**Indonesian Journal of Electrical Engineering and Computer Science**
Vol. 11, No. 3, September 2018, pp. 1048~1059
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v11.i3.pp1048-1059


# FPGA Implementation of a Novel Gaussian Filter Using Power Optimized Approximate Adders

**Jamshid M Basheer[1], Murugesh V[2]**
[1]Manonmaniam Sundaranar University, Tirunelveli, India
[1,2]Department of Computer Engineering, College of Computer Science, King Khalid University, Saudi Arabia

**Article Info**

***Article history:***

Received May 23, 2018
Revised Jun 21, 2018
Accepted Jul 2, 2018

***Keywords:***

Approximate adders
FPGA
Gaussian filter
Image processing
Low power

**ABSTRACT**

Smoothing filters are essential for noise removal and image restoration. Gaussian filters are used in many digital image and video processing systems. Hence the hardware implementation of the Gaussian filter becomes a reliable solution for real time image processing applications. This paper discusses the implementation of a novel Gaussian smoothing filter with low power approximate adders in Field Programmable Gate Array (FPGA). The proposed Gaussian filter is applied to restore the noisy images in the proposed system. Original test images with 512x512 pixels were taken and divided in to 4x4 blocks with 256x256 pixels. The proposed technique has been applied and the performance metrics were measured for various simulation criteria. The proposed algorithm is also implemented using approximate adders, since approximate adders had been recognized as a reliable alternate for error tolerant applications in circuit based metrics such as power, area and delay where the accuracy may be considered for trade off.



***Corresponding Author:***

Jamshid Basheer. M,
Department of Computer Engineering,
College of Computer Science,
King Khalid University, Kingdom of Saudi Arabia.
Email: jamsvlsi@gmail.com

## 1. INTRODUCTION

Computer vision and Image processing are attainig tremendous scope in various real life applications like security systems, robotics, quality control, and automation. Gaussian smoothing filter is the most common filter which is applied for smoothing images. In real life scenario, images are recorded to display the pictorial information. The recorded original image might be degraded due to imperfections in the imaging and the method of capture. Eliminating the blur or noise from the image to be processed is essential operation to many of the image processing techniques. Quality of the images is reduced by different degradations like instantaneous noise, blur, color imperfections, geometrical degradations and imperfect illumination. The intension of the image restoration process is to estimate the difference between the blurred image and the original image to apply the respective technique. Image restoration is used in medical, astronomy and forensic field [1].

The process of restoring the image is nothing but extracting the good image from the blurred and noisy image. The blind image deconvolution process is often referred to as a combination of image restoration [2]. If the original image which does not contain any noise or blur is denoted by i(a1, b1), then the recorded image k(a1, b1) is modeled as follows.

$$k(a1, b1) = m(a1, b1)* i(a1, b1) + j(a1, b1) \quad (1)$$

where, m(a1, b1) represents the blurring function; j(a1, b1) represents the noise.

Figure 1, depicts the representation of spatial and Fourier domain images. The other way of describing Equation (1) is through its spectral equivalence. By using Discrete Fourier transforms to Equation (1), mentioned in Equation 2,

$$K(u1, v1) = M(u1, v1)I(u1, v1) + J(u1, v1) \quad (2)$$

Where, (*u1*,*v1*) are spatial coordinates and Fourier transforms of the spatial coordinates were represented by the uppercase. Either Equation (1) or Equation (2) are used to represent image restoration algorithm [3]. Image restoration can be realized in Fourier domain by its spectral representation. In Equation (1) and Equation (2), the noise J(a1, b1) is represented as an additive term. White noise is taken with zero mean [4].

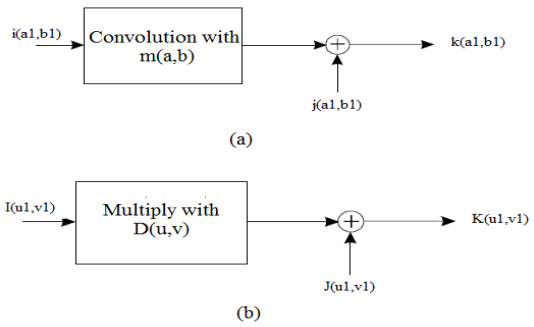



Figure 1. (a) Spatial domain Image (b) Fourier domain Image in

The general image restoration process is depicted in Figure 2. The actual test image i(a1, b1) is blurred by the blurring function m(a1, b1) and then additive Gaussian noise j(a1, b1) is added the original test image. The degraded image k(a1, b1) is then passed through the restoration filter R(a1, b1) and the restored image is represented as i'(a1, b1) [5].

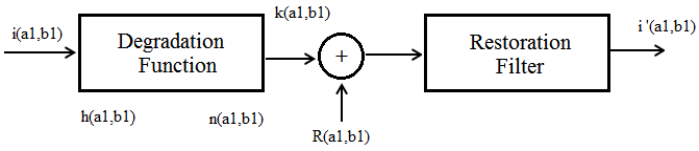



Figure 2. Image Restoration Process Model

Many image restoration algorithms or methods are found in the review of literature. Every algorithm has its own merits and demerits. A nonlocal and local consistency based mixed Gaussian impulse noise removal technique is discussed [6]. Hyper Laplace prior method is used to measure the local consistency by exploiting the local smoothness of images. Image restoration with texture preserving and dyadic bounded mean method is proposed. In this method the proposed model is transformed into wavelet domain. Space adaptive Lagrange multipliers are used to control the denoising work [7].

Image restoration by a non-local means (NLM) filter is proposed in this article and noisy image is processed by Gaussian filter. Preprocessed image based on the spatial distance method is used to calculate the similarity by the search window. The proposed NLM method restores intensity of each pixel [8].Classification decision based adaptive filter is applied to restore the degraded images. The restored images were represented by finite normal-density mixture (FNM) models by applying fixed number of Gaussian components [9]. Segmentation algorithm based on Gaussian blurring is applied to investigate the filtered images. As the segmentation method consumes more time, a proper filtering technique is to be selected to get improved results [10]. Adopted towards trend based tool, database and techniques and repeatedly used attack scenerios. Significant open research were explored after many reviews from existing systems [11]. The work choosen three algorithms they are: OMP, StOMP, CoSaMP as the research [12]. A hybrid statistical method related noise suppression techniques have been developed for the purpose to increase the quality of the impulse noisy of the color images. It proved the performance in image enhancement methodology by using the advanced performance metrices.

## 2. FPGA IMPLEMENTATION

In recent years Field Programmable Gate Array (FPGA) is mostly used to realize the hardware for the image processing algorithms for real time applications. It is desirable for various real time video image processing applications. FPGA can be used to implement any image processing algorithm. Efficient modified guided filter architecture is implemented on Xilinx FPGA device which offers less cost, speed and low power. Modified guided filter for real time video has been implemented in FPGA [13].

Bilateral filters are used for denoising in real time videos since it preserves the details of the image while the denoising operation is performed. A synchronous bilateral filter for real time image denoising has been implemented in FPGA [14]. Most of the image processing algorithms were implemented in FPGA to speed up the image processing applications. Recently approximate computing is getting popular because of reduced hardware cost and the low power consumption. Approximate adders were being used in digital image processing and digital signal processing to achieve low power consumption and optimized hardware by sacrificing the accuracy as tradeoff. Reduction of logic complexity for approximate adders is applied to test the area, delay and power consumption. The circuits were implemented in transistor level [15].

Approximate computing is a promising technique but the accuracy is slightly degraded. A novel method to reduce the error characteristic of the approximate adders has been implemented. In this work a framework based error characteristic was analyzed [16]. Hardware for neural network has many constraints like, low area, reduced delay and low power consumption. Since the neural networks can do parallel processing, high speed multipliers were used to minimize the delay. As the multipliers consume more area and power, it is necessary to use approximate computing elements as the building blocks for the approximate multiplier. Iterative logarithmic multiplier has been implemented in hardware with approximate adders and the performances were analyzed [17].

Approximate computing is used in signal processing using distributed arithmetic to optimize the performance metrics of the target hardware at the cost of accuracy. A high precision distributed arithmetic circuits were implemented using Micro Electro Mechanical Systems for medical applications. Distributed arithmetic based micro power programmable approximate signal processing was implemented [18].

Low power and high efficiency hardware is needed for modern in battery operated image processing applications to provide more battery backup for other applications. Approximate adders and multipliers were used to achieve low power consumption with reduced hardware architecture where accuracy is a tradeoff. But still the percentage of the error contributed by the approximate computing is negligible and hence it will not degrade the overall system performance. Modified approximate adders were used in Discrete Cosine Transform (DCT) to minimize the power consumption and reduce the hardware cost for image compression. By applying the approximate adders, the logic functions of the DCT algorithm are simplified and the proposed hardware consumes less area and power [19].

Approximate computing is especially used where the effect of error in the proposed system does not degrade the system performance much. The proposed system becomes error tolerant and performs similar as conventional system. Approximate adders for error tolerant applications was designed and implemented. The carry propagation is eliminated in the proposed design. The error tolerant adder outperforms the conventional adder in terms of the performance metrics like area, power and delay at the cost of accuracy [20].

## 3. PROPOSED ALGORITHM

The proposed novel Gaussian filter is implemented in FPGA with the approximate adders to reduce the size of the hardware and the power consumption. The error caused by the approximation degrades the accuracy slightly but this is negligible in the error tolerant system like image processing. In the proposed algorithm, the original image will be divided into 4x4 cells each with pixel size of 256x256 pixels before applying the proposed Filtering technique. By applying this technique, the proposed scheme achieves PSNR of 29.75 dB and SSIM as 0.714. The steps involved the proposed algorithm is depicted below.

Algorithm:

1. Original test image of 512x512 pixel size is taken as input image.
2. The input image is divided in to four cells of each 256x256 pixels.
3. Mean and variance for each cell is calculated to make the decision.
4. Median of variance of each cell is calculated and checked for following criteria,

```
if var > median
    flag=2;
else
    flag=1;
 end
```

5. For flag = 2, deconvolution and Gaussian filter of 5x5 window is applied

6. For flag = 1, deconvolution and Gaussian filter of 3x3 window is applied
Modified Gaussian filter is represented as follows,

$$h_{x,y}\frac{1}{K}\exp\left[-L\frac{(\sigma_{i,j}')^2(x^2+y^2)}{\sqrt{\mu_{i,j}'+1}}\right] \quad (3)$$

where, K and L denote the normalizing constant. The variation in the standard deviation of the original and blurred images is denoted by $\sigma_{i,j}'$ and $\mu_{i,j}'$ denotes the difference between the mean of original and the blurred images. “Baboon”, “Boat”, “Barbara”, “Pepper” and Zelda images were taken to test the proposed scheme.

## 4. SIMULATION RESULTS

Various original test images of 512x512 pixels were taken to validate the proposed scheme. Various levels of noises were added and the noisy or blur images were restored by the proposed filtering scheme. Simulations were carried out in MATLAB and the FPGA implementation is performed in Xilinx ISE. The original images were divided in to 4x4 cells of size 256x256 pixels. Mean, variance and the standard deviations were calculated and then the proposed scheme was applied. Median of the variance is calculated and this is used to decide the flag in the proposed scheme. When the variance is more than the median, flag2 is selected and when the variance is less than the median, then the flag1 is selected. Original test images were taken to validate the proposed algorithm by adding blur and noise. The original test images with their blurred version and the restored version are depicted below. The test images were blurred with different noise levels with different variance and then the blurred images were recovered by the proposed technique.

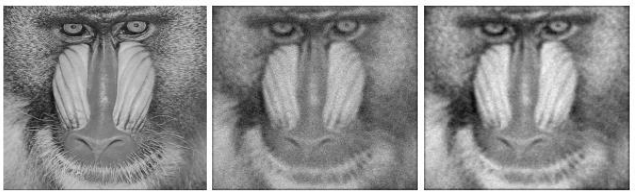
Test Image -1: Baboon

Variance: 0.001

(i) (ii) (iii)

Figure 3. Baboon: (i) Actual (ii) Noisy (iii) Restored

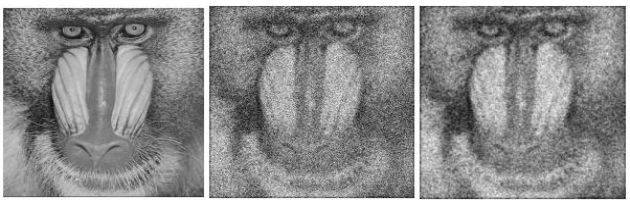
Variance: 0.0



(i) (ii) (iii)

Figure 4. Baboon: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.1



(i) (ii) (iii)

Figure 5. Baboon: (i) Actual (ii) Noisy (iii) Restored

Original images, blurred versions and the restored versions of the test image “Baboon” are depicted in Figure 3, Figure 4, Figure 5 with variance of 0.001, 0.01 and 0.1 respectively.

Test Image -2: Boat

Variance: 0.001

(i) (ii) (iii)

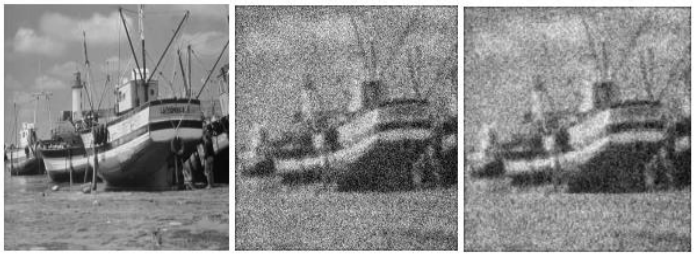
Figure 6. Boat: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.01

(i) (ii) (iii)

Figure 7. Boat: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.1



(i) (ii) (iii)

Figure 8. Boat: (i) Actual (ii) Noisy (iii) Restored

Original images, blurred versions and the restored versions of the test image “Boat” are depicted in Figure 6, Figure 7, Figure 8 with variance of 0.001, 0.01 and 0.1 respectively.

Test Image -3: Barbara

Variance: 0.001

(i) (ii) (iii)

Figure 9. Barbara: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.01

(i) (ii) (iii)

Figure 10. Barbara: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.1

(i) (ii) (iii)

Figure 11. Barbara: (i) Actual (ii) Noisy (iii) Restored

Original images, blurred versions and the restored versions of the test image "Barbara" are depicted in Figure 9, Figure 10, Figure 11 with variance of 0.001, 0.01 and 0.1 respectively. Original images, blurred versions and the restored versions of the test image "Peppers" are depicted in Figure 12, Figure 13, Figure 14 with variance of 0.001, 0.01 and 0.1 respectively.

Test Image -4: Peppers

Variance: 0.001

(i) (ii) (iii)

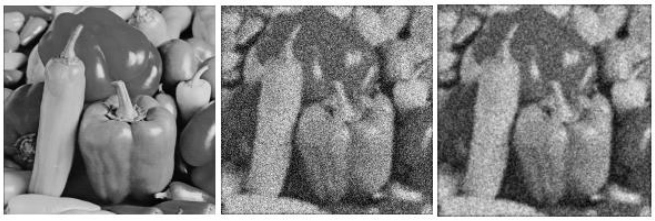
Figure 12. Peppers: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.01

(i) (ii) (iii)

Figure 13. Peppers: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.1



(i) (ii) (iii)

Figure 14. Peppers: (i) Actual (ii) Noisy (iii) Restored

Original images, blurred versions and the restored versions of the test image "Zelda" are depicted in Figure 15, Figure 16, Figure 17 with variance of 0.001, 0.01 and 0.1 respectively.

Test Image -5: Zelda

Variance: 0.001

(i) (ii) (iii)

Figure 15. Zelda: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.01

(i) (ii) (iii)

Figure 16. Zelda: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.1

(i) (ii) (iii)

Figure 17. Zelda: (i) Actual (ii) Noisy (iii) Restored

Test Image -6: Lena

Original images, blurred versions and the restored versions of the test image "lena" are depicted in Figure 18, Figure 19, Figure 20 with variance of 0.001, 0.01 and 0.1 respectively.

Variance: 0.001

(i) (ii) (iii)

Figure 18. Lena: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.01

(i) (ii) (iii)

Figure 19. Lena: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.1

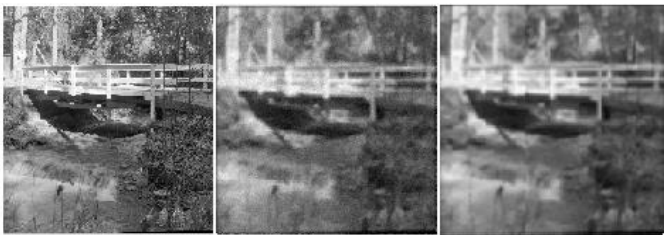
(i) (ii) (iii)

Figure 20. Lena: (i) Actual (ii) Noisy (iii) Restored

Test Image -7: Stream Bridge

Original images, blurred versions and the restored versions of the test image "Stream Bridge" are depicted in Figure 21, Figure 22, Figure 23 with variance of 0.001, 0.01 and 0.1 respectively.

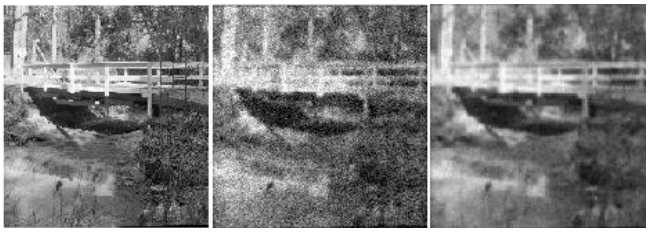
Variance: 0.001



(i) (ii) (iii)

Figure 21. Stream Bridge: (i) Actual (ii) Noisy (iii) Restored

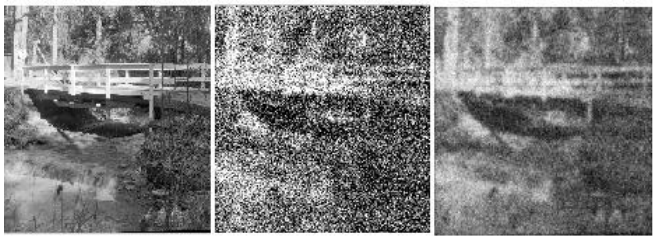
Variance: 0.01



(i) (ii) (iii)

Figure 22. Stream Bridge: (i) Actual (ii) Noisy (iii) Restored

Variance: 0.1



(i) (ii) (iii)

Figure 23. Stream Bridge: (i) Actual (ii) Noisy (iii) Restored

Simulation of the proposed novel Gaussian filter is validated with different levels of variance 0.001, 0.01 and 0.1 on the original test images viz. “Baboon”, “Boat”, “Barbara”, “Peppers”, “Zelda”, “Lena” and “Stream Bridge”. Original test image of 512x512 pixel size is taken as input image. The input image is divided in to four cells of each 256x256 pixels. Mean and variance for each cell is calculated to make the decision. Median of variance of each cell is calculated and checked for following criteria, if the variance is greater than the median, then the falg2 is selected else flag1 is selected. Mean and variance is calculated for each cell in the proposed algorithm. As the mean of the variance increases, the original image is degraded to the maximum but still the proposed Gaussian filter is able to restore the noisy image and gives better structural similarity index. The image quality of the restored test images proves that the proposed technique provides the better restoration compared to other filtering.

The proposed Gaussian filter has been realized as hardware in FPGA using Xilinx ISE Version 14.2 EDA software. Long-Term Maintenance, Time to Market, Performance, Cost and Reliability are the major advantages of the FPGA implementations. Performance: In the recent years the hardware parallelism are gaining attention, with that advantages, the FPGAs outperforms the computing power of digital signal processors (DSPs) by parallel execution and achieving more data processing speed. Time to market: New technologies in FPGA offer flexibility and rapid prototyping capabilities towards the faster time-to-market. Cost: The FPGA implementation becomes cost effective solution than Application Specific Integrated Circuits (ASIC) implementation when very few number of prototypes are needed. Reliability: FPGAs are more reliable than that of processer, since the processor needs software. Long-term maintenance: FPGA devices are field-upgradable and they there reconfigurable. They are very cheaper and consume very less time to market compared to the ASIC redesign. Figure 24 shows the Register Transfer Level (RTL) schematic of the proposed Gaussian filter which is implemented in the Xilinx FPGA xc3s100e-5vq100.

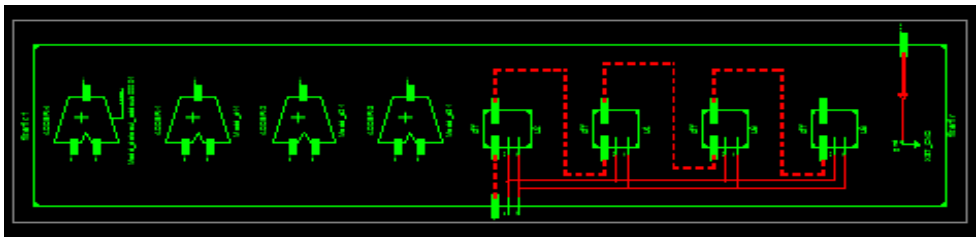


Figure 24. RTL_Schematic of the proposed filter

The more detailed component level schematic which is also called as technology schematic is depicted in Figure 25. Technology schematic shows the number of Look Up Tables (LTUs) which are used to implement the proposed algorithm. Technology schematic also shows how the functions are implemented in the LUTs of the FPGA. The proposed algorithm consumes very few resources of the FPGA and the execution time is also very less. The numeric values are listed out in the Table 1 and in text.

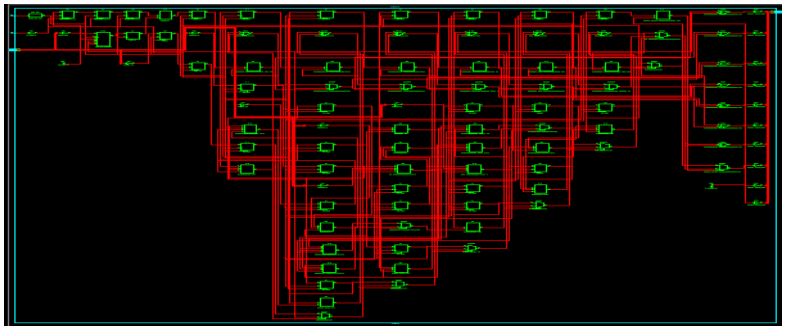


Figure 25. Technology Schematic of the proposed Modified Gaussian Filter

| Device Utilization Summary | | | |
|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** |
| Number of Slice Flip Flops | 29 | 1,920 | 1% |
| Number of 4 input LUTs | 26 | 1,920 | 1% |
| Number of occupied Slices | 30 | 960 | 3% |
| Number of Slices containing only related logic | 30 | 30 | 100% |
| Number of Slices containing unrelated logic | 0 | 30 | 0% |
| Total Number of 4 input LUTs | 26 | 1,920 | 1% |
| Number used as logic | 25 | | |
| Number used as Shift registers | 1 | | |
| Number of bonded IOBs | 19 | 66 | 28% |
| Number of BUFGMUXs | 1 | 24 | 4% |
| Average Fan-out of Non-Clock Nets | 2.18 | | |

Figure 26. Summary of device utilization

The summary of device utilization of the proposed design is shown in Figure 26. Where the number of flip-flops, number of Look Up Tables (LUT), shift registers and other details are mentioned clearly. HDL Synthesis Report shows the number of adders and subtractors which were used to implement the proposed algorithm. In this design three 8-bit adders were used in the algorithm to perform the computation. These adders are conventional adders which consumes more area and power.

**Macro Statistics**

# Adders/Subtractors : 4
8 bit adder : 3
8 bit adder carry out : 1
# Registers : 4
# Registers : 24
Flip-Flops : 24

# Shift Registers : 1
4 bit shift register : 1

The maximum period, minimum input arrival time before the clock, maximum output required time after clock and maximum combinational path delay were calculated for the proposed design and they are mentioned in Table 1.

Table 1. Minimum period 3.492ns (Maximum Frequency: 286.369MHz)

| Minimum period | 3.492ns (Maximum Frequency: 286.369MHz) |
|---|---|
| Minimum input arrival time before clock: | 2.973ns |
| Maximum output required time after clock | 12.243ns |
| Maximum combinational path delay | 12.752ns |
| Worst case slack | 0.809ms |
| Best case achievable slack | 1.70ns |

The proposed modified work was carried out using approximate adders.

**Approximation 1**

In general the first approximate mirror adders are designed to reduce the transistor count in the proposed schematic till the predefined error constraints are met. When the error constraints are safisfied, the last modified schematic is utilized [21]. This technique is utilized to the other approximate adders also. The schematic diagram of Approximate Mirror Adder1 (AMA) is shown in Figure 27.

**Approximation 2**

The second Approximate Mirror Adder (AMA2) gives two errors in Count and three errors in Sum, as shown in Table 2. Figure 28 shows the approximate adder.

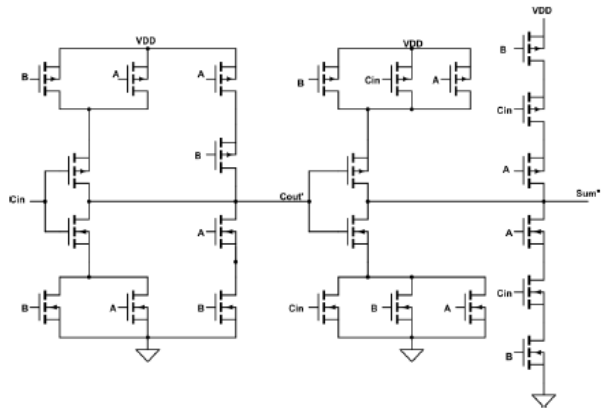



Figure 27. Conventional Mirror Adder [22]

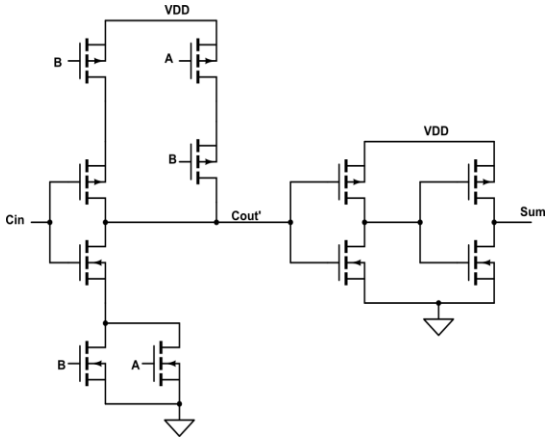



Figure 28. Approximate Adder [20]

Table. 2 Truth Table of Conventional Full Adder and Approximate Full Adders 1& 2 [23]

| Inputs | | | Accurate Outputs | | Approximate Outputs | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $A$ | $B$ | $C_{in}$ | $Sum$ | $C_{out}$ | $Sum_1$ | $C_{out1}$ | $Sum_2$ | $C_{out2}$ | $Sum_3$ | $C_{out3}$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

By applying the approximate adders in the proposed design large amount of dynamic power dissipation is minimized (25% - 40%, from the literature) moreover area and delay are also optimized at the cost of little bit accuracy but still this error is negligible.

## 5. CONCLUSION

Simulation of the proposed modified Gaussian filter is validated with different levels of variance 0.001, 0.01 and 0.1 on the original test images viz. “Baboon”, “Boat”, “Barbara”, “Peppers” and “Zelda”. Mean and variance is calculated for each cell. Median of variance of each cell is calculated and checked for the decision to be taken. If the calculated variance is greater than the median then the deconvolution and Gaussian filtering is applied on 5x5 window else 3x3 window is applied. Achieved Peak Signal-to-Noise Ratio (PSNR) by the proposed algorithm is 29.75 and Structural SIMilarity (SSIM) is 0.714. The proposed design was implemented in the FPGA and more over the adders used in the design were replaced by the approximate adders to optimize the area, power and delay. By applying approximate adders around 40% power can be saved in the proposed design at the cost of little bit accuracy but still this error will not cause a big difference in the performance of the proposed design.

## REFERENCES


[1] Banha, M.R., and Katsaggelos, A.K., "Digital Image Restoration", *IEEE Signal Processing Magazine*, vol. 14, no. 2, pp.24-41, March, 1997.

[2] Kundur D., and Hatzinakos, D., "Blind Image Deconvolution", *IEEE Signal Processing Magazine*, vol. 13, no. 3, pp. 43-64, May, 1996.

[3] Jamshid, M.B., and Murugesh, V., "Image Restoration using Modified Gaussian Filter", *International Journal of Applied Engineering Research*, vol. 10, no. 13, 2015.

[4] Reginald, L. L., and Jan, B., "Basic Methods for Image Restoration and Identification", February, 1999.

[5] Swati, S., Shipra S., and Rajesh M., "Image Restoration using Modified Lucy Richardson Algorithm in the Presence of Gaussian and Motion Blur", *Advance in Electronic and Electric Engineering*, vol. 3, no. 8, pp. 1063-1070, 2013.

[6] Jian, Z., Ruiqin, X., Chen, Z., Siwei, M., and Debin, Z., "Exploiting image local and nonlocal consistency for mixed Gaussian-impulse noise removal", *IEEE International Conference on Multimedia and Expo*, pp. 592-597, 2012.

[7] Tao, Z., Qiuli, G., and Gaoshan, T., "Texture Preserving Image Restoration with DyadicBounded Mean Oscillating Constraints", *IEEE Signal Processing Letters*, vol. 22, no. 3, pp. 322-326, March, 2015.

[8] Yi, Z., Mingyue, D., Feng, X., and Xuming, Z., "An Improved Non-local Means Filter for Image Denoising", *International Conference on Intelligent Computation and Bio-Medical Instrumentation*, pp.31-34, 2011.

[9] Alexia, G., "Classification-Based Adaptive Filtering for Multiframe Blind Image Restoration", *IEEE Trans. Image Processing*, vol. 20, no. 2, pp. 382-390, 2011.

[10] Estevão, S., Gedraite, M. H., "Investigation on the Effect of a Gaussian Blur in Image Filtering and Segmentation", *53rd International Symposium ELMAR-2011, Zadar, Croatia*, pp. 393-396, 14-16 September, 2011.

[11] Shashidhar, T. M., and Ramesh, K.B., "Reviewing the Effectivity Factor in Existing Techniques of Image Forensics", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no.6, pp. 3558-359, 2017.

[12] Yubo, Z., Dongmei, W., Lingling, K., Panpan, Z., "A Study on Image Reconfiguration Algorithm of Compressed Sensing ", *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol.15, no.1, pp. 299-305, 2017.

[13] Soorya, B., and Dhanalakshmi, S., "VLSI Implementation of Modified Guided Filter for Real Time Video", ARPN Journal of Engineering and Applied Sciences. vol. 10, no. 7, pp. 3067-3071, April, 2015.

[14] Anna, G. R., Matthias, K., Robert, W., and Richard, R., "An FPGA-Based Fully Synchronized Design of a Bilateral Filter for Real-Time Image Denoising", IEEE Trans. Indust. Electron, vol. 61, no. 8, 2014.

[15] Vaibhav, Gupta., Debabrata, M., Anand, R., and Kaushik, R., “Low-Power Digital Signal Processing Using Approximate Adders”, IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 1, pp. 124 -137 January, 2013.

[16] Cong, L., Jie, H., and Fabrizio, L., "An Analytical Framework for Evaluating the Error Characteristics of Approximate Adders", IEEE Trans. On Computers, vol. 64, no. 5, pp. 1268-1281, May, 2015.

[17] Lotric, U., Bulic, P., "Applicability of approximate multipliers in hardware neural networks", Neurocomputing, (2012).

[18] Rajeevan, A., and Anantha P. C., "A Micropower Programmable DSP Using Approximate Signal Processing Based on Distributed Arithmetic", IEEE Journal of solid-state circuits, vol. 39, no. 2, pp. 337-347, February, 2004.

[19] Jeyalakshmi, D., and Bernatin, T., "Modified Approximate DCT For Image Compression Using Efficient Binary Adder", ARPN Journal of Engineering and Applied Sciences, vol. 10, no. 9, pp. 3905-3908, May, 2015.

[20] Ashish, T., Raghvendra, C., and Vijay kumar, M., "Design of Approximate Adder for Error Tolerant Application", International Journal of Emerging Technology and Advanced Engineering, vol 3, no. 6, pp. 578-582, .2013.

[21] Gupta, V., Mohapatra, D., Park, S. P., Raghunathan, A., and Roy, K., "Impact: imprecise adders for low-power approximate computing", In Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design, pp. 409–414, IEEE Press, 2011.

[22] Rajkumar, S., Malathi, G., "An Efficient Image Denoising Approach for the Recovery of Impulse Noise", Bulletin of Electrical Engineering and Informatics(BEEI), vol. 6, no. 3, pp. 281-286, 2017.

[23] Gupta, V., Mohapatra, D., Raghunathan, A., and Roy, K., “Low-power digital signal processing using approximate adders,” Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 32, no. 1, pp. 124–137, 2013.