Ministry of Education and Science of Ukraine

Odessa National Academy of Food Technologies

International Competition of Student Scientific Works

BLACK SEA SCIENCE 2021

Information Technology, Automation and Robotics

Proceedings

Odessa, ONAFT 2021

UDC 004.01/08

Editorial board:

Prof. B. Iegorov, D.Sc., Rector of the Odessa National Academy of Food Technologies, Editor-in-chief
Prof. M. Mardar, D.Sc., Vice-Rector for Scientific and Pedagogical Work and International Relations, Editor-in-chief
Dr. S. Kotlyk, Ph.D., Assoc. Prof., Director of the P.M. Platonov Educational-Scientific Institute of Computer Systems and Technologies "Industry 4.0", Editor-in-chief
O. Sokolova – Senior Lecturer of the Department of Information Technology and Cybersecurity, ONAFT, Technical Editor

Black Sea Science 2021: Proceedings of the International Competition of Student Scientific Works. Information Technology, Automation and Robotics. / Odessa National Academy of Food Technologies; B.Yegorov, M. Mardar, S.Kotlyk (editors-in-chief.) [*et al.*]. – Odessa: ONAFT, 2021. – 526 p.

These materials of International Competition of Student Scientific Works «Black Sea Science 2021» contain the works of the contest participants in the section «Information technologies, automation and robotics» (not winners).

The author of the work is responsible for the accuracy of the information.

Odessa National Academy of Food Technologies, 2021

Organizing committee:

Prof. Bogdan Iegorov, D.Sc., Rector of Odessa National Academy of Food Technologies, Head of the Committee

Prof. Maryna Mardar, D.Sc., Vice-Rector for Scientific and Pedagogical Work and International Relations of Odessa National Academy of Food Technologies, Deputy Head of the Committee

Prof. Stefan Dragoev, D.Sc., Vice-Rector for Scientific Work and Business Partnerships of University of Food Technologies (Bulgaria)

Prof. Baurzhan Nurakhmetov, D.Sc., First Vice-Rector of Almaty Technological University (Kazakhstan)

Prof. Mircea Bernic, Dr. habil., Vice-Rector for Scientific Work of Technical University of Moldova (Moldova)

Prof. Jacek Wrobel, Dr. habil., Rector of West Pomeranian University of Technology (Poland)

Prof. Michael Zinigrad, D.Sc., Rector of Ariel University (Israel)

Dr. Mei Lehe, Ph.D., Vice-President of Ningbo Institute of Technology, Zhejiang University (China)

Prof. Plamen Kangalov, Ph.D., Vice-Rector for Academic Affairs of "Angel Kanchev" University of Ruse (Bulgaria)

Dr. Alexander Sychev, Ph.D., Assoc. Professor of Sukhoi State Technical University of Gomel (Belarus)

Dr. Hanna Lilishentseva, Ph.D., Assoc. Professor, Head of the Department of Merchandise of Foodstuff of Belarus State Economic University (Belarus)

Prof. Heinz Leuenberger, Ph.D., Professor of the Institute of Ecopreneurship of University of Applied Sciences and Arts (Switzerland)

Prof. Edward Pospiech, Dr. habil., Professor of the Institute of Meat Technology of Poznan University of Life Sciences (Poland)

Prof. Lali Elanidze, Ph.D., Professor of the Faculty of Agrarian Sciences of Iakob Gogebashvili Telavi State University (Georgia)

Dr. V. Kozhevnikova, Ph.D., Senior Lecturer of the Department of Hotel and Catering Business of Odessa National Academy of Food Technologies, Secretary of the Committee

The jury for the section «Information technologies, automation and robotics»

Head of the jury:

Sergii Kotlyk – Ph.D., Associate Professor, Director of the P.M. Platonov Educational-Scientific Institute of Computer Systems and Technologies "Industry 4.0" of Odessa National Academy of Food Technologies (Ukraine)

Members of the jury:

Piotr Artiemjew - Dr hab., Associate Professor in Decision Systems of the Faculty of Mathematics and Computer Science,University of Warmia and Mazury in Olsztyn (Poland)

Francisco Antonio Augusto – Dr., International Relations Manager of Higher Institute of Information and Communication Technologies (Angola)

Andrey Kuprijanov – Ph.D., Associate Professor of the Department of Software for Computers and Automated Systems of Belarusian National Technical University (Belarus)

Simon Milbert – Vice-President of Xtra Information Management, Inc. (USA)

Ivan Palov – D.Sc., Professor of University of Ruse "Angel Kanchev" (Bulgaria)

Degla Gérard Hugues – Communications and Training Manager of "MAPCOM solutions informatiques" company group (Benin)

Nugzar Kereselidze - Academic Doctor of Informatics (Computer Science), Associate Professor of the Department of Natural Sciences, Mathematics, Technology and Pharmacy, Sukhumi State University (Georgia)

Etibar Seyidzade - Associate Professor of the Department of Computer and Information Technolohies, Baku Engineering University (Azerbaijan)

Vladimir Golenkov, D.Sc., Professor of the Department of Intelligent Information Technologies, Belarusian State University of Informatics and Radio Electronics (Belarus)

Zhanar Omirbekova - Ph.D., Associate Professor of the Department of Automation and Management, Satbayev University (Kazakhstan)

Ivan Palov - D.Sc., Professor of the Department of Power Supply and Electrical Equipment, University of Ruse "Angel Kanchev" (Bulgaria)

Siarhei Palavenia - Ph.D., Associate Professor, Head of the Department of Telecommunication Systems, Belarusian State Academy of Communications (Belarus)

Alexander Goloskokov - Ph.D., Professor of the Department of Software Engineering and Information Technology Management, National Technical University "Kharkiv Polytechnic Institute" (Ukraine)

Peter Nikolyuk - D.Sc., Professor of the Department of Computer Technology, Vasyl Stus Donetsk National University (Ukraine)

Vladimir Palagin - D.Sc., Professor, Head of the Department of Radio Engineering, Telecommunications and Robotics Systems, Cherkasy State Technological University (Ukraine)

Viktor Khobin – D.Sc., Professor, Head of the Department of Technological Processes Automation and Robotic Systems of Odessa National Academy of Food Technologies (Ukraine)

Valeriy Plotnikov – D.Sc., Professor, Head of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies (Ukraine)

Sergii Artemenko – D.Sc., Professor, Head of the Department of Computer Engineering of Odessa National Academy of Food Technologies (Ukraine)

Fedir Trishyn - Ph.D., Associate Professor, Vice-Rector on Scientific and Educational Work, Odessa National Academy of Food Technologies (Ukraine)

Valerii Levinskyi – Ph.D., Associate Professor of the Department of Technological Processes Automation and Robotic Systems of Odessa National Academy of Food Technologies (Ukraine)

Viktor Yehorov – Ph.D., Supervisor of the Laboratory of Mechatronics and Robotics of Odessa National Academy of Food Technologies (Ukraine)

Pavlo Lomovtsev – Ph.D., Associate Professor of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies (Ukraine)

Yurii Kornienko – Ph.D., Associate Professor of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies (Ukraine)

Serhii Shestopalov – Ph.D., Associate Professor of the Department of Computer Engineering of Odessa National Academy of Food Technologies (Ukraine)

Anatoly Galiulin - Ph.D., Associate Professor, Acting Head of the Department of Electromechanics and Mechatronics, Odessa National Academy of Food Technologies (Ukraine)

Secretary of the jury:

Oksana Sokolova – Senior Lecturer of the Department of Information Technology and Cybersecurity of Odessa National Academy of Food Technologies (Ukraine)

ПРОЕКТ

IMPLEMENTATION OF IMAGE PROCESSING AND OUTPUT USING DIGITAL FILTERS USING FPGA

Author: **A. *A. Mukhanbet***
Advisors :***Y. S. Nurakhov, T. S. Imankulov***
Kazakh National University named after Al-Farabi (Almaty, Kazakhstan)

Abstract. *Digital image processing is an ever-growing industry, including medicine, video surveillance, and more. To improve the performance of image processing systems, image processing algorithms are implemented not programmatically, but hardware. The idea here is to achieve results by processing images at the hardware level faster and more efficiently than software processing. Thus, digital signal processing algorithms and operations were implemented using the Basys 3 FPGA board. FPGA provides the advantage of parallelism, low costs, and low energy consumption. In this article, the advantages of using FPGA for the implementation of image processing algorithms such as median filter, morphological, convolution, smoothing and edge detection, etc.were demonstrated and the results were obtained. The picture is provided .we send it to the FPGA block RAM in .coe format, and then we can check the processing operations performed depending on our choice. Then we output the images to the monitor using VGA. Image types use eight bits to encode the value of each pixel, resulting in shades of gray with a black value of 256 to 0 and a w hite value of 255.*

Keywords: *FPGA, digital image processing, algorithms, filter, VGA*

Introduction. Digital image processing means processing and displaying images. Image editing-used to change the image. There are three main categories of image processing: image enhancement, Image recovery, and image classification. image enhancement, which allows you to effectively display data for visual interpretation. This helps the user see the image and recognize different segments of the image. An example of this is the processing of shades in an image. This method is very useful to help you distinguish between different objects in an image. Image correction and recovery is another important aspect of image processing. It mainly deals with image correction, which can affect the image due to geometric distortions or noise. It can also eliminate blurring, as a result of which a low-quality image can be updated to an image with high-quality and clear features.

There are many useful image processing applications. It is used as an emotion sensor to guide the robot and recognize goals. It is also used for industrial control and in medical technologies such as X-ray amplification. A very useful application for digital image processing is to look at the color intensity in the image and divide the image into segments based on the results obtained. The biggest performance challenge is the time it takes to process the images captured by the camera. Such applications may be easy to implement on a general-purpose computer, but they are not very time-consuming due to additional memory limitations and other peripherals.

This leads to the study of possible hardware alternatives. Image processing algorithms, which have recently been introduced in hardware, have become the most viable solution for improving the performance of image processing systems. FPGAs are often used as platforms for real-time image processing. An FPGA is a programmable device [1] in which the logical structure can be configured directly by the user. An FPGA consists of an array of unwritten elements that can be programmed or interconnected and have virtually unlimited ways according to the user's description. Reprogrammed and easily updated, the FPGA offers a compromise between the flexibility of universal processors and the speed of ASIC hardware. They allow you to quickly create a prototype of the system and provide a cheaper option for testing system requirements [2]. Placing the functionality of image processing applications on the hardware makes it possible to speed up processing, since there is no need to break down individual instructions into the sampling, decoding, and application cycles that are required on a typical computer processor. In addition, taking into account the speed of the recorded process and the obvious dynamics of the variability of its flow structure, it is necessary to quickly capture at a high frame rate, which creates a large flow of video frames that must be analyzed and processed in real time. An effective solution to this problem is possible by parallel processing of the image stream. The construction of parallel image processing systems can be carried out using SystemOn-Chip technology based on the FPGA architecture.

This paper provides an overview of the implementation of image processing applications in FPGA, focusing on the features inherent in FPGA. The rest of the article describes in detail the processing algorithms and operators. In addition, various filtering algorithms were described, such as accumulation filter, Gauss filter [3], etc.finally, the results obtained by implementing image processing algorithms in FPGA were shown.

Image processing algorithm. Many studies have been conducted in the field of hardware image processing using FPGA. The image processing algorithms considered for hardware implementation include aggregation, image filtering, and edge detection (Sobel, Prewitt, and edge detection). These algorithms are based on image processing. Grinding filters are widely used in many applications, such as object recognition, identification, classification, etc.they are used as pretreatment to eliminate unnecessary particles and noise. We focus on image filtering based on the Gauss filter.

Gauss mask. The Gauss filter is one of the most important and widely used filtering algorithms for image processing [4]. The Gauss filter (G) is defined in equation 1.

$$G(x,y) = \frac{1}{2\pi\sigma\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \qquad (1)$$

where G is a Gaussian mask at a point with coordinates x and y, σ is a parameter that determines the Gaussian standard deviation. If the σ value is greater, then the image smoothing effect is higher.

Accumulation operation. In general, smoothing is obtained by calculating the sum of the products between the input image and a small Gaussian Matrix (3 × 3). Using a 3 × 3 mask and a 3 × 3 input image, you can achieve a 2D convolution image [5].

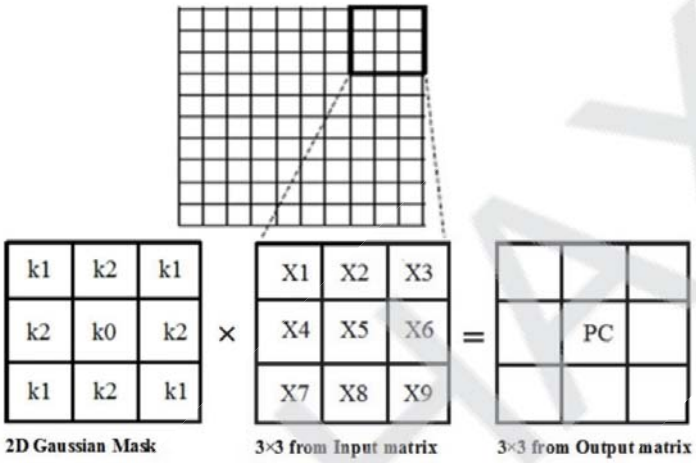

Figure 1. Accumulation operation

In image processing, the core, convolution matrix, or mask is a small matrix. It is used for blurring, sharpening, drawing patterns, defining edges, and more. This is achieved by performing a connection between the core and the image.

If the core is symmetrical, then you need to set the center of the core to the current pixel. The core overlaps with adjacent pixels around the place of origin. Each core element must be multiplied by its overlapping pixel value, and all the resulting values must be added. This resulting amount will be the new value of the current pixel, which currently overlaps the center of the core [6].

Implementation of hardware image filtering

Blocking memory. To upload an image to Verilog, you need to convert it to binary. It is created using python, and it is a format .it's called coe. The converted image is as many lines as the total number of pixels, and each line has 24 bits (8X3). Thus, the 160x115 image will have 18400 lines. Then the block memory module is executed, which has many addresses, such as strings and 24-bit data. So, for an image in 160x115 format, it will have 215 addresses. This memory module, like other modules, can be designed and used in the main module. The module contains input signals such as clock, address, datain, and read-write and dataout commands as outputs. Thus, it can only output one data set at a time, in which case it is one pixel. To summarize, you need several pixels at the same time, which use cores that are used by adding and subtracting values to the address[7-8].

VGA interface. For VGA, a 480p display code with a refresh rate of 60 Hz was written. The refresh rate is how many times the screen is updated per second. With each update, each pixel (480X640) is updated one at a time. To do this, the counter starts with the origin (0,0) and up to (0,799) and so on. They include the invisible area in the images, as well as the reverse path. They are displayed on the screen using hsync and vsync signals. The Hsync signal will be "0" after the counter reaches the end of the right border, where it will start tracking again. After re-searching, the signal will be "1" once, starting the search for the left border and the display area. In addition, for the screen, we only need to start the display from one point and end with the number of pixels connected to it in both directions [7].

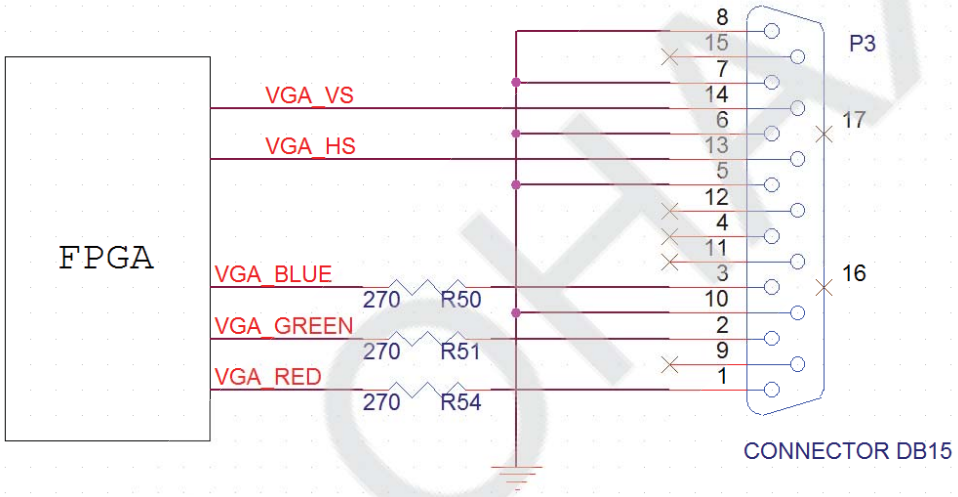

Figure 2. FPGA connection scheme with VGA

You need to convert the image to a list of binary numbers, where each binary number represents the Pixel value of the image. Using a specially designed file, you can perform basic image operations such as increasing/decreasing brightness, RGB2Gray, color inversion, and various color filters. To perform these basic image processing operations, we need to reach one pixel at a time. For example, we can convert a color image to black and white using the operation R+g+b/3 [9] for this Pixel. Therefore, we can read one pixel at a time to perform an action and display the result on the screen or save the image to your computer. We can't create any blur or define borders, because we need access to the pixels around this pixel to get the result. To apply blurring, border detection, and other filters, you need to stock up on the image using different cores. So, in the course of work, using python, the image was converted to grayscale and all the necessary pixels [10] to perform kernel operations in a single line, i.e. up, down, right, etc., pixel values were placed around the Pixel. Basically, this is not an effective method, as it uses more memory than a single image in grayscale.

In the course of our work, we will use Basys 3, a first-level FPGA development board developed specifically for the Vivado Design Suite, which is distinguished by the Xilinx Artix-7 FPGA architecture. [11].

Figure 3. FPGA Basys 3 Xilinx Artix-7 board

We use the Vivado programming environment to create a common FPGA architecture. We will create a circuit using the Clock module and VGA [12] generators.

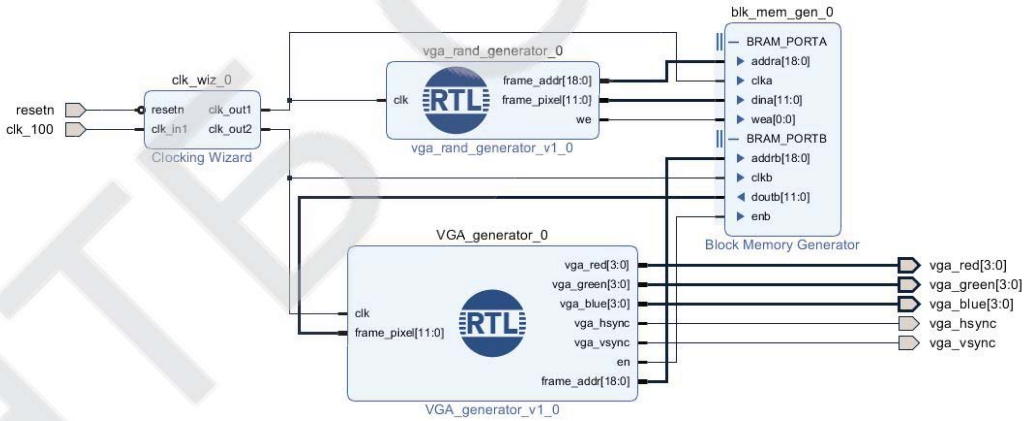

Figure 4. Functional design and circuit of VGA

Since the scope of this work is limited to real-time image processing, this means that the image must be stored in the chip's memory. This paper deals with the upper vhd module, which used a block memory generator to create a single-port ROM with a read width of 8 bits and a read depth of 128x128 pixels. Well .coe files are provided as an image file and must be loaded into the IP kernel generator when

creating ROM [13]. When tracking address bits, pixels are removed from the ROM one after the other. These bytes are then sent to another VHD file, which is responsible for storing these bytes in a separate FIFO along with the processed pixel bytes. After the FIFOs is filled in, the data is displayed on the screen and the program starts again with the ROM address 0.

After creating a VGA circuit, we can implement the synthesis and implementation process and see the simulation result in the following image.

Figure 5. Results of simulation in Vivado

Input signals can be set as binary or hexadecimal values for input contacts or ports, and the result can be seen on the screen in a time diagram similar to the output of a logic analyzer. An example of a time diagram can be seen in Figure 6, which shows the actual output of the image processing code created for the project. A detailed analysis of the simulation results is shown in the figure. The first result of Pixel reproduction occurs at 525ns[14].

The "bad negative failure (WNS)" reported by teams like Report_timing_summary is actually a bad positive failure. If the WNS is positive, then this path will pass. If it is negative, then this path is doomed to failure. According to the results of my work, this indicator was positive. To be precise, it was 6,459 ns. "Total Negative Slack (TNS)" [15] is a (real) negative weak amount in the design. If 0, then the construction corresponds to time. If this is a positive number, then this design has a negative weakness, the design is doomed to failure. It can't be negative. According to the results of my work, it was 0.

Results. As a result, we can see images created using image processing algorithms using VGA. The results are shown in Figure 6. In addition, the time of image processing algorithms on the computer and on the FPGA was calculated. The computer had 0.17708635330200195 seconds, the Basys 3 board had 0.00195 seconds.

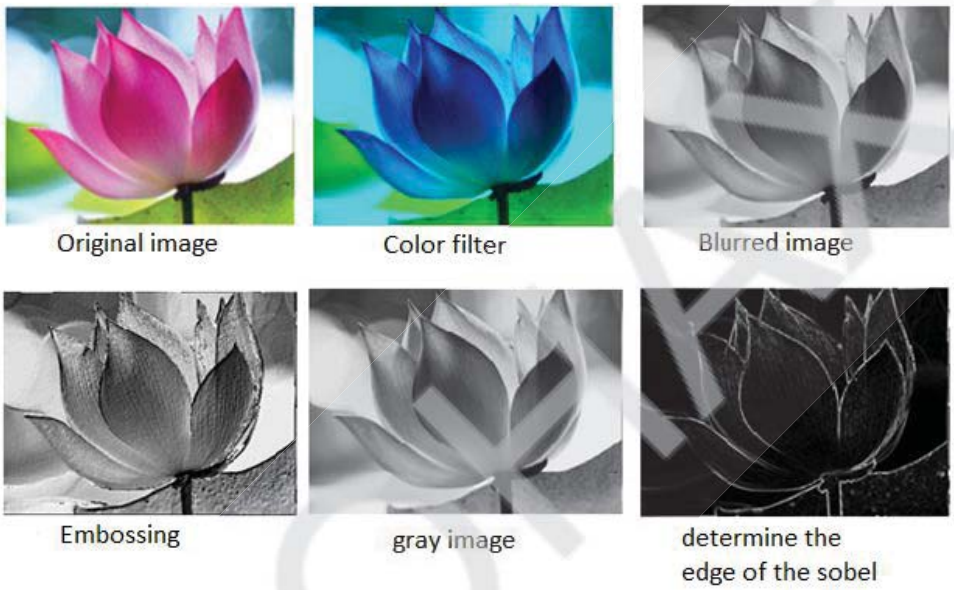

Figure 6. Images obtained as a result of processing

The most commonly used in processing is the edge detection algorithm for creating an image that highlights the edges of the image. First, we use the sobel [15] operator in the direction of x, then in the direction of y, and get the average value of both values to create the final image.

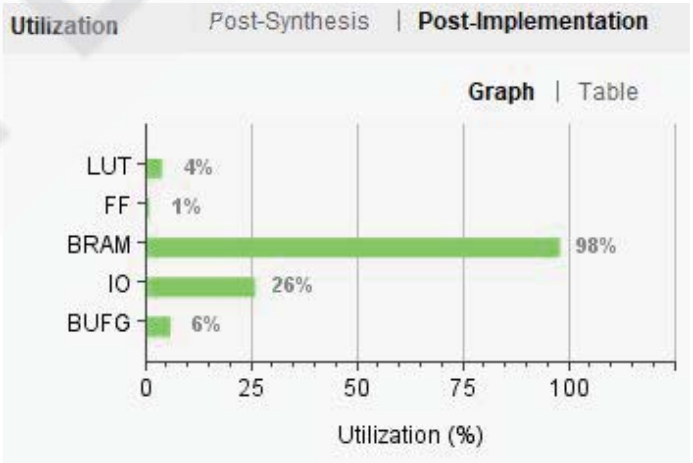

Figure 7. The place of image processing in FPGA memory

This graph shows the size of all the signals used, the amount of memory. In other words, data on resource usage and performance are provided. Since I work in Verilog and Python, the amount of memory is 98%. If we add the C language to it, it means that the size will be larger. The amount of input and output signals was 26%. This is directly related to the blocks used during processing.

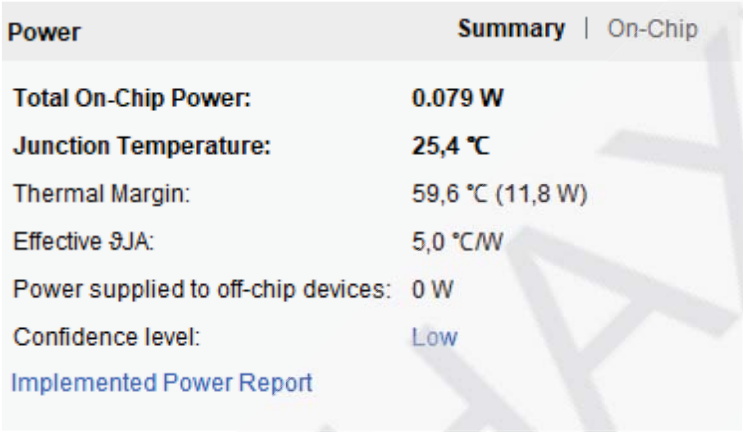

Figure 8. The amount of current and performance spent during image processing on the FPGA Basys 3 board

This is a detailed description of the power used in my work. A current of 0.079 W was spent on everything. The transition temperature was 25.4 C. And the maximum operating temperature was 59.6 C.

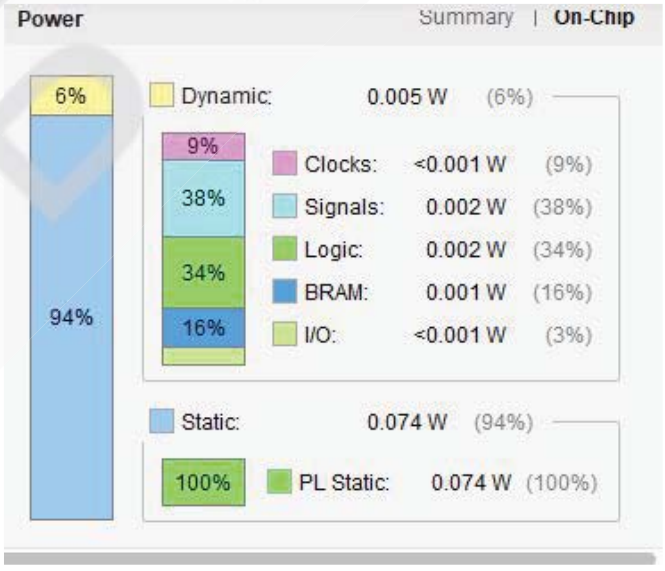

Figure 9. Graphic amount of power and performance spent during image processing on the FPGA Basys 3 panel

This is the amount of current in the chip. And the amount of current spent on time and signals is indicated.

Conclusion. With the help of Basys3 FPGA, image processing algorithms were implemented in Verilog. In this article, image processing operations were performed on the image transmitted by FPGA Basys-3, including operations related to the console. We send the transmitted image in binary form to the FPGA block RAM, and then, depending on the user's choice, perform some special processing applications, and then display it via a VGA display. To convert a given digital image to binary, we use Verilog as a hardware description language and python. Vivado software was used to do this. As a result, the processing time of the image on the computer and FPGA was calculated. And so it was made sure that image processing works quickly on the FPGA. Finally, the amount of memory spent on processing images was calculated. In addition, a current of 0.079 W was spent on processing. The transition temperature was 25.4 C. We can see that this is a very effective value for productivity.

REFERENCES

[1] Zhou, H.; Machupalli, R.; Mandal, M. Efficient FPGA implementation of automatic nuclei detection in histopathology images. J. Imaging 2019.
[2] Donald G. Bailey, Anoop S. Ambikumar. «Border Handling for 2D Transpose Filter Structures on an FPGA». School of Engineering and Advanced Technology, Massey University, Palmerston North 4442, New Zealand; 2018
[3] Kevin Morris Cray Goes FPGA. Algorithm Acceleration in the New XD1. FPGA and Programmable Logic Journal. April 5, 2005.
[4] Paulo Garcia, Deepayan Bhowmik, Robert Stewart, Greg Michaelson, Andrew Wallace. «Optimized Memory Allocation and Power Minimization for FPGA-Based Image Processing». Received: 19 November 2018; Accepted: 27 December 2018; Published: 1 January 2019
[5] Aiman Badawi, Muhammad Bilal. «High-Level Synthesis of Online K-Means Clustering Hardware for a Real-Time Image Processing Pipeline». Received: 29 November 2018; Accepted: 7 March 2019; Published: 14 March 2019.
[6] Lysakov K. F., Shadrin M. Y. SLSP – Special Processor for Image and Video Processing // Proceedings of the IEEE International Siberian Conference On CONTROL AND COMMUNICATIONS SIBCON-2007. Russia, Tomsk, April 20–21, 2007. P. 140–144.
[7] Haonan Zhou, Raju Machupalli and Mrinal Mandal. «Efficient FPGA Implementation of Automatic Nuclei Detection in Histopathology Images». Received: 30 November 2018; Accepted: 11 January 2019; Published: 17 January 2019

[8] Fahad Siddiqui, Sam Amiri, Umar Ibrahim Minhas , Tiantai Deng , Roger Woods,Karen Rafferty, Daniel Crookes. «FPGA-Based Processor Acceleration for Image Processing Applications». Received: 27 November 2018; Accepted: 7 January 2019; Published: 13 January 2019.
[9] Devjataikin A. M., Lysakov K. F., Shadrin M. J. FPGA implementation of sub-pixel images matching algorithm // Proceedings of the Second IASTED International Multi-Conference AUTOMATION, CONTROL, AND APPLICATIONS (ACIT-ACA). Russia, Novosibirsk, June 20–24, 2005. P. 93–97.
[10] Mohassin .M, Najeeb-Ud-Din Hakim. Review on Image Processing: FPGA Implementation perspective. International Journal of Innovative Research in Computer Science & Technology (IJIRCST), ISSN: 2347-5552, Volume-2, Issue-1, January 2014
[11] Stefano Di Carlo, Giulio Gambardellay, Marco Indaco, Daniele Rolfoy, Gabriele Tiotto, Paolo Prinetto, "An Area-Efficient 2-D Convolution Implementation on FPGA for Space Applications," Proceedings of the IEEE 6th International Design and Test Workshop (IDT), Vol. 7,Issue 3, 2011.
[12] Madhuri Gundam, Dimitrios Charalampidis, "Median Filter on FPGAs," 44th IEEE Southeastern Symposium on System Theory , University of North Florida, Jacksonville, FL March 11-13, 2012.
[13] Elmoncef Benrhouma, Meddeb Souad, Abdulqadir Alaqeeli, Hamid Amiri, "Study and Design of Median Filter," SIDOP 2nd Workshop on Signal and Document Processing, 2012.
[14] R.Arunmozhi, G.Mohan, "Implementation of Digital Image Morphological Algorithm onFPGA using Hardware Description Languages," International Journal of Computer Applications, Vol. 57, No.5, pp 0975 – 8887, Nov. 2012.
[15] Runbin Shi , Justin S.J. Wong and Hayden K.-H. So . «High-Throughput Line Buffer Microarchitecture for Arbitrary Sized Streaming Image Processing». Received: 21 January 2019; Accepted: 25 February 2019; Published: 6 March 2019

TABLE OF CONTENTS

International Competition of Student Scientific Works

BLACK SEA SCIENCE 2021

Information Technology, Automation and Robotics

Proceedings

Odessa National Academy of Food Technologies

The collection includes student works of the participants of the competition, which were not included in the number of prize-winners. The texts of the competitive works are published in the form in which they were submitted by the authors. The authors of the articles are responsible for the content and form of submission of the material.

Responsible for the issue: Sergii Kotlyk

Computer typesetting and layout: Oksana Sokolova

Odessa 2021