Design of Approximate Bilateral Filters for Image Denoising on FPGA's

[1]Dr M.SHARMILA, [2]SRI SAI DINESH, [3]THUPAKULA GANGA HARIKA, [4]SHAIK SIDDIK, [5]RAMAVATHU DURGA PRASAD , [6]SHAIK ZAID HUSSAIN

[1]Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering, Tirupati, India ,
[23456]B.Tech – IV year student, Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering, Tirupati, India.

Guide: Dr M.SHARMILA, M.Tech, Ph.D.

Abstract : Image denoising is a very important process in image processing, the purpose of which is to eliminate noise without affecting important details of the image. An example of an edge-preserving denoising filter that is popular and suitable in the denoising process in that it can preserve sharp edges and minimize noise is the bilateral filter (BF). Nevertheless, the bilateral filter has a relatively high cost of computation that restricts its real-time use especially where the environment has limited resources. This paper will provide a power-efficient FPGA implementation of a bilateral filter (denoising of images) in its approximate form. The given design employs hardware-based approximations to decrease the complexity of computationally intensive processes involved in the bilateral filter, namely the range filter part. Using the paralleling nature of FPGA and optimizing access patterns in memory we will record great gains in power and processing speed without compromising image quality. Experimental evidence shows that the suggested FPGA implementation is more efficient in power consumption and execution time compared to the traditional methods based on the CPU and it is therefore applicable to real-time image processing applications in embedded systems and mobile devices. Scalability and flexibility are also made possible by the design as it can be adapted to various levels of resolution and noise. This strategy opens the avenue towards viable energy efficient image denoising in hardware based applications.

Index Terms - bilateral filter, Hardware Optimization, Low- Power, Vivado Simulation, Xilinx Vivado Design Suite Image processing.

I. INTRODUCTION

In modern digital systems, images are widely used in applications such as medical imaging, satellite communication, surveillance, and autonomous systems. However, during image acquisition, transmission, or storage, images often get corrupted by noise. This noise may arise due to several reasons such as low illumination conditions, sensor imperfections, environmental interference, or transmission errors. Image denoising is the process of removing unwanted noise from an image while preserving important features such as edges, textures, and fine details. Traditional filtering techniques like mean and Gaussian filters tend to blur the image and degrade edge information, which is not desirable for high-quality image processing. The bilateral filter is an advanced non-linear filtering technique that effectively smoothens images while preserving edges. It works by considering both spatial and intensity differences between pixels. Despite its advantages, the bilateral filter involves complex computations such as exponential functions and large kernel operations, making it computationally expensive. There has been a rising need to use efficient and high-performance digital systems hence the FPGA (Field- Programmable Gate Array) technology has been widely utilized in numerous applications such as image processing, communication systems, and embedded systems. A bilateral filter (BF) is one of the fundamental operations in the image processing used in large amounts to achieve edge-preserving denoising. The bilateral filter is a useful tool because it allows one to smooth an image and keeps sharp edges intact, thus finding use in noise reduction, object recognition, and image enhancement applications. Its computational complexity and huge resource needs, particularly in real-time processing, however, pose problems in implementing the filter in hardware platforms such as FPGA [1] [2]. On the FPGA, power consumption is a critical issue because of the heavy computations that are required to run the bilateral filter on large data sets of images. In cases where a lot of real-time high-speed processing is required, such as FPGA platforms, there is usually a high dynamic power dissipation, since the logic gates are switched on and off all the time. This is even aggravated in embedded systems where power efficiency is critical in extending battery life, minimized heat production and optimum energy usage efficiency [3]. Therefore, the development of power efficient FPGA implementations of the bilateral filter is essential in addressing the high standards of the modern image processing applications [4]. Clock gating (Selectively disabling the clock signal to the inactive parts) is one of the promising methods to handle the power consumption, basically, it aims at eliminating the unnecessary switching activities in the system by turning off the clock signal to components that are not required [5]. This power-saving method comes in especially handy in FPGA-based image-processing systems, where the image-processing system uses large numbers of flip-flops and registers to store the intermediate results in storage during image filtering operation [6]. Within the framework of the bilateral filter, these flip-flops are continuously switching particularly during real-time processing thus consuming a lot of power. Clock gating can be introduced to selectively switch off these flip-flops during other operations which are not participating in the active operations and there by useless power without impacting on the functionality or accuracy of the bilateral filter [7]. In this paper, the integration of pipeline optimized method in FPGA based bilateral filter design is researched to achieve power optimization. The proposed solution is expected to minimize switching traffic and total power leakage by selectively enabling flip-flops upon the need to perform active processing. The experiment is a comparison between the conventional bilateral filter design based on FPGA and one which implemented to evaluate the effect of power consumption ,processing speed, and resource utilization[8][9].The experimental outcomes are conducted with the help of Artix-7 FPGA platform that has low power consumption and high performance that can be used successfully to save power without deteriorating the performance of the bilateral filter..Though FPGA devices are not as endowed as the other devices, the pipelining techniques will ensure that any methods used to save power are utilized without wasting the resources [10].

II. RELATED WORK

Image processing systems based on FPGA, and especially power efficiency, have become a critical field of research in recent years with the rise in demands of energy-efficient embedded systems. Numerous researches have been on different approaches to lower power consumption of FPGA- based designs, particularly computationally intensive image processing, such as bilateral filtering[11].The bilateral filter is widely applied in the automatic processing of digital images because it allows removing noise and the edges of the image are not distorted. A number of FPGA-based implementations of bilateral filters have been offered, with the initial efforts being on optimization of the hardware implementation to enhance the processing speed and reduce the usage of resources. The designs use parallelism methods usually to get high throughput but they fail to put into consideration the very crucial factor of power consumption. A comparative study of bilateral filter variants to filter medical imaging where the computational cost of bilateral filters is very high in medical image enhancement but power consumption is not explicitly optimized by FPGA [12]. With the increased adoption of FPGA-based design researchers started concentrating on the power efficiency versus speed. Application of fast bilateral filters with arbitrary range and domain kernels, which offer a solution to the issue of increasing speed of processing without compromising on performance [13]. This design however failed to deal with the power dissipation problems of high- speed operations. FPGA designs with many flip-flops and many registers to store intermediate values during filtering operations consume a lot of power particularly in real-time image processing [14]. Although it was faster in processing power efficiency was still an issue because a lot of resources were consumed in such high-performance designs. Pipelining techniques along with clock gating is one of the useful methods to lower power consumption in the FPGA design and it is a technique that enables the clock signal to be disenabled to inactive circuit components [15]. Clock gating greatly minimizes dynamic power dissipation by eliminating needless switching of flip- flops and registers. This paper presents the sensitivity of using a combination of approximate techniques and optimize the bilateral filters of FPGA in-terms of power consumption [16]. Dynamic clock gating application on the implementation of bilateral filters in FPGA based real time image processing systems. They proposed an intelligent clock-gating mechanism that adjusted to the nature of the input signal of the bilateral filter, where the clock was turned off when the signal was not needed. This method had the benefit of consuming a lot less power without affecting the performance of filtering. Their experiment showed that they could save up to 30 percent of their power consumption, which has made it one of the possible solutions to real-time image processing problems where speed and power efficiency are highly important [17] [18]. A small hardware design of the bilateral filter that integrates approximate computing with a look up table. This method resulted in saving of large amounts of power since the number of active components was reduced when less computationally intensive tasks were being performed without reducing the performance of the filter [19]. Their effort demonstrated that pipelining technique was able to result in high power savings, particularly in high performance design where dynamic power dissipation was of interest. The work by Robertson was a foundation of the future research in the areas of designing power-efficient FPGA-based solutions, especially in signal processing such as image filtering [20].

III. EXISTING MRTHODOLOGY

Bilateral filter is a non-linear image processing method, which is popular in edge-saving smoothing and grass removal. Contrary to the normal linear filters, it is based on the consideration of the spatial distance between the pixels and the differences in intensities as well. The filter averages the intensity value of the neighboring pixels, however, it weighs them considering their spatial closeness and resemblance to the central pixel in terms of intensity.

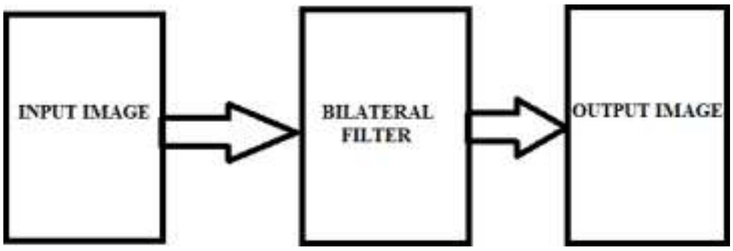

Fig.1.Existing Block Diagram

This is the method that can maintain sharp edges even though it is also effective in reducing noise levels, hence especially in those applications where sharp edges are essential, including medical imaging, computer vision, and graphics. Mathematically, the bilateral filter implements two Gaussian functions, one of which uses the distance between pixels, and the other one uses the difference between the intensity of pixels. The spatial component makes sure that pixel close to the center pixel have a greater impact whereas the range component makes sure that the pixels with similar intensity have greater effects on the average. The two-fold consideration of this enables the filter to maintain significant features such as edges, and to eliminate noise in homogeneous parts of the image. The outcome is that, the details are maintained and noise has been reduced successfully without distorting the edges.

Fig.2.Base Filter Internal RTL Schematic Diagram

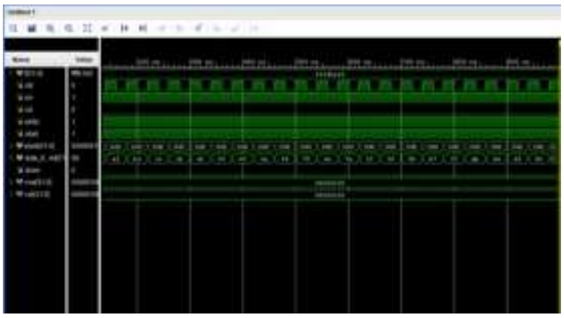

Fig.3.Base Filter Simulation Waveform

This is expensive in computation ,as the filter has to perform spatial and intensity kernels of each pixel and its neighbors, and this turns out to be a large number of calculations. Hence, although bilateral filter is a great tool in activities where an edge must be maintained in an improved image, it cannot always be used in real-time processing unless an optimized version is implemented. Bilateral filter has been highly valued in its ability to smooth an image and at the same time retain the edges, and it is a powerful tool used in image enhancement and denoising tasks. The thing is that its preserving edge quality is provided with the help of paying attention to the spatial distance and the intensity contrast among pixels. Such a property is especially useful in applications in which sharp edges are important and in which the aesthetic appearance is essential like in medical imaging, computer vision, and artistic effects. This makes its computational cost higher, particularly when large images are being operated on or in real-time uses. In an attempt to reduce the computational cost a number of optimized versions of the bilateral filter have been suggested. Such methods consist of methods such as fast bilateral filtering which approximates calculations of the filter by use of less efficient data structures or simplifying range and spatial kernels. The other optimization technique is hardware accelerations or GPUs or FPGAs which can use parallel processing to accelerate the bilateral filtering process.

IV. PROPOSED METHODOLOGY

The proposed system focuses on designing an approximate bilateral filter for image denoising using FPGA architecture. The bilateral filter is widely used for image denoising because it preserves edges while smoothing noisy regions. However, conventional bilateral filters require complex arithmetic operations, exponential functions, and large computational resources. To overcome these limitations, the proposed method uses approximate computing and pipeline optimization techniques. The proposed architecture reduces computational complexity while maintaining acceptable image quality. Approximation techniques simplify mathematical calculations and reduce hardware resource usage. Pipeline optimization improves throughput and processing speed, making the system suitable for real-time image processing applications. The proposed system consists of the following main stages:

1. Input Image Acquisition
2. Pre-processing Stage
3. Approximate Bilateral Filtering
4. Pipeline Optimization
5. Output Image Generation

These stages work together to provide efficient image denoising with reduced hardware complexity and improved performance.

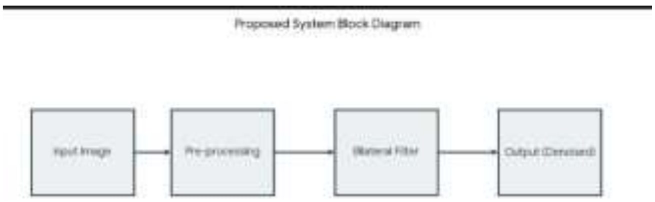

Fig.4:Proposed design

The bilateral filter is well known for its ability to preserve edges while removing noise; however, its exact implementation involves complex exponential operations and extensive multiplications. To overcome these challenges, the proposed approach introduces approximate computing techniques into the bilateral filtering process. The key idea is to simplify the computation of spatial and range kernels without significantly degrading image quality. By approximating Gaussian functions and optimizing arithmetic operations, the proposed design achieves a balance between denoising performance and hardware efficiency. The bilateral filter output for each pixel is computed using a weighted sum of neighbouring pixels, where the weights depend on both spatial distance and intensity difference.

In the proposed method:

- The spatial kernel is pre-computed and stored using fixed-point representation.
- The range kernel is approximated using simplified arithmetic operations, avoiding complex exponential calculations.
- Low-impact computations with negligible contribution to the final result are selectively reduced or eliminated.

This approximation significantly reduces the number of multiplications and memory accesses required during filtering.

A. Input Image Processing

The first stage of the proposed method involves input image acquisition. The input image may contain noise due to sensor limitations, environmental conditions, or transmission errors. The image is first converted into gray scale format for easier processing. This step reduces computational complexity and simplifies hardware implementation.

Pre-processing steps include:-

1. Image resizing
2. Gray scale conversion
3. Noise detection
4. Pixel normalization

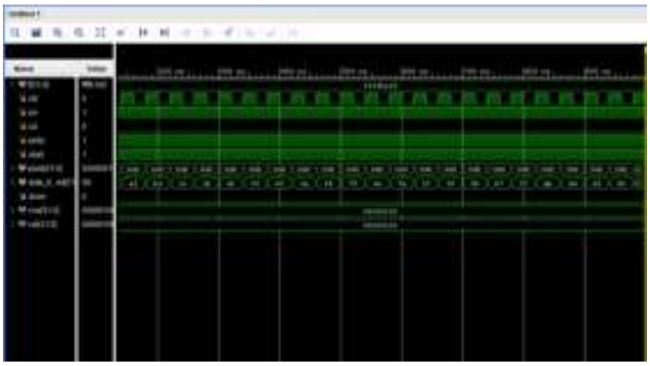

Fig.5.Proposed Pipeline - optimized Based Simulation Waveform

B. Approximate Bilateral Filtering

The bilateral filter is used to remove noise while preserving image edges. The bilateral filter considers both spatial distance and intensity difference between pixels. The approximate bilateral filter simplifies complex exponential functions using approximation techniques.

The bilateral filter operation includes:

- Spatial kernel calculation
- Range kernel calculation
- Weight computation
- Pixel intensity update

Approximation techniques reduce the complexity of experimental calculations and reduce hardware utilization

C. Pipeline Optimization Architecture

Pipeline optimization improves processing speed by dividing the filtering operation into multiple stages. Each stage performs part of the filtering operation. This allows parallel processing and improves throughput.

Pipeline stages include:

- Input stage
- Kernel computation stage
- Weight calculation stage
- Pixel update stage
- Output stage

Pipeline architect' increases processing speed & reduces latency. The main subunit of the system is the Filter Processing Unit in which the actual filtering operations are found. This unit contains several examples of Filter 1 blocks, each of them depicting a separate filter stage. The fundamental signal processing operations, e.g. convolution, smoothing, or noise removal, are done by these filters. With the help of multiple filter stages, the system can perform various kinds of filtering operations consecutively until the required output is achieved. Digital multipliers and adders generate part products and have to be summed up and finally added.

Fig.6.Proposed Implemented Algorithm

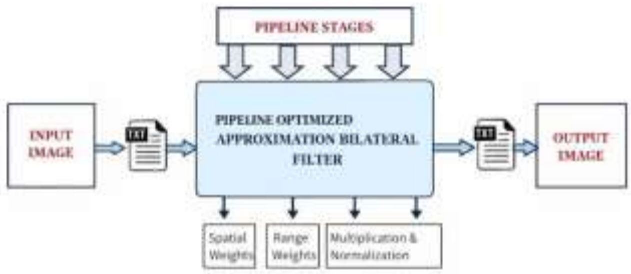

Fig.7.Proposed Block Diagram

This is broken down into two key phases, image preprocessing and filtering with Verilog to do the bilateral filter implementation and finally post-processing to restore the final image that is denoised. The image made noisy is first turned into digital form in MATLAB where pixel data is taken out and worked with on an FPGA board using a Verilog based bilateral filter. Depending on how the system is designed, the data of the noisy picture is sent over to the FPGA either using serial communication of a parallel communication interface. The hardware description of the bilateral filter in this case is being done in Verilog. All the filter tap coefficients are preset according to the required filtering properties. The system will make sure that the FPGA will enable the sections of the filter processing unit that are required to perform the calculation in question only, thus consuming significantly fewer power. The bilateral filter is used as follows: the values of the incoming pixels are processed according to their spatial distance and contrast with their neighboring pixels. The modified method also means that only a set of registers and multipliers are needed at any point at any point in time within a filter operation, resulting in improved efficiency.

V. RESULTS AND DISCUSSION

A synthesis and simulation of the proposed efficient power FPGA design of bilateral filter of the image denoising process were performed in the Xilinx Vivado Design Suite 2020.2. The idea behind the hardware implementation was to use the Xilinx Artix-7 FPGA (xc7a100t-1csg324) which is a common high-performance platform characterized by power-efficient digital systems. The design aimed on maximizing computationally heavy operations of bilateral filter especially the range filter aspect to gain energy efficient performance that can be used in real time application. The paralleling processing of the FPGA was used to accelerate the functions, and optimization of access patterns to the memory assisted to minimize power consumption which made it appropriate to embedded systems and mobile devices. The results indicated that a great power savings was achieved at the same time preserving the quality of the image processing, and hence the filter can be used effectively even with large images and in resource limited settings. Besides, the design is scalable and flexible, and is suitable to a wide range of image resolutions and noise levels, and opens the way to viable, energy efficient image denoising in hardware-accelerated applications.

TABLE.1:COMPARISON OF EXISTING AND PROPOSED WORK IN PARAMETERS

Method	**Area**	**Delay(ns)**	**Power(W)**
Existing method	1332	19.25	26.338
Proposed method	1330	19.1	2.052

It is a table comparing the bilateral filters that are currently used and the proposed ones based on the area, delay, and power. The suggested method has 1130 area units that is nearly similar to the current method with 1132 units. The delay of both techniques is identical (19.2ns), and shows no timing deterioration. There is a major enhancement in power consumption.

Fig.8.Input Sample Image 1(B) Output Image1(C) Input Sample Image 2 (D) Output Image

In the initial two images (A → B), there is granular noise that can be observed all over the surface of the input image. The result of the bilateral filter (B) has a lot of noise reduction and retained the edges and object shapes. The filter blurs continuous areas but does not blur edges and this is clearly visible in the preserved sharpness of the features of circles and structure. The performance measures voice PSNR = 19.5990 dB SNR = 13.9549 dB, and show that the signal quality was improved over the noisy input. A similar effect is witnessed in the second pair (C→D). The high frequency texture in the input image (C) is lowered in the output (D) that is filtered. Linear patterns and detail elements can also be seen and it is seen that the bilateral filter is effective in balancing noise minimization and edge conservation. The values of PSNR = 19.2602 dB and SNR = 14.5975 dB reported prove that the filtering process did not overly smooth the images and it provided an improvement in the picture quality.

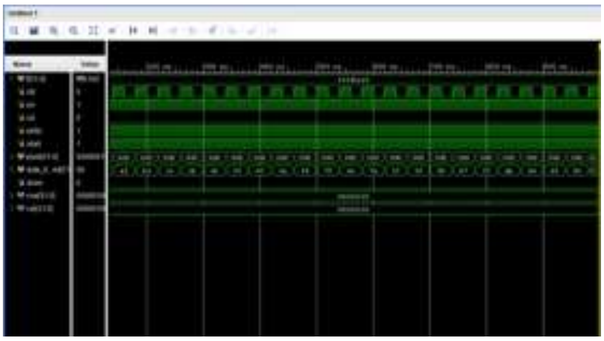

Fig.9.Simulation waveform

Fig:9 image presents the output of the bilateral filter design simulation waveform in case of functioning verification. Some of the signals that are found in the waveform are clock, write, start, pixel input data, and output signals. Based on the timing diagram, the clock signal is running in an uninterrupted manner and this implies a synchronous work mode of the filter. The write and start control signals are asserted to start the process of filtering. The values on the pixel input are fed one at a time as observed in the evolving values of the data as time goes by. Following a processing, the processed signal is stabilized to a filtered signal (in the waveform 256). The done signal is used to signal the end of the filtering process, and is used to show that the bilateral filtering logic was properly implemented.

Name	Slice LUTs (134600)	Slice Registers (269200)	F7 Muxes (67300)	Slice (33650)	LUT as Logic (134600)	LUT Flip Flop Pairs (134600)	Bonded IOB (400)	BUFGCTRL (32)
Bilateral_Filter	1332	8	34	370	1332	8	178	1

Fig.10.Area

Fig: 10 indicates resource usage of your Bilateral Filter design when synthesized in the FPGA hardware. It does not actually display the filtered image, but instead the amount of FPGA area that one of your designs occupies, and the type of hardware blocks that you are using. Your design has 1332 Slice LUTs and that implies that there is high level of combinational logic needed. This reasoning is primarily based on comparing pixels, arithmetic, and making decisions that are required in the bilateral filter. The use of LUT is an indication that your implementation is quite efficient and not too complex. The design employs 34 F7 multiplexers indicating that conditional choices are present in the design, e.g. between pixel values or dealing with boundary conditions in the filter window. Bonded IOBs are 178, and this implies that numerous input and output pins are utilized. This tends to occur when pixel information and control signals are attached directly to the pins of FPGA rather than being stored and read via internal memory such as BRAM.

Fig.11.Delay

Fig:11 overall delay is approximately 19.2 ns on the majority of paths. This implies that the highest safe clock rate is approximately 52.08MHz (1/ 19.2 ns). Any attempt to run the design even faster than this can result in timing violations. High value of fan out 33 depicts that many logic blocks are driven by the signal pixel [2] concurrently. Great fan out upsurges latency and is typical in picture-processing designs where a single value of the pixel is utilized in numerous calculations.

Total On-Chip Power:	2.189 W
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	29.1°C

Fig:12 Total on-chip power

The output of this Fig:13 indicates the power and thermal analysis of your bilateral filter FPGA design. The overall power on-chip is 2.189 W and of this is the dynamic power (switching activity in logic, routing, and I/O) and the static power (leakage). It is an indication that your design is relatively power-intensive, probably as a result of the high switching activity in the bilateral filter calculations and the large number of I/O signals. The definition of power budget is useful to assess the fact of tenure of whether the design would be safe and efficient. The cross-over temperature is 29.1 o C and that is well within the safe operating temperatures of an FPGA. This is to say that, despite the significant power consumption, the device is not overheating which is why the thermal conditions are stable.

VI. CONCLUSION

The suggested power-efficient FPGA implementation of the bilateral filter contributes in a significant manner to the viability of the real-time image denoising software and its use in the context of resource-limited settings. The design ensures that a significant power reduction is realized and a significant power-consumption controlled by the bilateral filter is specifically the range filter component and, by taking advantage of the parallel processing capabilities of FPGA, the operation of the bilateral filter is optimized to reduce power consumption without affecting the quality of the image. The experimental data confirms the fact that FPGA implementation even with a simple case of power consumption as well as running speed beats the traditional methods that are based on the CPU and therefore should be used in the embedded systems and mobile devices. Moreover, the design is scaled and flexible in its implementation, which improves its applicability in practice by multiple real-time image processing applications of different noise levels and image resolutions. The strategy has enabled the use of hardware-accelerated image denoising solutions, which are energy-efficient, to be contemplated, thus offering a way forward to more developments in FPGA-based image processing systems.

Acknowledgment

The authors would like to express their sincere gratitude to Dr. M. Sharmila, Department of Electronics and Communication Engineering , Sri Venkateswara College of Engineering for her valuable guidance, encouragement and continuous support throughout the development of this project . The authors also extend their appreciation to the faculty members of Department of ECE, for providing the necessary academic resources and support required for completing the project. Finally, the authors would like to thank their institution for providing the infrastructure and learning environment that enabled the successful completion of this project.

VII. REFERENCES

[1] P.K.R.Maddikunta,Q.V.Pham, B.Prabadevi, N.Deepa,and K.Dev,''Industry 5.0: A survey on enabling technologies and potential applications,'' J. Ind. Inf. Integr., vol. 26, pp. 1–31, Mar. 2022.

[2] J. Nakamura, Image Sensors and Signal Processing for Digital Still Cameras .New York, NY ,USA : Taylor & Francis, 2006, ch.3, pp.55–90.

[3] D.-H. Shin, R.-H. Park, S. Yang, and J.-H. Jung, ''Block-based noise estimation using adaptive Gaussian filtering,'' IEEE Trans. Consum.Electron., vol. 51, no. 1, pp. 218–226, Feb. 2005.

[4] C.Liu,W.T.Freeman,R.Szeliski, and S.B.Kang,''Noiseestimationfrom a single image,'' in Proc. IEEE CVPR, Jun. 2006, pp. 901–908.

[5] B.Y.Kong and I.C.Park, "Improved Sorting Architecture for K-Best MIMO Detection," IEEE Trans Circuits Syst. II, Exp. Briefs,vol.64,no. 9, pp. 1042–1046, Sep. 2017.

[6] N. Govindaraju, J. Gray, R. Kumar, and D. Manocha, "GPUT era Sort: High-Performance Graphics Co-Processor Sorting for Large Database Management," in Proc. ACM SIGMOD Int. Conf. Manag. Data, Jun.2006, pp. 325–336.

[7] J. Casper and K. Olukotun, "Hardware Acceleration Of Database Operations,"in Proc.ACM/SIGDA FPGA, Monterey,CA, USA, 2014, pp.151–160.

[8] H. Inoue, T. Moriyama, H. Komatsu, and T. Nakatani, "AA-Sort: A New Parallel Sorting Algorithm for Multi-Core SIMD Processors,"inProc.16thInt.Conf.ParallelArchit.CompilationTechn.(PACT),Sep.2007, pp. 189–198.

[9] D.Merrill and A. Grimshaw, "High Performance and Scalable Radix Sorting: A Case Study of Implementing Dynamic Parallelism for GPU Computing," Parallel Process. Lett., vol. 21, no. 2, pp. 245–272, Jun.2011.

[10] A. Davidson, D. Tarjan, M. Garland, and J. D. Owens, "EfficientParallel Merge Sort for Fixed and Variable Length Keys," in Proc.Innov. Parallel Comput. (InPar), 2012, pp. 1–9.

[11] J. Casper and K. Olukotun, "Hardware Acceleration of Database Operations, "inProc.ACM/SIGDAFPGA,Monterey,CA,USA,2014, pp.151–160.

[12] J. Chhugani, A. D. Nguyen, V. W. Lee, W. Macy, M. Hagog, Y.-K.Chen, A.Baransi, S.Kumar, and P.Dubey, "Efficient Implementation of Sorting on Multi-Core SIMD CPU Architecture," Proc. VLDBEndowment, vol. 1, no. 2, pp. 1313–1324, 2008.

[13] R. Marcelino, H. C. Neto, and J. M. Cardoso, "Unbalanced FIFO Sorting for FPGA-Based Systems," in 2009 16th IEEEICECS, IEEE,2009, pp. 431–434.

[14] D. Koch and J. Torresen, "FPGA-Sort: A High-Performance Sorting Architecture Exploiting Run-Time Reconfiguration on FPGAs for Large Problem Sorting," in Proc. 19th ACM/SIGDA International Symposium on FPGAs, ACM, 2011, pp. 45–54.

[15]R.Mueller, J.Teubner, and G.Alonso,"Sorting Networks on FPGAs, "VLDBJ.-Int.J.Very Large Data Bases ,vol.21,no.1,pp.1–23,2012.

[16] D. Merrill and A. Grimshaw, "High Performance and Scalable Radix Sorting :A Case Study of Implementing Dynamic Parallelism for GPU Computing,"Parallel Process.Lett.,vol.21,no.2,pp.245–272,2011.

[17] A. Davidson, D. Tarjan, M. Garland, and J . D. Owens, "EfficientParallel Merge Sort for Fixed and Variable Length Keys," in 2012Innov. Parallel Comput. (InPar), IEEE, 2012, pp. 1–9.

[18] N.Tsuda,T.Satoh,and T.Kawada,"APipelineSortingChip,"in1987IEEE Int. Solid-State CircuitsConf. Dig. Technical Pap., vol. 30, IEEE,1987, pp. 270–271.

[19] B.Y.Kong,H.Yoo,andI.-C.Park,"Efficient Sorting Architecture for Successive-Cancellation-List Decoding of Polar Codes," IEEE Trans.Circuits Syst. II, Exp. Briefs, vol. 63, no. 7, pp. 673–677, 2016.H.Jiang, F. J. H. Santiago, H. Mo, L. Liu and J. Han, "Approximate arithmetic circuits:A survey characterization and recent applications",Proc. IEEE, vol. 108, no. 12, pp. 2108-2135, Dec. 2020.

[20] S. Saha Ray and S. Ghosh, "K-degree parallel comparison-freehardware sorter for complete sorting," IEEE Trans. Comput.-Aided Design Integr.CircuitsSyst.,vol.42,no.5,pp.1438–1449,May 2023.