3rd International Conference on Automation, Control, Engineering and Computer Science (ACECS'16)

Proceedings of Engineering & Technology (PET)
pp. 430-434

Software Implementation Of an image Processing Chain on System on Programmable Chip (SOPC) with OpenCV library integration

Marwa Fradi
Physic Department of Faculty of Sciences of Monastir
fradimarwa89@gmail.com

Wajih Elhadj Youssef, Machout Mohsen
Physic Department of Faculty of Sciences of Monastir
Elhadj_youssef_wajih@yahoo.fr
Mohsen.Machhout@fsm.rnu.tn

***Abstract*— This paper presents a new method that makes accessible the implementation of an image processing program coded with OpenCV library on the Altera Cyclone III platform. The main idea consists of a design methodology of a SOPC making its hardware implementation on the NEEK platform. We make also the implementation of a uClinux Operating System on the Nios II processor based SOPC to facilitate the implementation of our proposed image processing chain on an Altera reconfigurable platform.**

Keywords*— *NiosII, SOPC, uClinux, OpenCV, Image processing, FPGA.

I. INTRODUCTION

The area of embedded system is evolving with the evolution of technological progress. It can generate many multimedia applications such as video application and image processing. Therefore, we are interested to the domain of SOPC for image processing. It is a very important area of research for the useful information that gives an image as a means of communication. Thus, it is crucial for the market to develop an image processing chain and implement it on an embedded system for various daily utilities applications. In fact, the image processing consists on applying a set of techniques for improving the quality of images. In this context, we will make a complete chain whose goal is to have the sharpness, quality and reproducibility. For our application, we will present a System on Programmable Chip (SOPC). Then we will develop a code describing an image processing chain in C/C++ language using the functions provided by the OpenCV library. After that, we will achieve substantive implementation of the embedded Operating System (OS) uClinux on the Nios II processor under Cyclone III FPGA. After preparation of the work environment, we will implement the developed image processing chain under our proposed SOPC. Finally, we will display of these images processed on the LCD of Embedded evaluation Kit (NEEK) of Altera.

This paper presents 5 sections divided as follows: in section I, we are interesting in the hardware/Software environment.

Section II presents the proposed processing chain. In the third section III describes the elaborated SOPC and the implementation of embedded operating system uClinux for image processing.
Section IV provides the results and observations and the last section gives the conclusion.

I. HARDWARE/SOFTWARE ENVIRONMENT

A. Hardware environment

The Nios II Embedded Evaluation Kit of Altera (NEEK) is a low-cost platform. It has a low power and a high performance. It contains all the hardware and software tools for development of several applications that are ready to be used. Hardware designers can use this platform to build complex embedded systems. Software developers can use the hardware reference design for the implementation of several software applications [1].

B. Software environment

- *The Quartus II software* from Altera Company is a comprehensive tool for System on Chip (SOPC) development [2].
- *The Qsys tool of Quartus II* can add NIOS II devices, input/output interfaces for the embedded system. It allows the generation of our SOPC and the creation of hardware design.

II. PROPOSED IMAGE PROCESSING CHAIN

The C/C++ language is an interesting tool for the application of image processing algorithms. Now, we proceed to the integration of the OpenCV library to construct an entire image processing chain as depicted in Figure 1.

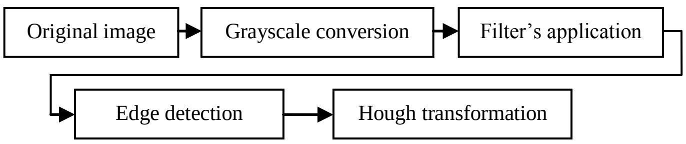


Figure 1: Image processing Chain

ISSN: 2356-5608

A. Techniques of image processing

a) Median Filter

The method is based on the median filter is adapted when the noise is composed of fine lines or isolated points. However, it is applicable only for images in grayscale. The idea is to replace a pixel with the median of its neighbors. Thus, even if several neighboring pixels are noisy, you can correct the current pixel. This induces a smoothing filter, however, since even the correct pixels can be changed. The sharpness is preserved but the median filter removes fine details irreversibly [3].

The methodology for the application of this filter is described by the following steps:

- Storing a list of values created in a table.
- The set of values constitutes a mask which can vary.
- Move the mask on the image.
- Make the sort of the table.

b) Bilateral Filter

It is a nonlinear filter that is well known in digital imaging. It is widely used for good noise removal with preservation of edges. The principle of this filter is to replace the intensity value of each pixel with a weighted average of neighboring pixel intensity values. It consists in a first step of applying a low-pass filter [4].

c) Canny Filter

The principle is based on the fixation of two thresholds, a high threshold *Sh* and a low threshold *Sb*. We begin by selecting the points that exceed the upper threshold and then we apply the lower threshold, keeping only the annexes components that contain a point above *Sh*. In other words from each point above *Sh* it follows a path consisting of points above *Sb*, this path has the desired outline [5].

d) Hough Transform

The technique of recognition of shapes by the Hough transform was invented for digital image processing. This algorithm allows the detection of lines in an image. Also, it is used for the detection of other geometrical shapes (circles, ellipses ...).

B. OpenCV Library

OpenCV (Open Source Computer Vision) is an Open Source library C/C ++ which contains over than 500 image processing functions and computer vision. It can process real-time still images, video files or video streams from cameras [6]. In the conception of our processing chain of images, we choose the OpenCV library for its crucial advantages. In fact it is a very rich library that contains several functions for image processing specialized in the treatment of real-time images using interfaces for major programming languages C/C++, Java and Python. It allows reading, writing and displaying the image, also having the function of calculating a histogram of gray.

But it is difficult to implement the OpenCV library on Cyclone III FPGA with Nios II processor [7]. In fact , It is designed for other platforms generations of Altera such as Cyclone V and it can be implemented on ARM processor. As a solution we will make the design of a SOPC achieving its implementation on Cyclone III plateform. Then we will implement a uClinux operating system on Nios II processor. Thus, uClinux will facilate the implementation of our processing image chain on Cyclone III FPGA .

III. SOPC DESIGN AND UCLINUX IMPLEMENTATION ON NIOS II PROCESSEUR

A. Proposed System on programmable Chip (SOPC)

A SOPC is a system on a programmable chip for solving development problems and achieving a rapid prototyping. The structure of our SOPC is designed by *Qsys* as described by the figure 2.

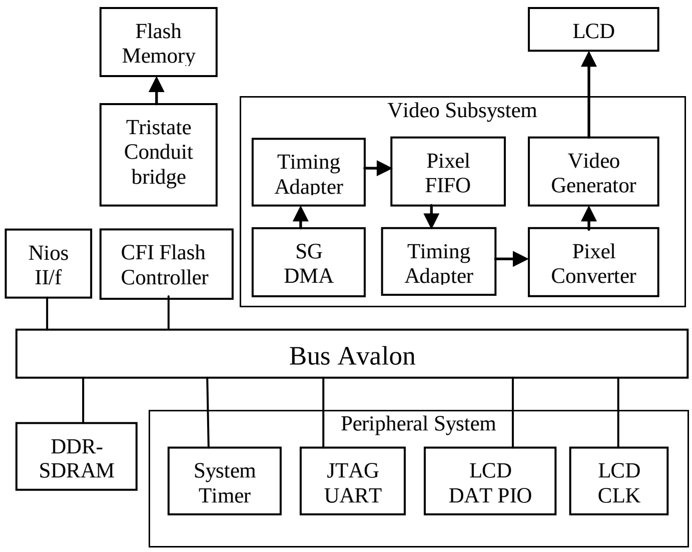


Figure 2 : Architecture of our designed SOPC

B. Implementation uClinux on Nios II processor

The implementation of an operating system is necessary because it facilitates the physical design. It manages multiple tasks and enables a fast generation of software applications. Thus we have to implement uClinux operating system on the Nios II processor of NEEK platform for facilating the implementation of our processing chain. In fact we need to prepare the work-environment such as:

- OS: Linux Mint 17.2
- Hard Disk Space: 60 GB FPGA
- Development tools: Quartus II Web Edition 13.1.

To implement uClinux on Nios II processor, we have to add some necessary libraries for hardware and software development of our application. In fact uClinux Linux is behind a Linux project. Currently, this distribution supports both of systems with and without MMU.

C. Integration of OpenCV library into uClinux

The integration of OpenCV library consists of the choice of the target processor on which is the implementation of our code and the operating system, different settings and to use different libraries of OpenCV. It should also select the cross-compilator.

D. Implementation of image processing chain with OpenCV library into Cyclone III FPGA

After generating our code under uClinux, we will implement it on the NEEK platform by means of a cross-compilation (nios2-linux-gcc). Following the implementation of our code under uClinux/Nios II, the original image and the image processing chain will be displayed on the LCD screen of the NNEK platform.

IV. EXPERIMENTAL RESULTS

A comparative study of our elaborated SOPC with the state of the art is described in table I.

Table I: SOPC compilation results

System Name	This work	[8]
Family / Type	***Cyclone III***	***Stratix II***
DSP	0	8
Logic elements	1415	6873
Switching elements	1.415/24.624	6873/48352
Memory Space	43.008/608.256	1,154,816 / 2,544,192
Pins	1/216	209/493
Frequency	102 MHz	120MHz

If we compare our proposed SOPC design with the [8], we confess that:

- The memory space used by Stratix II and the number of pins are very important in relation to that used with the Altera Cyclone III FPGA.
- The execution frequency of instructions in our map Cyclone III FPGA is about 102 MHz. It is similar to that found with the Altera Stratix II FPGA which shown in [8].

A. Loading the uClinux operating system on NiosII processor of NEEK

The first step is to Load the uClinux kernel image as shown in Figure 3.

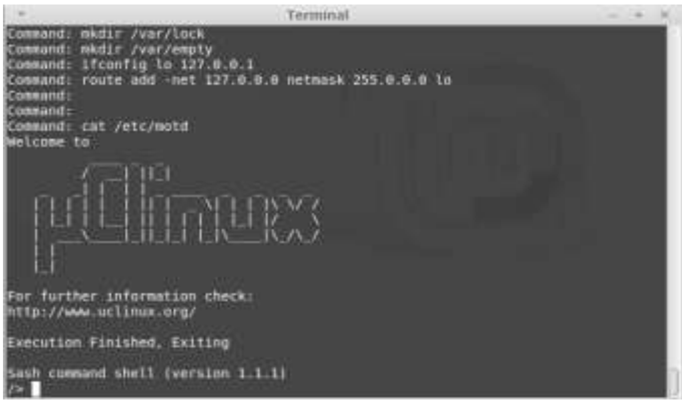


Figure3 : uClinux loading on Nios II processor

The next step is the integration of OpenCV library into uClinux as illustrated by figure 4.

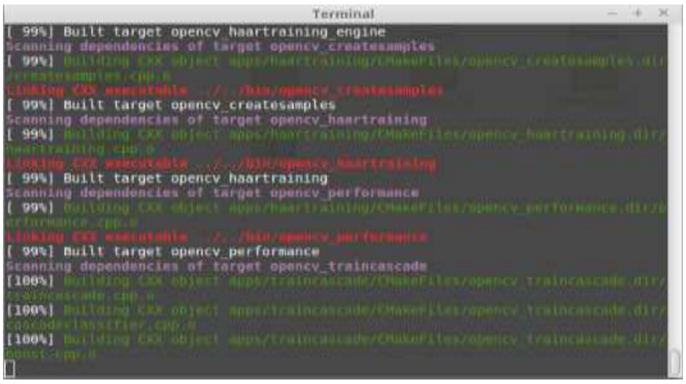


Figure 4: Generation of OpenCV libraries under uClinux

Finally, the configuration of this library under uClinux is made as shown is figure 5.

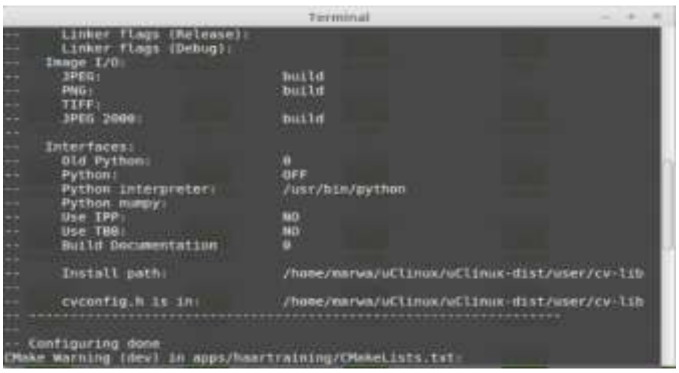
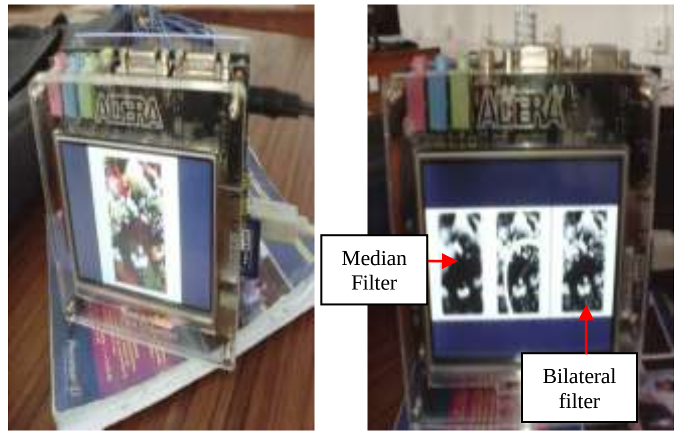


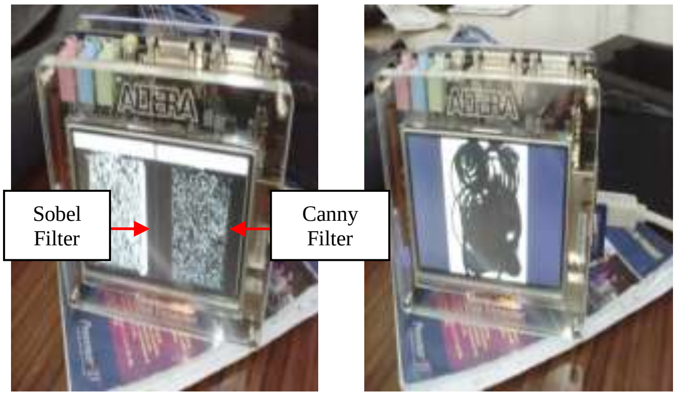
Figure 5 : Configuration of OpenCV under uClinux

B. Viewing the processing image on LCD screen of NEEK

After the implementation of our code under the Altera NEEK, the original image will be displayed on the LCD screen as shown in (a) (Figure 5). In the following, the display of images on which is applied the edge detection filter shown in the figure below. Finally, there was used a detection shapes (circles) by applying the Hough algorithm as displayed on the LCD screen



(a): Original image (b): Filter application



(c): Edge detection (d): Hough transformation

Figure 6: Displaying image processing on NEEK LCD screen

It is found that the quality of image obtained with the bilateral filter is optimal, better and sharper than treated with median filter as presented in (b) (Figure 6).

The detected circles are done by the coordinates of the given point as the center and radius of the circle (c) (Figure6). The Canny filter gives a very good precision with respect to the Sobel filter. In fact, we note that the distance between different points detected is minimal. The Sobel filter gives poor quality and shows more noise compared to the Canny filter (d) (Figure 6).

In [7], Hentai et al. make an implementation of an algorithm for iris recognition on Cyclone II. In fact they remove the OpenCV library out of their algorithm because it's so difficult to implement an OpenCV program in Cyclone II FPGA. This problem is solved in this work, and the implementation of a program based on OpenCV library is accessible on Cyclone III with Nios II processor.

V. CONCLUSION

In this paper, we make the implementation of an OpenCV library based processing chain on Altera Cyclone III platform. The interest dedicated to the integration of the uClinux system NIOS II aims to facilitate the implementation of the image processing chain made with OpenCV library on the evaluation kit. The SOPC design with Quartus II allows the implementation of its hardware. The image processing is a very important area of research for the useful information that gives an image as a means of communication

REFERENCES

[1] Altera Corporation, Nios II Embedded Evaluation Kit, Cyclone III Edition, User Guide, July 2010.

[2] Altera Corporation, Nios II Hardware Development: tutorial, May 2011.

[3] Pierre Bonnet, Filtrage median sur Lagis à l'université de Lile, 2009.

[4] Ming Zhang, Bilateral Filter In Image Processing,Thèse, 2006.

[5] Chantal St-Pierre, Détection Automatique De Stries De Croissance: Application Aux Écailles De Poissons, Université du Qubec, Mémoire 2005.

[6] Gary Bradsk and Andrian Kaehler, Livre Learning OpenCV, Gary Bradsk and Andrian Kaehler.

[7] Hentati.R , Res.Unit CES, Nat.sch.of Eng of Sfax, Tunisia, Abid.M.Dorrizi.B, Software implementation of the OSIRIS iris recognisation algorithm in FPGA, International Conference on microelectronics ICM11, tunisie, december 2011.

[8] Ben Atitallah.Ahmed, Etude et implantation d'algorithmes de compression d'images dans un environnement Mixte Matériel et Logiciel, Thèse, UniversitéBordeaux, 2007.