technologies

Article

Development of Power-Delay Product Optimized ASIC-Based Computational Unit for Medical Image Compression

Tanya Mendez [1,2], Tejasvi Parupudi [3], Vishnumurthy Kedlaya K [2] and Subramanya G. Nayak [2,*]

[1] Department of Robotics and Artificial Intelligence Engineering, NMAM Institute of Technology, NITTE (Deemed to Be University), Nitte 574110, India; tanya.mendez@learner.manipal.edu or mendez.tanya@gmail.com
[2] Department of Electronics & Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India; kv.kedlaya@manipal.edu
[3] Department of Computer science and engineering, University of North Texas, Discovery Park, E245A, Denton, TX 76227, USA; tejasvi.parupudi@unt.edu
* Correspondence: gs.nayak@manipal.edu

Abstract: The proliferation of battery-operated end-user electronic devices due to technological advancements, especially in medical image processing applications, demands low power consumption, high-speed operation, and efficient coding. The design of these devices is centered on the Application-Specific Integrated Circuits (ASIC), General Purpose Processors (GPP), and Field Programmable Gate Array (FPGA) frameworks. The need for low-power functional blocks arises from the growing demand for high-performance computational units that are part of high-speed processors operating at high clock frequencies. The operational speed of the processor is determined by the computational unit, which is the workhorse of high-speed processors. A novel approach to integrating Very Large-Scale Integration (VLSI) ASIC design and the concepts of low-power VLSI compatible with medical image compression was embraced in this research. The focus of this study was the design, development, and implementation of a Power Delay Product (PDP) optimized computational unit targeted for medical image compression using ASIC design flow. This stimulates the research community's quest to develop an ideal architecture, emphasizing on minimizing power consumption and enhancing device performance for medical image processing applications. The study uses area, delay, power, PDP, and Peak Signal-to-Noise Ratio (PSNR) as performance metrics. The research work takes inspiration from this and aims to enhance the efficiency of the computational unit through minor design modifications that significantly impact performance. This research proposes to explore the trade-off of high-performance adder and multiplier designs to design an ASIC-based computational unit using low-power techniques to enhance the efficiency in power and delay. The computational unit utilized for the digital image compression process was synthesized and implemented using gpdk 45 nm standard libraries with the Genus tool of Cadence. A reduced PDP of 46.87% was observed when the image compression was performed on a medical image, along with an improved PSNR of 5.89% for the reconstructed image.

Keywords: application-specific integrated circuits; low-power VLSI; error-tolerant adder; medical image compression; peak-signal-to-noise-ratio; vedic arithmetic; Iterative carry save adder; power delay product optimized multiplier

Citation: Mendez, T.; Parupudi, T.; Kedlaya K, V.; Nayak, S.G. Development of Power-Delay Product Optimized ASIC-Based Computational Unit for Medical Image Compression. *Technologies* **2024**, *12*, 121. https://doi.org/10.3390/technologies12080121

Academic Editors: Tamás Haidegger, Gerard Ghibaudo and Francis Balestra

Received: 19 May 2024
Revised: 22 July 2024
Accepted: 26 July 2024
Published: 29 July 2024

1. Introduction

The onset of Very Large-Scale Integration (VLSI) has led to the advent of chips designed to perform a particular operation, resulting in the emergence of Application-Specific Integrated Circuits (ASIC) designs. Mastering the ASIC design process in digital VLSI is the cornerstone to successfully designing chips. The advent of sophisticated processing nodes has ushered in a new era of increased transistor density and given rise to complex and power-efficient ASICs. ASIC design engineers have consistently tried to break the barriers

in the ever-evolving realm of VLSI design by adopting emerging trends and exploiting cutting-edge technological innovations.

The arithmetic computational unit is an indispensable part of processors suited to image and signal processing applications. Several commercial processors have emerged as a result of the broadening of Digital Image Processing (DIP) applications. These processors have efficient computational units that boost overall performance, paving the way for faster and low-power computing. Extensive recurrent mathematical computations are required for these operations, resulting in increased delay and power consumption. High-speed computational units with low power consumption are desired for high performance in digital image and signal processing applications. The advent of emerging technologies and device scaling demands computational units to be constructed with lower power consumption, less propagation delays, and reduced dimensions.

The computational unit's architectural design and efficiency are deciding factors for the speed and power of the Central Processing Unit (CPU). This research explores different adder and multiplier architecture designs by incorporating low-power techniques and implements the computational unit design that is most suitable for DIP applications. The proposed computational unit comprises an Error-Tolerant (ET) adder, a Power Delay Product (PDP) optimized fixed-point multiplier, and a low-power logical unit. This research presents the design, development, and implementation of a PDP optimized ASIC-based computational unit for image processing applications.

Through the low-power concept of approximate computing, an ET adder is implemented in this research, which significantly improves both performance and power at the expense of accuracy. Multipliers form an indispensable part of almost all high-speed processors and CPUs, as a result of which research on the low-power design of multipliers makes a significant contribution to society. There is a rapid surge in the market demand for compact and lightweight electronic devices. An apparent low power demand emerges from such advanced forces of electronic circuits. Clock frequencies have been increasing for higher performance. Faster circuits are required to work at high clock frequencies, which increases the chip complexity, power consumption, and area. This has led to innovative techniques of low-power VLSI. The frequency has been limited to 4.5 GHz due to power limitations [1].

If the power consumption can be reduced with an acceptable increase in delay, then the frequencies can be increased, which is the necessity of high-performance devices, especially mobile phones. PDP optimized multipliers and arithmetic units are essential. The image processing algorithms implemented in contemporary times with software techniques are slower due to the limitations imposed by the processor's speed. A dedicated computational unit is necessary for image processing implementations to keep up with the speeds of high-performance applications operating at high clock frequencies. Due to the limitations imposed by General-Purpose Processors (GPP), implementing image processing algorithms in real time is complex as it requires high computational power and throughput rate. ASICs are suitable for stand-alone image processing applications due to their small size, lower power consumption, and greater functionality when used for specific purposes. ASICs can be customized to implement specific algorithms in a single small-sized, low-power integrated circuit, unlike GPP [1].

The computational units designed for image and signal processing applications comprise adders, multipliers, and logical units to perform various computations. A broad range of adder and multiplier architectures were proposed to meet the demands of high-performance devices. Complex tasks were executed in software or micro-code with minimal assistance from hardware during the early years of processor development. A review of the different architectures of Arithmetic Logic Unit (ALU) that were designed in the past to meet the demands of reduced power, area, enhanced efficiency, and high speeds is discussed in this section.

Lachireddy et al. [2] focused on the design of an ALU using reversible logic to reduce the power and Vedic multiplication to increase the speed. Two multiplier designs were

modeled, one using the Vedic multiplier and the other using the Booth multiplier. A drop of 6.7% in dynamic power and a 2.2% reduction in area was observed in the design that used the proposed Vedic algorithm.

Suchismita et al. [3] presented the design for a custom 4-bit ALU that consumed less power and was fast. The power consumption was lower compared to the designs implemented using conventional techniques. Since different modules were used for each operation, the overall complexity was more in terms of area.

Madhukar et al. [4] proposed the design of a 16-bit ALU for high-speed and low-power applications. A parallel prefix adder, the Brent Kung adder, was utilized to perform the addition operation as it consumed less power. The multiplication operation was performed using the Booth multiplier, involving fewer additions than the existing array and Wallace multipliers. The restoring algorithm used for division consumed less power than subtractive methods. The design exhibited reduced delay and power consumption at a slightly increased area.

Suhas et al. [5] designed a hybrid 32-bit ALU using irreversible and reversible gates. The modified Carry Save Adder (CSA) was implemented using reversible gates to reduce the power further. The Urdhva Tiryagbhyam (UT) sutra of Vedic mathematics was used to design the multiplier, and the addition of the partial products was performed using the delay-modified CSA to further reduce the delay. Reversible logic gates were used to design the subtractor. The design exhibited an overall reduction in power and delay but with an increased area. Kamaraj et al. [6] designed an ALU that was fault-tolerant using KMD, Toffoli, and Fredkin reversible gates, which were functionally realized in Quantum Cellular Automata. A single module performed the arithmetic and logical operations. The design utilizes more gates but with less quantum cost.

Rashmi et al. [7] implemented two 8-bit ALUs for low-power by incorporating the techniques of clock gating and operand isolation at the Register Transfer Level (RTL) level. Compared to conventional designs, the design with operand isolation showed that it consumed more power and occupied more area with less delay. Similarly, the design with clock gating consumed less power, but the area and delay were more than those of conventional designs. Nagarjuna et al. [8] proposed a low-power 8-bit ALU with two 4×1 data selectors, a 2×4 decoder, a full adder, and a logic unit. The design seemed complex in terms of area, as it involved decoders and data selectors.

Maan et al. [9] compared different clock gating techniques in 8-bit ALUs. The newly proposed technique of achieving clock gating using a tristate buffer resulted in more power and area savings as the idle parts were eliminated. The use of the tristate buffer showed an improved performance. Moresh et al. [10] presented a 4-bit ALU utilizing the Gate Diffusion Input (GDI) approach. GDI is a novel method to design low-power combinational circuits that reduce power and area. In contrast to conventional designs, the proposed concept was fast and efficient.

Priyanka et al. [11] suggested an 8-bit ALU with a Knowles adder for addition and a Vedic multiplier constructed using the Ladner-Fisher approach. The multiplier used in the design had better power, area, and delay efficiency than the Vedic multipliers used in conventional designs. The design utilized the Knowles adder to reduce the routing complexity and area. The efficiency of the adder and multiplier used can be further improved.

Abdul et al. [12] designed two types of architectures of 32-bit ALUs using the Sklansky and Ripple Carry Adder (RCA) adders. In addition, the power and area of the architecture with RCA were compared to those of the architecture constructed without a demultiplexer. This was achieved by employing a demultiplexer to cut off the power to the blocks that were not in use. The results showed that the architecture designed with a demultiplexer proved to be more power-efficient. A power-saving method for an 8-bit ALU utilizing clock gating by incorporating a negative latch was presented by Gunjan et al. [13]. The design was implemented at the RTL level on an FPGA, and the results of power consumed were compared with a D Flip Flop-based clock gated design (243 mw at 100 MHz), which

showed a significant power reduction. The design that used the negative latch exhibited an increased area and speed.

An 8-bit ALU was proposed by Akram et al. [14] utilizing a method to lower leakage power. The power supply was decreased, and the ground potential was raised to compensate for the delay overhead that lowered the leakage power. Bishwajeet et al. [15] proposed an ALU of size 8 bits that utilized AND gate-based clock gating to reduce the dynamic and clock power consumed. Power savings were achieved by powering off the inactive modules using the clock gating approach. The analysis of power and area was conducted at different clock frequencies, and it was found that even though there was a reduction in clock and dynamic power, there was an overhead of increased area.

Sushma et al. [16] have presented a Vedic multiplier by proposing an area and speed-efficient design for an ALU. Unlike conventional designs, the novel Multiplier Adder and Subtractor (MAS) unit was implemented by aggregating the adder, subtractor, and multiplier, thereby reducing the area, and by modifying the UT formula for Vedic multiplication. The MAS-based multiplier proved to be faster than the Booths algorithm but slower than the UT multiplier. The MAS-based multiplier was found to occupy slightly more area than the Urdhva method of multiplication and Booth's algorithm.

To increase efficiency, M. Kamaraju et al. [17] presented the design of a 16-bit ALU with lower power consumption that used the clock gating approach. The design was divided into four functional blocks, each designed to perform four operations. Clock gating was achieved using a decoder that generated the coded clock signals based on an opcode to the desired functional block. Power dissipation can be reduced further by using improved techniques for clock gating, and high-speed adders and multipliers can be used to enhance the speed further. Substantial work has been conducted to design ALUs to meet the expectations of high-speed processors. The overall power can be lowered by minimizing the power of the adders and multipliers used in the ALUs.

Medical images like Magnetic resonance imaging (MRI), Computed Tomography (CT) scans, and X-rays in the healthcare domain have an increased size owing to substantial high-resolution information packed into a single image. The processing of these high-resolution images for various applications requires the images to be compressed in size to reduce the memory occupied by these images, especially for applications that demand high speed and less area. Researchers have been consistently working on different algorithms to compress the images. Most of the existing work focused on reducing the size of the images with less emphasis on the computational complexity of the processor. The growing demand for efficient transmission and storage of medical images at a reduced computational complexity in the medical field is the driving force of research designing a computational unit for image compression.

The compression of medical images based on transforms has recently resulted in a huge demand due to their growing applications in the medical domain, attracting researchers. The efficiency and computational speed of Discrete Cosine Transform (DCT) are among the most sought-after algorithms, especially for medical images. Some of the DCT algorithms used in different medical image compression applications are listed here to give a clear idea about the existing work.

An image compression algorithm using DCT was proposed by Wu et al. [18], and the significant coefficients were obtained using adaptive sampling techniques. The proposed algorithm resulted in compressing the images and preserving the information fidelity. Chen et al. [19] proposed a Two-Dimensional (2D) image compression method for medical images utilizing DCT and hierarchical tree partitioning. An approach with a reduced bit rate, resulting in minimizing the memory requirement for storing the remote diagnosis and for the rapid transmission, was also adopted in this work.

An approximation technique for 8-point DCT with 14 addition operations utilizing the separability property of DCT was presented by Potluri et al. [20]. A reduction in the algorithm and computational complexity was observed. Kaur et al. [21] have proposed context tree lossless compression for the Region Of Interest (ROI) image part and a lossy

compression fractal for the non-ROI part of the image. The proposed method provided much faster and more accurate results even though the PSNR was slightly lower.

The research carried out by Suma et al. [22] focused on the utilization of the UT sutra in the design of the multiplier that was incorporated into the image compression algorithm. A significant reduction in the size of the image was observed with less data loss. The compression speed was improved with the use of the Vedic multiplier as the computation time of the image compression algorithm was reduced. In their work, Liu et al. [23] discussed the different standards used in image compression, mainly for medical images. A comparison of the performances of the different compression standards on publicly available datasets was also conducted.

A DCT architecture without multipliers was proposed for medical applications targeting capsule endoscopy by Xing et al. [24]. Approximate adders, threshold setting, and coefficient optimization were incorporated into the DCT algorithm design. A DCT architecture utilizing inexact adders was proposed by Kumar et al. [25]. Considerable savings in area and power were observed due to the absence of multipliers in the architecture. Hosny et al. [26] presented a 2D medical image compression algorithm in their work. The algorithm proposed exhibited high compression rates and good image resolution.

Alzahrani et al. [27], in their work, compared two lossless transforms, DCT and Haar Wavelet Transform (HWT), with medical images in terms of image size and compression ratio. It was observed from their inferences that while using grayscale images, DCT proved to perform much better than HWT. HWT was found to be a better choice when color images were used and in terms of compression time. A discussion on image compression algorithms using DWT and Huffman coding was presented by Mydin et al. [28] in their work. In their work, Rahman et al. [29] analyzed and evaluated the strength of different lossless compression techniques based on more than one parameter. In an attempt to improve the compression ratios, the medical image compression techniques discussed resulted in a reduced quality of images and loss of minute details, which is undesirable in medical images.

Medical image compression primarily aims to reduce the image size to minimize the memory required for processing and storing the image without affecting its quality. Even though considerable research strategies for medical image compression were developed, an algorithm specific to medical image compression that focuses on reducing computational speed, area, and power has not been developed.

The digital compression of medical images, especially for telemedicine, has gained significant importance in recent years due to medical advancements. Primary health care, especially in villages, is challenging owing to the limited facilities available and the lack of well-trained health personnel. With telemedicine, most of the problems faced by people living in villages can be solved. The main idea behind telemedicine is the efficient transmission of medical images to healthcare professionals in modern cities. A critical factor that comes into play in telemedicine is to handle the transmission of cumbersome medical images with comparatively lower bandwidth. This demands that the medical images be compressed to enable the efficient and error-free transmission of information. This research explores the DCT that can be utilized for digital image compression by incorporating the computational unit to reduce computational complexity and minimize power as well as area. In the DCT coding technique followed in this research, a considerable set of highly correlated pixels are transformed to a comparatively smaller set of transform coefficients that are decorrelated. The quantization and encoding of the coefficients are then performed to obtain the compressed image.

This research aims to design and develop a computational unit for medical image compression comprising a modified Carry Select Adder (CSLA)-based Selector-Based Error Tolerant Adder (SBETA)/Low-Power Selector-Based Error Tolerant Adder (LPSBETA)/Optimized Error Tolerant Adder (OETA) using RCA, a divide and conquer multiplier, a PDP optimized multiplier using iterative carry save SBETA and a logic unit using operand isolation. In the proposed research work, an architecture of a computational unit

incorporated in the Loeffler 8-point DCT/IDCT algorithm and used for medical image compression applications is proposed, designed, and implemented. The ET adders designed using the low-power concept of approximate computing substitute the addition operations involved in Loeffler's fast algorithm's DCT/IDCT computations. This reduces the hardware complexity, resulting in significant savings in area and power. The Binary to Excess 1 Converters (BEC) used in the modified CSLA reduces the number of gates involved, leading to lesser power and area as the transitions rely on only one bit. The multiplications involved are minimized by approximating some of the cosine constants involved and using the proposed low-power multipliers and shifters to reduce the computational complexity. An efficient encoding and decoding block is proposed using the improved Loeffler algorithm. The contributions of this research are as follows:

1. A DCT/IDCT architecture based on the Loeffler algorithm by incorporating the computational unit to perform the additions involved in the butterfly diagram using the ET adders and the low-power multipliers for the multiplications involved.
2. An improved encoding and decoding block is used to reduce the circuit complexity and improve the compression ratio, SSIM, and PSNR without affecting the quality of the reconstructed image.
3. An approach to building a computational unit by integrating VLSI ASIC design and the concepts of low-power VLSI compatible for medical image compression used in telemedicine was adopted in this research.

The VLSI ASIC design flow that is followed in this research work is presented in Section 2. Section 3 elaborates on the development of the proposed computational unit designed and implemented in this research. The modified CSLA-based SBETA/LPSBETA/OETA using RCA, a divide and conquer multiplier, a PDP optimized multiplier using iterative carry save SBETA, and a logic unit using operand isolation incorporated in the computational unit is also discussed in Section 3. The digital image compression application using the proposed computational unit for the medical image is demonstrated in the last portion of Section 3. The results of the proposed computational unit are discussed in Section 4. The concluding remarks of the research work are given in Section 5.

2. VLSI ASIC Design Flow

The ASIC design flow comprises several steps, from conceptualization to specification and tape-outs, as shown in Figure 1. The end product usually is small in the order of nanometers. The ASIC design flow starts with the specification definition of the system [30].

The current emphasis of low-power VLSI design is on designing and implementing power-efficient algorithms of arithmetic units such as adders, subtractors, multipliers, and dividers. Due to the rapid growth of wireless devices in the electronics market, total power consumption, a core design constraint, must be addressed. The efficient design of an Integrated Circuit (IC), considering low power, high speed, and small area, is quite challenging [31]. Since maximizing the run time of these portable devices is desirable, reducing power consumption becomes a critical factor that requires careful consideration [32].

The most demanding constraint when it comes to the design of an efficient, high-performance system in nanometer technologies is power dissipation. ASICs, having optimized performance, offer better power efficiency than other implementation schemes. ASIC offers lower power solutions and higher speeds for the same processor technology due to its interconnect architecture compared to FPGA. Due to the limitations imposed by general-purpose processors regarding computational power and throughput, implementing image processing algorithms in real time is complex as it requires high computational power and throughput rate. ASICs are semiconductors that can be specially designed for a particular function. They can reduce silicon area with high efficiency in speed and power, making them suitable candidates for image processing applications.

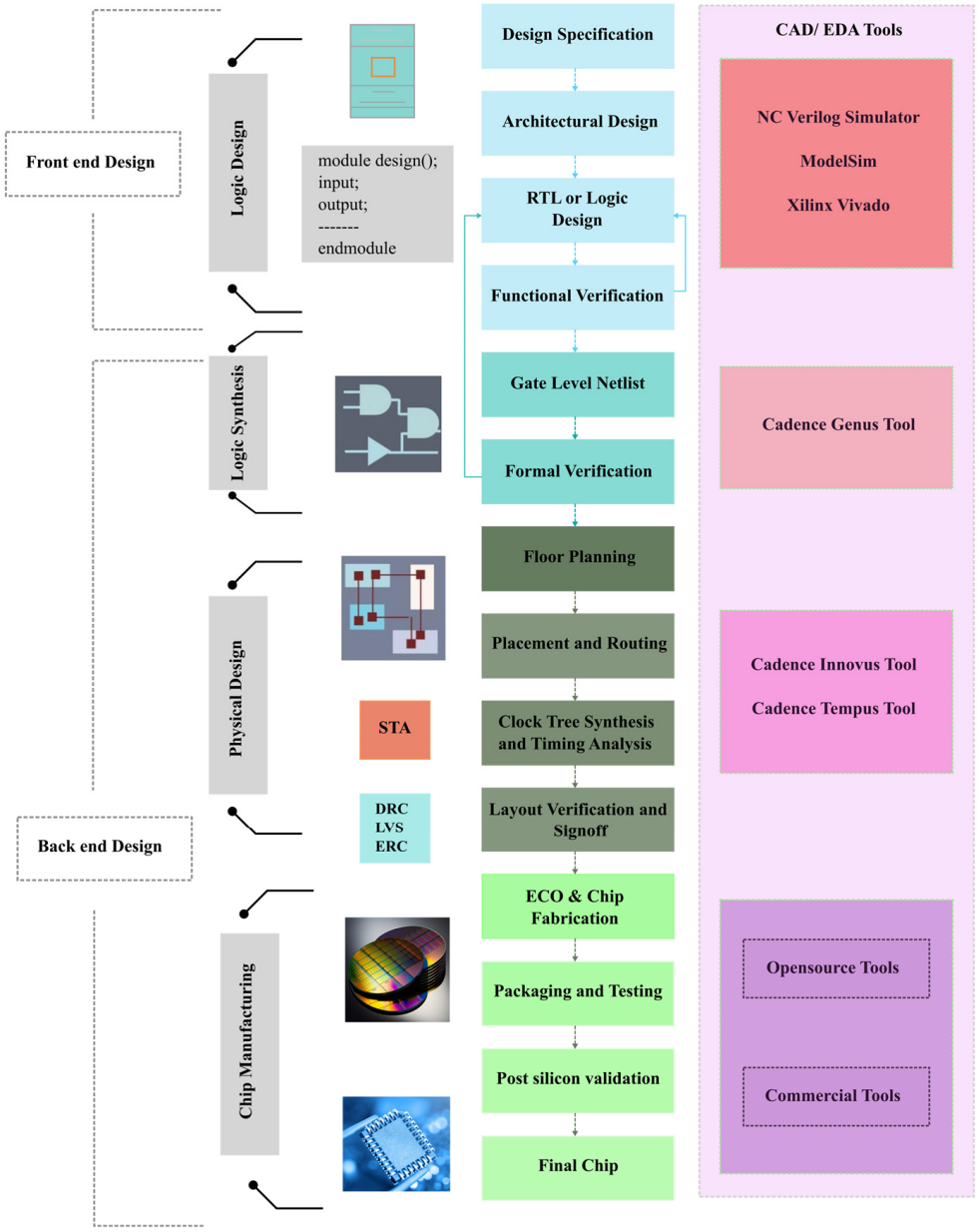

Figure 1. VLSI ASIC Design Flow.

The efficient implementation of the different arithmetic circuits of the computational unit for executing distinct algorithms is realizable with ASIC. As the complexity of arithmetic circuits increases, so does the power consumption and transistor density, giving rise to the low-power design techniques of VLSI with state-of-the-art concepts. With the ever-growing popularity of battery-powered electronic devices, area, power, and delay from the perspective of a circuit play a significant part in VLSI. As the area increases, the number of logic gates also increases, slowing the circuit and increasing power consumption. A trade-off often exists among power, delay, and area in VLSI.

The PDP is the design metric that gauges the effectiveness and performance of low-power devices. As the complexity of the devices increases, the power dissipation keeps rising, impacting the performance and reliability. In such a scenario, power consumption becomes an integral part of the design that requires the design constraints to be further optimized. The latest technological advancements have led researchers to work to design power-efficient, high-speed, and smaller devices with a focus on Power Delay Area (PDA) constraints.

3. Development of the Proposed Computational Unit

A universal integrated logic device, the computational unit performs arithmetic and logical computations. The computational unit is an essential microprocessor component due to its versatility. The computational unit proposed in this research is illustrated using the block diagram shown in Figure 2. The computational unit takes two operands, A and B, as inputs and produces an output by applying predefined operations to the inputs, as presented in Table 1. The selection input S(2) decides the arithmetic or logic operation. The inputs S(1) and S(2) decide the operations to perform within the arithmetic or logic block. The arithmetic computational unit comprises an ET adder and a low-power multiplier. An OR, AND, XOR, and NOT logical gate comprises the logic unit using operand isolation.

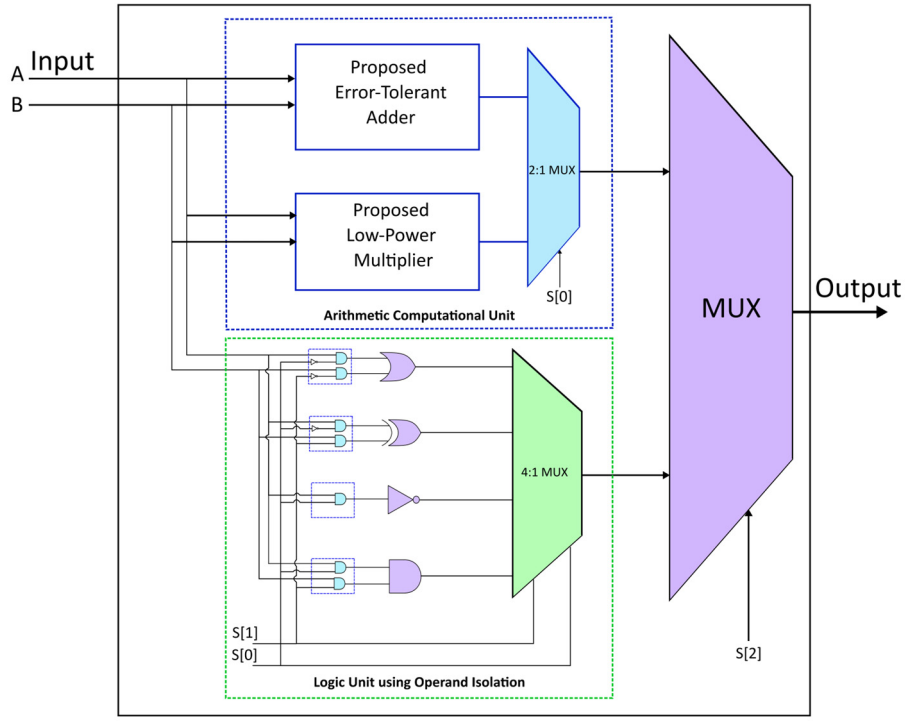

Figure 2. Proposed Computational Unit Block Diagram.

Table 1. Operation Table for Proposed Computational Unit.

S(2)	S(1)	S(0)	Operation	Function
0	0	0	A + B	Addition using ET Adders
0	0	1	A × B	Multiplication using LP Multipliers
1	0	0	A | B	Logical operation using logic unit
1	0	1	A ⊕ B	
1	1	0	~A	
1	1	1	A & B	

The computational unit is built by integrating the ET adders, the low-power multipliers, and the logic unit for 8-bit, 16-bit, and 32-bit computations. The computational unit has been configured so that the logical module activates when S(2) is 1, and the arithmetic module activates when the select input S(2) is 0. Depending on the S(0) input, addition using the ET adders or multiplication utilizing the low-power multipliers is performed. When the input S(2) is 1, the inputs S(1) and S(0) determine which logical operation has to be carried out. The different combinations of designs are mentioned in Table 2.

Table 2. Different Designs of Computational Units.

Computational Unit	Modified CSLA-Based Adder	Multiplier
CU1	SBETA Using RCA	Divide and Conquer Multiplier
CU2	LPSBETA Using RCA	
CU3	OETA Using RCA	
CU4	SBETA Using RCA	PDP Optimized Multiplier Using Iterative Carry Save SBETA
CU5	LPSBETA Using RCA	
CU6	OETA Using RCA	

3.1. Modified CSLA-Based Adders Using Error Tolerant Adders

The arithmetic computations are accomplished utilizing the ET Adders presented in our previous research [33]. Three designs of ET adders were incorporated in the computational unit as illustrated in Figure 3, where 1 depicts the logic for the Selector-Based ET Adder (SBETA), 2 describes the logic for Low-Power Selector-Based ET Adder (LPSBETA), and 3 depicts the logic for the Optimized ET Adder (OETA). The pseudocode for SBETA is also depicted.

Pseudocode: Selector-Based ET Adder

```
if (A == 1)
        Carry Output = B AND Cin
else
        Carry Output = B OR Cin
end
```

Sum Output = **NOT** (cout)

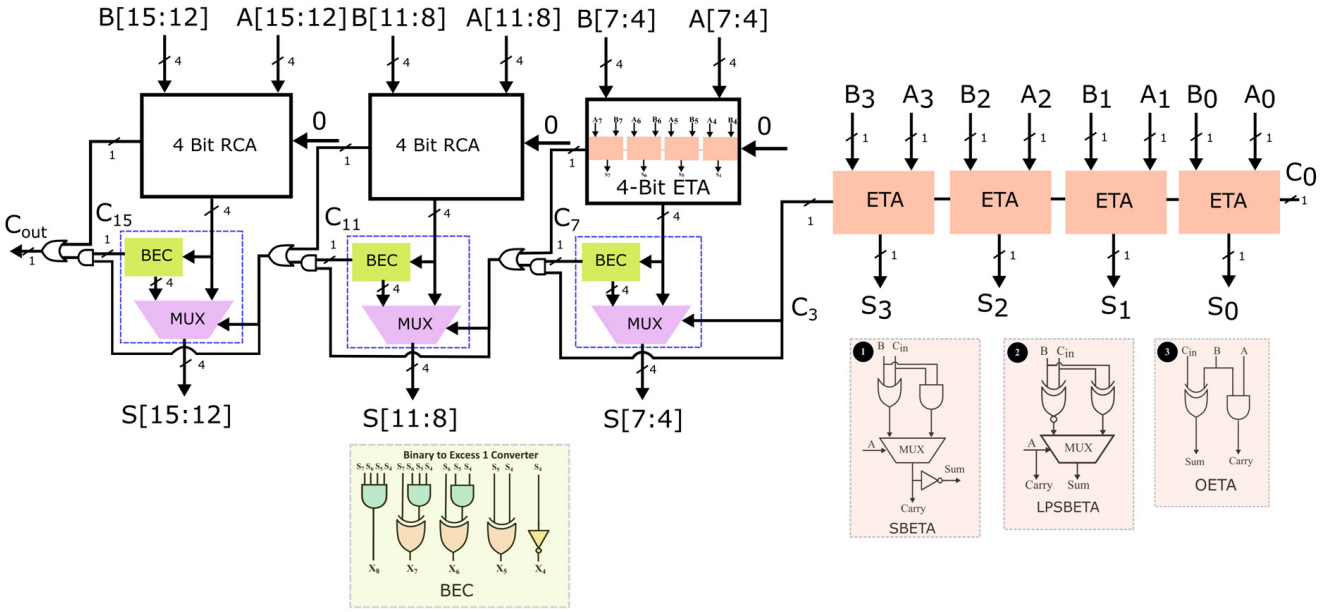

Figure 3. Modified Carry Select Adder (CSLA) using ET Adders.

While using the approximate computing paradigm, we noted that errors are introduced in the sum and carry output of the ET adders to minimize the complexity of the Boolean expressions, thereby reducing the power and area. The conventional CSLA is modified by incorporating the ET adders for the lower order bits, as illustrated in Figure 3. The RCA is used for addition when the previous carry is zero. A selector unit is utilized for the higher order bits to reduce the switching activity whenever the previous carry is 1. The power is reduced further by the BEC, that is utilized whenever the carry generated is one.

3.2. Multiplier Based on Divide and Conquer Approach

The proposed multiplier based on the divide and conquer approach is illustrated in Figure 4 and Algorithm 1 depicts the algorithm used. Phase 1 determines the type of multiplier operation to be performed depending on the input type.

Algorithm 1: Multiplier based on Divide and Conquer Approach

One Hot Block: *Checks if multiplier/multiplicand is one hot (Input has exactly one bit set to 1).*
Two-part Multiplicand Block: *Checks if the multiplier is 3 or less than 99.*
Divide and Conquer Block: *Checks if multiplier/multiplicand is an even number.*
Four Range Block: *This block is enabled when the conditions mentioned above are not met.*

In the second phase, just one block is activated, reducing power.

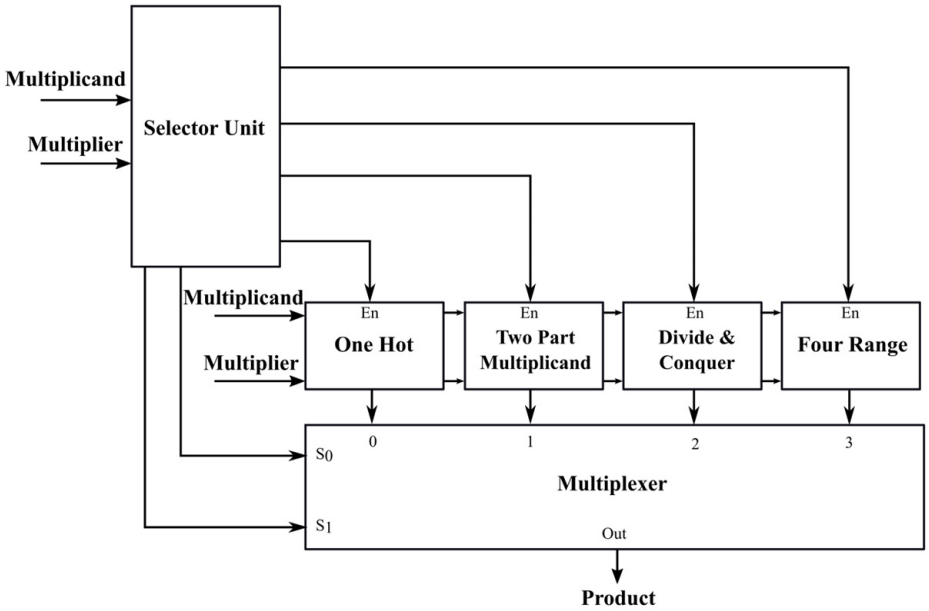

Figure 4. Multiplier based on Divide and Conquer approach.

The detailed algorithm and block diagram is explained here.

i. One Hot Block:

Figure 5 illustrates the block diagram of one hot block. The "Selector unit" determines whether the multiplier/multiplicand is one hot or not.

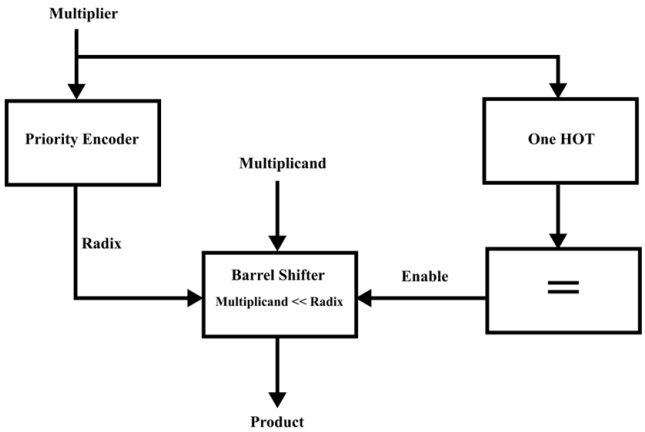

Figure 5. Block diagram of One Hot Block.

A one hot flag checks if the multiplier/multiplicand is one hot. A priority encoder determines the radix of the multiplier and multiplicand. The smaller number is made the multiplier based on the radix obtained. The product is obtained by left shifting the multiplicand by the radix of the multiplier when the multiplier is one hot. Likewise, when the multiplicand is one hot, the product is obtained by left shifting the multiplier by the radix of the multiplicand. The shifting operation can be accomplished using a barrel shifter.

ii. Two-part Multiplicand Block:

The block diagram of the two-part multiplicand block is depicted in Figure 6. The "Type Determinant" block determines whether the multiplier is a multiple of 3 or if the multiplier is less than 99. The two-part multiplicand block is enabled if the multiplier represented by B is in the range specified or is a multiple of 3. First, the multiplier is divided into two parts of 4 bits each—B[7:4] and B[3:0]. Each part is multiplied with the multiplier using the MUL block to obtain M1 and M2. A barrel shifter is utilized to left shift M1 by four-bit positions. Finally, the adder block adds the shifted P1 with P2 to obtain the product.

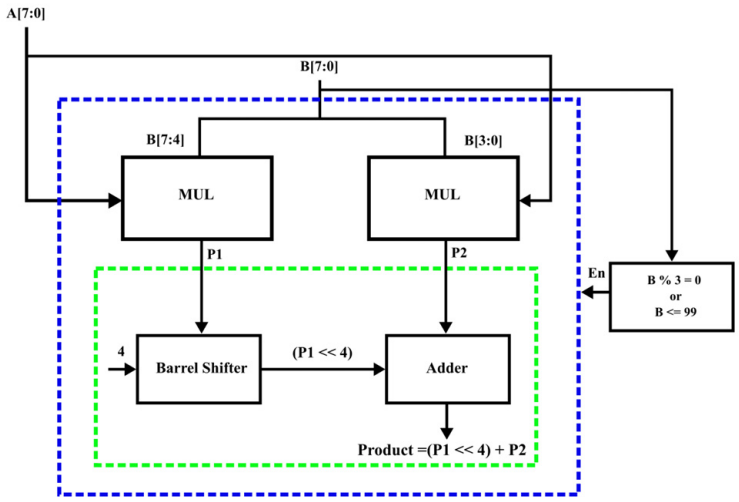

Figure 6. Two-part Multiplicand Block.

iii. Divide and Conquer Block:

The priority encoder block in the proposed architecture determines the range of the multiplicand and multiplier to identify their radix. The numbers to be multiplied represented by A and B are examined to identify if they are even or odd. The even number is designated as the multiplier, and the trailing zeros count is determined. The multiplicand is shifted to the left by the trailing zeros count of the multiplier. The multiplier is shifted to the right by the trailing zeros count of the multiplier. Multiplication is now performed on multiplier and multiplicand to obtain the product. The shifting operation reduces the multiplier range, reducing the power consumption. Figure 7 shows the block diagram of the divide and conquer block.

iv. 4-Number Range Block:

The block diagram for the 4-power range method is illustrated in Figure 8. The radix of the multiplier and multiplicand are determined using a priority encoder. A comparator module then compares the radix of the two numbers to decide which is the smaller number, which is assigned as the multiplier.

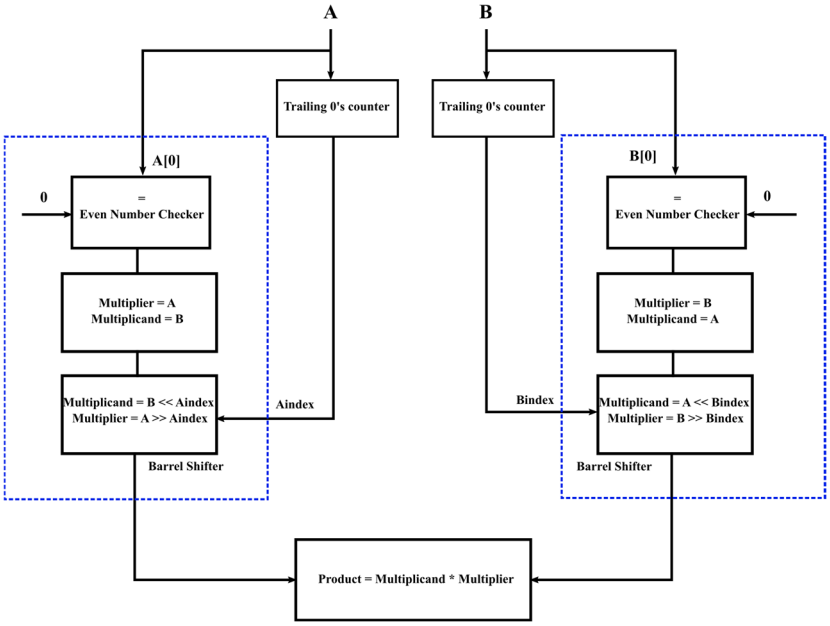

Figure 7. Divide and Conquer Block diagram.

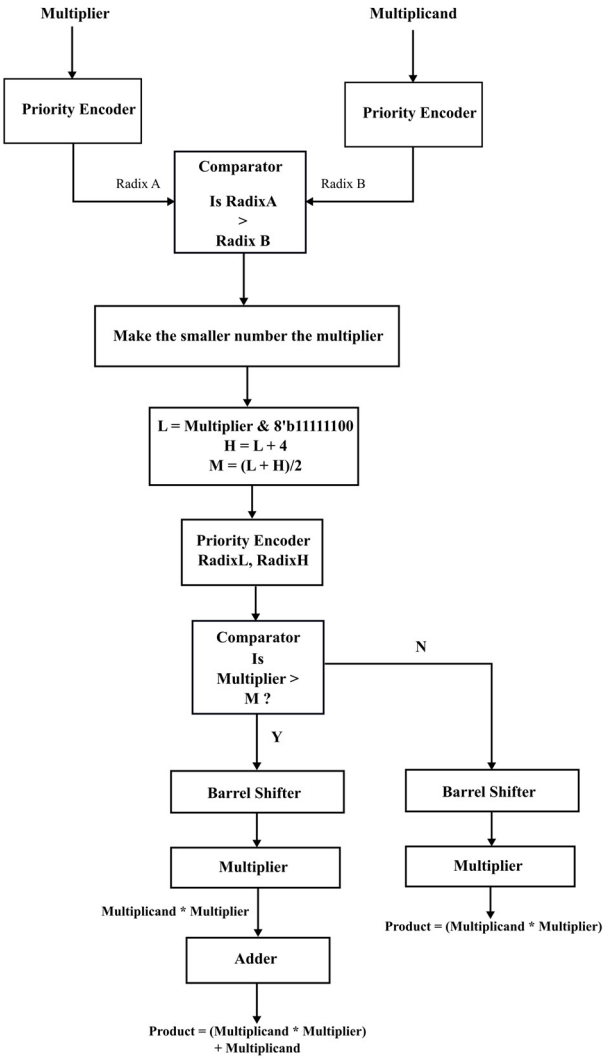

Figure 8. 4 number range Block diagram.

The multiplier's range [L, H] is then decided based on the radix of the multiplier obtained from the priority encoder. The range of the L value is 2n, and 2n + 4 is the H value range. The H and L value average in the range gives the midpoint value (M). A priority encoder is used to determine the radix of H and L. A comparator is used to check if the multiplier is greater than the midpoint value M. If the multiplier is greater than M, the multiplicand is shifted to the left by the radix of H, and the multiplier is shifted to the right by the radix of H. A multiplier calculates the product, which is then added to the multiplicand using an adder. If the multiplier is smaller than the midpoint value, the multiplicand is shifted to the left by the radix of L, and the multiplier is shifted to the right by the radix of L. The product is obtained by multiplying the multiplicand and the multiplier.

3.3. PDP Optimized Multiplier Using Iterative Carry Save SBETA

The demand for image and signal processing systems in portable electronic devices has been on the rise owing to the growing innovations in the field of semiconductor technology. An indispensable component of almost all image processing applications is multipliers. The low-power implementation of multipliers is tricky as it calls for high performance and speed to keep up with modern electronic devices' increasing clock frequencies.

The proposed work discusses about the design and implementation of a multiplier, based on the UT sutra of Vedic arithmetic, incorporating an Iterative Carry Save Selector-Based Error Tolerant Adder (ICS SBETA) for the partial product addition.

In the first phase of multiplication, a 4 × 4 multiplier using ICS SBETA is used as illustrated in Figure 9. The SBETA design and the partial products generated are also depicted in Figure 9.

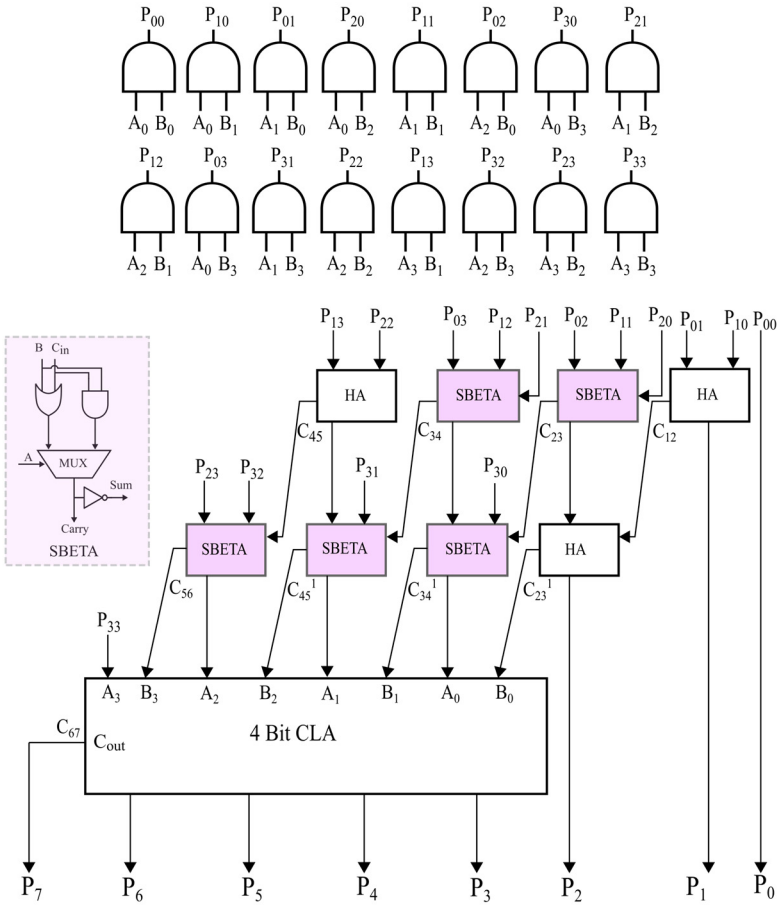

Figure 9. 4 × 4 Multiplier incorporating Iterative Carry Save SBETA.

In the second phase, an 8 × 8 multiplier using Iterative Carry Save SBETA is utilized to compute partial products generated in the first stage, as illustrated in Figure 10.

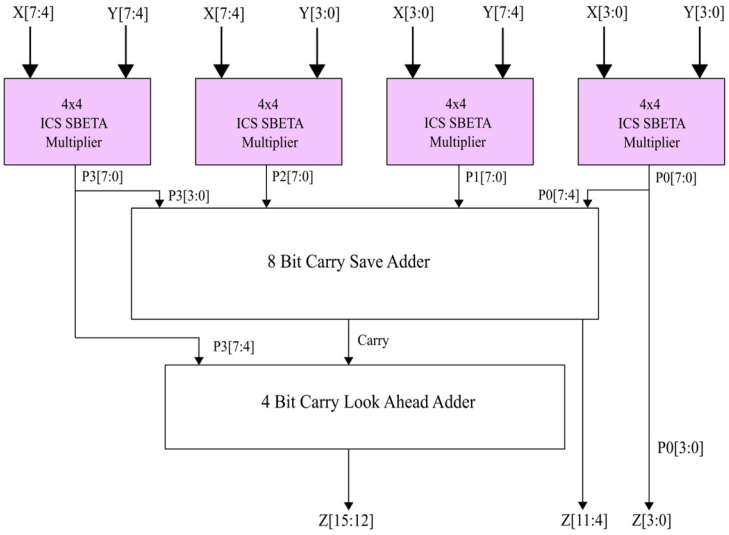

Figure 10. 8 × 8 Multiplier incorporating Iterative Carry Save SBETA.

Finally, in the third phase of multiplication, a 16 × 16 multiplier using ICS SBETA is utilized as depicted in Figure 11. Since ICS SBETA is incorporated into the multiplier design, the processing time for the summing of partial products is nearly halved. The arithmetic computation using the ICS SBETA is as given by Equation (1).

$$t_{ICS\ SBETA} = (k-2)t_{SBETA} + t_{CLA} \tag{1}$$

where k denotes the bit size.

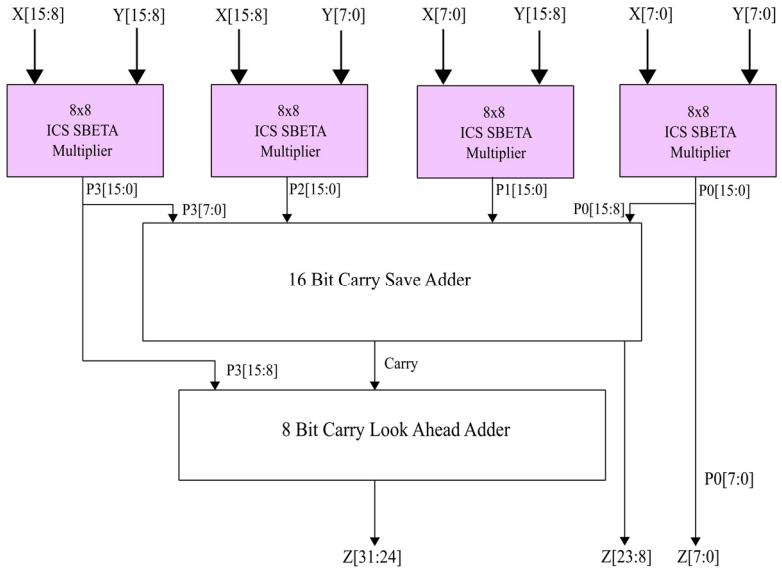

Figure 11. 16 × 16 Multiplier incorporating Iterative Carry Save SBETA.

3.4. Logic Unit Using Operand Isolation

The proposed logic module uses the operand isolation technique [34] to selectively restrict switching activity that propagates through the circuit, hence avoiding unnecessary operations. Operand isolation prevents specific circuit components from detecting changes to their input operands until the desired result is achieved. AND gates, which become

transparent when the computation results are required, are used at the circuits' inputs to achieve this. Operand isolation's fundamental concept is to power off specific logic blocks while no calculations are being processed. This can be achieved by shutting off the inputs to logic blocks when their outputs are unnecessary. As a result, the unwanted switching activity is minimized, reducing the power.

3.5. Application of the Proposed Computational Unit for Medical Image Compression

This research considers a medical image, a CT scan of the brain for aneurysm detection of size 512 × 512 taken from the open-access database of medical images [35]. Block style of DCT coding is followed in the DCT approach, in which the DCT is applied over blocks of a fixed size of 8 × 8. So, for an image of size 512 × 512, there are 4096 blocks to which DCT is applied to get the DCT of the entire image. The structure of the encoding of the digital image compression referred to in this research work is depicted in Figure 12.

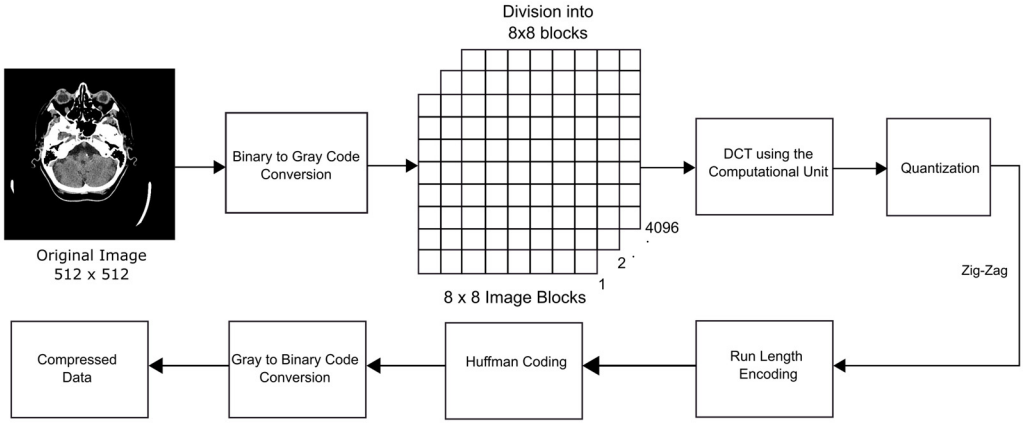

Figure 12. Encoding block of DCT Image Compression.

The compression ratios can be improvised by replacing the gray code with the binary code when the original digital image is represented and utilized in the image compression process [36]. A considerable improvement in the compression ratio was observed in this research work by utilizing gray code [37]. The original image in binary code is first converted to gray code using a binary-to-gray code converter. Gray coding offers a vibrant property of requiring a single bit change from a value "G" to "G + 1" or "G − 1" unlike binary codes, making gray coding an attractive addition to applications demanding lesser power consumption. Figure 13 illustrates the transitioning of bits for both gray and binary codes from numbers 0 to 15.

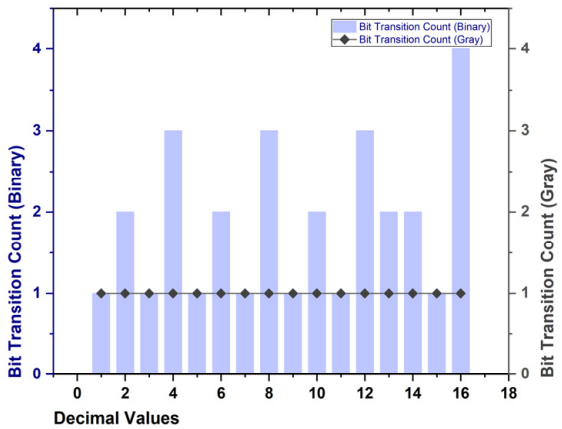

Figure 13. Bit Transitioning for Binary and Gray Codes.

The image is then divided into 8 × 8 blocks, to which DCT is applied. Several fast DCT algorithms have been proposed recently for the practical computation of image compression. Most of the proposed algorithms were focused on minimizing the addition and multiplication operations associated with the butterfly blocks incorporated in the DCT algorithms. The algorithm proposed by Loeffler proved to be the fastest, with just 11 multiplications and 29 additions, making it the ideal choice for this research work [38]. The structure of the 8-point DCT structure incorporating the computational unit in Loeffler's algorithm is illustrated in Figure 14. The adders used for computations in the butterfly diagram are replaced by the ET adders by incorporating the computational unit designed in this research work, reducing the power. The multiplications involved are reduced by approximating the cosine constants as given in Table 3 to one wherever applicable, and in cases where multiplications were required, it was achieved using the low-power multipliers and shifters, as illustrated in Figure 14.

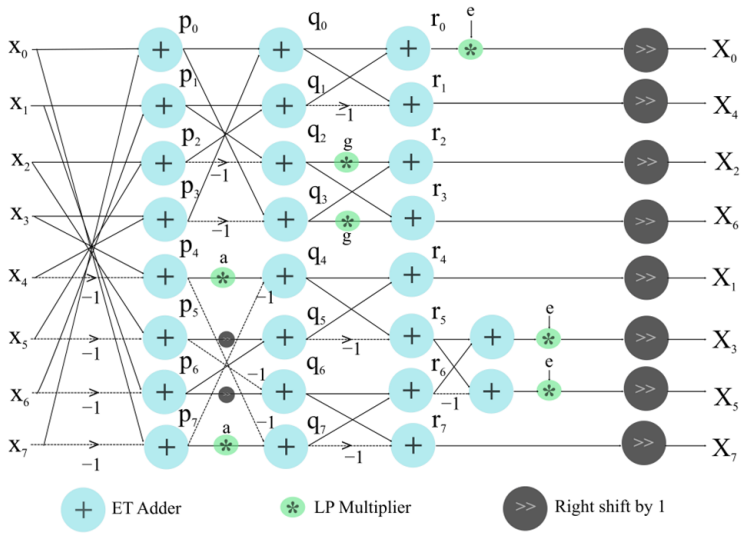

Figure 14. An 8-point Loeffler's DCT flow diagram incorporating the computational unit.

Table 3. Constant Factors used in DCT.

a	b	c	d	e	f	g	h
$sin\frac{\pi}{16}$	$cos\frac{\pi}{16}$	$sin\frac{3\pi}{16}$	$cos\frac{3\pi}{16}$	$cos\frac{\pi}{4}$	$cos\frac{\pi}{16}$	$sin\frac{\pi}{8}$	$cos\frac{\pi}{8}$
0.2	0.98	0.5	0.83	0.707	0.98	0.38	0.92

The bit-width of the digital input images is 10 bits, and after the row DCT, the output images' bit-width is 14 bits. The second column DCT and first row IDCT had an input and output bit-width of 14 bits. Finally, 14 was the bit-width of the last column IDCT, with the output image having a bit-width of 10 bits.

The next step is quantization, where the high-frequency DCT coefficients that are not imperative are reduced to 0. The quantization process is achieved by dividing each 8 × 8 block by a coefficient Q (u,v) obtained using the quantization matrix. As the division involved is an integer computation, most coefficients are 0, whereas the remaining coefficients are smaller numbers. A situation ideal for run length encoding and entropy encoding is obtained now. The quantization matrix determines the quality of the compression.

Larger values in the quantization table are chosen for lower quality, resulting in higher compression rates.

$$Q(u,v) = \begin{matrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 33 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{matrix}$$

A quality factor of 50 was utilized for the quantization process. Each matrix element should be multiplied by $(100 - q)/50$ for higher qualities; the multiplication factor is $50/q$ for lower qualities. The high-frequency components are eliminated without compromising the image quality. Higher numbers are located at the bottom right of the quantization table, whereas data with high-frequency and low-frequency values are located at the top left. This makes the high-frequency components to be rounded off to zero. Once the DCT coefficients are quantized, a zig-zag scanning pattern, as depicted in Figure 15, is followed to maintain the redundancy of zeroes due to the presence of several zeroes at the bottom left corner. The high-frequency components are reduced to zero once quantization is fulfilled, due to which the zig-zag pattern of scanning is pursued.

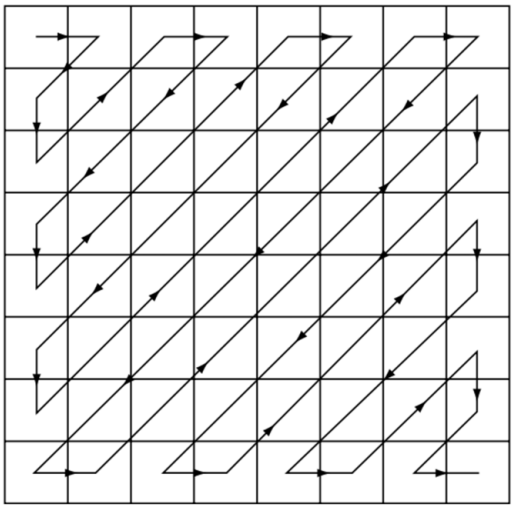

Figure 15. Zig-zag scan pattern.

The list of integer values obtained after zig-zag scanning generates the sequence of symbols. Run Length Encoding (RLE) is the next step since there are many zeroes from which each number, along with the number of times it is repeated, is obtained. The quantized coefficients are sampled using the zig-zag scan pattern from the lowest to highest frequencies to generate longer runs. The data stream obtained has a length of 64 bytes, which has a high probability of zeros toward the end. The RLE of the stream obtained is highly efficient.

After this, Huffman Coding must be carried out, where the numbers with higher frequency are denoted with lesser bits than the other numbers. RLE and Huffman Coding are performed to reduce the space required without removing any data. The data obtained are then converted back to binary format using the gray-to-binary code converter, after which the compressed image is obtained [38–42].

The reverse process is applied to obtain the reconstructed image using the decoding process illustrated in Figure 16. The Huffman-coded data are decoded, and the run-length encoded data are expanded into an 8×8 matrix to restore the matrix of quantized

coefficients. These are multiplied by the values from the quantization matrix Q (u,v). Next, the Inverse Discrete Cosine Transform (IDCT), which is the reverse of the DCT, is carried out on the values obtained in a similar manner using the computational unit for the computations involved. At this instant, data loss becomes evident as the DCT coefficients that were set to zero due to rounding remain zero after decompression.

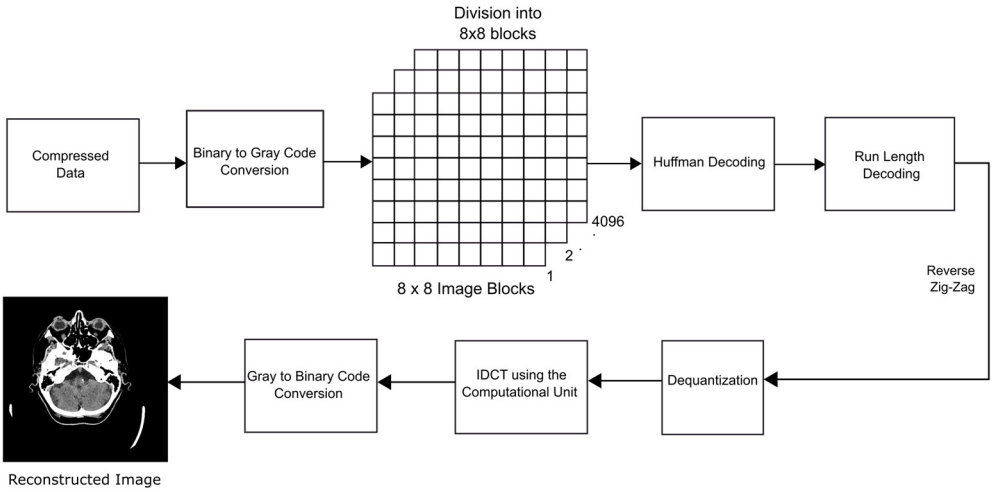

Figure 16. Decoding block of IDCT Image Compression.

4. Results and Discussion

The metrics used to evaluate and assess the architecture's performance are estimated by calculating the reconstructed image's Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Compression ratio (CR). The power, delay, area, PDP, and PSNR of various architectures based on DCT architectures are also assessed in this section.

Implementation of the Proposed Architecture for Image Processing Application

The implementation adopts standard-cell-based ASIC or semi-custom design built on preassembled library cells. Custom-designed standard cells are included in the library after being created. A few standard cells are available as a file in the IEEE library format. Through "place and route" Computer Aided Design (CAD) tools, these cells are subsequently used in the design by being arranged in rows and wired together. These cells are arranged physically and validated for specific process nodes. The Verilog HDL is used for the functional design in the RTL, which is mapped through a logic synthesis process into preassembled cells.

The designs proposed and from the literature are verified for their functionality by applying random inputs and examining the actual and anticipated results. It was observed that the anticipated and actual results matched for each design. The RTL verification of the design was performed using Cadence's incisive tool. The toggle information of all inputs and outputs is captured by a Value Change Dump (VCD) file generated during the design's functional verification. The VCD file is given as one of the inputs to the logic synthesis process. Logic synthesis is the process that transforms the technology-independent RTL into a technology-specific gate-level netlist optimized for a set of predefined constraints by using the process library. Logic synthesis starts with the RTL design, standard cell libraries, and design constraints. It completes with a gate-level netlist, mapped to the standard cell library that is efficient in terms of power, delay, and area.

The RTL defines how the design functions. The process library describes the logic gates' physical, electrical, and logical information. The logic synthesis Electronic Design Automation (EDA) tool converts the RTL into an optimized netlist. The process library serves to map this netlist into a technology-specific netlist. The Genus tool of Cadence with gpdk 90 and 45 nm standard cell libraries is used for the logic synthesis of the designs

proposed and from the literature. On completion of the logic synthesis, the next step of the ASIC design flow is physical design.

The 16-bit integrated computational units are designed, implemented, and synthesized using 45 nm gpdk technology libraries. The statistical parameter PSNR, as defined in Equation (2), calculates the ratio of the image strength to the noise power that affects the image quality.

$$PSNR = 10\, log_{10} \frac{Max^2}{MSE} \tag{2}$$

where Max is the maximum value of the image pixel. MSE denotes the Mean Squared Error (MSE) defined by Equation (3).

$$MSE = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} [I_o(x,y) - I_r(x,y)]^2 \tag{3}$$

where m and n denote each output image's number of rows and columns and I_o = Original image and I_r = reconstructed image.

A PSNR of higher value indicates that the reconstructed image is of better quality. The SSIM that predicts the reference and noisy image quality is another quality metric that is used in this research denoted by Equation (4).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{4}$$

where μ_x is the pixel sample mean of x, μ_y is the pixel sample mean of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, σ_{xy} is the covariance of x and y, $C_1 = (k_1 L)^2$, $C_2 = (k_2 L)^2$ are two variables to stabilize the division with a weak denominator, L is the dynamic range of the pixel values (typically, this is $2^{\#\ of\ bits\ per\ pixel} - 1$), k_1 is 0.01, and k_2 is 0.03 by default.

SSIM values vary between 0 and 1. An SSIM value of 0 signifies that the two images are different. A higher SSIM value indicates a higher-quality image. The metric that evaluates the compression algorithm performance is defined by the CR given by Equation (5).

$$Compression\ Ratio\ (CR)(\%) = \frac{Output\ Image\ size\ (bytes)}{Input\ Image\ size\ (bytes)} \tag{5}$$

The performance in terms of power, critical delay, area, PDP, SSIM, and PSNR is tabulated in Table 4, and the PDP-PSNR Plot of the computational unit and the reconstructed image is presented in Figure 17.

The PDP of the overall image encoding and decoding computations are reduced by the incorporation of the modified CSLA-based ET adders, the divide and conquer multiplier, and the iterative CSA multiplier in the proposed computational unit using the Loeffler algorithm for DCT and IDCT computations. The analysis of Table 4 indicates an improvement in performance metrics of the power, delay, area, and PDP of CU4 in comparison to the other implemented designs and the DCT designs from the existing literature considered in this research work due to the low-power concept of approximate computing that is used in the butterfly diagram computations of the DCT/IDCT operations involved in the encoding and decoding block of the architecture. A substantial improvement in PSNR was also observed in the reconstructed image obtained using CU4, as depicted in Figure 17. The reconstructed image obtained using the computational unit exhibited less distortion, making this design the ideal choice for medical image compression applications.

The synthesis results performed using 45 nm standard technology libraries indicated that the proposed implementation is better than the existing architectures listed in [46–48] in terms of PDP by 46.87%, 44.113%, and 33.88%, respectively, due to the incorporation of the computational unit in the Loeffler algorithm for the DCT/IDCT block used in the image compression process. Reducing the multiplications involved and utilizing the low-power

multipliers and the shifters minimize the power. The PSNR is considered as a Figure of Merit (FOM) to evaluate the performance of the computational unit. The PSNR of the proposed design was better than the PSNR of the existing architectures in [46–48] by 5.89%, 9.88%, and 6.14%. Compared to the existing DCT architectures, the SSIM obtained was 0.967, which was quite good.

Table 4. Power, Delay, Area, PDP, SSIM, and PSNR Analysis for various DCT Structures.

Technology: 45 nm						
Architecture	**Power (mW)**	**Delay (nS)**	**Area (μmm^2)**	**PDP (pJ)**	**SSIM**	**PSNR (dB)**
High-Performance 2D Transform [43]	11.89	32.86	23.78	393.10	0.898	37.766
Loeffler using AIE [44]	11.77	26.45	10.12	311.32	0.901	31.23
Cost-Efficient Loeffler algorithm [45]	8.17	21.5	9.5	175.65	0.886	31.45
EHA for DCT—Imp. Loeffler [46]	13.64	23.76	18.98	320.32	0.928	38.33
DIC using Approx. Addition [47]	15.31	19.89	21.34	304.51	0.915	36.94
DCT using Imp. B.G Lee [48]	9.72	26.48	17.69	257.38	0.906	38.24
CU1	11.16	22.12	19.07	246.86	0.953	40.24
CU2	11.27	22.13	19.06	249.4	0.961	39.67
CU3	11.29	22.14	19.05	249.96	0.965	39.54
CU4	9.31	18.28	14.63	170.18	0.967	40.59
CU5	9.32	18.31	14.62	170.65	0.958	39.24
CU6	9.23	18.61	14.61	171.77	0.962	39.14

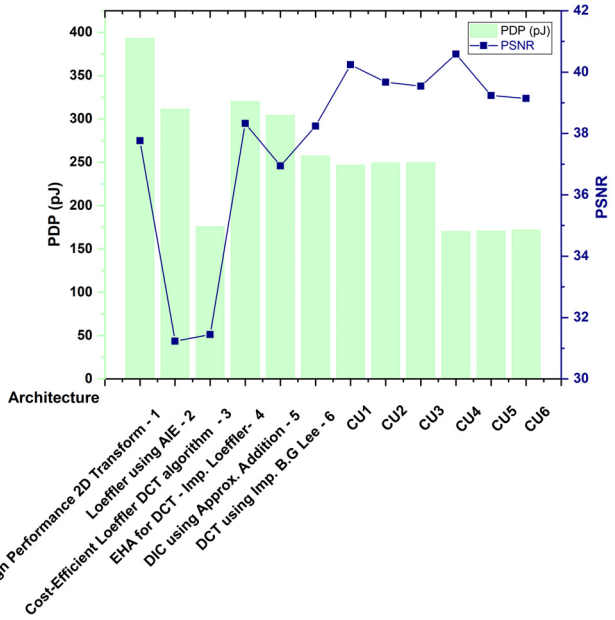

Figure 17. PDP-PSNR Plot of various DCT architectures and the proposed Computational units 1—[43], 2—[44], 3—[45], 4—[46], 5—[47], 6—[48].

The SSIM, PSNR, and CR values of the reconstructed and original CT scan images obtained by incorporating the computational unit CU4 in the Loeffler algorithm for the DCT and IDCT computations are illustrated in Figure 18. SSIM and PSNR are used in image processing as the figure of merit. An image with less noise and distortion is generally preferable as it gives a high signal-to-noise ratio. In image processing, high values of SSIM and PSNR indicate a good-quality image.

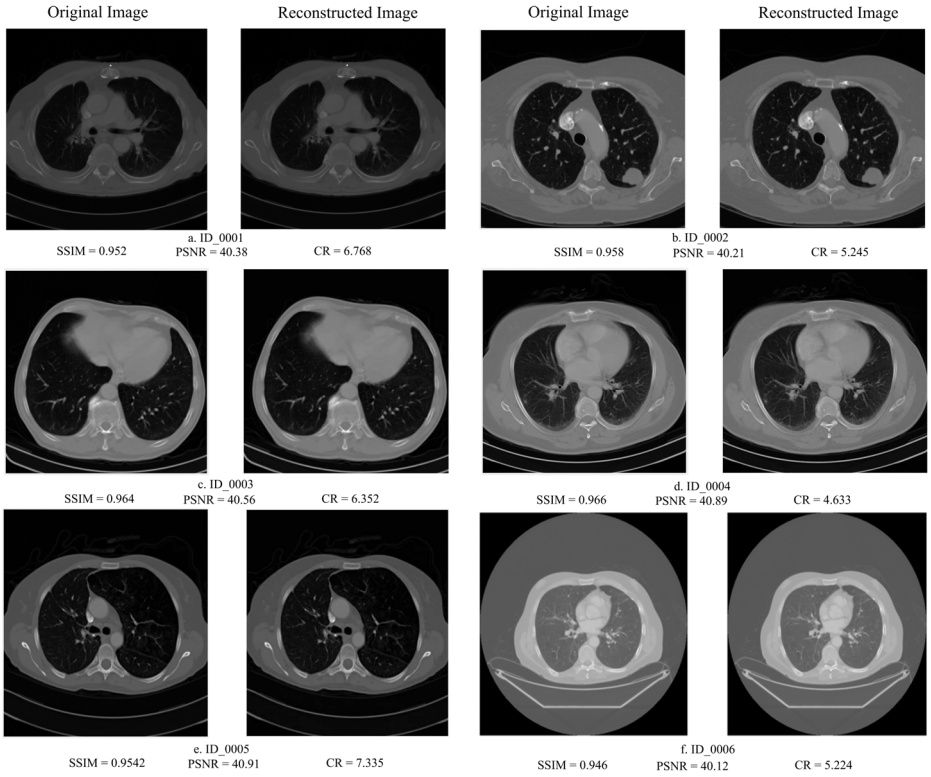

Figure 18. Original and reconstructed CT scan images obtained using image compression.

The physical design was obtained using the place and route option of the Cadence Innovus tool. The physical design of the computational unit proposed and selected for the final implementation of the digital image compression process is depicted in Figure 19.

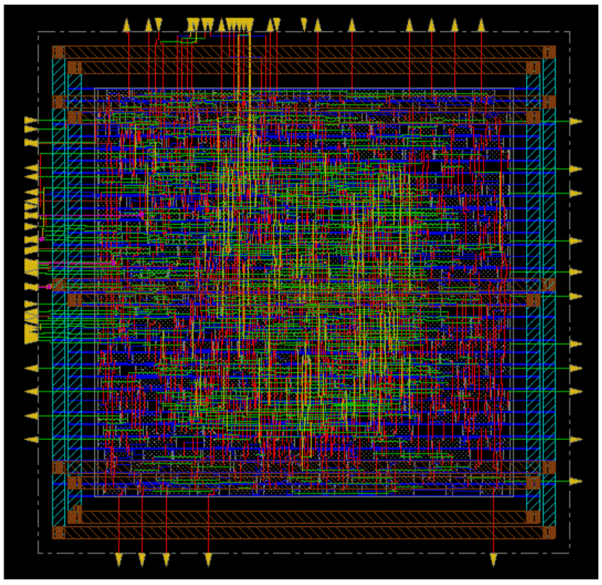

Figure 19. Physical Layout Design of the computational unit used for image compression.

5. Conclusions

The ASIC design of computational units is vital for processors, especially in contemporary technologies. The research work's prime focus was designing, developing, and implementing an ASIC-based computational unit using the concepts of low-power VLSI, mainly approximate computing, multipliers based on UT sutra utilizing ICS SBETA,

operand isolation, and low-power multipliers for stand-alone medical image processing applications. This research covers implementing an ASIC-based computational unit for medical image compression applications. The synthesis was performed with standard semi-custom ASIC design flow using the Genus tool of Cadence. In addition, 45 nm gpdk technology libraries were used to synthesize the image compression architecture designed using the computational unit with Loeffler's algorithm, resulting in improved performance compared to existing architectures. The PSNR and SSIM of the proposed design were better than the existing architectures due to the incorporation of the computational unit. A significant reduction in PDP and an improved PSNR for VLSI implementations was observed in the synthesis results when carried out on a medical image

The research work can be explored further by including the ET adders in convolution or digital filter applications involving addition. Future work involves replacing the standard libraries with the optimized digital libraries from Taiwan Semiconductor Manufacturing Company Limited (TSMC) and realizing the design up to chip fabrication. The design can be extended further to make it suitable for the high real-time performance required for video compression coding. The incorporation of low-power techniques can be explored to reduce power. Other multiplier and adder designs can be analyzed to form different combinations of computational units suitable for image processing applications. Different signal processing algorithms can also be explored and implemented using the proposed computational unit.

The developed computational unit for image compression that was utilized in the DCT algorithm demonstrates significant potential for direct application in medical image compression. The DCT-based computational unit can substantially reduce the storage requirements, enabling healthcare providers to maintain comprehensive image archives without the necessity of extensive physical storage infrastructure. The hospitals can save on storage costs and enhance data management practices, facilitating quicker retrieval and review of patient images by efficiently compressing images. Compressing medical images reduces their file size, which can be advantageous for encryption processes. Smaller file sizes can be encrypted more quickly and with potentially less computational overhead, thus enhancing the overall security of medical image data. Ensuring the confidentiality and integrity of medical images is critical, and the DCT-based computational unit can play a vital role in facilitating secure storage and transmission protocols.

The computational unit can be used to efficiently handle medical images on mobile platforms, allowing healthcare providers to review and share images using smartphones and tablets. This flexibility supports a wide range of healthcare scenarios, from emergency response to routine check-ups, enhancing the accessibility and responsiveness of medical services. In wearable medical devices, DCT compression can compress and transmit data, reducing power consumption and increasing device lifespan.

The implementation of this computational unit not only addresses current challenges but also paves the way for future advancements in medical image processing. Continued development and optimization could lead to even more efficient compression algorithms, further enhancing the utility and applicability of this technology in diverse medical environments. Additionally, integrating machine learning techniques could refine the compression process, ensuring that diagnostically relevant features are preserved with even greater precision.

The DCT-based computational unit designed for image compression presents an effective tool for medical image compression, with direct applications that can significantly enhance storage, transmission, and overall management of medical images. Its integration into healthcare systems promises to improve operational efficiencies, reduce costs, and ultimately contribute to better patient care outcomes.

Author Contributions: Conceptualization, T.M. and S.G.N.; methodology, T.M.; software, T.M.; validation, T.M., S.G.N., T.P., and V.K.K.; formal analysis, T.M.; investigation, T.M., T.P., and S.G.N.; resources, T.M. and S.G.N.; writing—original draft preparation, T.M.; writing—review and editing,

T.P. and V.K.K.; visualization, T.M.; supervision, S.G.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors thank Manipal Academy of Higher Education for offering them the T.M.A. Pai scholarship to complete this research. The authors would also like to thank the Department of Electronics and Communication Engineering at Manipal Institute of Technology for providing the necessary laboratory environment and resources.

Conflicts of Interest: The authors declare no conflict of interest.

List of Abbreviations

ALU	Arithmetic Logic Unit
ASIC	Application-Specific Integrated Circuits
BEC	Binary to Excess 1 Converters
CAD	Computer Aided Design
CT	Computed Tomography
CPU	Central Processing Unit
CR	Compression ratio
CSA	Carry Save Adder
CSLA	Carry Select Adder
DCT	Discrete Cosine Transform
DIP	Digital Image Processing
EDA	Electronic Design Automation
ET	Error-Tolerant
FOM	Figure of Merit
GDI	Gate Diffusion Input
GPP	General-Purpose Processors
HWT	Haar Wavelet Transform
IC	Integrated Circuit
ICSBETA	Iterative Carry Save Selector-Based Error Tolerant Adder
IDCT	Inverse Discrete Cosine Transform
LPSBETA	Low-Power Selector-Based Error Tolerant Adder
MAS	Multiplier Adder and Subtractor
MRI	Magnetic resonance imaging
OETA	Optimized Error Tolerant Adder
PDA	Power Delay Area
PDP	Power Delay Product
PSNR	Peak Signal-to-Noise Ratio
RCA	Ripple Carry Adder
RLE	Run Length Encoding
ROI	Region Of Interest
RTL	Register Transfer Level
SBETA	Selector-Based Error Tolerant Adder
SSIM	Structural Similarity Index
TSMC	Taiwan Semiconductor Manufacturing Company Limited
2D	Two-Dimensional
UT	Urdhva Tiryagbhyam
VCD	Value Change Dump
VLSI	Very Large-Scale Integration

References

1. Roy, K.; Prasad, S.C. *Low-Power CMOS VLSI Circuit Design*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
2. Lachireddy, D.; Ramesh, S. Power and delay efficient alu using vedic multiplier. In *Advances in Electrical and Computer Technologies: Select Proceedings of ICAECT 2019*; Springer: Coimbatore, India, 2020; pp. 703–711.
3. Sengupta, S.; Sarkar, P.; Dastidar, A. Design of a 4 bit arithmetic & logic unit, evaluation of its performance metrics & its implementation in a processor. In Proceedings of the 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 5–7 June 2020; pp. 1–8.
4. Kulkarni, M.A.L.; Baligar, J. Asic implementation of high speed and low power alu. *Int. J. Eng. Res. Technol.* **2019**, *8*.
5. Shirol, S.B.; Ramakrishna, S.; Shettar, R.B. A novel design and implementation of 32-bit hybrid alu. In *Computing and Network Sustainability: Proceedings of IRSCNS 2018*; Springer: Hubballi, India, 2019; pp. 239–249.
6. Kamaraj, A.; Marichamy, P. Design of integrated reversible fault-tolerant arithmetic and logic unit. *Microprocess. Microsyst.* **2019**, *69*, 16–23.
7. Samanth, R.; Chaitanya, C.; Nayak, G.S. Power reduction of a functional unit using rt-level clock-gating and operand isolation. In Proceedings of the 2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Manipal, India, 11–12 August 2019; pp. 1–4.
8. Telagam, N.; Kandasamy, N. Low power delay product 8-bit alu design using decoder and data selector. *Majlesi J. Electr. Eng.* **2018**, *12*, 103–108.
9. Hameed, M.; Khmag, A.; Zaman, F.; Ramli, A. Cmos technology for increasing efficiency of clock gating techniques using tri-state buffer. *Walailak J. Sci. Technol. (WJST)* **2017**, *14*, 327–338.
10. Mukhedkar, M.; Pandurang, W.B. A 180 nm efficient low power and optimized area alu design using gate diffusion input technique. In Proceedings of the 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), Pune, India, 24–26 February 2017; pp. 47–51.
11. Priyanka, M.; Ravi, T. Design and analysis of competent arithmetic and logic unit for risc processor. *ARPN J. Eng. Appl. Sci.* **2016**, *11*, 7141–7146.
12. Buzdar, A.R.; Sun, L.; Buzdar, A. Comparative analysis of alu implementation with rca and sklansky adders in asic design flow. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*. [CrossRef]
13. Shrivastava, G.; Singh, S. Power optimization of sequential circuit based alu using gated clock & pulse enable logic. In Proceedings of the 2014 International Conference on Computational Intelligence and Communication Networks, Bhopal, India, 14–16 November 2014; pp. 1006–1010.
14. Akram, S.M.; Rani, V.L.; Sailaja, K. Implementation of low leakage and high performance 8-bit alu for low power digital circuits. *Int. J. Comput. Appl.* **2013**, *82*, 24–28. [CrossRef]
15. Pandey, B.; Yadav, J.; Pattanaik, M.; Rajoria, N. Clock gating based energy efficient alu design and implementation on fpga. In Proceedings of the 2013 International Conference on Energy Efficient Technologies for Sustainability, Nagercoil, India, 10–12 April 2013; pp. 93–97.
16. Huddar, S.R.; Rupanagudi, S.R.; Janardhan, V.; Mohan, S.; Sandya, S. Area and speed efficient arithmetic logic unit design using ancient vedic mathematics on fpga. In *International Conference on Advances in Computing, Communication and Control*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 475–483.
17. Kamaraju, M.M.; Kishore, K.L.; Tilak, A. Power optimized alu for efficient data path. *Int. J. Comput. Appl.* 2010; 11, 39–43. [CrossRef]
18. Wu, Y.-G. Medical image compression by sampling DCT coefficients. *IEEE Trans. Inf. Technol. Biomed.* **2002**, *6*, 86–94.
19. Chen, Y.-Y. Medical image compression using DCT-based subband decomposition and modified SPIHT data organization. *Int. J. Med. Inform.* **2007**, *76*, 717–725. [CrossRef]
20. Potluri, U.S.; Madanayake, A.; Cintra, R.J.; Bayer, F.M.; Kulasekera, S.; Edirisuriya, A. Improved 8-point approximate DCT for image and video compression requiring only 14 additions. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2014**, *61*, 1727–1740. [CrossRef]
21. Kaur, M.; Wasson, V. ROI based medical image compression for telemedicine application. *Procedia Comput. Sci.* **2015**, *70*, 579–585. [CrossRef]
22. Sridhar, V. Design of multiplier for medical image compression using Urdhava Tiryakbhyam sutra. *Int. J. Electr. Comput. Eng.* **2016**, *6*, 1140–1151.
23. Liu, F.; Hernandez-Cabronero, M.; Sanchez, V.; Marcellin, M.W.; Bilgin, A. The current role of image compression standards in medical imaging. *Information* **2017**, *8*, 131. [CrossRef] [PubMed]
24. Xing, Y.; Zhang, Z.; Qian, Y.; Li, Q.; He, Y. An energy-efficient approximate DCT for wireless capsule endoscopy application. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–4.
25. Kumar, U.A.; Jain, N.; Chatterjee, S.K.; Ahmed, S.E. Evaluation of Multiplier-Less DCT Transform Using In-Exact Computing. In Proceedings of the Second International Conference on Machine Learning, Image Processing, Network Security and Data Sciences, Silchar, India, 30–31 July 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 11–23.
26. Hosny, K.M.; Khalid, A.M.; Mohamed, E.R. Optimized medical image compression for telemedicine applications. In *Artificial Intelligence and Data Mining in Healthcare*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 119–142. [CrossRef]

27. Alzahrani, M.; Albinali, M. Comparative Analysis of Lossless Image Compression Algorithms based on Different Types of Medical Images. In Proceedings of the 2021 International Conference of Women in Data Science at Taif University (WiDSTaif), Taif, Saudi Arabia, 30–31 March 2021; pp. 1–6.
28. Mydin, M.A.; Alkawaz, M.H.; Ghafoor, K.Z.; Mohammad, O.F.; Johar, M.G. A Study on Medical Image Compression Techniques based on Huffman Coding and Discrete Wavelet Transform (DWT). In Proceedings of the 2021 IEEE 9th Conference on Systems, Process and Control (ICSPC 2021), Melaka, Malaysia, 10–11 December 2021; pp. 86–91.
29. Rahman, M.A.; Hamada, M.; Shin, J. The Impact of State-of-the-Art Techniques for Lossless Still Image Compression. *Electronics* **2021**, *10*, 360. [CrossRef]
30. LaMeres, B.J.; LaMeres, B.J. The modern digital design flow. In *Quick Start Guide to Verilog*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1–12. [CrossRef]
31. Taraate, V. ; *Taraate*; Meherishi. Advanced HDL Synthesis and SOC Prototyping; Springer: Berlin/Heidelberg, Germany, 2019.
32. Natarajan, V.; Nagarajan, A.K.; Pandian, N.; Savithri, V.G. Low power design methodology. In Very-Large-Scale Integration; 2018; p. 47. [CrossRef]
33. Mendez, T.; Nayak, S.G.; Kumar, P.V.; Kedlaya, K.V. Performance Metric Evaluation of Error-Tolerant Adders for 2D Image Blending. *Electronics* **2022**, *11*, 2461. [CrossRef]
34. Yeap, G.K. Practical Low Power Digital VLSI Design. Kluwer Academic Publishers: New York, NY, USA; Springer: Berlin/Heidelberg, Germany, 1998.
35. Computed Tomography (CT) of the Brain. Available online: https://www.kaggle.com/datasets/trainingdatapro/computed-tomography-ct-of-the-brain (accessed on 15 December 2023).
36. Abdat, M.; Bellanger, M.G. Combining gray coding and JBIG for lossless image compression. In Proceedings of the 1st International Conference on Image Processing, Austin, TX, USA, 13–16 November 1994; Volume 3, pp. 851–855.
37. Rabbani, M.; Jones, P.W. *Digital Image Compression Techniques*; SPIE Press: Bellingham, WA, USA, 1991.
38. Ochoa-Dominguez, H.; Rao, K.R. *Discrete Cosine Transform*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2019.
39. Haidekker, M. *Advanced Biomedical IMAGE Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
40. Jayaraman, S.; Esakkirajan, S.; Veerakumar, T. *Digital Image Processing*; Tata McGraw Hill Education: New Delhi, India, 2009.
41. Gonzalez, R.C.; Woods, R.E.; Masters, B.R. *Digital Image Processing*; Pearson International Edition: Bellingham, WA, USA, 2009.
42. Coelho, D.F.; Cintra, R.J.; Kulasekera, S.; Madanayake, A.; Dimitrov, V.S. Error-free computation of 8-point discrete cosine transform based on the Loeffler factorisation and algebraic integers. *IET Signal Proc.* **2016**, *10*, 633–640. [CrossRef]
43. Mert, A.C.; Kalali, E.; Hamzaoglu, I. High performance 2D transform hardware for future video coding. *IEEE Trans. Consum. Electron.* **2017**, *63*, 117–125. [CrossRef]
44. Coelho, D.F.; Nimmalapalli, S.; Dimitrov, V.S.; Madanayake, A.; Cintra, R.J.; Tisserand, A. Computation of 2D 8 $\times$ 8 DCT based on the Loeffler factorization using algebraic integer encoding. *IEEE Trans. Comp.* **2018**, *67*, 1692–1702. [CrossRef]
45. Chung, R.L.; Chen, C.W.; Chen, C.A.; Abu, P.A.; Chen, S.L. VLSI implementation of a Cost-Efficient Loeffler DCT algorithm with recursive CORDIC for DCT-based encoder. *Electronics* **2021**, *10*, 862. [CrossRef]
46. Zhou, Z.; Pan, Z. Effective hardware accelerator for 2d DCT/IDCT using improved Loeffler architecture. *IEEE Access* **2022**, *10*, 11011–11020. [CrossRef]
47. Balasubramanian, P.; Nayar, R.; Maskell, D.L. Digital Image Compression Using Approximate Addition. *Electronics* **2022**, *11*, 1361. [CrossRef]
48. Mendez, T.; Kedlaya, K.V.; Nayak, D.; Mruthyunjaya, H.S.; Nayak, S.G. A Novel ASIC Implementation of Two-Dimensional Image Compression Using Improved BG Lee Algorithm. *Appl. Sci.* **2023**, *13*, 9094. [CrossRef]