A Study of FPGA-based System-on-Chip Designs for Real-Time Industrial Application

Ali Al-Mahmood
Dept. Computing and Immersive Technologies
University of Northampton,
College of information technology

Michael Opoku Agyeman
Dept. Computing and Immersive Technologies
University of Babylon

ABSTRACT
This paper shows the benefits of the Field Programming Gate Array (FPGAs) in industrial control applications. The author starts by addressing the benefits of FPGA and where it is useful. As well as, the author has done some FPGA's evaluation researches on the FPGA performing explaining the performance of the FPGA and the design tools. To show the benefits of the FPGA, an industrial application example has been used. The application is a real-time face detection and tracking using FPGA. Face tracking will depend on calculating the centroid of each detected region. A DE2-SoC Altera board has been used to implement this application. The application based on few algorithms that filter the captured images to detect them. These algorithms have been translated to a Verilog code to run it on the DE2-SoC board.

Keywords
FPGA; System-on-Chip(SoC); Real-Time face detection and tracking Application; algorithims; images filtering; Industail control Application; Design Tool.

1. INTRODUCTION
Modern industries systems need to respond to various requirements to compete on their products which must be fast, high performing, reliable and very flexible [1].

Another key issue is the cost, in order to minimize it, time-to-market must be shortened and the price of controller device should be cheap.

The cost reduction, increasing the complexity of the control algorithms and reduce the time execution are the most challenging aspects for a new industrial control system to make a high and quick market impact.

To achieve these challenges, developers need to depend on the most advanced digital electronics technologies which come now with familiar software developments tools [1]. To develop a high quality real-time industrial control systems, developers have to choose one of the main two families of digital device technologies.

The first family relies on a pure software platform. Digital Signal Processors controllers (DSP controllers) and the microcontroller are the associated devices. These components integrate a performing microprocessor core with external devices which are very important to implement the targeted systems in real-time and to communicate with the industries environment.

There are few differences between DSP controllers [39] and microcontrollers [2]. The main difference is, for a specific surface of silicon, the proportion between the communication and control peripherals and the processing units.

The alternative family of digital devices for the industrial control systems is Field Programmable Gate Array(FPGA)[3].

FPGAs are programmable silicon chips, Using prebuilt logic blocks and programmable routing resources. It is easy to configure these chips to execute some hardware functionalities with pick up a soldering iron or breadboard.

FPGAs consist of pre-design elementary cells and interconnections which are completely programmable and it allows users to program their hardware according to their requirements.

The recent advances of the FPGA devices offering thousands of GFLOPs with high power quality. FPGA considered as a implementation platform between the hardware and the software. It's been developed on specially hardware design connected with the dynamic circuit switching interconnections, efficient reconfiguration and offering hardware performance and software flexibility. FPGA based on a hardware programming languages like VHDL and Verilog.

In order to promote designer productivity and decrease time-to-market modern design techniques such High Level Synthesis(HLS), Electronic System Level(ESL) design or, in easier terms, C based hardware design can be adopted. ESL, HLS, and C are based hardware design [4], all less or more cover the automatic translation of untimed C/C++ algorithmic descriptions into Register-Transfer Level (RTL) HDL architectural descriptions, ready for FPGA implementation.

Digital industrial control Implementation technologies and methodologies, like FPGAs, DSPs and microcontrollers are getting popular during the recent years. Especially FPGAs have produced various types of efficient and advanced hardware design solutions into the industrial control arena. These involve HDLs [5], C based design and HLS[6], PLC code to HDL translator [7], software hardware design[10] and run-time reconfiguration[11], SoC [8] and Multiprocessor SoC (MPSoC) [9] architectures.

FPGAs have been developed to be used for the implementation of different types of controller [12].

The application that has been used to in this paper is a real-time face detection and tracking application [38] which has been designed for FPGA on SoC. It is going to be used to prove the performance of the FPGA-SoC within industrial applications. This application functionality is to detect and track persons' faces to check whether it is human face or not. The application developed by Thu-Thao Nguyen which is taken as a part of a master degree.

2. FPGA-BASED CONTROLLER FOR EMBEDDED INDUSTRIALS AND ROBOTIC

Aircraft and automotive embedded systems are the most challenging applications for digital electronics [13]. The most important issue for these systems is safety. Therefore in [14] and [15], the authors have suggested some techniques to deal with this issue to increase the reliability of the FPGA-based controllers.

Another concept of safety is the possibility of the vehicle manufacturers and their providers to deal with the obsolescence of the critical embedded systems. Electronic devices now have a very finite life time and it is compulsory to find a completely secure FPGA based intellectual Property(IP) module in the market to change then when they are longer available [37] [16]. The existing tendency for modular architecture within the FPGA-based embedded controller has also drive the developer to enhance IP to use the advanced encryption techniques [17].

Another anxiety of the aircraft and the automotive embedded systems is the implementations of the real-time design of high date rate and the dependable of the protocols in the network of the vehicle like FlexRay or SpW[18]-[19].

Concerning handled embedded systems, the main issue is the decreasing of the power consumption [20]- [21]. About this issue few studies have been done in [21], which presents understanding of the source of power consumption.

FPGA devices are widely used to manage the communication distrusted applications by the effective real-time Ethernet protocol [22]- [23].

3. PRESENTATION OF FPGA

FPGA hardware always increasing interest and have strongly disrupted the early digital development trends [37]. These devices belong to the semi-custom ASICs. The latter low cost consists of predesigned interconnections and elementary cells which could be programmed and interconnected by users.

3.1.Generic structure of on FPGA

Fig 1 shows the basic structure of FPGA which consists of many logic blocks(LBs), of an I/O blocks and interconnection network. Recent FPGA on a very sophisticated level which integrates the hardwired DSP blocks, memory blocks, communication blocks [28] and clock manager blocks.

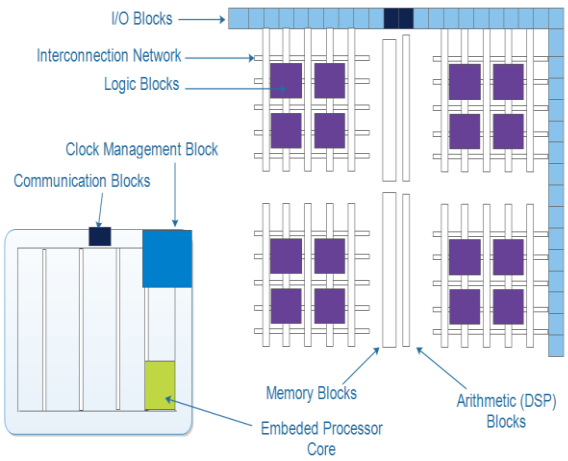

Fig. 1. Generic structure of an FPGA

According to the proposed function to implement, every LB is assigned to execute combinatorial or/and sequential operations. For the combinatorial operations, a set of Look-Up-Tables (LUTs) are included which is same with the sequential operation of the D-Flip-Flops. An LB is capable to to proceed a local storage function (DRAM), multiplexer, shift register(SR) and adder/subtractor operations. The interconnections network is also re-programmable by the developer to interconnect as many LBs as required.

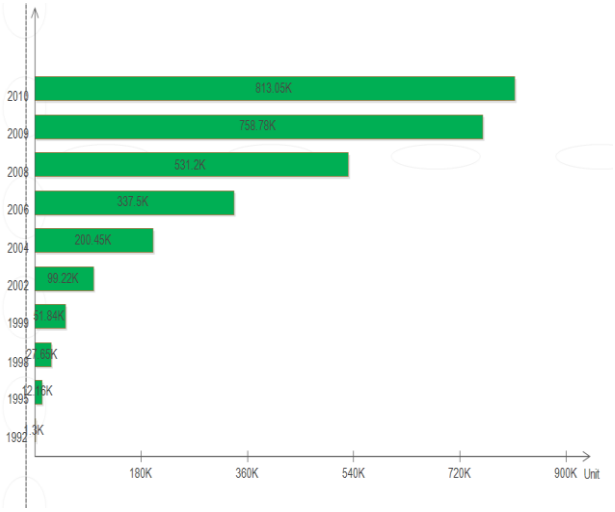

Fig. 2. Evolution of the FPGA density

A normalized FPGA density metric has been accepted because internal structure od LGs are different from FPGA to another. In fact, Logic Cell has been realized. LC consists of D-Flip-Flop, 4-bit LUT, multiplexer and a carry chain (for arithmetic operations) [29] [30]. Fig 2 shows the evaluations of the FPGA density.

I/O configurable blocks allow the connection between the external environments and the internal architecture.

The integrated clock manager blocks allow the management of the clocking resources. Which are commonly based on the Phase-Locked-Lools (PLLs). The latter provides some features like frequency multiplication and division, phase shift correction and propagation compensation.

4. FPGA SYSTEM-ON-CHIP TRENDS

SoC devices integrate microprocessors with FPGAs technology. There are currently three companies shipping SoC FPGA devices which are Altera, Xilinx and MicroSemi. The SoC FPGAs devices from these venders combines fully dedicated microprocessors with the FPGAs technology on the same device. As mentioned that the SoC approaches shows that the new designed features combine the hardware design with the software. There are two groups of processor cores are considered, the “synthesizable” and the “non- synthesizable” cores.

The synthesizable (also called Soft Cores) like Altera’s Nois, Xilinx’s MicroBlaze processors and Actel’s ARM7 or Cortex-M1 are using existing FPGA logic cells (LC) to perform the processor core. The feature of such approach is the ability that allows the designer to specify and configure the number, the memory width and the kinds of the peripherals. However, these cores show the clock rates.

The non-synthesizable (also called hard processor cores) have a tradition VLSI layout. Which is combined within the FPGA. In general, a hard processor core produces an efficient and

high clock speed with low flexibility. For instance, Altera supports an ARM9 processor core embedded in its EPXA10 series which marketed as an Excalinur™ devices [31]. The Xilinx Virtex-5 also integrates a hardwired PowerPC 440 processor cores on-chip [32]. Actel has offered the first hardwired Cortex-M3 processor core integrated into its Fusion FPGA family [33].

5. EVALUATING SOC FPGAS

Designers need to look at the following framework to evaluate venders

5.1.System Performance

There is no simple single way to look at the performance, it is certainly important to look at the typical data sheet aspects like CPU clock speed, maximum memory speed and system considerations. For instance, FPGA is getting popularity in industries. As fig 3 shows some kind of applications, designer can combine multiple functions like Motor control, Network and safety all in a single device as shown in fig 4. That saves the power and the cost as

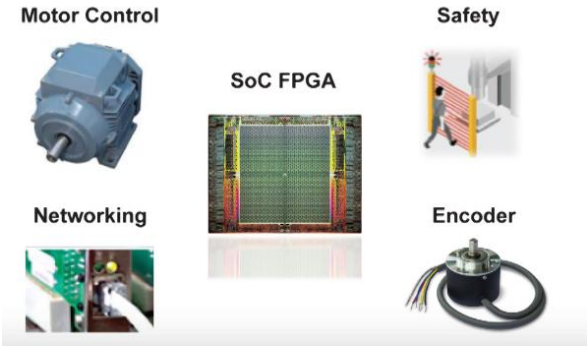

Fig3 SoC FPGAs industrial Drives.

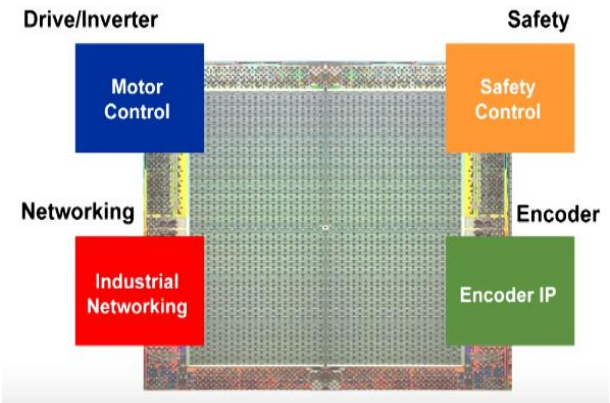

Fig4 SoC FPGAs industrial Drives.

In system like that system performance might be all about how quickly the processor can calculate multi control loops in the same time which means all the processing must be deterministic.

5.2.Design tool

During the evaluation features of the FPGA, the design tool has been developed as well. Nowadays, FPGA venders provide a complete set of tools which allow a very good quality design process start from the hardware description, using the Verilog or VHDL languages to final but stream generation [30]-[32]. A simple FPGA design process it shown in Fig 5

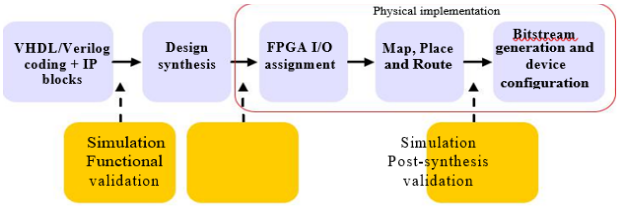

Fig5. Simplified synoptic of the FPGA design process

Several examples of the Integrated Software Environment (ISE) tools from Xilinx, Quartus tools from Altera and Libero Integrated Design Environment (LiberoIDE) tools from Actel. All of them support flexible and complete design features with additional associated tools for simulations (e.g. ModelSim tools) and for debugging (e.g. ChipScope tools from Xilinx).

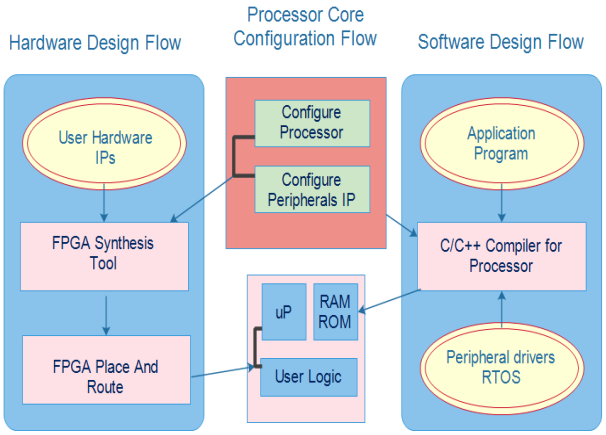

Fig. 6. Typical FPGA SoC design flow

In addition, to suit SoC trends, FPGA vendors provide software development tools (editor, compiler, assembler, linker and debugger), software vendor IPs and processor customization tools. For example, Altera provides Embedded Design Suite (EDS) platform, Xilinx provides Embedded Development Kit (EDK) platform and Actel provides Soft Console platform. Fig. 6 presents a standard design flow for developing SoC applications

The design flow consists of two main procedures: the hardware design flow and the software design flow. It presents a friendly user interface which allows the designer to specify the processor for a particular project.

After finishing the configuration, the processor core is created in the form of an HDL file (in the case of Altera and Actel tools) or a netlist file (in the case of Xilinx tools). Then, this file can be associated to custom user logic and integrated within the hardware design flow to be synthesized, placed and routed.

FPGA also can be configured with the resulting bitstream file. Therefore, the program is going to be integrated on the soft processor cores which can be interpreted with the associated library files. The compiler of C/C++ has been proposed to this processor which is also provided for the development system.

6. FPGA-BASED DESGIN OF AN INDUSTRIAL CONTROLLER CASE STUY: A REAL-TIME FACE DETECTION AND TRACKING USING SoC FPGA.

FPGA technology provides the development of the hardware architectures a very flexible programmable environment. This feature allows designers to be more free compared to software implementations based on microcontroller and DSPs [30]-[34][38]. That is because FPGAs are outperforming these software solutions by exploiting the inherent parallelism of the algorithm. Consequently, designer can develop a hardware architecture that is fully dedicated to the algorithm to implement.

In this section, FPGA based design of an industrial controllers is explored.

FPGA technology has been used to implement a real-time face detection and tracking [38]. This purpose of this application is to detect and track a human's face, the face detection algorithm includes image filtering and colour-based skin segmentation. The location of the face is selected by calculating the centroid of the detected region. At the testing stage the experimental result was very accurate and effectiveness of the real-time system, even under various rules of facial poses, lighting, and skin colours. The calculations of the hardware execution have been done in real-time with less computational effort.

Face detection and tracking have become very important in many applications for example, in the biometrics, video surveillance or video coding.

The aim of this application is to create a FPGA system to discover and track a person's face in a real-time. This application has been developed using Verilog language, a camera, Altera DE2 board and a VGA monitor. The algorithm of the face detection has been executed here which is based on image filtering and skin detection. After the detection of the face region, the location of the human's face is determined by calculating the centroid of the very close skin pixels

7. SOC FPGA APPLICATION DESIGN AND IMPLEMENTATION

7.1.Algorithm

The algorithm of the skin detection has been derived from a method that's been well described in [35]. The colour segmentation demonstrated to be efficient method to detect the face region in spite of the implementation case and the low computational requirements. The colour based algorithm desired a very little training compared to the feature based-method. Fig 7 shows the general design stages.

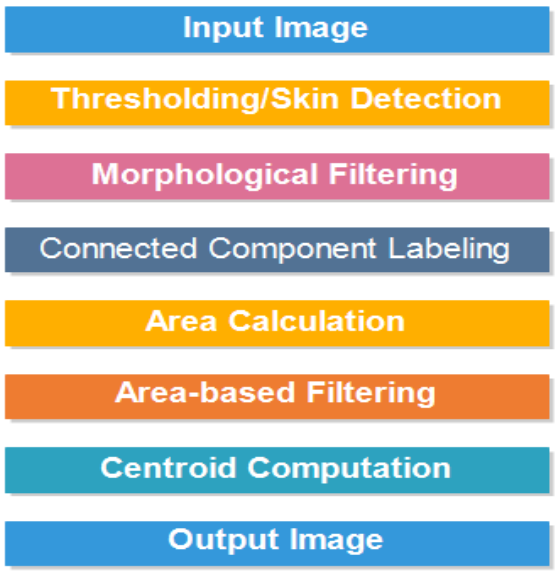

Fig 7 Software algorithm

Firstly, the original image is going to be a different color space, namely modified YUV. After that the skin pixels will be segmented based in the algorithm on the proper rang. Some morphological filters have been applied to decrease the false positives. Then every detected pixel is going to be labelled, the area of each labelled region was computed and an area-based filtering was applied. Only regions with large area were considered face regions. The centroid of each face region was also computed to show its location.

7.2.Modified YUV Colour space

Transforming the skin pixels' data into modified YUV colour space it is more beneficial cause human skin tones resort to fall within a particular range of chrominance values (i.e. U-V), ignoring the type of the skin.

The following equations illustrate the conversion [35].

$$Y= \frac{R+"G+B}{4}$$

U= R-G, V=B-G

45 34 30	#2D221E
60 46 40	#3C2E28
75 57 50	#4B3932
90 69 60	#5A453C
105 80 70	#695046
120 92 80	#785C50
135 103 90	#87675A
150 114 100	#967264
165 126 110	#A57E6E
180 138 120	#B48A78
195 149 130	#C39582
210 161 140	#D2A18C
225 172 150	#E1AC96
240 184 160	#F0B8A0
255 195 170	#FFC3AA
255 206 180	#FFCEB4
255 218 190	#FFDABE
255 229 200	#FFE5C8

Fig 8 shows the different skin tone samples [35].

7.3.Thresholding/Skin Detection

After the conversion of the skin pixels to the modified YUV space, the skin pixels could be segmented according to the following experimented threshold:

$10 < U < 74$

$-40 < V < 11$

As Fig 2 displays that the blue channel has the minimal contribution to the skin colour of the human. Regarding to [2], ignoring the blue channel would have a bit of effect on the skin filtering and the thresholding. It also reveals the insignificance of the V component in the YUV format. Therefore, the algorithm of the skin detection which is used here based on the U component only. After applying the segmentation threshold for the U component will introduce a binary image with raw segmentation output, as shown the Fig 9.

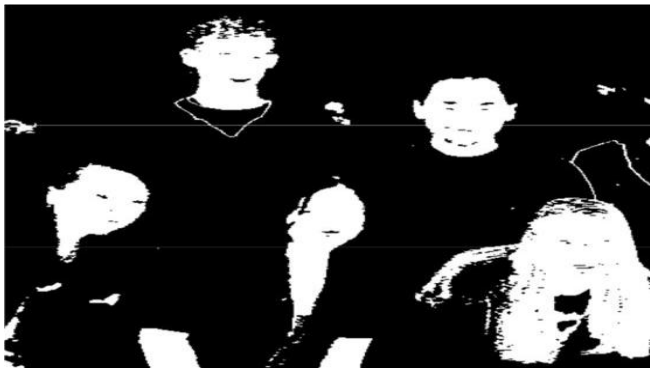

Fig 9 Result after thresholding [38].

8. SOC FPGA APPLICATION HARDWARE IMPLEMENTATION

Fig 10 shows the FPGA system setup.

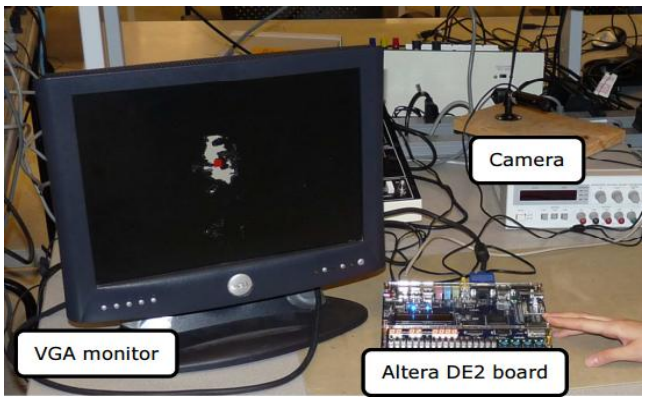

Fig 10 shows the FPGA system setup

The camera captures each video frame to send it to the FPGA's decoder chip via a composite video able. Then the video signal is going to be processed in various modules in Verilog, after that the output passed through the VGA driver to be showed on the VGA screen.

The algorithm of the hardware has been modified to be as fig 11.

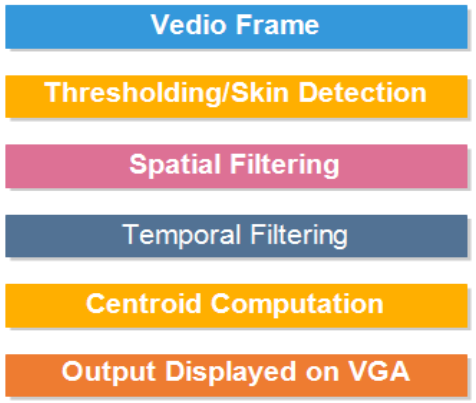

Fig 11 hardware algorithm

8.1.Thresholding

10-bit colour used in the Verilog, setting a aforementioned U range yields $40 < U < 296$. Each video frame in this stage is changed to be a "binary image" displaying the segmented raw result.

8.2.Spatial Filtering

This stage is almost similar to the erosion operation that are used in the software algorithm. However, the structuring element used here don't have any particular shape. Instead, for every pixel p, its neighboring pixels in a 9x9 neighborhood were checked. If more than 75% of its neighbors were skin pixels, p was also a skin pixel. Otherwise p was a non-skin pixel. This allowed most background noise to be removed because usually noise scattered randomly through space, as shown in Figure 12. In Figure 13, because p only had 4 neighboring pixels categorized as skin, p was concluded to be a non-skin pixel and, thus, converted to a background pixel.

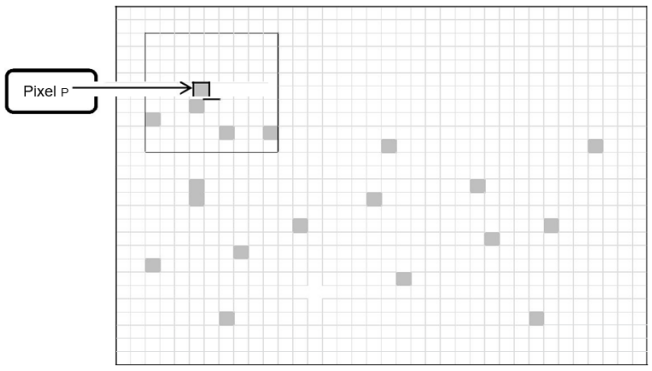

Figure 12 - Example of spatial filtering for a pixel P— before filtering

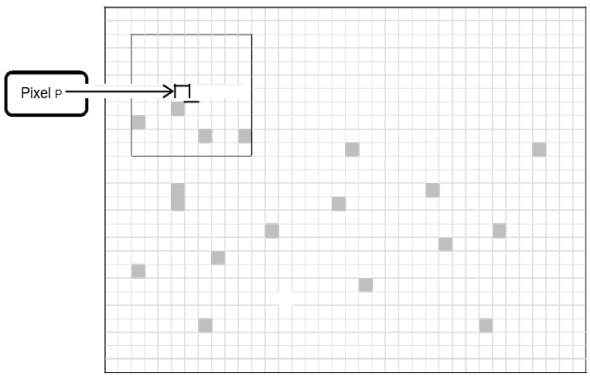

Figure 13 - Example of spatial filtering for a pixel P—after filtering

In order to inspect the neighbour pixel, the values of them must be stored. Therefore, ten shift registers that have been created to buffer the values of ten consecutive rows in each frame.

8.3.Temporal Filtering

Sometimes a small change in lighting may cause flickering and make the output on the VGA monitor unstable. Using temporal filtering gives flickering the ability to be decreased significantly. The idea of designing such a filter which is borrowed from the project "Real-Time Cartoonifier".
The following equation need to be used to apply the temporal filter.

Average_out = (3/4) average _in + (1/4) data

Data :the filtered output comes from the last stage of a pixel, namely P, in current frame average_in: average value of P from previous frame

Average out: average value of P in current frame.

It is nearly equal to averaging four consecutive frames over time. To ease the computational effort, the equation above can be re-written as

Average _out = average _in – (1/4) average _in + (1/4) data

Average _out = average _in – average _in >> 2 + data >> 2

8.4.Centroid Computation

Lastly, centroid has been computed to locate the region of the human's face.

First suppose that only one face is presented. Therefore, its centroid would just be the centroid of all detected pixels, as it is shown in Fig 15. Note that this calculation would only be true in case if one face is presented.

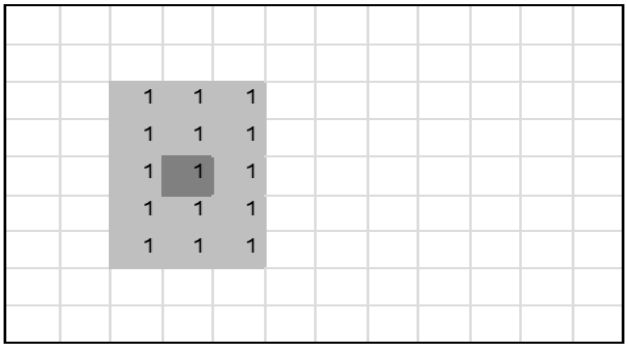

Figure 15 - Centroid of all detected pixels

In spite of the pixels of one human's face may not be connected and labelled, simply the calculating of the centroid of all detected still gives a pretty guessing for the face location, as it is shown in Fig 16.

Since area-based filtering also has not been, other skin palces—mostly the hands were not entirely removed. However, even if the hands were present, calculating the centroid of all detected pixels still allowed us to locate the face region. This was a suitable estimate because, compared to the face area, the area of the hand/hands was much smaller.

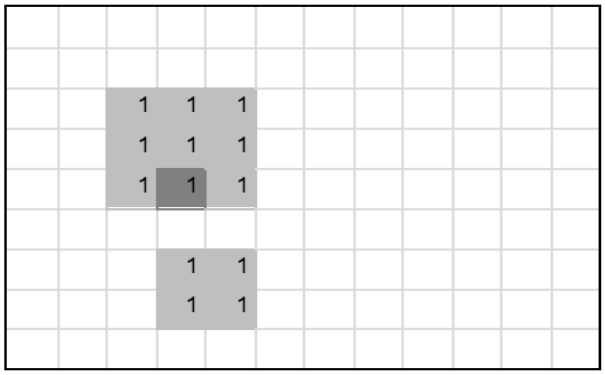

Figure 16 - Centroid of all detected pixels—one person

When there are two faces presented to the camera, the calculating of the centroid of all the detected pixels would track the location between the two faces, instead of tracking each face separately. To separate the track of each face of two-person frame, additional steps are required. First the neighboring pixels around the centroid are checking to see if they are any skin pixels. If they are, it means that the centroid carefully located the region of the face. However, if the neighboring pixels of the centroid are not skin pixels, it means that the centroid is somewhere in the background located between two detected face regions, as it is shown in Fig 17.

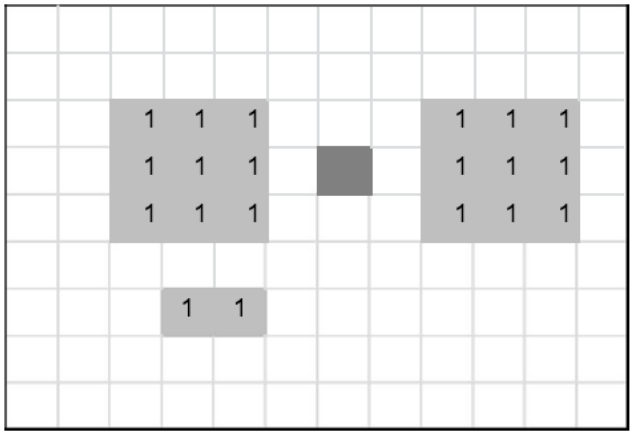

Fig 17 - Centroid of all detected pixels—two people

To solve this issue, the frame of the video needs to be divided into two according to the centroid location, as it is shown in Fig 18.

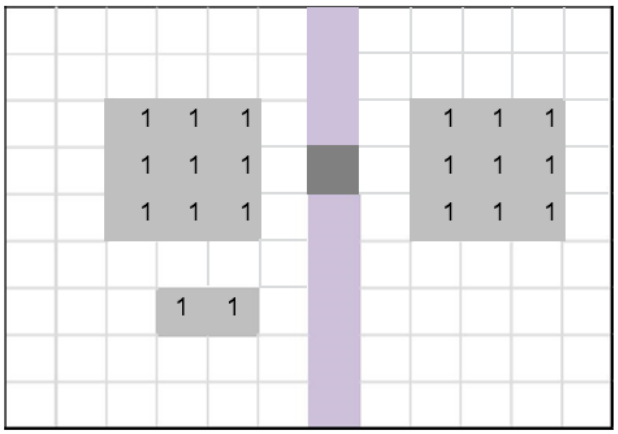

Fig 18 shows the calculation separation of the centroid of each detected region

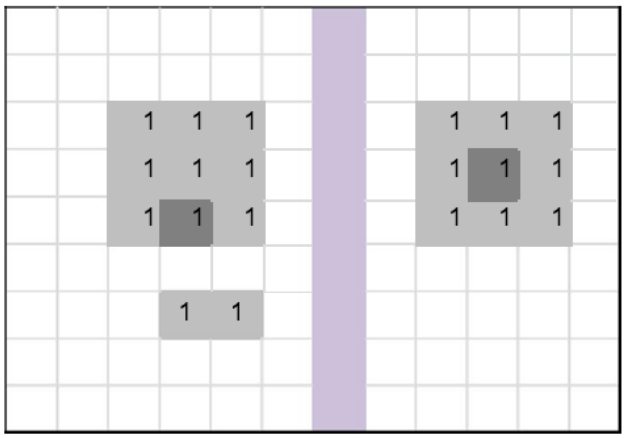

Fig 19 - Centroid of each detected face

Getting the centroid of every face region allowed us to locate the face of each human introduce in a two-human video frame. [38]

Applying the temporal filtering allowed the box to move smoothly. The implementation of the temporal filter here was slightly different from the one shown previously.

$Y_N = (1 - \alpha)X_N + \alpha Y_{N-1}$

X_N: current input

Y_N: current output

Y_{N-1}: previous output

9. THE RESULT OF THE REAL-TIME FACE DETECTION AND TRACKING APPLICATION

Face detection and tracking application is now able to detect and track two people faces in a real-time. In spite of tracking each face separately if there are three faces or more, it still able to detect the presence of their faces.

Sample Results and Analysis are going to be presented in this part.

When there is no one (Figure 20, Figure 21, Figure 22)

- From Figure 20 to Figure and 21 show many false positives have been removed.
- It clear from figure 21 and 22 that the scarf is eroded. And the centroid of the detected pixels didn't appear because the number of the detected pixel is below the threshold. In this case, it shows that no face has been tracked or detected.

Images from a book (Figure 23 and Figure 24)

- Images from a book. The system can detect and track a face on an image in a book if the book moved manually.

Presence of one person in (Figure 25, Figure 26, Figure 27, and Figure 28)

- It is clear from the pictures that the scarf is completely eroded and it didn't affect the final result

Presence of two people in (Figure 29, Figure 30, Figure 31, and Figure 32)

- In case, if there are more than one person, the system can detect and track person's faces [38].

These images illustrate the testing of the application:

Figure 20 - When there was no one—natural [38]

Figure 21 - When there was no one—skin detection

Figure 22 - When there was no one—spatial filtering

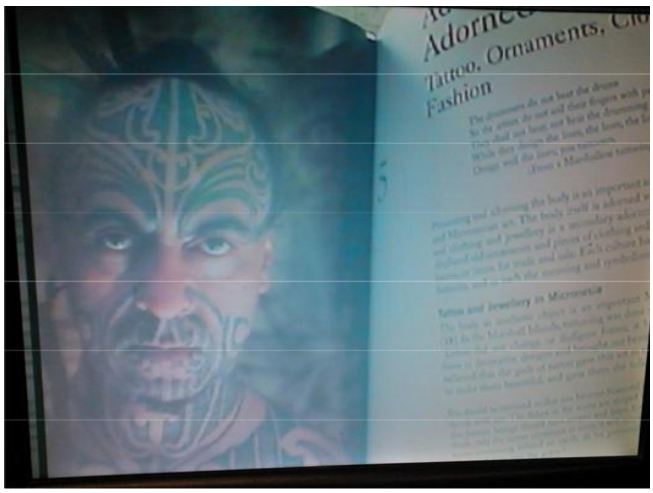

Figure 23 – Still image taken from a book—natural

Figure 24 - Still image taken from a book—Final Result

Figure25- Presence of one person (with scarf)—natural

Figure 26 - Presence of one person (with scarf)—skin detection

Figure 27 - Presence of one person (with scarf)—spatial filtering

Figure 28 - Presence of one person (with scarf)—temporal filtering + centroid computation

Figure 29- Presence of two people—natural

Figure 30 - Presence of two people—skin detection

Figure 31 - Presence of two people—spatial

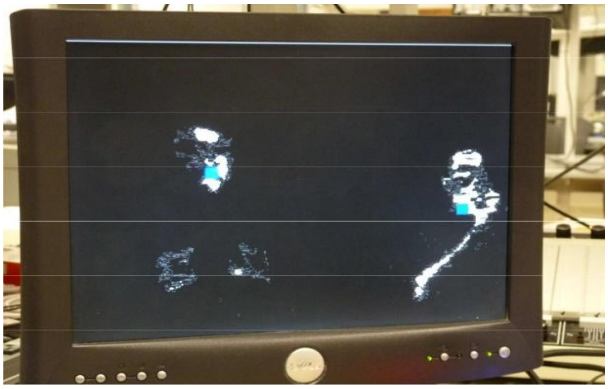

Figure 32- Two People recognized by the system

10. CONCLUSION

The main goal of this paper was to explain the FPGA's technologies and to illustrate the uses of FPGAs in industries. This paper showed the benefits of using FPGA and how it is going to help industries to produce the best products. As mentioned in the introduction part that the industrials' applications should be on a high performance, reliable, fast, and very flexible. The contributions of the controller embedded systems have discussed and declared the main issues with the applications. These issues have been discussed to find the best solutions to avoid them. FPGA-SoC trends have been discussed to compare between the two groups of processors which are the "synthesizable" and the "non-synthesizable and highlights the benefits of them and the uses.

The Evaluating of FPGA-SoC has been discussed to evaluate the performance of it within the industrials' applications. A real-time application has been used to prove the high performances of the FPGA-SoC. The application was a face detection and tracking system to detect and truck human faces. It has been developed using Verilog hardware programming language to run it on DE2-SoC Altera board. Some testing has been done on a live human face and on a picture that has a human face. Some algorithms of images' filleting have been implemented on the captured image in order to detect the right pixels to present the expected results.

11. REFERENCES

[1] C. Economakos, G. Kiokes and G. Economakos, "Using advanced FPGA SoC technologies for the design of industrial control applications," *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*, Corfu, 2015, pp. 1-6

[2] Phil Lapsley, "DSP processor fundamentals: architecture and features", IEEE Press, Computers, 1997.

[3] Wayne Hendrix Wolf, "FPGA-based system design", *Prentice Hall PTR*, 2004.

[4] Coussy and A. Morawiec, *High-level Synthesis: From Algorithm to Digital Circuit*. Springer-Verlag, 2008.

[5] S. Ghosh, R. K. Barai, S. Bhattarcharya, P. Bhattacharyya, S. Rudra, pp Dutta, and R. Pyne, "An FPGA based implementation of a flexible digital PID controller for a motion control system," in *International Conference on Computer Communication and Informatics*. IEEE, 2013.

[6] D. Navarro, O. Lucia, L. A. Barragan, I. Urriza, and O. Jimenez, "High-level synthesis for accelerating the FPGA implementation of computationally-demanding

control algorithms for power converters," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1371– 1379, 2013.

[7] S. Subbaraman, M. M. Patil, and P. S. Nilkund, "Novel integrated de-velopment environment for implementing PLC on FPGA by converting ladder diagram to synthesizable VHDL code," in *11th International Conference on Control Automation Robotics and Vision*. IEEE, 2010, pp. 1791–1795.

[8] A. Ben Said, M.and Hemdani, M. W. Naouar, E. Monmasson, and I. Slama-Belkhodja, "Standard FPGA-based or full cSoC controllers for three-phase PWM boost rectifier, two viable solutions," in *15th International Power Electronics and Motion Control Conference*. IEEE, 2012.

[9] S. Ben Othman, A. K. Ben Salem, H. Abdelkrim, and S. Ben Saoud, "MPSoC design approach of FPGA-based controller for induction motor drive," in *International Conference on Industrial Technology*. IEEE, 2012, pp. 134–139.

[10] E. Monmasson, I. Bahri, L. Idkhajine, A. Maalouf, and W. M. Naouar, "Recent advancements in FPGA-based controllers for AC drives appli-cations," in *13th International Conference on Optimization of Electrical and Electronic Equipment*. IEEE, 2012, pp. 8–15.

[11] M. W. Naouar, E. Monmasson, A. A. Naassani, and I. Slama-Belkhodja, "FPGA-based dynamic reconfiguration of sliding mode current con-trollers for synchronous machines," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1262–1271, 2013.

[12] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems - a review," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4, pp. 1824–1842, 2007.

[13] J. Munoz-Castaner, R. Asorey-Cacheda, F.J. Gil-Castineira, F.J. Gonzalez-Castano, P.S. Rodriguez-Hernandez, "A Review of Aeronautical Electronics and its Parallelism with Automotive. Electronics", *IEEE Trans. On Ind. Electron.*, vol. 54, no. 99, April 2010.

[14] P. Conmy, I. Bate, "Component-Based Safety Analysis of FPGAs", *IEEE Trans. On Ind. Informatics*, Vol.6, n° 2, pp. 195, May 2010

[15] F. Salewski, S. Kowalewski, "Hardware/Software Design Considerations for Automotive Embedded Systems", *IEEE Trans. On Ind. Informatics*, Vol. 4, n° 3, pp. 56, August 2008.

[16] H. Guzman-Miranda, L. Sterpone, M. Violante, M.A. Aguirre, M. Gutierrez-Rizo, "Coping With the Obsolescence of Safety- or Mission-Critical Embedded Systems Using FPGAs", *IEEE Trans. On Ind. Electron.*, vol. 58, no. 3, pp. 814, Marsh 2011

[17] M.A. Gora, A. Maiti, P. Schaumont, "Coping With the Obsolescence of Safety- or Mission-Critical Embedded Systems Using FPGAs", *IEEE Trans. On Ind. Informatics*, Vol. 6, n° 4, pp.

[18] F. Simonot-Lion, "Guest Editorial Special Section on In-Vehicle Embedded Systems", *IEEE Trans. on Ind. Informatics*, vol. 56, no. 4, pp. 372 - 374, Oct 2009.

[19] F. Baronti, E. Petri, S. Saponara, L. Fanucci, R. Roncella, R.Saletti, P. D'Abramo, R. Serventi, "Design and Verification of Hardware Building Blocks for High-Speed and Fault-Tolerant In-Vehicle Networks", *IEEE Trans. On Ind. Electron.*, vol. 58, no. 3, pp.792, Marsh 2011

[20] J.-J. Chen, X. S. Hu, Mossé, D. , L. Thiele, "Guest Editorial Special Section on Power-Aware Computing ," *IEEE Trans. on Ind. Informatics*, vol. 57, no. 3, pp. , July 2010.

[21] J. Choi, H. Cha, "A Processor Power Management Scheme for Handheld Systems Considering Off-Chip Contributions ," *IEEE Trans. on Ind. Informatics*, vol. 57, no. 3, pp. , Jyly 2010.

[22] P. Ferrari, A. Flammini, D. Marioli, A. Taroni, "A Distributed Instrument for Performance Analysis of Real-Time Ethernet Networks", *IEEE Trans. On Ind. Informatics*, Vol. 4, N° 1, pp. 16-25, Feb. 2008

[23] S. Fischmeister, R. Trausmuth, Insup Lee, "Hardware Acceleration for Conditional State-Based Communication Scheduling on Real-Time Ethernet ," *IEEE Trans. on Ind. Informatics*, vol. 56, no. 3, pp 325 - 337, July 2009.

[24] L. Vachhani, K. Sridharan, "Hardware-Efficient Prediction-Correction-Based Generalized-Voronoi-Diagram Construction and FPGA Implementation", *IEEE Trans. On Ind. Electron.*, vol. 55, no. 4, pp. 1558, April 2008

[25] L. Vachhani, K. Sridharan, P.K. Meher, "Efficient FPGA Realization of CORDIC With Application to Robotic Exploration", *IEEE Trans. On Ind. Electron.*, vol. 56, no. 12, pp.

[26] H. Tanaka, K. Ohnishi, H. Nishi, T. Kawai, Y. Morikawa, S. Ozawa, T. Furukawa, "Implementation of Bilateral Control System Based on Acceleration Control Using FPGA for Multi-DOF Haptic Endoscopic Surgery Robot", *IEEE Trans. On Ind. Electron.*, vol. 56, no. 3, pp. 618, March 2009

[27] Li. Tianjian, Y. Fujimoto, "Control System With High-Speed and Real-Time Communication Links", *IEEE Trans. On Ind. Electron.*, vol. 55, no. 4, pp. 1548, April 2008.

[28] J. J. Rodriguez-Andina, M. J. Moure, M. D. Valdes, "Features, design tools, and application domains of FPGAs", *IEEE Trans. On Ind. Electron.*, vol. 54, no. 4, pp. 1810–1823, August 2007.

[29] Xilinx on-line literature. Available in www.xilinx.com

[30] E. Monmasson, M. Cirstea, "FPGA design methodology for industrial control systems – A review," *IEEE Trans. On Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, August 2007.

[31] Altera on-line literature. Available in www.altera.com

[32] Xilinx on-line literature. Available in www.xilinx.com

[33] Actel on-line literature. Available in www.actel.com

[34] E. Monmasson, L. Idkhajine, I. Bahri, M.W. Naouar, L. Charaabi, "Design methodology and FPGA-based controllers for Power Electronics and drive

applications", *In Proc. ICIEA'2010 Conf*., pp 2328-2338, Taichung, Taiwan.

[35] M. Ooi, "Hardware Implementation for Face Detection on Xilinx Virtex-II FPGA Using the Reversible Component Transformation Color Space," in Third IEEE International Workshop on Electronic Design, Test and Applications, Washington, DC, 2006.

[36] S. Paschalakis and M. Bober, "A Low Cost FPGA System for High Speed Face Detection and Tracking," in Proc. IEEE International Conference on Field-Programmable Technology, Tokyo, Japan, 2003.

[37] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan and M. W. Naouar, "FPGAs in Industrial Control Applications," in *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 224-243, May 2011.

[38] Thu-Thao Nguyen, MEng "Field Advisor: Bruce Robert Land", School of Electrical and Computer Engineering, Cornell University,2012.

[39] Ajay V Deshmukh, "Microcontrollers, Theory and Applications", the *MacGraw-Hill companies, Computer Engineering series*, 2007.