

PERFORMANCE EVALUATION OF FPGA ACCELERATOR FOR VISION BASED FIRE DETECTION SYSTEM

Abubakar Hassan, Christopher Umerah Ngene and Peter Yusuf Dibal

Department of Computer Engineering, University of Maiduguri, Maiduguri, P.M.B 1069, Nigeria
Corresponding Author: abubakarhassan@unimaid.edu.ng

ABSTRACT

Fire disasters are endangering human lives and pose a great threat to economic infrastructure, and the environment around the world. As a result, there is need for robust early detection of fire outbreaks to minimize loss of human lives, protect properties and the ecosystem from great damages. Conventional fire detection systems are based on electronic sensors. As a result, they suffer from the problems of transport delays, conduction delays, limited detection range, high false alarm, and not suitable for outdoor applications. Image processing and computer vision techniques for fire detection have been proposed in recent years to address the limitations of such sensor-based systems due to technological advances in electronics and computer science. However, most of these computer vision-based fire detection solutions are implemented in software platforms and as result, they have shortcomings of inefficiency, high hardware requirements and high cost. In this research study, a vision fire detection accelerator based on Field Programmable Gate Array (FPGA) was developed. MATLAB R2021a software was used for decoding the image dataset into pixel stream data. The design was captured in very high-speed integrated circuit HDL (VHDL). The design was synthesized with Xilinx Vivado 2021 design suite and simulated with Xilinx ISIMI. It has been shown that the design has achieved better resource utilization and power consumption compared to a similar work (15% Look-up Table (LUT), 1% Digital Signal Processor (DSP), 24% Input-Output (IO), 92.062W Dynamic and 1.029W Device Static). The hardware accelerator which was developed as an Intellectual Property (IP) core can also be employed to speed up image processing and computer vision algorithms in embedded vision, smart camera and video analytics applications.

Keywords: Fire detection; FPGA; Performance Analysis; Image Processing; Hardware Accelerator.

1. INTRODUCTION

Fire has a vital role in human society. It is used for cooking, keeping warm, black smithing and other whole host of domestic and industrial applications by human beings. Despite its great benefits, Fire can be disastrous if left unchecked. Such kind of fire has the capability to lead to loss of human lives, physical and economic infrastructure damage, and serious environmental degradation. Fire is characterised by the chemicals it releases, heat it generates, smoke it emits, its colour, its motion and its geometric shape (Hassan & Audu, 2022; Rabi, 2022).

Fire disaster causes huge number of human deaths and massive property destruction. A great number of fire incidents were reported across Nigeria in recent times. For instance, the Kano State Fire Service records that 134 human lives and 635-million-naira assets were lost due to fire outbreaks Also available online at <https://www.bayerojet.com>

in the state between January 2021 to December 2021. Another fire event was also reported at Sabon Gari Market in Kano City in which properties and shops worth millions of Naira were lost and took more than 10 hours to be put out completely by the Kano State Fire Service team. Report from News Agency of Nigeria (NAN) on 23 May 2021, showed that not less than 64 people, including eight fire fighters, got different degrees of injuries from a petroleum tanker explosion at a filling station in Sharada area of Kano State (Rabi, 2022).

Conventional fire detection systems are electronic sensor based, which usually depend on infrared radiation, heat emitted, or chemicals and gases released by fire combustion (Solórzano et al., 2022; Wang et al., 2020). These sensor-based systems have several shortcomings which include

transport delay, conduction delay, limited detection range, high false alarm, inappropriate for outdoor applications and lack of capability to give rich additional information on location, growth rate, size of the fire incident (Hassan & Audu, 2022; Rabi, 2022). With advances in computing and electronics engineering, image processing and computer vision-based approaches to fire detection have been proposed in the literature to address the limitations of the conventional systems which are traditionally sensor-based. These image processing and computer vision-based approaches have advantages of larger coverage area, faster response, rich descriptive information, appropriate for both indoor and outdoor applications and relatively cheaper (Hassan & Audu, 2022b). However, majority of them are purely software-based solutions. As a result, they suffer from the problems of inefficiency, high cost and high resources requirement (Hassan & Audu, 2022).

Given the fact that computer vision applications involve the computation of a large number of repetitive operations over the entire input image to analyse the image contents and recover useful information. In addition, this task could be

complex and computationally intensive. An alternative solution to software implementation is design of specific hardware for image processing and computer vision applications to perform a high rate of operations per second. Advances in programmable logic devices have led to the development of Field programmable Gate Arrays (FPGAs) which allow integrating large number of programmable logic elements on a single chip. The size and speed of FPGAs are comparable to ASICs, but FPGAs are more flexible and have shorter design cycle. An FPGA is an array of logic blocks arranged in an infrastructure of interconnections, which can be programmed at three distinct levels: the function of the logic blocks, the interconnections between blocks and the inputs and outputs. An FPGA is programmable at hardware level, thus combining the advantages of both general-purpose processors and specialized circuits (Torres-Huitzil & Arias-Estrada, 2000). Therefore, this research study aims to develop FPGA based fire detection accelerator using simulation, synthesis and configuration tools of Xilinx Inc USA and MATLAB.

2. RELATED WORKS

With advances in electronic engineering and computer science, image processing and computer vision approaches to fire detection have been proposed to address the limitations of the conventional systems which are traditionally sensor-based. The image processing and computer vision-based systems have advantages of larger coverage area, faster response, rich descriptive information, appropriate for both indoor and outdoor applications and relatively cheaper (Hassan & Audu, 2022; Rabi, 2022). For example, authors in (Foggia et al., 2015) proposed fire detection algorithm employing a combination of experts based on colour, shape and flame motion information. The method has been tested on a broad database to evaluate its performance in terms of sensitivity and specificity. Experiment results obtained show the effectiveness of the approach. A robust algorithm based on information on colour and shape was proposed by (Rabi, 2022). The algorithm starts with loading video image database developed to detect presence or absence of fire in video frames. Background subtraction is used by this method to compare the current frame with the reference frame. When the outcome of the background subtraction is less than the set point, the frame is discarded, and the next frame is checked. But if the difference is equal or greater than the set point, the frame is then subjected to colour and shape test. This is achieved by employing a combined RGB colour space model and shape feature. The proposed algorithm was extremely robust in detecting fire compared to those algorithms based on colour or motion features. In the work of (Saeed et al.,

2020), a hybrid approach was presented that consists of an Adaboost MLP neural network model to forecast fire events. Subsequently, an Adaboost Local Binary Pattern (LBP) model was developed to extract the region of interest (ROIs) and finally a convolutional neural network for fire detection with videos and images captured from installed surveillance cameras. In (Xie et al., 2020), a method was proposed that employs motion-flicker based dynamic features and deep static features for video-based fire detection. First, the dynamic features are segmented by analyzing the differences in motion and flicker features between fire and non-fire objects in video scenes. Second, an adaptive lightweight convolutional neural network is proposed to segment the deep static features of fire. Finally, the dynamic and static features of fire are cascaded to develop a video-based fire detection system with enhanced efficiency in terms of accuracy and run time.

However, the majority of these image processing and computer vision methodologies proposed are purely software-based solutions (Kaabi et al., 2018; Majid et al., 2022; Oluwarotimi, 2022; Pujangga et al., 2019; J. Zhang et al., 2021). These software-based solutions have problems of high hardware requirement, high cost and inefficiency (Abid et al., 2021; Choudhuri & Roy, 2016; Liu et al., 2020). In this research study, an FPGA accelerator for vision-based fire detection using VHDL and MATLAB was developed to address the shortcomings of the software-based design of

image processing and computer vision approaches to fire detection.

3. METHODOLOGY

3.1. Introduction

This section describes the architectural design of the system in detail. The hardware architecture consists of three key components: image decoding, grayscale image conversion

and Sobel edge detection. Modular and hierarchical approach was adopted for this design and the model was captured using VHDL. The conceptual diagram of the system is depicted in Figure 1.

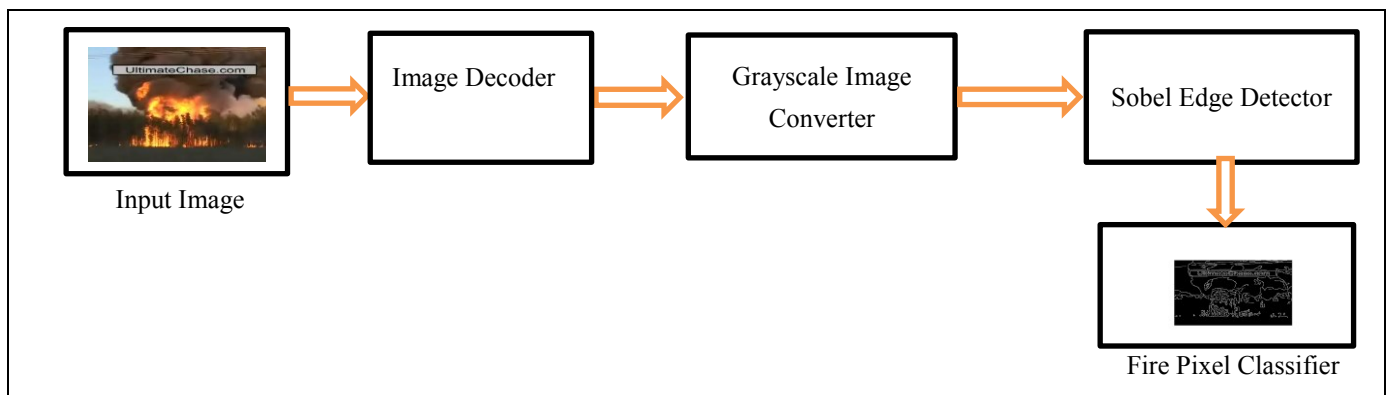


Figure 1. Conceptual Diagram of the Fire Detection Architecture

3.2. Image Decoding

Video dataset from VisiFire database was used in this research study. The VisiFire dataset is a commonly used public video dataset for fire and smoke detection. It consists of four categories of videos: flame, smoke, forest smoke and other video data. The fire set comprises 13 videos, 21 smoke videos, 21 forest smoke video set, and the other 2 videos. The dataset obtained was then converted into frames using MATLAB image processing and computer vision toolboxes. The pseudocode for the frame conversion in MATLAB environment is shown in Algorithm 1.

Algorithm 1: Video Dataset Conversion into Frames

Input: Dataset containing fire and non-fire videos

Output: Video frames

- 1 Import video which is to be converted into frames into the current MATLAB environment.
- 2 Extract the total number of frames in the video.
- 3 Make an empty directory named frames before the execution.
- 4 Run for loop and start extracting the frames into the directory.
- 5 end

We computed the respective average RGB values and decoded each video frame into pixel stream data which was fed into grayscale image converter. The coding was done in MATLAB environment. The pixel stream data is read from memory and fed into the grayscale image converter. The Algorithmic State Machine (ASM) chart for the image decoding into pixel data stream is shown in Figure 2. The ASM was coded in Vivado HLS 2021.1.

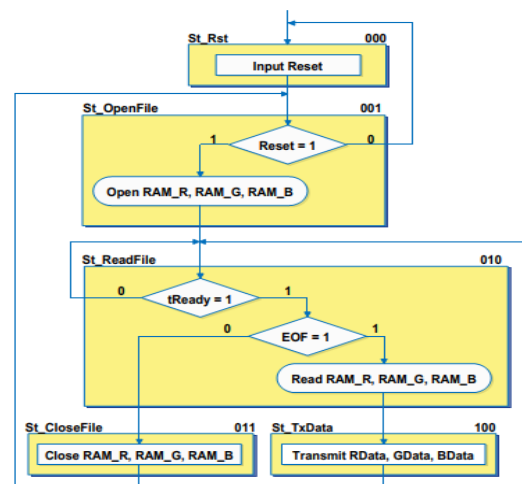


Figure 2. FSM chart for Image Pixel Generation.

3.3. Grayscale Image Conversion

The RGB image pixel data was converted to grey scale image for easy computation which is expressed as equation (1) as given in (Zhang, 2018) .

$$Y = 0.299R + 0.587G + 0.114B \quad \dots \quad (1)$$

For the hardware implementation, we avoided the floating-point multiplications and efficiently computed the grayscale value of the pixel as in equation (2):

$$Y = (306 \times R + 601 \times G + 117 \times B)/1024 \quad \dots \quad (2)$$

Figure 3 illustrates the architecture of the RGB to Gray scale image converter.

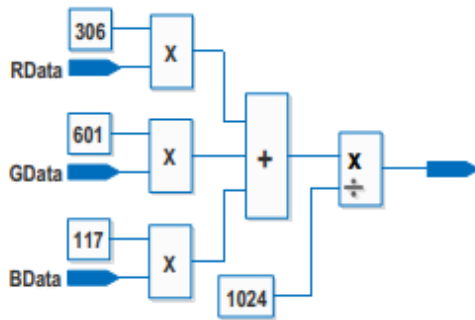


Figure 3. Architecture of the RGB to Grayscale Image Converter

3.4. Sobel Edge Detection

Sobel edge detection is a classical algorithm in the field of image and video processing for the extraction of object edges. Edge detection using Sobel operators works on the premise of computing an estimate of the first derivative of an image to extract edge information. By computing the x and y direction derivatives of a specific pixel against a neighbourhood of surrounding pixels, it is possible to extract the boundary between two distinct elements in an image.

The Sobel operator performs a 2-D spatial gradient measurement on an image and hence emphasizes regions of high spatial frequency that correspond to edges. Typically, it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image. The operator consists of a pair of 3 by 3 convolution kernels as shown in equation (3). One kernel is simply the other rotated by 90 degrees.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -2 \end{bmatrix} * A \quad \dots \quad (3)$$

Where A is the input image, G_x , G_y are images where each point represents the approximate derivatives along the x and y axes.

We assume that the original image is given by (Zhang, 2018) in equation (4).

$$\begin{bmatrix} f(x-1, y-1) & f(x, y-1) & f(x+1, y-1) \\ f(x-1, y) & f(x, y) & f(x+1, y) \\ f(x-1, y+1) & f(x, y+1) & f(x+1, y+1) \end{bmatrix} \quad \dots \quad (4)$$

The vertical and horizontal gradients of the Sobel operators are G_x and G_y , which denote the gray level of the edge detection image in vertical and horizontal directions respectively which are computed as given by (Zhang, 2018) in equation (5) and (6)

$$G_x = Sobel_x \times A \quad \dots \quad (5)$$

$$G_y = Sobel_y \times A \quad \dots \quad (6)$$

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The approximate gradient value of brightness can be obtained by element-wise summation of the resulting images given by (Zhang, 2018) in equation (7).

$$|G| = \sqrt{G_x^2 + G_y^2} \quad \dots \quad (7)$$

Typically, an approximate magnitude is computed using equation (8) (Zhang, 2018) .

$$|G| = |G_x| + |G_y| \quad \dots \quad (8)$$

3.5. Fire Detection

In this research work, we employed thresholding-based technique to detect fire edge detected images which are expressed as equation (9).

$$g(x, y) = \begin{cases} Fire, & \text{If } f(x, y) \geq TH \\ Not\ Fire, & \text{Otherwise} \end{cases} \quad \dots \quad (9)$$

Where $f(x, y)$ is the input image and TH is the threshold value.

The threshold value was determined using the Otsu's method which is used to perform automatic image thresholding in computer vision and image processing (Zhang, 2018). In the simplest form, the algorithm returns a single intensity threshold that separate pixels into two classes, foreground and background. This threshold is determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance.

4. RESULT AND DISCUSSION

The results of the actual design of the hardware architecture are first presented by examining the synthesis and the simulation results. Then, the overall performance of the device was evaluated and compared with a similar work.

4.1 Synthesis

The hardware was synthesized with Xilinx Vivado 2021.1 design suite and simulated using Xilinx ISIM. Test benches were used to verify the functionality of the design, which was implemented in Kintex-7 series FPGA by using Xilinx configuration tools. The Synthesis and summary for the targeted Kintex-7 are shown in Figure 4. The motivation behind the use of Kintex 7 was because it is robust, highly reconfigurable and has lower power consumption. The input/output lines were large due to the high level of parallelism of the design, which generally enhanced the speed of operation of the device at the expense of area or huge number of primitives employed.

4.2 Verification of Results

The test benches used to verify the functionality of the design is not shown in this paper because of limited space. The

FIRESourceSignal entity was instantiated to provide RGB image data. The test stimuli were driven to the FIRE-SourceSignal inputs. The simulation in Figure 5 shows that the RGB image pixel data were read simultaneously from memory into the grayscale image conversion unit at the rising edge of the clock.

After the image pixel stream data were read and transferred into the grayscale conversion unit in parallel and concurrently executed. The result of the grayscale image conversion was depicted in Figure 6.

The result of Sobel Magnitude Computation is depicted in Figure 6 and that of Sobel Square Root Computation is shown in Figure 7. After the RGB images have been converted into grayscale they are transferred to the Sobel edge detection unit in parallel and concurrently executed by computing the Sobel magnitude and the square root values. The simulation results for fire pixel classification operations were similarly carried out.

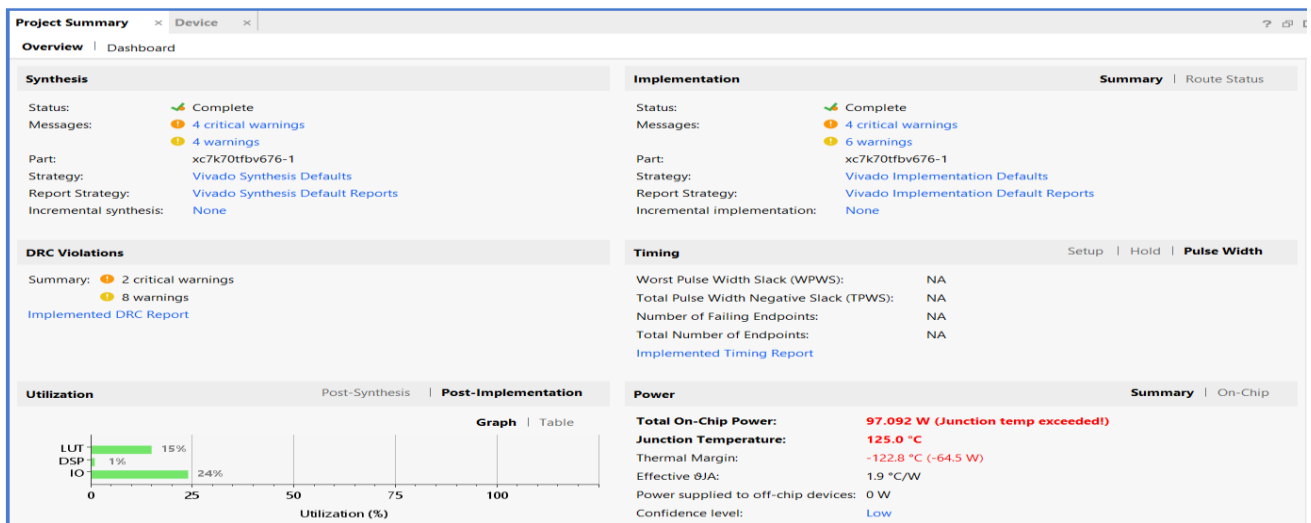


Figure 4. Synthesis Summary

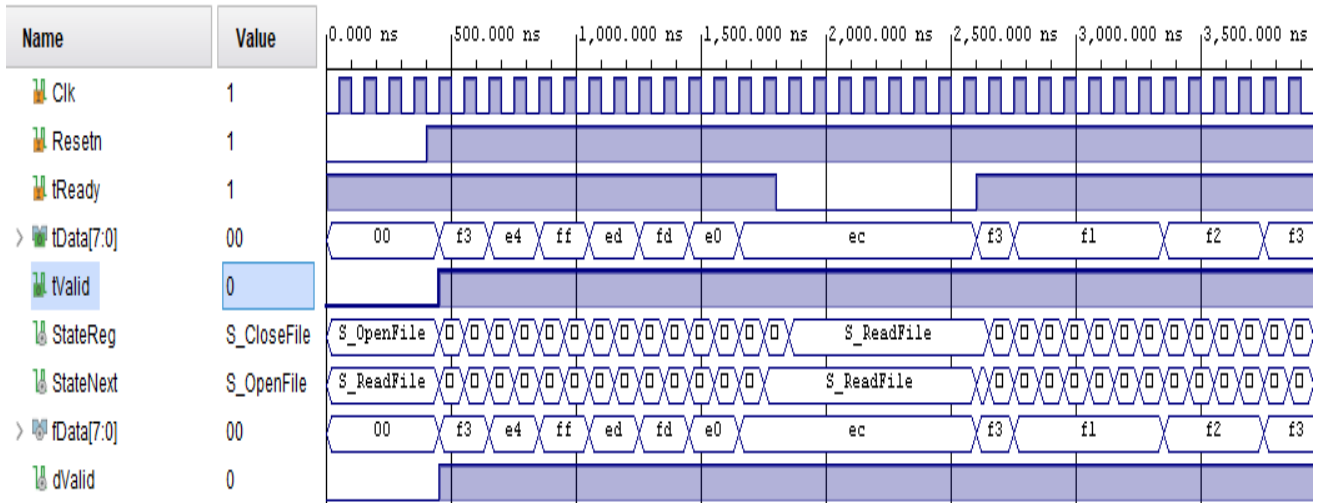


Figure 5. Image Pixel Generation Verification Waveforms

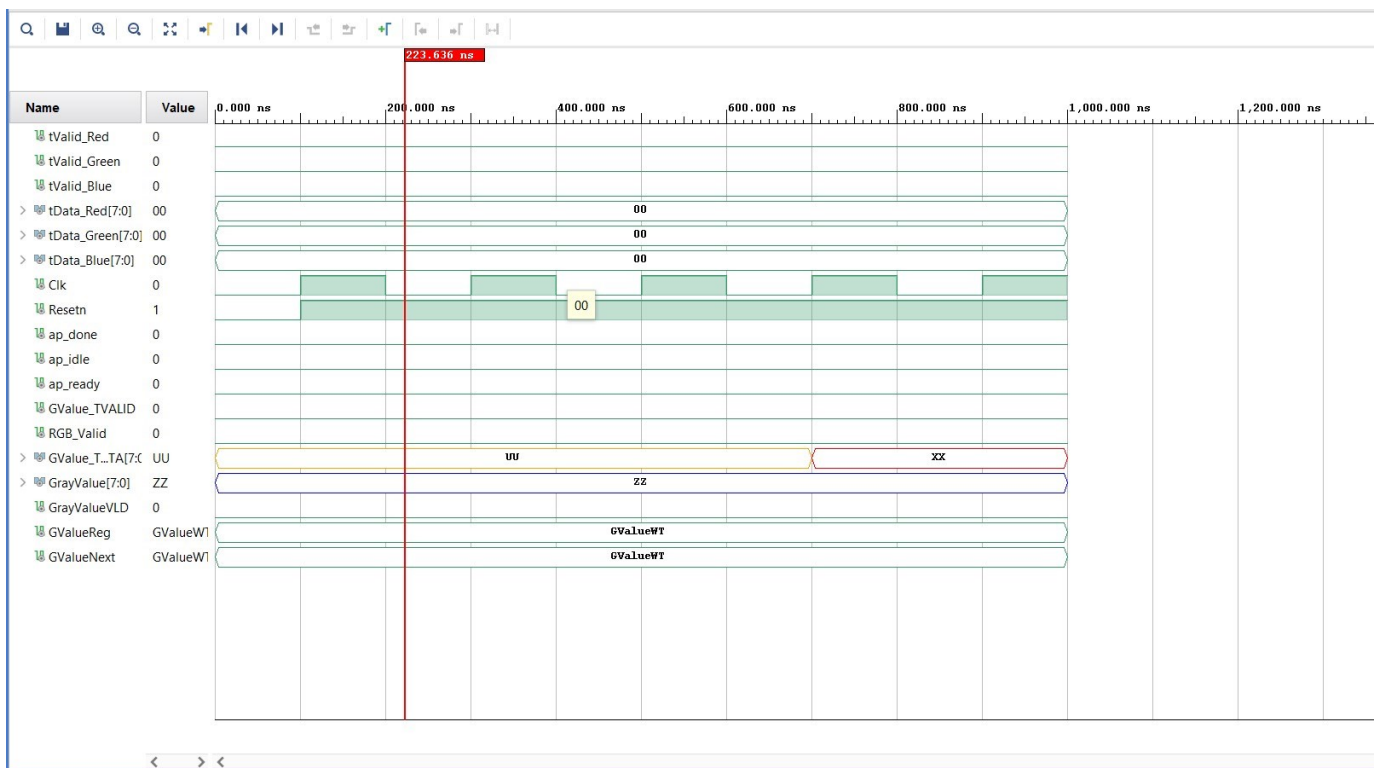


Figure 6. Grayscale Image Conversion Verification Waveforms

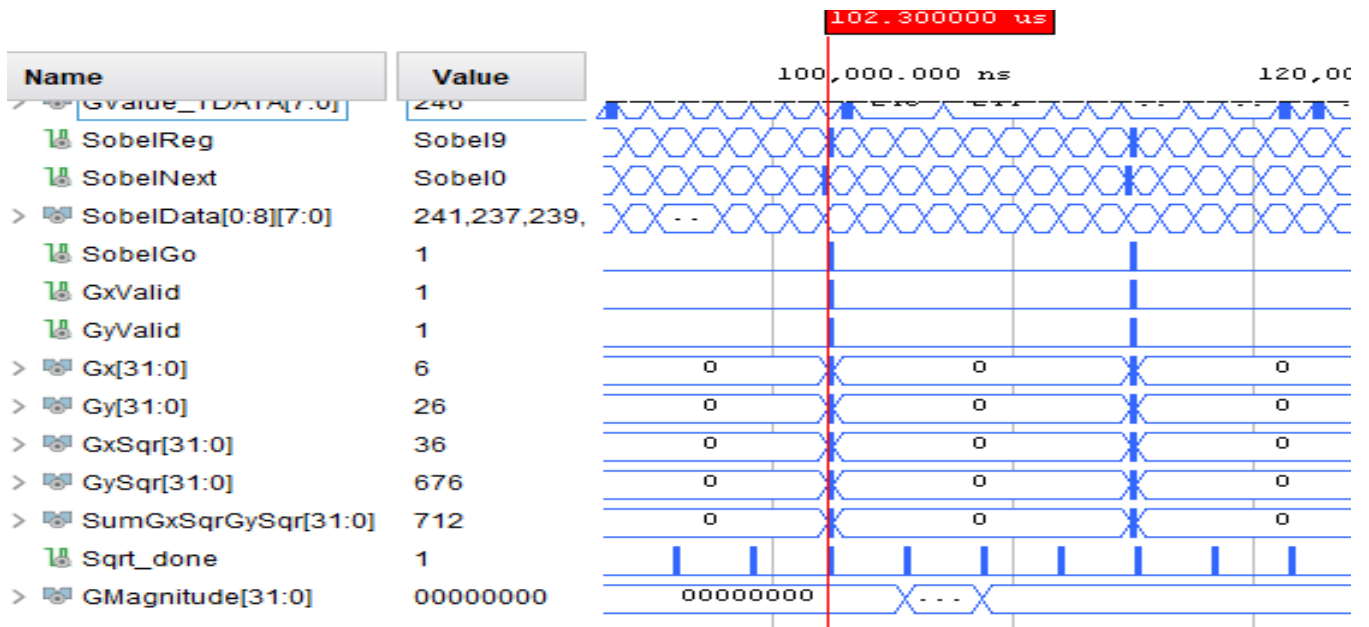


Figure 7. Sobel Magnitude Computation Verification Waveforms

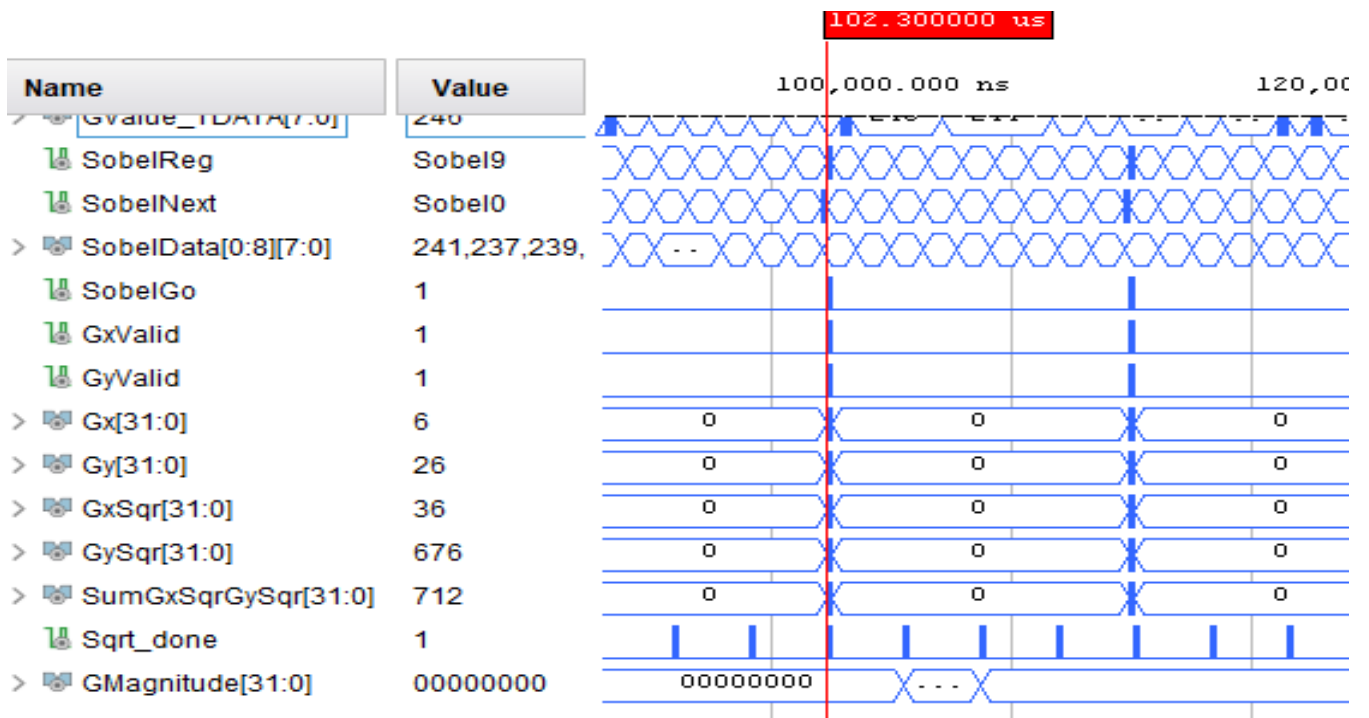


Figure 8. Sobel Square Root Computation Verification Waveforms

4.3 Performance Evaluation and Comparison

The performance of the design was evaluated and compared with the research work by authors in (Choudhuri & Roy, 2016). Results show a significant efficient resource utilization. Also available online at <https://www.bayerojet.com>

tion and the overall verification time. The performance comparison is shown in Table 1.

Table 1. Performance Comparison

Author and Date	LUT (%)	IO (%)	DSP (%)	Latency (ms)
Choudhuri & Roy, 2016.	1	7	1	3.2
Our work	15	24	1	2.97

It is evident that our design has efficient resource utilization, and the input/output lines were large due to the high level of parallelism of the design, which generally enhanced the speed of operation of the device at the expense of area or huge number of primitives employed. The comparison statistics shows that 15% of the available Look Up Tables (LUTs), 24% of available Input-Output(I/Os), and 1 % of available Digital Signal Processors (DSPs) were utilized by the design. The power consumption of the design is depicted in Figure 8. The dynamic power consumption is due to the

switching between digital states and the static power consumption is because of leakages in the circuit.

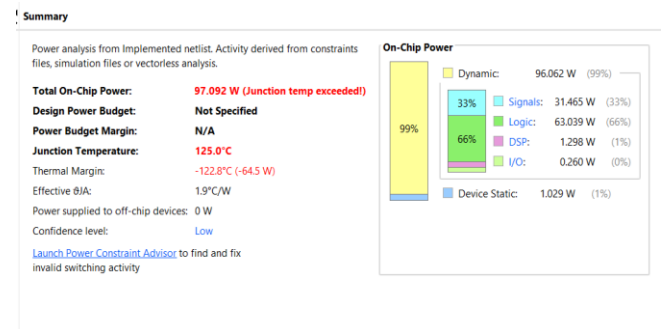


Figure 9. Power Analysis of the Design.

5. CONCLUSION

This research work was embarked upon to develop FPGA accelerator for vision-based fire detection system to address the limitations of the conventional systems which are electronic sensor based. MATLAB software was used for decoding image dataset into pixel stream data. The design was verified and implemented using simulation, synthesis and configuration tools of Xilinx Inc. USA. The images were acquired from Visifire video datasets which were converted into frames to generate the representative RGB images using MATLAB software.

It has been shown that the design has efficient resource utilization and power consumption (15% Look-up Table(LUT), 1% Digital Signal Processor(DSP), 24% Input-Output(IO), 92.062W Dynamic and 1.029W Device Static)(15% Look-up Table(LUT), 1% Digital Signal Processor(DSP), 24% Input-Output(IO), 92.062W Dynamic and

1.029W Device Static) compared to the work in (Choudhuri & Roy, 2016).(Choudhuri & Roy, 2016). The hardware accelerator which was developed as an Intellectual Property (IP) core can also be employed to speed up image processing and computer vision algorithms in embedded vision, smart camera and video analytics applications.

In this research, hardware accelerator based on FPGA was developed and evaluated using Vivado and MATLAB environment to address the shortcomings of software-based solutions for vision-based fire detection. we are going to consider interfacing camera and display monitor to the hardware design for real-life fire detection, enhancing detection performance by employing advanced image processing and computer vision algorithms and embedding machine learning algorithms into the design for fire prediction in future further research.

6. ACKNOWLEDGEMENT

This study was supported by Petroleum Technology Development Fund (PTDF) with grant number PTDF/ED/LSS/MSC/AH/0657/19.

REFERENCES

Abid, F., Izeboudjen, N., & Louiz, F. (2021). Hardware/Software Development of a Machine Learning based Forest Fires Prediction System. *Proceedings - 2021 International Conference on In-*

formation Systems and Advanced Technologies, ICISAT

2021.

<https://doi.org/10.1109/ICISAT54145.2021.967848>

6

- Choudhuri, A., & Roy, P. (2016). VLSI architecture for real time fire detection. *2015 IEEE International Conference on Computer Graphics, Vision and Information Security, CGVIS 2015*, 27–32. <https://doi.org/10.1109/CGVIS.2015.7449887>
- Foggia, P., Saggese, A., & Vento, M. (2015). Real-Time Fire Detection for Video-Surveillance Applications Using a Combination of Experts Based on Color, Shape, and Motion. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(9), 1545–1556. <https://doi.org/10.1109/TCSVT.2015.2392531>
- Hassan, A., & Audu, A. I. (2022). TRADITIONAL SENSOR-BASED AND COMPUTER VISION-BASED FIRE DETECTION SYSTEMS: A REVIEW. *Arid Zone Journal of Engineering, Technology and Environment(AZOJETE)*, 18(3), 469–492. www.azojete.com.ng
- Kaabi, R., Sayadi, M., Bouchouicha, M., Fnaiech, F., Moreau, E., & Ginoux, J. M. (2018). Early smoke detection of forest wildfire video using deep belief network. *2018 4th International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2018*, 1–6. <https://doi.org/10.1109/ATSIP.2018.8364446>
- Liu, Y., Sun, R., Zhang, T., Zhang, X., Li, L., & Shi, G. (2020). Fast Fire Identification Soft-Core Package Design Based on FPGA. *2020 International Conferences on Internet of Things (IThings) and IEEE Green Computing and Communications (Green-Com) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 642–647. <https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00112>
- Majid, S., Alenezi, F., Masood, S., Ahmad, M., Gündüz, E. S., & Polat, K. (2022). Attention based CNN model for fire detection and localization in real-world images. *Expert Systems with Applications*, 189. <https://doi.org/10.1016/j.eswa.2021.116114>
- Oluwarotimi, G. (2022). *A NOVEL VISION-BASED FIRE DETECTION ALGORITHM*. University of PortsMouth.
- Pujangga, R. S., Irawan, B., & Setianingsih, C. (2019, December). Fire Detection Using Image processing Techniques with convolutional Neural Networks. *2nd International Seminar on Research of Information Technology and Intelligent Systems 2019*.
- Rabiu, H. (2022). Robust Fire Detection Algorithm using Color and Shape Information. *BAYERO JOURNAL OF ENGINEERING AND TECHNOLOGY (BJET)*, 17(1), 2022–2044.
- Saeed, F., Paul, A., Karthigaikumar, P., & Nayyar, A. (2020). Convolutional neural network based early fire detection. *Multimedia Tools and Applications*, 79(13–14), 9083–9099. <https://doi.org/10.1007/s11042-019-07785-w>
- Solórzano, A., Eichmann, J., Fernández, L., Ziems, B., Jiménez-Soto, J. M., Marco, S., & Fonollosa, J. (2022). Early fire detection based on gas sensor arrays: Multivariate calibration and validation. *Sensors and Actuators B: Chemical*, 352. <https://doi.org/10.1016/j.snb.2021.130961>
- Torres-Huitzil, C., & Arias-Estrada, M. (2000). *An FPGA Architecture for High Speed Edge and Corner Detection*. <http://cseg.inaoep.mx/~ariasm,ariasm@inaoep.mx>
- Wang, X., Zhou, H., Arnott, W. P., Meyer, M. E., Taylor, S., Firouzkouhi, H., Moosmüller, H., Chow, J. C., & Watson, J. G. (2020). Evaluation of gas and particle sensors for detecting spacecraft-relevant fire emissions. *Fire Safety Journal*, 113(March), 102977. <https://doi.org/10.1016/j.firesaf.2020.102977>
- Xie, Y., Zhu, J., Cao, Y., Zhang, Y., Feng, D., Zhang, Y., & Chen, M. (2020). Efficient video fire detection exploiting motion-flicker-based dynamic features and deep static features. *IEEE Access*, 8, 81904–81917.

<https://doi.org/10.1109/ACCESS.2020.2991338>

Zhang, J., Zhu, H., Wang, P., & Ling, X. (2021). ATT Squeeze U-Net: A Lightweight Network for Forest Fire Detection and Recognition. *IEEE Access*, 9, 10858–10870.

<https://doi.org/10.1109/ACCESS.2021.3050628>

Zhang, S. (2018). *Real Time Image Processing on FPGAs*. University of Liverpool.