multiplier refers to the compound products, or outcomes, of two integer operations. Multiplication begins with the multiplicand, which is an integer. One of the most contemporary uses for multipliers is in digital signal processing. With minimal power consumption, a repeater ought to operate rapidly. The unexpected 2 x 2 Partial Product Generator (PPG) 16 x 16 multiplier guarantees the minimum and maximum precision specified during the design process. PPGs compromise accuracy in exchange for reduced leakage and dynamic energy, as they employ fewer transistors than ideal 2x2 ones. On the basis of Leading One Detectors (LODs), we implement precision modifications in a novel iterative logarithmic estimate multiplier using Leading One Detectors (LODs). Digital devices such as computers use binary multipliers, which are electrical circuits that multiply two binary integers. The assembly process utilizes adders.

There are numerous computational mathematics-based approaches to constructing a digital repeater. Most approaches begin by identifying a series of incomplete products, which they then combine. To accommodate the binary numerical system, we adjusted this methodology. The approach bears a resemblance to the way in which young children in primary education examine long division using whole numbers and the decimal system.

If the multiplier's most significant number is 1, store the multiplicand in an accumulator. Initiate a rightward shift of an equivalent magnitude and augment the multiplier with a minimum value. If every element of the multiplier is zero, give up.

Three types of circuitry are required for a multiplier circuit: Determining whether a bit signifies a 0 or a 1 necessitates prior knowledge. Additionally, move any unfinished business to the left. The array multiplier efficiently facilitates the construction of combinational multipliers employing a combinational circuit capable of simultaneously generating the product bit and multiplying two binary integers in a single micro-operation. This solution is extremely efficient, as it requires no more time than the duration required for signals to traverse the gates comprising the multiplication array. Check the array multiplier's two binary values. B and A, respectively, consist of n and m bits. A collection of M_n AND gates can generate M_n summands concurrently. Two AND gates, n half-adders, and n (n-2) complete adders are necessary to implement a multiplier with dimensions of n x n. It is worth mentioning that the array multiplier is capable of producing a maximal delay of (2n+1) t_d. The array multiplier produces a sufficient quantity of components despite its increased power consumption and prolonged execution time. Moreover, the array multiplier's heightened power and physical space requirements limit its practicality. The apparatus operates as a rapid signal amplifier despite incorporating state-of-the-art technology.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

II.LITERATURE REVIEW

Using a simple and efficient estimation adder, it is possible to build an innovative approximate multiplier configuration. By interfering with the carry propagation sequence, this cutting-edge adder can process multiple data inputs at once. Nonetheless, some mistakes have occurred here. When compared to a traditional one-bit full adder, the critical route takes less time. An approximation adder, acting as a partial summation device, adds the products in a simple hierarchical structure. Furthermore, we use error signals to adjust for potential inaccuracies, thereby improving accuracy. A person aspires to improve their precision. Some argue that the multiplier is far less exact in terms of critical routes and complex circuits than the Dada and Wallace trees.

It is critical to remain focused on the need to improve energy efficiency. Fixed-point arithmetic has a high tolerance for processing errors due to its widespread use in a variety of digital settings. We introduce a multiplier in this paper to optimize the trade-off between energy usage and performance. Bit-width reduction, forced voltage manipulation, and defective component estimates for multiplication are examples of alternative methods. In contrast, this strategy approximates multiplication by using a large number of operands.

A unique and cutting-edge Dynamic Range Unbiased Multiplier (DRUM) built exclusively for approximation purposes. The multiplier's dynamic architecture improves its capacity to magnify the most significant digits in a number, increasing its adaptability and scalability. Our multiplier's impartial design results in an almost zero average error rate, as numerous calculations demand repeated multiplications. Excellent work. We have implemented measures to ensure the balance of errors and prevent their accumulation from influencing the final conclusion, thereby promoting fairness and impartiality. Because something is true, it must be so. Our design's high degree of adjustability allows us to create a seamless trade-off between power consumption and the level of precision sacrificed, depending on the designer's individual requirements. Notably, the magnitude of the core multiplier in our approximation method is smaller than the entire multiplier. As a result, designers can keep the multiplier design they choose for the primary component while still using traditional design methodologies.

Common mathematical units have applications in a variety of areas. We investigate the voltage over scaling performance of three unique traditional designs used as essential building blocks, namely adders. We classify adders into three types: Han-Carlson adders, ripple carry adders, and carry look-ahead adders. This paper also investigates three computational approximations. The core concepts of these systems include reverse carry propagation, truth table-based approximation, and dynamic segmentation with error correction. We provide new insights into the operation of these approximation arithmetic units, as well as their advantages and downsides. The actual results through our methodology, which shows how a functional correlation between the output bits reduces the maximum error due to excessive scaling. They established that the empirical mean error was significantly less than the maximum error. Furthermore, it is clear that different designs exhibit significant variance in the frequency with which specific value errors due to voltage over-scale occur. If this approach proves successful, it will facilitate the discovery and correction of errors with ease. This and subsequent research suggest an architecture that has the potential to significantly simplify the implementation of approximation computing techniques.

Adding wrong or approximation FA cells lets you make multi-bit arithmetic units that get the most out of quality, space, and power use in digital signal processing (DSP) systems that can handle errors. Our solution reduces the number of transistors and load capacitances in a standard mirror adder cell, simplifying its construction. When they are obvious, approximation-related errors have little impact on the output quality of a traditional DSP process. Our technique fundamentally differs from previous approaches hindered by VOS-related inaccuracies. When the effective switching capacitance decreases. The proposed FA cells allow the system to run at lower supply voltages. This is a comparison to a typical occurrence. An extra-quadratic factor reduces power loss. Created a number of approximate multipliers to reduce the computational complexity. Essentially, we construct these multipliers by applying techniques 1) and 2) sequentially. To create a broken-array multiplier, we reduced the number of carry-save adders used in the accumulation stage. We had to address the concerns of all parties involved. This solution reduced the vertical and horizontal partial-product columns, lowering the amount of space required. To achieve the lowest error rate, utilize a compressor with a compression ratio of around 4:2. At the PPG stage of the PPR stage, this technique leverages the probabilistic feature of the AND gate output to its full potential. The precision of the previously designed compressor by incorporating a basic error-correcting circuit into the PPR stage. We advised using an efficient compressor to achieve a small footprint and low electricity use. A compressor-based approximation multiplier was one suggested method for obtaining the desired output with the least amount of error margin. In mode 4 (AWTM-4), Bhardwaj introduced the concept of an approximation Wallace tree multiplier for constructing recursive blocks. This multiplier is

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

based on the second number. The MSB computation used correct multiplier modules, while the LSB component used exact blocks. We generated approximation inputs using a quick bit selection method to reduce power consumption for error-tolerant applications.

III. PROPOSED METHODOLOGY

The proposed DeBAM system's primary goal is to increase PP production efficiency by incorporating a specialized 2-bit encoder logic circuit into a block design. The DEBAM has committed to this goal. The PPs referenced in this email originate from one of the early circuits. The accumulator, which condenses the encoder logic's PPs into two rows, completes the reduction process. The two-operand adder eventually completes the reduction procedure. We demonstrate the proposed unsigned multiplier using two operands, each containing sixteen bits. We have completed this job with the highest level of simplification possible. We will provide more information on raising the bit count later. To proceed, multiplication of the 16-bit values A and B is required. The typical method for computing parallel products is to utilize an AND gate to perform bitwise multiplications on each bit of multiplier A and the matching bit of multiplicand B. In contrast, this means that each row will have sixteen bits available for use in total. Reducing the number of PP sections on the PPR stage will limit the number of gear options. To achieve this goal, we recommend implementing the DeBAM architecture. The system uses separate decoder logic blocks to estimate likely values for the least significant multiplier bits (a0a1, a2a3, a4a5, a6a7, a8a9, a10a11, and a12a13), while it produces exact probable values for the most significant bits (a14 and a15). The system manages the decoder logic through the seven-group multiplier A, which consists of bit pairs (a0a1, a2a3, a4a5, a6a7, a8a9, a10a11, and a12a13). Each set approximates the prediction point (PP) using two bits, despite the AND gate logic reducing error and identifying the precise PP for a14 and a15. Choice logic restricts the connections between two bits to four options, necessitating the construction of each pair separately.

Table I clearly shows that the pick inputs (ax, ax+1) have a direct impact on how the decoding logic unit functions. The table shows that the decoder accepts a two-bit choice input. There are four possible outcomes that could occur. If we set the multiplier bits to "01," the encoder block might output the multiplicand (B). For the decoder blocks to provide an estimated PP, they must set both of the chosen inputs, ax and ax+1, to "11". One can make PP by fusing left-shifted multiplicand B with B (BT1). This addition makes use of the OR logic concept. Each multiplier bit's relative relevance affects the order of the output from each processor logic block. By appropriately employing the multiplicand (B) and the most significant bits (MSB) a7a6, we can increase the precision of the proposed multiplier. Both basic and reasoning skills can accomplish this.

The inputs to the proposed decoder logic blocks (with the multiplicand as the other input) produce approximate PPs, but the major components (a15–a14) yield accurate PPs. Following the PPG stage, the accumulation stage employs three-to-two carry-save adders to divide the nine PP rows into two halves. To achieve low power consumption (PP), one reduces the number of adders during the collection stage. This significantly impacts both the saved space and the amount of electricity utilized. To reduce the last two rows from the collecting stage, we use a ripple carry adder. We therefore present the final solution as p31-p0.The input selection bits provide seven rows of parallel processing for the seven decoder units. Logically, the AND gate produces the final two rows. To combine these rows into the final product, a reduction structure is required. The results that DeBAM provides are undoubtedly very close to reality. The implementation of the proposed 16x16 multiplier design with a circuit-level decoder. When ax plus 1 equals "00," the decoder logic block outputs zero. Adding the binary representation (b15→b0) to a left-displaced "1," yields the number "10". The number "01" is written as b15→b0. It is evident that only a few gates are required for each situation because all three require the conditional transfer of input bits or zero. While we can evaluate the antecedent scenario (ax ax+1 = "11") with the number 10, the current case (ax ax+1) results in a PPG value of 0.01. This time, we constructed the logic for the adder using an OR gate. The accompanying image illustrates the justification for using an 8 by 8 multiplier. A 16-bit multiplier fed with input (b15-b0) produces PP (p16-p0).

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

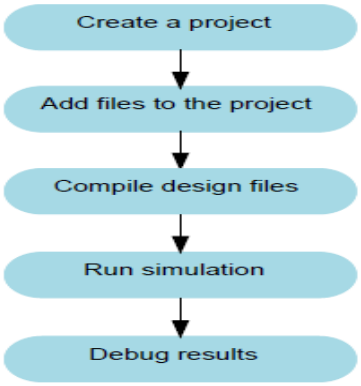


Figure 1. Project Flow

Figure 1 and 2 represents the project and multiple library flow. Table I and II shows the Multiplication of two number (5 X -4) in step by step using both Algorithm and Truth table for booth multiplier.

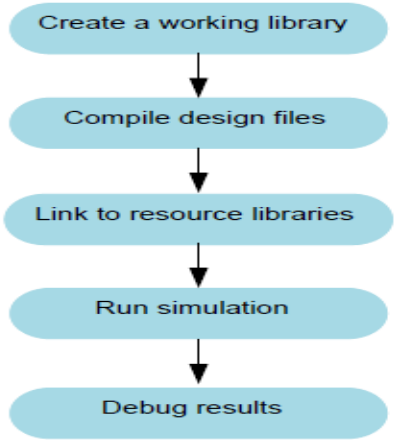


Figure 2. Multiple Library Flow

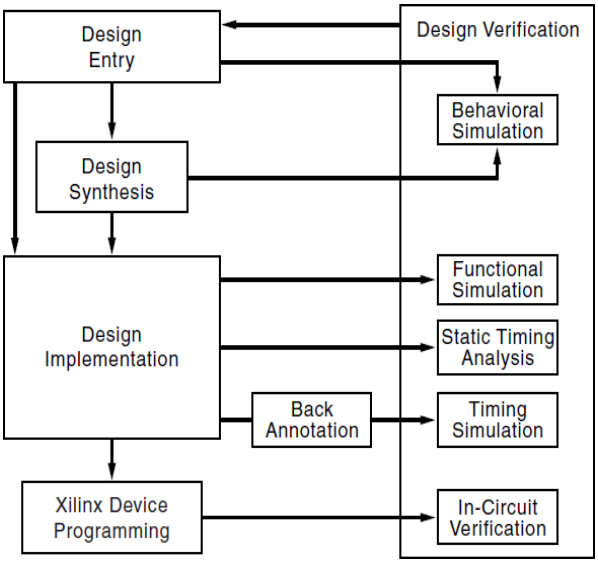


Figure 3. Block Diagram of Xilinx ISE

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

Table I. Multiplication of two number (5 X -4) in step by step using both Algorithm

Count	A	Q	Qn-1	Operations
1	0000	1100	0	Initialization
2	0000	0110	0	Right shift
3	0000	0011	0	Right shift
4	1011	0011	0	A=A-M=A+(M`+1)
5	1101	1001	1	Right shift
6	1110	1100	1	Right shift
	Ans = 00010100			2`s complement of 20

Table II. Truth table for booth multiplier

Q	Qn+1	Recorded bits	Operation
0	0	0	Shift
0	1	+1	Add
1	0	-1	Subtract
1	1	0	Shift

IV. RESULTS AND DISCUSSION

XILINX ISE offers an extensive array of functionalities that surpass the mere generation of design summaries and implementations. In addition, XILINX ISE carries out route synthesis, translation, and mapping. The design summary grants access to all the messages and extensive reports produced by the synthesis and implementation tools. Moreover, it offers a succinct overview of the crucial design details. The summary offers a thorough comprehension of our project, encompassing a complete plan, statistics on device utilization, and performance evaluations.

This processor takes an input and outputs a sixteen-bit partial product. After reducing nine rows with the load-saving adder, three rows remain. Employing parallel product (PP) creation reduces the frequency of adders used during the accumulation process. Reduced energy and space are thus required. Using a ripple carry adder, we obtain the final result (P31-P0) by adding the three rows obtained during the accumulation phase. There are three divisions altogether. We analyzed the power and area outputs using the Xilinx ISE software. Using the Model Sim application, one could view the output pattern by multiplying two 16-bit multipliers. Its development included the Very High Speed Integrated Circuit Hardware Description Language (VHDL). This programming language facilitates the implementation of dataflow models for digital systems.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

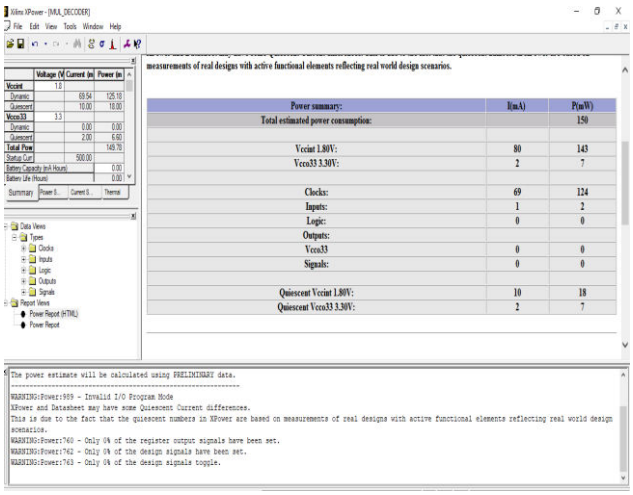


Figure 4. Power Consumption of Exact Multiplier

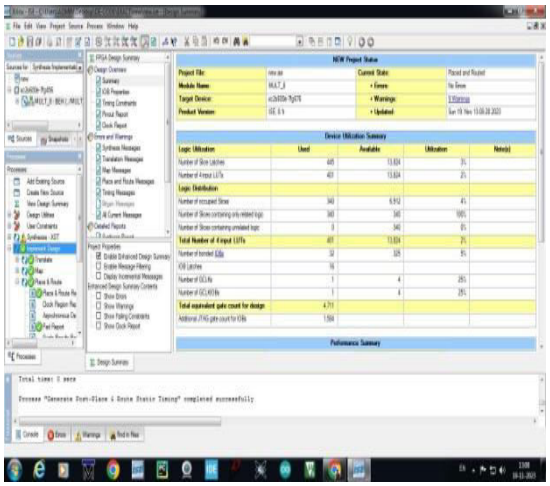


Figure 5. Output of Area in 16bit Multiplier

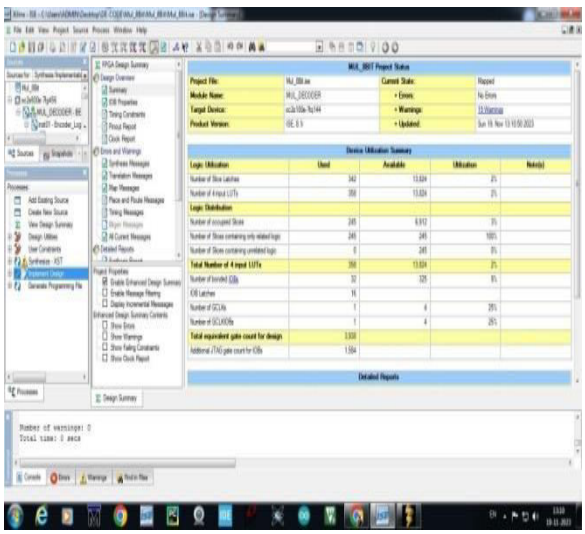


Figure 6. Gate Count of Proposed Multiplier1

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

Figures 4, 5 and 6 power consumption, output, and gate count of proposed multiplier 1.

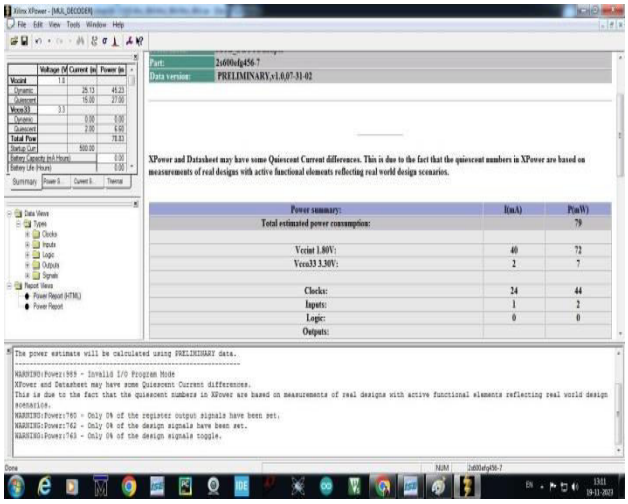


Figure 7. Power Consumption of Proposed Multiplier1

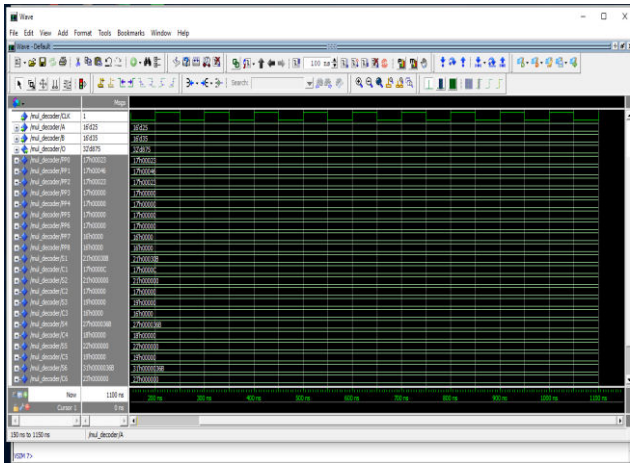


Figure 8. Output wave form of generating PP

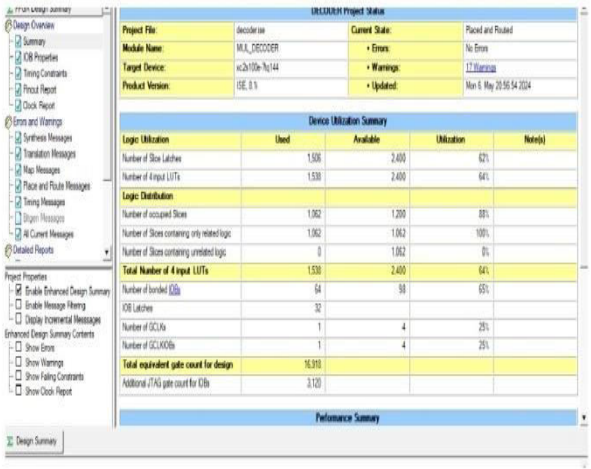


Figure 9. Output of Area of Proposed Multiplier 1

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

EXACT16 Project Status			
Project File:	exact16.ise	Current State:	Synthesized
Module Name:	multiplier1616	• Errors:	No Errors
Target Device:	xc3s100e-3q144	• Warnings:	58 Warnings
Product Version:	ISE, 8.1i	• Updated:	Mon 6 May 20:52:54 2024

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Latches	1,867	2,400	77%	
Number of 4 input LUTs	1,851	2,400	77%	
Logic Distribution				
Number of occupied Slices	1,198	1,200	99%	
Number of Slices containing only related logic	1,103	1,198	92%	
Number of Slices containing unrelated logic	95	1,198	7%	
Total Number of 4 input LUTs	1,851	2,400	77%	
Number of bonded IOBs	64	98	65%	
IOB Latches	32			
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	28,601			
Additional JTAG gate count for IOBs	3,120			

Figure 10. Output of Area of Proposed Multiplier 2

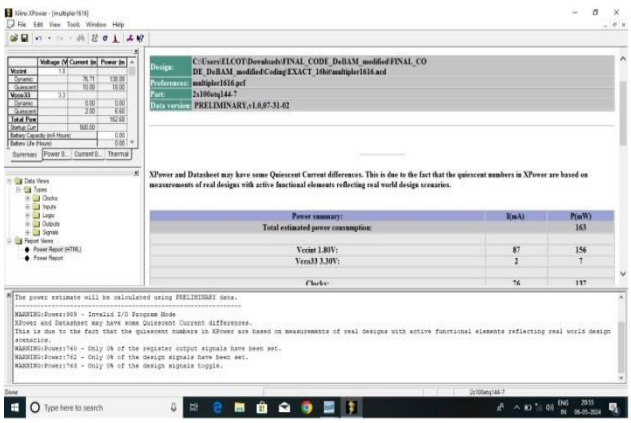
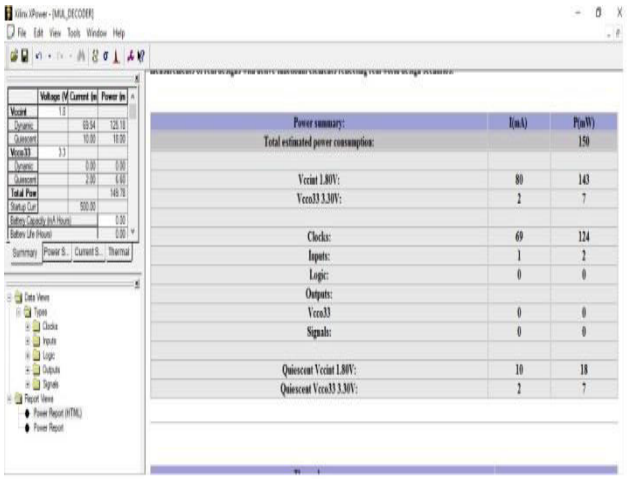


Figure 11. Area Utilization of Proposed Multiplier



Power summary:	I(mA)	P(mW)
Total estimated power consumption:		159
Vccint 1.80V:	80	143
Vcco33 3.30V:	2	7
Clocks:	69	124
Inputs:	1	2
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	10	18
Quiescent Vcco33 3.30V:	2	7

Figure 12. Power Summary of Proposed Multiplier

Figures 7, 8 and 9 power consumption, output wave, and output. Figures 10, 11 and 12 output of area, area utilization, and power summary of proposed multiplier.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

Table III. Recoding table of Radix-4 Booth multiplier

Block	Recoded digits	Operation
000	0	0
001	+1	+1
010	+1	+1
011	+2	+2
100	-2	-2
101	-1	-1
110	-1	-1
111	0	0

Table IV. Multiplier Comparison

Multiplier	Area	Power
Exact 16 bit	20,601	163
Proposed 16 bit	16,918	150

Table III shows the Recoding table of Radix-4 Booth multiplier. Table IV represents the comparison of exact and proposed 16 bit multiplier which shows the proposed 16 bit multiplier has better performance than existing multiplier.

V. CONCLUSION AND FUTURE WORK

Power consumption, delay, and space utilization are the three areas where approximation computers may be useful. The proposed DeBAM (N,M) design decreases the quantity of PPs produced while optimizing the PP accumulation phase. Hardware and error statistics reveal that the proposed multiplexer consumes less energy and occupies less space than existing choices. As demonstrated by the previously explored photo compression and enhancement techniques, the proposed strategy improves the trade-off between computational effort and precision.

REFERENCES

[1] A. Kumari, Ch. Vandana, Chowta Mallikharjuna Rao, Uppugunduru Anil Kumar, and Syed Ershad Ahmed, "Low-Power Approximate Multiplier Architecture for Error Resilient Applications," *2023 11th International Conference on Intelligent Systems and Embedded Design (ISED)*, Dec. 2023, doi: https://doi.org/10.1109/ised59382.2023.10444585.

[2] S. P. Joy Vasantha Rani, J. R. Lourdu Jennifer, and P. Sudhanya, "Approximate Multipliers Design Using Approximate Adders for Image Processing Applications," *Journal of Circuits, Systems and Computers*, vol. 31, no. 15, Jun. 2022, doi: https://doi.org/10.1142/s0218126622502565.

[3] Parthibaraj Anguraj and T. Krishnan, "Design and realization of area-efficient approximate multiplier structures for image processing applications," *Microprocessors and microsystems*, vol. 102, pp. 104925–104925, Oct. 2023, doi: https://doi.org/10.1016/j.micpro.2023.104925.

[4] F. Frustaci, "Improving the Quality Degradation of Dynamically Configurable Approximate Multipliers via Data Correlation," *Electronics*, vol. 10, no. 17, p. 2063, Aug. 2021, doi: https://doi.org/10.3390/electronics10172063.

[5] Pruthvy Yellu, Nishanth Chennagouni, and Q. Yu, "Leveraging Intermediate Node Evaluation to Secure Approximate Computing for AI Applications," *2022 IEEE International Symposium on Technologies for Homeland Security (HST)*, Nov. 2022, doi: https://doi.org/10.1109/hst56032.2022.10025430.

[6] Pruthvy Yellu, Nishanth Chennagouni, and Q. Yu, "INEAD: Intermediate Node Evaluation-Based Attack Detection for Secure Approximate Computing Systems," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 43, no. 3, pp. 716–727, Mar. 2024, doi: https://doi.org/10.1109/tcad.2023.3328826.

[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 Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 6, pp. 1180–1184, Jun. 2015, doi: https://doi.org/10.1109/tvlsi.2014.2333366.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Refereed Journal |

|| Volume 13, Issue 5, May 2024 ||

| DOI:10.15680/IJIRSET.2024.1305113 |

[8] Suresh Nambi, U. Anil Kumar, K. Radhakrishnan, M. Venkatesan, and Syed Ershad Ahmed, "DeBAM: Decoder-Based Approximate Multiplier for Low Power Applications," *IEEE embedded systems letters*, vol. 13, no. 4, pp. 174–177, Dec. 2021, doi: https://doi.org/10.1109/les.2020.3045165.

[9] Parthibaraj Anguraj and T. Krishnan, "Design of area-efficient modified decoder-based imprecise multiplier for error-resilient applications," *Microelectronics*, vol. 141, pp. 105957–105957, Nov. 2023, doi: https://doi.org/10.1016/j.mejo.2023.105957.

[10] Muhammad Hamis Haider and S.-B. Ko, "Booth Encoding-Based Energy Efficient Multipliers for Deep Learning Systems," *IEEE transactions on circuits and systems. II, Express briefs*, vol. 70, no. 6, pp. 2241–2245, Jun. 2023, doi: https://doi.org/10.1109/tcsii.2022.3233923.

[11] B. Vakili, O. Akbari, and B. Ebrahimi, "Efficient approximate multipliers utilizing compact and low-power compressors for error-resilient applications," *AEÜ. International journal of electronics and communications*, vol. 174, pp. 155039–155039, Jan. 2024, doi: https://doi.org/10.1016/j.aeue.2023.155039.

[12] Uppugunduru Anil Kumar, Goli Naga Sandesh, Dhoti Ojusteja, and Syed Ershad Ahmed, "Lower part OR based Approximate Multiplier for Error Resilient Applications," *2021 IEEE International Symposium on Smart Electronic Systems (iSES)*, Dec. 2021, doi: https://doi.org/10.1109/ises52644.2021.00046.

[13] U. Anil Kumar, Pavankumar Bikki, Sreehari Veeramachaneni, and Syed Ershad Ahmed, "Power Efficient Approximate Multiplier Architectures for Error Resilient Applications," *2022 IEEE 19th India Council International Conference (INDICON)*, Nov. 2022, doi: https://doi.org/10.1109/indicon56171.2022.10039748.

[14] Pramod Alamuri, U. Anil Kumar, Vallepu Vannuru, and Syed Ershad Ahmed, "Improved approximate multiplier architecture for image processing and neural network applications," *Microprocessors and microsystems*, vol. 101, pp. 104909–104909, Sep. 2023, doi: https://doi.org/10.1016/j.micpro.2023.104909.

[15] Sahith Guturu, Uppugunduru Anil Kumar, S. Vignesh Bharadwaj, and Syed Ershad Ahmed, "Design methodology for highly accurate approximate multipliers for error resilient applications," *Computers & electrical engineering*, vol. 110, pp. 108798–108798, Sep. 2023, doi: https://doi.org/10.1016/j.compeleceng.2023.108798.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details