I.J. Information Technology and Computer Science, **2025, 2, 10-23** Published Online on April 8, 2025 by MECS Press (http://www.mecs-press.org/) DOI: 10.5815/ijitcs.2025.02.02

An efficient floating point multiplier for low power single precision FPGA implementation

Thangamani C
PG Scholar
ME-VLSI Design
K.S.Rangasamy college of Technology,
Namakkal, India
nithinthangam2016@gmail.com

R.Poovarasan M.E.,
Assistant Professor
ECE, K.S.Rangasamy college of Technology,
Namakkal, India
poovarasan@ksrct.ac.in

ABSTRACT

A rapid and energy-efficient floating point unit is constantly required for significant applications such as digital signal processing, image processing, real-time data processing, and multimedia applications. This project suggests using a single-precision Floating Point Multiplier engine to speed FPGA applications. The engine has a modified approach for virtual zeropadding to conserve memory space, and it also includes customizable settings for filter and picture size. To improve the suggested multiplier design, a low power multiplier with lower dynamic power, notably while acting on pixels, has been presented, as well as a quicker increment by one circuit based on gates. Finally, the project displays the post-synthesis power dissipation, area estimation, and picture quality comparison achieved from the suggested architecture's RTL simulation.

1. INTRODUCTION

In the current world, a processor's total performance is governed solely by the adder and multiplier circuits. At the same time, most real-world issues are much more complicated than fixed-point multipliers; floating-point multipliers are ideally suited for high precision. However, floating point processing is extremely complicated in terms of storage and representation. At the same time, most scientific computations rely heavily on floating point values, which may be customized in embedded systems or hardware implementations such as field programmable gate arrays (FPGAs). Many authors reported their research on these circuits in order to improve VLSI performance matrices such as area and power optimization, as well as delay reduction. Multipliers are the most often used arithmetic blocks in many digital signal processing applications, providing the foundation for operations such as filtering, correlation, de-noising, and domain transformation. Fixed-point implementation is widely used in a variety of electronic systems, including transceivers, FPGA accelerators, digital phase locked loops, and spread spectrum clock generators, because of its advantageous hardware properties. set-point arithmetic uses a set number of bits to represent the integer and fractional components of a signal. The integer part's bit length is connected to the range of representable values, but the fractional part's bit length influences operation precision. To regulate numerical range and precision, the designer must select appropriate signal bitwidths and resolutions.

Floating-point (FP) arithmetic, while difficult in terms of hardware implementation, provides a

versatile method of doing numerical computations while maintaining a wide range of representable values and high accuracy. It is so widely utilized in fields like as scientific computing, digital signal processing, and computer graphics. Most computer systems, including CPUs, GPUs, and microcontrollers, use the standardized IEEE 754 format. This standard defines an FP number as having a sign S, an exponent E, and a mantissa M. The FP format encodes the value as A = (−1)S·(1 + M)·2E-bias, where M ranges from 0 to 1.

2. LITERATURE SURVEY

Towhidy, Ahmad, Reza Omidi, and Karim Mohammadi. " On the Plan of Iterative Inexact

Drifting Point Multipliers." IEEE Exchanges on PCs (2022).

Surmised multipliers save power and space in applications that need mistake resistance. Two two-dimensional pseudo-Booth encoded approximate floating-point multipliers are first presented in this paper: drifting point pseudoCorner (PB) and drifting point iterative pseudoStall (IPB). Three parameters can be used to adjust the accuracy of the proposed multipliers: word length following truncation (W), iteration, and the encoder radix (R). Then, we created customary iterative multipliers with an improved on guiding circuit for their rectification part to diminish multiplier power utilization. The proposed iterative multipliers are contrasted with customary iterative number multipliers that utilization a worked on guiding circuit in the drifting point region. When compared to the exact floating-point multiplier, the proposed PB-R4-W4 and IPB-R16-W19 reduce power consumption in TSMC 180nm CMOS technology by up to 98.9 percent and 67.5 percent, respectively. Also, their MRED values are 2.9% and (7.4×10−4)%.

Bin Zhou, Guangsen Wang, Guisheng Jie, Qing Liu, Zhiwei Wang. A rapid drifting point duplicate collector in view of fpgas." IEEE Exchanges for Exceptionally Enormous Scope Combination (VLSI) Frameworks 29, no. 10 (2021): 1782-1789.

A novel high-speed floating-point multiplyaccumulator (FPMAC) is proposed in this article. It comprises of a marked delicate multiplier and a solitary cycle drifting point gatherer (FAAC). The multiplier is carried out utilizing a radix-4 Stall encoding in light of sign-extent inputs. The FAAC incorporates a bidirectional shift arrangement, a quick 3:1 blower, and a threeoperand driving zero indicator (LZP). Our FPMAC maintains its excellent delay and resource consumption performance despite its quick critical path, full-pipelined structure, and straightforward implementation processes. On two Xilinx devices, our FPMAC has been developed and examined. Our FPMAC's maximum clock frequencies rise by 33.4%– 74.4% in single-precision floating-point (SFP) and 15.8%–86.1% in double-precision floatingpoint (DFP) architectures, according to the findings.

Lyu, Fei, Yan Xia, Yuheng Chen, Yanxu Wang, Yuanyong Luo, and Yu Wang. " Highthroughput low-inactivity pipelined divider for single-accuracy drifting point numbers." IEEE Exchanges for Extremely Huge Scope Mix (VLSI) Frameworks 30, no. 4 (2022): 544-548.

In this concise, we propose a completely pipelined divider for single-accuracy drifting point numbers in light of a widespread piecewise straight (PWL) estimation technique and a changed Goldschmidt calculation. The cutting edge general PWL strategy utilizes a reasonable number of sections and partial piece widths to meet the necessity of the predefined most extreme outright mistake. The modified Goldschmidt algorithm makes use of relatively insignificant multipliers. In the equipment execution, the multipliers are enhanced with the radix-4 and radix-8 stall encoding techniques to decrease the quantity of fractional items. Likewise, the amount of the incomplete items and different information are determined by a blower and a viper to abbreviate the basic way. Integrated results show that the greatest reachable recurrence of our plan is superior to those of the current techniques. In addition, our design outperforms other approaches by a wide margin in terms of latency and throughput.

Seok-Bum Ko, Hao, and Zhang Proficient Estimated Place Multipliers for Profound Learning Calculation." IEEE Diary on Arising and Chose Themes in Circuits and Frameworks 13, no. 1 (2022): 201-211.

Set numeric configuration has acquired ubiquity lately. Its tightening precision makes it exceptionally ideal in different applications including profound picking up registering. However, posit arithmetic is more expensive to implement in hardware than its floating-point counterpart due to its dynamic bit-width. To resolve this cost issue, this work proposes set multipliers with estimate processing properties. The essential idea of the proposed plan is to shorten the part multiplier as per the normal division bit-width of the item. With the goal that the asset utilization of the division multiplier and thus the part snake might be incredibly diminished. Multipliers are used to implement the proposed strategy in both the linear and logarithm domains. The suggested

approximation posit multipliers in 8/16/32 bits are implemented and examined. For the regularly utilized 16-bit set design in profound picking up figuring, the recommended estimation set multiplier can use 16% less power contrasted with the ongoing place multiplier plan. The recommended 16-cycle rough logarithm multiplier might deliver a 15% improvement as far as power utilization contrasted with the best in class place estimated logarithm multiplier.

Haroon Waris, Weiqiang Liu, Ke, Yue Gao, and Fabrizio Lombardi Inexact Softmax Capabilities for Energy-Proficient Profound Brain Organizations." The 31st issue of IEEE Transactions on Very Large Scale Integration (VLSI) Systems 1 (2022): 4-16.

A new paradigm known as approximate computing has emerged, resulting in highperformance, energy-efficient, and less stringently required mathematical systems. Nonlinear capabilities, (for example, softmax, amended direct unit (ReLU), Tanh, and Sigmoid) are widely utilized in profound brain organizations (DNNs). Be that as it may, they experience significant power squander because of the great circuit intricacy. Estimate direct capabilities may and ought to be planned utilizing DNNs since they are blunder open minded. A proposal for an approximation softmax function (AxSF) is made in this article. A double hybrid structure (DHS) underpins AxSF. In order to accommodate various processing techniques, AxSF divides the softmax function's input into two parts. The main pieces (MSBs) are taken care of utilizing query tables (LUTs) and a precise reestablishing cluster divider (EXDr). Less critical pieces (LSBs) are addressed utilizing Taylor's development and a logarithmic divider. A better DHS (IDHS) is likewise proposed to limit equipment intricacy. In IDHS, the half breed technique utilizes an exceptional Stall multiplier to increment fractional item creation and pressure, while the divider unit utilizes the shortened execution. The recommended DHS and IDHS are considered in contrast to current softmax plans. The discoveries exhibit that the recommended guess softmax design brings down equipment by 48%, delays by 54%, and keeps up with great exactness.

3. EXISTING SYSTEM

The radix-2 Baugh-Wooley multipliers are somewhat slow yet fast multipliers that employ Booth coding. A BR4 multiplier has been proposed, in which the PPR is supplied with less than half of the partial products used in the BW2 multiplier, resulting in a substantially shorter critical path. In contrast to the symmetric BW2 multiplier, the BR4 architecture processes the two input operands differently, as x is given to the recoding logic, which determines which multiples of y should be supplied to the PPR. For the proposed implementation, a carry-save adder with (m,2) compressors is used for the PPR and a carry-propagate adder for the VMA, similar to the BW2 multiplier. The 8 × 8-bit BR4 multiplier was analyzed using the same nonzero partial product per input operand analysis as the BW2 multiplier. Booth multiplication, which is based on string recoding, may multiple a pair of two's complement integers regardless of their signs. Due to string recoding, the radix-4 Booth multiplier requires just (n+1)/(2) multiplicand multiples to be added together (Table 1). In this technique, bxc represents the greatest integer less than or equal to the real number x.

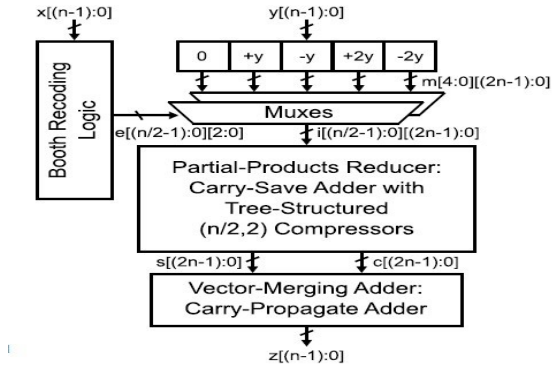

FIG.1 A signed bit radix-4 Booth-recoded multiplier has the following structure.

4. PROPOSED SYSTEM

This research proposes Floating Point Approximate designs for a Booth Multiplier (FPAM), which are based on an approximation approach that deals not only with partial product accumulation but also with the production of recoded multiplicands. A 2-bit approximation recoding adder is first developed to eliminate the extra latency found in prior radix-4 systems, enhancing the performance of the radix-4 Booth algorithm. A Wallace tree is used to compute the sum of partial products and further minimize the

addition time. The least important partial products are then truncated to decrease power and delay. The floating 8 × 8-bit approximate radix-4 Booth multipliers 1 and 2 (abbreviated ABM1 and ABM2) are presented. The simulation results demonstrate that ABM1 is 20% quicker than the correct radix-4 Booth multiplier. ABM2 saves up to 44% in power dissipation when compared to the accurate multiplier. This approximation multiplier is also 31% quicker and uses 43% less circuit space. Finally, the ABMs are applied to a Median filter, demonstrating that the suggested approximate multipliers outperform those provided in the technical report.

b_{2i+1}	b_{2i}	b_{2i-1}	neg_i	two_i	$zero_i$	PP_i
0	0	0	0	0	1	0
0	0	1	0	0	0	$+A$
0	1	0	0	0	0	$+A$
0	1	1	0	1	0	$+2A$
1	0	0	1	1	0	$-2A$
1	0	1	1	0	0	$-1A$
1	1	0	1	0	0	$-1A$
1	1	1	0	0	1	0

To ensure proper alignment routes, we chose the SEL architecture with normalization as a contrast to our BLOCK design. Both comparative designs are not identical to their originals. In addition, we have recreated two contrast designs in the text to provide a fair and straightforward comparison. The differences between our FPMAC design and the CON and SEL architectures are outlined below.

1) A signed soft multiplier using sign-magnitude inputs is implemented.

2) To ensure alignment, a bidirectional shift substitutes the unidirectional right-shift.

3) A 4:1 compressor works as the accumulator and the final adder.

4) Using a three-operand LZP over a twooperand LZP speeds up the normalization process.

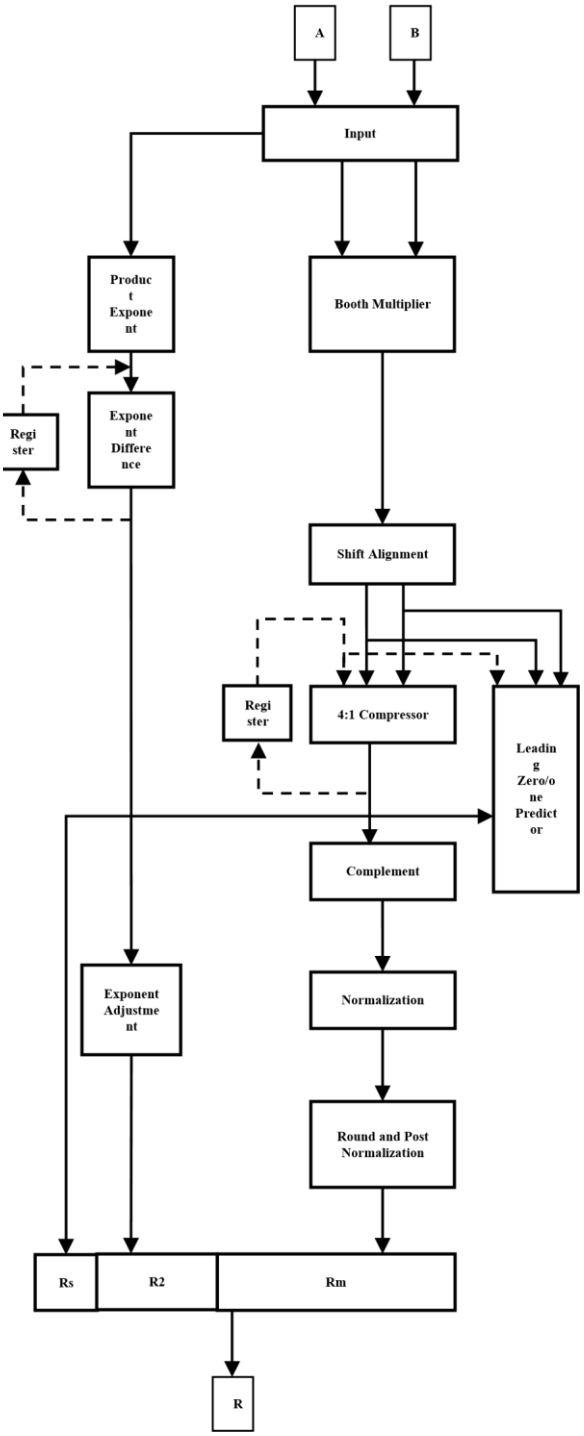

FIG.2. Proposed Block Diagram

The input B is classified into sets of bits {b2i+1, b2i, b2i-1} that correspond to one of the values from 0, 1, or 2. The radix-4 Booth encoder converts these bit groups into three signals: negi, twoi, and zeroi, which represent the values 0, 1, or 2, as illustrated in Table 1. Negi represents the sign of each partial product operation, twoi denotes if the resulting partial product is to be shifted, and zeroi defines whether the partial

product is zero or non-zero. Using the signals negi, twoi, and zero, the rowwise partial product P Pi is chosen from the set {2A,1A, 0, 1A, 2A}. Radix-4 encoding generates eight partial product rows for i ranging from 0 to 7 from 16-bit signed inputs. Figure 1 depicts how the partial product generator outputs each element in the partial product row. The logic equation of the partial product element pij created by the partial product generator is provided by h mi . (2) Here, mi is the multiplexer output and aj is the multiplicand input row. The partial product matrix for a 16-bit precise Booth multiplier after sign-extension reduction is displayed.

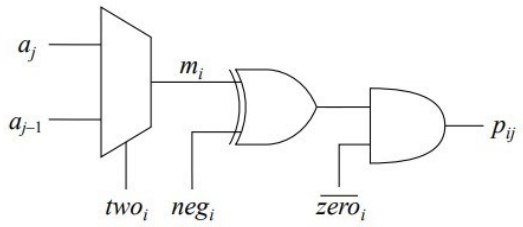

FIG.3. Circuit schematic for the partial product generator

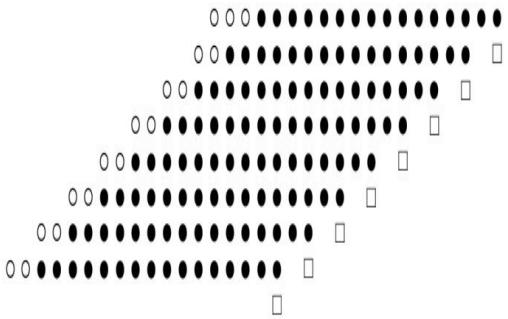

FIG.4. Partial product matrix of a 16-bit radix-4 Booth multiplier (●: a partial product, o: a sign-extension term, □: a correction term)

Approximation in Bootstrap Multipliers

Booth multipliers are good candidates for using approximation in both partial product accumulation and generation. An exact radix-4 partial product generator requires all three signals, negi, twoi, and zeroi, to produce the partial product. For ABM-M1 and ABMM2, an approximate partial product generator is created with just two of the three signals, negi and twoi. In ABM-M3, a partial product generator is developed that just employs the signal zeroi. In ABM-M1, an approximation is used in partial product accumulation by coupling the correction term to its associated row in the partial product matrix, lowering the depth of the matrix. In ABM-M2 and ABM-M3, approximation in generation is achieved by replacing a collection of partial product generators with a single approximate partial product generator, resulting in a reduction in the number of accumulation components.

4.1 VERILOG

In electronics, a hardware description language (HDL) is a specialized computer language used to program the structure, design and functioning of electronic circuits, and most typically, digital logic circuits. A hardware description language provides for the accurate, formal definition of an electrical circuit, as well as automated analysis, modeling, and simulated testing of the circuit. It also enables the compilation of an HDL program into a lower-level specification of actual electrical components, such as the masks needed to construct an integrated circuit. A hardware description language is similar to a programming language like C; it is a textual description made up of expressions, statements, and control structures. The fact that HDLs clearly contain the concept of time distinguishes them from the majority of computer languages. HDLs are an essential aspect of electronic design automation (EDA) systems, particularly for complicated circuits like microprocessors.

HDLs are text-based expressions that describe the structure of electronic systems and their behavior across time. Similar to concurrent programming languages, HDL syntax and semantics allow explicit notations for expressing concurrency. However, unlike most software programming languages, HDLs have an explicit concept of time, which is a key feature of hardware. Languages that simply represent circuit connection between a hierarchy of blocks are appropriately categorized as netlist languages used in electric computer-aided design (CAD). HDL can be used to describe designs in structural, behavioral, or registertransfer architectures for the same circuit functionality; in the latter two cases, the synthesizer determines the architecture and logic gate layout.

5. RESULTS

SIMULATION RESULTS FOR MULTIPLIER

In this part, we evaluate the hardware overheads and accuracy of the suggested approximate 16-bit multiplier designs. Meanwhile, the performance of exact and various other approximate Booth multipliers is compared to the suggested designs. The multipliers are architecturally described in Verilog HDL, and the simulation and synthesis are performed using Xilinx® ISE 14.7. Virtex-6 (XC6VLX75TL FF784 -1) is compared in terms of area, power, and delay. The structure is identical for the floating point multiplier designs, with the exception of the 24x24 multiplier block, which will be created using distinct integer multipliers. Thus, the size and speed of floating point multipliers are determined by the performance of integer multipliers.

The Vedic Multipliers require more space than the suggested booth multipliers. It is predicted given the presence of three adder structures and four 16x16 multipliers, each of which contains an 8x8 multiplier structure, and so on.

The Booth Multipliers were found to be faster than other multipliers. This is because the partial products are created and added in concurrently. Also, the delay caused by multipliers is mostly due to the propagation of carry bits through adders. Booth multiplication is a smaller, quicker multiplication method that encodes signed integers, which is also a popular approach used in chip design. It delivers considerable gains by lowering the amount of partial products by half compared to "long multiplication" techniques.

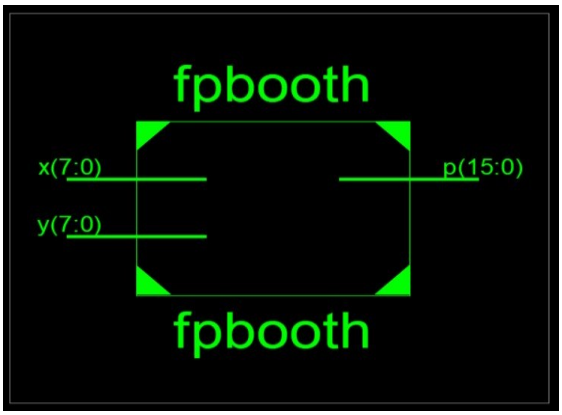

FIG.6. RTL Schematic for Proposed MVU

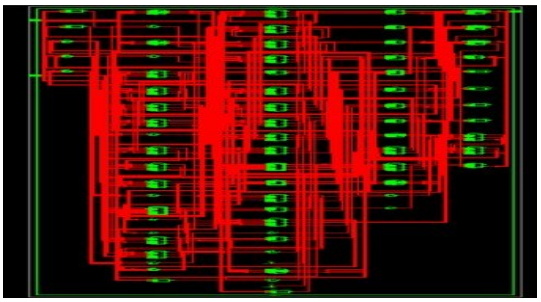

Fig. RTL Schematic

The module includes an RTL schematic design for the proposed Radix-4 polar encoder, as well as an output for the proposed Radix-4
Reconfigurable polar encoder. The figure depicts the RTL schematic and output for the proposed Radix-4 multiplier.

Image Processing Applications:

We prove the correctness of our concept by investigating the performance of segmented FPMs in an image filtering application. The image filtering algorithm applies the following operation on the input picture I.In this example, we analyze a smoothing application with a gaussian kernel of size 3 × 3 and standard deviation 2. The coefficients are floating-point integers with values normalized to one. To obtain the filter mask, use the MATLAB function fspecial('gaussian', 5, 2). The table shows the findings in terms of structural similarity index measure (SSIM) and peak signal-to-noise ratio (PSNR, in dB), which were produced by filtering three photos (Lena, Cameraman, and Lady). For each example, we average the SSIM and PSNR acquired from processing the test pictures, and we display the overall average SSIM and PSNR as a synthetic value in the table's last column.

Module	PSNR [dB]	SSIM
Existing Vedic	47.5	0.995
Proposed Booth Multiplier	56.1	1

Table: MATLAB Output Comparison.

In this module, power, area, and delay analyses are performed and the findings compared. The power, area, and delay analyses were performed

using Xilinx ISE 14.7. The findings are reported in Table 1.

LDPC Module	Power Consumption (mW)	Area (Device Utilization)			Delay (nS)
		IOB	LUT	FF	
Existing Vedic	1.801	32	140	120	7.758
Proposed Booth Multiplier	1.065	32	96	96	7.137

Table. 1. Comparative Analysis of Power, Area, and Delay on Proposed FPAM

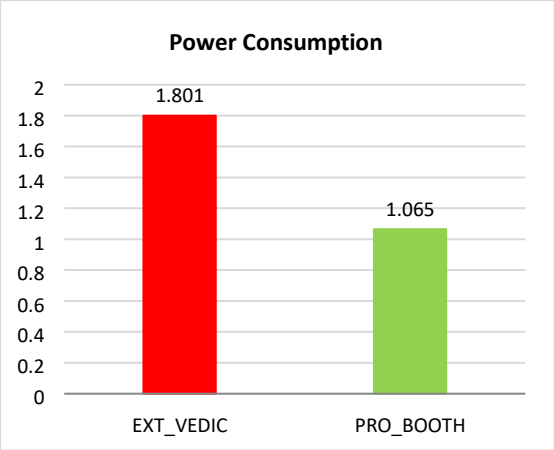

Figure: Power Analysis - LDPC vs. Proposed LDPC MCV-4

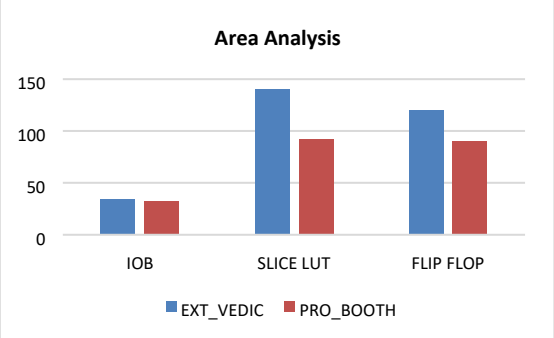

Figure Area Analysis

Power analysis shows that the Proposed has less power than the Regular. The proposed booth multiplier reduces power by 53% compared to the set regular multiplier.

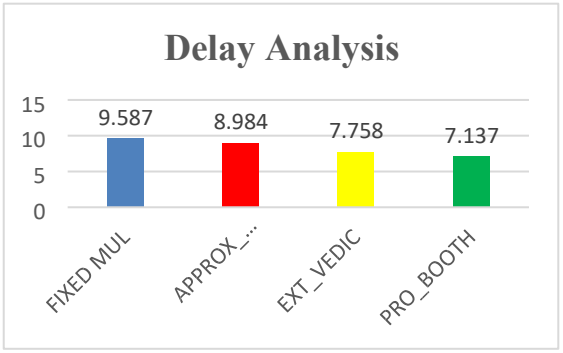

Figure: Delay Analysis

6. CONCLUSION AND FUTURE WORK

Three types of approximate Booth multipliers are suggested, with the focus on partial product production. This project describes the development of a floating point multiplier that supports the IEEE 754 binary exchange standard. The whole design was recorded using Verilog Hardware Description Language (HDL). The calculation delays achieved using Vedic and modified booth multipliers are compared. The design is implemented using a Xilinx ISE 14.7 tool with the Vertex 6 device XC6VLX75TL 1FF784. The advantage of this technique in this work is that utilizing parallelism becomes trivial, and substantial dynamic power may be saved using the low power multiplier. Areapower product and error metrics of the proposed multipliers are compared with accurate multipliers and existing state-of-the-art approximation Booth multipliers. When compared to the precise Booth multiplier

7. REFERENCES

[1] Towhidy, Ahmad, Reza Omidi, and Karim Mohammadi. " On the Plan of Iterative Inexact Drifting Point Multipliers." IEEE Computer Transactions, 2022.

Zhou, Bin, Guangsen Wang, Guisheng Jie, Qing Liu, and Zhiwei Wang are the second group. A

rapid drifting point duplicate collector in view of fpgas." IEEE Exchanges for Exceptionally Enormous Scope Combination (VLSI) Frameworks 29, no. 10 (2021), 1782–1789.

[3] Lyu, Fei, Yan Xia, Yuheng Chen, Yanxu Wang, Yuanyong Luo, and Yu Wang. " High-throughput low-inactivity pipelined divider for singleaccuracy drifting point numbers." IEEE Exchanges for Extremely Huge Scope Mix (VLSI) Frameworks 30, no. 4 (2022): 544-548.

[4] Hao, Seok-Bum Ko, and Zhang. Proficient Estimated Place Multipliers for Profound Learning Calculation." IEEE Journal on Emerging and Selected Topics in Circuits and Systems, Volume 13.1 in the year 2022: 201-211.

[5] Haroon Waris, Weiqiang Liu, Fabrizio Lombardi, Chen, Ke, Yue Gao, and Inexact Softmax Capabilities for Energy-Proficient Profound Brain Organizations." The 31st issue of IEEE Transactions on Very Large Scale Integration (VLSI) Systems 1 (2022), pp. 4-16.

[6] "Fast Data Delivery for Many-Core Processors," IEEE Transactions on Computers, by M. Bakhshalipour, P. Lotfi-Kamran, A. Mazloumi, F. Samandi, M. Naderan, M. Modaresi, and H. Sarbazi-Azad, IEEE Transactions on Computers, 2018.

[7] "Multipliers-Driven Perturbation of Coefficients for Low Power Operation in Reconfiguration FIR Filters," by Andrea Bonetti, Adam Teman, Philippe Flatresse, and Andreas Burg, IEEE Trans. 2017.

[8] E. Viegas, A. O. Santin, A. Franc¸a, R. Jasinski, V. A. Pedroni, and L. S. Oliveira. " IEEE Transactions on Computers (TC) published "Towards An Energy-Efficient Anomaly-Based Intrusion Detection Engine for Embedded Systems" in 2017.

[9] M. Salehi, A. Ejlali, and B. M. Al-Hashimi, "Two-Stage Low-Energy N-Measured Overt repetitiveness for Hard Constant Multi-Center Frameworks," IEEE Exchanges on Equal and Disseminated Frameworks (2016).

[10] Y. Huan and others IEEE Transactions on Computers, vol. 5, 2005), "A 101.4 GOPS/W reconfigurable and scalable controlcentric embedded processor for domain-specific applications." Circuit Framework. I, Reg. Volume of papers. 63, no. 12, pp. 2245-2256, December 2016.