Area efficient 2D FIR filter architecture for image processing applications

Soumica Cheemalakonda
Department of Electronics and Communication
V R Siddhartha Engineering College
Vijayawada 520007, India
soumicacheemalakonda@gmail.com

Sudeeksha Chagarlamudi
Department of Electronics and Communication
V R Siddhartha Engineering College
Vijayawada 520007, India
sudeekshach@gmail.com

Bhasita Dasari
Department of Electronics and Communication
V R Siddhartha Engineering College
Vijayawada 520007, India
bhasitadasari@gmail.com

Shaik. Sameera Sulthana
Department of Electronics and Communication
V R Siddhartha Engineering College
Vijayawada 520007, India
sameersultana9@gmail.com

Kishore Kumar Gundugonti
Department of Electronics and Communication
V R Siddhartha Engineering College
Vijayawada 520007, India
gkishorekumar@vrsiddhartha.ac.in

***Abstract*— The 2D- finite impulse response filters are efficient and are of low complexity and they are often used for image restoration, image enhancement and denoising applications. The 2D FIR filter architecture usually includes multipliers which are implemented using network of adders/ subtractors and shifters, which can be further optimized. In this study, an area efficient 2D FIR filter architecture is implemented using Radix – 2^r multiplication which reduces the no. of partial products, reduces the no. of multipliers and thus optimizes area. This symmetric filter is implemented in gate-level Verilog HDL for three kernel sizes, 3x3, 5x5 and 7x7, in direct form architecture. Intel DSP Builder is used to check the proposed filter's functionality and synthesis is carried out using Intel Quartus Standard edition. When the FPGA results of hardware resources used are compared to previous works, it is discovered that the proposed filter takes up less space for different kernel sizes.**

Keywords— Radix-2^r multiplication, Intel DSP Builder, FPGA.

I. INTRODUCTION

Two-Dimensional FIR (finite impulse response) filters are highly applicable across different domains and are especially useful in image processing applications. The high efficiency and low complexity of the 2D FIR filters are favourable for applications like image restoration, image enhancement, denoising applications, etc. 2D FIR filter optimization is a technique adapted to make the most effective use of the filter for a specific application.

Area is one of the major concerns in the hardware structure implementation of 2D FIR filters. Most of the area in the symmetric filter structures is occupied by the multipliers. This issue can be resolved by adapting a multiplier-less filter operation in which, the filter coefficients are represented in the form of an optimized binary format, consequently reducing the number of adders/subtractors. Previously, a variety of realization methods and concepts associated with area-efficient filters were presented [2,3]. [3] suggested a multiplier-less 2D IIR filter structure for small- and high-order filters using distributed arithmetic (DA) technique. In [2], the same distributed arithmetic (DA) algorithm was used to design a systolic architecture for 2D FIR filters. Another effective method of optimization was presented in [4] in which, the optimization of design symmetric FIR filter is preformed using the coefficients derived from Fast Converging Cuckoo Search Algorithms.

Recently, a circular symmetric 2D FIR filter architecture was proposed in [1] using Park-McClellan transformation method. This work focused on presenting an efficient hardware implementation model in which, the filter coefficients were derived using modified Park-McClellan transformation and were then quantized using canonical signed digit (CSD) binary number format. Though the model has proven to be highly efficient compared to the previously proposed models in terms of area, power, and speed, it can be optimized further.

A study on resource usage of FPGA for different sizes of Gaussian filter to state the differences between implementations of fixed point and floating point is carried out in [14]. This study has proven that, floating point consumes a greater number of resources than fixed point. An interesting concept of stationing a 2D convolver on an FPGA which is divided as static region and partially reconfigurable region, so that processing of an image can be done according to the image characteristics is presented in [13].

In the current work, an area optimized 2D FIR filter architecture is designed with Radix 2^r arithmetic multiplication technique. In this approach, the filter coefficients are scaled up to obtain an integer and then are multiplied by Radix 2^r multiplication constant [5]. This process minimizes the number of additions which results in reduced number of adders essential for the implementation of the filter. The reduction of adders will in turn reduce the area. The additional advantage in reducing the adder count is the reduction of simulation time of the filter. The proposed structures are implemented in gate-level Verilog HDL for three different 2D FIR filter kernel sizes, 3x3, 5x5, and 7x7 in the direct form. The functionality of the filter is tested using Cyclone® V 5CSEMA5F31C6 board and the board description file is created using Intel DSP Builder, which can be operated on MATLAB command prompt. The synthesis is carried out using Quartus Standard edition.

The results obtained in the present work are compared to those of [12-14] in the utilization of FPGA resources. The structures of the present model utilize a relatively smaller number of available FPGA resources compared to the previously proposed models, thereby reducing the area. And also, the Verilog HDL codes of filter architectures are applied on a 256x256 image and obtained images are presented and compared with each other.

978-1-6654-8094-9/22/$31.00 ©2022 IEEE

This paper is further divided into six sections. Section II discusses the 2D FIR filter architecture and coefficients considered for the present work, Section III shows the proposed model, that is, Radix 2^r, Section IV shows the simulation and functional verification using DSP Builder and Quartus Standard edition, and Section V discusses the FPGA implementation, hardware utilization results with an explanation, followed by Sections VI which concludes the paper.

II. 2D FIR FILTER

In this study, the symmetric filters of different kernel sizes are implemented in Fully Direct form which is one among 4 different 2D filter architectures, fully direct form, fully transpose form, transpose-direct form (hybrid-I), and direct-transpose form (hybrid-II). This architecture is easy to implement. The general architecture of this direct form is shown in Fig. 1. Direct form realization can also be referred as transversal or tapped-delay- line filter as the delayed inputs is tapped by the outputs. The utilized delay elements and the order of the filter are same in the Direct form realization hence the structure is also called a canonical structure.

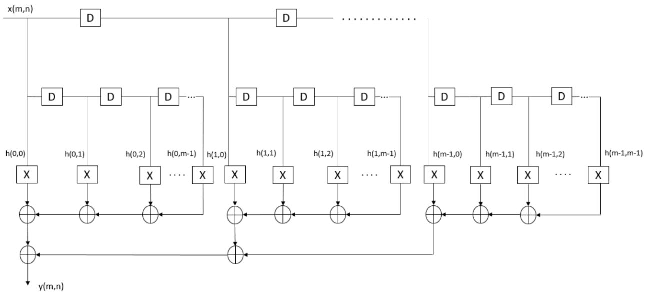


Fig. 1. General architecture of fully direct form

The kernels considered for this study are presented in Fig. 2 which are taken from [15]. The filter orders considered are 3x3, 5x5 and 7x7.

1/16

1	2	1
2	4	2
1	2	1

1/273

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

1/1003

0	0	1	2	1	0	0
0	3	13	22	13	3	0
1	13	59	97	59	13	1
2	22	97	159	97	22	2
1	13	59	97	59	13	1
0	3	13	22	13	3	0
0	0	1	2	1	0	0

Fig. 2. Approximations of Gaussian kernel 3x3, 5x5, 7x7

III. PROPOSED METHOD FOR MULTIPLIER-LESS FILTER OPERATION

Multiplication of data is a fundamental operation in 2D FIR filters. The multiplier block of the filter, where the filter coefficients and inputs are multiplied, has a significant impact on the cost, complexity, and performance. To increase the efficiency of the design, a full-block generic multiplier can be implemented using adders/subtractors. There are few techniques to optimize the filter architecture such as CSD, VCSE, HCSE etc. [1]. The optimization can be further improved by using a more efficient multiplication method, that is Radix-2^r for constant multiplication. Radix 2^r algorithm is a technique used to synthesize multiplier block in order to optimize the design of the 2D FIR filter.

Radix-2^r aims at reducing the number of steps involved in performing multiplication when compared to conventional methods [10]. Various radix multiplication techniques were discussed such as radix 2, radix 4, radix 8 depending on r value among which, Radix 8 can reduce the number of partial products by a factor of 1/3rd and 1/6th when compared to radix 4 and radix 2 respectively [11]. Thus, considering Radix 2^r multiplication as discussed in [5,9], the N-bit constant H is given as,

$$H = \sum_{j=0}^{(N+1)/r-1} [h_{rj-1} + 2^0 h_{rj} + 2^1 h_{rj+1} + \ldots. + 2^{r-2} h_{rj+r-2} - 2^{r-1} h_{rj+r-1}] \times 2^{rj} \qquad (1)$$

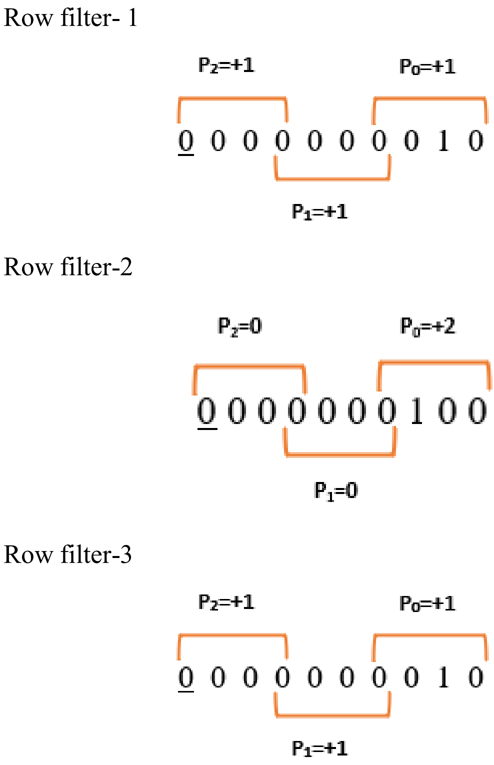
$$H = \sum_{j=0}^{(N+1)/r-1} P_j \times 2^{rj} \qquad (2)$$

where N represents the number of bits, and Pj in (2) represents the value in the booth encoder corresponding to r+1 bits considered. Here we are considering values of N and r as 8 and 3 respectively. The process groups the bits of the 2's compliment representation of coefficient H into 4 (r+1) bits and assigns Pj values calculated from (1) [10]. The grouping starts from LSB after appending a 0 before LSB. Thus, for all values of j, the Radix 8 multiplication table with all Pj values is given in Table I.

TABLE I. RADIX 8 BOOTH ENCODING TABLE

Bits grouped				**P_j**
0	0	0	0	0
0	0	0	1	+1
0	0	1	0	+1
0	0	1	1	+2
0	1	0	0	+2
0	1	0	1	+3
0	1	1	0	+3
0	1	1	1	+4
1	0	0	0	-4
1	0	0	1	-3
1	0	1	0	-3
1	0	1	1	-2
1	1	0	0	-2
1	1	0	1	-1
1	1	1	0	-1
1	1	1	1	0

Thus, from (1) and (2) and from Table 1, encoding of Pj for the 3x3 kernel considered from Fig. 2, is done as shown in Fig. 3.


Row filter- 1

Row filter-2

Row filter-3

Fig. 3. Encoding of coefficient using Radix 8 multiplication

Similarly, 5x5 and 7x7 kernels' coefficients can also be encoded the way presented above. The coefficient can thus be obtained from (2) where N=8 and r=3.

$$H = \sum_{j=0}^{3} P_j \times 2^{3j} \tag{3}$$

Thus, the number of partial products and hence the number of adders required can be lowered by encoding all the coefficients of various kernel sizes using Radix 8 multiplication and implementing them in direct form architecture.

IV. SIMULATION AND FUNCTIONAL VERIFICATION

This section discusses the simulation of Verilog HDL codes of 3x3, 5x5 and 7x7 filters using Radix 8 multiplication performed in DSP Builder and the results of resource allocation on Field-programmable gate array.

A. *DSP Builder*

DSP Builder is a programme that connects the MATLAB and Simulink software from MathWorks with the Intel Quartus II software from Intel. Automatic Verilog HDL Test Bench creation and Quartus II compilation control are included in DSP builder. Quick prototyping is facilitated in Intel DSP development boards with the help of fixed-point arithmetic and logical operators in conjunction with the Simulink software. Signals from Intel device on DSP board are detected by SignalTap II Logic analyzer and the images are analyzed by loading data into MATLAB workspace. Hardware in the loop (HIL) and HDL import enable the Verilog HDL design and FPGA hardware accelerated co-simulation with Simulink [16].

Fig. 4 illustrates the block diagram representing the steps involved in the development of model file in DSP Builder and Simulink. The Simulink simulates the model (MDL) file, the compile block compiles the file to generate output files on which Register transfer level (RTL) synthesis is performed. For automated simulation flow, simulation tools like Quartus II software are used. Then a program is generated to load into DE1-SoC hardware development board [16].

Main reason behind using DSP builder is that it provides built-in-board components of DE1-SoC board. It contains board libraries, which are defined by XML board description file, that has all FPGA pin assignments [16]. For this study, Cyclone® V 5CSEMA5F31C6 board file is used which can be operated on MATLAB command prompt.

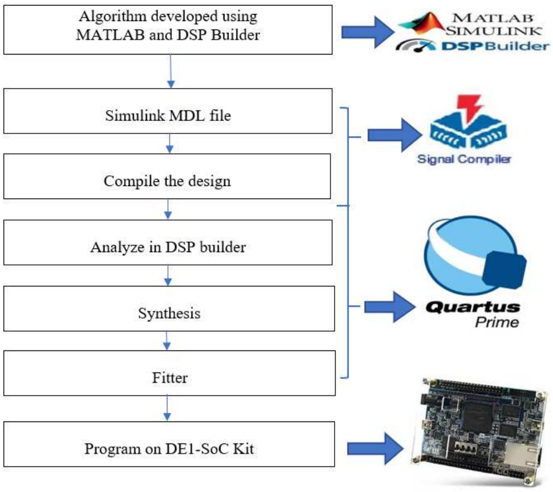


Fig. 4. Block diagram of simulation

B. *FPGA implementation*

Fig. 5 shows the verification of functionality using Intel DSP Builder. The subsystem contains the Verilog HDL of filter coefficients of order 3x3 implemented using Radix 8 for direct form architecture. It is given 8-bit input and output ports along with a step input. A 256x256 image as shown in Fig. 7(a) is given as input, which is pre-processed by the pre-processing block shown in Fig. 6(a) and given to the subsystem. The output of the subsystem is then post processed by the block shown in Fig. 6(b).

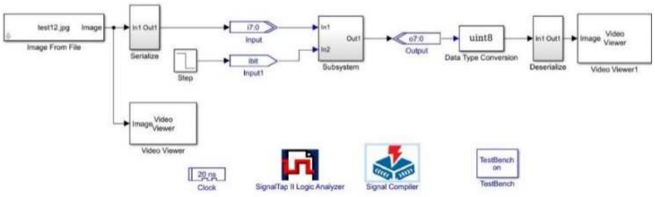
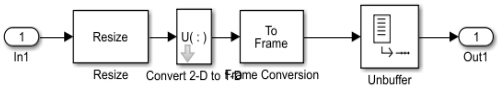
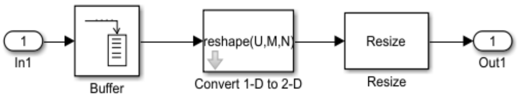


Fig. 5. FPGA implementation of proposed filter



(a)

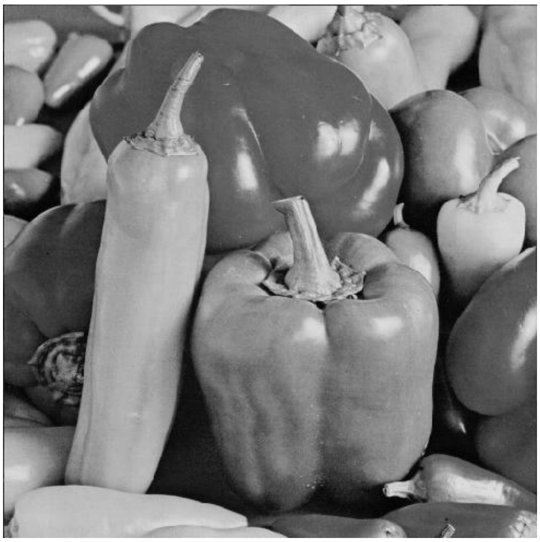


(b)

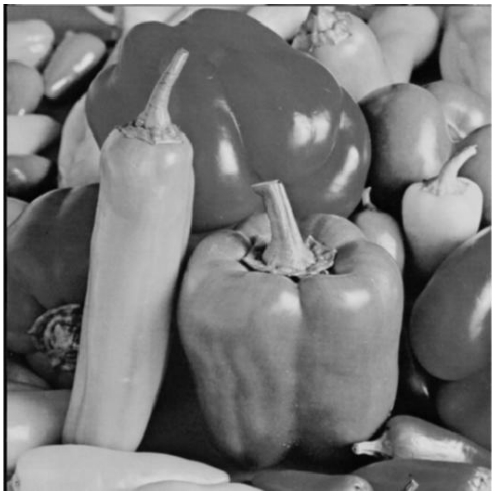
Fig. 6. (a) pre-processing block, (b) post processing block

For the 256x256 image given as input, the output image obtained for 3x3 filter implemented using Radix 8 is shown in Fig. 7(b). Similarly, Verilog HDL of filters with orders 5x5 and 7x7 are loaded into the DSP Builder block and are simulated. For the filter orders 5x5 and 7x7, the output images can be obtained similarly as obtained for 3x3 filter for a 256x256 image input. According to [1], when compared to higher order filters, lower order filters are preferable. Though filters with higher orders tend to remove more noise, they add unnecessary artifacts which may blur out the edges of the output image. Therefore, as the order of the filter increases, the output image's clarity degrades.

For the same coefficients and same architecture, MATLAB implementation is carried out and Fig. 7(a) is given as input and output image is obtained. The Structural Similarity Index (SSIM) is computed between the image obtained from hardware implementation using DSP builder and software implementation using MATLAB and is observed that the SSIM value is considerably more and the two images are very much similar.



(a)



(b)

Fig. 7. (a) input 256x256 image, (b) output obtained by the implementation of 3x3 filter using Radix 8.

V. SYNTHESIS RESULTS

In this section, the synthesis results, or resources utilized on the FPGA, are shown. The simulation of proposed model for different kernel sizes is carried out using Quartus 18.0 supported by DSP Builder. The amount of look up tables (LUTs), flipflops (FF), and digital signal processing (DSP) blocks are often used to quantify the area occupancy of an FPGA platform [12]. The resources utilized by the proposed filter are compared with other filters considered from [12-14]. [12] presented an area efficient reconfigurable convolver implemented on Xilinx Virtex-4 (XC4VLX25) FPGA platform. [13] proposed splitting of FPGA into a partially reconfigurable region which can adapt to arbitrary kernel sizes used Xilinx XCV4LX25 FPGA. And [14] implemented a 2D Gaussian filter using fixed-point and floating-point arithmetic for efficiency used Xilinx XC6SLX16. For fair comparison, these filters are computed. Table II shows the comparison between proposed model and other filter models taken from [12-14].

Table II shows that the resources used by the proposed model are fewer than those used by other models. This is because the proposed model used Radix 8 multiplication technique which groups 4 bits, assigns the group a value from booth encoding table. This reduces the number of partial products and thus adders. DSP slices are not used as this is a multiplier less design [1]. They can be used for other applications. And also, as direct form architecture is being used, LUTs used will be less than other architectures. Thu-s flipflops, LUTs utilized are fewer than other models presented in [12-14].

Between [12], [14], and the suggested one, the LUTs of the 3x3 filter have decreased by 82% and 56%, respectively. Similarly, between [13],[14] and proposed one, the LUTs of 5x5 filter have decreased by 94% and 88%, respectively. And, between [13],[14] and proposed model, the LUTs of 7x7 filter have decreased by 89% and 86%, respectively. The adders count increases as the order of the filter increases, hence the resources used increase as well, however they are significantly less when compared to other models of the same order. Thus, this approach lowers the count of resources used for any order.

TABLE II. COMPARISON OF RESOURCE UTILIZATION OF DIFFERENT FILTERS FOR VARIOUS KERNEL SIZES

Device	**Proposed 5CSEMA5F31C6**			**[12] XC4VLX25**	**[13] XCV4LX25**		**[14] XC6SLX16**		
Resource Utilization	*3x3*	*5x5*	*7x7*	*3x3*	*5x5*	*7x7*	*3x3*	*5x5*	*7x7*
FF	104	248	394	396	4978	4892	NR	NR	NR
LUT	91	256	358	532	4612	3265	209	2296	2626
DSP blocks	0	0	0	14	20	0	0	0	0

VI. CONCLUSION

This study presents an efficient technique to optimize the area occupied by a 2D FIR filter architecture for 3 different filter orders, 3x3, 5x5 and 7x7 taken in direct form. The optimization is carried out using Radix 2^r multiplication. This is applied on standard coefficients of various kernel sizes. This technique reduces the number of adders required thus reduces the hardware utilized. The synthesis is carried out using DSP Builder into which the Verilog HDL of different filter orders implemented in Radix 8 are loaded. A 256x256 image is given as input and filtered outputs for different kernel sizes are obtained. It is seen that, as filter order increases, output image quality decreases. Then the simulation of this Verilog HDL design of different kernel sizes is carried out using Cyclone® V 5CSEMA5F31C6 board file and the utilization of resources is obtained. The obtained values are compared with other models presented in [12-14] and is concluded that, the proposed model lowers the number of hardware resources needed when compared to other models for the same filter order.

ACKNOWLEDGEMENT

We express our heartfelt gratitude to our project guide, Dr. G. Kishore Kumar (Department of ECE, V R Siddhartha Engineering College) who guided us through the process of publishing this work and offering valuable suggestions. Our sincere thanks to our valued organization for allowing us to work on this project. We also appreciate our teammates' important contributions to our endeavors.

REFERENCES

[1] Venkata Krishna Odugu, C. Venkata Narasimhulu and K. Satya Prasad, “Design and implementation of low complexity circularly symmetric 2D FIR filter architectures”, Multidimensional Systems and Signal Processing, LLC, part of Springer Nature 2020.

[2] Prashant Kumar, Prabhat Chandra Shrivastava, Manish Tiwari and Ganga Ram Mishra, “High-throughput, Area-efficient architecture of 2-D block FIR filter using Distributed Arithmetic Algorithm”, Circuits, Systems, and Signal Processing, part of Springer Nature 2018.

[3] Prashant Kumar, Prabhat Chandra Shrivastava, Manish Tiwari and Amit Dhawan,“ASIC implementation of Area-efficient, High-throughput 2-D IIR filter using Distributed arithmetic,” Circuits Syst Signal Process, LLC 2017.

[4] Poulami Das, Sudip Kumar Naskara and Sankar Narayan Patra,“Fast converging Cuckoo search algorithm to design symmetric FIR filters”, International journal of computers and applications (2019).

[5] Abdelkrim K. Oudjida and Nicolas Chaillet,“Radix-2^r arithmetic for multiplication by a constant”, IEEE transactions on circuits and systems—ii: express briefs, vol. 61, no. 5, may 2014.

[6] Palnitkar S, Verilog HDL: a guide to digital design and synthesis. Upper Saddle River: Prentice Hall Professional; 2003.

[7] Chandra, A., and Chattopadhyay, S. (2016),“A new strategy of image denoising using multiplier-less FIR filter designed with the aid of the differential evolution algorithm”, Multimedia Tools and Applications, 75(2), 1079–1098.

[8] Bindima, T., and Elias, E. (2017),“Design and implementation of low complexity 2-D variable digital FIR filters using single-parameter-tunable 2-D farrow structure”, IEEE Transactions on Circuits and Systems I: Regular Papers, 65(2), 618–627.

[9] Kishore Kumar Gundugonti and Balaji Narayanam,“Efficient Haar Wavelet Transform for Detecting Saccades and Blinks in Real-Time EOG Signal”, SN Computer Science (2021) 2:156.

[10] Sukhmeet Kaur, Suman and Manpreet Signh Manna, “Implementation of Modified Booth Algorithm (Radix 4) and its comparison with booth algorithm (Radix 2)”, Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 3, Number 6 (2013), pp. 683-690.

[11] Peter-Michael Seidel, Lee D. McFearin and David W. Matula,“Secondary Radix recordings for higher Radix multipliers”, IEEE TRANSACTIONS ON COMPUTERS, VOL. 54, NO. 2, FEBRUARY 2005.

[12] Abbas Dehghani, Ali Kavari, Mahdi Kalbasi and Keyvan RahimiZadeh, “A new approach for design of an efficient FPGA-based reconfigurable convolver for image processing”, The Journal of Supercomputing, LLC, part of Springer Nature 2021.

[13] Francisco Fons n, Mariano Fons and Enrique Canto, “Run-time self-reconfigurable 2D convolver for adaptive image processing”, Microelectronics J 42(1):204–217 (2010).

[14] Cabello F, León J, Iano Y and Arthur R (2015), “Implementation of a fixed-point 2D Gaussian filter for image processing based on FPGA”, In: 2015 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), Poznan, pp 28–33.

[15] O.S. Shipitko and A.S. Grigoryev, “Gaussian filtering for FPGA based image processing with High-Level Synthesis tools”, Conference: IV International Conference on Information Technology and Nanotechnology (ITNT-2018).

[16] N. Habibunnisha and D. Nedumaran, “Hardware-Based Document Image Thresholding Techniques Using DSP Builder and Simulink”, Artificial Intelligence and Evolutionary Computations in Engineering Systems, Advances in Intelligent Systems and Computing 1361 (2020).