International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 3 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Reconfigurable Symmetric Dyadic Filter VLSI Architecture for High-Fidelity Medical Image Fusion

Nathi Manasa[1], Raja kumar Rudrarapu[1*]
[1]Department of Electronics & Communication Engineering, Vaagdevi Engineering College, Warangal, 506005, Telangana, India.
*Correspondence: Raja kumar Rudrarapu (raj.rudrarapu@gmail.com)

Abstract

The rapid growth of medical imaging has significantly increased the volume of diagnostic data generated worldwide, with recent healthcare reports indicating that more than 5 billion medical imaging examinations are performed annually and nearly 80% of clinical decisions rely on imaging-based diagnosis. However, existing wavelet-based Medical Image Fusion (MIF) systems often rely on fixed filter architectures that require many arithmetic units, memory resources, and dedicated hardware blocks, resulting in increased area utilization, power consumption, and processing latency. These limitations reduce scalability and restrict real-time deployment in portable and resource-constrained medical imaging devices. To address these challenges, this work proposes a Reconfigurable Symmetric Dyadic Filter (SDF) VLSI Architecture for High-Fidelity MIF, designed to provide flexible and efficient wavelet decomposition and reconstruction. The proposed architecture incorporates a controller-driven reconfigurable framework, filter coefficient ROM, optimized Multiply-Accumulate (MAC) unit, internal register structure, and coefficient storage RAM for approximation and detail coefficients. The controller dynamically configures filter operations, enabling adaptive processing while minimizing hardware redundancy. The symmetric dyadic filtering structure significantly reduces computational complexity, memory access overhead, and critical path delay, thereby improving throughput and energy efficiency. By supporting high-speed wavelet processing with reduced hardware resources, the proposed SDF architecture enables accurate and real-time medical image fusion while achieving enhanced image quality, lower power consumption, and improved VLSI implementation efficiency suitable for next-generation healthcare and IoMT-based diagnostic systems.

1. Introduction

Medical imaging has become one of the most important components of modern healthcare systems, with the global medical imaging market exceeding USD 45 billion and continuing to grow due to the increasing prevalence of chronic diseases, cancer, neurological disorders, and cardiovascular complications. According to healthcare reports, more than 5 billion medical imaging examinations are conducted annually worldwide, and nearly 80% of clinical decisions are supported by imaging data. Technologies such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT), Positron Emission Tomography (PET), Ultrasound, and Single Photon Emission Computed Tomography (SPECT) generate massive volumes of diagnostic information that must be processed efficiently to assist healthcare professionals in making accurate diagnoses. As healthcare systems continue to digitize, the need for rapid and reliable image processing solutions has become increasingly critical.

The growth of telemedicine, Internet of Medical Things (IoMT), cloud-based healthcare platforms, and AI-assisted diagnosis has further increased the demand for high-performance image processing systems. Modern hospitals and diagnostic centers routinely handle thousands of images every day, requiring efficient processing, storage, and transmission mechanisms. Medical image

fusion has emerged as an effective solution for integrating complementary information from different imaging modalities into a single image, thereby enhancing visualization and diagnostic accuracy. Such fused images provide physicians with better anatomical and functional information, enabling improved treatment planning and disease monitoring.

The increasing complexity of medical imaging workflows has simultaneously created significant challenges in computational processing. Advanced image fusion algorithms often involve intensive mathematical operations, multiresolution decomposition, and filtering processes that demand substantial hardware resources. Traditional software-based implementations frequently struggle to meet real-time requirements, particularly in portable medical devices and embedded healthcare systems. Consequently, VLSI-based image processing architectures have gained considerable attention due to their ability to provide high-speed computation, low latency, reduced power consumption, and efficient hardware utilization for next-generation healthcare applications.

1.1 Problem Definition

Existing medical image processing architectures frequently rely on conventional wavelet filter implementations that require multiple arithmetic units, memory buffers, and complex data transfer mechanisms. These architectures consume a significant amount of hardware resources during implementation on FPGA and ASIC platforms. The extensive use of multipliers, adders, and storage elements increases LUT utilization and register requirements, leading to inefficient resource allocation. As image resolutions continue to increase, the hardware complexity grows substantially, making traditional implementations less suitable for modern healthcare applications.

Another major limitation of existing architectures is the presence of long critical paths caused by multiple sequential arithmetic operations. The accumulation of delays across filtering, decomposition, and reconstruction stages restricts the maximum achievable operating frequency. Such latency becomes particularly problematic in real-time applications where rapid image acquisition and immediate diagnostic analysis are required. High-resolution multimodal image fusion further intensifies computational demands, often resulting in performance bottlenecks that limit system scalability and responsiveness.

Power consumption represents an additional challenge in current image processing architectures. Large numbers of switching components, memory accesses, and arithmetic operations contribute to increased dynamic and static power dissipation. Excessive power consumption not only reduces energy efficiency but also creates thermal management issues in compact embedded systems. Consequently, existing approaches often suffer from increased LUT utilization, higher propagation delays, elevated power consumption, reduced throughput, and limited scalability, thereby restricting their effectiveness in next-generation medical imaging platforms.

2. Literature Survey

Samantaray et al. [1] introduced a symmetric Daubechies wavelet filter bank architecture for image processing applications. The study focused on developing a VLSI-friendly implementation capable of performing wavelet decomposition and reconstruction with reduced computational complexity. The architecture utilized symmetry properties of Daubechies filters to simplify arithmetic operations and improve hardware efficiency. The design was evaluated in terms of area utilization, processing speed, and filtering accuracy. The architecture exhibited limited adaptability to varying image processing requirements and lacked dynamic reconfiguration capability.

Kumar and Gawande [2] presented a review of VLSI architectures for two-dimensional Discrete Wavelet Transform using multiplier-less computation techniques. The work analyzed various hardware implementations that replace multipliers with shift-and-add operations to reduce circuit complexity.

Different architectural approaches were compared with respect to hardware resources, speed, and power consumption. The study highlighted the advantages of arithmetic simplification in FPGA implementations. Comparative evaluation was performed to identify efficient design alternatives. The review primarily focused on architectural comparisons and did not provide a unified solution for optimizing multiple performance parameters simultaneously. Sahu et al. [3] developed a selective level skip DWT architecture for efficient lossy image compression. The methodology selectively omitted specific decomposition levels to reduce computational requirements while maintaining acceptable image quality. The architecture was implemented using a hardware-efficient framework to support real-time compression applications. Performance was analyzed through compression ratio and resource utilization metrics. The design demonstrated improved processing efficiency for image storage applications. Skipping decomposition levels may lead to loss of important image details, affecting reconstruction quality. Dai et al. [4] designed an area-efficient VLSI architecture for high-throughput computation of two-dimensional DWT. The framework emphasized optimized data flow management and parallel processing techniques to improve throughput. Hardware resources were carefully allocated to minimize silicon area while maintaining computational efficiency. The architecture supported continuous image processing operations with reduced latency. The design prioritizes throughput and area reduction but provides limited flexibility for adaptive filtering applications.

Kumar and Gawande [5] proposed a VLSI implementation of a 5/3 two-dimensional DWT using multiplier-less computation combined with Kogge-Stone adder structures. The architecture replaced complex multiplication operations with efficient arithmetic units to reduce processing overhead. Fast adder structures were incorporated to accelerate wavelet coefficient computation. FPGA implementation results demonstrated improvements in speed and hardware efficiency. The design targeted image compression and signal processing applications. The use of advanced adder networks increases routing complexity and may elevate overall design overhead. Tiwari et al. [6] developed a low-resource FPGA implementation of DWPT for power quality indices estimation. The methodology employed MAC-based wavelet filters to extract frequency-domain information from electrical signals. The architecture was optimized to operate under resource-constrained FPGA environments while maintaining estimation accuracy. Performance evaluation included hardware utilization and signal analysis metrics. The system demonstrated effectiveness for real-time power monitoring applications. The MAC-intensive operations can introduce additional latency and power consumption during continuous processing. Anju et al. [7] presented an energy-efficient image compression and encryption framework based on adaptive lifting wavelet transform and CORDIC optimization. The methodology integrated compression and security functionalities within a unified VLSI architecture. Adaptive wavelet decomposition enhanced compression efficiency, while CORDIC optimization reduced arithmetic complexity. The framework targeted low-power image transmission systems. The integration of compression and encryption stages increases architectural complexity and design verification effort. Kannan et al. [8] developed a low-power VLSI architecture for ECG signal detection using a transpose-form retimed delayed LMS filter integrated with an advanced contextual convolutional attention network. The framework combined adaptive signal filtering with intelligent feature extraction mechanisms. Hardware optimization techniques were employed to reduce power consumption while preserving detection accuracy. The architecture was validated using biomedical signal datasets. The incorporation

of complex neural processing modules increases hardware resource requirements.

Arunlal and Priyadharson [9] introduced a recurrent adaptive wavelet-enhanced noise removal framework for smart grid meter integrated circuits. The methodology utilized wavelet-based signal enhancement combined with adaptive regulation mechanisms to suppress measurement noise. The system continuously adjusted processing parameters based on signal variations. Hardware implementation focused on improving signal quality and measurement reliability. The architecture supported smart grid monitoring applications. Adaptive processing mechanisms increase computational overhead and may affect real-time responsiveness. Priyanka and Satheesh [10] developed a VLSI-based medical image fusion framework employing an average fusion rule to enhance diagnostic visualization. The methodology combined information from multiple medical images to generate a fused image with improved interpretability. Hardware implementation focused on accelerating the fusion process for real-time healthcare applications. Performance evaluation included image quality assessment and processing efficiency. The framework demonstrated improved visualization of diagnostic features. Average-based fusion may suppress important modality-specific information and reduce feature preservation accuracy. George and Jayakumar [11] presented a hardware-efficient FrWF-based architecture for joint image dehazing and denoising. The methodology utilized fractional wavelet processing to simultaneously remove haze and noise from visual sensor images. The architecture optimized arithmetic operations to achieve real-time performance. Hardware evaluation demonstrated efficient image restoration capabilities. Experimental results confirmed improvements in visual quality under degraded imaging conditions. Simultaneous dehazing and denoising operations increase computational complexity and memory requirements. Islam et al. [12] developed an FPGA implementation of an image enhancement algorithm based on a novel Daubechies wavelet filter bank. The methodology applied wavelet decomposition and enhancement strategies to improve image contrast and visibility. Hardware implementation focused on accelerating enhancement operations using FPGA resources. The architecture supported real-time image processing applications. The architecture remains dependent on fixed filter structures, limiting runtime adaptability. Agarwal et al. [13] introduced the FF-GFA architecture for power- and area-efficient Gabor filter implementation. The methodology employed flipped and folded processing structures to reduce hardware resource utilization. The design optimized filter computation through efficient data reuse strategies. Hardware evaluation demonstrated reductions in power consumption and silicon area. The architecture was targeted toward embedded image processing systems. The design is application-specific and offers limited flexibility for other wavelet-based processing tasks. Sophia and Kavitha [14] developed an architecture-based framework for medical image compression using dyadic wavelet filters and context-adaptive encoding. The methodology combined multiresolution decomposition with adaptive compression mechanisms to improve storage efficiency. Matrix-based post-processing was incorporated to enhance reconstructed image quality. Hardware-oriented implementation supported efficient medical image handling. Experimental analysis demonstrated favorable compression performance. Additional encoding and post-processing stages increase computational latency and memory usage.

3. Proposed System

The VLSI-based Reconfigurable SDF architecture as shown in Figure 4.1 designed for mother wavelet selection in integrated data monitoring enables efficient real-time signal processing for biomedical applications such as MIF analysis. In data monitoring systems, accurate detection of subtle waveform variations is critical for diagnosing cardiac abnormalities like myocardial ischemia.

Implementing the wavelet transform in hardware using VLSI techniques significantly improves processing speed, reduces latency, and minimizes power consumption compared to software-based implementations. The proposed architecture uses a controller, filter coefficient memory, MAC unit, and coefficient storage modules to perform wavelet decomposition efficiently. By selecting appropriate mother wavelet coefficients and performing fast convolution operations, the system extracts approximation and detail components of the MIF signal, which are essential for detecting data deviations in real time.

Step 1: Input Signal Acquisition and Routing: The operation begins with the arrival of the input signal, denoted as "in", which represents the digitized MIF samples obtained from the biomedical acquisition system. The input passes through a multiplexer-based routing stage controlled by the controller. This stage manages the selection of incoming data streams and ensures that the correct sample sequence is forwarded to the processing pipeline. The routing mechanism allows sequential MIF samples to be fed into the computation unit while maintaining synchronization with the system clock.

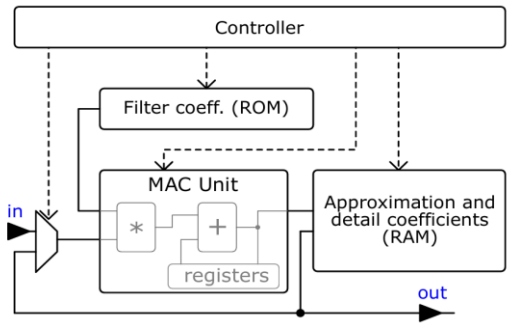

Figure 1: Proposed Reconfigurable SDF Architecture.

Step 2: Filter Coefficient Selection from ROM: Once the input signal enters the processing stage, the controller activates the filter coefficient memory (ROM) that stores predefined coefficients corresponding to the selected mother wavelet. These coefficients represent the low-pass and high-pass filters used in the wavelet transform. The controller determines which coefficient set to use depending on the selected wavelet basis for data analysis. The ROM supplies these coefficients sequentially to the processing unit so that the signal can be convolved with the appropriate filter kernels.

Step 3: MAC Processing: The filtered processing is performed by the MAC unit, which is the core computational component of the architecture. In this stage, each incoming MIF sample is multiplied with the corresponding filter coefficient retrieved from the ROM. The multiplication results are then accumulated using an adder and temporary registers within the MAC unit. This iterative multiplication and accumulation process implements the convolution operation required for the SDF. The use of dedicated hardware multipliers and adders ensures high-speed execution and efficient utilization of silicon resources.

Step 4: Register Storage and Intermediate Data Handling During the MAC operations, intermediate computation results are temporarily stored in registers inside the MAC unit. These registers hold partial sums and intermediate values required for completing the convolution process. The register stage also ensures pipeline synchronization, allowing continuous data flow without interruption. This mechanism reduces computational delays and supports high-throughput signal processing required for real-time MIF monitoring.

Step 5: Generation of Approximation and Detail Coefficients: After completing the MAC computations, the resulting values correspond to the wavelet decomposition outputs, which are divided into approximation and detail coefficients. These coefficients represent the low-frequency and high-frequency components of the MIF signal respectively. The generated coefficients are stored in the RAM module labeled "Approximation and detail coefficients". This storage allows subsequent stages of the monitoring system to analyze the signal characteristics related to data elevation or depression.

Step 6: Output Generation and Data Transmission: Finally, the processed wavelet coefficients are transferred to the output stage labeled "out." The controller manages the data flow from RAM to the output interface, ensuring that the processed information is available for further MIF analysis modules or decision-making algorithms. These outputs can then be used for feature extraction, abnormality detection, and clinical interpretation in integrated data monitoring systems.

Through this structured architecture, the proposed VLSI-based SDF efficiently performs wavelet decomposition while maintaining low hardware complexity and high computational speed. The design enables accurate real-time MIF signal analysis, making it highly suitable for portable and embedded cardiac monitoring devices.

3.1 Reconfigurable Controller

The controller serves as the supervisory unit of the fast wavelet architecture and is responsible for coordinating the operation of all hardware modules. It manages the timing, synchronization, and control signals required for data flow between the input interface, ROM, MAC unit, and RAM modules. By controlling the sequential execution of operations, the controller ensures that the wavelet transform process is performed accurately and efficiently within the hardware environment.

Step 1: System Initialization The controller begins the operation by initializing the system and enabling the required hardware modules. It activates the clock signals and prepares the processing units to receive the incoming MIF data samples.

Step 2: Input Data Control Once the system is initialized, the controller manages the routing of the incoming signal sample through the input interface. It ensures that the correct data is forwarded to the MAC unit for processing.

Step 3: Memory Access Coordination The controller generates address and enable signals for the ROM to retrieve the appropriate filter coefficients associated with the selected mother wavelet.

Step 4: Processing Synchronization During computation, the controller synchronizes the MAC unit operations with the coefficient retrieval process to maintain a continuous flow of data.

Step 5: Output Data Management Finally, the controller manages the transfer of computed results to the RAM module and ensures that the output data is stored correctly for further analysis.

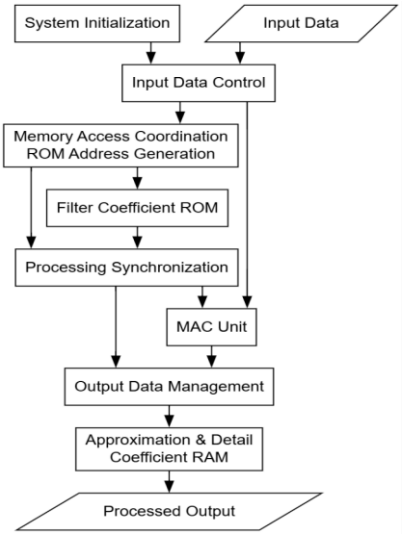

Figure 2: Proposed Reconfigurable Controller

3.2 Filter Coefficient Memory- ROM

The ROM module stores the predefined filter coefficients corresponding to the selected mother wavelet used in the Reconfigurable SDF. These coefficients represent the parameters of the low-pass and high-pass filters required to decompose the input MIF signal into its approximation and detail components. Since these coefficients remain constant during operation, ROM provides a reliable and efficient storage mechanism.

Step 1: Coefficient Storage: The ROM initially contains the filter coefficient values for the selected mother wavelet. These coefficients are stored in predefined memory locations and remain unchanged during operation.

Step 2: Address Generation The controller sends address signals to the ROM to access specific coefficient values required for the filtering operation.

Step 3: Coefficient Retrieval Upon receiving the address signal, the ROM retrieves the

corresponding coefficient and sends it to the MAC unit.

Step 4: Sequential Coefficient Delivery The ROM supplies the coefficients sequentially according to the filtering order required for the wavelet transform process.

Step 5: Continuous Data Supply This process continues until all coefficients required for the convolution operation are delivered to the MAC unit for signal processing.

3.3 MAC Processing Unit

The MAC unit as shown in Figure 3 performs the core arithmetic operations required to implement the wavelet filtering process. It executes the convolution between the input signal samples and the filter coefficients retrieved from the ROM. The MAC unit consists of multipliers, adders, and registers that work together to perform high-speed arithmetic computations.

Step 1: Input Data Reception: The MAC unit receives the input MIF signal samples from the input interface along with the filter coefficients from the ROM.

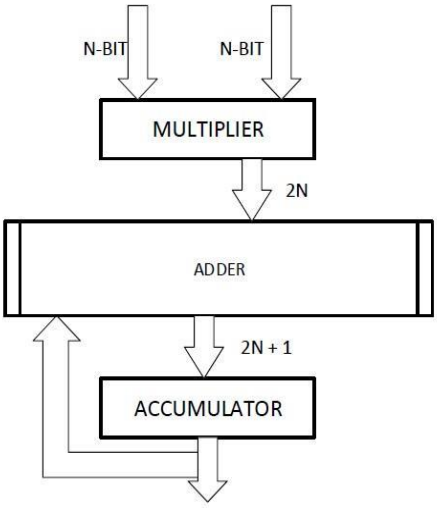

Figure 3: MAC Processing Unit

Step 2: Multiplication Operation: Each input sample is multiplied with the corresponding wavelet filter coefficient using the multiplier circuit.

Step 3: Accumulation Process: The resulting products are then accumulated using an adder to generate partial sums required for the convolution operation.

Step 4: Intermediate Result Storage: Intermediate computation results are stored in internal registers within the MAC unit to maintain data continuity during successive operations.

Step 5: Final Computation Output: After completing all multiplication and accumulation operations, the MAC unit produces the final filtered output corresponding to the wavelet transform stage.

3.4 Approximation and Detail Coefficient RAM

The Approximation and Detail Coefficient RAM module as shown in Figure 4 stores the filtered outputs generated by the MAC unit. In wavelet decomposition, the output signals are categorized into approximation coefficients, which represent the low-frequency components, and detail coefficients, which represent the high-frequency components of the MIF signal.

Step 1: Data Reception from MAC Unit: The RAM module receives the computed wavelet coefficients produced by the MAC unit.

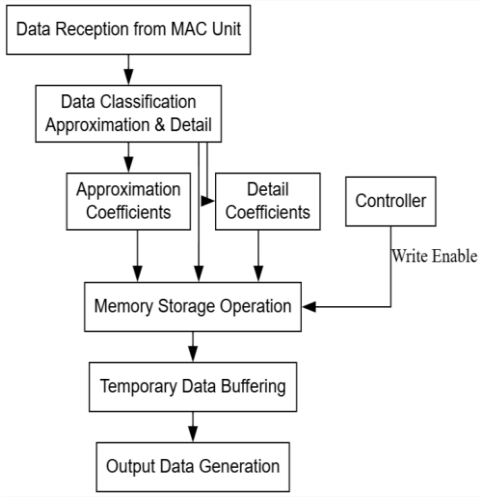

Figure 4: Approximation and Detail Coefficient RAM.

Step 2: Data Classification The received values are classified into approximation and detail components based on the filtering stage of the wavelet transform.

Step 3: Memory Storage Operation These coefficients are stored sequentially in RAM using write-enable signals generated by the controller.

Step 4: Temporary Data Buffering: The RAM acts as a temporary buffer that preserves the decomposition results until they are required for further analysis or signal reconstruction.

Step 5: Output Data Availability Finally, the stored coefficients are made available at the output stage for further MIF signal analysis, enabling accurate monitoring of data variations.

3.5 Decomposition steps of an SDF

Figure 5 illustrates the proposed decomposition architecture of a SDF with depth 2, which performs hierarchical signal decomposition using low-pass and high-pass filtering operations combined with down-sampling. The main objective of this architecture is to separate the input signal into approximation and detail components at different resolution levels. In the first stage, the signal is divided into low-frequency and high-frequency subbands, producing approximation and detail coefficients. The approximation component is then further decomposed in the second stage to obtain deeper frequency resolution. This two-level decomposition enables efficient multiresolution signal analysis while reducing computational redundancy compared with deeper multi-level implementations. The architecture generates three output coefficients: A_2 (second-level approximation), D_2 (second-level detail), and D_1 (first-level detail), which collectively represent the signal information across different frequency bands.

Step 1: Input Signal Initialization: The decomposition process begins with the input signal labeled "in." This signal contains the complete information of the original signal and is simultaneously fed into the first stage of the decomposition architecture.

Step 2: First-Level Low-Pass Filtering (LoD): The input signal is first passed through the Low-pass Decomposition filter (LoD). This filter extracts the low-frequency components of the signal, which represent the slowly varying or approximation part of the signal.

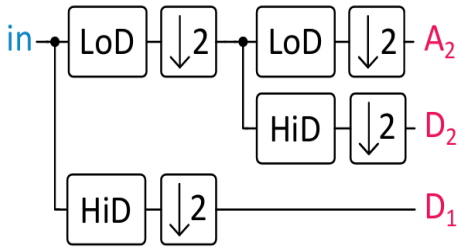

Figure 5: Decomposition steps of an SDF of depth 2.

Step 3: First-Level Down-Sampling (↓2): The output of the first LoD filter is then passed through a down-sampling block (↓2). This operation reduces the sampling rate by a factor of two, eliminating redundant samples and decreasing computational complexity. The resulting signal becomes the first-level approximation component, which will undergo further decomposition.

Step 4: First-Level High-Pass Filtering (HiD): Simultaneously, the input signal is also passed through the High-pass Decomposition filter (HiD). This filter extracts the high-frequency components, representing edges, rapid variations, or fine details of the signal.

Step 5: First-Level Detail Coefficient Generation: The output of the HiD filter is down-sampled using a down-sampling block (↓2). This produces the first-level detail coefficient denoted as D_1, which represents the high-frequency information captured at the first resolution level.

Step 6: Second-Level Decomposition of the Approximation Component: The approximation output obtained from the first stage is further processed in the second decomposition stage. This signal again passes through:

- **Low-pass Decomposition filter (LoD)**
- **High-pass Decomposition filter (HiD)**

These filters further separate the approximation signal into finer frequency components.

Step 7: Second Level Down-Sampling: The outputs of the second-stage filters are down-

sampled using down-sampling blocks (↓2). This produces two additional coefficients:

- **A_2** – Second-level approximation coefficient representing the coarse signal structure.
- **D_2** – Second-level detail coefficient representing intermediate frequency components.

Step 8: Final Decomposition Outputs: At the end of the depth-2 decomposition process, the architecture generates three wavelet coefficients:

- **A_2** – Second-level approximation component (lowest frequency band)
- **D_2** – Second-level detail component (middle frequency band)
- **D_1** – First-level detail component (highest frequency band)

These coefficients collectively represent the signal across multiple resolution levels and can be used for further processing such as compression, denoising, feature extraction, or hardware-based signal analysis.

4. Results and Discussions

Figure 6 illustrates the simulation results of the proposed Reconfigurable Symmetric Dyadic Filter architecture. The waveform shows the input signals p0[7:0] to p8[7:0] being applied at different simulation intervals, while the corresponding output is observed through Filter_out[8:0]. Initially, the output remains at 000 when all inputs are zero. As the input vectors change over time, the filter successfully generates corresponding output values such as 148, 0B8, 020, 054, 118, and 11A, demonstrating correct arithmetic processing and data propagation through the architecture. The simulation index signal increments sequentially from 00000000 to 00000005, confirming proper execution of the filtering operations. The obtained waveforms verify the functional correctness of the proposed design and demonstrate that the architecture accurately processes incoming data samples while producing stable and deterministic filter outputs.

Figure 6: Proposed Simulation Outcome

Figure 7 presents the FPGA area utilization summary of the proposed architecture. The design utilizes only 114 LUTs out of the available 134,600 LUT resources, resulting in an LUT utilization of merely 0.08%. Additionally, the architecture uses 73 I/O pins from the available 500 I/O resources, corresponding to an I/O utilization of 14.60%. Compared with the existing implementation that required 175 LUTs, the proposed architecture significantly reduces hardware resource consumption while maintaining identical I/O requirements. The reduction in LUT utilization demonstrates the effectiveness of the optimized filtering structure in minimizing logic complexity and improving hardware efficiency. Such low resource occupancy enables the architecture to support larger system integrations and multiple parallel processing modules within the same FPGA device.

Resource	Estimation	Available	Utilization...
LUT	114	134600	0.08
IO	73	500	14.60

Figure 7: Proposed Area Outcome

Figure 8 illustrates the power consumption characteristics of the proposed architecture. The total power consumption is measured at 12.034 W, comprising 11.888 W dynamic power (99%) and 0.146 W static power (1%). Within the dynamic power component, I/O power contributes 8.684 W (73%), representing the dominant source of power consumption, while signal power accounts for 2.369 W (20%) and logic power contributes only 0.834 W (7%). The static power consumption remains minimal at 0.146 W. Compared with the existing architecture, the logic power is significantly reduced from 1.580 W to 0.834 W,

indicating a substantial reduction in switching activity within the computational circuitry. The results demonstrate that the proposed architecture achieves improved energy efficiency through optimized hardware utilization while maintaining stable operational performance.

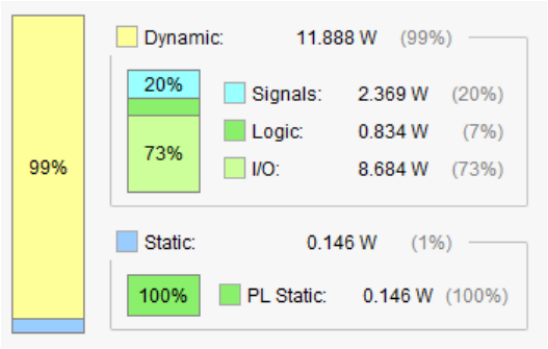

Figure 8: Proposed Power Summary

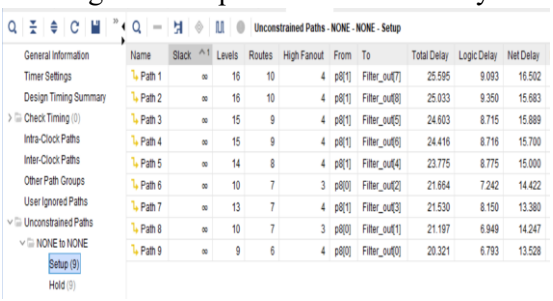

Figure 9: Proposed Setup Delay Outcome

Figure 9 shows the setup timing analysis of the proposed architecture. The most critical setup path, identified as Path 1, exhibits a total delay of 25.595 ns, consisting of 9.093 ns logic delay and 16.502 ns net delay, between source signal p8[1] and destination signal Filter_out[7]. Other significant paths show delays of 25.033 ns, 24.603 ns, 24.416 ns, 23.775 ns, and 21.664 ns. The setup paths contain between 9 and 16 logic levels, with routing stages ranging from 6 to 10 routes, while the maximum high fanout value is only 4. Compared with the existing architecture, which exhibited a maximum setup delay of 29.314 ns, the proposed architecture achieves a considerable reduction in timing delay. The reduced logic depth and routing complexity contribute to improved operating frequency, enhanced throughput, and better real-time processing capability.

Figure 10 presents the hold timing analysis results of the proposed architecture. The minimum hold delay is observed in Path 10, with a total delay of 6.038 ns, including 2.028 ns logic delay and 4.010 ns net delay, between p0[2] and Filter_out[3]. The remaining hold paths exhibit delays ranging from 6.059 ns to 6.750 ns, with logic delays varying between 1.998 ns and 2.106 ns, while net delays range from 3.961 ns to 4.643 ns. All hold paths contain only 5 logic levels and 3 routing stages, demonstrating a highly optimized timing structure. Compared with the existing architecture, which showed hold delays up to 7.729 ns, the proposed design significantly reduces hold timing overhead. The reduction in both logic and routing delays confirms that the proposed architecture provides faster signal stabilization, improved timing reliability, and enhanced overall performance for high-speed medical image fusion applications.

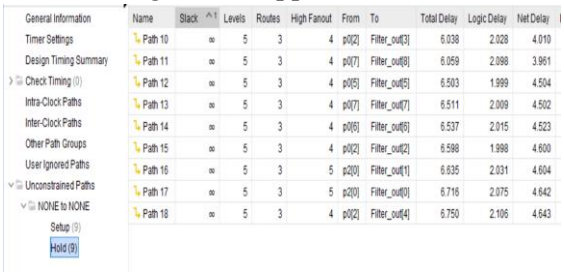

Figure 10: Proposed Hold Delay Outcome

4.1 Comparative Analysis

Table 1 presents the area utilization comparison between the existing architecture and the proposed SDF architecture. The existing design requires 175 LUTs, whereas the proposed SDF architecture utilizes only 114 LUTs, achieving a significant improvement of 34.86% in hardware resource utilization. Similarly, the LUT utilization percentage is reduced from 0.13% in the existing architecture to 0.08% in the proposed architecture, corresponding to an improvement of 38.46%. The reduction in LUT count demonstrates that the proposed SDF architecture employs a more efficient hardware structure with lower logic complexity. This optimization minimizes FPGA resource consumption and enables the integration of additional processing modules within the same device, making the proposed architecture more suitable for large-scale and high-performance medical image fusion applications.

Table 1 Area Comparison Between Existing and Proposed Architectures

Resource Metric	Existing Architecture	Proposed SDF Architecture	Improvement (%)
LUT Utilization	175	114	34.86
LUT Utilization (%)	0.13	0.08	38.46

Table 2 compares the power consumption characteristics of the existing and proposed SDF architectures. The dynamic power consumption decreases from 12.309 W in the existing architecture to 11.888 W in the proposed architecture, resulting in an improvement of 3.42%. Signal power is reduced from 2.567 W to 2.369 W, providing a 7.71% reduction. A substantial improvement is observed in logic power, which decreases from 1.580 W to 0.834 W, achieving a significant 47.22% reduction due to optimized computational circuitry. Static power consumption is also slightly reduced from 0.147 W to 0.146 W, corresponding to an improvement of 0.68%. Consequently, the total power consumption decreases from 12.456 W to 12.034 W, resulting in an overall improvement of 3.39%. These results confirm that the proposed SDF architecture effectively lowers power dissipation while maintaining high processing efficiency, making it more suitable for energy-constrained medical imaging systems.

Table 2 Power Consumption Comparison Between Existing and Proposed Architectures

Power Metric (uW)	Existing Architecture	Proposed SDF Architecture	Improvement (%)
Dynamic Power	12.309	11.888	3.42
Signal Power	2.567	2.369	7.71
Logic Power	1.580	0.834	47.22
Static Power	0.147	0.146	0.68
Total Power	12.456	12.034	3.39

Table 3 presents the setup timing comparison between the existing architecture and the proposed SDF architecture. The maximum setup delay is reduced from 29.314 ns to 25.595 ns, resulting in a 12.69% improvement. The logic delay decreases from 11.213 ns to 9.093 ns, achieving an improvement of 18.91%, while the net delay is reduced from 18.101 ns to 16.502 ns, corresponding to an improvement of 8.83%. The number of logic levels is significantly reduced from 22 to 16, providing a 27.27% improvement and indicating a shorter critical path. Similarly, the number of routing stages decreases from 14 to 10, resulting in a 28.57% improvement. The high fanout value is reduced from 13 to 4, representing the highest improvement of 69.23%, which contributes to reduced signal propagation overhead. Overall, the proposed SDF architecture demonstrates superior timing performance through reduced logic depth, lower routing complexity, and shorter setup paths, thereby enabling higher operating frequencies.

Table 3 Setup Delay Comparison Between Existing and Proposed Architectures

Timing Metric (ns)	Existing Architecture	Proposed SDF Architecture	Improvement (%)
Maximum Setup Delay	29.314	25.595	12.69
Logic Delay	11.213	9.093	18.91
Net Delay	18.101	16.502	8.83
Logic Levels	22	16	27.27
Routes	14	10	28.57
High Fanout	13	4	69.23

Table 4 compares the hold timing characteristics of the existing and proposed SDF architectures. The maximum hold delay is reduced from 7.729 ns in the existing architecture to 6.750 ns in the proposed architecture, achieving an improvement of 12.67%. The logic delay decreases from 2.551 ns to 2.106 ns, resulting in a 17.44% improvement, while the net delay is reduced from 5.178 ns to 4.643 ns, corresponding to an improvement of 10.33%. The number of logic levels decreases from 9 to 5, yielding a significant 44.44% reduction in timing complexity. Likewise, the number of routing stages is reduced from 5 to 3, providing an improvement of 40.00%. The high fanout value is reduced from 12 to 4, resulting in a substantial 66.67% improvement. These results indicate that the proposed SDF architecture achieves faster signal stabilization and improved timing reliability by minimizing logic depth, routing overhead, and fanout effects, thereby enhancing the overall performance of the medical image fusion system.

Table 4 Hold Delay Comparison Between Existing and Proposed Architectures

Timing Metric (ns)	Existing Architecture	Proposed SDF Architecture	Improvement (%)
Maximum Hold Delay	7.729	6.750	12.67
Logic Delay	2.551	2.106	17.44
Net Delay	5.178	4.643	10.33
Logic Levels	9	5	44.44
Routes	5	3	40.00
High Fanout	12	4	66.67

5. Conclusion

This work presented a Reconfigurable SDF architecture for high-fidelity medical image fusion and demonstrated its effectiveness through FPGA-based implementation and performance evaluation. The proposed architecture achieved significant hardware optimization by reducing LUT utilization from 175 LUTs to 114 LUTs, corresponding to an improvement of 34.86%, while LUT utilization percentage decreased from 0.13% to 0.08%, achieving a 38.46% reduction. In terms of power consumption, the architecture reduced dynamic power from 12.309 W to 11.888 W, signal power from 2.567 W to 2.369 W, logic power from 1.580 W to 0.834 W, and total power from 12.456 W to 12.034 W, providing an overall power improvement of 3.39% and a substantial 47.22% reduction in logic power. Timing analysis further confirmed the efficiency of the design, with setup delay reduced from 29.314 ns to 25.595 ns and hold delay reduced from 7.729 ns to 6.750 ns, resulting in improvements of 12.69% and 12.67%, respectively. Additionally, reductions in logic levels, routing complexity, and fanout values contributed to improved timing reliability and higher operating speed. These results demonstrate that the proposed SDF architecture provides a compact, power-efficient, and high-performance solution for real-time medical image fusion applications, making it highly suitable for FPGA-based healthcare systems, IoMT platforms, and next-generation VLSI image processing devices.

References

[1]. Samantaray, Aswini K., Amol D. Rahulkar, Satyajeet Sahoo, and Pradeep Gorre. "A new design and VLSI implementation of symmetric Daubechies wavelet filter bank for image processing applications." *Circuits, Systems, and Signal Processing* 44, no. 5 (2025): 3625-3640.

[2]. Kumar, Deepak, and Suresh S. Gawande. "Review Paper on VLSI Architecture for 2-D Discrete Wavelet Transform using Multiplier-less Technique." *International Journal of Research & Technology* 14, no. 1 (2026): 654-661.

[3]. Sahu, Prabhat Ku, Noel Prashant Ratchagar, Anubhav Bhalla, Abhinav Mishra, and Pallavi Pallavi. "VLSI Architecture Implementation of Selective Level Skip Discrete Wavelet Transform for Efficient Lossy Image Compression." In *2025 IEEE Madhya Pradesh Section Conference (MPCON)*, pp. 463-468. IEEE, 2025.

[4]. Dai, Yuzhou, Wei Zhang, Lin Shi, Qitao Li, Zhuolun Wu, and Yanyan Liu. "An area-efficient VLSI architecture for high-throughput computation of the 2-D DWT." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 33, no. 5 (2025): 1292-1303.

[5]. Kumar, Deepak, and Suresh S. Gawande. "VLSI Architecture for 5/3 2-D Discrete Wavelet Transform using Multiplier-less and Kogge Stone Adder Technique." *International Journal of Engineering Science & Humanities* 16, no. 1 (2026): 793-802.

[6]. Tiwari, Vinay K., D. Prakash Reddy, and Sachin K. Jain. "Low-Resource FPGA Implementation of DWPT for Power Quality Indices Estimation Using MAC-Based Wavelet Filters." *IEEE Transactions on Instrumentation and Measurement* 74 (2025): 1-10.

[7]. Anju, M. I., Rajesh Dey, and Sai Kiran Oruganti. "VLSI-based energy-efficient image compression and encryption framework using adaptive lifting wavelet transform and CORDIC optimization: A literature review." *SGS-Engineering & Sciences* 1, no. 1 (2025).

[8]. Kannan, Loganathan Mohana, Krishna Prakash Arunachalam, Rudramani Bhutia, and Elangovan Muniyandy. "Low-Power VLSI Architecture for ECG Signal Detection Using Transpose Form Retimed Delayed LMS Filter with Improved Progressive Cross Sea Lion Graph Contextual Convolutional Attention Network." *Biomedical Materials & Devices* (2026): 1-23.

[9]. Arunlal, C. K., and A. Selwin Mich Priyadharson. "Recurrent adaptive wavelet-enhanced noise removal with synchronized major-minor voltage regulator for smart grid meter IC." *Sādhanā* 50, no. 4 (2025): 343.

[10]. Priyanka, Endravath, and D. Satheesh. "VLSI-Based Medical Image Fusion Using Average Fusion Rule for Enhanced Diagnostic Visualization." *Excel International Journal of Technology, Engineering and Management* 12, no. 4 (2025): 28-40.

[11]. George, Anuja, and E. P. Jayakumar. "Hardware-efficient FrWF-based architecture for joint image dehazing and denoising framework for visual sensors." *Journal of Real-Time Image Processing* 22, no. 1 (2025): 14.

[12]. Islam, Md Tarikul, Aswini Kumar Samantaray, Pradeep Gorre, Deepak Naik Jatoth, Sandeep Kumar, AlaaDdin Al-Shidaifat, and Hanjung Song. "FPGA implementation of an effective image enhancement algorithm based on a novel daubechies wavelet filter bank." *Circuits, Systems, and Signal Processing* 44, no. 1 (2025): 365-386.

[13]. Agarwal, Priyanka, Pruthvi Parate, and Madhav Rao. "FF-GFA: Flipped Folded Architecture enabled Power-and Area-Efficient Gabor Filter Design." In *2025 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, vol. 1, pp. 1-6. IEEE, 2025.

[14]. Sophia, D., and S. Kavitha. "An architecture-based framework for enhancing medical image compression using dyadic wavelet filters and context adaptive encoding with matrix-based post-processing." *AIP Advances* 15, no. 9 (2025).