IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

# DATA COMPARATOR FOR MEDIAN FILTER IN DENOISING APPLICATION BASED ON AN AREA-EFFICIENT AND LOW-POWER MULTIPLEXER

**Mrs.T.SRUJANA**
Assistant professor,
Department of ECE,
Sree Dattha Institute of Engineering and Science,Hyderabad.

***Abstract :*** Filters are used to remove the different types of noises including salt and pepper, gaussian, and random noises from image. Therefore, the VLSI oriented hardware implementation of filters plays the crucial role in real time applications. However, the conventional hardware-based filters are failed to reduce the look-up-table (LUT)s, path delays, and power consumption. Therefore, this work is focused on implementation of Hybrid Median Filter (HMF) using Decision-based multiplexer logic, and adaptive neighborhoods logic. Initially, the multiplexer selection logic-based decision-based multiplexer is used to identify the high and low values from two numbers. Then, decision-based multiplexer is repeated for multiple number of times for nine pixels combinations,which identifies the median value from nine pixels. The subjective and objective valuation shows that the proposed real time impulse noise removal (RTINR) resulted in superior performance in terms reduced noise, hardware metrics like LUTs, delay, and power consumption as compared to state of art approaches.

***IndexTerms*** **– Decision based multiplexer, Real time impulse noise removal(RTINR), Hybrid Median filter.**

## I.INTRODUCTION

Noise is undesirable information which degrades image quality [1]. The image can be noisy because of dust present on the lens, electronic noise in camera, imperfection present in the image sensor or can be introduced when image data is transmitted over communication channel. The motive of image processing is to get rid of noise from a digital image while keeping its features unaltered. Image filter is the key blog of Image processing system. An impulsive noise can be added when image data transmitted over an insecure communication channel. [2]. It causes small size dots or dark/black spot on an image. Impulse noise is uniformly distributed and the most often mentioned noise in digital images. Further, Impulse noise can be divided into two parts. The first one is salt and pepper noise which is a type of impulse noise having noisy pixel intensity either 0 (minimum) or 255 (maximum) in the case of gray scale images. It appears as randomly scattered black or white dots over the images [3]. The second one is the random-valued shot noise which has arbitrary valued noisy pixels. To remove these noises, it is necessary that the acquired image must pass through an image pre-processing stage defined as a filter [4]. Spatial and frequency domain are two categories of the filtering operation. Generally, filters are implemented by MATLAB, OCTAVE software's in real time systems [5]. As it is a well-known fact that software implementation offers less processing speed in comparison to hardware implementation [6]. Hardware implementation has become better alternative after the boost in the VLSI technology. To reduce the power consumption in the systems, more cooling devices have to be incorporated results in the costly system. Keeping the same functional capabilities with the reduction in power factors are heavily demanding. Yet in that context, battery and power optimizing technology have not matured up to that target. Most of these products include embedded microprocessors, DSPs and ASICs [7]. It is a provoking undertaking to accomplish low force plan of any VLSI circuit. There are various degrees of advancement in VLSI configuration measure for low force applications. For battery operated portable products [8], power has been the main concern. As System-on-Chip (SoC) developing with more power transistors, it requires less power consumption. Power consumption reduction in highly integrated SoC cut down the heating problem. It reduces the cost of expensive packing and cooling mechanism [9]. In this work, VLSI architecture for noise reduction in different imaging applications is proposed to deal with the above issues of power and cost reduction, respectively. To achieve low resources, this work mainly focusing on Verilog based coding mechanisms with FPGA prototype [10].

Then, the subjective and objective image statistics are measured by using MATLAB environment. The major contributions of this work are as follows:

- Implementation of decision-based multiplexer for identifying the high, low values using multiplexer selection logic.
- Implementation of multi-level network for selection of median value from nine pixels in a window.
- Implementation of RTINR for removal of different types of noises from image using hybrid switching of data blocks.

## II.RESEARCH METHODOLOGY

Nikitha, Mylaram, and Divya Gampala , focused on implementation of Hybrid Median Filter (HMF) using Decision-based multiplexer (DC) logic. Initially, the multiplexer selection logic-based data comparator is used to identify the high and low values from two numbers. Then, decision-based multiplexer is repeated for multiple number of times for nine pixels combinations, which identifies the median value from nine pixels. The subjective and objective evaluation shows that the proposed MF-DC resulted in superior performance in terms reduced noise, hardware metrics like LUTs, delay, and power consumption as compared to state of art approaches . Bevara, Vasudeva, Bevara Srinu, and Pradyut Kumar Sanki (2022) proposed a new Decision-based Adaptive Denoising Filter (DBADF) algorithm & hardware architecture are proposed for restoring the digital image that is highly corrupted with impulse noise. The proposed DBADF detects only the corrupted pixels and that pixel is restored by the noise-free median value or previous value based upon the noise density in the image. The proposed DBADF uses a window initially and adaptively goes up to window based on the noise corruption more than 50% by impulse noise in the current processing window. The proposed architecture was found to exhibit better visual qualitative and quantitative evaluation based on PSNR, IEF, EKI, SSIM, FOM, and error rate. The proposed architecture has been simulated using the VIRTEX7 FPGA device and the reported maximum post place and route frequency are 149.995MHz and the dynamic power consumption is 179mW. Mounika, G., and K. Vasanth (2022) proposed on a 45nm based Decision-based multiplexer using different combinational style implementation is proposed. A Decision-based multiplexer is a combinational circuit that uses a carry as reference to select greatest and smallest of two numbers. The proposed decision-based multiplexer has carried generation unit implemented with a logic of subtraction using different combinational style such as decoder, multiplexer, basic subtractor logic etc. The proposed architecture was implemented on Virtuoso (Cadence) 45nm technology. Six Architectures were developed and compared with each other. It was found that carry select decision-based multiplexer requires low power. Borrow look ahead architecture operates at high speed and finally Area efficient architecture was decoder-based decision-based multiplexer.

Srinu, Boni, Srinu Bevara, and M. Nagendra Kumar (2019) proposed on the median filter with high throughput and good latency to suppress the impulse-based noise on real time signal and image processing applications. It is partially affected by the median filter and its bias of the input stream is directly above the average of mathematical analysis. An efficient VLSI suitable hardware implementation of a median filter is presented, that uses compare and exchange unit. The proposed hardware structure reduces the hardware requirements and has a faster processing speed, when compared with some other existing techniques. The input numbers or streams are used to construct an algorithm. By using this algorithm, the median number can be found out. The proposed technique can be implemented with perfect shuffle interconnects between active stages of compare and exchange elements. In this paper, all the designs are synthesized and created using MAX PLUS- II from ALTERA with fma = 486.38 MHz.

Karanam Rajini and K.Vasanth novel, low-power, multiplexer-based data comparator is proposed for use with the median filter in de-noising applications. The suggested solution makes use of a complete subtractor, which serves as the fundamental processing unit of a data comparator, to implement the borrow equation using multiplexers. For various technologies, a Novel Multiplexers-based Data Comparator was suggested. Only 2X1 multiplexers were used in the whole subtractor implementation for the multiplexer-based data comparators. The inputs will also be compared using a specially created Multiplexer based on the propagating borrow. Implementations based on multiplexers required 87% more power and consumed 63% more space. The proposed logic is ideal for low-power and low-area applications since it takes less space and power. Krishna Chaitanya Data comparators are key arithmetic units used in digital systems to know whether two numbers are equal, or one number is less than or greater than other number. which acts as a elementary processing element of a Data Comparator. The proposed designs are implemented in Dsch tool for gate level designs of AND gate ,OR gate, Multiplexer, Barrow generation unit and high and low value output units and verilog file is generated . Microwind tool is used to get layouts for different Models of Mosfet and different technologies. The multiplexer based data comparators contains full subtractor these full subtractor are implemented only using 2X1 multiplexers. The Multiplexer based data comparator implementations resulted in 65% of area and 88% of power consumption. The proposed logic requires less area and power compared to existing.

## III.PROPOSED METHOD

Commotion is signal-subordinate and is hard to be eliminated without disabling image subtleties. Various sorts of error influence the image, like Gaussian, drive, dot and Rician commotions. In the image denoising measure, data about the sort of error present in the original image assumes a huge part. The image error can be delegated either added substance or multiplicative. The image is a 2-D function $f(x, y)$ of light intensities, where $f$ is amplitude at any spatial coordinate $x$ and $y$. The beam of light falls on an object and reflected light reaches to eyes. It makes human to see the object. The smallest element of the image is pixels. Each pixel represents intensity value at a particular location. Mathematically, image can be represented as Equation (1).

$$F(x, y) = I(x, y).R(x, y) \quad (1)$$

Here, $I(x, y)$ is intensity of incident light on object, $R(x, y)$ is reflected light from object in intensity and $F(x, y)$ is intensity of resultant image. The image restoration is the process to denoising a image, which has been distorted by prior knowledge of degradation model. Once the degradation model is known, by applying inverse process to recover the desired imagery. image restoration is different than traditional image enhancement techniques. It is a subjective process, which produces more effective results to an observer with and without using degradation model. The image degradation process is shown in Fig 1. image degradation model in the spatial domain is achieved by performing the convolution between $f(x, y)$ and model function $(h(x, y))$.

$$F(x, y) = h(x, y) * f(x, y) + \eta(x, y) \quad (2)$$

Here, $\eta\ (x, y)$ represents the speckle noise. Further, degradation model in the frequency domain is achieved by applying the Fourier transform as follows:

$$F\ (u, v)\ = h\ (u, v) * f\ (u, v)\ +\ \ \eta(u, v) \qquad (3)$$

Here, $u, v$ represents the frequency domain coefficients.

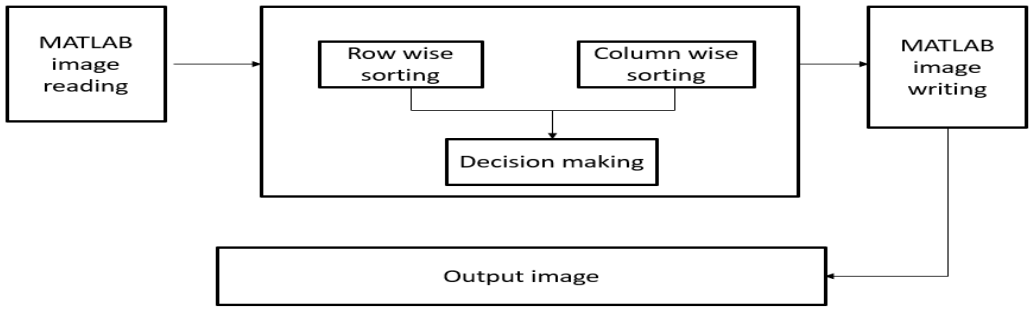


Fig 1: Proposed model

The RTINR is a digital non-linear method used to eliminate noise, similar to that of the medium filter. However, by keeping valuable details in the image, it typically does better than the mean filter. This filter class belongs to the class of filter that preserves the edge. These filters smooth down the data while maintaining the details. The median is only the average of all the pixel values in the area. It doesn't correspond to the mean (or average), but the median is half bigger and half smaller in the neighborhood. The median is a "center indication" stronger than the average. Like the medium, every pixel in the image is taken into account by the RTINR and its close neighbors are examined to determine if it is typical of their surroundings. It replaces the median value with those values instead of just replacing the pixel value by the mean of the next pixel value. Particularly better than the typical filter is to take away impulsive noise. The RTINR eliminates the noise as well as the fine details as the difference between them cannot be identified. Anything that is comparatively tiny in size with the area size will minimize and filter out the median value. In other words, the RTINR can differentiate between fine detail and noise.

**3.1 Decision-based multiplexer**

Fig 2 shows the block diagram of decision-based multiplexer, which is used to perform the selection of highest and lowest values from the given two input data. Further, the decision-based multiplexer block contains inputs as A, B and outputs are High (H) and Low (L).

Step 1: Initially, A < B condition is verified, if condition is satisfied selection line of multiplexer becomes one, else condition failed selection line becomes zero.

Step 2: Input-A is applied as Data-input-0 and Input-B is applied as Data-input-1 to 2to1 multiplexer. If A value is smaller than B, then selection line becomes one and multiplexer generates H as input-B through selection switching. If A value is higher than B, then selection line becomes zero and multiplexer generates H as input-A through selection switching

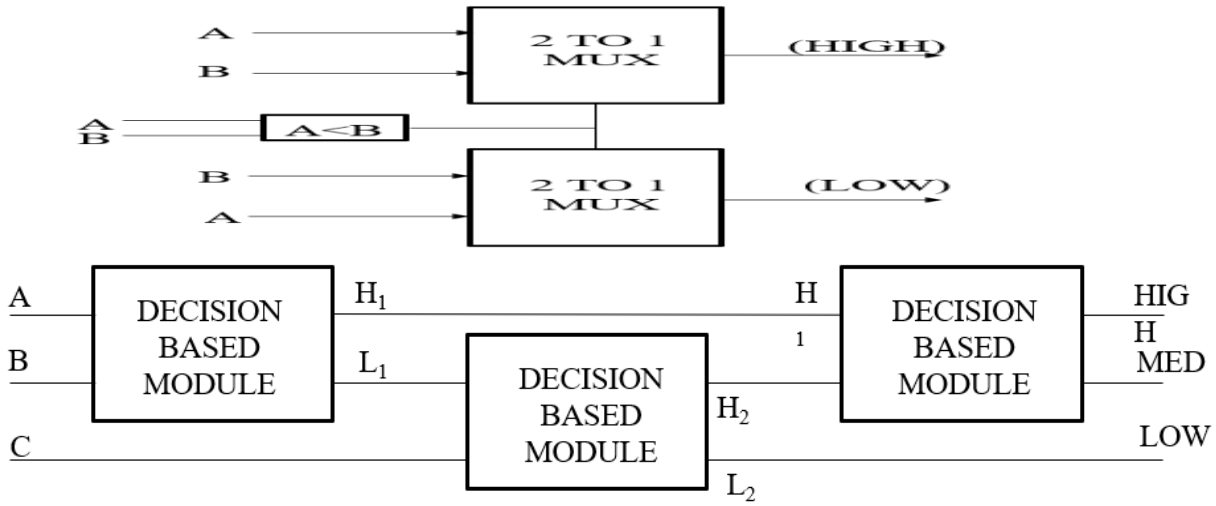


Fig 2. Block diagram of sub modules, (a) decision-based multiplexer, (b) adaptive neighborhood multiplexer.

Step 3: Input-B is applied as Data-input-0 and Input-A is applied as Data-input-1 to 2to1 multiplexer. If A value is smaller than B, then selection line becomes one and multiplexer generates L as input-A through selection switching. If A value is higher than B, then selection line becomes zero and multiplexer generates L as input-B through selection switching.

### 3.2 Hardware architecture of RTNIR

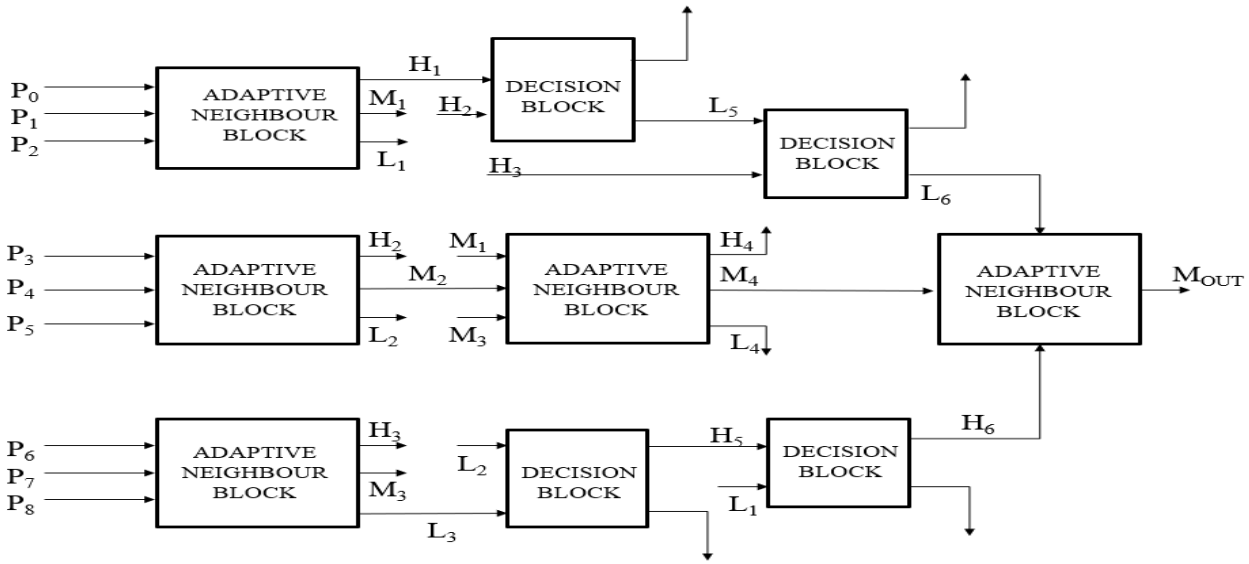


Fig 3. Hardware architecture of RTINR

Fig 3 shows the hardware architecture of RTINR, which contains the fourteen number of hardware resource blocks. Here, inputs P0, P1, P2, P3, P4, P5, P6, P7, and P8 are applied to RTINR, which generates the median value as M. Here, DC-1, DC-2, DC-3 are grouped together and performs the selection of high (H1), low (L1) and median (M1) values. Similarly, DC-6, DC-7, DC-8 and DC-15, DC-16, DC-17 performs the generation of high, low and median values. Further, DC-4 is used to select the lowest value from H1, H2, H3 outcomes. Furthermore, DC-18 is used to select the highest value from L1, L2, L3 outcomes. Similarly, DC-9, DC-10, DC-11 are grouped together and performs the selection of high, low and median values. Like this, the process will continue and generates the median value (M) from DC-14 low outcome.

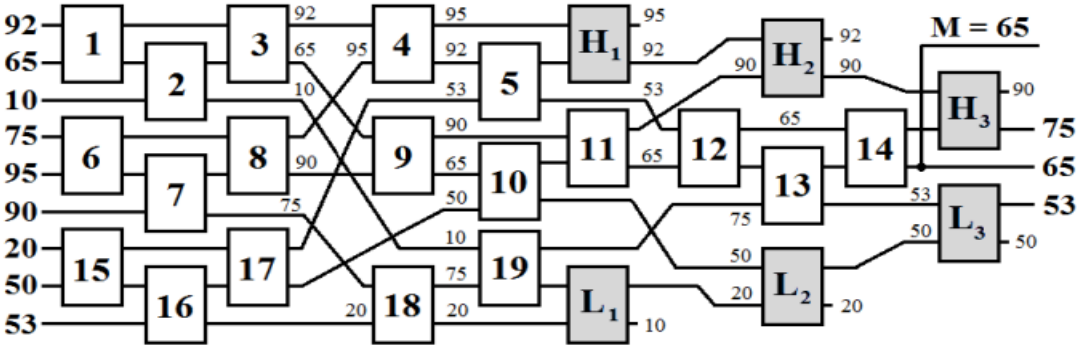


Fig 4. Example of RTINR.

Fig 4 provides a numerical example that might help better illustrate how the RTINR system works. In this case, the median value is defined by the two non-median outputs that are located the closest to it. As can be seen in this diagram, the H1, H2, and H3 blocks are used to sort the four highest pixel values (95, 92, 90, and 75), which results in 75 being the upper range. At the same time, the three lower ranges (L1, L2, and L3) are used to sort the four lowest values (10, 20, and 50), which results in 53 being the lower range.

## IV SIMULATION RESULTS AND DISCUSSION

Xilinx ISE software was used to create all of the RTINR designs. This software programmed gives two types of outputs: simulation and synthesis. The simulation results provide a thorough examination of the RTINR architecture in terms of input and output byte level combinations. Decoding procedure approximated simply by applying numerous combinations of inputs and monitoring various outputs through simulated study of encoding correctness. The use of area in relation to the LUT count will be accomplished as a result of the synthesis findings. In addition, a time summary will be obtained with regard to various path delays, and a power summary will be prepared utilizing the static and dynamic power consumption. Further, MatlabR2020a software is used to evaluate the subjective performance of RTINR.

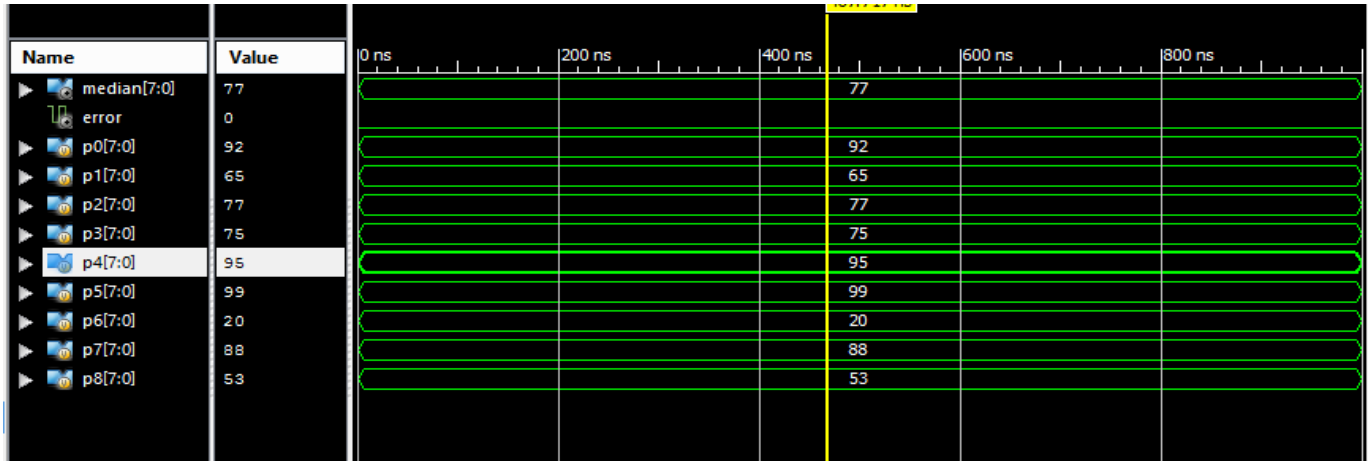


Fig 5. Simulation outcome of RTINR.

Fig 5. presents the simulation outcome of RTINR. Here, P0, P1, P2, P3, P4, P5, P6, P7, P8 are the inputs to RTINR and median is the output value.

| Device Utilization Summary (estimated values) | | | [-] |
|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** |
| Number of Slice LUTs | 437 | 303600 | 0% |
| Number of fully used LUT-FF pairs | 0 | 437 | 0% |
| Number of bonded IOBs | 81 | 700 | 11% |

Fig 6 Design summary

```
LUT6:I0->O            1    0.043    0.350   c14/a[7]_b[7]_LessThan_1_o22
LUT6:I5->O           14    0.043    0.422   c14/a[7]_b[7]_LessThan_1_o24
LUT4:I3->O            3    0.043    0.507   c14/a[7]_b[7]_LessThan_1_o25
LUT6:I3->O            1    0.043    0.613   H3/a[7]_b[7]_LessThan_1_o3 (H3
LUT6:I0->O            1    0.043    0.405   H3/a[7]_b[7]_LessThan_1_o4 (H3
LUT3:I1->O            2    0.043    0.410   H3/a[7]_b[7]_LessThan_1_o1_SW2
LUT5:I3->O            1    0.043    0.000   H3/a[7]_b[7]_LessThan_1_o1_G
MUXF7:I1->O           1    0.178    0.405   H3/a[7]_b[7]_LessThan_1_o1 (H3
LUT5:I3->O            6    0.043    0.631   H3/a[7]_b[7]_LessThan_1_o21 (H
LUT5:I0->O            1    0.043    0.613   H3/Mmux_13 (h31<2>)
LUT6:I0->O            1    0.043    0.405   median[7]_h31[7]_LessThan_1_o1
LUT5:I3->O            1    0.043    0.613   median[7]_h31[7]_LessThan_1_o1
LUT6:I0->O            1    0.043    0.339   error5 (error_OBUF)
OBUF:I->O                  0.000           error_OBUF (error)
----------------------------------------
Total                     20.375ns (1.726ns logic, 18.649ns route)
                                   (8.5% logic, 91.5% route)
```

Fig 7. Time summary

Fig 7 shows the design (area) summary of proposed method. Here, the proposed method utilizes the low area in terms of slice LUTs i.e., 437 out of available 303600. Fig 7 shows the time summary of proposed method. Here, the proposed method consumed total 20.375ns of time delay, where 1.726ns is logical delay, and 18.649ns is route delay. Fig 5.4 shows the power consumption report of proposed DCM-RTPG-BFD. Here, the proposed DCM-RTPG-BFD consumed power as 32.83 milli watts.

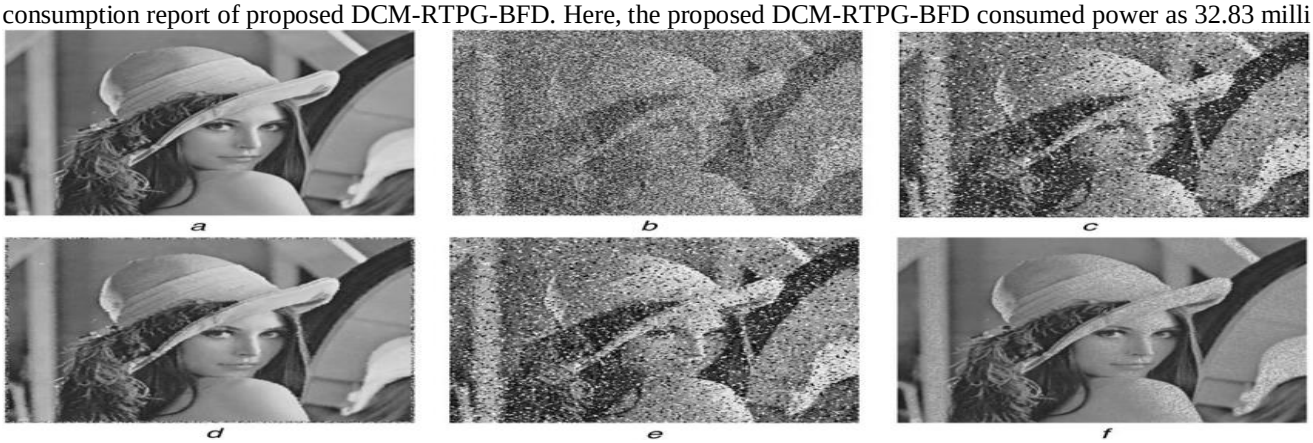


Fig 8. Visuval performmace of RTINR. (a) original image, (b) noisy image, (c) SMF [18], (d) DMF [21], (e) AMF [25], (d) proposed RTINR.

Fig 8 shows the filtering performance of various methods like SMF [18], DMF [21], AMF [25], and proposed RTINR. Here, SMF [18] and AMF [25] methods resulted outcome still contains the highher noises, DMF [21] method outcome contains the low

level noises. But, the proposed RTINR method resulted outcome is looks similar to the original image. Table 1 compares the performance evaluation of proposed RTINR method. Here, the proposed RTINR resulted in superior (reduced) hardware performance in terms of LUTs, time-delay, and power consumption as compared to conventional approaches such as SMF [18], DMF [21], and AMF [25]. Further, the proposed RTINR resulted in improved subjective performance in terms of peak signal to noise ratio (PSNR), stctural similarity index metric (SSIM) as compared to conventional approaches such as SMF [18], DMF [21], and AMF [25]. Further, the graphical representation of performance comparison is presented.

Table 1. Performance evaluation.

| **Metric** | **SMF [18]** | **DMF [21]** | **AMF [25]** | **Proposed RTINR** |
|---|---|---|---|---|
| LUTs | 767 | 655 | 542 | 437 |
| Time delay (ns) | 51.927 | 43.837 | 32.735 | 20.37 |
| Power consumption (mw) | 82.61 | 73.41 | 58.26 | 32.83 |
| PSNR (dB) | 37.34 | 42.45 | 48.38 | 54.53 |
| SSIM | 0.827 | 0.893 | 0.927 | 0.992 |

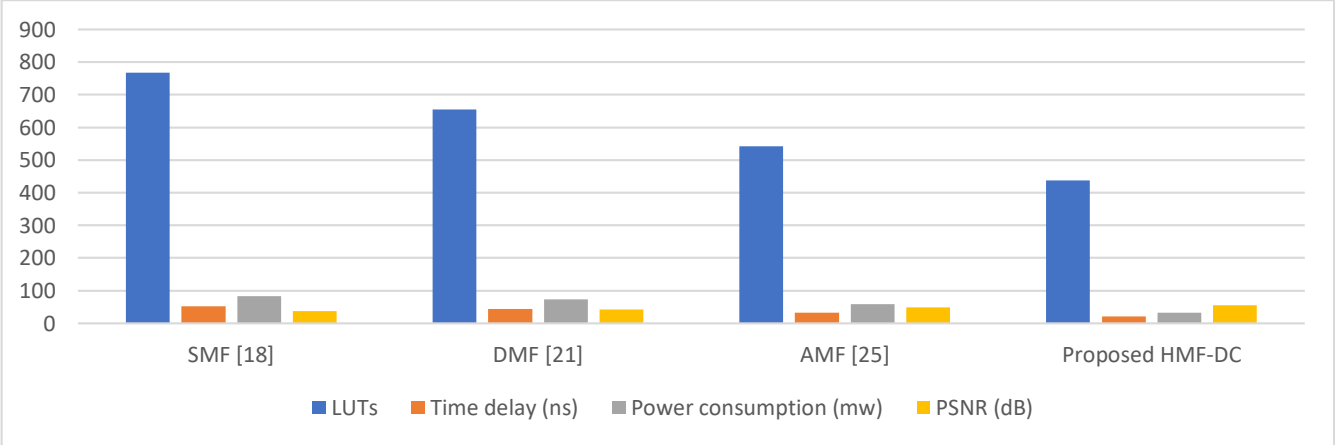


Fig 9. Graphical representation of performance.

## V CONCLUSION

The development of a Hybrid Median Filter by making use of Decision-based multiplexer logic is the primary emphasis of this study. In the beginning, a multiplexer selection logic-based decision-based multiplexer is used in order to determine which of two numbers have high and low values. After then, the decision-based multiplexer is carried out a number of times for the nine different possible combinations of pixels, which determines the median value for all nine of those values. The subjective and objective evaluations both reveal that the suggested RTINR resulted in greater performance when compared to the state-of-the-art techniques in terms of decreased noise, latency, and power consumption. Hardware metrics such as LUTs were also reduced and software metrics such as PSNR, SSIM are improved using the proposed RTINR approach. Further, this work can be extended with the hybrid adaptive filters for improved PSNR performance.

## REFERENCES

[1] Goel, Anish, M. Omair Ahmad, and M. N. S. Swamy. "Design of a 2D median filter with a high throughput FPGA implementation." 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2019.
[2] Ngo, Dat, Gi-Dong Lee, and Bongsoon Kang. "A 4K-Capable FPGA Implementation of Single Image Haze Removal Using Hazy Particle Maps." Applied Sciences 9.17 (2019):
[3] Rahnama, O., Cavallari, T., Golodetz, S., Tonioni, A., Joy, T., Di Stefano, L., ... & Torr, P.H. (2019). Real-time highly accurate dense depth on a power budget using an fpga-cpu hybrid soc. IEEE Transactions on Circuits and Systems II: Express Briefs, 66(5), 773-777.
[4] Sambamurthy, N., & Kamaraju, M. (2020). Power optimised hybrid sorting-based median filtering. International Journal of Digital Signals and Smart Systems, 4(1-3), 80-86.

[5] Arnal, J., & Súcar, L. (2019). Hybrid filter based on fuzzy techniques for mixed noise reduction in color images. Applied Sciences, 10(1), 243.
[6] Ahmad, Rais, and Som Pal Gangwar. "Rain Streaks Elimination Using Hybrid Median Filter and Contrast Stretching." Advances in VLSI, Communication, and Signal Processing.Springer, Singapore, 2021. 157-178.
[7] Rini, C., Perumal, B., & Rajasekaran, M. P. (2019). Eradication of Rician Noise in Orthopedic Knee MR Images Using Local Mean-Based Hybrid Median Filter.In Proceedings of the 2nd International Conference on Data Engineering and Communication Technology (pp. 693-702). Springer, Singapore.
[8] Banerjee, S., Sinha Chaudhuri, S., Mehra, R., & Misra, A. (2021). A Comprehensive Review of Median Filter, Hybrid Median Filter and Their Proposed Variants. Advances in Smart Communication Technology and Information Processing, 393-406.
[9] Elsharif, R. I., Ibraheem, B. A., Mustafa, Z. A., & Abass, S. K. (2018). Wavelet decomposition–based speckle reduction method for ultrasound images by using speckle-reducing anisotropic diffusion and hybrid median. Journal of Clinical Engineering, 43(4),163-170.
[10] Kannan, L. M., & Deepa, D. (2021). Low power very large-scale integration (VLSI) design of finite impulse response (FIR) filter for biomedical imaging application. DYNA-Ingeniería.