IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 41, No. 5, October, 2024, pp. 2403-2414
Journal homepage: http://iieta.org/journals/ts

# Efficient Parallel Median Filter for Image Denoising: Implementation and Performance Evaluation

Mahendrakan Kantharimuthu[1*], Prema Selvaraj[2], Harish Sankar[3], Gokulavasan Brindavanam[4]

[1] Department of ECE, Hindusthan Institute of Technology, Coimbatore 641032, India
[2] Department of ECE, Sri Krishna College of Technology, Coimbatore 641042, India
[3] Department of ECE, Rathinam Technical Campus, Coimbatore 641021, India
[4] Department of ECE, Sri Eshwar College of Engineering, Coimbatore 641202, India

Corresponding Author Email: dr.mahendrakan.k@hit.edu.in

https://doi.org/10.18280/ts.410515

**Received:** 27 November 2023
**Revised:** 20 February 2024
**Accepted:** 23 April 2024
**Available online:** 31 October 2024

***Keywords:***
*sorting network, parallel median filter, precision computations, data sorter, modified magnitude comparator, equality check, processing pixels*

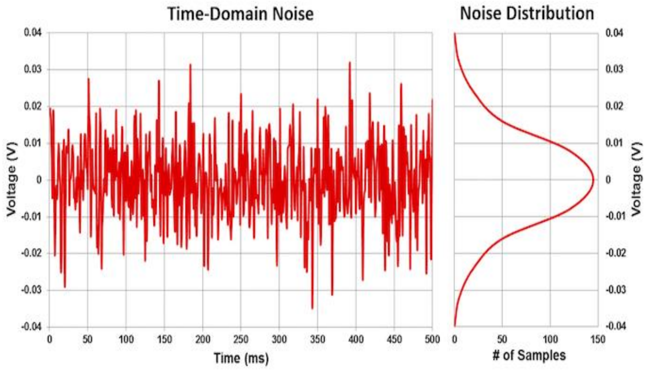
## ABSTRACT

The sorting network forms the foundation of the suggested parallel median filters and the cutting-edge filter produced noise-free images. To enhance such filters noise-elimination abilities, a particular comparator is created. Signal leaps can be preserved while noise is reduced with the help of parallel median filtering. The noise elimination determines how much noise is reduced. The filter does a better job of minimizing noise the heavier the distribution tail. Median filtering preserves edge signal, which is a crucial aspect of images, more effectively than average filtering. New median filters have a consistent, modular architecture. Limited precision computations are allowed in applications that process audio and images. The approximate computing can be implemented in the digital system with sufficient precision. This paper proposes a novel technique for the low-cost area, power and speed-efficient manufacturing of 2-bit magnitude comparators. The new technology created larger comparators with tunable error characteristics. Further, parallel median filter is designed with additional 2 ternary data sorter for high speed application which processes the data in parallel. From Simulation results, the proposed filter achieves more power, area and speed. The filters output value is essentially equivalent to the particular one when it comes to filtering precision and circuit features. When compared to serial median filters, parallel median filters dynamic power consumption is 36.37% higher and also total estimated power consumption of parallel median filter is 30.80% more compare to with serial Median Filter. In logic distribution, 5% of number of occupied slices is reduced in the parallel median filter. Parallel design uses 26.76% fewer total equivalent gates than serial design. Simulations show that inexact filter implementations can save up to 30% and 26% of energy and space, respectively, and can accelerate operations by 15% when compared to standard accurate ones.

## 1. INTRODUCTION

Image denoising is a pre preparatory step in image processing. Various noises including thermal noise, impulse noise and salt and pepper noise will affect digital images. When sending photos via any type of electronic communication noise is a major worry. In electronic transmission, voltage instability can lead to frequent impulse noise issues. When picture acquisition, incorrect hardware memory locations, malfunctioning camera sensor pixels, or transmission over a noisy channel, Noise occur in the Image. Image [1] can be affected by Thermal noise. Even in low noise density the image property can change due to appearance of thermal noise in image. A more advanced form of impulse noise, familiarized by the name salt-and-pepper noise, occurs when the corrupted pixels are constrained to take on only the maximum and minimum intensity values within the dynamic range. It seems to be grayscale images, with a dispersion of pixels displaying maximum and minimum intensity values, resembling a distribution of white and black dots. Impulse noise also called salt-and-pepper noise is a common phenomenon that can occasionally be seen in digital photos. This noise can result from abrupt and unforeseen disturbances in the image, manifesting as isolated black and white pixels dispersed across the image. Other names for thermal noise include Johnson noise, Nyquist noise and Johnson-Nyquist noise. In an electrical conductor, independent of the applied voltage, if the charge carriers vibrate proportional to temperature, it is known as thermal noise. Figure 1 shows the graphic representations of salt and pepper noise and thermal noise. The secret to a median filter is to swap the grey level of each pixel instead of using an average operation by taking the median of the grey levels of every neighboring pixel surrounding every single pixel. For dipping salt and pepper noise, the median filter (MF) is regarded as the top choice [2]. Parallel Median filtering reduces blurring effects from frame capturing speed overlaps, which is one of the key uses of median-finding in image and audio processing.

(a) Salt and pepper noise



(b) Thermal noise

**Figure 1.** Graphic representations

Substituting the median of the adjacent entries for each signal entry, entry by entry is the straightforward method of implementing the median filter. The window that shifts entries one after another during the signal is referred to as the neighbors pattern. The window can be as simple as the first few entries before and after for one-dimensional signals, but it must contain all entries inside of an elliptical region or radius for two-dimensional (or higher-dimensional) data. For any length greater than two, a median filter cannot be divided into a product of two vectors, making it non-separable. This is because the median function that makes up the filter lacks an equivalent sum-product form. Whether or not it has a square mask is irrelevant. An additional level of complexity in image processing is provided by the employment of non-separable filters, which enable better denoising outcomes while maintaining crucial edges and image elements. These filters have two limitations: For low signal-to-noise ratios, they often produce misleading noise artifacts and disrupt image edges, failing to effectively reduce medium-tailed (Gaussian) noise distributions.

One type of smoothing method is the median filter, commonly known as the linear Gaussian filter. All smoothing techniques have an adverse effect on edges, however smooth patches or specific features of a signal can be efficiently denoised. However, it is difficult to both sustain the edges of image as well as reducing the noise in it simultaneously. For example, edges play a vital role in how images appear visually. The median filter performs better in terms of noise reduction without losing edges for minimum to sensible amount of Gaussian noise for a given window size than Gaussian blur. The Gaussian noise performs superior to Gaussian blur at high noise levels but it excels at reducing speckle and salt-and-pepper noise. Therefore, median filtering is a common method in processing digital images. Image noise is often referred to as electrical noise which is the unpredictable change in brightness or color details in photos. It can be produced using the image sensor and circuitry of a digital camera or a scanner. Additional types of image noise include film grain and the unavoidable shot noise associated with an ideal photon detector. The unwanted side effect of image capture is image noise, which hides the required details of the image. In digital photography, noise is a prevalent problem that can seriously degrade an image's quality. Reduced dynamic range, colour distortion, and detail loss are some of the impacts. The quality of the lens, the distance from the subject, and camera settings like ISO, shutter speed, and aperture can all have an impact on how clear an image.

Although they are not the same, resolution and clarity both affect an image's overall quality. Salt-and-pepper noise and spike noise are two other names for fat-tail dispersed or "impulsive" noise. Reducing impulse noise is a major issue in image processing. In a noisy image there will be bright spots in dark areas and vice versa. This kind of noise can be caused by bit errors that occur during transmission. It can be largely removed with dark frame subtraction combined median and mean filtering dark/bright interpolation and median filtering. An LCD monitor with dead pixels can create this kind of display though not at random.

Median filters are particularly effective at removing periodic patterns and random noise when the noise amplitude probability density has heavy tails. The filtered image can be produced by putting the values of the output image in the middle of the input window. The location with the highest likelihood estimate is represented by the median when working with a Laplace operator of noise distribution. In reasonably uniform areas, the median filter is especially useful for estimating the grey-level value when long-tailed noise is present. When an edge crosses, the output undergoes a sudden transition between values as one side occupies a larger portion of the filtering window than the other. The median filter is seen as beneficial because, in contrast to the averaging filter, which blurs the image's edges in order to eliminate noise, the median filter eliminates noise alone, leaving the edges undisturbed. The only and best filter for eliminating pepper and salt sounds is the median filter.

## 2. LITERATURE SURVEY

The output of the linear filter changes linearly when the input changes. Images with impulse noise are often subjected to a non-linear digital filter. One popular pre-processing method for enhancing the outcomes of additional image processing is noise reduction of this kind. In order for the median filter to work each value in the image is iteratively replaced with the median value of the pixels next to it. The arrangement of adjacent pixels that moves across the image, pixel by pixel, is known as the filtering window. A non-linear filtering method called median filtering can preserve sharp features in an image. One of the disadvantages of a median filter is that it can be difficult to analyze its effects analytically.

Erkan et al. [2] introduced Applied Median Filter which is used to eliminate salt and pepper noise at all concentrations. The information about the colour or brightness of the image can be affected by noise. Commonly, image sensor of a digital camera or scanner can produce noise in images. Film grain can also cause noise and shot noise is a necessary component of any good photon detector. The intended information is hidden by noise in the image an unsought byproduct of image capture. The peak signal-to-noise ratio and structural similarity of the median filter types are compared in this study using the applied median filter. Through complex processing a tiny amount of information can be extracted from nearly noise-free optical and radio astronomical images as well as from barely perceptible specks on a digital snapshot captured in optimal lighting [3]. In a technical context, such a level of noise would be problematic because it would make it impossible to distinguish or identify the subject. MF is suited for low-density noise images. All of the image's pixels, even the noisy ones, are processed. This is the median filter's primary drawback.

Vasicek and Mrazek [4] proposed a unique approach that uses genetic programming (GP)-based genetic augmentation to trade off non-functional and functional features of current software. The study looks into ways to enhance non-functional parameters, such as median functions implemented by comparator systems' execution time, code size, or power consumption. In general, because optimal implementations are available, it is not viable to change non-functional median function parameters without occasionally accepting results that are incorrect. To address this issue, the System [4] proposed a solution that effectively balances accuracy, power consumption and execution time. Random test vectors assess the fitness of GP individuals. The author proposed novel quality metrics based on positional error from the permutation approach to accurately assess and mitigate median value bias. When minimizing the errors in image, the suggested approach makes it possible to find solutions that significantly outperform the correct median function in terms of power consumption, execution time and size. The solutions that are disclosed have non-functional properties that are confirmed by means of authentic measurements on real microcontrollers and calculated with various data settings.

Monajati et al. [5] presented the demonstration of a novel approach to design 2-bit approximate magnitude comparator with low requirements for speed area and power. This method produced the comparators with programmable error features. Compared with precise comparators, our approximation comparators [5] can save up to 7-46% and 10-50% of power and space, respectively. This study characterizes the architectures of our comparators and analyzes their error profiles. The system generates multiple approximate median filters using these comparators to suppress salt-and-pepper noise. The simulation results show that noise deterioration is negligible because these filters output quality is remarkably close to the real ones. The approximation filters operate 18% more quickly than the precise ones while using up to 30% less power and space.

HosseinKhani et al. [6] designed one median filter for solving the problem of SAP. The quality of all photographs, including medical photos, is significantly lowered by noise. A typical impulsive noise called SAP is brought on by faulty sensor components or mistakes in the transmission of images [7]. Because many of the pixels in medical photos have the same brightness in relation to impulse noise and because the background is frequently black with white foreground it can be challenging to distinguish between noisy and normal pixels. Developing a method to accurately eliminate this kind of interference like SAP is essential [8]. Process complexity is crucial for hardware implementation but accuracy is also vital [9]. The de-noising technique suggested in this study distinguishes between noisy and non-noisy pixels.

Brioua et al. [10] provided an effective wavelet thresholding and wiener filtering association for an R-Peaks recovery and a median filter smoother which are included in the ECG denoising. In this paper, we introduce an advanced denoising technique designed to mitigate the effects of additive white Gaussian noise, which uniformly affects signal frequencies across the entire spectrum [11]. Our method provides an innovative combination of Wiener filtering methods with wavelet transform. The proposed approach involves hard thresholding of detail coefficients, single-level DWT, and wavelet-domain Wiener filtering of approximation coefficients [10]. The initial denoised signal is obtained by applying the inverse Discrete Wavelet Transform. A median filter is applied to enhance the precision of the signal restoration process even more.

Li et al. [12] introduced a median filter algorithm to eliminate high- concentration salt and pepper noise by filtering signal points surrounding noise pixels after noise detection. The wavelet denoising technique by Ismael and Baykara [13] decomposes images into four frequency sub-bands for processing. Reliable performance depends on obtaining the original image content. With this technique undesirable noise in an image with noise is removed while maintaining the images key elements.

## 3. EXSITING WORK

### 3.1 Fundamentals

Ternary Data Sorter (TDS) consists of two-bit magnitude comparator (TBC) and equality checker (EC). It searches the first and last thirds of the domain first and then it searches the remaining two thirds confirming that the minimum or maximum cannot exist in either of those two thirds. Ternary search is an example of a divide and conquer algorithm. Figure 2 displays the TDS internal block diagram. From this figure, a two-bit comparator unit is the TBC. The least significant bits are compared by the first top TBCs. Each EC block verifies that the bits that came before it was equal. The filter structure has 12 comparators [14]. The median filters physical characteristics are significantly impacted by TBC. While assessing approximate comparators, the Error Dispersion Plot (EDP), which more properly depicts the error features, it is a crucial qualitative metric. A magnitude comparator determines the relative size of two binary numbers. The succeeding outputs are indicated for inputs X and Y, as shown in Figure 3. Greater is shown by X > Y, less than is shown by X < Y and Equal is shown by X = Y. Each modes remaining outputs are all zero with any one output being one. In this section a few approaches to constructing an approximate comparator are presented and thoroughly examined. There should be fewer logic gates in the approximation MCs than in the precise one. Speed, complexity, size need, and power dissipation of a circuit are all affected by the logic implementation [13]. The objective is to reduce the gate requirements by MCs logic function so that its hardware can be made simpler by accepting small truth table errors [15].

The circuit shouldn't be subjected to any significant errors and the results should be satisfactory. The system suggests various 2-bit comparator function simplifications. We demonstrate how to expand these 2-bit MCs into larger ones that have customizable error characteristics. The system design makes advantage of the less than (L) and greater than

(G) approximation logic of the two-bit comparator's (TBC) output. This is achieved by modifying a few cells in the corresponding K-maps from 0 to 1 or 1 to 0 [16]. This lowers the number of gates required. Four different kinds of approximation terms are used for each of L and G as Ln and Gn. Figure 4 shows the exact and approximate K-maps for the outputs L and G. Table 1 lists the Ln and Gn minimums that have been added or removed from the precise ones. If input values are uniformly distributed, the likelihood of error is kept below 12.5% for all methods of approximation. The two-bit comparator (TBC) outputs are utilized in the system design for approximation logic circuits [17] with less than (L) and larger than (G) values.

This is accomplished by minimizing the gate count requirement by converting various cells in their equivalent K-map Boolean method from 1 to 0 or vice versa. For each of L and G, we define four distinct forms of approximations as Ln and Gn. The exact and approximative K-maps for L and G are displayed in Figure 5. Table 1 lists the Ln and Gn minimums that have been subtracted or added to the exact ones. If input values are equally distributed, the likelihood of error is maintained below 12.5% for all approximation techniques. A 2-bit comparator called TBC which is used to detect whether one integer is greater than or less than another. Each equality check (EC) block confirms that the bits that came before it are equal. This method simply applies the approximation to one or more TBC blocks, leaving the structure intact.

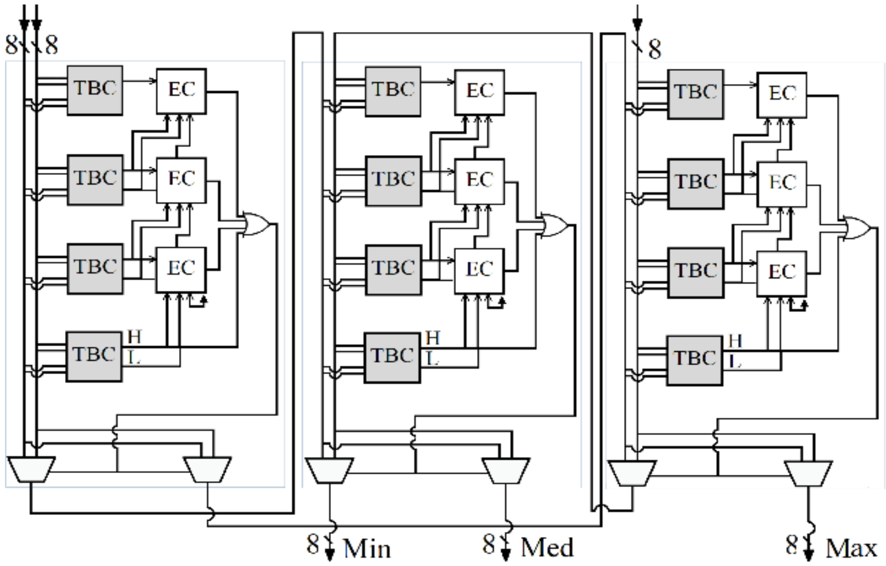


**Figure 2.** Schematic of ternary data sorter

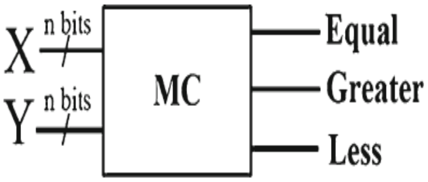


**Figure 3.** Magnitude comparator

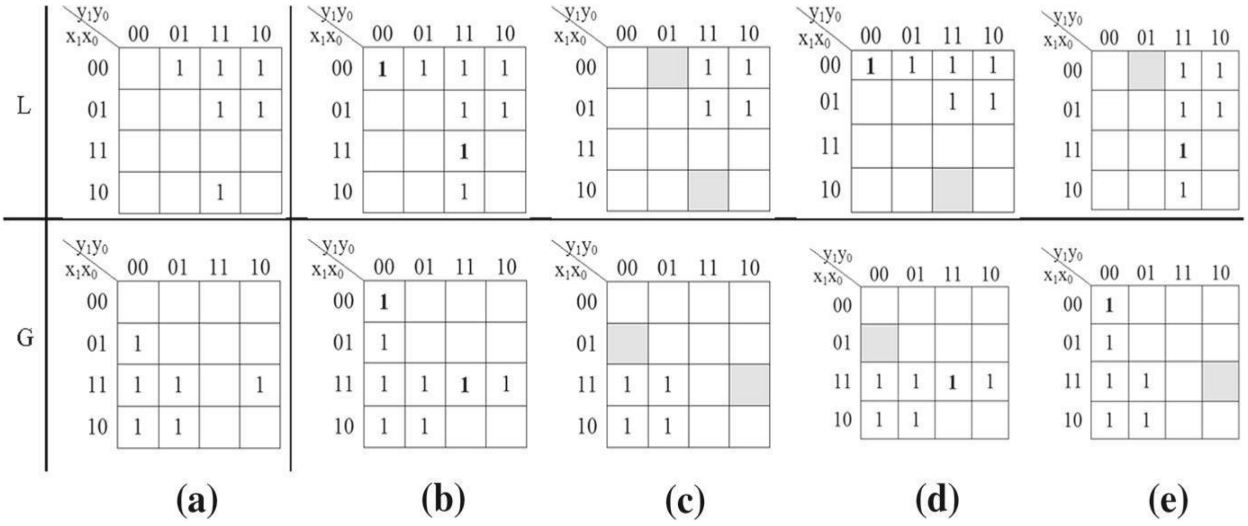


**Figure 4.** Karnaugh map for Ln and Gn with approximate values of a, b, c, d, and e. Bolded additions are highlighted, whereas deletions are coloured

**Table 1.** Logic utilization, logic distribution and number of 4-input look up tables (LUTs) of serial median filter

| Description | Utilized | Accessible | Utilization |
|---|---|---|---|
| | **Logic Utilization** | | |
| Flip-flop count with a Slice | 80 | 13824 | 1% |
| 4-input LUTs Count | 353 | 13824 | 2% |
| | **Logic Distribution** | | |
| Occupied Slices count | 216 | 6912 | 3% |
| Count of slices solely contain related logic | 216 | 216 | 100% |
| Count of slices with disconnected logic | 0 | 216 | 0% |
| **Total count of 4-input LUTs** | **386<br>(Number used as logic – 353<br>Number used as a route thru-33)** | **13824** | **2%** |
| Count of bonded IOBs | 33 | 510 | 6% |
| Count of IOB Flip Flops | 24 | | |
| No of GCLKs | 1 | 4 | 25% |
| Gate complexity of the design | 3187 | | |
| IOB JTAG overhead | 1632 | | |

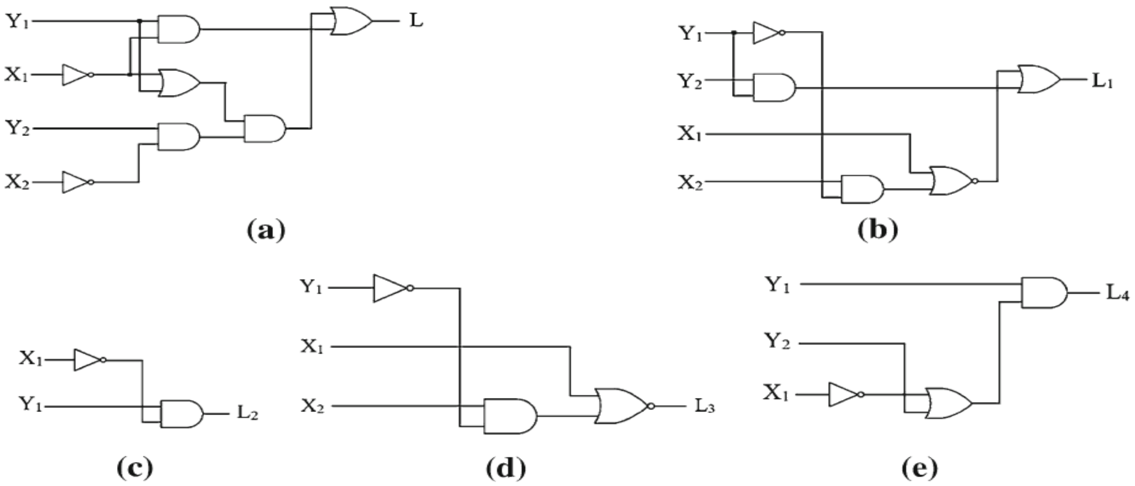


**Figure 5.** Diagrams of the logic for exact and approximate L. exact, L1, L2, L3, and L4

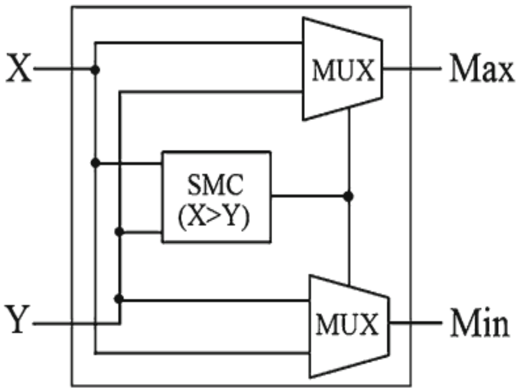


**Figure 6.** Modified magnitude comparator

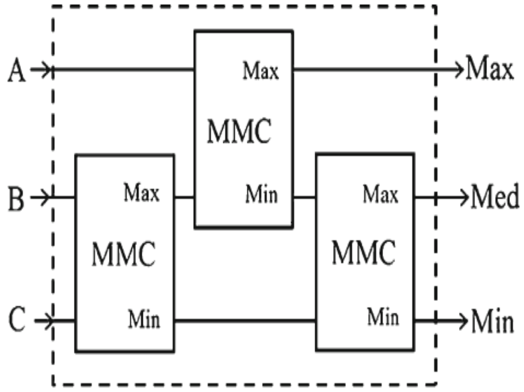


**Figure 7.** Ternary data sorter (TDS) with MMC

3.1.1 Modified magnitude comparator

Sort two numbers using the MMC block (Modified Magnitude Comparator). If the inputs are the same, the MMC's two outputs will equal zero. As a result, no equal mode circuit design is required. Figure 6 shows how the MMC works. It is determined whether or not X is greater than Y via the sub-magnitude comparator, or SMC.

Due to the value of the SMC output, the newly suggested comparator is processed to create a new median filter [18] that will swiftly and effectively remove salt and pepper noise from grey-level images. As the central element of the median filter the data sorter (DS), we optimize the newly proposed comparator here to function well with it. Multiplexers are used to route the larger and smaller inputs to the proper outputs. The TDS sorts the data according to magnitude after receiving three integer inputs. Figure 7 shows the Ternary data sorter (TDS) with MMC. The variables for inputs are A, B, and C. The Max, Min, and Median values can be used to predict the output. The MMC is the TDS's main beating heart. The TDS consists of three MMC, each having two inputs and two outputs.

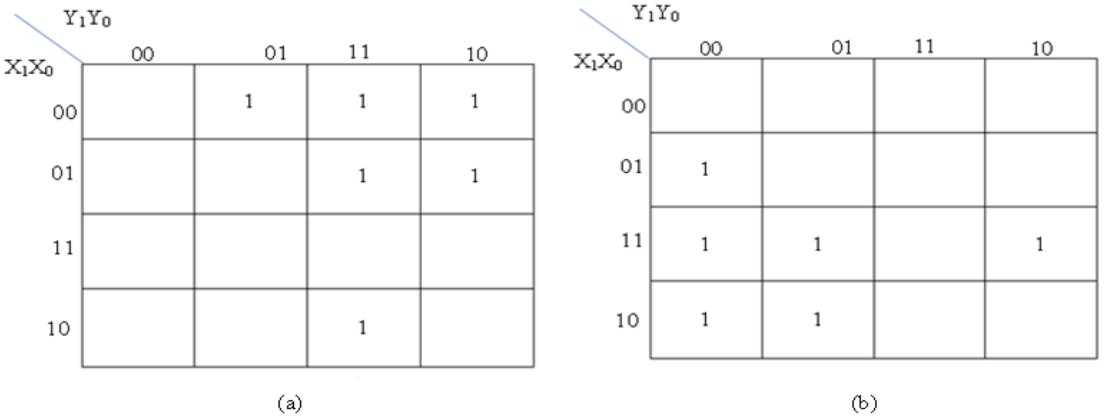
### 3.2 Inexact arithmetic for removing peppers

A less complicated filter results from a simpler TBC. A simpler logic function requires less resource to build, which reduces switching activity and power consumption. By adding a small amount of error to its truth table the method which is similar to simplifies the logic implementation of TBC. The key distinction between this method and others is the deliberate approximation used to attain accuracy when comparing an 8-bit number with the common salt and pepper noise. Although suggests a number of straightforward magnitude comparator units, APMFs are unable to remove noise as effectively as an accurate filter.

Both APMF and our imperfect filter are equally good at eliminating salt-and-pepper noise. We accomplish low-complexity, simplified arithmetic units by adopting imprecision, instead of typical low-power design strategies, which enables us to save more power [19]. The viewer should find the IMF image quality acceptable even though it isn't as excellent as the precise filter.

As previously mentioned, the median filter hardware is simplified by simplifying the TBC hardware. To reduce their impact on noise removal in the median filter, deliberately errors were introduced into a subset of TBCs Boolean map cells. The pictorial method of the precise TBC is shown in Figure 8. In the illustration, peppers are represented by the black dots with no intensity. Consequently, the focus was on the K-map cells that represent two-bit zeros. They are referred to as crucial cells engaged with peppers, or CCPs, and they ought to be error-free. In Figure 9, they are displayed as coloured cells.

Errors should either be avoided in critical cells or, if they occur, they should be corrected. Errors are allowed in the critical cells of L only when the corresponding cell in H contains zero, as H is essential for determining the circuit's output. Remember that none of the H's crucial cells can contain the mistake. The inaccuracies in L in this instance have no impact on the TBC. We have L = H = 0 for two MSBs of inputs as a result of the explanations. The comparator recognizes them as equal as a result. As a result, the comparison was made with the subsequent less noteworthy couples. In accordance with Figure 9 (b), the comparator successfully finds X lower than Y if these pairings of Y contain all values other than 1. Conversely, in the event that Y = 01 01 01 01, X is assigned by the comparator. Consequently, we get High = 0, Maximum = Y, and Minimum = X. Consequently, the comparator operates as intended, and peppers are effectively eliminated by the filter that was developed using this incorrect TBC. Because it works well for pepper, we call this filter IMFP.



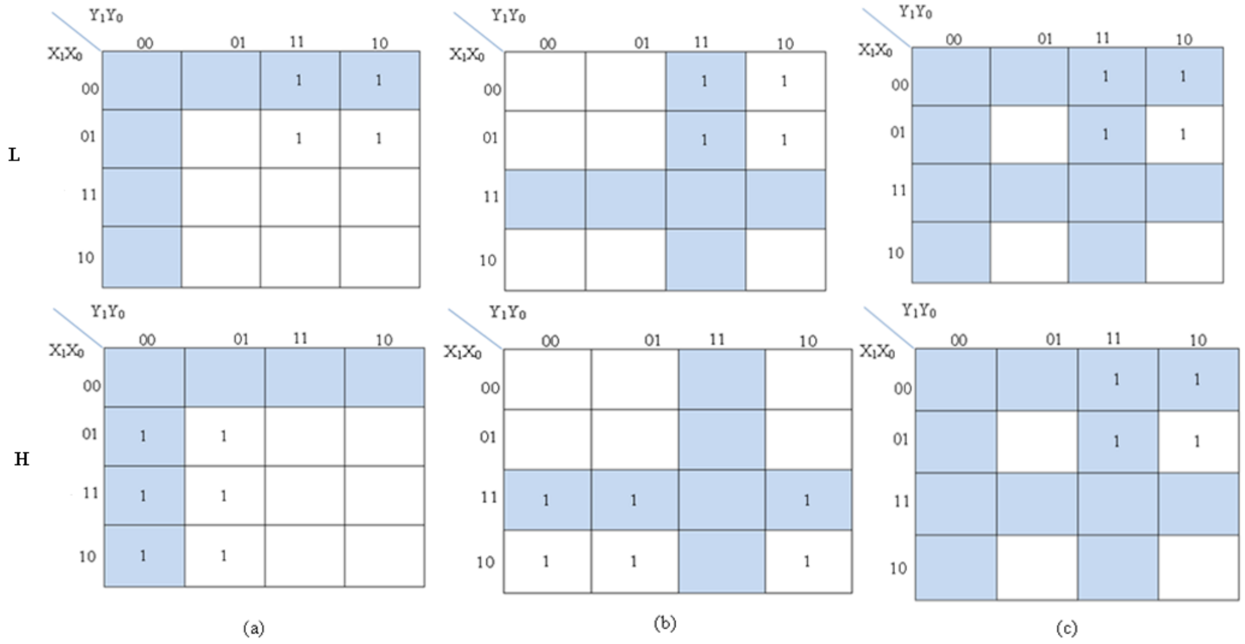
(a)

| $Y_1Y_0$<br>$X_1X_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 |  | 1 | 1 | 1 |
| 01 |  |  | 1 | 1 |
| 11 |  |  |  |  |
| 10 |  |  | 1 |  |

(b)

| $Y_1Y_0$<br>$X_1X_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 |  |  |  |  |
| 01 | 1 |  |  |  |
| 11 | 1 | 1 |  | 1 |
| 10 | 1 | 1 |  |  |

**Figure 8.** Karnaugh-maps of 2-bit comparator for: a) L and b) H



L

(a)

| $Y_1Y_0$<br>$X_1X_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | ▨ | ▨ | ▨ 1 | ▨ 1 |
| 01 | ▨ |  | 1 | 1 |
| 11 | ▨ |  |  |  |
| 10 | ▨ |  |  |  |

(b)

| $Y_1Y_0$<br>$X_1X_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 |  |  | ▨ 1 | 1 |
| 01 |  |  | ▨ 1 | 1 |
| 11 | ▨ | ▨ | ▨ | ▨ |
| 10 |  |  | ▨ |  |

(c)

| $Y_1Y_0$<br>$X_1X_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | ▨ | ▨ | ▨ 1 | ▨ 1 |
| 01 | ▨ |  | ▨ 1 | 1 |
| 11 | ▨ | ▨ | ▨ | ▨ |
| 10 | ▨ |  | ▨ |  |

H

(a)

| $Y_1Y_0$<br>$X_1X_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | ▨ | ▨ | ▨ | ▨ |
| 01 | ▨ 1 | 1 |  |  |
| 11 | ▨ 1 | 1 |  |  |
| 10 | ▨ 1 | 1 |  |  |

(b)

| $Y_1Y_0$<br>$X_1X_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 |  |  | ▨ |  |
| 01 |  |  | ▨ |  |
| 11 | ▨ 1 | ▨ 1 | ▨ | ▨ 1 |
| 10 | 1 | 1 | ▨ | 1 |

(c)

| $Y_1Y_0$<br>$X_1X_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | ▨ | ▨ | ▨ 1 | ▨ 1 |
| 01 | ▨ |  | ▨ 1 | 1 |
| 11 | ▨ | ▨ | ▨ | ▨ |
| 10 | ▨ |  | ▨ |  |

**Figure 9.** Imprecise Karnaugh-maps of two-bit comparator for removing: a) peppers, b) salts, c) salts and peppers

### 3.2.1 Inaccurate calculus for eliminating salts

Salts are the 255-intensity white dots seen in the picture. Since 255 in binary equals 11111111 we need to focus on the cells that represent the number 11 (x1x0 = 11 y1y0 = 11 etc.). They are known as salt-related hazardous cells or CCSs and they must be flawless. In Figure 9 (b), they are depicted as coloured cells.

This section continues until the cell number 11 is correctly related to other two-bit numbers. Inaccurate L and H must meet the requirements outlined in the preceding section. H is recommended, as Figure 9 (b) illustrates. That figure shows that H's CCSs are error-free. Furthermore, there is zero in the cell that H's label 1011, which represents the problematic CCs of L. Consequently, the precise one and other salts are eliminated by the inexact median filter. We refer to the filter as IMFS because of how it specifically works on salts. to create an imperfect median filter that effectively filters out salts and peppers. We recommend L and H as indicated in Figure 9 (c); the CCs of H are correct. Additionally, there is zero in two cells of H with the labels 0001 and 1011 that correspond to the problematic CCs of L. As a result, neither salts nor peppers cause the filter to malfunction. However, some portions of the image may have mistaken as a result of the inaccurate computations. It should be noted that the suggested imprecise arithmetic can be used with one or multiple TBCs. It is evident that decreasing circuit area and power consumption by increasing the number of erroneous blocks comes at the expense of image quality [20].

## 4. PROPOSED METHODOLOGY

A synchronous parallel implementation of a two-dimensional median filtering algorithm is shown in Figure 10. This implementation demonstrates how well a novel data partitioning strategy distributes the workload among the numerous processors in a multiprocessor system. For algorithms that depend on data features that, like in normal photos, could differ significantly from area to region, the suggested data partitioning technique has decreased the idling time of the processors at synchronization points.

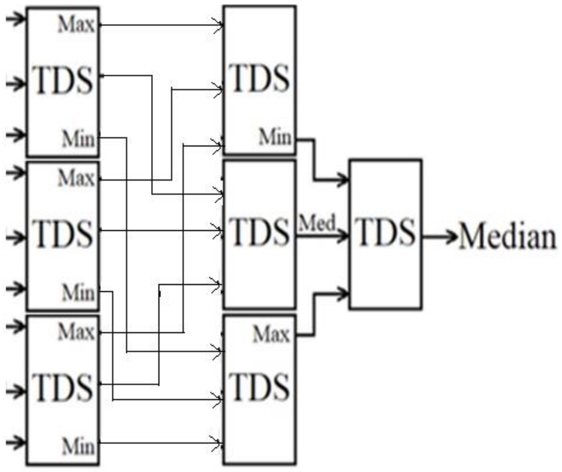


**Figure 10.** Parallel median filter

A fresh method for creating compact, error-resistant arithmetic units for portable applications is approximate computing. In this work, a novel median filter architecture aimed at concurrent digital image processing has been suggested. The future parallel median filter (PMF) replaces defective processing pixels (PP) by the pixel median in the 3×3 processing window using pre-sorter and post-merge units. The suggested PMF uses approximate comparison and swap in TDS blocks, which can trade off area at the penalty of accuracy. The parallel architecture does not use any intermediate registers which reduces the overall power consumption. TDS constructed with modified magnitude comparator.

In order to function, the parallel median filter takes into account a sliding window—also referred to as a kernel or mask—that changes every single pixel throughout the image. The window's pixel intensities are sorted at each place, and the central pixel's new intensity is determined by taking the median value. The median value is an effective choice for noise reduction because it is resistant to extreme values and represents the central value in a sorted list of pixel intensities. The median filter's ability to maintain edges and minute features gives it a significant advantage over the mean filter. The parallel median filter effectively reduces noise while maintaining the sharpness of image borders, while the mean filter tends to muddy edges and outlines when noise is present. Because of this property, the median filter is very useful in situations when maintaining significant structural information is essential. Erratically seeming bright and dark pixels are called "salt-and-pepper noise". They greatly reduce image quality. When dealing with this kind of noise, the median filter excels because it replaces noisy pixels with the median value, efficiently eliminating the highest and lowest noise levels without affecting the image's overall appearance.

In the parallel median filtering procedure, selecting the window size—also referred to as the kernel size—is crucial. Although a smaller window will help to reduce fine, localised noise, there's a chance that crucial details will be lost. However, a bigger window size may result in visual blurring even though it can better reduce broader noise patterns. Achieving the optimal balance between noise reduction and detail preservation is crucial, depending on the specific application and the characteristics of the noise. The robustness of the median filter to outliers is one of its key advantages. In situations where irregular extreme noise values are present the median filter eliminates these outliers from its calculation of median value. Therefore, it can handle noise that would otherwise cause other filtering methods to perform poorly. In many different domains such as computer vision applications surveillance remote sensing and medical imaging the median filter is employed. For instance, it is applied to improve CT and MRI scans in order to guarantee precise diagnosis and analysis. By reducing noise in satellite imagery, the median filter enhances the quality of data for environmental monitoring and mapping.

In summary reducing noise in digital photos without sacrificing important edges and features can be accomplished simply but effectively with the median filter. They find widespread application in image processing due to its versatility in handling noise and resilience to outliers. However, it’s important to weigh the trade-off between noise reduction and detail retention when determining the right window size for a given use case. Well examine increasingly complex variants as we delve deeper into this fascinating subject and figure out how to maximize median filterings effectiveness in various contexts.

Through pixel-by-pixel movement a sliding window also known as a kernel function as a median filter. The window frame is a key factor that affects how much filtering occurs.

By filtering out more nearby pixels a larger window can reduce noise more effectively but there is a chance that crucial information will be lost. The pixel values need to be sorted in order to find the median value which is exactly in the middle of the sorted list. Pixel intensities within the moving window are collected and sorted at each position. The median value is ascertained subsequent to the sorting of the windows pixel values. The middle value in the ordered list represents the median for windows containing an odd number of pixels. The median of an even-sized window is obtained by averaging the two center values. The median operation ignores outliers and concentrates on the center trend of the pixel intensities making it resistant to high noise values. The original pixel intensity is replaced with the calculated median pixel value of the sliding windows central pixel. Using this technique the sliding window covers the entire image resulting in a newly filtered less noisy image. In contrast to popular filtering methods such as the mean filter the median filter exhibits exceptional ability to retain image edges and minor details. This is done to prevent the extreme noise values—which the median procedure successfully removes—from blurring the images edges. Because of its selective noise reduction which maintains crucial features like the borders and curves of the undistorted image the median filter is perfect in situations where maintaining structural information is essential.

A prevalent challenge in image processing is salt-and-pepper noise or irregularly bright and dark pixels. In noise like this the median filter works incredibly well. A smooth transition between the extreme extremes is produced when the sliding window passes through areas with salt-and-pepper noise because the median value will naturally fall between the bright and dark pixels. This produces an image that is sharper and more coherent by effectively eliminating the noise that salt and pepper cause. Although median filtering typically uses a fixed window size, adaptive median filters allow the window size to be dynamically altered based on the specific local characteristics of the picture. Adaptive techniques are useful when there are structures with varying sizes or when noise characteristics fluctuate throughout the image. We can effectively use the median filter to improve image quality, boost feature visibility, and enable more accurate image analysis in a variety of image processing and computer vision domains by grasping its basic concepts.

## 5. RESULTS AND DISCUSSION

An effective trade-off between accuracy and the physical characteristics of a median filter is presented, when developing imprecise two-bit magnitude comparators. The amount of noise that is retained in the images by these filters varies. Further, parallel median filter is designed with additional 2 ternary data sorter for high speed application which processes the data in parallel. Simulation data shows that the output image quality, though slightly reduced compared to the exact filter, is practically indistinguishable from it to human observers. The construction and error characteristics of median filters and imprecise comparators were discussed. The imperfect median filters have far superior physical characteristics than the precise one. According to simulations, inexact filter implementations are essentially inexpensive, can reduce energy and space usage by up to 26% and 30%, respectively, and can speed up processes by 15% compared to traditional accurate ones. The serial median filter simulation result, area report and power report are represented in Figure 11, Figure 12 and Figure 13.

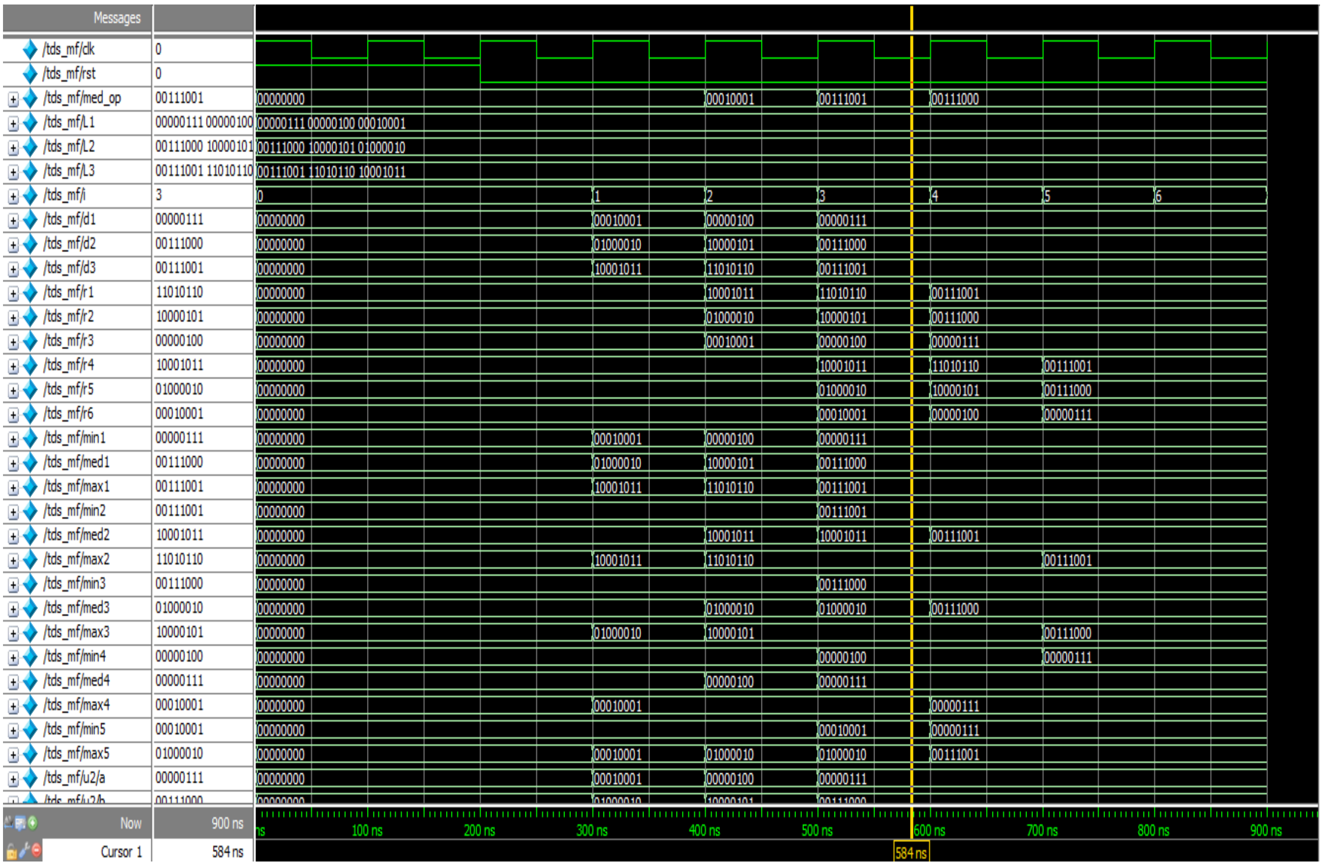
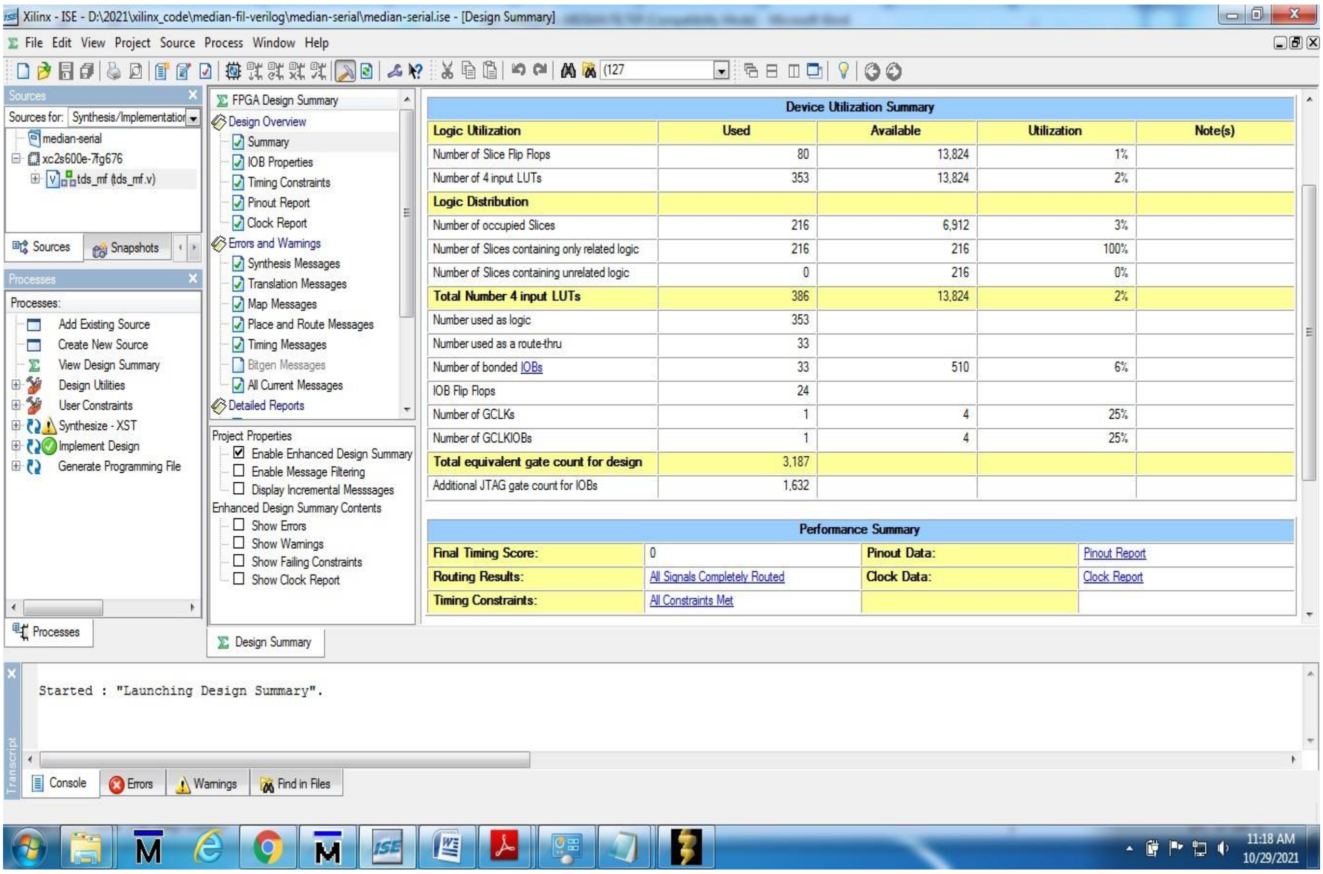


**Figure 11.** Simulation results of serial median filter

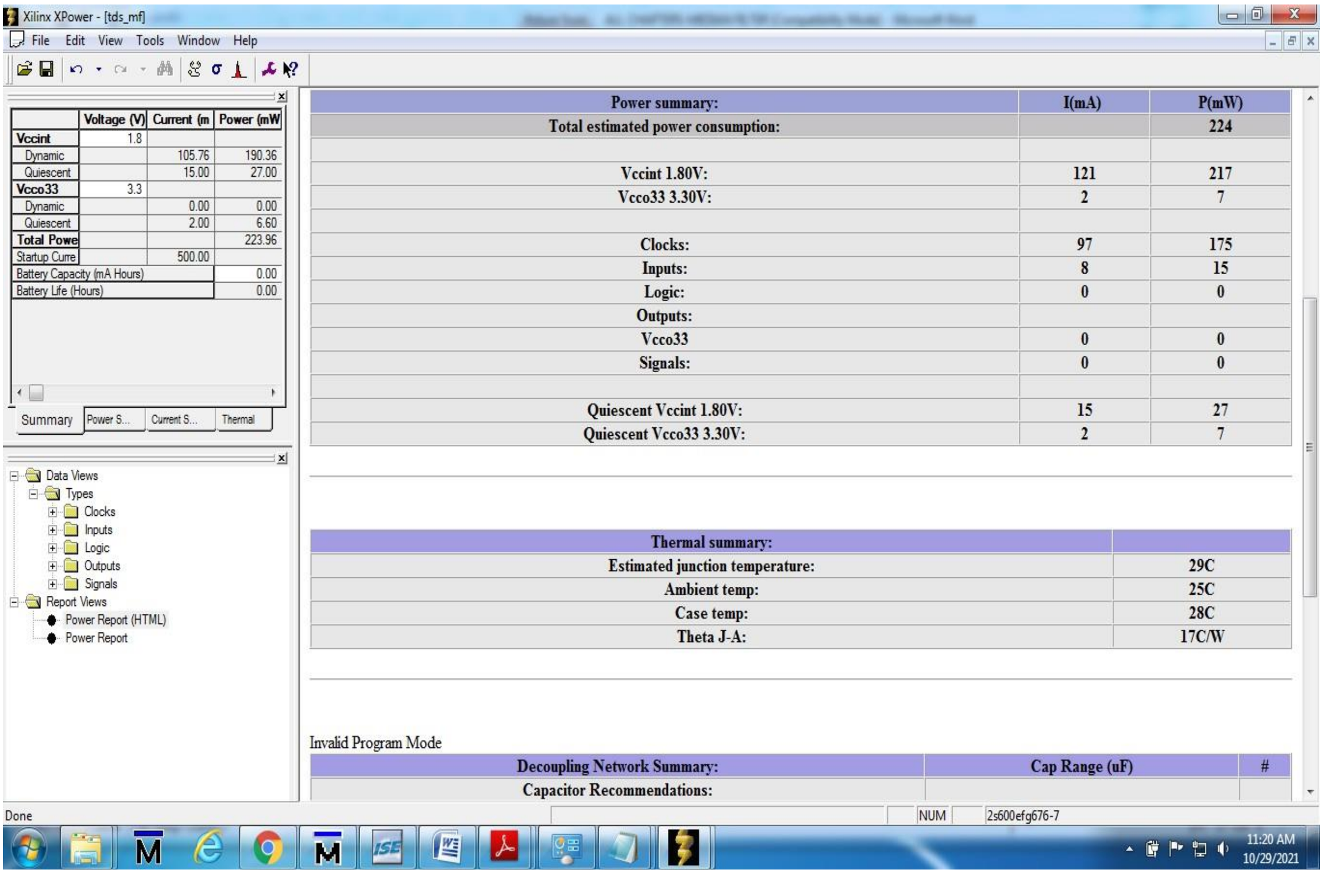


| Device Utilization Summary | | | | |
|---|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** | **Note(s)** |
| Number of Slice Flip Flops | 80 | 13,824 | 1% | |
| Number of 4 input LUTs | 353 | 13,824 | 2% | |
| **Logic Distribution** | | | | |
| Number of occupied Slices | 216 | 6,912 | 3% | |
| Number of Slices containing only related logic | 216 | 216 | 100% | |
| Number of Slices containing unrelated logic | 0 | 216 | 0% | |
| **Total Number 4 input LUTs** | 386 | 13,824 | 2% | |
| Number used as logic | 353 | | | |
| Number used as a route-thru | 33 | | | |
| Number of bonded IOBs | 33 | 510 | 6% | |
| IOB Flip Flops | 24 | | | |
| Number of GCLKs | 1 | 4 | 25% | |
| Number of GCLKIOBs | 1 | 4 | 25% | |
| **Total equivalent gate count for design** | 3,187 | | | |
| Additional JTAG gate count for IOBs | 1,632 | | | |

| Performance Summary | | | |
|---|---|---|---|
| **Final Timing Score:** | 0 | **Pinout Data:** | Pinout Report |
| **Routing Results:** | All Signals Completely Routed | **Clock Data:** | Clock Report |
| **Timing Constraints:** | All Constraints Met | | |

```
Started : "Launching Design Summary".
```

**Figure 12.** Area report of serial median filter



| | Voltage (V) | Current (m | Power (mW |
|---|---|---|---|
| **Vccint** | 1.8 | | |
| Dynamic | | 105.76 | 190.36 |
| Quiescent | | 15.00 | 27.00 |
| **Vcco33** | 3.3 | | |
| Dynamic | | 0.00 | 0.00 |
| Quiescent | | 2.00 | 6.60 |
| **Total Powe** | | | 223.96 |
| Startup Curre | | 500.00 | |
| Battery Capacity (mA Hours) | | | 0.00 |
| Battery Life (Hours) | | | 0.00 |

| Power summary: | I(mA) | P(mW) |
|---|---|---|
| **Total estimated power consumption:** | | **224** |
| | | |
| **Vccint 1.80V:** | **121** | **217** |
| **Vcco33 3.30V:** | **2** | **7** |
| | | |
| **Clocks:** | **97** | **175** |
| **Inputs:** | **8** | **15** |
| **Logic:** | **0** | **0** |
| **Outputs:** | | |
| **Vcco33** | **0** | **0** |
| **Signals:** | **0** | **0** |
| | | |
| **Quiescent Vccint 1.80V:** | **15** | **27** |
| **Quiescent Vcco33 3.30V:** | **2** | **7** |

| Thermal summary: | |
|---|---|
| **Estimated junction temperature:** | **29C** |
| **Ambient temp:** | **25C** |
| **Case temp:** | **28C** |
| **Theta J-A:** | **17C/W** |

Invalid Program Mode

| Decoupling Network Summary: | Cap Range (uF) | # |
|---|---|---|
| **Capacitor Recommendations:** | | |

**Figure 13.** Power report of serial median filter

The logic Utilization is calculated by estimating how many slices of Flip-Flops and four input LUTs to fit a serial median filter design. Only pertinent logic is present in the 216 slices that are utilized. The four input LUTs are divided into two categories: 33 are used as a route-thru and 353 which are used as logic. The serial median filters logic utilization logic distribution and number of four input LUTs are displayed in Table 1.

Table 2 shows that Logic Utilization, Logic Distribution and Number of four input LUTs of Parallel Median Filter. In logic distribution, 5% of Number of occupied Slices is reduced in the Parallel Median Filter. When compared to serial design, parallel design has a 26.76% lower total equivalent gate count. The parallel median filter simulation result, area report and power report are mentioned in Figure 14, Figure 15 and Figure 16.

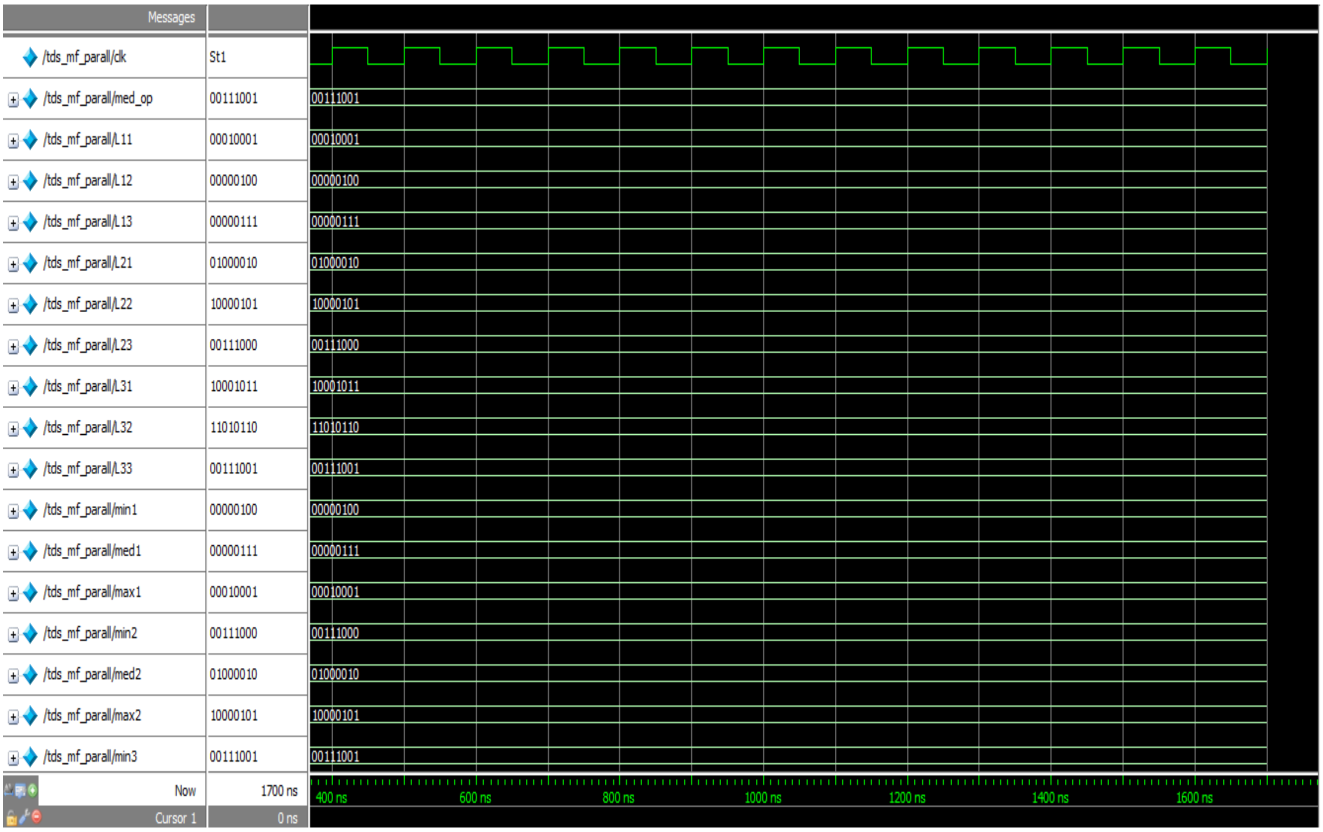
When all inputs are maintained at a valid logic level and the circuit is not in a charging condition, the dynamic power consumption happens. However, dynamic power usage can considerably increase total power consumption when switching at a high frequency. According to Table 3 and Table 4, a parallel median filter uses less dynamic power than a serial median filter. Area Utilization of parallel Median Filter is 26.76% improvement when compare with serial Median Filter. Dynamic power consumption of parallel Median Filter is 36.37% improvement when compare with serial Median Filter and also total estimated power consumption of parallel median filter is 30.80% improvement when compare with serial Median Filter.

**Table 2.** Logic utilization, logic distribution and number of 4-input LUTs of parallel median filter

| Description | Utilized | Available | Utilization |
|---|---|---|---|
| **Logic Utilization** | | | |
| Count of Flip-Flops with a Slice | 389 | 13824 | 2% |
| **Logic Distribution** | | | |
| Count of occupied Slices | 205 | 6912 | 2% |
| Count of slices solely contain related logic | 205 | 205 | 100% |
| The quantity of pieces that lack coherent reasoning | 0 | 205 | 0% |
| **Total Number 4-input LUTs** | **389<br>(Number used as logic – 353<br>Number used as a route thru-36)** | **13824** | **2%** |
| Count of bonded IOBs | 80 | 510 | 15% |
| Count of equivalent gates in the design | 2334 | | |
| IOB JTAG overhead | 3840 | | |

**Table 3.** Power consumption comparison of serial, parallel median filter

| Description | Vin=1.8V | | Vin=3.3V | | Total Estimated Power Consumption (mW) |
|---|---|---|---|---|---|
| | Dynamic (mW) | Quiescent (mW) | Dynamic (mW) | Quiescent (mW) | |
| Serial Median Filter | 190.36 | 27 | 0 | 6.6 | 224 |
| Parallel Median Filter | 121.12 | 27 | 0 | 6.6 | 155 |

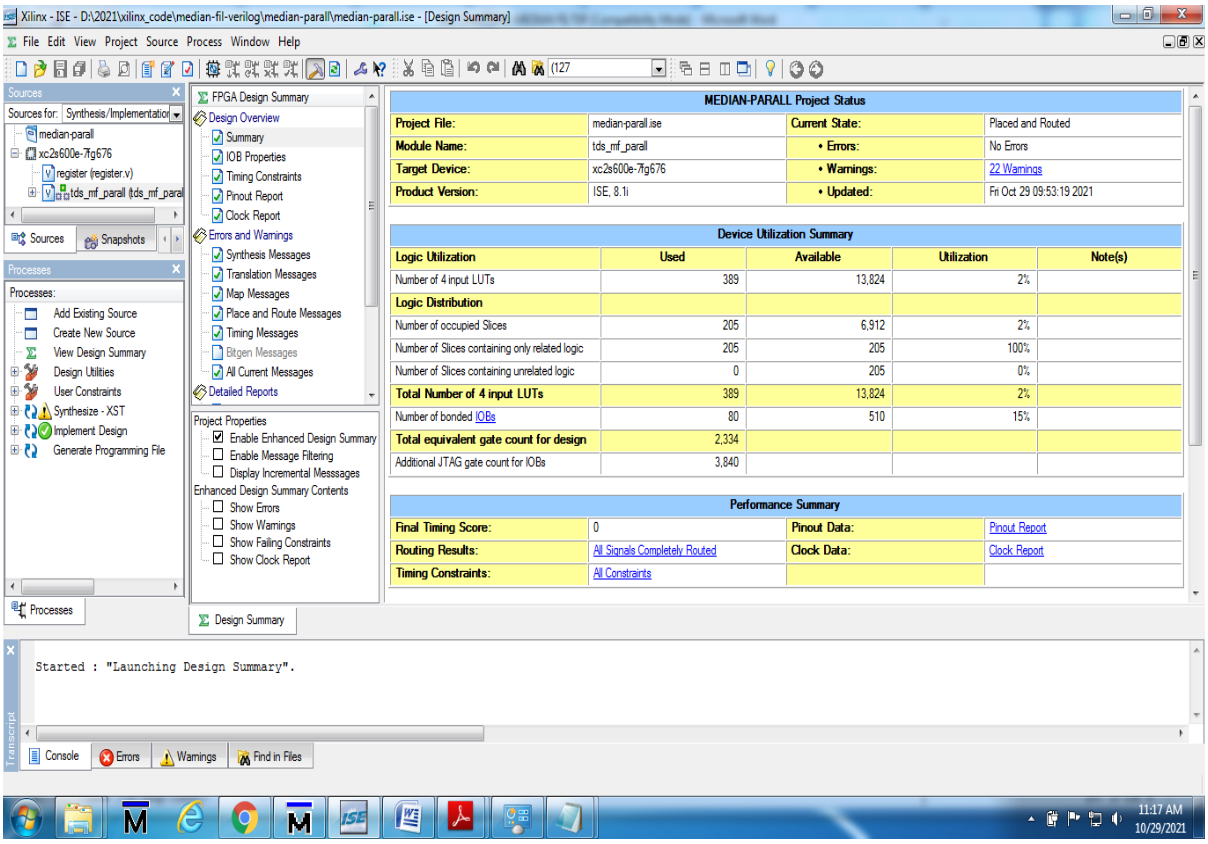


```
Messages
/tds_mf_parall/clk      St1
/tds_mf_parall/med_op   00111001
/tds_mf_parall/L11      00010001
/tds_mf_parall/L12      00000100
/tds_mf_parall/L13      00000111
/tds_mf_parall/L21      01000010
/tds_mf_parall/L22      10000101
/tds_mf_parall/L23      00111000
/tds_mf_parall/L31      10001011
/tds_mf_parall/L32      11010110
/tds_mf_parall/L33      00111001
/tds_mf_parall/min1     00000100
/tds_mf_parall/med1     00000111
/tds_mf_parall/max1     00010001
/tds_mf_parall/min2     00111000
/tds_mf_parall/med2     01000010
/tds_mf_parall/max2     10000101
/tds_mf_parall/min3     00111001
Now                     1700 ns
Cursor 1                0 ns
```

**Figure 14.** Simulation report of parallel median filter

**Table 4.** Area utilization comparison of serial, parallel median filter

| Types of Median Filter | Total Number of Equivalent Gates in the Design | JTAG Gate Count Increase for IOBs |
|---|---|---|
| Serial Median Filter | 3187 | 1632 |
| Parallel Median Filter | 2334 | 3840 |



Xilinx - ISE - D:\2021\xilinx_code\median-fil-verilog\median-parall\median-parall.ise - [Design Summary]

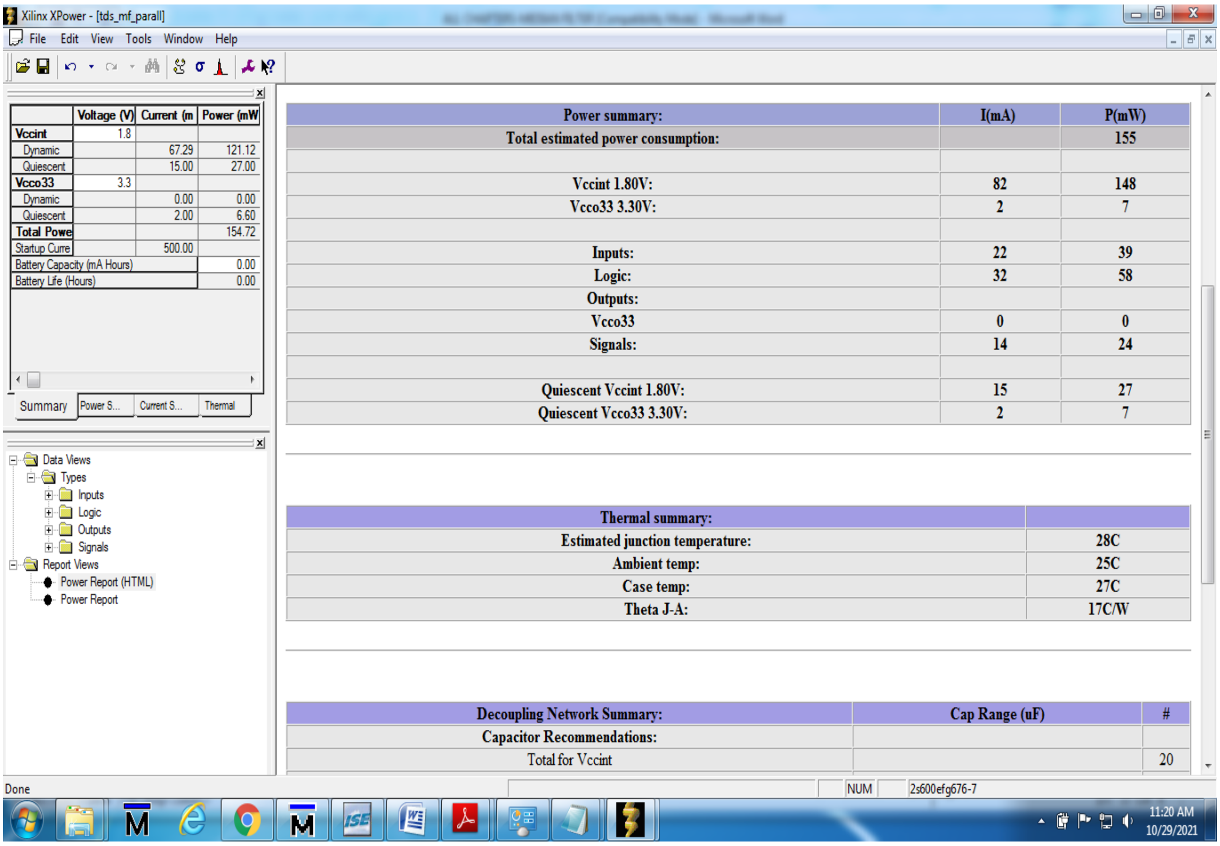
| MEDIAN-PARALL Project Status | | | |
|---|---|---|---|
| **Project File:** | median-parall.ise | **Current State:** | Placed and Routed |
| **Module Name:** | tds_mf_parall | **• Errors:** | No Errors |
| **Target Device:** | xc2s600e-7fg676 | **• Warnings:** | 22 Warnings |
| **Product Version:** | ISE, 8.1i | **• Updated:** | Fri Oct 29 09:53:19 2021 |

| Device Utilization Summary | | | | |
|---|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** | **Note(s)** |
| Number of 4 input LUTs | 389 | 13,824 | 2% | |
| **Logic Distribution** | | | | |
| Number of occupied Slices | 205 | 6,912 | 2% | |
| Number of Slices containing only related logic | 205 | 205 | 100% | |
| Number of Slices containing unrelated logic | 0 | 205 | 0% | |
| **Total Number of 4 input LUTs** | 389 | 13,824 | 2% | |
| Number of bonded IOBs | 80 | 510 | 15% | |
| **Total equivalent gate count for design** | 2,334 | | | |
| Additional JTAG gate count for IOBs | 3,840 | | | |

| Performance Summary | | | |
|---|---|---|---|
| **Final Timing Score:** | 0 | **Pinout Data:** | Pinout Report |
| **Routing Results:** | All Signals Completely Routed | **Clock Data:** | Clock Report |
| **Timing Constraints:** | All Constraints | | |

```
Started : "Launching Design Summary".
```

**Figure 15.** Area report of parallel median filter



Xilinx XPower - [tds_mf_parall]

| | Voltage (V) | Current (m | Power (mW |
|---|---|---|---|
| **Vccint** | 1.8 | | |
| Dynamic | | 67.29 | 121.12 |
| Quiescent | | 15.00 | 27.00 |
| **Vcco33** | 3.3 | | |
| Dynamic | | 0.00 | 0.00 |
| Quiescent | | 2.00 | 6.60 |
| **Total Powe** | | | 154.72 |
| Startup Curre | | 500.00 | |
| Battery Capacity (mA Hours) | | | 0.00 |
| Battery Life (Hours) | | | 0.00 |

| Power summary: | I(mA) | P(mW) |
|---|---|---|
| Total estimated power consumption: | | 155 |
| | | |
| Vccint 1.80V: | 82 | 148 |
| Vcco33 3.30V: | 2 | 7 |
| | | |
| Inputs: | 22 | 39 |
| Logic: | 32 | 58 |
| Outputs: | | |
| Vcco33 | 0 | 0 |
| Signals: | 14 | 24 |
| | | |
| Quiescent Vccint 1.80V: | 15 | 27 |
| Quiescent Vcco33 3.30V: | 2 | 7 |

| Thermal summary: | |
|---|---|
| Estimated junction temperature: | 28C |
| Ambient temp: | 25C |
| Case temp: | 27C |
| Theta J-A: | 17C/W |

| Decoupling Network Summary: | Cap Range (uF) | # |
|---|---|---|
| Capacitor Recommendations: | | |
| Total for Vccint | | 20 |

**Figure 16.** Power report of parallel median filter

## 6. CONCLUSION AND FUTURE WORK

An efficient way to trade-off the physical attributes like, area, power and speed of a median filter with precision is proposed by creating a novel inaccurate two-bit magnitude comparator. Every one of these filters leaves a different amount of noise in the images. For high-speed applications that handle data in parallel the parallel median filter also includes a second ternary data sorter constructed in. Even though it is not visible to the human eye simulation data indicates that there is a slight loss in image quality when compared to the exact filter. The inaccuracy performance and structure of median filters and imprecise comparators were determined. The physical characteristics of the proposed inaccurate median filters were far improved than those of the accurate one. Simulations reveal that using imprecise filters can have a 15% speed boost over using standard accurate filters and cost savings of up to 26% and 30% respectively in the context of power and area usage.

Look-ahead clock gating (LACG) can be used in subsequent work. An intermediate register calculates the clock enabling signals for each flip-flop (FF) one cycle in advance, based on the current cycle data of the FFs it relies on. Its ability to compute the enabling signals for a full clock cycle and allow them to propagate to their gates eliminates a significant drawback associated with earlier approaches strict timing requirements.

## REFERENCES

[1] Gonzalez, R.C., Woods, R.E. (2017). Digital Image Processing. 4ed. Pearson.

[2] Erkan, U., Gökrem, L., Enginoğlu, S. (2018). Different applied median filter in salt and pepper noise. Computers & Electrical Engineering, 70: 789-798. https://doi.org/10.1016/j.compeleceng.2018.01.019

[3] Sekanina, L., Vasicek, Z., Mrazek, V. (2019). Automated search-based functional approximation for digital circuits. In: Reda, S., Shafique, M. (eds) Approximate Circuits, Springer, Cham, pp. 175-203. https://doi.org/10.1007/978-3-319-99322-5_9

[4] Vasicek, Z., Mrazek, V. (2016). Trading between quality and non-functional properties of median filter in embedded systems. Genetic Programming and Evolvable Machines, 26(3): 45-82. https://doi.org/10.1007/s10710-016-9275-7

[5] Monajati, M., Fakhraie, S.M., Kabir, E. (2015). Approximate arithmetic for low-power image median filtering. Circuits, Systems, and Signal Processing, 34(10): 3191-3219. https://doi.org/10.1007/s00034-015-9997-4

[6] HosseinKhani, Z., Hajabdollahi, M., Karimi, N., Soroushmehr, R., Shirani, S., Najarian, K., Samavi, S. (2018). Adaptive real-time removal of impulse noise in medical images. Journal of Medical Systems, 42(11): 216-225. https://doi.org/10.1007/s10916-018-1074-7

[7] Vairaprakash, S., Karuppaiah, J., Buvaneswari, B., Alghamdi, A. (2023). Removal of salt and pepper noise using hybrid adaptive switching median filter with ant colony optimization technique in nano electronic applications. Journal of Nanoelectronics and Optoelectronics, 18(1): 85-94. https://doi.org/10.1166/jno.2023.3365

[8] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P. (2004). Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4): 600-612. https://doi.org/10.1109/TIP.2003.819861

[9] Liang, H., Zhao, S. (2019). Removal of salt-and-pepper noise from bimodal images using a non-symmetry and anti-packing model. Journal of Medical Imaging and Health Informatics, 9(1): 84-90. https://doi.org/10.1166/jmihi.2019.2544

[10] Brioua, A., Benzid, R., Saidi, L. (2023). Efficient wavelet thresholding and wiener filtering association incorporating a median filter smoother followed by R-peaks recovery for ECG denoising. Traitement du Signal, 40(3): 1211-1217. https://doi.org/10.18280/ts.400338

[11] Kalaimani, G., Manojkumar, K., Kumar, S.S. (2019). Median filtering for removal of maximum impulse noise from images with a decision based model. Journal of Computational and Theoretical Nanoscience, 16(2): 562-567. https://doi.org/10.1166/jctn.2019.7769

[12] Li, Q., Chen, C.X., Huang, T., Huang, Z., Hua, J.X., Li, L.L. (2021). A median filtering algorithm for removing high density salt and pepper noise. Computer Knowledge and Technology, 17(35): 137-139. https://doi.org/10.14004/j.cnki.ckt.2021.3470

[13] Ismael, A.A., Baykara, M. (2022). Image denoising based on implementing threshold techniques in multi-resolution wavelet domain and spatial domain filters. Traitement du Signal, 39(4): 1119-1131. https://doi.org/10.18280/ts.390404

[14] Alaoui, N., Mashat, A., Adamou-Mitiche, A.B.H., Mitiche, L., Djalab, A., Daoudi, S., Bouhamla, L. (2021). Impulse noise removal based on hybrid genetic algorithm. Traitement du Signal, 38(4): 1245-1251. https://doi.org/10.18280/ts.380436

[15] Shrestha, S. (2014). Image denoising using new adaptive based median filter. Signal & Image Processing: An International Journal (SIPIJ), 5(4): 1-13. https://doi.org/10.5121/sipij.2014.5401

[16] Abadi, H.Z.H., Samavi, S., Karimi, N. (2013). Image noise reduction by low complexity hardware median filter. In 2013 21st Iranian Conference on Electrical Engineering (ICEE), Mashhad, Iran, pp. 1-5. https://doi.org/10.1109/IranianCEE.2013.6599571

[17] Sánchez, R.M., Rodríguez, P.A. (2012). Bidimensional median filter for parallel computing architectures. In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, pp. 1549-1552. https://doi.org/10.1109/ICASSP.2012.6288187

[18] Draz, H.H., Elashker, N.E., Mahmoud, M.M. (2023). Optimized algorithms and hardware implementation of median filter for image processing. Circuits, Systems, and Signal Processing, 42(9): 5545-5558. https://doi.org/10.1007/s00034-023-02370-x

[19] Ismael, A.A., Baykara, M. (2021). Digital image denoising techniques based on multi-resolution wavelet domain with spatial filters: A review. Traitement du Signal, 38(3): 639-651. https://doi.org/10.18280/ts.380311

[20] Ouali, M.A., Ghanai, M., Chafaa, K. (2020). TLBO optimization algorithm based-type2 fuzzy adaptive filter for ECG signals denoising. Traitement du Signal, 37(4): 541-553. https://doi.org/10.18280/ts.370401