DESIGNING A MAC UNIT USING APPROXIMATE MULTIPLIER

Phase-1 Project report submitted in partial fulfillment of the requirement for the award of the Degree of B.Tech

By

KOTTE PRATHYUSHA (20241A0431)

MUNTHA BALARAJU (20241A0436)

BATHINENI AKASH (20241A0405)

Under the Guidance

Of

Dr. K. JAMAL

Department of Electronics and Communication Engineering

Gokaraju Rangaraju Institute of Engineering and Technology (AICTE Approved; Autonomous under JNTU)

Bachupally, Kukatpally, Hyderabad-500090

Gokaraju Rangaraju Institute of Engineering & Technology

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that this thesis entitled "Designing a Mac unit Using Approximate Multiplier" submitted by Kotte Prathyusha (Roll No:20241A0431), Muntha Balaraju(Roll No: 20241A0436), Bathineni Akash(Roll No: 20241A0405), in partial fulfillment of the requirements for the degree of Bachelor of Technology in Electronics and Communication Engineering of JNTUH, during the academic year 2023- 24, is a bonafide record of research work carried out by his/her under our guidance and supervision. The contents of this thesis, in full or in parts, have not been submitted to any other university or Institution for the award of any degree or diploma.

Internal Guide External Examiner Head of the Department

DECLARATION

I hereby declare that the mini project entitled “Designing a Mac Unit Using Approximate Multiplier” is the work done during the period from June 2023 to November 2023 and is submitted in the partial fulfillment of the requirements for the award of Bachelor of Technology in Electronics and Communication Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad). The results embodied in this project have not been submitted to any other university or Institution for the award of any Degree or Diploma.

Kotte Prathyusha(Roll No: 20241A0431)

Muntha Balaraju(Roll No: 20241A0436)

Bathineni Akash(Roll No: 20241A0405)

ACKNOWLEDGEMENT

I gratefully thank my project guide Dr. K. Jamal, Professor, Department of Electronics and Communication Engineering, Gokaraju Rangaraju Institute of Engineering and Technology for her excellent guidance and timely suggestions throughout the project work.

I wish to express my sincere gratitude to Project Review Committee Members (Dr. V. Aravind, Dr. T. Jagannadha Swamy, Dr. G. Mamatha), Department of Electronics and Communication Engineering for providing valuable guidance and suggestions during the course of work.

I wish to express my sincere thanks to Ms. A.Usha Sree, Assistant Professor, Department of Electronics and Communication Engineering, Gokaraju Rangaraju Institute of Engineering and Technology for reviewing my work time to time and giving necessary suggestions needed during the project work.

I would like to convey my sincere thanks to Dr. Ch Usha Kumari, Professor and Head of the Department, Electronics and Communication Engineering, Gokaraju Rangaraju Institute of Engineering and Technology for valuable suggestions and motivation during the work.

I convey my deep sense of gratitude to Dr. J. Praveen, Principal for the kind encouragement and support.

I would like to thank all teaching and non-teaching staff of the Department of Electronics and Communication Engineering for their kind support and suggestions which in turn helped me to complete the project successfully.

Finally, I would like to convey my sincere thanks to each and every person who have supported and helped me for the success of the project.

Kotte Prathyusha
Muntha Balaraju
Bathineni Akash

ABSTRACT

In numerous applications, multipliers play a crucial role as arithmetic functional units, often requiring extensive multiplications that contribute significantly to power consumption. Employing an approximate multiplier represents a novel strategy to reduce critical path time and power usage in error-tolerant systems. The trade-off involves sacrificing accuracy for enhanced performance and reduced energy consumption. To cater to varying precision requirements, this article not only introduces a highly accurate approximate 4-2 compressor but also proposes a adaptable approximation multiplier capable of dynamically truncating partial products. Additionally, a Multiplier and Accumulation (MAC) unit is recommended. Depending on user needs, the suggested MAC unit, paired with an approximate multiplier, can dynamically adjust power and accuracy for multiplications during runtime.

Contents

DECLARATION 3

ACKNOWLEDGEMENT 4

ABSTRACT 5

TABLE OF CONTENTS..... 6

LIST OF FIGURES..... 7

LIST OF TABLES..... 8

CHAPTER 1 9

1.1 INTRODUCTION 9

1.2 PROJECT OBJECTIVES 11

1.3 PROBLEM DEFINITION 11

1.4 FEATURES 11

1.5 STRUCTURE OF THESIS 12

CHAPTER 2 13

2.1 LITERATURE SURVEY 13

2.2 SOFTWARE REQUIREMENTS 18

CHAPTER 3 22

3.1 EXISTING METHOD..... 22

3.2 PROPOSED METHOD..... 27

CHAPTER 4 31

4.1 RESULTS 31

4.2 CONCLUSION 35

CHAPTER 5 36

5.1 FUTURE SCOPE 36

5.2 REFERENCES 36

LIST OF FIGURES

Fig 1.1:Basic MAC…….10

Fig .2.2.1:Create new folder for design..……18

Fig 2.2.2: Set family and device before design a project……19

Fig 2.2.3: Ready to design a project and create module name(.v files names)for design..…19

Fig 2.2.4: Declaration of input and output ports with their bit lengths and The schematic was created by its ports.……..19

Fig 2.2.5: Ready to write the code for design and check the syntax.…….20

Fig 2.2.6: Simulation of design to verifying the logics of design……20

Fig 2.2.7: Apply inputs through force constant or force clock for input signals……20

Fig 2.2.8: Run the design after applying inputs……21

Fig 3.1.1: Gatelevel approximate compressor……..22

Fig 3.1.2 Modified Partial Product……24

Fig.3.1.3: An example of gate sharing to minimize the number of gates…….25

Fig 3.1.4: Proposed approximate multiplier……..25

Fig 3.2.1: MAC- Fundamental diagram..…….28

Fig 3.2.2: Ripple Carry Adder……..29

Fig 3.2.3: Circuit diagram of register.……..30

Fig.4.1: RTL Schematic of Approximate Multiplier..……..31

Fig.4.2: RTL Analysis of MAC Using Approximate Multiplier..……32

Fig.4.3: Technology Schematic of Approximate Multiplier……32

Fig.4.4: Technology Schematic of MAC Using Approximate Multiplier..…….33

Fig.4.5: Simulation waveforms of approximate multiplier……..…34

Fig.4.6 : Simulation waveforms of approximate multiplier……34

LIST OF TABLES

Table - 3.1.1-The approximate 4-2 compressor truth table that we have presented……24

Table - 4.1 : Device utilization summary of approximate multiplier……..……..34

Table - 4.2 : Device utilization summary of MAC using approximate multiplier..…..……35

CHAPTER - 1

1.1 INTRODUCTION

In a number of fields, including artificial intelligence, computer vision, multimedia processing, image recognition, and digital signal processing (DSP), multipliers play a crucial role as essential arithmetic functional units. These applications often demand a significant number of multiplications, lead- ing to substantial power consumption, especially in mobile devices, presenting a challenge for implementation. Many studies have suggested ways to lower multiplier circuit power consumption in order to overcome this problem. When error tolerance is permissible or when applications are linked to human senses, one approach to mitigating multiplier power consumption is through the approximation of multiplication. Given the limited sensory capabilities of humans, precise computational outcomes are not always necessary, enabling the approximation-based decrease of cell space, time delay, and power consumption. Approximate multipliers come in two primary varieties: those utillizing dynamic voltage scaling to regulate the multiplier's time path and those that involve reworking specific multiplier circuits, such as the Wallace Tree Multiplier and Dadda Tree Multiplier, to alter operational properties. Previous research on rebuilding multipliers frequently suggested erroneous m:n compressors, in where m stands for inputs and n for outputs. During multiplication, these compressors were used to compress partial products, which resulted in high energy consumption and considerable path latency. The majority of earlier approximation multipliers had set output accuracy and power requirements. However, there is a need for dynamic adjustments in accuracy and power consumption for applications like artificial intelligence, whose requirements evolve over time. It's essential to note that achieving an adjustable multiplier architecture incurs additional hardware costs. In order to create a high-accuracy approximation multiplier, this work presents a high accuracy 4:2 compressor. Furthermore, a dynamic input truncation technique is introduced to modify the needed power and precision as necessary.

MAC'S DESIGN

The Multiplier-accumulator (MAC) device plays a crucial role in supporting various digital signal processing (DSP) applications. It empowers the microcontroller to handle signal processing tasks, including servo and audio control. Positioned as an execution unit within the CPU, the MAC device employs a 3-stage pipelined arithmetic architecture to optimize 8x8 multipliers. It is capable of processing both signed, fixed-point fractional input operands, and signed and unsigned integers.

The MAC unit serves three primary functions:

1. Signed and unsigned integer multiplication.

2.Perform multiply-accumulate operations, supporting signed, unsigned and signed fractional operands.

3.Execute other register functions, comprising an accumulator adder, multiplier, and adder within the MAC unit.

Given the high-speed requirements of DSP applications, adders are predominantly used in the MAC unit. To enable more multiply-accumulate operations, the memory collects inputs from its location to the multiplier.The output of the MAC unit is kept in a pertinent memory location. and the entire process must be completed within a single clock cycle due to specific constraints.

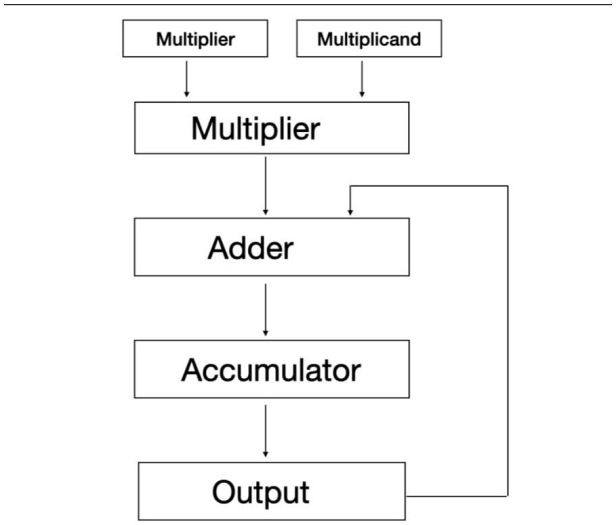

Fig 1.1. Basic MAC

1.2 PROJECT OBJECTIVES

The paper makes the following key contributions:

1.Proposing the construction of the recommended approximate multiplier, which is based on a high-precision approximate 4-2 compressor.

2. Presenting a straightforward circuit for error compensation intended to further minimize error lengths.

3.Presenting a dynamic input truncation method that allows for the adjustment of power and precision requirements in multiplication.

4.Developing a Multiply and Accumulate (MAC) unit that incorporates an approximate multiplier in this project.

1.3 PROBLEM DEFINITION

Designing a MAC unit with less power consumption, less delay and less area.Determine if the MAC unit needs to support different operational modes, such as fixed-point or floating-point arithmetic, or different types of multiplication (e.g., signed/unsigned, fractional).

1.4 FEATURES

Designing a MAC (Multiplier and Accumulator) unit involves considering several key features to ensure its functionality, efficiency, and performance.

- The MAC unit must perform accurate multiplication of input operands. This includes supporting various data types (e.g., integer, fixed-point, floating-point) and signed/unsigned numbers.
- The MAC unit should accumulate the results of multiple multiplications to produce a final output. This involves adding the results of each multiplication to an accumulator register.

- Design the MAC unit to minimize power consumption by employing low-power circuit techniques, such as clock gating, power gating, and voltage scaling. This is particularly important for energy-constrained applications and mobile devices.
- Optimize the hardware implementation of the MAC unit to minimize silicon area while meeting performance and functionality requirements. This involves architectural optimizations, efficient resource utilization, and layout optimizations.
- Decide on the data representation format for input operands, accumulator, and output. This includes choosing between binary, two's complement, or floating-point representations based on the application requirements.

1.5 STRUCTURE OF THESIS

CHAPTER-1: This chapter provides an overview of multipliers and highlights the significance of examining several varieties to comprehend their operation.

CHAPTER-2: This chapter deals with details regarding the results of the literature review, which helps to design a Mac unit and also includes tool that is used.

CHAPTER-3:The peripheral components of multipliers, such as carry propagation and performance issues, are examined in this chapter.Explains several methods and complex architectures for designing multipliers. In order to provide readers a thorough grasp of the functionality and design of architectures, this guide examines them using block diagrams and functional representations.

CHAPTER-4: The Mac design analysis results are summarized in this chapter. The practical implications and applications of the analysis are discussed, with a focus on how these results might be used in actual situations.

CHAPTER-5: The project's final conclusions are quickly summarized in this chapter, which also includes a simulation analysis of Mac design and approximate multiplier and its potential for future applications.

CHAPTER-2

2.1 LITERATURE SURVEY

"Comparison and extension of approximate 4-2 compressors for lowpower approximate multipliers," A.G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo

Since recursive multipliers (RMs) offer a large number of power-quality tuning options, they are categorized as a low-power multiplier class. This recursive topology's fundamental building blocks are 2x2 multipliers, while the majority of modern approximate recursive designs are built using 4x4 building blocks. As such, the design space exploration of AxRMs with 2x2 multipliers remains an unsolved research issue. High-performance, low-area 2-bit multipliers are needed to bring configurability and flexibility to the design of AxRMs[1]. This article proposes two approximate 2x2 multipliers that display a double-sided error distribution characteristic.The proposed design has a bounded error behaviour and delivers a 52% decrease in area and a 25% improvement in delay when compared to the current best-approximated 2x2 multiplier. Next, three 8x8 multipliers with varying accuracy are created by utilizing various arrangements of an approximate 2x2 multiplier. The most accurate design is AxRM1, which achieves a 50% improvement in mean relative error distance (MRED) over the best MRED-optimized design currently in use. AxRM3's MRED is comparable to that of the finest 2x2-based AxRM that came before it, known as MACISH[2]. However, because AxRM3 uses high-performance, low-power 2x2 multipliers to generate larger multipliers, it has 13% greater PDP than MACISH.Convolutional neural networks, a state-of-the-art error-tolerant application, use the suggested approximate multipliers. The best quality-power trade-off is offered by AxRM2, which achieves 32.64 percent power savings and 1.10 percent higher classification accuracy[3].

"Design and analysis of approximate 4-2 compressors for high-accuracy multipliers," by T. Kong and S. Li

Approximate multipliers can be used in multimedia, data recognition, image processing, and other error-tolerant applications with loose precision requirements. Such multipliers can provide a commensurate improvement in electrical performance at the expense of some precision[4]. In order to examine the accuracy and performance of previously proposed compressors, an analysis of their architectures is presented in this paper. Five high-accuracy approximation 4-2, with improved latency, area, power, and performance-accuracy tradeoff, are proposed in this article. Pro5 is derived from the modified sorting technique, whereas Pro1-Pro4 rely on the critical route optimization[5]. In this article, 8X8 and 16X16 multipliers are implemented using the approximate compressors that are suggested for TSMC 28 nm. According to the testing results, our designs have an approximate 18% delay, a 43%-52% decrease in area-delay product (ADP) when compared to the precise multiplier, and an ADP optimization of 20%-55% when compared to compressors with identical precision. This article uses applications for image blending and matrix multiplication to further confirm the effectiveness of the suggested compressors[6].

The study "Design and analysis of approximate compressors for multiplication" by A. Menimi, J. Han, P. Montuschi, and F. Lombardi

For digital processing at nanometric scales, approximation (or imprecise) computing provides a compelling paradigm. Computer arithmetic designs find special importance in inexact computing. The analysis and design of two new approximate 4-2 compressors for use in multipliers are the topics of this paper[7]. These designs rely on various compression features, allowing calculation imprecision (as indicated by the error rate and the so-called normalized error distance) to be balanced against circuit-based design merit metrics (transistor count, delay, and power consumption).For a Dadda multiplier, four distinct plans for making use of the suggested approximation compressors are put out and examined[8]. Comprehensive simulation outcomes are offered, and a presentation of the approximation multipliers' applicability to image processing is made. In addition, two of the proposed multiplier designs offer excellent image multiplication capabilities with respect to average normalized error distance and peak signal-to-noise ratio (more than 50 dB for the considered image examples). The results demonstrate that the proposed designs achieve significant reductions in power dissipation, delay, and transistor count when compared to an exact design[9].

"A majority-based imprecise multiplier for ultra-efficient approximate image multiplication," by F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad

In error-tolerant applications, approximate multipliers are utilized, which compromises result precision in order to reduce power consumption or delay. In this study, we use static segmentation to investigate approximation multipliers. In these circuits, each of the two n-bit operands yields a set of m contiguous bits (a segment of m bits). The two segments are fed into a small mÃ—m internal multiplier, the output of which is appropriately shifted to yield the desired result. We study signed and unsigned multipliers, and we suggest a novel segmentation strategy for the latter. Additionally, we offer straightforward and efficient correction methods that can lower hardware costs while considerably reducing approximation error[10]. We examine a thorough comparison with earlier suggested approximations of multipliers while taking into account a hardware implementation in 28 nm technology.According to the comparison, static segmented multipliers using the suggested correction method have the desirable property of both power vs. normalized mean error distance and power vs. mean relative error distance trade-off charts being on (or near) the Pareto-optimal frontier. For situations where their error performance is acceptable, these multipliers are therefore viable options. The outcomes attained for image processing and image classification applications support this[11].

"Design of ultra-low power consumption approximate 4-2 compressors based on the compensation characteristic," by H. Pei, X. Yi, H. Zhou, and Y. He

In some digital signal processing applications, approximate computing is used in a hesitant manner because it naturally tolerates inaccurate computing outputs. They use the approximation arithmetic blocks in them to enhance these circuits' electrical performance. One of the basic building elements of computer arithmetic is the multiplier. Furthermore, in order to speed up the partial product compression process, 4-2 compressors are frequently used in parallel multipliers. Three new approximate 4-2 compressors are proposed and applied to 8-bit multipliers in this research.Concurrently, with the suggested 4-2 compressors, an error-correcting module (ECM) is included to enhance the approximation multiplier's error performance[12]. This work presents a unique reduction of the approximate 4-2 compressor's outputs to one, resulting in further gains in energy-efficiency. The simulation results show that the approximation compressors UCAC1,

UCAC2, and UCAC3 accomplish 24.76%, 51.43%, and 66.67% decrease in delay, 71.76%, 83.06%, and 93.28% reduction in power, and 54.02%, 79.32%, and 93.10% reduction in area, respectively, when compared to the exact 4-2 compressors. Additionally, the average power consumption of 8-bit multipliers is reduced by 49.29% when these suggested compressors are used[13].

Using novel approximate compressors as a basis, D. Esposito, A. G. M. Strollo, E. Napoli, D. de Caro, and N. Petra developed an approximation multiplier.

An emerging trend in digital design is approximate computing, which trades off accuracy in processing for increased speed and power efficiency. In order to construct efficient approximation multipliers, this study suggests new approximate compressors and a method to take use of them. We have created approximation multipliers for a number of operand lengths using a 40-nm library by applying the suggested method[14]. When compared to previously presented approximated multipliers, the suggested circuits offer superior power or speed in exchange for a higher degree of goal precision. The research also presents applications to adaptive least mean squares filtering and image filtering.

In “Low power compressor-based approximate multipliers with error correcting module," U. Anil Kumar, S. K. Chatterjee, and S. E. Ahmed

The scientific literature, which suggests many circuits constructed using approximate 4-2 compressors, shows a great deal of interest in approximation multipliers. The challenge for the designer who wants to employ an approximation 4-2 compressor is choosing the appropriate topology because there are so many alternatives available. We provide a thorough analysis and comparison of approximate 4-2 compressors that have been previously suggested in the literature in this research. In order to investigate a total of twelve distinct approximate 4-2 compressors, we also offer a novel approximate compressor. Utilizing the circuits under investigation, 8x 8 and 16x 16 multipliers are designed and built in 28nm CMOS technology.We examine two multiplier configurations, both signed and unsigned, with varying degrees of approximation for every operand size. Our analysis shows that there is no one optimal approximate compressor topology since the optimal answer varies on the needed precision, the multiplier's signedness, and the error metric taken into account.

"Design of an energyefficient approximate compressor for error-resilient multiplications," by X. Yi, H. Pei, Z. Zhang, H. Zhou, and Y. He[15].

Approximate computation is utilized since accurate results are not needed in most practical applications. System performance parameters like area, power, and speed can be increased by approximating calculations. In this study, an approximation circuit constructed by altering the circuit architecture but not the circuit operation is proposed. Through the use of Wallace tree reduction, AND-OR logic approximation, and 3:2 inexact additive designs for partial product generation and addition, we present an approximate multiplier using these approximate circuits. We demonstrate the entire suggested notion in this paper using an 8X8 bit multiplication as an example. Furthermore, the area and delay will be significantly improved by the suggested multipliers[16].

"Multipliers with approximate 4-2 compressors and error recovery modules," by M. Ha and S. Lee

One often utilized operation in approximate computing techniques for high performance and low power computing is approximate multiplication. An approximation 4-2 compressor can be used to produce power-efficient circuits for approximate multiplication. This letter offers a new design that incorporates an error recovery module and is based on an earlier approximate 4-2 compressor concept. Compared to other proposed 4-2 compressor-based approximate multiplier systems, the suggested design is more accurate, uses less hardware, and uses less power even with the added error recovery module[17].

"Energy and area efficient imprecise compressors for approximate multiplication at nanoscale," by M. Ahmadinejad, M. H. Moaiyeri, and F. Sabetzadeh

A new paradigm for creating nanoscale integrated circuits that are energy-efficient is approximate computing. In order to create simpler logic functions with fewer output mistakes, we propose efficient imprecise 4:2 and 5:2 compressors in this study by altering the truth table of the exact compressors[18]. The suggested method produces inaccurate compressors that perform better than their predecessors and need a great deal fewer transistors. Furthermore, the suggested

imprecise compressors serve as the foundation for the creation of effective approximation multipliers. One of the most advanced industrial technologies, the FinFET, is used in the circuit design process. HSPICE at the 7 nm technology node is used for simulation.In addition, MATLAB is used to compute two crucial quality metrics: the mean structure similarity index meter (MSSIM) and the peak signal to noise ratio (PSNR). These approximate multipliers are employed in image processing applications such as image multiplication, sharpening, and smoothing[19]. The outcomes show notable gains in terms of energy-efficiency, transistor count, and performance over the other precise and approximation designs that are already in use. In compared to the previous designs, the suggested 4:2 and 5:2 compressors improve the area by 60% and 75% and the power delay product (PDP) by an average of 59% and 68%. Furthermore, for approximative computing, the suggested multipliers offer a noteworthy trade-off between hardware efficiency and accuracy.When evaluating image quality (based on PSNR and MSSIM) and energy efficiency, the suggested approximation multiplier which employs both imprecise 4:2 and 5:2 compressors improves the figure of merit by 2.35 times over its predecessors[20].

2.2 TECHNOLOGY

SOFTWARE REQUIREMENT

XILINX

The VHDL/VERILOG designers use Xilinx software to carry out the Synthesis operation. On FPGA, any simulated code may be configured and synthesized. The conversion of HDL code into a gate level net list is known as synthesis. It is a crucial component of the design flows used today.

Click the XILINX ISE icon to launch the ISE Software.Open a New Project and you should see the properties listed below.It is possible to alter the family, device, and package adjustments if the design requires a lot of LUTs.

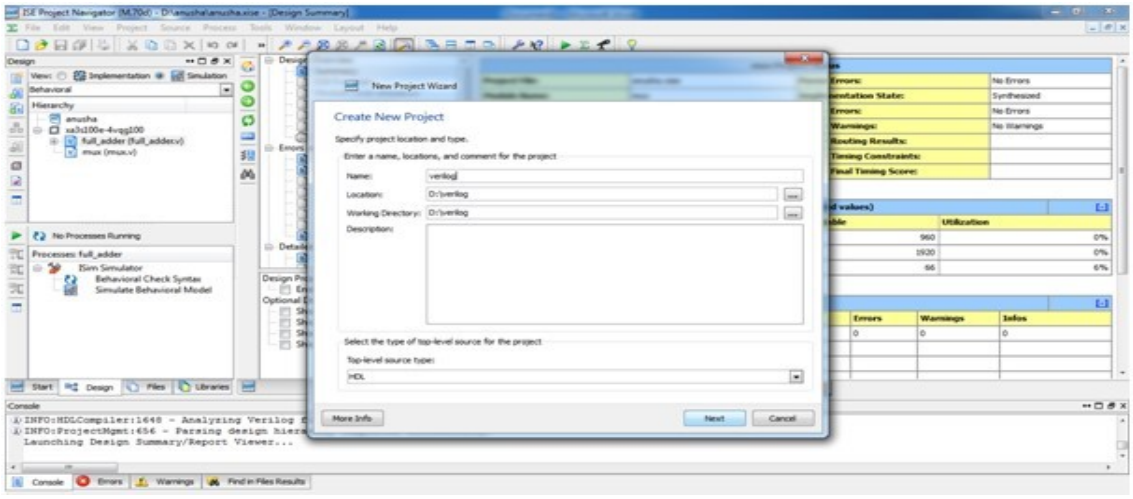

Fig .2.2.1.Create new folder for design

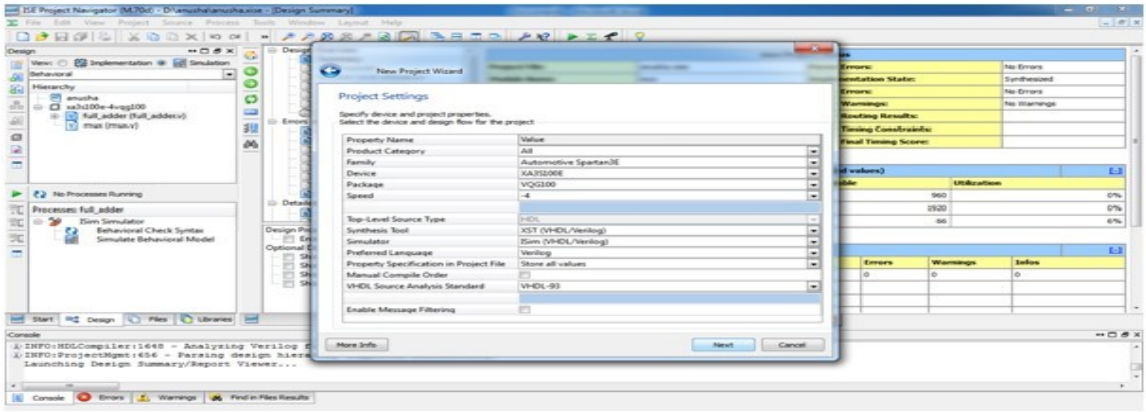

Fig 2.2.2. Set family and device before design a project

Make an HDL source and format all of the inputs, outputs, and buffers (if needed). It offers a window for writing the HDL code that will be synthesized.

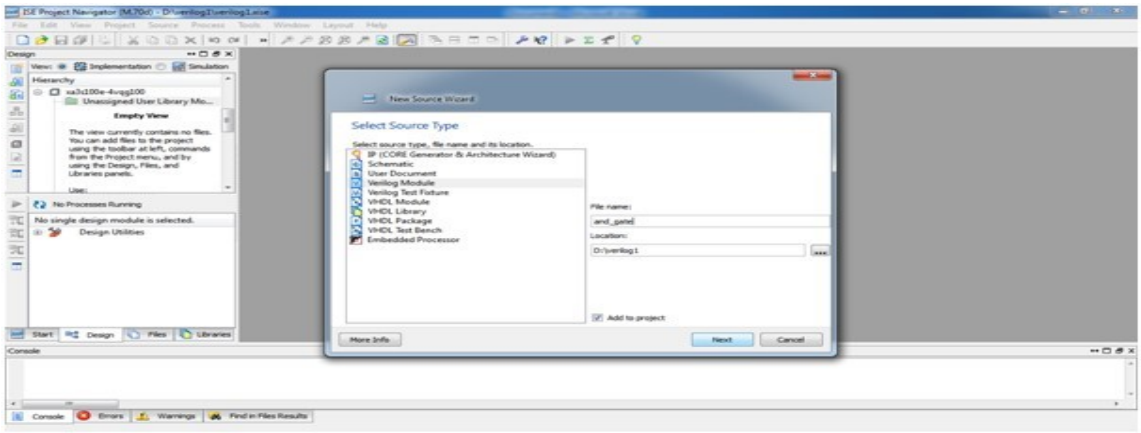

Fig 2.2.3. Ready to design a project and create module name(.v files names)for design

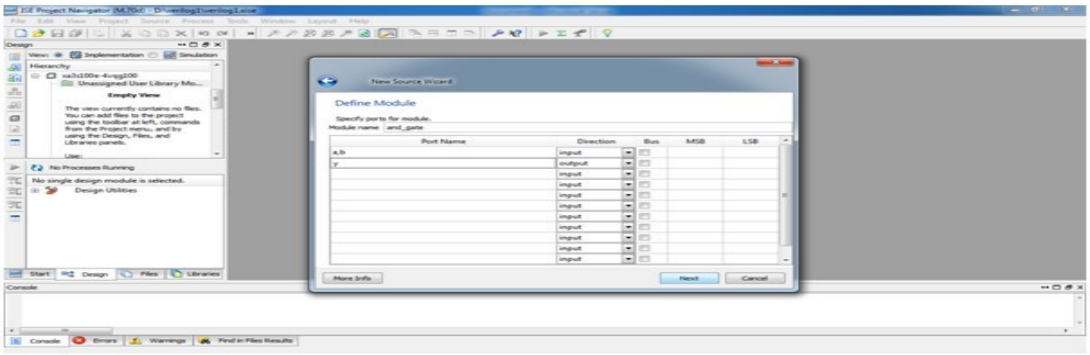

Fig 2.2.4. Declaration of input and output ports with their bit lengths and The schematic was created by its ports

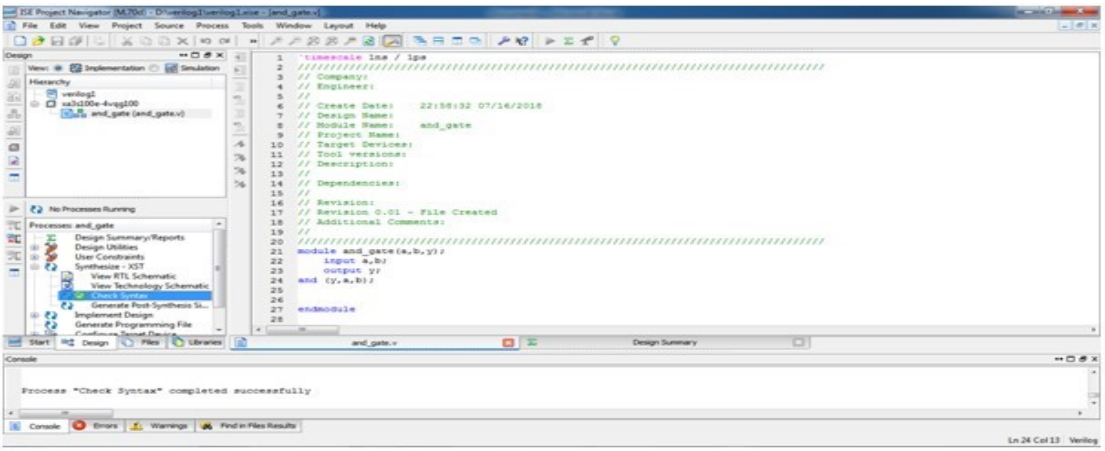

Fig 2.2.5: Ready to write the code for design and check the syntax

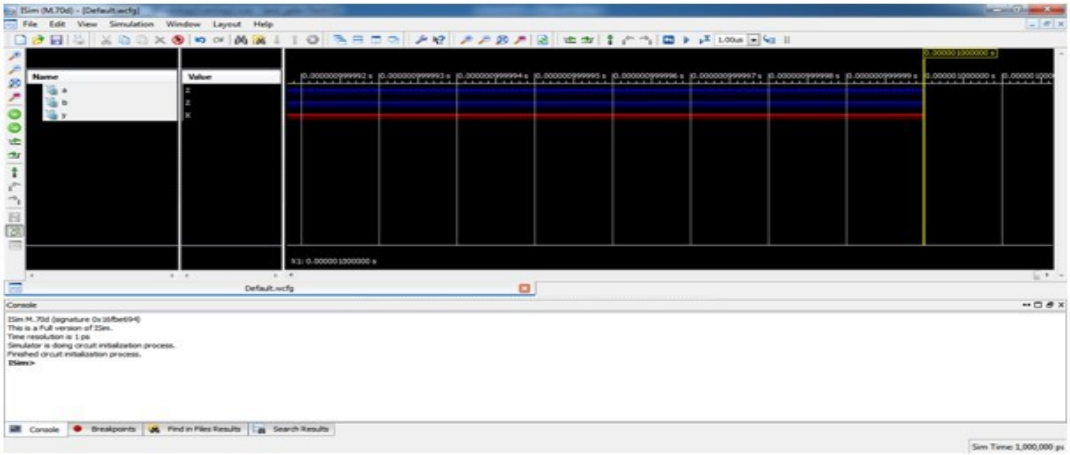

Fig 2.2.6: Simulation of design to verifying the logics of design.

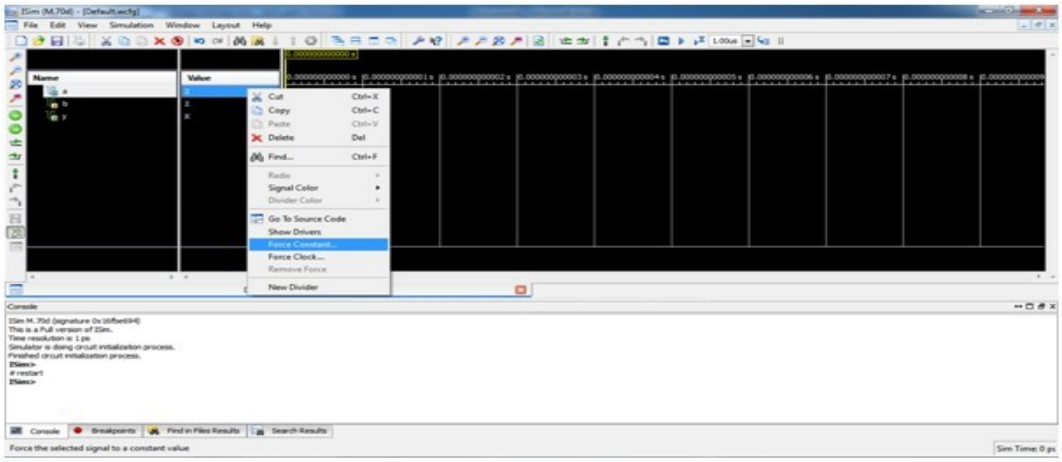

Fig 2.2.7: Apply inputs through force constant or force clock for input signals

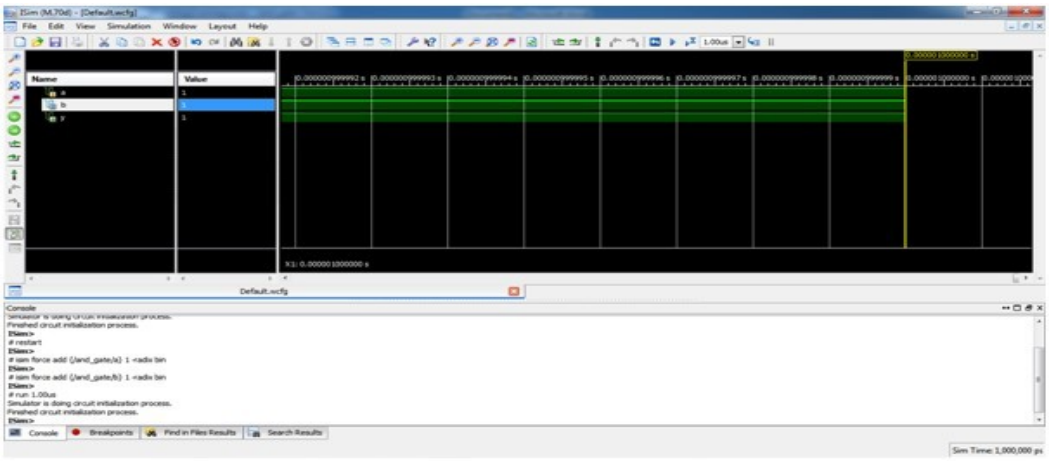

Fig 2.2.8: Run the design after applying inputs

CHAPTER-3

3.1 EXISTING METHODS

An error correction circuit and the current, very accurate 4-2 compressor are introduced. It then moves on to the dynamic input truncation step, which builds the adjustable multiplier. Lastly, it describes the general design of the suggested approximation multiplier and the reasons behind its applicability to CNNs.

A. PROPOSED FLOW AND APPROXIMATE MULTIPLIER

Using 2-input AND gates, correct partial products are first generated. The accurate compressors then compress these partial products. Ultimately, the output is produced by precise adders adding the compressed partial products.

B. PROPOSED HIGH-ACCURACY 4-2 COMPRESSOR

A low-power, high-accuracy approximate 4-2 compressor is suggested. FIGURE depicts the suggested 4-2 approximation compressor.

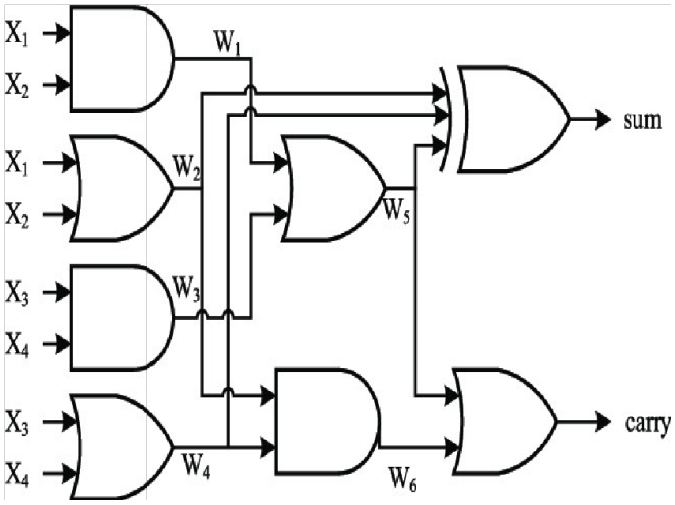

Fig 3.1.1. Gatelevel approximate compressor.

The following is a description of the proposed 4-2 approximation compressor's design. Equations below are used to generate W1 âˆ¼ W4 from four inputs, X1 âˆ¼ X4. The carry bit in the suggested compressor is intended to always be created correctly since an incorrectly computed carry bit has a larger error distance than the sum bit, meaning that an inaccurate carry bit creates twice the ED of that produced by an incorrect sum bit. There are three situations where the carry bit becomes 1. X1 and X2 are the same. Another is that both X3 and X4 equal 1. The third is that one of X1 or X2 and one of X3 or X4 are both 1.

W1 =X1 ANDX2

W2 =X1 ORX2

W3 =X3 ANDX4

W4 =X3 ORX4

W5 =W1 ORW3

W6 =W2 ANDW4

Carry = W5 OR W6

Sum = W5 XOR W2 XOR W4

Table:3.1.1-The approximate 4-2 compressor truth table that we have presented.

X_3	X_3	X_2	X_1	carry	sum	diff.
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	1	0
1	0	0	0	0	1	0
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	1	1	-1

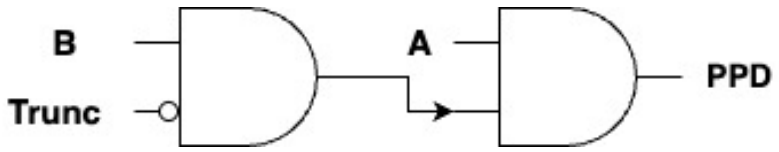

Fig 3.1.2 Modified Partial Product.

C. DYNAMIC INPUT TRUNCATION

Through the employment of two 2-input AND gates and a dynamic input truncation approach, a customizable approximate multiplier may be achieved at runtime. to generate a partial product, the equation for which is displayed. where the multiplier is B and the multiplicand is A. To decide whether to truncate the partial product, one uses the Trunc signal.The partial product is trimmed to zero if the Trunc value is 1. Put another way, the Trunc signals preserve energy by setting the PPDs in the multiplications to zeros. Put differently, we can consider the Trunc signals to be acting as a means of turning off the hardware units in the respective columns.

PPDij = (Trunc AND Bi)ANDAj

Our solution is to incorporate an additional AND gate to share gates and lower hardware costs because every bit in an 8 × 8 multiplier corresponds to 8 bits in the multiplicand. For instance, ppd01 and ppd00 are computed as trunc0·B0·A1 and trunc0·B0·A0, respectively.In this scenario, three 2-input AND gates are required, and trunc0·B0 can be precomputed to serve as a mask.

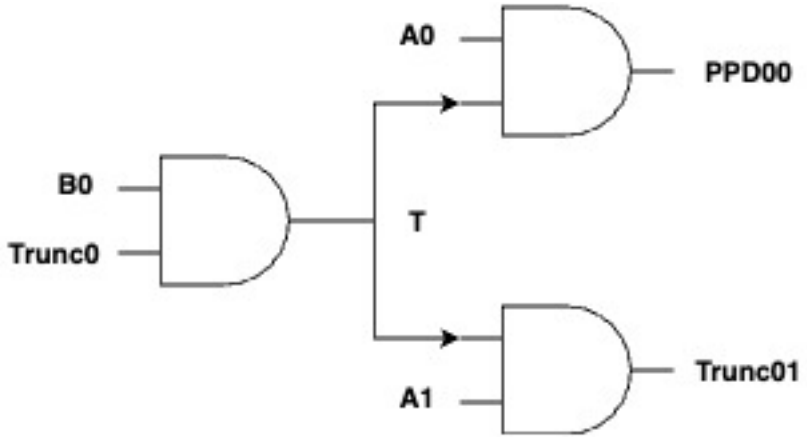

Fig.3.1.3 An example of gate sharing to minimize the number of gates

D. APPROXIMATE MULTIPLIER

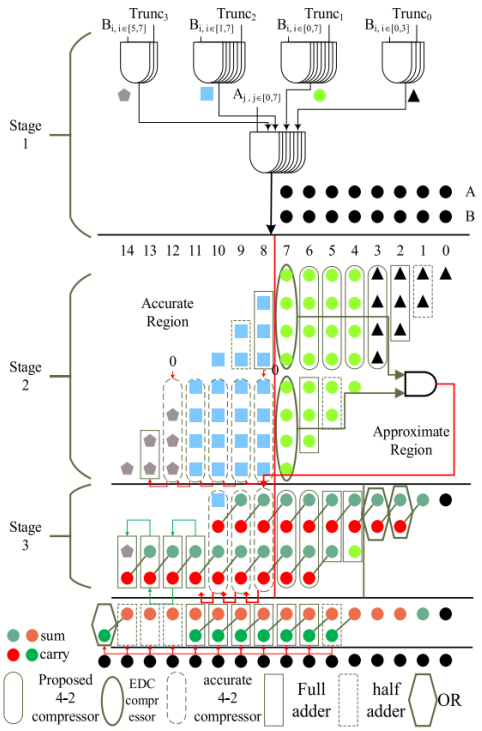

Fig 3.1.4 Proposed approximate multiplier

Two 2-input AND gates generate each partial product in the first step. hardware expenses can be further decreased by utilizing the gate sharing technique. The Trunc signal can be used to determine the accuracy of the created partial product, depending on the needs. Our suggested approximate multiplier uses a 4-bit Trunc signal, which we refer to as the "3-4-4-4 partition," with each bit controlling several partial product columns in order to improve control efficiency and lower hardware costs.specifically, each bit from MSB to LSB to control column 14th~12th, 11th~8th, 7th~4th and 3rd~0th respectively, corresponding to the color of khaki, sky blue, green and black in Stage 2 . In the event when the Trunc(3-0) is 01012, for instance, the 14th ~ 12th and 7th ~ 4th columns are accurate, while the 11th ~ 8th and 3rd ~ 0th columns are truncated. The partitioning of the columns provides several possibilities for regulating Trunc signals, enabling users to adapt the suggested multiplier to suit their needs. We have experimented with several partitions in multiple trials; the findings indicate that the 3-4-4-4 and 3-3-3-3-3 partitions maintain the equilibrium between accuracy, power efficiency, and area overhead. It is more flexible to adjust the quantity of electricity to be saved and accuracy losses the finer the split. On the other hand, it will experience high area overhead in return. Consequently, we employ 3-4-4-4 partition in all of our trials. We'll contrast the 3-4-4-4 partition's CNN findings with the 3-3-3-3-3 partition in more detail. The incomplete product compression stages are displayed in the second stage.Following generation, the partial products are split into two regions: the exact region, which is column 14th-8th, and the approximation region, which is column 7th-0th. The most logical half-half separation determines the division of the region into correct and approximate halves. For instance, there will be a noticeable loss in accuracy if we perform a 30-70 split and the approximation multiplier performs too many calculations. However, the impact of approximating the computation for power reduction will be minimal if we choose a 70-30 split. We employ accurate 4-2 compressors to compress the partial products in the accurate zone since their weight is greater and more significant there. Conversely, we compress the partial products in the approximate region using our proposed approximation 4-2 compressors and error compensation circuit. In the third stage, we create results using OR gates in columns 3rd-0th, ignoring carry

propagation because they are close to LSB and their errors have less of an impact on the outcome. In order to decide whether to generate the compensation bit or not, we use the EDC, or a single AND gate, to identify faults in the second stage. To compress the partial products in the remaining columns, we employ the proposed approximate 4-2 compressors, exact 4-2 compressors, full adders, and half adders.The last two partial product rows are obtained when the third stage is complete, and they are added up using precise adders to yield the final results.

3.2 PROPOSED METHODOLOGY

A.DESIGN OF MAC USING TRUNCATED MULTIPLIER

A MAC unit is a common component found in numerous digital signal processing systems. that use accumulation and multiplication. It also operates digital DSP systems with high speed. DSP is utillized in a variety of procedures, including convolution, filtering, and inner products. DSP techniques often utillize nonlinear functions such as Discrete wavelet transforms (DWT) and Discrete cosine transforms (DCT). The whole speed of the arithmetic computations, which includes addition and multiplication, is established by the speed of execution and the performance of the full calculation since addition and multiplication are efficiently finished by cyclic application of addition and multiplication. Procedures that multiply and aggregate are unique to digital filters.Next, high- speed filtering and other special processing units for DSP applications are made possible by the MAC unit's basic features. To reduce the stress on the CPU, the MAC unit processes each piece of data individually, operating entirely independently of the CPU.One of the best uses is the optical communication system, which is entirely dependent on the DSP and needs a large amount of data to process digital data quickly. For the Fast Fourier Transform, the addition and multiplication operations are also necessary (FFT). The 16-bit MAC unit requires additional memory because it can handle a lot of bits. Its basic components are an accumulator unit that holds the total of the preceding sequential product phrases and a multiplier. The matching memory address that is attached to the multiplier block provides the inputs for the MAC unit.

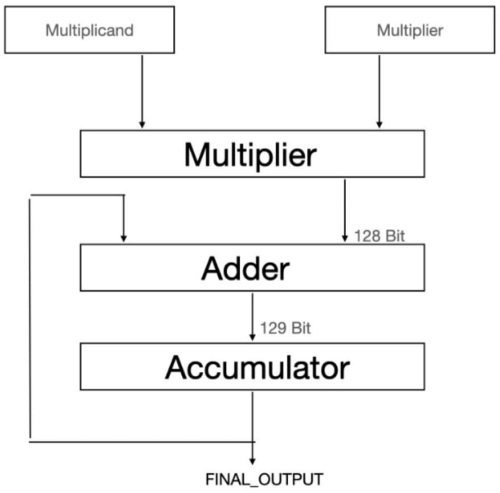

Fig 3.2.1 MAC- Fundamental diagram

B.MAC OPERATION

Furthermore, the Multiply-Accumulate (MAC) operation is important in a variety of applications related to multimedia information processing and other domains, outside of its relevance in digital signal processing (DSP). As was previously mentioned, the register/accumulator, multiplier, and adder comprise the MAC.In this work, a Vedic multiplier was used. The matching memory address that is attached to the multiplier block is where the MAC inputs are obtained. This is beneficial for DSP that is 16-bit.We can connect the input from the position of the 16-bit memory. After processing the 16-bit input, the multiplier produces a 16-bit output when the computation is successful. These 16-bit multiplier outputs are then fed into an adder to perform additional addition operations. An extra bit is added for carry to the 16-bit output that the adder unit generates. The pertinent data in the accumulator register is then linked to this output. A Parallel in parallel out (PIPO) register method is used by the accumulator register, where input bits are received and output bits are generated simultaneously in parallel. Large sets of bits can be handled well using this PIPO technique, making it easier to generate adder output values in parallel. The output of the accumulator register receives inputs to equivalent adders.

C.RIPPLE CARRY ADDER

A Ripple Carry Adder (RCA) is an essential building block in digital circuits employed in computer processors and similar digital systems for binary number addition. Its name, ”ripple

carry," stems from the way the carry bit, produced by adding each pair of bits, sequentially propagates through successive stages of the adder. The truth table of a complete adder, the fundamental building block of a ripple carry adder (RCA), can be used to obtain the Sum (S) and Carry-out (Cout) equations for a single stage of the RCA. For a single full adder stage, the sum (S) and carry-out (Cout) can be expressed as follows:

Sum = A xor B xor Cin

Cout = (AB)+(ACin)+(BCin)

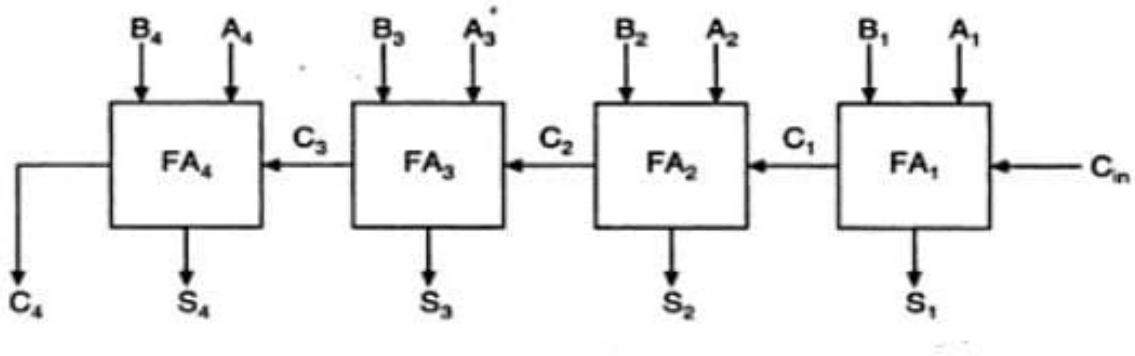

Fig 3.2.2 Ripple Carry Adder

The fundamental logic processes needed to calculate the sum and carry out for a single stage of a ripple carry adder are represented by these equations. These equations are applied iteratively to every step in a multi-bit RCA, where the carry-in of one stage is influenced by the carry-out of the previous one.

D.REGISTER

A register in digital electronics is a type of data storage element used to hold binary information temporarily. It's commonly implemented using flip-flops or other similar circuitry. Registers play a crucial role in digital systems, such as microprocessors, where they are utillized for various purposes including data storage, data movement, and control signal generation. The "D" register, sometimes referred to as a "data register", is an essential part of microprocessor designs and digital systems. Its main purpose is to store binary data temporarily.It contains binary data that is being moved, processed, or altered across several system components.The D register can hold

operands, intermediate results, or calculation final outcomes during arithmetic and logic operations. It offers a workspace for carrying out multiplication, division, addition, and subtraction operations.It facilitates the processing of several instructions at once by segmenting the instruction execution process into sequential steps.In digital systems and microprocessor designs, the D register is essential for enabling data processing, storage, and transfer. Because of its adaptability and flexibility, It is an essential part of the design of a wide range of digital systems and circuits.

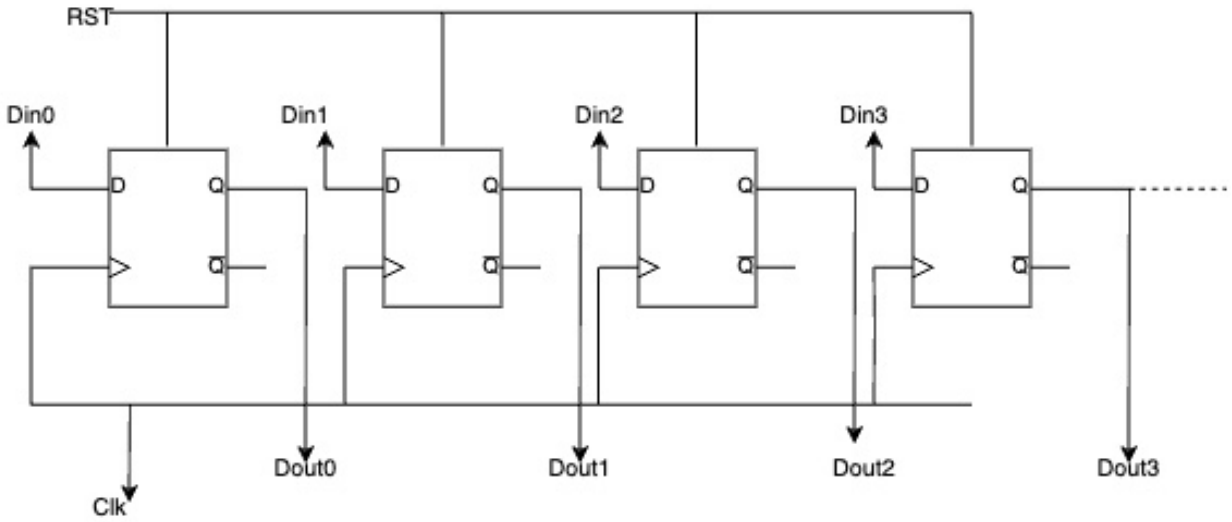

Fig 3.2.3 Circuit diagram of register

CHAPTER - 4

RESULT &CONCLUSION

4.1 RESULTS:

A. RTL Schematic :

The architecture's blueprint, known as the register transfer level schematic, or RTL schematic for short, is used to contrast the current, intended design with the ideal architecture that still needs to be created.The working summary is created by utillizing the HDL language to transfer the description or summary of the architecture into a coding language, such as verilog or vhdl. The RTL diagram even specifies the internal connection blocks for further analysis.The graphic below shows the planned architecture's RTL schematic diagram.

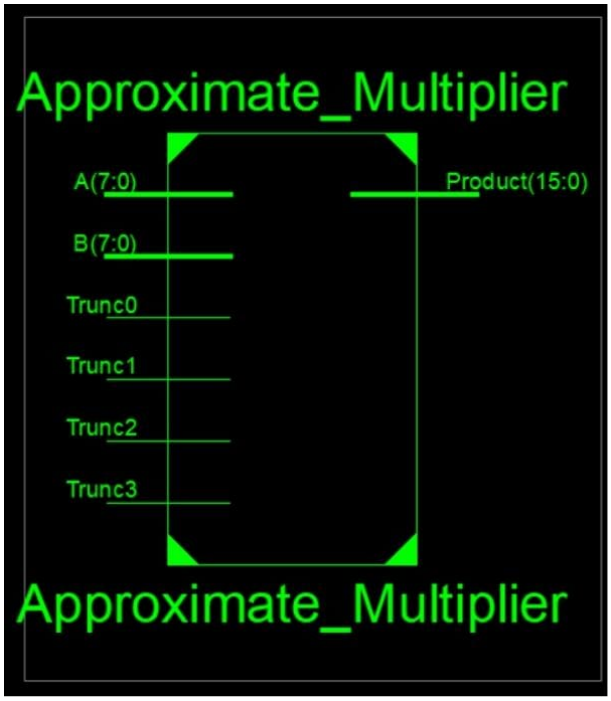

Fig.4.1: RTL Schematic of Approximate Multiplier.

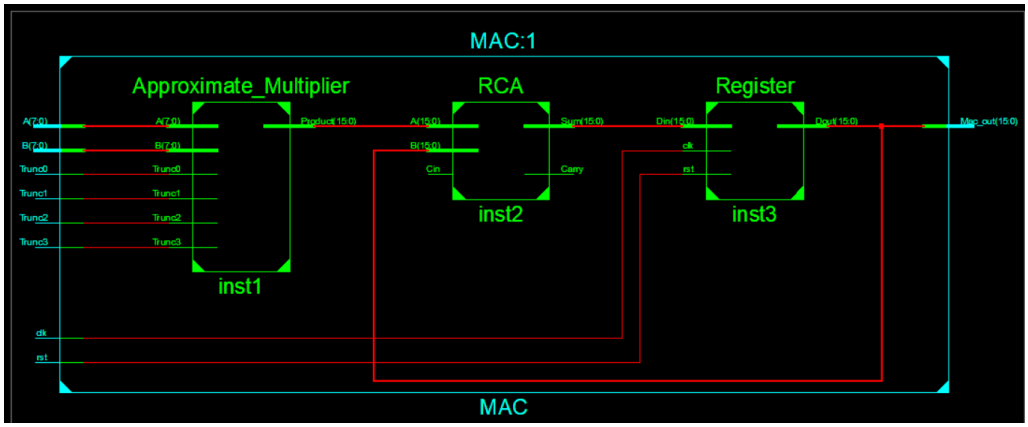

Fig.4.2: RTL Analysis of MAC Using Approximate Multiplier.

B. Technology Schematic

The architectural layout is depicted in the technology diagram using the LUT (Look-Up Table) format, where LUTs serve as a measure or parameter in VLSI (Very Large Scale Integration) to gauge the design of the architecture. These FPGA LUTs, often referred to as square units, illustrate the memory allocation of the code.The LUT technology schematic serves as a reference for identifying potential issues or errors in the logic implementation. Designers can trace signal paths, verify connections, and diagnose any anomalies that may arise during operation.

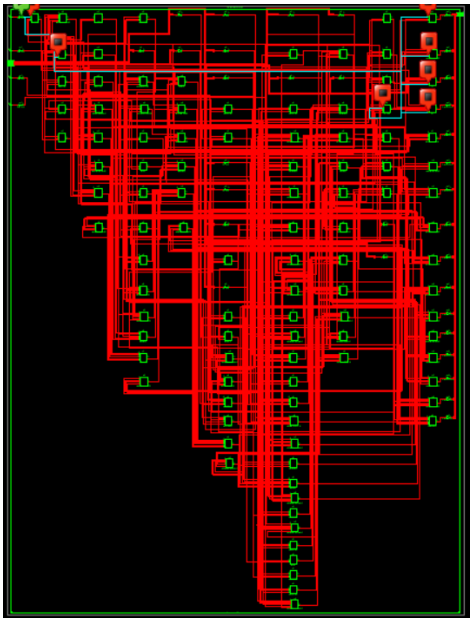

Fig.4.3 Technology Schematic of Approximate Multiplier.

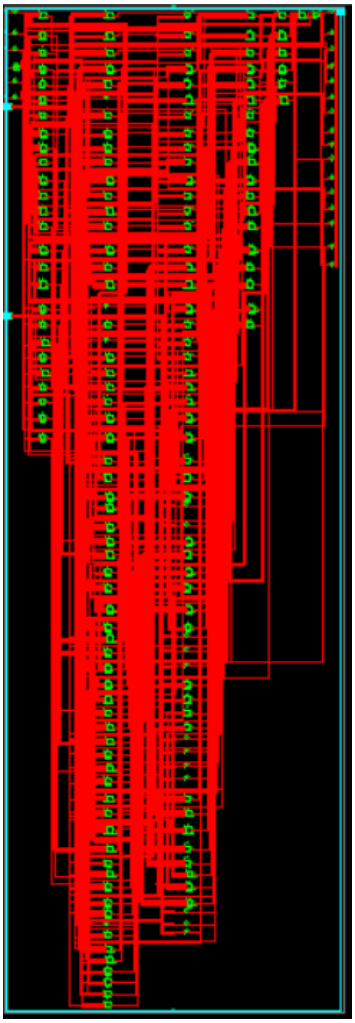

Fig.4.4 Technology Schematic of MAC Using Approximate Multiplier.

C. Simulation

The simulation is the procedure that is thought of as the final check to make sure that everything works as planned, while the schematic confirms the connections and blocks as shown in figure 13 and figure 14. To access the simulation window, navigate from implantation to simulation on the tool's home screen. Wave patterns represent the output that is limited by the simulation window.This is where its flexibility in offering many radix number systems comes in handy. Additionally, simulations offer insights into system behaviour under different conditions, enabling thorough validation before actual deployment.

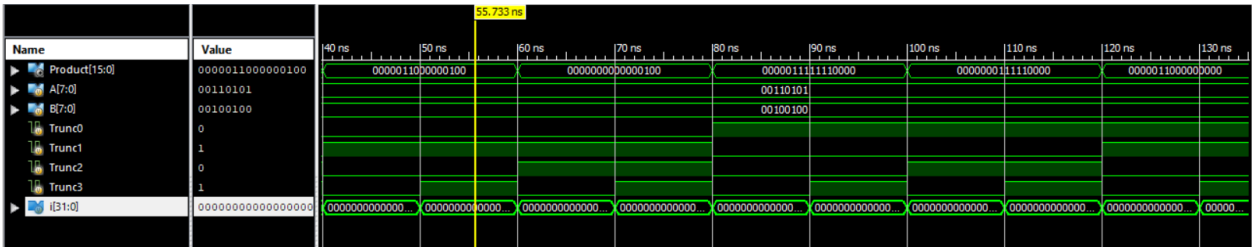

Fig.4.5: Simulation waveforms of approximate multiplier.

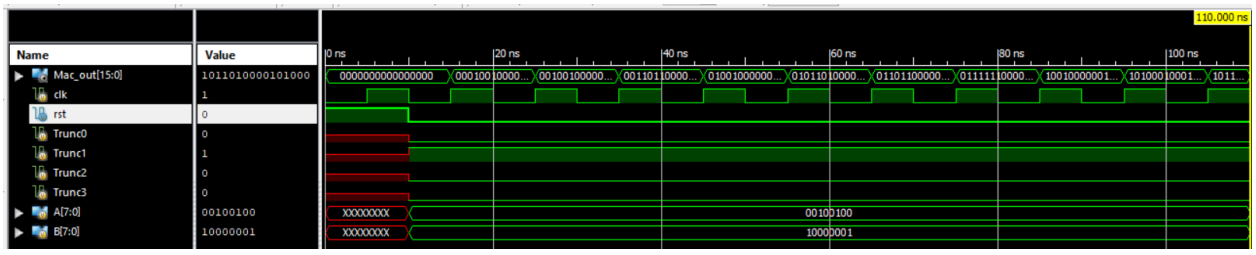

Fig.4.6 : Simulation waveforms of approximate multiplier.

D. Parameters

In VLSI, area, delay, and power are taken into account. These variables can be used to compare various architectures. Here, the value that considers both area power and latency is derived using the tool XILINX 14.7. Verilog is the HDL language utilized.

Table - 4.1 : Device utilization summary of approximate multiplier

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	103	46560	0%
Number of fully used LUT-FF pairs	0	103	0%
Number of bonded IOBs	36	240	15%

Delay: 7.673ns

Table - 4.2 : Device utilization summary of MAC using approximate multiplier

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	16	93120	0%
Number of Slice LUTs	145	46560	0%
Number of fully used LUT-FF pairs	16	145	11%
Number of bonded IOBs	38	240	15%
Number of BUFG/BUFGCTRLs	1	32	3%

Minimum period: 2.179ns (Maximum Frequency: 458.926MHz)

Minimum input arrival time before clock: 8.410 ns

Maximum output required time after clock: 0.735 ns

4.2 CONCLUSION:

This study introduces a new method involving approximate 4:2 compressor designs coupled with an approximate multi- plier for constructing MAC (Multiplier-Accumulator) units. The aim is to cater to different precision requirements. We describe an extremely accurate Approximate 4-2 Compressor and suggest a flexible approximation multiplier that may dynamically truncate partial products. Additionally, we propose a MAC unit that can adjust power consumption and accuracy levels for multiplication tasks based on user requirements during runtime.In essence, designing an approximation multiplier with absolute advantage is quite difficult, and the best solution is usually the one that best fits the intended use. We provide a contender with a competitive error-electrical performance tradeoff in our approximate multiplier design.

CHAPTER - 5

5.1 FUTURE SCOPE:

In order to obtain more precise or mathematical representations of the relationship between hardware prices, accuracy, and power usage, future study will go deeper into the examination of various partition techniques. It is evident from the current study that in order to get satisfactory results, the suggested adjustable approximate multiplier needs distinct Trunc signals for various networks, or more precisely, different convolutional layers. Future research will give this component more weight. Examining the characteristics of various convolutional layers to determine the best set of parameters based on the types of layers is a workable solution. In that manner, the users can apply the suggested multipliers in any CNN after discovering those pre-examined factors.

5.2 REFERENCES:

[1] O. Sentieys, D. Me nard, and A. Bosio, eds. Techniques for Approximate Computing: From the Component to Application Level. Springer, Cham, Switzerland, 2022. [Online]. The following URL is available: https://link.springer.com/book/10. 1007/978-3-030-94705-7

[2] A flexible approximate multiplier designed for use in quantized Convolutional Neural Network (CNN) applications is presented by C. Guo, L. Zhang, X. Zhou, W. Qian, and C. Zhuo in the Proceedings of the 25th Asia South Pacific Design Autom. Conf. (ASP-DAC), January 2020, pages 235–240.

[3] Low-power high-speed multiplier for error-tolerant application, K. Yin Kyaw, W. Ling Goh, and K. Seng Yeo, Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1-4.

[4] ”A truncated multiplier with programmable functionality and low power consumption,” by M. de la Guia Solaz, W. Han, and R. Conway, IEEE Trans. Circuits Syst., vol. 59, no. 11, pp. 2555–2568, Nov. 2012.

[5] Manohar, P. S., Rohan, B., Jamal, K., Kumar, M. O., Reddy, B. V. (2023, May). Ramana, P. V. S. Using a 2-bit approximate adder, a Carry Look Ahead adder is implemented. Applied

Artificial Intelligence and Computing, 2nd International Conference (ICAAIC), 2023, pp. 1543–1547. IEEE.

[6] In December of 2021, Kumar, U. A., Ahmed, S. E., Jamal, K., Kumar, S., and Chintakunta, R. K. An approximation of multiplier architectures for applications that are error resilient. Pages 89–92 in the IEEE International Symposium on Smart Electronic Systems (iSES) 2021. IEEE.

[7] ”4:2 carry-save adder module,” A. Weinberger IEEE Technology Development Bulletin, vol. 23, no. 8, pp. 3811–3814, 1981.

[8] ”Estimated multiplier with high precision and error correction,”authored by C.-H. Lin and I.-C. Lin, was published in the IEEE 31st International Conference on Computer Design (ICCD) Proceedings, October 2013, pages 33–38.

[9] ”Development of approximate multiplier utilizing innovative dual-stage 4:2 compression techniques.” P. J. Edavoor, S. Raveendran, and A. D. Rahulkar, IEEE Access, vol. 8, pp. 48337–48351, 2020.

[10] ”Design of voltage-scalable meta-functions for approximate computing,” by D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, was published in Proc. Design, Autom. Test Eur., Mar. 2011, pp. 1–6.

[11] Jamal, K., and Nayeema, S. (2013). Using FSM and Verilog coding techniques, a MAC transmitter is designed and implemented to transmit UDP packets. 338–342, Int. J. Eng. Res. Appl., 3(1).

[12] In October 2020, Supritha, B., Mannem, K., Reddy, B. V., Jamal, K., Kumar, M. O. ALU with a customised booth multiplier that operates quickly and effectively. IoT in Social, Mobile, Analytics, and Cloud: Fourth International Conference on I-SMAC (2020) (pp. 1196-1200). IEEE.

[13] “CNN inference using a preprocessing precision controller and approximate multipliers with various precisions," I. Hammad, L. Li, K. El-Sankary, and W. M. Snelgrove, IEEE Access, vol. 9, pp. 7220-7232, 2021.

[14] A reconfigurable approximate multiplier for quantized CNN applications is presented by C. Guo, L. Zhang, X. Zhou, W. Qian, and C. Zhuo in the Proceedings of the 25th Asia South Pacific Design Autom. Conf. (ASP-DAC), January 2020, pages 235-240.

[15] Dual-quality 4:2 compressors for use in dynamic accuracy adjustable multipliers, O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, IEEE Trans. Very Large Scale Integer. (VLSI) Syst., vol. 25, no. 4, pp. 1352-1361, Apr. 2017.

[16] “A low-power high-speed accuracy controllable approximate multiplier design," by T. Yang, T. Ukezono, and T. Sato, was published in Proceedings of the 23rd Asia South Pacific Design Autom. Conf. (ASP-DAC), January 2018, pp. 605-610.

[17] "RoBa multiplier: A rounding-based approximate multiplier for highspeed yet energy-efficient digital signal processing," by R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, and M. Pedram Very Large Scale Integral (VLSI) Systems Technical Transactions, IEEE, vol. 25, no. 2, pp. 393-401, Feb. 2017.

[18] Wallace, C. S. "A proposal for a fast multiplier," In February 1964, IEEE Trans. Electron. Comput., vol. EC-13, no. 1, pp. 14-17.

[19] Low-power high-speed multiplier for error-tolerant application, K. Yin Kyaw, W. Ling Goh, and K. Seng Yeo, Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1-4.

[20] “A flexible low power DSP with a programmable truncated multiplier," by M. de la Guia Solaz, W. Han, and R. Conway, IEEE Trans. Circuits Syst., vol. 59, no. 11, pp. 2555-2568, Nov. 2012.

Course Outcomes (Mini and Major Projects) Project Objectives (PrOs)	Formulate hypothesis for the problem statement with sound technical knowledge from selected project domain. (CO1)	Design Engineering Solution to the problem statement with systematic approach. (CO2)	Analyse and develop an efficient solution for implementation of the project. (CO3)	Apply the theoretical concepts while providing solution to the problem statement with teamwork and multidisciplinary approach. (CO4)	Demonstrate professionalism with ethics while preparing and presenting the project work. (CO5)
To propose a high accuracy approximate 4:2 compressor that can used to construct the proposed approximate multiplier.	X	X	X	X	
To design a simple error compensation circuit to further reduce the error distance.		X	X	X	X
To propose a dynamic input truncation techinique that can be used to adjust accuracy and power required for multiplication. .	X	X	X		
To apply the proposed X technique is to suitable for CNNs as a power consumption.			X		
To implement the proposed approximate multiplier for design of a Mac unit	X	X	X	X	

S.No	Project Outcomes	CO	PO	Blooms Level
1	Proposed a high accuracy approximate 4:2 compressor that used to construct the proposed approximate multiplier.	CO 1	PO1,PO2,PO4	Understand, Analyse,Evaluate
2	Designed a simple error compensation circuit to further reduce the error distance.	CO 2	PO3,PO4,PO5	Analyse,Create
3	Proposed a dynamic input truncation technique that can be used to adjust accuracy and power required for multiplication.	CO 3	PO3,PO5,PO8,PO, PO10	Apply,Analyse, Create
4	Applied the proposed technique and is suitable for CNNs as a power consumption.	CO 4	PO8,PO9,PO11	Create
5	Implemented the proposed approximate multiplier for designing of mac unit	CO 5	PO9,PO11,PO12	Create, Apply, Evaluate