International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 10, No. 3, November 2021, pp.212~220
ISSN: 2089-4864, DOI: 10.11591/ijres.v10.i3.pp212-220

Enhanced MAC controller architecture for 2D processing based on FPGA with configurable resource allocation

Chiranjeevi G. N., Subhash Kulkarni
Department of Electronics and Communication Engineering, PESIT-Bangalore South Campus, Karnataka, India

Article Info

Article history:

Received Jun 20, 2021
Revised Jul 27, 2021
Accepted Aug 11, 2021

Keywords:

2D convolution
Image enhancement
Image processing FPGA
MAC
Xilinx Vertex
Zynq7000 SOC

ABSTRACT

The bulks of image processing algorithms are either two-dimensional (2D) or confined by their very nature. As a result, the 2D convolution function has a large impact on picture processing requirements. The methodology of 2D convolution and media access control (MAC) design can also be used to perform a variety of image processing tasks, and even as picture blurring, softening, and feature extraction. The main goal of this research is to develop a more efficient MAC control block-based 2D convolution architecture. This 2D algorithm can be implemented in hardware using fewer modules, multipliers, adders, and control blocks, resulting in significant hardware savings and look up table (LUT) reductions. The simulations were run in Verilog, and the Xilinx Vertex family field programmable gate array (FPGA) was used to build and test them. The recommended 2D convolution architectural solution is significantly faster and consumes significantly less hardware resources than the traditional 2D convolution implementation. The proposed architecture will result in technology that saves a substantial amount of processing time when it comes to LUTs.

Corresponding Author:

Chiranjeevi G. N.
Department of Electronics and communication Engineering
PESIT-Bangalore South Campus
Affiliated to Visveswaraya Technological University, Belgaum
Bangalore, Karnataka, India
Email: chiranjeevign@pes.edu

1. INTRODUCTION

In the industry, field programmable gate array (FPGA) is commonly utilized for testing. And also has a wide range of applications such as DSP applications, medical field [1], and close observation like surveillance. Three key applications drive the development of digital image processing technology. The very first function was to promote better interpretation of input images. As a result, every image that intends to improve the quality of the image is implying that the picture wishes to improve the characteristics of the input image. As a result of which the image will have a better appearance than the original. The autonomous machine is the second key use of digital image processing technology. The efficient storage and transmission of data is a third use. If the image is stored on a computer, it will necessitate the storing of a specific amount of data. In order to limit the quantity of data space necessary to save the photographs otherwise, additional memory will be required to store the photos. Two-dimensional (2D) convolution is used in image and video processing applications to cover a broad area. Picture smoothing, image sharpening, and edge detection all require the use of the 2D convolution block in digital image processing [2].

MAC design is among the most important operations (like Edge Detection) [3], [4] in image processing applications and algorithms. It could work with various values of kernel based on user

requirement and constraints, allowing for different voice, image processing applications respectively. Convolution and MAC design is an image retrieval approach extensively employed in image analysis and recursive sum value, as well as for biomedical devices [5]. As a result, the optimal trade-off between area (reduced LUTs), speed, and specific strength is desired. On the Xilinx Vertex FPGA [6] platform, a 2D convolution framework is proposed to be implemented. When developing image processing techniques with memory components [7], it is crucial for embedded systems to even have low power consumption and high throughput [8]. There are various steps to the implementation process [9]. The target platform is the Vertex FPGA and higher versions. Every pixel value, such as hue, brightness, and so on, has its own meaning [10]. Data is kept in a matrix format. The first step in acquiring a picture out of any input is to transcribe that into image pixels and store it inside this matrix. MAC operations (viz, reconstruction and regression) [11] on those matrices are then executed using a kernel selected by the user.

The remainder of this work is arranged in the following manner. The design methodology for convolution operator is introduced in Section 2. The proposed architecture and hardware implementation for MAC design as convolution operator are discussed in Section 3. Section 4 of this paper explains the computational results, the simulation data is discussed in Section 5, tracking systems and performance evaluation parameters are discussed in Section 6, and the article's conclusion is discussed in Section 7.

2. DESIGN METHODOLOGY

Image processing is useful in a wide range of situations [12], [13]. For particular, whenever an image is sent or acquired, or when an image is compressed, noise signals are easily injected into the original image. As a result, reducing the disturbances in the original image necessitates an additional step [14]. The technique [15], [16] of image filters is critical in digital image processing. A kernel, which is a tiny array applied to each neighboring pixels in the image, can be used to define a filter. The kernels centric are usually aligned with the current pixel throughout many applications. The final image is obtained by convolving an image with the kernel. Kernel matrices play a significant part in the convolution idea, different kernel matrices produce distinct image resultants. Blurring, smoothing, contrast enhancement, and other operations based on convolution can all be accomplished depending upon on kernel matrices selected [17], [18].

The Image pixel principle stating I(x, y), the kernel convolution measure shows K(x, y), and the convolution resulting for the frame O(x, y). The convolution technique is depicted in image kernel matrices are regarded 3*3 in Figure 1, as well as image convolution the resultant matrices convolution is provided by 4*4 matrices. The resultant of the convolution output values [19], [20] is substituted for the pixels' original values, and the resultant array's portions, which are the same size as the kernel, were ultimately averaged. The process will be repeated, with the convolution [21], [22] of the generated values replacing the entire picture in the matrices, pixel values. The convolution pixel value resultant should be 0 whenever the kernel pixel value is outside the matrix. The block is fed with the core pixel values to be analyzed, as well as its neighboring pixels. The convolution operator is used in real-time applications for various edge detection algorithms [23], [24] and also edge filter for video processing system on reconfigurable platform [25], [26].

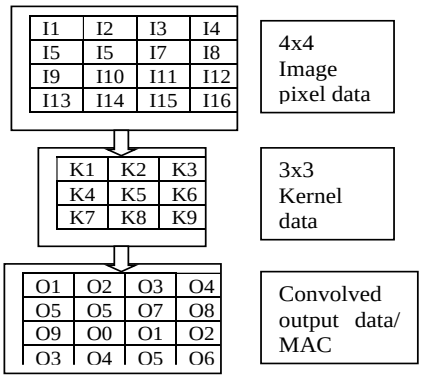

Figure 1. Matrices of the image and the kernel are convolved

Figure 1 depicts the convolution of the picture with kernel matrices, where the picture represents 4x4 matrices, the kernel matrices represent 3x3 and the final picture represents 4x4. The convolution process should be conducted as shown in the equation. The recursive multiplication and addition operators to derive all of the outputs

of convolve data. Get the output convolution resultant (O1 to O16) values based on this. The technique is repeated, other values are discovered. When performing the convolution operation, the mean multiplication and additions rise, implying that the propagation latency increases as well. As a result, create an efficient architecture for 2D convolution. The convolution of the picture and the kernel matrices is shown in Figure1, where the picture represents 4x4 matrices, the kernel matrices represent 3x3, and the final picture represents 4x4. The convolution process should be conducted as shown in the equation.

$$O10 = (I5 \times k1) + (I6 \times K2) + (I7 \times k3) + (I9 \times k4) + (I10 \times k5) + (I11 \times k6) + (I13 \times k7) + (I14 \times k8) + (I15 \times k9).$$

Get the O10 convolution resultant values based on this. The technique is repeated, other values are discovered. When performing the convolution operation, the mean multiplication and additions rise, implying that the propagation latency increases as well. As a result, create an efficient architecture for 2D convolution.

3. 2D CONVOLUTION ARCHITECTURE AND MAC HARDWARE IMPLEMENTATION DESIGN

Multiplication and summation will be performed by the model, which is necessary for image processing applications. Essentially, this approach obtains the pixels and kernel values, then activates the multiplier pixels with kernel and adds them together. Different kernels, such as identification, edge detection, sharpen, box blur, Gaussian blur 3x3, and so on, must be chosen based on user preferences.

Blur = 1/9 [1, 1, 1: 1, 1, 1: 1, 1, 1]

I.e. each pixel is multiplied by a factor of one. We totaled all of a pixel's eight neighbors and divided the result by nine. Alternatively, you might use a smoothing effect or the average of one pixel. Box blur is used as a case study for our MAC hardware implementation. Figure 2 depicts the MAC unit's implementation in the Matlab design pipeline.

In FPGA, the same mechanism is used. Assuming row-by-row data from the preprocessing unit. For MAC operation, 24 bit per row and a total of three rows of data are regarded input pixels data. Finally, 72 bit pixel data is handled as system data at each clock cycle. Figure 3 depicts the external view of 2D MAC unit with external input output pin access.

An image kernel is a tiny matrix that can be used to perform effects like blurring, sharpening, outlining, and embossing, similar to those seen in Photoshop. They're also employed in a process called 'feature extraction,' which determines the most essential parts of an image. In this context, the process is referred to as "convolution" in a broader sense. To begin, a nine-element kernel is chosen, with each value having an 8-bit wide representation at most. To fit this into memory, a two-dimensional array is created using the following declaration. Figure 4 depicts the 2D array declaration for defineing the width and number of locations for kernel register.

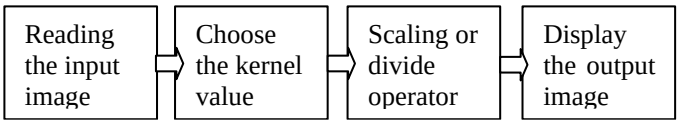

Figure 2. Implementation of MAC hardware in MATLAB

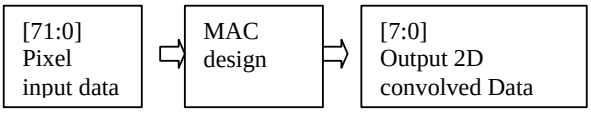

Figure 3. External view of the 2D MAC implementation design

```
Reg [7:0] kernel [8:0]
```

Figure 4. 2D array declaration for kernel

This begins the process of assigning or writing the nine values to kernel memory. Because all of the elements in our case study have the same value, we can use a recursive technique to assign a common value to all of the positions in the 2D array or memory. In the proposed design, this is task 1 of the pipeline stage.. Figure 5 depicts the assignment of nine functional values to the kernel register.

The very next activity is multiplication (task 2 of pipeline stage un suggested architecture), which entails retrieving all nine values from a 2D array memory and multiplying them with the incoming pixel data. Tasks 1 and 2 are seen to be multitasking and executed in pipeline mode. Figure 6 depicts the activation of multiplication operator for all nine values with the input pixel data.

```
Always@ (posedge (clk))
For (i=0;i<9;i=i+1)
Begin
Kernel[i] =1;
End
```

Figure 5. 2D array declaration for kernel with values assigned as a case study of "blur" kernel 9 values

```
Always@ (posedge (clk))
for (i=0;i<9;i=i+1)
Begin
Multiplied_data<= Kernel[i]* input pixel data
[i*+:8];
End
```

Figure 6. Multiplication operation as a case study of "blur" kernel 9 values

To generate the output convolved resulting pixels, the next operation is addition (add them all together). Figure 7 depicts the successive addition operation for the multiplied data.

```
Always@ (posedge (clk))
for (i=0;i<9;i=i+1)
Begin
sum_data<= sum_data+multiplied_data[i];
End
```

Figure 7. Addition operation as a case study of "blur" kernel 9 values

The matrix is divided with a value of [1/9] at the final stage of MAC design, which is indicated in Figure 8

```
Always@ (posedge (clk))
Begin
Convolved_output<= Sum_data/9;
End
```

Figure 8. Division operation as a case study of "blur" kernel 9 values

4. RESULTS AND DISCUSSION

The RTL schematic, which is the transformation of an HDL description together into formally digital circuit schematic, or in plain terms, a net list, is shown in Figure 9. Figure 10 shows a look up table-based technological schematic. It is a representation of a net list superimposed over the structural properties of a specific device (Vertex board).

Figure 11 depicts the comprehensive components of a report that were used in the synthesis. The area utilization report in terms of inside structural components like logic, registers and mux which is shown in Figure 12. The primary finding is that selecting the vertex as a target consumes less than 1% of the total available resource on the target device. The power consumption of the capacity in the targeted system is depicted in Figure 13.

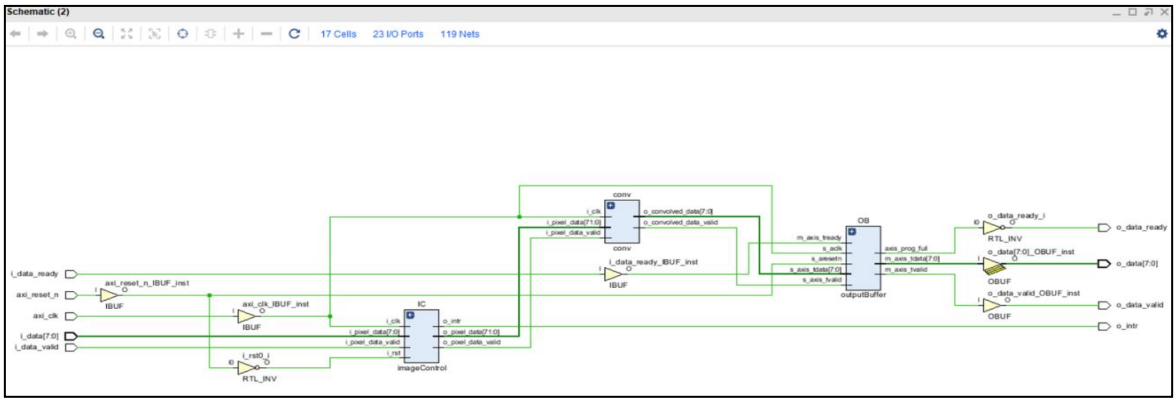

Figure 9. RTL schematic

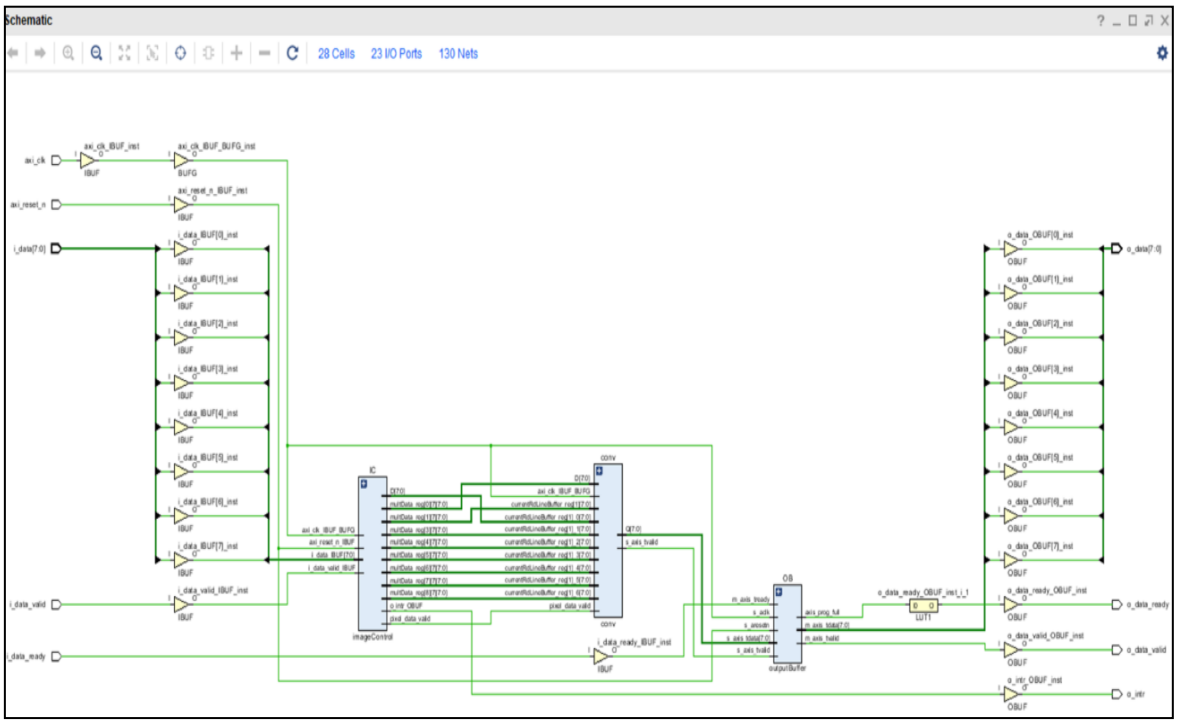

Figure 10. Technologic schematic analysis in terms of LUT's

```
Detailed RTL Component Info :
+---Adders :
      9 Input     12 Bit       Adders := 1
      2 Input      9 Bit       Adders := 18
      2 Input      2 Bit       Adders := 2
+---Registers :
                  12 Bit    Registers := 1
                   9 Bit    Registers := 10
                   8 Bit    Registers := 10
                   2 Bit    Registers := 2
                   1 Bit    Registers := 5
+---Muxes :
      4 Input     72 Bit         Muxes := 1
      4 Input      4 Bit         Muxes := 1
      2 Input      1 Bit         Muxes := 4
      3 Input      1 Bit         Muxes := 1
```

Figure 11. Detailed RTL component information

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	1647	0	53200	3.10
LUT as Logic	494	0	53200	0.93
LUT as Memory	1153	0	17400	6.63
LUT as Distributed RAM	1152	0		
LUT as Shift Register	1	0		
Slice Registers	201	0	106400	0.19
Register as Flip Flop	201	0	106400	0.19
Register as Latch	0	0	106400	0.00
F7 Muxes	96	0	26600	0.36
F8 Muxes	0	0	13300	0.00

Figure 12. Report on the use of space components inside target device

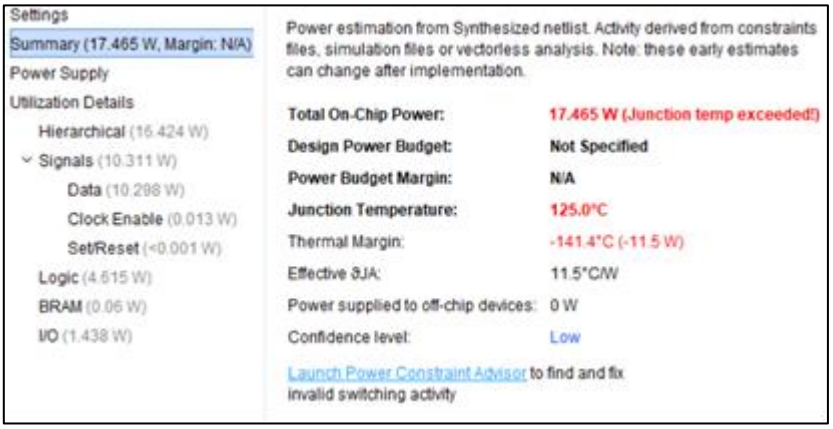

Figure 13. Report on energy consumption

Figure 14 shows the complete on-chip power data on dynamic and static power dissipation. The essential point to note is that the power slot is split into two categories: dynamic and static. Where the dynamic power consumption is 16.24W (94%) and the static power consumption is 1.041W (which is 6 percent). Signals and logic are responsible for the device's dynamic power usage. I/O and RAM are blocked.

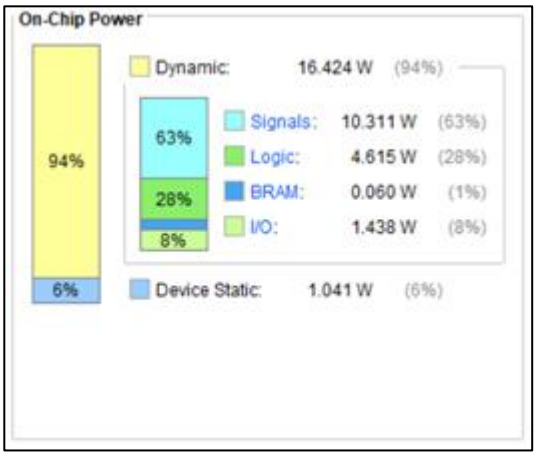

Figure 14. Report on on-chip power

Multiple test cases are produced by using test bench, validated and used the vivado tool, then hardware synthesis is triggered for various testing vectors by using test bench application, and also the architecture is generated and used in the Verilog building design. For comparison purposes, a few characteristics such as power, area, and efficiency can be extracted from those in the synthesis results.

The reporting system for synthesis and execution is shown in Figure 15 and Figure 16. In terms of the number of LUTs utilized for the algorithms, the reconfigurable internal components are employed 39 instances.For synchronization mostly with external clock, yet another flip-flop is used.

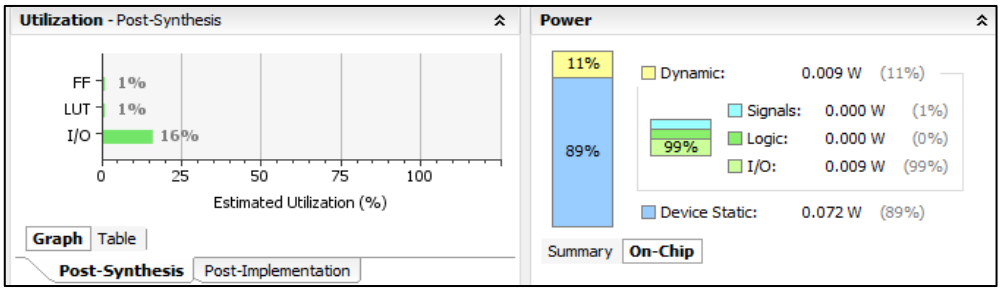

Figure 15. Report on the synthesis for the target device

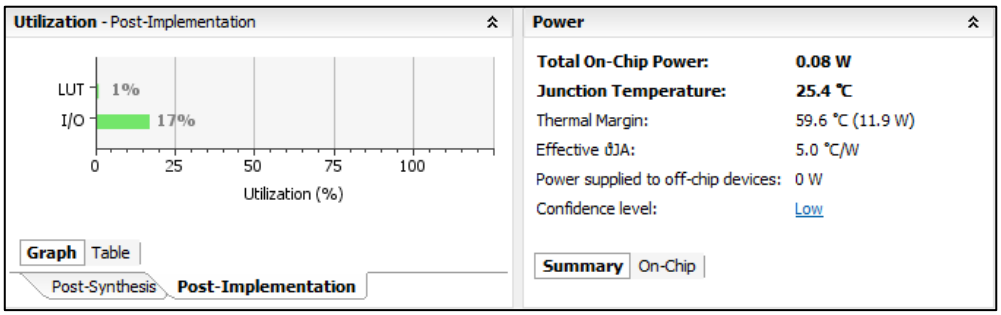

Figure 16. Report on the Results of the Implementation

5. CONCLUSION

This paper discusses the hardware implementation of MAC design with convolution operator in Verilog Platform. The suggested hardware design is modelled in Verilog and synthesized using an FPGA device with target device as vertex and higher versions SOC. When compared to conventional architecture, critical path time is lowered more in this device and it also uses less slices. The propagation delay was determined to be 5.030 ns for a picture of an order of 16*16 resolutions. When the image resolution is 720*480 or 600*560, the delay will be longer. The resulting convolution values are stored in memory when the convolution procedure is completed. This paper predominantly describes a new architecture for the MAC implementation with a convolution operator is proposed. With only a few mux, ALU blocks for multiplication, addition, division, and control logic blocks for generation of control signals in each stage of pipeline stage, it is a basic and efficient hardware implementation with pipeline techniques for reconfigurable platform and helps in reducing the critical path delay.

ACKNOWLEDGMENT

This task has been completed at PESIT Bangalore South Campus (PESIT-BSC) is an approved and affiliated campus of Visvesvaraya Technological University (VTU) in Belagavi, Karnataka, India.

REFERENCES

[1] T. F. Smith and M. S. Waterman, “Identification of common molecular subsequences,” *Journal of Molecular. Biology*, vol. 147, no. 1, pp. 195–197, 1981. doi: 10.1016/0022-2836(81)90087-5.

[2] R. Nikhil, "Bluespec System Verilog: efficient, correct RTL from high level specifications,"*Proceedings. Second ACM and IEEE International Conference on Formal Methods and Models for Co-Design, 2004. MEMOCODE '04.*, 2004, pp. 69-70, doi: 10.1109/MEMCOD.2004.1459818.
[3] R. Shoup, "Parameterized convolution filtering in a field programmable gate array interval," Technical Report1993.
[4] H. S. Neohand A. Hazanchuk, "Adaptive edge detection for real-time video processing using FPGAs," *Global Signal Processing*, 2004, vol. 7, no. 3, pp. 2-3.
[5] J. Wang, S. Zhong, L. Yan, and Z. Cao, "An embedded system-on-chip architecture for real-time visual detection and matching," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 525-538, March 2014, doi: 10.1109/TCSVT.2013.2280040.
[6] P. Mondal, P. Biswal, and S. Banerjee, "FPGA based accelerated 3D affine transform for real-time image processing applications," *Comput. Electr. Eng*., vol. 49, pp. 69-83, 2016, doi: 10.1016/j.compeleceng.2015.04.017.
[7] E. Kadric, D. Lakata, and A. Dehon, "Impact of parallelism and memory architecture on fpga communication energy," *ACM Trans. Reconfigurable Technol. Syst*, vol. 9, no. 4, 2016, doi: 10.1145/2857057.
[8] I. Kaur, L. Rohilla, A. Nagpal, B. Pandey, and S. Sharma, "Different configuration of low-power memory design using capacitance scaling on 28-nm field-programmable gate array," In: Muttoo S. (eds) *System and Architecture. Advances in Intelligent Systems and Computing*, vol 732.Singapore: Springer, doi: 10.1007/978-981-10-8533-8_15.
[9] L. Pezzarossa, A. T. Kristensen, M. Schoeberl, and J. Sparsø, "Using dynamic partial reconfiguration of FPGAs in Real-Time Systems," *MicroprocessandMicrosystems*, vol. 61, pp. 198-206, 2018,doi:10.1016/j.micpro.2018.05.017.
[10] S. D. Brown, R .J. Francis, J. Rose, and Z. G. Vranesic, *Field Programmable Gate Arrays*, Berlin: Springer Science & Business Media,1992.
[11] C. T. Huitzil and M. A. N. Maganda, "Areatime efficient implementation of local adaptive image thresholding in reconfigurable hardware," *ACM SIGARCH Comput. Arch. News*, vol. 42, no. 4, pp. 33–38, 2014, doi: 10.1145/2693714.2693721.
[12] A. Sungheetha and R. S. Rajendran, "A novel CapsNet based image reconstruction and regression analysis," *Journal of Innovative Image Processing*, vol. 2, no. 03, pp. 156-164, Jul. 2020, doi: 10.36548/jiip.2020.3.006.
[13] S. Dutta and A. Banerjee, "Highly precise modified blue whale method framed by blending bat and local search algorithm for the optimality of image fusion algorithm," *Journal of Soft Computing Paradigm*, vol. 2, no. 04, pp. 195-208, 2020, doi: 10.36548/jscp.2020.4.001.
[14] P. Lysaght, B. Blodget, J. Mason, J. Young, and B. Bridgford, "Invited paper: Enhanced architectures, design methodologies and CAD tools for dynamic reconfiguration of Xilinx FPGAs*," 2006 International Conference on Field Programmable Logic and Applications*, 2006, pp. 1-6, doi: 10.1109/FPL.2006.311188.
[15] G. Dougherty, "Image analysis in medical imaging: Recent advances in selected examples*," Biomed Imaging Interv. J.*, vol. 6, no. 3, pp. E32, 2010, doi: 10.2349/biij.6.3.e32.
[16] J. C. Russ, *The Image Processing Handbook*, 6th Ed., Florida: CRC Press, 2011.
[17] Z. Navabi, *Digital Design and Implementation with Field Programmable Devices*, Boston: Kluwer Academic Publishers, 2005.
[18] G. N. Chiranjeevi and S. Kulkarni, "Pipeline architecture for N==K*2L Bit modular ALU: Case study between current generation computing and vedic computing," *2021 6th International Conference for Convergence in Technology (I2CT)*, 2021, pp. 1-4, doi: 10.1109/I2CT51068.2021.9417917.
[19] B. S. Durgakeri and G. N. Chiranjeevi, "Implementing image processing algorithms using xilinx system generator with real time constraints," *2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, 2019, pp. 230-234, doi: 10.1109/RTEICT46194.2019.9016962.
[20] C. G. Narasimhamurthy and S. Kulkarni, "Fast architecture for low level vision and image enhancement for reconfigurable platform," *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 2021, pp. 1-4, doi: 10.1109/ICAECT49130.2021.9392425.
[21] C. G. Narasimhamurthy and S. Kulkarni, "Validation of the FPGA-based image processing techniquesusing the efficient tool like xilinx device generators," *International Journal of Emerging Trends in Engineering Research*, vol. 9. No. 4, Apr. 2021, doi: 10.30534/ijeter/2021/16942021.
[22] S. Hirai, M. Zakouji, and T. Tsuboi, "Implementing image processing algorithms on FPGA-based realtime vision system," *Proc. 11th Synthesis and System Integration of Mixed Information Technologies (SASIMI 2003)*, Apr. 2003, pp.378-385, .
[23] B. H. Ramyashree, R. Vidhya, and D. K. Manu, "FPGA implementation of contrast stretching for image enhancement using system generator," *2015 Annual IEEE India Conference (INDICON)*, 2015, pp. 1-6, doi: 10.1109/INDICON.2015.7443730.
[24] G. B. Reddy and K. Anusudha, "Implementation of image edge detection on FPGA using XSG," *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 2016, pp. 1-5, doi: 10.1109/ICCPCT.2016.7530374.
[25] M. Alareqi, R. Elgouri, K. Mateur, A. Zemmouri, A. Mezouari and L. Hlou, "Optimization of high-level design edge detect filter for video processing system on FPGA," *2017 Intelligent Systems and Computer Vision (ISCV)*, 2017, pp. 1-8, doi: 10.1109/ISACV.2017.8054900.
[26] P. G. Deepesh, G. Pramod, B. Kapil," Implementation of FPGA based image processing algorithm using xilinx system generator," *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no:1, pp. 1457-1462, January 2018

BIOGRAPHIES OF AUTHORS

Chiranjeevi G. N. received his Bachelor in Electronics and Communication Engineering from Visvesvaraya Technological university, Belgaum, Karnataka, India, in 2010 and M.Tech in VLSI Design from National Institute of Technology Karnataka (NITK Surathkal), Karnataka, India in 2012. Currently working as Assistant Professor at PES University EC Campus (PESIT Bangalore South Campus), Bangalore, Karnataka, India. Areas of interest are in the field of VLSI, FPGA, and Image processing applications. Currently the research interest in low power VLSI, FPGA and Medical Image processing.

Subhash Kulkarni obtained PhD degree in 2002 from E&ECE Department, Indian Institute of Technology, Kharagpur, India. He completed his Mastersdegree from CEDT, Indian Institute of Science, Bangalore in 1995 with Electronic Design and Technology specialization. He completed B.E in ECE from PDA College of Engineering, Gulbarga, Karnataka. His research interests include Image Processing, Control Systems, and High Speed Architectures using Vedic Maths. He has been into teaching in Electronics and Communication Engineering since 1989. He is presently working as Principal and Professor in ECE at PESIT Bangalore South Campus, Bangalore. He has guided 9 PhD Scholars till date and has published 120 Research Articles in Journals and Conferences.