An FPGA Synthesis of Face Detection Algorithm using HAAR Classifier

Archit Gajjar[1], Xiaokun Yang[1], Lei Wu[1],
Hakduran Koc[1],
Ishaq Unwala[1], Yunxiang Zhang[1]
University of Houston-Clear Lake[1]
2700 Bay Area Blvd, Houston, TX77058, USA
{GajjarA7402, YangX, WuL, KocHakduran,
Unwala, ZhangY0552}@uhcl.edu

Yi Feng[2]
Algoma University[2]
1520 Queen Street East Sault Ste. Marie,
ON P6A 2G4, Canada
Feng@algomau.cau

ABSTRACT

This paper presents a synthesis of well-known Viola-Jones face detection algorithm on Xilinx software and platform - Vivado and field programmable gate array (FPGA) as Nexys 4 Artix-7 device. Compared with the prior work on the Altera platform proposed in [1], our work reduces the slice count by 1018. And additionally, the power consumption of the implementation is 714 mW, including 15% as the static cost and 85% as the dynamic power dissipation.

Furthermore, the design details of the components of the structure, such as generation of integral image, multiple pipelined classifiers, as well as the parallel processing, are discussed in this work, in order to provide a potential improvement for the future work. This paper not only provides successful synthesis of a face detection system but also ignites intriguing ideas in terms of improvement aspects, such as approximating the design for finding an optimal energy-quality tradeoff corresponding to different applications as our future work.

CCS Concepts

• Hardware→ Application specific integrated circuit

Keywords

Face detection; field programmable gate array (FPGA); Viola-Jones algorithm.

1. INTRODUCTION

Due to the benefits of face/object detection/recognition [2][3][4], it plays an important role in our daily life. Although, with such surveillance privacy is compromised, the benefits over rule the denials of the system.

On a second note, the application-specific computation on field programmable gate arrays (FPGA) is usually higher efficiency than that on software like general MCU or embedded systems, due to the hardware parallelism and specific design. Such scenario has already been proved by Microsoft for their search engine

"Bing" [5][6]. Briefly, they offloaded some complex computation on pure hardware platforms by reconfiguring traditional servers with multiple grids of FPGAs. At the end, the latency was reduced with throughput of almost ×2.25.

Under this consideration, Nexys 4 FPGA is adopted in our work to implement the famous Viola-Jones face detection algorithm [7], in order to find an optimal tradeoff between quality and speed-energy constrains. The main contributions are:

- First of all, we offer an implementation of Viola-Jones algorithm with Very High Speed Integrated Circuit Hardware Description Language (VHDL) on Nexys 4 Artix-7 FPGA using Vivado v2017.4. This work is derived from the existing prototype which is fully functional demo of face detection using Altera DE2-115 FPGA [1].

- The main contribution of our work is to provide an alternative version of face detection demo with the implementation on Xilinx by using Vivado and Nexys 4 FPGA. Except for the data path of the Viola-Jones algorithm, we created memory blocks including RAMs and ROMs, 44-bit signed comparator, clock module, etc. Experimental results show that the slice count of our work on Xilinx is reduced by 1018 compared with the prior work on Altera in [1].

- The preliminary results, a demonstration of OV7670 camera-VGA monitor displaying, have been presented in our previous work [8]. The existing demonstration is developed based on the Basys 3 with Artix-7 which is an entry-level FPGA board, and is available on http://sceweb.sce.uhcl.edu/xiaokun/# -> project as open source code. In this study, not only do we synthesize the Viola-Jones design with VHDL, but we also evaluate the FPGA performance in terms of slice count and power cost by using the estimation methodology in [9].

- The details of the implementation have been discussed in this work, in order to offer a potential of optimizations as our future work like [10][11][12]. The improvement of approximate designs on this platform will be coming soon: by providing the high-performance architectures [13][14] and inaccurate designs on the computation components such as [15], it is able to find an optimal energy-quality tradeoff corresponding to different applications [16].

The organization of this paper is as follows: section 2 briefly introduces the background and challenges of face detection, and section 3 reviews the related works of our proposed design. In section 4, the design architecture and details of implementation are explained. Finally, section 5 concludes this paper.

2. BACKGROUND AND CHALLENGES OF FACE DETECTION

Face detection simply means, to identify if provided input, whether it be image or video, contains face in it or not. It is an old concept and has grown in huge manner. Yet, is still does not provide full satisfactory results due to different factors involved in the process of detecting face such as pose variation, feature occlusion, imaging condition and what not [4].

- Pose Variation - The ideal scenario would be one in which only frontal images are involved. Unlikely, this is not the usual case in the real case.

- Feature Occlusion – The presence of features like spectacles, beards, cap etc. can be hindrance in the face detection.

- Facial Expression – Facial features may vary greatly due to different facial gestures.

- Imaging Conditions – Different cameras and ambient conditions can affect the quality of image which can eventually, create problems in detecting faces.

Face detection using FPGA is not an easy task and there are not globally accepted grouping criteria. There are different detection methods while keeping in the mind about the various scenarios.

- Controlled Environment – It is most straightforward case. Photographs are taken under the controlled light, background, facial pose and angle etc.

- Color Images – Typically, skin color is used to find faces. But the drawback would be if light conditions are weak, this does not work quite well [14].

- Images in Motion – Real-time videos give the chance to use motion detection to localize faces. Nowadays, most commercial systems must locate faces in videos. There continuing challenge to achieve the best detecting results with best performance, possibly.

3. PRIOR WORK

In initial phase of this project, we have already developed a platform which is able to show provided input on monitor via VGA interface recorded by camera [8]. The project was, fundamentally, implementation on the integration of camera OV7670 and monitor with VGA interface.

There were numerous implementations based on Zed board zynq-7000 and Terasic DE2-115 development boards such as [17][18], with a big memory to buffer the 640x480 RGB image, where else our prior platform in [8] employs the entry-level Basys 3 (Artix-7) with limited memory resource (suitable for the 320x240 RGB standard).

The main challenges are: 1) the resolution configuration button provided by the original code doesn't work on the Basys 3 board; 2) the display image is incomplete due to limited buffer size. Therefore, we reconfigure the functional registers and finally make it work on Basys 3. We also presented 12 approximates of Red-Green-Blue (RGB) to grayscale converters, in order to provide different energy-speed requirements with different quality constrains.

This paper is entirely based on our previous OV7670-VGA platform of Xilinx implementation in [8]. Furthermore, the open source code in [1], which is a prototype on Altera platform, is adopted and optimized for extending the work of face detection available on Xilinx.

4. APPROACHED METHOD & IMPLEMENTATION

In this section, the design architecture on FPGA is introduced, and the details of the implementation is further explained as well.

4.1 Hardware Architecture

Before moving onto the design, let us explain about the architecture which would make the understanding of design much easier. As shown in Fig. 1, the design can be separated in three main components.

First, the OV7670 camera module consists controller and capture units which is configured by the top module. Second, where the Viola-Jones face detection algorithm is implemented. Output generated by the algorithm would be sent through the last but not the least part of the system, VGA interface and monitor.

4.2 Details of Implementation

As per Fig. 1, the input image/video is being captured by OV7670 camera module which is 12-bit RGB, 4-bit for each color. Data captured by camera is in 320×240 pixel resolution by default. Here, camera is manipulated by the top module which contains different capture modes as required and capture button to take a snapshot of the image. Later, that image is stored in the image buffer.

The stored data in the image buffer is used to generate integral image which speeds up the processing time and efficiently produces the sum of the values, rather pixels in this case, in rectangular subset. Note that, before producing integral image, input, 12-bits RGB colored image is converted into 8-bits grayscale image. Not only integral image but integral image square is also generated in the followed up step. Each integral image possesses dual buffers containing data read through memory. Word size for integral image is 21-bits ($16 \times$word size) where else integral image square is 29-bits ($16 \times$word size). Addressed chunks enable routing of any data set ($16 \times$word size) to mux which are followed by buffer.

The method was introduced by Viola-Jones face detection algorithm [17]. To find an integral image, you need to take sum of all pixel values that are on left and top side. Considering the example from image, to find value for 2, you need to sum 9+5+4+2 which gives 20. Seeing, integral image, while using formula $D = (S+P) - (Q+R)$, one would be able to find pixel value at the same place from original image. This saves a lot of processing cost and decreases the computation time of the system.
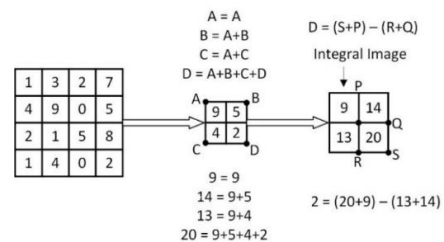

Figure 1 : Integral Image Example

There is a sub module in the main algorithm which is subwindow_top. There are 16 sub windows processing in parallel on the data chunks provided from the addresses. Now, these 16 sub windows described as sw[0], sw[1], sw[2] up to sw[15]

classifiers make decision whether the provided image contains any face/faces. In more detail, theses classifiers create weak classifiers first. Weak classifiers would only be considered as strong classifiers once they pass weak threshold. Furthermore, strong classifiers are compared with strong threshold value. If the value for strong classifier is greater than strong threshold value, system consider that the image has face in it. More details about the threshold are explicitly mentioned in the Viola-Jones face detection algorithm [2].
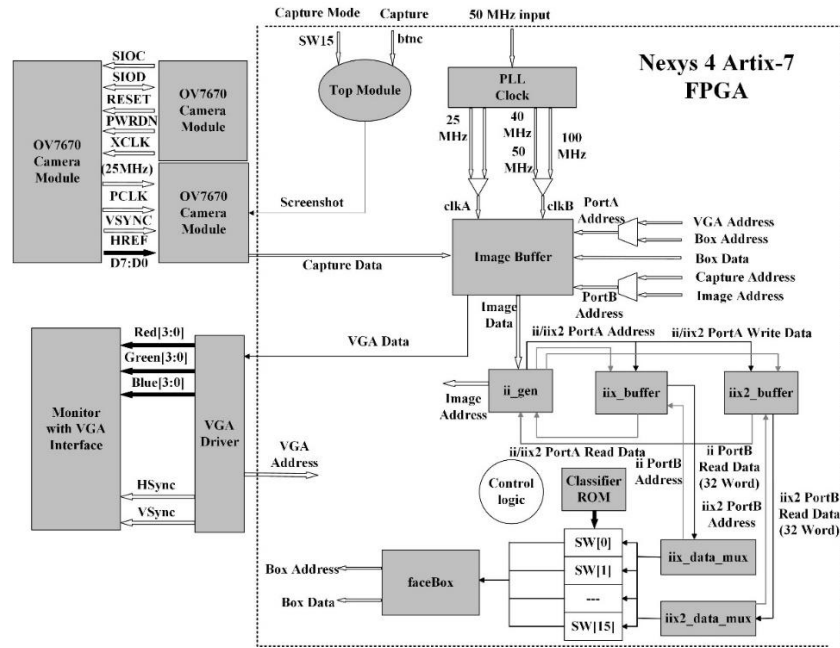

Figure 2 : Hardware Architecture

Once we get the positive values from the strong classifiers or in other words those strong classifiers which are beyond the strong threshold values, FaceBox will be fetched those data and draw rectangles around all the detected faces. This image is displayed on the monitor since data was fetched to image buffer from faceBox via VGA interface. The exhibited output is 12-bits RGB colored image with rectangles around the detected faces.

5. DESIGN FLOW AND EXPERIMENTAL RESULTS

In this section, we present the design-under-test (DUT) and evaluate the performance using a performance evaluation methodology [9].

We employ Mentor Graphic ModelSim 10.4d as the simulator in our study. After simulation we can obtain the waveforms (VCD) with switching activities of signals, IOs, and logics. And after the synthesis the gate-level net lists (NCD) is able to be collected. These files are needed to analyze the power consumption using XPower Analyzer (XPA). Finally, Xilinx Vivado is used as the synthesis tool with the target device Nexys 4 Artix-7 FPGA.

5.1 FPGA Resource Cost

Using Vivado 2017.4 and the constraint file with the Nexys 4 Artix-7 FPGA board, we synthesized the code. In our design we use 32,309 Look Up Tables (LUTs) which is 50.96% of the LUTs available on the FPGA board. Compared to the implementation in [1] where the number is 33,327, our proposed work reduces the number of LUT by 1018. In addition, the D flip-flop cost in our work is 38130, which is 30.07% of the total resource provided.

Moreover, we used 1 Phase-Locked Loop (PLL) out of 6 which is similar to the other design, and 35 DSPs employed as some complex computation components like variances and floating-point multiplications.

Since Nexys 4 Artix-7 FPGA does not have ADV7123 Digital to Analog Converter (DAC), used IO pins are quite less, which are 41 compared to [1] where their total number of pins is 56. Such change does not make a bigger difference in terms of quality of results which was reduced to 12-bit VGA output instead 24-bit VGA output, however, the reduced number of IOs is able to lower the switching activities of signals and logics, resulting in lower power consumption on FPGA development.

Table 1. Summary of Resource utilization

Resource	Utilization	Available	Utilization %
LUT	32309	63400	50.96
LUTRAM	644	19000	3.39
FF	38130	126800	30.07
BRAM	94	135	69.63
DSP	35	240	14.58
IO	41	210	19.52
MMCM	1	6	16.67

Fig. 2 depicts the resource cost as a percentage of the entire resource on the board, in such a way a clear results are able to be described. The figure is generated by Vivado.
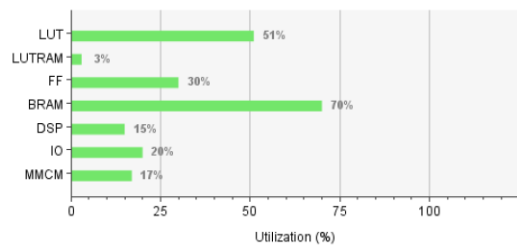
Figure 3 : Resource Utilization Graph

5.2 Power Consumption

Fig. 4 demonstrates the abbreviated and graphical breakdown of power estimation report generated by the power analyzer. It can be observed that the static power cost is 15% and the majority proportion of the total power is dissipated by the dynamic power. For our design, static power consumption is 104 mW and dynamic power cost is 610 mW with total power consumption of 714 mW. Since the power cost has not been estimated in [1], it is not able to compare the power dissipation with the prior work between the two different platform.

The dynamic power is mainly affected by the toggle rates of clock, signals, logics, and IOs, etc. More specifically, the switching activities of signals, logics, and BRAMs cost 22%, 23%, and 24% of the dynamic power dissipations, respectively. And the remaining power consumption is mainly come from the mixed-mode clock manager module (MMCM) and clock input, totally 30% of the dynamic power cost. Due to the small number of IO and DSP utilizations, the power cost is very low (around 1%) compared to the aforementioned FPGA components.
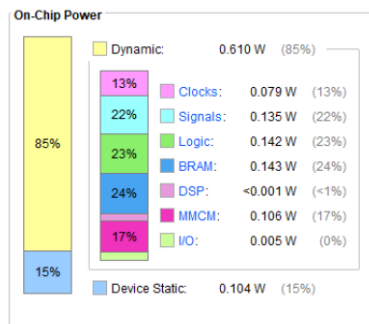

Figure 4 : Power Estimation

6. CONCLUSION

Based on the existing implement in [1] and our prior work in [8], not only do we offer an alternative prototype on the Viola-Jones face detection algorithm with FPGA development, but we also evaluate the computation performance in terms of slice cost and power consumption. Compared to [1], our proposed work on Xilinx platform achieves a reduction on slice count by 1018, and costs a typical power consumption as 104 mW for static and 610 mW for dynamic dissipation.

Since the prototype is in the emerging artificial intelligent field, it is likely to create a big impact to the design of such application-specific integrated circuit for offering low-latency and inexpensive computations. We firmly believe that this project has certain potential improvement, which will be our future work. Since the project demonstrates such potential enhancement our next step is to target probable growth areas in the system.

7. REFERENCES

[1] P. Irgens, C. Bader, Theresa L, D. Saxena, C. Ababei, "an efficient and cost effective FPGA based implementation of the Viola-Jones face detection algorithm," *HardwareX*, (1), 2017, pp 68-75. https://doi.org/10.1016/j.ohx.2017.03.002

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. CVPR 2001, 2001, pp. I-511-I-518 vol.1. DOI >10.1109/CVPR.2001.990517.

[3] H. He, L. Wu, X. Yang, et al, "Dual Long Short-Term Memory Networks for Sub-Character Representation Learning," The 15th Intl. Conference on Information Technology-New Generations (ITNG), PP. 1-6, Jan. 2018.

[4] M-H Yang, D. Kriegman, N. Ahuja, "Detecting Faces in Images: A Survey", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 1, January 2002.

[5] A. M. Caulfield, E. S. Chung, A. Putnam, etc., "A Cloud-Scale Acceleration Architecture," 2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), PP: 506-511, Oct. 2016.

[6] A. Putnam, A. M. Caulfield, E. S. Chugn, etc., "A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services," IEEE MICRO, Vol. 35, No. 3, PP. 10-22, May 2015.

[7] Nexys 4 Artix-7 FPGA Trainer Board, 2017. Available at (https://reference.digilentinc.com/_media/nexys4-ddr:nexys4ddr_rm.pdf).

[8] Y. Zhang, X. Yang, L. Wu, K. Sha, etc., "Exploring Slice-Energy Saving on An Video Processing FPGA Platform with Approximate Computing", *Intl. Conf. on Algorithms, Computing and Systems (ICACS2018), Accepted, In Press, 2018*

[9] X. Yang, N. Wu, and J. Andrian, "A Novel Bus Transfer Mode: Block Transfer and A Performance Evaluation Methodology," Elsevier, Integration, the VLSI Journal, Vol. 52, PP. 23-33, Jan. 2016. DOI=10.1016/j.vlsi.2015.07.012

[10] X. Yang, N. Wu, and J. Andrian, "Comparative Power Analysis of An Adaptive Bus Encoding Method on The MBUS Structure," Journal of VLSI Design, Vol. 2017, Article ID 4914301, PP. 1-7, May 2017. DOI=10.1155/2017/4914301

[11] X. Yang, W. Wen, and M. Fan, "Improving AES Core Performance via An Advanced IBUS Protocol," ACM Journal on Emerging Technologies in Computing (ACM JETC), Vol. 14, No. 1, PP. 61-63, Jan. 2018. DOI=10.1145/3110713

[12] X. Yang and W. Wen, "Design of A Pre-Scheduled Data Bus (DBUS) for Advanced Encryption Standard (AES) Encrypted System-on-Chips (SoCs)," The 22nd Asia and South Pacific Design Automation Conference (ASP-DAC 2017), PP. 1-6, Chiba, Japan, Feb. 2017. DOI=10.1109/ASPDAC.2017.7858373

[13] X. Yang and J. Andrian, "A High Performance On-Chip Bus (MSBUS) Design and Verification," IEEE Trans. Very Large Scale Integr. Syst. (TVLSI), Vol. 23, Issue: 7, PP. 1350-1354, Sept. 2015. DOI> 10.1109/TVLSI.2014.2334351

[14] X. Yang and J. Andrian, "A Low-Cost and High-Performance Embedded System Architecture and An Evaluation Methodology," IEEE Compt. Society Annual Symposium on VLSI (ISVLSI), PP. 240-243, Sept. 2014.

[15] Y. Zhang, X. Yang, and L. Wu, et al, "Hierarchical Synthesis of Approximate Multiplier Design for Field-Programmable Gate Arrays (FPGA)-CSRmesh System," *Intl. Journal of Compt. Applications* (IJCA), Vol. 180, No. 17 PP. 1-7, Feb. 2018. DOI> 10.5120/ijca2018916380

[16] M. Fan, Q. Han, and X. Yang, "Energy Minimization for On-Line Real-Time Scheduling with Reliability Awareness," Elsevier Journal of Systems and Software (JSS), Vol. 127, PP. 168–176, May 2017. DOI=10.1016/j.jss.2017.02.004

[17] P. Viola, M. Jones, "Robust Real-Time Face Detection", Intl. Journal of Computer Vision, Vol. 57, PP.137-154, May 2004.

[18] J. Matai, A. Irturk and R. Kastner, "Design and Implementation of an FPGA-Based Real-Time Face Recognition System," *2011 IEEE 19th Annual International Symposium on Field-Programmable Custom Computing Machines*, Salt Lake City, UT, 2011, pp. 97-100. DOI >10.1109/FCCM.2011.53.