International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.14, No.1 (Jul-23)

http://dx.doi.org/10.12785/ijcds/140129

Design and Implementation of Image Processing Application with Zynq SoC

Dharmavaram Asha Devi[1], Niharika Reddy Kathula[2], Gopinath Kalluri[3] and Leela Sai Bondalapati[1-4]

[1]*Department of ECE,Sreenidhi Institute of Science and Technology, Hyderabad, India*

Received 27 Mar. 2023, Revised 6 May. 2023, Accepted 8 May. 2023, Published 1 Jul. 2023

Abstract: Human brains can depict images much easier than texts and images convey a lot more information that is the reason why images are used as branding and advertisements. Digital image processing is the major technology that needs more simple algorithms to implement. In this paper, we implemented Image processing operations like Brightness, Inversion, and Threshold operations on FPGA using the Xilinx Vivado tool. The prototype verification is done with Zynq processing system (ZPS) interfaced with OLEDrgb peripheral module in Xilinx and Software Development Kit (SDK) tool environment. The proposed design is Zynq architecture based technology that is efficient for real-time embedded applications because of the ARM9 processing system with Software Development Kit is used for application software. The advantage of the proposed system is power efficient, high performance and flexible enhancement of the functionalities because of the re-configurable Zynq SoC.

Keywords: Field Programmable Gate Array, Inversion, Software Development Kit, Threshold, Zynq SoC.

1. INTRODUCTION

Artificial Intelligence and Machine Learning are the booming technologies in the world right now. Machine Learning means making machines think like a human brain. One of the major applications of machine learning is image processing. The machines need to understand the pattern of images and analyze the information in them. Image Processing is a technique which is used to enhance the images and to extract the required information from the images. But Implementing the Image Processing Techniques on the general Microcontroller boards is not suitable for real-time applications. But implementing them using the FPGA boards have better accuracy and high performance. Especially Zynq Architectures proved to be best suitable for real-time embedded applications. We implemented various image enhancement techniques like brightness, threshold and inversion operations using Xilinx Vivado tool. The Image Enhancement techniques of images of any size [1] can be implemented using FPGA boards unless it has the memory size which supports it.

FPGA boards provide greater efficiency and they are used for implementing complex designs in various applications. It is very adaptable to various real-world applications. A high - level design language (HDL) that is simple to design and debug in Verilog. Typically, it is more helpful than schematics, particularly for large circuits. Hence, we are implementing image processing techniques such as brightness, and contrast threshold on reconfigurable Zynq SoC and Xilinx software tools. Xilinx is one of the EDA tool useful for designing various VLSI circuits. Electronic equipment makers in end sectors like medical, communications, industrial, consumer, automotive, and data processing applications are implemented with re-configurable System on chips like FPGAs. We can also see some unique applications like coin counting by using various image processing techniques implemented on FPGA [2].

2. LITERATURE SURVEY

Image processing is a trending technology that is used almost everywhere, where the enhancement of images plays a crucial role in various applications. Especially in the medical field, image processing is vital and requires greater accuracy and performance. In [3], various image processing techniques were implemented on FPGA for enhancing the medical images (MRI—Magnetic Resonance Imaging). This showed that FPGA-based implementation can be more accurate and power efficient. Similarly, the authors of reference[4] represents how medical image compression can be done by using 3 dimensional Daubechies, and the entire design is being implemented on one of the FPGAs. This proposed methodology is designed using VHDL, and the implementation was done on Altera® Cyclone II (EP2C35F672C6). This paper showed how they achieved maximum power efficiency and maximum frequency. Another major area of concentration in the current world is security. The image processing also plays a vital role in security applications.

E-mail address:

The proposed methodology in [5] shows the plate recognition of vehicles by using CNN, implemented on the PYNQ-Z2 board. This shows how machine learning can be integrated with image processing and can be implemented on FPGAs for efficient real-time applications. The advantages of using reconfigurable FPGAs like Zynq architectures are tremendous. Since these can be re-programmable, thus reducing the NRE cost [6].

There are various hardware implementations of FPGA, each with its own merits and demerits. The approach of combining the merits of two microcontroller boards, like the Raspberry Pi and FPGA [7], has enhanced the design by many times. This led to the acceleration of image processing by almost 350%. This methodology achieved maximum power efficiency and accuracy. The use of ASIC for various embedded applications is efficient but time-consuming and expensive. The alternative is the use of SOCs [8]. Implementing image processing applications can reduce the cost and result in greater efficiency in the design. That is why we prefer FPGAs over ASICs for implanting our design.

The ability to enhance or edit a given image requires knowledge of algorithms and high computing power. The features of FPGA (Field Programmable Gate Array) have made it easier to exploit its resources for real-time image processing.

Ref.[9] reports on the realisation of the design on Xilinx XC200 series FPGA and details the high level programming of the model. The significance of this paper is on building algorithms for image processing rather than spending more time on hardware implementation. The high level instruction is based on image-level neighbourhood operators of image Algebra. This implementation needed configuration of the XC2600 series FPGA chip. The paper concludes with further research and analysis on a sophisticated tool to easily implement any order of algorithms.

The contemporary image/video processing demands for complex algorithms and fuzzy logic along with high computational power from the hardware. In addition, the processing needs to be quick, consuming less amounts of power. These requirements can be accomplished by re-configurable FPGAs [10] addresses all the concerns found while implementing the design and executing it in the hardware. This paper also implements Sobel Edge Detection using ZYNQ-702 board. The implementation of software and hardware and the approach for the design is very well explained in this paper. It also demonstrates acceleration of hardware models in image or video processing and the scope of OpenCV tool for relatively better design in the future. Details on implementing a blood sample recognition using an image processing algorithm demonstrated in Ref.[11].

The main focus of the algorithm is on agglutinated samples to determine the type of blood. It is achieved by contrasting and detecting the edges of the sample. To characterise a vast number of blood samples, the system uses parallel processing architecture for FPGA implementation. Initially, the design is coded in MATLAB software and then transferred to Vertex 6 FPGA. The project also implies the time taken by various hardware systems to recognise the blood sample. It also succeeds with nearly 100% accuracy tested on more than 500 different blood samples.

The design in [12] is to fasten the image processing application performed using robust hardware implementation. The project also requires a host computer along with a FPGA chip. It is implemented on XILINX Kintex-7. It stands out from other papers because of utilisation of appropriate image filters increasing the overall performance. This paper compares the transfer rates through various communication channels for different frame sizes. The advantage of the design is due to its high accuracy and less processing time for every frame. [13] is based on the parallel dual template strategy implemented on FPGA for faster parallel and hardware computing. Yanping Wei and Hailiu Xiao experimented with various evaluative indices of results obtained on image processing using different algorithms. The paper explains the advantage of dual template strategy over single template in the implementation of mean filter algorithm. The design turns out to be 678 times better performing than the conventional approach. And the proposed design was analysed and the optimization of the parallel dual-template strategy was feasible.

The thesis in [14] aims at designing a real-time tracking and detection of vehicles with hardware implementation. The image processing algorithms are deployed on an image captured using a camera. The algorithms include Fixed, Re-configurable and Variable Block size Motion Estimations for processing captured images of the vehicle. The results proved that the power consumption was reduced by 45% when compared with the conventional VBSME algorithm. A Mixture of Gaussian (MoG) algorithm in SoC architecture fulfilled image segmentation requirements. The design was implemented on Xilinx XtremeDSP Video Starter Kit Spartan-3ADSP 3400A Edition under 25MHz clock frequency and it could easily process a resolution of 640x480 (VGA) at 30 frame-per-second (fps).

Highlights the VLSI implementation [15]for image processing applications overcoming the limited speeds observed in conventional processors of computers. This paper demonstrates three operations namely Contrast Stretching, Thresholding and Negative Transformation for image enhancement. The implementation requires knowledge of MATLAB and VHDL and operating Xilinx Sparten 3E500K. The proposed design can be applied in Biomedical image analysis, Remorse sensing, Fault detection etc. The paper addresses point processing techniques which are operated on RGB pixels directly.

The difference of the proposed work with respect to the existed work is illustrated in Table.1. The proposed work

clearly demonstrated the accurate display of the results at the target OLEDrgb display which is interfaced with Zed board. Therefore, the prototype implementation of the work has shown up to TRL4.

TABLE I. Comparison of proposed work with the existed work

Comparison parameter	Existed Work	Proposed Work
Implemented on	Spartan [Ref.2] Virtex-6 [Ref.3 & 11] Altera Cy-cloneII[Ref.4] Zynq [Ref.5,6,8 and Ref.10] Kintex 7[Ref.11 & Ref.12]	Zynq SoC
Operations /application	Sobal edge ditection[Ref.10]	Image Enhancement operations: Brightness increase and decrease, threshold, inversion etc.
Technology	28nm and 18nm	28nm
TRL	TRL1/TRL2/TRL3	Up to TRL4
Platform	Verilog [Ref.3]/ VHDL/C[Ref.5,6,8,10]	Block Level Design with IP integration and application project in C and matlab

3. PROPOSED METHODOLOGY

The proposed methodologies are design and develop the controlling parameters like brightness, threshold and inversion of an image with Verilog HDL. It is a simpler and efficient way and more suitable for real-time applications. For simulation and verification purposes, Xilinx Vivado System Design Suite software is used. In the proposed work, Matlab software is used to convert the BMP image file to a Hex file. For reading an image and producing an image with various enhancements like brightness, threshold and inversion mechanisms are implemented and simulations are done with Verilog HDL using Xilinx 2018.3 version. The flowcharts of these mechanisms are illustrated in Figure 1.

The prototype implementation and verification is done with Zed Board interfacing with OLEDrgb in SDK environment.

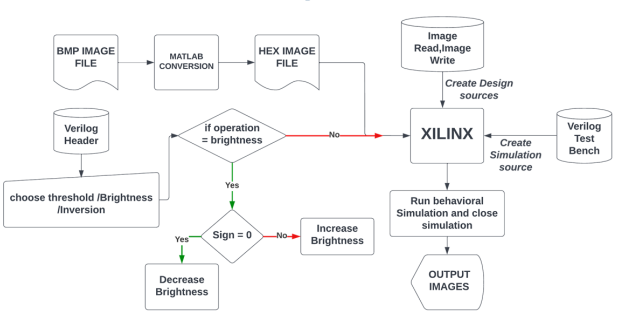


Figure 1. Flowchart of Proposed Image Processing Operations

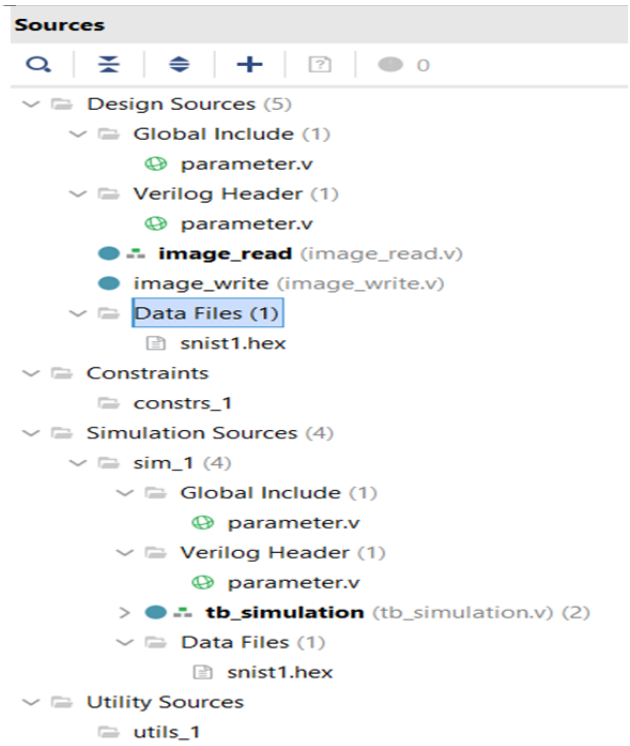


Figure 2. View of added/created source files in Xilinx tool environment

4. EXPERIMENTAL APPROACH

The first step is to convert BMP image file into HEX file using Matlab software tool. The design source files like image read and write operations are to be created. In the proposed work, these are developed in Verilog HDL. The image enhancement operations like brightness, threshold and inversions, used in the proposed work is defined in the parameter file as Verilog header and then made it as global include. In the proposed design, the image width X height is 768 X 512 pixels are considered. Now add the image file that is to be processed (in hex format) as design source, then change it to the data type file. Added source files can be viewed in the sources menu under project manager as illustrated in Figure 2, in the Xilinx tool environment.

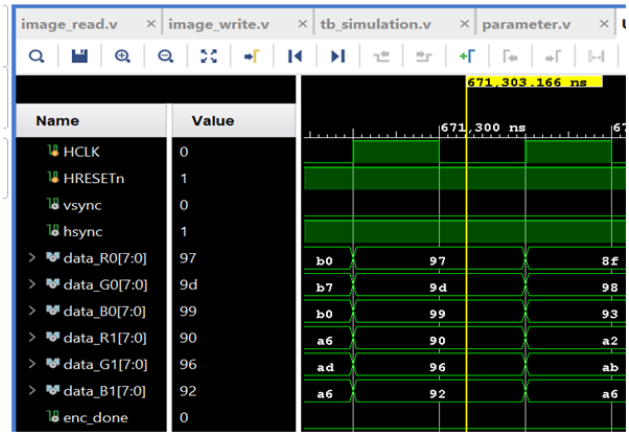


Figure 3. Simulation results for increased brightness of image data RGB values

Figure 4. Original Image

After creation of the source files, test bench file is to be created under simulation sources to check the logical functionality of the proposed design. To get the simulation results, choose run behavioral simulation from the flow navigator, and can monitor the waveform results. To view the output images, we must stop or close the simulation process and open the output.bmp image generated in the xsim (Xilinx Simulation) folder after the simulation. Figure 3 is illustrated as the image data RGB values after increasing the brightness. The corresponding output image is illustrated in Figure 5 With the enhanced brightness. It can be observed by comparing the brightness of the original image as shown in Figure 4.
Similarly the decreased brightness of the image output can be viewed in Figure 6. The simulation results of inversion process and the inverted image outputs are shown in Figure 7 and Figure 8 respectively.

The threshold values of the image are ranging from 0 to 255. Therefore, the corresponding threshold values are illustrated in Figure 9 after simulation. The generated corresponding output.bmp image is illustrated in Figure 10.

5. DESIGN WITH ZYNQ SOC

Block design is created by using IP Integrator in the flow navigator. It is created by using soft core IP. The

Figure 5. Image output after increasing the brightness level

Figure 6. Image with decreased brightness

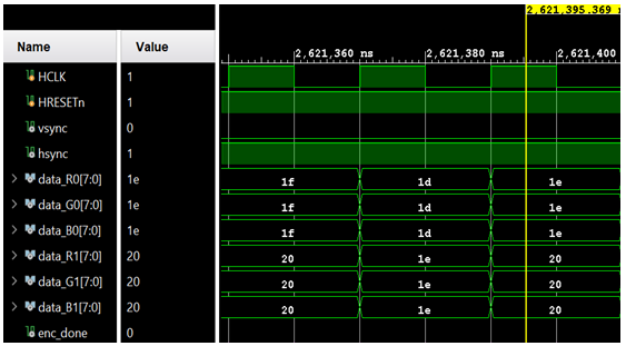


Figure 7. Simulation Results of Inverted Image

Figure 8. Inversion of the Image

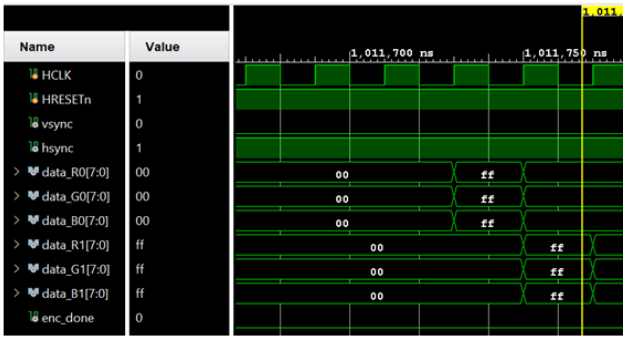


Figure 9. Threshold operation simulation

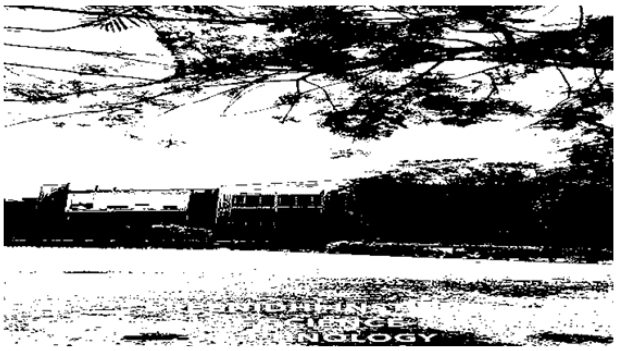


Figure 10. Threshold Operation Results

Zynq7 Processing System has to be interfaced with the peripheral module OLEDrgb IP. In the Zynq architecture-based designs, the input and output devices are interfaced with an AXI interconnect. There are three types of AXI interconnects available, depending on the requirement. Those are AXI Lite, AXI Stream, and AXI Full. In the proposed system, AXI-Lite interconnect is selected. This is used to interface the OLEDrgb PMOD IP with ZPS via the serial peripheral interface protocol connected through the AXI interconnect. Therefore, this AXI interconnect is configured as a slave AXI interconnect to the master ZPS. This ZPS is configured with DDR, Fixed_IO, M_AXI_GP0, one FCLK_CLK0, FCLK_RESET0_N, and M_AXI_GP0_ACLK in order to connect the other blocks such as Processor System Reset and OLEDrgb through the AXI interconnect as illustrated in Figure 11. The internal architecture of ZPS is an All Programmable SoC (APSOC) and is a sandwich of processing system and programmable logic [16]. It can interface various I/O peripherals with the related interface protocols like USB, GigE, SDIO, CAN, I2C and SPI. In our proposed design architecture, we have interfaced one peripheral OLEDrgb is interfaced through SPI. Therefore, one UART is enabled and rest of all the peripheral interfaces are disabled in the ZPS. The OLEDrgb is connected to the output port JA. This block design is a semicustom design using ZPS, AXI interconnect in slave mode, and processor reset blocks. After the completion of the block design, ”Run Block Automation”

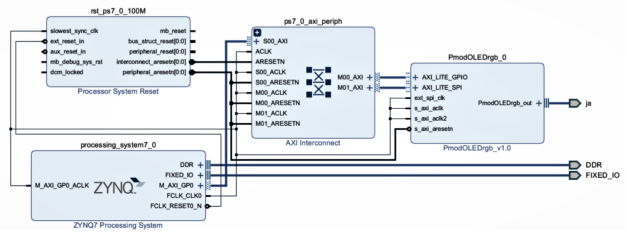


Figure 11. Block Design with Zynq SoC and OLEDrgb

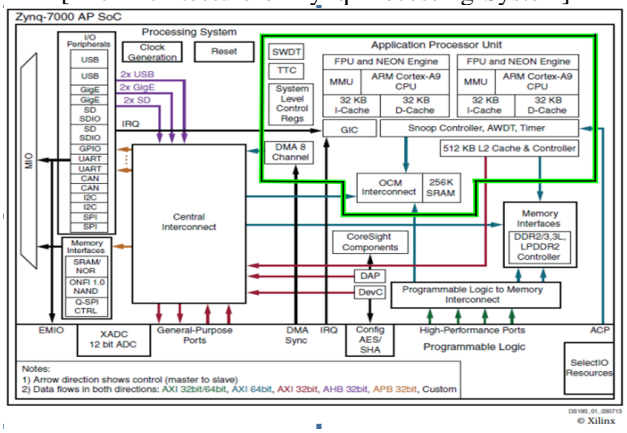
and validation of the design steps are to be performed. The internal architecture of ZPS is illustrated in Figure 12.a. It is an All Programmable SoC (APSOC). It is a sandwich of processing system and programmable logic[16] as shown in Figure 12.b. It can interface various I/O peripherals with the related interface protocols like USB, GigE, SDIO, CAN, I2C and SPI. In our proposed design architecture, we have interfaced one peripheral OLEDrgb is interfaced through SPI. Therefore one UART is enabled and rest of all the peripheral interfaces are disabled in the ZPS.

The complex hardware design is programmed in Artix7 FPGA known as programmable logic and the complete application software is developed using Software Development Kit (SDK) software and programmed on processing system as illustrated in Figure 12.c. Here, dual core ARM cortex 9 processor is used as a processing system. In this Zynq architecture based designs, GPIOs, external peripherals, memory and SD card all are interfaced through AXI interconnect only.
In the proposed design, OLEDrgb is connected to the output port JA. This block design is a semi-custom design using ZPS, AXI interconnect in slave mode, and processor reset blocks. After the completion of the block design, ”Run Block Automation” and validation of the design steps are to be performed. After the validation process, an HDL wrapper will be generated by using the appropriate option in the tool. The next processing steps are elaboration, synthesis, implementation, and bit stream generation. The elaboration process converts the HDL wrapper into an RTL schematic, as illustrated in Figure 13. Synthesis generates logic gate level net-list, utilisation, power and timing reports. The utilization report demonstrates the 2.23% of slice LUTs,1.45% of slice registers, 7% of Bonded IO blocks, 100% of Bonded IO pads and 3.13% of BUFGC-TRL (Buffer Control pins) as illustrated in Figure 14.a.

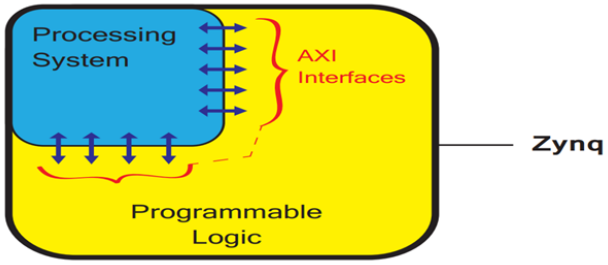
The power report of the proposed design is illustrated in Figure 14.b.Total on-chip power is 1.636 W, out of which PS7 consumption is 1.527 W, very less amount ¡1% of power (0.007W) is due to Signals, logic and IOs and the rest of 0.101 W is due to device static power.

The timing report generated in the synthesis process is illustrated in the Figure 15. As there is no negative slack in neither setup and hold nor pulse width, the constraints are met satisfactorily. After the verification of synthesis reports,

[The Architecture of Zynq Processing System]



[A simplified model of the Zynq architecture]



[Relationship of the software-hardware and Zynq architecture]

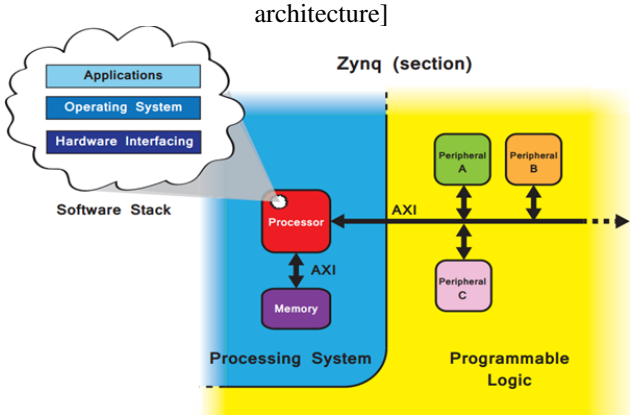


Figure 12. The Architecture details of Zynq Processing System

implementation process and Bitstream generation will be followed. If all the constraints are satisfied, then implementation will be completed without errors and Bitstream generation will be successful.

6. RESULTS

After the successful generation of the Bitsream of the proposed design which is to be programmed in the programmable logic known as Artix7 FPGA available in Zed board. The task to be done by the design is processed by processing system which is dual core ARM9 available in this board.

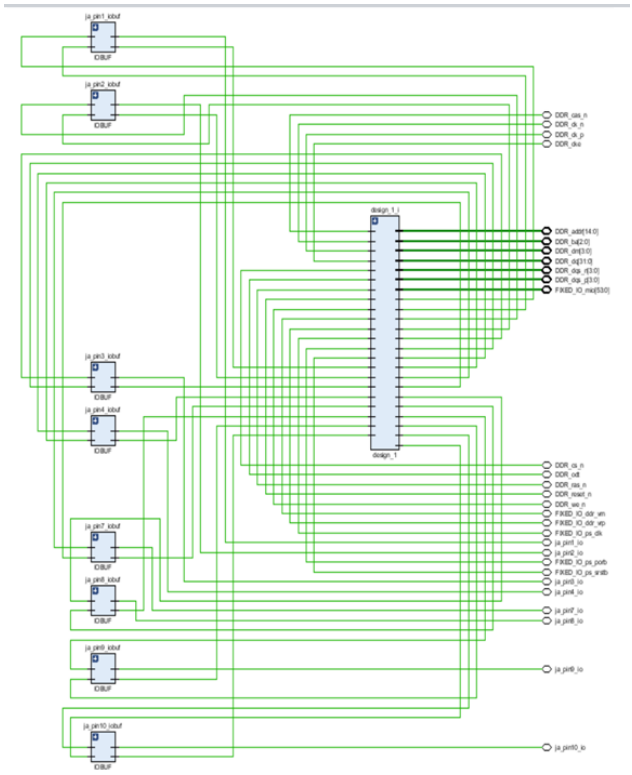


Figure 13. The RTL Design after Elaboration

[The Utilization Report]

Name	Slice LUTs (53200)	Slice Registers (106400)	Bonded IOB (200)	Bonded IOPADs (130)	BUFGCTRL (32)
Oled_design_1_wrapper	1186	1545	14	130	1
Oled_design_1_i (Oled_design_1)	1186	1545	0	0	1
PmodOLEDrgb_0 (Oled_desig...	512	775	0	0	0
processing_system7_0 (Oled_...	112	0	0	0	1
ps7_0_axi_periph (Oled_desig...	543	730	0	0	0
s00_couplers (s00_coupler...	420	600	0	0	0
auto_pc (Oled_design_1...	420	600	0	0	0
xbar (Oled_design_1_xbar_0)	123	130	0	0	0
inst (Oled_design_1_xba...	123	130	0	0	0
rst_ps7_0_100M (Oled_design...	19	40	0	0	0

[Power Report]

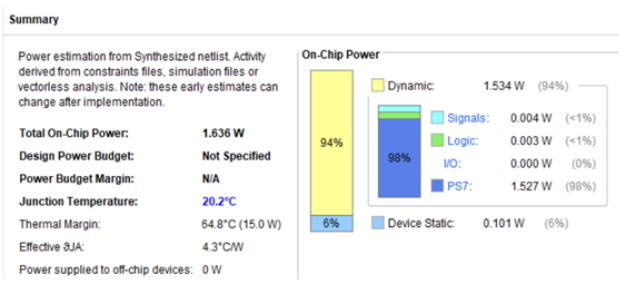


Figure 14. The Synthesis Reports: Utilization and Power

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	3.110 ns	Worst Hold Slack (WHS):	0.025 ns	Worst Pulse Width Slack (WPWS):	3.750 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	2972	Total Number of Endpoints:	2972	Total Number of Endpoints:	1495

All user specified timing constraints are met.

Figure 15. Timing Report of the Proposed Design

Figure 16. Wel-Come message display

These two steps are performed in SDK environment. Therefore, Hardware Design will be exported including the generated bit stream file to SDK tool and then launch it. At SDK, create new application project, and name the application project. Here it is named as OLEDrgbdemo. Now, create or add Bitmap.h file to display the specific image whose size will be 96X64 Pixels. Because, in the proposed research, a tiny PMOD OLEDrgb is used as a display device that can support the 96X64 pixels.

As we have connected port JA to the OLEDrgb in the block design, we need to connect the OLEDrgb to JA port of the Zed board and make a connection of the working system with it via two USB cables. One is for JTAG connectivity and the other is for programming purpose. Once hardware configuration is completed, power on the Zed board. Then create one application project in the SDK along with application source file either in C or CPP. Then develop the board supporting package file .bsp in the same environment. Once save all the created files, then program FPGA.

Here, the application is developed to display text information as “WelCome to VLSI_Lab OLEDrgb_Demo”, followed by desert rose or also known as Adenium flower will be displayed with image compression.

To verify the results in real time, run application project and launch on hardware system debugger.Then Welcome message is observed on OLEDrgb display as shown in Figure 16. The original Adenium flower image is illustrated

Figure 17. Original Image before compression

Figure 18. Image display on OLEDrgb after compression

in Figure 17 as specified in the application software and compressed Adenium flower image is displayed on OLEDrgb as shown in Figure 18.

7. Conclusion and Future Work

key advantages of our research are the flexibility, cost effective, high re configurable embedded application demonstrated on image processing opera- tions with Zynq SoC and Xilinx Vivado System Design Suite and SDK software tools. The case study of this research work successfully demonstrated the simulation and emulation results as well. Even though the proposed work has been demonstrated with sample personnel images, it is a better suitable solution for many real time embedded applications in our daily life. In the proposed paper, a few image processing operations such as brightness variations, inversion, and threshold operation of the input image are demonstrated clearly. However, many other operations can be verified without modification of the hardware design. This is achieved because of the high re configurable FPGA and an efficient ARM processing system.

Acknowledgment

The authors of this paper express their gratitude to the management of Sreenidhi Institute of Science and Technology for the continuous encouragement and support in doing the research work for the teaching faculty and students.

References

[1] M. I. AlAli, K. M. Mhaidat, and I. A. Aljarrah, "Implementing image processing algorithms in fpga hardware," *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, vol. doi: 10.1109/AEECT.2013.6716446., pp. 1–5, 2013.

[2] B. V. B. S. R. P. V. D. K. Dhanabal R, S. K. Sahoo and V. Divya, "Fpga based image processing unit," *IEEE 9th International Conference on Intelligent Systems and Control*, pp. 1–4, 2015.

[3] I. Chiuchisan, "A new fpga-based real-time configurable system for medical image processing," *E-Health and Bioengineering Conference (EHB), Iasi, Romania*, pp. 1–4, 2013.

[4] N. H. J. A. Ahmad and A. Amira, "Fpga-based implementation of 3-d daubechies for medical image compression," *2019 Third International Conference on Inventive Systems and Control (ICISC)*, pp. 683–688, 2012.

[5] M. Veldurthi and A. D. Dharmavaram, "Automatic vehicle identification and recognition using cnn implemented on pynq board," *6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India*, pp. 1302–1306, 2022.

[6] S. T. B. M. Devi, D.A., "Monitoring and controlling of icu environmental status with wifi network implementation on zynq soc," *Inventive Systems and Control Lecture Notes in Networks and Systems*, vol. 204, pp. 643–655, 2021.

[7] P. Kokotis and J. Vourvoulakis, "Acceleration of image processing algorithms based on a single board computer and fpga co-design," *11th International Conference on Modern Circuits and Systems Technologies*, pp. 1–5, 2022.

[8] D. A. Devi, T. S. Savithri, and S. Sugun.L, "Design and implementation of real time data acquisition system using reconfigurable soc," *International Journal of Advanced Computer Science and Applications*, vol. 11(9), pp. 325—-331, 2020.

[9] P. D. K. A. K. B. A. Bouridane, D. Crookes, "A high level fpga-based abstract machine for image processing," *Journal of Systems Architecture*, vol. 45, pp. 809–824, 1999.

[10] P. Soma and R. K. Jatoth, "Hardware implementation issues on image processing algorithms," *4th International Conference on Computing Communication and Automation*, vol. doi: 10.1109/CCAA.2018.8777564., pp. 1–6, 2018.

[11] M. M. Telnaz Zarifi, "Fpga implementation of image processing technique for blood samples characterization," *Computers Electrical Engineering*, vol. 40, pp. 1750–1757, 2013.

[12] C. Wang and S. Zhu, "A design of fpga-based system for image processing," *Review of Computer Engineering Studies*, vol. 2, pp. 25–30, 2015.

[13] Y. Wei and H. Xiao, "Research and hardware design of image processing algorithm based on fpga," *Journal of Physics*, vol. 10.1088/1742-6596/1648/4/042104, pp. 1–7, 2020.

[14] P. Li, "Fpga-based hardware implementation of image processing algorithms for real-time vehicle detection applications," *University of Minnesota*, 2012.

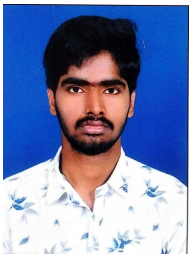
[15] V. S. Surwase and S.N.Pawar, "Vlsi implementation of image processing algorithms on fpga," *International Journal of Electronic and Electrical Engineering*, vol. 3, p. 139—145, 2010.

[16] M. A. E. L. H. Crockett, R. A. Elliot and R. W. Stewart, "The zynq book: Embedded processing with the arm cortexa9 on the xilinx zynq-7000 all programmable soc," *First Edition*, 2014.

Author Details

Dharmavaram Asha Devi Senior Member IEEE, Fellow IETE and Fellow IE. The Author is working as a Professor in the Department of Electronics Communication Engineering, SNIST, Hyderabad. She became an Associate Member of IETE in Dec-2000 from the Institute of Electronics and Telecommunication Engineering, New Delhi, India. She awarded M.Tech. Digital Systems and Computer Electronics from JNTU Anantapur in 2005. She achieved a Doctorate Degree from Sri Krishnadevaraya University, Anantapur in Dec 2011, for her research contribution in Embedded System based Soil Analysis under the supervision of Prof. K. Malakondaiah, from SKU, Anantapur. She had 18 years of academic, including 12 Years of R & D Experience. She has published more than 25 papers in International Journals and presented 40 research papers in National and International level conferences on behalf of her research work done in Embedded Systems and VLSI Designs. Her area of research is SoC designs with FPGAs, Low power VLSI designs for embedded applications.

Niharika Reddy Kathula The author is UG research scholar in Electronics and Communications Engineering from Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India. She is also an IEEE student member of EDS, Communication society and Sensor Council. She did an internal RD project titled "Pranayaan-Low Cost Ventilator" in her second semester. This project received a fund of 400 USD from IEEE region 10 under the special call for proposals related to Covid-19 in the academic year 2021-2022. She also received a Silver medal in 1 IDEA 1 WORLD (1I1W) International Innovation Design Start-up competition in February 2021. She is also the President of The Electronix Club (TEC) and conducted various technical events and talks in her college. She did two internships on AI-ML from AICTE and one internship on Digital Design Through Verilog from Microchip Academy. Her area of interest on research lies in VLSI Design, SOC architectures, and AI-ML applications.



Gopinath Kalluri The author is an undergraduate scholar in Electronics and Communications Engineering from SNIST, completed his 6-Months internship from VEDAIIT in VLSI Domain and now he is working at AMD. He worked as a project head for the ELECTRONIX CLUB - SNIST, one of the prominent clubs of the ECE Department where he was a part of numerous club projects. He is also a student member of IEEE. He worked on a project entitled image processing on Zynq SoC. In his 4th semester. He did an internship at Microchip Academy. His area of interest lies in VLSI design verification and SoC design and verification and Embedded Systems.

Leela Sai Bondalapati The author is an undergraduate scholar in Electronics and Communications from Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India. He is also a student member in Cisco Networking Academy. He also completed the Introduction to Networks course in Netacad. He was also the Organizer of The Cisco Networking Club - SNIST and conducted various Technical Workshops in his college. He cleared Preliminary and Mains test for Industrial Ready Professional Training course in VLSI Design at VEDA IIT, Hyderabad. His area of interest and research lies in VLSI Design, Chip design and Computer Networks.