International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.4 (2025) pp. 8574-8584
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Power – Efficient Gaussian Filter Architecture Utilizing Approximate 4:2 Compressors for Edge Detection

V Bharathi Devarakonda[1]*, K Babulu [2], M Hema[3]

[1] Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Gurajada Vizianagaram (JNTU-GV), Vizianagaram, Andhra Pradesh, India
* **Corresponding Author Email:** vbharathid@adityauniversity.in - **ORCID:** 0009-0007-5828-0892

[2] Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Gurajada Vizianagaram (JNTU-GV), Vizianagaram, Andhra Pradesh, India
Email: mhemajntugvcev@gmail.com **ORCID:** 0000-0003-1097-9390

[3] Department of Electronics and Communication Engineering, Jawaharlal Nehru Technological University Gurajada Vizianagaram (JNTU-GV), Vizianagaram, Andhra Pradesh, India
Email: kapbbl.ece@jntugvcev.ac.in **ORCID:** 0000-0001-6712-062X

Article Info:

DOI: 10.22399/ijcesen.4092
Received : 10 September 2025
Revised : 03 November 2025
Accepted : 09 November 2025

Keywords

Approximate Compressors,
Gaussian Filter,
Low Power,
Image Processing,
Edge Detection.

Abstract:

Nowadays, approximate computing is an emerging trend widely applied in image processing, signal processing, and computer vision applications. Approximation is suitable for applications which are error tolerant. In this work, approximate computing is used in the Gaussian filtering stage of edge detection. In computer vision applications like medical imaging, transportation etc., Edge detection plays a crucial role. In edge detection algorithms, the Gaussian filtering block is a fundamental module. This block involves power-consuming addition operations, resulting in increased computational complexity, power consumption, and delay. To address these trade-offs, imprecise computation is applied in the Gaussian filter module. The aim of the work is to reduce computational complexity, reduction in usage of power and delay in the Gaussian filtering module using an approximate compressor-based architecture. The proposed Gaussian filter is designed using newly developed approximate compressors. Simulation outcome indicates that the Gaussian filter implemented with proposed newly developed compressors achieves decline in power and delay compared to exact and existing implementations in literature. Error analysis and Edge Detection is performed using different metrics in MATLAB, and the results demonstrate good PSNR values for the proposed compressors, confirming their suitability for edge detection in image processing applications.

1. Introduction

The term approximation is a commoly used technique in case of power consumption. This approximation is mostly used in many applications ranging from Image and signal processing to AI and neural networks. Imprecise computing is mostly used in the designs which are complex and consumes a more power. The concept of approximation defines that “in all application exact results are not necessary, some applications can bound to approximate results also”. The design of some circuits will be more complex which results in occupying more area. Due to this large circuit design the computational complexity will be also very high which leads to utilization of more resources and increase in amount of delay and usage of power. The concept of approximation reduces complexity of circuit in terms of complex circuits, also helps in consuming less power and increases speed of operation. The Gaussian filtering block is also complex and contains power hungry addition operations. A brief information about adders and compressors is mentioned by stating why compressors are better than adders.

Adders are basic digital circuits used for performing binary addition. These are very important for arithmetic operations, which are vital in many

applications like digital signal processing, arithmetic logic unit, memory systems etc. There are different adder topologies for e.g., ripple-carry, carry-lookahead, parallel prefix adders are used for efficient addition. Approximate adders introduce some in accurate results to achieve less power and area. These imprecise adders are used to perform addition with a certain level of imprecision to get reduction in power and area. Adders perform the operation in a sequential manner. But in compressors the parallel operation will be done for computing the circuit design. Compressors reduce multiple inputs to fewer outputs. Approximate compressors are used mainly in multipliers to lessen the stages of partial products. Using approximate compressors can lead to reduction in usage of power, Shows improvement in speed and area efficiency over exact compressor-based designs, in applications where some error tolerance is acceptable. Some of those error tolerant applications are image processing, Artificial Intelligence, Machine learning etc. Previous works used approximate compressors in multipliers but there is limited research of applied approximation in Gaussian filtering. In today's world edge detection plays a crucial role by enabling computers to identify and extract important boundaries and contours from images. This is used in applications like object detection and classification, autonomous driving, medical imaging, face recognition, security systems etc. Gaussian filter module is the main module in edge detection which is containing more power consumption addition operations. Improvements in area, performance, and power consumption are observed. Incorporating approximate computing in the filtering module effectively reduces hardware complexity and power usage, with a trade-off in computational accuracy. The highlights of this work include

- Design of two new 4:2 approximate compressors and one new 3:2 approximate compressor.
- Design of Gaussian filter architecture using newly developed approximate compressors.
- Design of Level 1, Level 2 and Level 3 in Gaussian filter Architecture.
- The simulation is carried out in Xilinx Vivado and error analysis is carried out in MATLAB Tool.

A brief background about Gaussian filter is mentioned. Gaussian filter eliminates the noise from input image and smoothens the image by averaging pixel values with a weighted mask where weights follow the Gaussian distribution. This maintains the integrity of edges and contours in the image. The weights of the weighted mask in both x and y directions are calculated using equation
Gaussian kernel Equation.

$$G(x,y) = \frac{1}{2\pi\sigma^2} * e\left(\frac{x^2+y^2}{2\sigma^2}\right) \qquad (1)$$

x, y are pixel positions relative to the centre of the filter. By computing the values of x and y in the above kernel equation the weights of the mask will be obtained. σ (sigma) is the standard deviation. By taking σ =1.4 as common choice and size of the window 5*5 the below matrix is obtained.

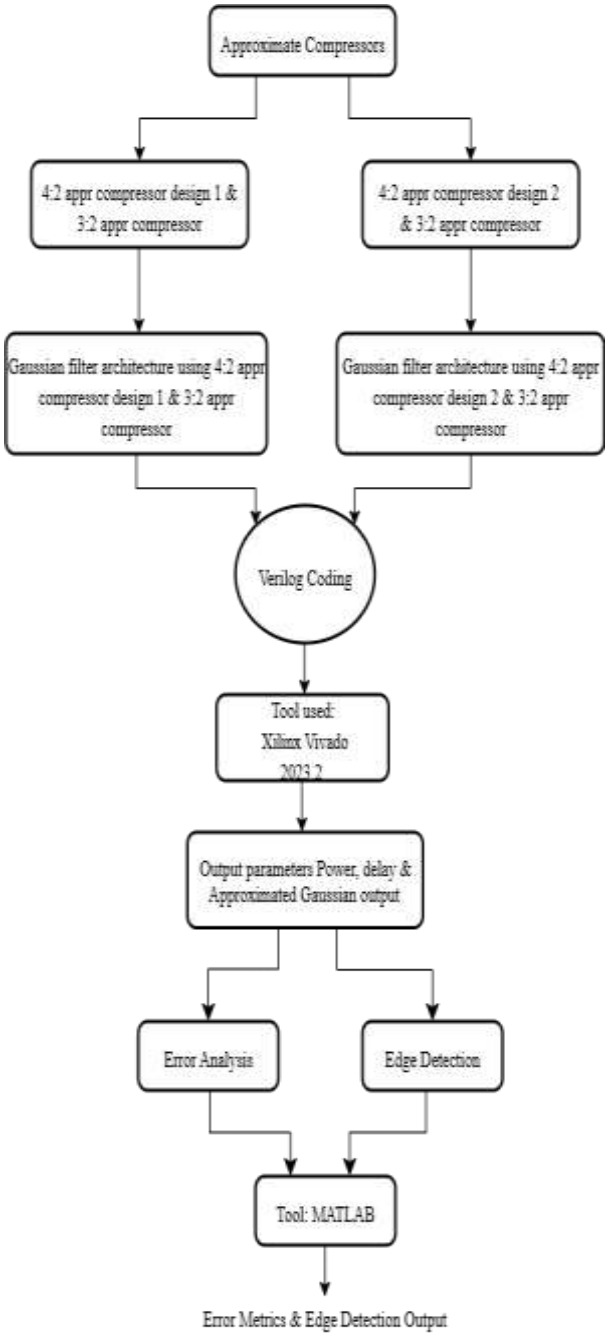

Figure 1. *Flow Diagram for Gaussian Filter Architecture.*

$$\begin{bmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \end{bmatrix} * \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

The left matrix contains the pixel values of input image and the right kernel contains the weights of the mask obtained from Gaussian kernel equation. The weighted averages obtained from the above filter gives the new smoothened image matrix. In next step the process edge detection is done by using convolution mask. Here Sobel gradient operator with two 3*3 convolution masks is used to obtain both x and y direction gradients. The obtained new weighted averages from above step are convoluted with the Sobel convolution masks and obtains the intensity gradients G_x and G_y. The equation to obtain

$$G_x = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} * \frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

The equation to obtain G_y

$$G_y = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} * \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

2. Theoretical Reference

The following is the literature review done on various approximate compressors for the better improvement of related work.

2.1 Literature work on approximate compressors in gaussian filtering and image processing

A Gaussian filter architecture with approximate compressors is designed to lower the usage of power and to reduce computational complexity of circuit [1]. A Gaussian image filter is developed using approximate hybrid adders. They described exact and compressed image filters in very high-performance ICs HDL concerning their developed methodology [2]. Design of a Gaussian blur filter using approximate arithmetic operations. Their simulation results indicated that the proposed Gaussian filter reduced energy consumption in comparison to existing designs [3]. A less energy consuming approximate multiplier is implemented using approximate compressor. The design has an 18.75% error rate and consumed 15% less energy and proposed two different multipliers for image and video processing applications [4]. Two multipliers are introduced using two approximate 4:2 compressors. The designed approximate multiplier is used in Sobel edge detection algorithm. The design was modeled and implemented in Verilog HDL alongside MATLAB simulations. and edge detected output is displayed [5]. A less energy consuming approximate multiplier is designed with new 4:2 approximate compressor. The simulation results showed that designed approximate multiplier achieved reduction in MED and maximum error distance. It was utilized in image processing application with increase in PSNR [6]. The inexact 4:2 and 5:2 compressors are designed and utilized in 8x8 Dadda multipliers. The error metrics for inexact 4:2 and 5:2 compressors and multipliers are analyzed and edge recognition is done utilizing Sobel operator [7]. The 8 transistor and 14-transistor approximate 4:2 compressors are designed and results that the 14-transistor architecture gives a lower error rate compared to 8-transistor with a tradeoff in lower area for higher accuracy. The compressor circuit is also evaluated using image processing application and the designed multiplier exhibits save in 50% area and 93% PDP when compared to exact compressors [8]. Design of two different 16-bit multipliers with approximate compressors. The results showed that two multipliers achieved 72% and 38% of power saving compared to exact multiplier. Also, the precision, mean relative error also resulted good and the multipliers are tested using image processing application [9]. The Dadda multiplier integrates a newly developed 4:2 approximate compressor aimed at reducing approximation error. The results showed that the multiplier with the developed approximate compressors demonstrate a notable decrease in error rate relative to other approximate designs. They also offer improvements in power efficiency, speed, and area utilization compared to exact compressors, and are effectively applied in image processing. [10].

2.2 Literature work on approximate compressors in multiplier designs

Design of 8x8 and 16x16 multipliers using high accuracy results indicate that the designed approximate 4 to 2 compressors provide an 18% improvement in delay and a 43–52% reduction in area–delay product, while consuming less power and offering superior performance. [11]. Design of an approximate multiplier architecture using developed 4 to 2 approximate compressor. The error is mitigated using a simple error-correcting unit. Experimental results indicate that both power consumption and the power–delay product are reduced compared to exact compressors. [12]. Comparison of various approximate 4 to 2

compressors which are already present and design 12 new approximate compressor designs. These designs were implemented in 8×8 and 16×16 multipliers, and it was noted that no identical winning approximate compressor topology exists. Each model is unique and the best solution depends on the required precision [13]. Design of an 8-bit multiplier using three novel approximate 4 to 2 compressors and an error correcting unit is designed to get better error performance. The simulated performance indicated that the designed approximate compressors achieved better energy efficiency and timing performance occupied less area [14]. We designed a new 4-gate 4:2 approximate compressor, complementary to earlier designs, and implemented a hybrid multiplier. Experimental results demonstrate that the hybrid multiplier provides an effective balance between accuracy and performance and exhibits a lower PDAP than the exact multiplier [15]. Two approximate 4 to 2 compressors are developed using circuits which reorders input and combinational input probabilities. An approximate multiplier is designed using this compressor. The multiplier designs utilized lower energy because of acceptable inaccuracy in the output, which suits for image processing applications [16]. Two novel approximate compressors are designed with zero mean error for applications which tolerates error. The performance evaluation (ED, MED, MRED, NMED & PDP) are analyzed and results showed the trade-off between accuracy and hardware efficiency [17]. Two 4:2 approximate compressors were proposed to enhance the energy efficiency of approximate multipliers. The proposed designs resulted in better improvement by maintaining low approximation errors compared with previous designs. These are applied in image sharpening application [18]. Two power aware 4:2 approximate compressors are designed. The designs resulted improvement in power and delay reducing the PDP by 66.7% and 79.1%. The efficiency of energy is also improved [19]. Less energy consuming designs of 3 to 2 approximate adder compressors and three different approximate 4 to 2 adder compressors are designed and compared with existing designs. The results showed the improvement in energy efficiency [20]. Analyzing various 4:2 approximate compressor architectures in terms of their performance characteristics are done and designed a 4 to 2 compressor. The power and timing metrics are performed and checked with the existing ones [21]. An approximated multiplier architecture is designed using approximate compressors. Low power circuits for approximate multiplication are designed using the approximated 4:2 compressor. The results showed that the design resulted in less power [22].

3. Materials And Methods

3.1 Exact Compressors

Before going to the new proposed approximate compressors, this is a brief information about exact compressors. Exact Compressors are basic combinational logic circuits. The use of compressors reduces the number of stages in partial product accumulation in arithmetic operations like multipliers or convolution in filters. 3:2 and 4:2 are the common compressors used in digital arithmetic circuits like multipliers, filters, etc. Exact compressors sum several binary numbers efficiently. When several operands are added the carry propagation should be avoided. These several operands can be efficiently reduced by using compressor structure.

Figure 2a shows the exact 3:2 compressor takes X1, X2, and X3 as inputs to generate sum and carry outputs, and its gate-level structure is depicted in Figure 2b.

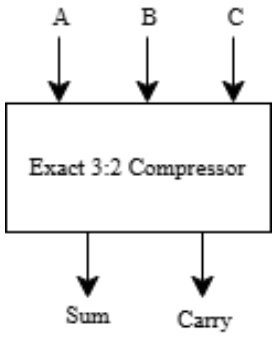

***Figure 2a**. Exact 3:2 compressor.*

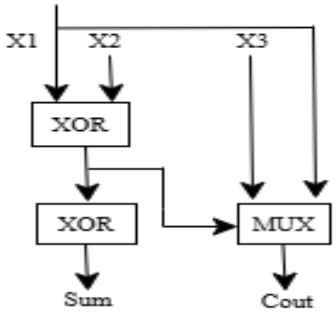

***Figure 2b**. Structural representation of Exact 3:2 compressors using logic gates.*

Figure 3a shows block level representation exact 4:2 compressor which takes inputs as X1, X2, X3, X4 and Cin and produces sum, C1 and Cout as outputs and Figure 3b shows the structural representation of Exact 4:2 compressors using logic gates.

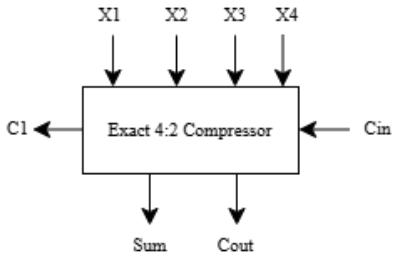

Figure 3a. *Exact 4:2 compressor.*

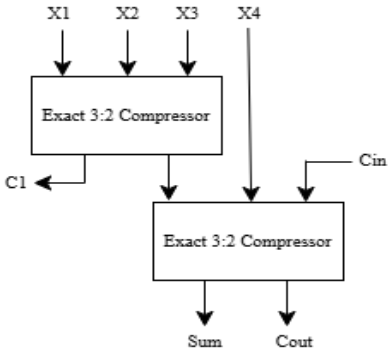

Figure 3b. *Structural representation of Exact 4:2 compressors using logic gates.*

Here Cin is the optional carry in input which comes from the previous stage of compressor and Cout is the output which goes to the next stage compressor. The precise 4:2 compressor is designed using two exact 3:2 compressors nothing but two full adders. The structure is not simply extension of a full adder but it helps to reduce the depth of the reduction network. This enables faster and parallel accumulation.

3. 2 Proposed Approximate Compressors

Presented here are two newly proposed 4:2 approximate compressors and a 3:2 approximate compressor, designed to achieve low error rates. These proposed designs help to reduce power consumption and also reduces computational complexity.

3.2.1 Proposed Approximate 3:2 Compressor

This presents a new 3:2 approximate compressor with less complexity in circuit and which gives better improvement than existing ones. Figure 4 represents the Structural representation of approximate 3:2 compressor using logic gates which takes inputs as A, B and C and producing approximate sum and Cout as outputs.

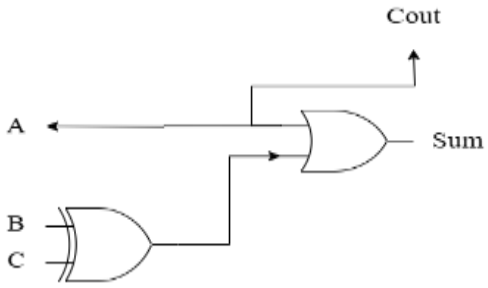

Figure 4. *Structural representation of Approximate 3:2 compressor using logic gates.*

This newly developed approximate 3:2 compressor is designed using one OR gate and one EX-OR gate which represents that the design is very simple with a smaller number of gates and transistors when differentiating with precise 3:2 compressor which is present in Figure 2a and 2b. This design gives improvement of power consumption and delay. The logic expressions for Sum and Cout are given as

Approximate Sum

$$Sum = A + (B \oplus C) \quad (2)$$

Approximate Carry

$$Cout = A \quad (3)$$

This design differs the output from exact 3:2 compressor only for two input combinations, “101” and “110”. Except these two input combinations for remaining all input combinations the output is same for both approximate and exact compressors. The rate of probability of error in this design in 2/8 which is a minimal error with reduction in delay and less usage of power.

3.2.2 Proposed 4:2 Compressors with Approximation

This section presents two newly developed approximate 4:2 compressors. The compressors give reduction in power and the design complexity is also less. The additional signals Cin and C1 which are present in exact 4:2 compressor in Figure 3a and 3b, are discarded in our proposed approximate 4:2 compressors. Figure 5 and Figure 6 shows the Structural representation of two approximate 4:2 compressors using logic gates. Figure 5 is the newly developed design 1.

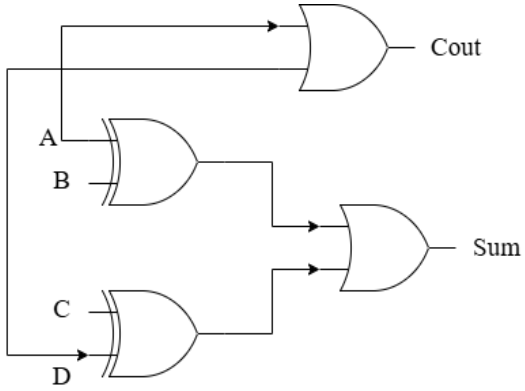

Figure 5. *Structural representation of newly developed Approximate 4:2 compressor Design 1 using logic gates.*

The design 1 consists of two EX-OR and two OR gates. The logic expressions for proposed design 1 are obtained from the truth table from Table1.

Approximate Sum

$$Sum = (A \oplus B) + (C \oplus D) \quad (4)$$

Approximate Carry

$$Carry = A + D \quad (5)$$

Table 1. *Truth table for newly developed Approximate 4:2 Compressor Design 1.*

Inputs				Approximate Outputs		Sum (Exact)
A	B	C	D	Sum (Appr)	Cout	
0	0	0	0	0	0	0
0	0	0	1	1	0	1
0	0	1	0	1	0	1
0	0	1	1	0	0	0
0	1	0	0	1	0	1
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	0	1
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	1	0	0
1	0	1	1	1	1	1
1	1	0	0	0	0	0
1	1	0	1	1	1	1
1	1	1	0	1	0	1
1	1	1	1	0	1	0

From Table I we can observe that the output of this design 1 approximate compressor differs with the output of exact compressor only for 4 input combinations, “0101”, “0110”, “1001”, “1010”. The rate of probability of error with this design is 4/16.

Figure 6 presents the newly developed approximate 4:2 compressor design 2.

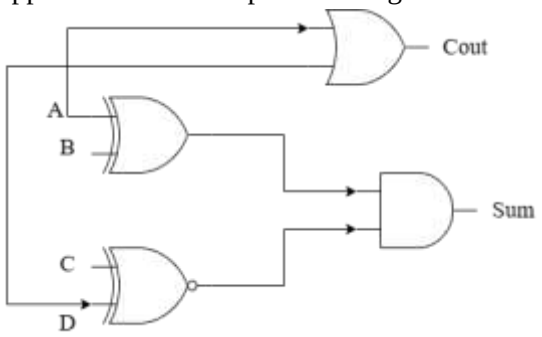

Figure 6. *Structural representation of Approximate 4:2 compressor Design 2 using logic gates.*

The design 2 consists EX-OR, AND, OR and EX-NOR gate of each which shows that the design is less complex. The logic expressions for design 2 are obtained from the truth table present in Table 2.

Table 2. *Truth table for newly developed Approximate 4:2 Compressor Design 2.*

Inputs				Approximate Outputs		Sum (Exact)
A	B	C	D	Sum (Appr)	Cout	
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	0
0	1	0	0	1	0	1
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	1	0	1
1	0	0	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	0	0	0
1	0	1	1	1	1	1
1	1	0	0	0	0	0
1	1	0	1	0	1	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

Approximate Sum

$$Sum = (A \oplus B)(C \odot D) \quad (6)$$

Approximate Carry

$$Carry = A + D \quad (7)$$

From Table II we can observe that the output of this design 2 approximate compressor differs with the output of exact compressor only for 4 input combinations, "0001", "0010", "1101", "1110". The rate of probability of error with this design is 4/16.

3.3 Design of Gaussian Filter Architecture Using Proposed Approximate Compressors

The general logic of Gaussian filter involves shift and add logic. In stage1 the pixel coefficients are shifted to left with a certain amount based on their corresponding weights. Generally, in filtering operations like Gaussian or any other filter the weights might not be the exact powers of two. So, to simply the hardware and to equivalent the weights the shifting operation is used instead of multiplication or division. Rather than multiplication or division this shifting operation is faster and also cheaper in hardware.

In stage 2 the left shifted coefficients are computed using compressors. The generic logic of Gaussian filter involves adder circuits in the place of compressors. But it is mentioned that using adder circuits results in more power consumption, computational complexity and usage of more resources. So, rather than using adders the stage 2 was replaced by using compressors. In stage 3 the compressed coefficients are given to final result using right shifting logic. The main aim is to design stage 2 with appropriate compressors. The Gaussian Filter Architecture using newly developed Approximate Compressor designs is shown in Figure 7.

This design of Gaussian filter architecture is developed using two proposed approximate compressor designs which are present in Figure 5 and Figure 6. The first Gaussian filter architecture is designed with newly developed 4:2 compressor design 1 with approximation and newly developed 3:2 compressor with approximation. The developed 4:2 compressor design 1 is placed in level 1, level 2 and developed 3:2 compressor design in level 3. The second Gaussian filter architecture is designed with newly developed 4:2 compressor design 2 with approximation and newly developed 3:2 compressor with approximation. The developed 4:2 compressor design 2 is placed in level 1, level 2 and developed 3:2 compressor in level 3.

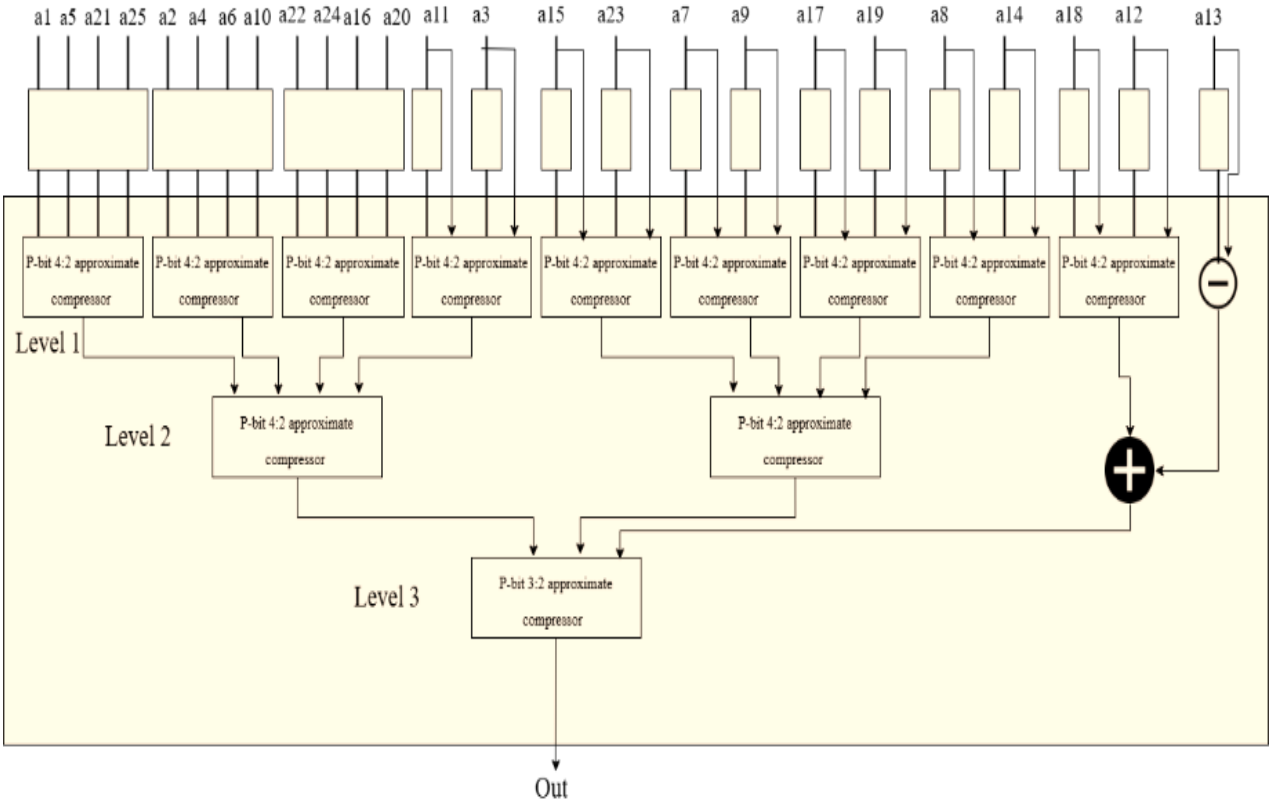

Figure 7. *Design of Gaussian Filter Architecture using newly developed Approximate Compressor designs.*

3.3.1 Design of Levels in Gaussian Filter architecture

The stage 2 contains mainly 3 levels. And the top input layer consists of 25 inputs, labelled as pixels from a 5×5 window. These inputs are grouped into clusters and each cluster is fed into a 4:2 approximate compressor in level 1. The level 1 is the first compression layer which consists of nine 4:2 approximate compressors. These compressors are labelled with bit widths like P-bit. These bits are labelled based on the pixel coefficient weights. Each compressor takes four values as input and compressors to two outputs. These outputs are grouped and passed to level 2 compressors. The level 2 consists of two approximate 4:2 compressors again labelled with bit widths P-bit. These take the compressed outputs from Level 1 and further reduce the number of partial sums. Level 3 consists of a single approximate 3:2 compressor of bit width P. The compressed outputs from level 2 are given this 3:2 compressor which comprises three inputs into two outputs. After this, additional basic arithmetic operations subtraction and addition are applied to adjust the final filtered value. These operations are used for weight corrections based on Gaussian weights. After these arithmetic operations the final output will be produced.

4. Results and Discussions

This section presents obtained results and observations of proposed designs. Code for proposed designs and code for Gaussian filter architecture with two proposed designs was written in Verilog and implemented using Xilinx Vivado 2023.2. After simulation the power and delay are observed for Gaussian filter architecture with both proposed designs. Table 3 gives the power and delay comparison parameters of Gaussian filter architectures with proposed designs against existing designs.

Table 3. *Power and Delay of Gaussian Filter architectures with newly developed against Existing designs.*

Compressor Design	Power (Watt)	Delay (Nano seconds)
S. Guturu [1]	1.548	2.436
Venkatachalam [9]	2.314	3.434
K. M. Reddy [10]	3.506	4.329
Proposed Design 1	1.306	2.302
Proposed Design 2	1.100	2.344

Figure 8 represents the graphical representation of power and delay comparison of Gaussian filter architectures with proposed designs against existing designs.

From Table 3 and Figure 8 it can be examined that the power and delay of the newly developed designs is comparatively less than the power and delay of existing designs.

Table 4: Comparison of Error evaluation for newly developed designs with existing designs. (MED, MRED, NMED)

Compressor Design	MED	MRED	NMED
S. Guturu [1]	1.00	0.1250	0.1250
Venkatachalam [9]	3.00	0.2750	0.3350
K. M. Reddy [10]	4.00	0.3980	0.4580
Proposed Design 1	1.00	0.1250	0.1250
Proposed Design 2	2.00	0.2500	0.2500

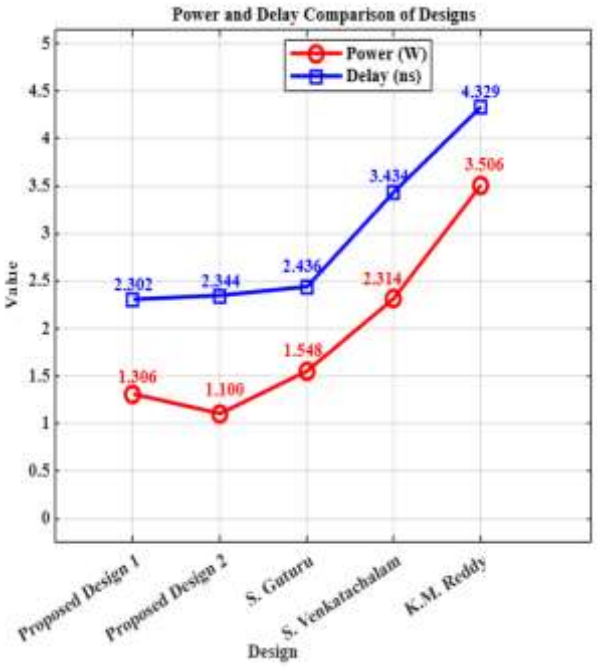

Figure 8. *Graphical representation of power and delay comparison of Gaussian filter architectures with proposed designs against existing designs.*

From Table 3 and Figure 8, it can be examined that the power and delay of the newly developed designs is comparatively less than the power and delay of existing designs. And also, it can be examined from Table III that Proposed imprecise 4:2 compressors are more power-efficient than

existing designs, with Design 1 outperforming Design 2 in power and delay.

The error analysis is also carried for newly developed designs and existing designs in MATLAB. Inbuilt test images, including Lena, were utilized to assess performance through metrics like Mean Error Distance, Mean Relative Error Distance, Normalized MED, Bit Error Rate, Mean Squared Error, and Peak Signal-to-Noise Ratio. Table IV gives the comparison of different error metrics for both proposed designs and existing designs.

Table 5. *Comparison of Error evaluation for newly developed designs with existing designs (BER, MSE, PSNR).*

Compressor Design	BER (%)	MSE	PSNR (dB)
S. Guturu [1]	12.50	64.00	30.07
Venkatachalam [9]	27.00	38.00	30.05
K. M. Reddy [10]	28.50	40.00	29.89
Proposed Design 1	12.50	16.00	36.09
Proposed Design 2	25.00	36.00	32.57

Figure 9 represents the graphical representation of error metrics comparison for both proposed designs and existing designs.

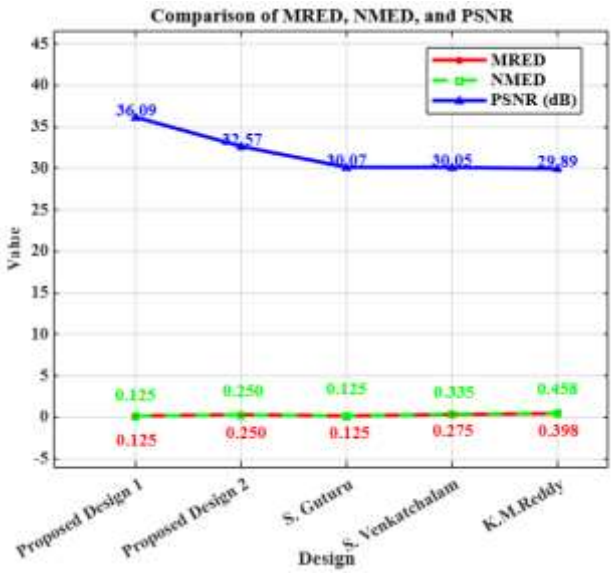

Figure 9. *Graphical representation of error metrics comparison for both proposed designs and existing designs.*

From Table 4,5 and Figure 9, it can be examined that the Proposed Design 1 results in the low MED of 1.00 bits and low Bit error rate of 12.50%. Also, it can be examined that the PSNR of the both proposed designs results good when compared with the PSNR of existing designs. This result of PSNR obtained from proposed designs is acceptable and suitable for better image quality in edge detection application.

The proposed design 2 has slightly a higher MED of 2.00 bits and BER of 25%, but surprisingly it resulted in high PSNR of 32.57dB. This suggests that even its bitwise accuracy is lower than proposed design 1 and existing design [1], its impact on overall image quality is less severe. This is probably due to its nature of approximation reducing high frequency noise. The existing design [1] having similar bitwise metrics with Proposed design 2 but results in lower PSNR. This highlights the effectiveness of proposed compressor-based architectures. The final analysis shows that both proposed designs are better and suitable than existing designs for hardware-efficient gaussian filtering in Edge detection application. The Edge Detection was carried out in MATLAB. From the obtained approximated output from above Gaussian filter architecture, the approximated 5x5 Gaussian kernel was obtained. This approximated Gaussian kernel is convoluted with original image 5x5 window with 256-pixel range. The sample image Lena is taken as original input image. After the image gets smoothened and obtained smoothened image is convoluted using Sobel operator for Edge detection.

Figure 10. Edge Detection using Exact Compressors.

Figure 11: *Edge Detection using Proposed Approximate Compressor Design 1.*

Figure 12. *Edge Detection using Proposed Approximate Compressor Design 2.*

From above figures it can be examined that both proposed designs are very near to the exact designs. It can be observed that each application is almost same with minimal change and with permissible error. This shows that the proposed designs are suitable for real world applications.

5. Conclusions

In computer vision applications like medical imaging, transportation etc., Edge detection plays a crucial role. In this edge detection, the Guassian filter module is the fundamental module. This gaussian filtering block involves more power consuming addition operations which results in more computational overhead, energy requirements, and propagation delay. This work aims to get the improvement in computational complexity, power consumption and delay. So instead of adders and exact computing, approximate compressors are utilized. Two 4:2 and one 3:2 approximate compressors are developed and deployed in a Gaussian filter design. The results showed that the usage of power and delay of our proposed designs are less compared to previous designs. The error analysis metrics are done in MATLAB and resulted better PSNR and accuracy with Proposed designs when compared with Existing design. The application part was carried out in Edge detection and observed that the proposed designs are very close to exact and existing designs which shows that these proposed designs can be used in real world scenarios.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The

References

[1]S. Guturu, A. K. Uppugunduru, A. Sharma, and S. E. Ahmed. (2024). A Power-Efficient Gaussian Filter Architecture Using Adder Compressors for Edge Detection Application. Circuits Syst Signal Process, 66(6): 2137–2150. DOI: 10.1007/s00034-024-02894-w.

[2]L. B. Soares, M. M. A. da Rosa, C. M. Diniz, E. A. C. da Costa, and S. Bampi. (2019). Design Methodology to Explore Hybrid Approximate Adders for Energy-Efficient Image and Video Processing Accelerators. IEEE Transactions on Circuits and Systems I: Regular Papers, DOI: 10.1109/TCSI.2019.2892588.

[3]I. Qiqieh, R. Shafik, G. Tarawneh, D. Sokolov, S. Das, and A. Yakovlev (2018). Significance-Driven Logic Compression for Energy-Efficient Multiplier Design. IEEE J Emerg Sel Top Circuits Syst 8(3) : 417–430, DOI: 10.1109/JETCAS.2018.2846410.

[4]L. H. Krishna, J. B. Rao, S. Ayesha, S. Veeramachaneni, and S. Noor Mahammad. (2021). Energy Efficient Approximate Multiplier Design for Image/Video Processing Applications. IEEE International Symposium on Smart Electronic Systems (iSES), IEEE, 8(3):417–430 DOI: 10.1109/iSES52644.2021.00056.

[5]A. K. P, J. N, M. R. Barusu, and M. D.(2023). Approximate Compressor for Efficient Multiplication in Image Processing Applications. International Conference on Next Generation Electronics (NEleX), 1–6 DOI: 10.1109/NEleX59773.2023.10421541.

[6]A. G. M. Strollo, D. De Caro, E. Napoli, N. Petra, and G. Di Meo.(2020). Low-Power Approximate Multiplier with Error Recovery using a New Approximate 4-2 Compressor IEEE International Symposium on Circuits and Systems (ISCAS) :1– 4 DOI: 10.1109/ISCAS45731.2020.9180767.

[7]Jain, S., & Singh, R. (2021). Implementation of approximate computing circuits for error-tolerant image processing applications. International Conference on Artificial Intelligence and Smart Systems (ICAIS), 1465–1469. IEEE. DOI:10.1109/ICAIS50930.2021.9395983

[8]Sadeghi, A., Rasheedi, R., Partin-Vaisband, I., & Pal, D. (2024). Energy efficient compact approximate multiplier for error-resilient applications. IEEE Transactions on Circuits and Systems II: Express Briefs,71(12),4989–4993. DOI:10.1109/TCSII.2024.3437235

[9]Venkatachalam, S., & Ko, S.-B. (2017). Design of power and area efficient approximate multipliers. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25(5), 1782–1786. DOI:10.1109/TVLSI.2016.2643639

[10]Manikantta Reddy, K., Vasantha, M. H., Nithin Kumar, Y. B., & Dwivedi, D. (2019). Design and analysis of multiplier using approximate 4-2 compressor. AEU - International Journal of Electronics and Communications, 107, 89–97. DOI:10.1016/j.aeue.2019.05.02

[11]Kong, T., & Li, S. (2021). Design and analysis of approximate 4-2 compressors for high-accuracy multipliers. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 29(10), 1771–1781. DOI:10.1109/TVLSI.2021.3104145.

[12]Anil Kumar, U., Chatterjee, S. K., & Ahmed, S. E. (2022). Low-power compressor-based approximate multipliers with error correcting module. IEEE Embedded Systems Letters, 14(2), 59–62. DOI:10.1109/LES.2021.3113005.

[13]Strollo, A. G. M., Napoli, E., De Caro, D., Petra, N., & Di Meo, G. (2020). Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers. IEEE Transactions on Circuits and Systems I: Regular Papers, 67(9), 3021–3034. DOI:10.1109/TCSI.2020.2988353.

[14]Pei, H., Yi, X., Zhou, H., & He, Y. (2021). Design of ultra-low power consumption approximate 4-2 compressors based on the compensation characteristic. IEEE Transactions on Circuits and Systems II: Express Briefs, 68(1), 461–465. DOI:10.1109/TCSII.2020.3004929.

[15]Zhang, M., Nishizawa, S., & Kimura, S. (2023). Area efficient approximate 4-2 compressor and probability-based error adjustment for approximate multiplier. IEEE Transactions on Circuits and Systems II: Express Briefs, 70(5), 1714–1718. DOI:10.1109/TCSII.2023.3257852.

[16]Krishna, L. H., Sk, A., Rao, J. B., Veeramachaneni, S., & Sk, N. M. (2024). Energy-efficient approximate multiplier design with lesser error rate using the probability-based approximate 4:2 compressor. IEEE Embedded Systems Letters, 16(2), 134–137. DOI:10.1109/LES.2023.3280199

[17]Rizwan, M. A., Waris, H., & Qadri, M. Y. (2022). Approximate multipliers based on low-power 4:2 compressors for error-tolerant applications. 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 432–437. IEEE. DOI:10.1109/IBCAST54850.2022.9990566.

[18]Liu, W., & Zhang, H. (2023). Efficient approximate 4-2 compressors and error compensation strategies for approximate multipliers. 6th International Conference on Electronics and Electrical Engineering Technology (EEET), 147–154. IEEE. DOI:10.1109/EEET61723.2023.00014.

[19]Zanandrea, V., & Meinhardt, C. (2023). New modified 4:2 approximate compressors for low-power applications. 36th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI), 1–6. IEEE. DOI:10.1109/SBCCI60457.2023.10261949.

[20]Lopes, R., Antonietti, L., Rosa, M. M. A., Costa, E., Soares, R., & Bampi, S. (2023). New energy-efficient 3-2 and 4-2 approximate adder compressors topologies. 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 1–4. IEEE. DOI:10.1109/ICECS58634.2023.10382943.

[21]Koshe, S. S., Rajgure, Y., & Sridevi, S. A. (2022). Novel implementation of low power and high performance 4-2 compressors for approximate multipliers. International Conference on Futuristic Technologies (INCOFT), 1–5. IEEE. DOI:10.1109/INCOFT55651.2022.10094489.

[22]Rai, S., Guntuku, D., Gupta, S., Savio, M. M. D., & Verma, M. (2023). Design of low power approximate multiplier based on compressors for multimedia application. International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), 1–6. IEEE. DOI:10.1109/RAEEUCCI57140.2023.10134205