International Conference on Control, Engineering & Information Technology
Proceedings- Copyright IPCO-2014, pp.1-13
ISSN 2356-5608

FPGA Implementation of PID Controller

Fares Sassi , Mehdi Abbes, Abdelkader Mami

Laboratory of Analyze, Conception and Control of Systems (LACCS), Team: Modelisation and Control of renewable Energies Systems, University El Manar, Faculty of Sciences of Tunis, Tunis, Tunisia

{ sassifares31@gmail.com ; mehdiabbes@yahoo.com; abdelkader.mami@gmail.com }

Abstract. In this paper, we propose an implementation of a synthesizable Very High Speed Integrated Circuits Description Language program of Proportional-Integral-Derivative controller on a map XC3S700A Xilinx Starter Kit using the Xilinx ISE 10.1 software. The control strategy was applied to a second order state system. The Very High Speed Integrated Circuits Description Language was used as a programming tool. The use of Field Programmable Gate Array circuits presents a good choice regarding to the problem of execution time. A Proportional-Integral-Derivative Matlab program was also implemented in order to make a performance comparison.

Keywords: FPGA, MATLAB/SIMULINK, Digital PID Controller,VHDL.

1 Introduction

The first realizations of digital implementation of control algorithms were performed using microprocessors. These numerical solutions have solved the problems associated with the use of analog controls. They also have a large economic interest and a better design flexibility. With technological advancement in the field of microelectronics, new digital solutions such as FPGAs (Field Programmable Gate Array) are available and can be used as targets for the implementation of digital control algorithms [1]. The programmable FPGA circuits (Field Programmable Gate Array) didn't escape to this law. The circuits FPGA were initially developed like a natural evolution of the CPLD (Complex Programmable Logic Devices), but with further complexity in every new integration until reaching one billion of transistors for the most recent generations. In fact, the level integration increase is mainly due in the similar power growth of the computations of these circuits. One of the proposed solutions was to use the FPGAs in order to make the fast samples and still find their place in many industrial and electronic domains. The majority of the industrial systems control controllers are PID controllers. From this point of view, it will be necessary to digitalize the PID algorithm. In general, the modern digital control systems require stronger and faster calculation components [13]. This type of elements becomes yet indispensable with the use of some new control algorithms as the adaptive control, the fuzzy control, and, the sliding mode control…[6] [7]. Despite the fact that the PID controllers are the oldest ones, they still represent the most used controllers in the industrial control systems.The target FPGA device used in this paper is Spartan-3A manufactured recently by Xilinx [4]. Design development and debugging is carried on a low-cost, full featured kit provided by Digilent. This board provides all the tools required to quickly begin designing and verifying Spartan-3 platform designs.

In this paper, we present an implementation of a PID controller applied to a second order state system using a synthesizable VHDL integer program [2]. It has the advantage to effectively represent the real type not synthesizable in VHDL better than fixed point libraries which take a lot of space on the FPGA, and they also require some clock cycles to generate results. The results are compared with those obtained in Matlab.

The paper is structured as follows: in Section II and Section III, we present a theoretical review of PID controller and system to control. In Sections IV and V, we present the controlled state system VHDL design, their simulations and implementation results. The last section is devoted to conclude this paper.

2 PID Controller

Proportional-Integral-Derivative (PID) controllers are widely used in automation systems. They are usually implemented either in hardware using analog components or in software using computer-based systems [8] [9].This paper outlines several modules necessary for building PID controllers on Field Programmable Gate Arrays (FPGAs) which improve speed, power and cost effectiveness.

The corresponding discrete PID equation is given [12]:

$$U(z)=K_iT_e(\frac{z}{z-1})E(z)-[K_p+\frac{K_d(z-1)}{T_ez}]y(z) \quad (1)$$

Where $$E(z)=Y_c(z)-Y(z) \quad (2)$$

$Y_c(z)$: The set point signal.

$Y(z)$: The feedback signal.

$E(z)$: The error signal.

$U(z)$: The control signal.

Where K_p, K_i, K_d are respectively the proportional, integral, derivative parameters. The PID controller is shown in Fig. 1.

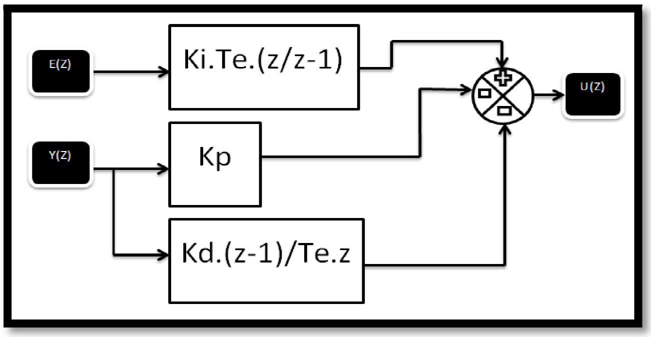


Fig. 1. A block diagram of PID controller

The PID parameters are computed using the Takahashi method in closed loop [11].

$K_p = 0.25$, $K_i = 0.18$ and $K_d = 10$

3 System To Control

In this example, the proposed process is a speed control which consists in a motor fitted with a speed sensor in order to regulate the speed of the motor by manipulation of the input voltage [10]. The mathematical model of the system with a sampling time of 0.1 sec is [5]:

$$
\begin{aligned}
x(k+1) &= \begin{pmatrix} 0.93 & -0.01 \\ 0.04752 & 0.9964 \end{pmatrix} x(k) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u \\
y(k) &= (-0.01 \quad 3.71)\, x(k)
\end{aligned}
\tag{3}
$$

4 Implementing A PID Controller On The Map Xilinx Spartan 3A Starter Kit XC3S700A

4.1 Simulink Model

The application of a PID controller in a feedback control system is shown in Fig. 2.

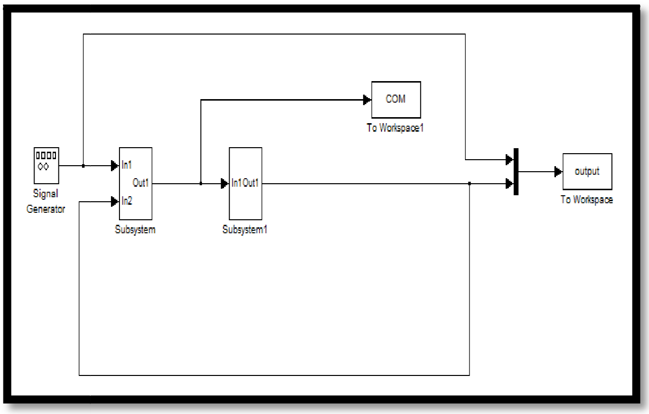


Fig. 2. Simulink implementation of controlled system

4.2 VHDL Programming

The design includes:

- The “Set_point”: integer type written in VHDL as a state machine.
- The “PID controller”: the VHDL algorithm block written in integer.
- The “system”: the VHDL algorithm block written in integer.

The three blocks are synchronized by the same clock “clk”.

The VHDL PID controller program has four inputs and output:

- Inputs: Clock (frequency 50 MHz), clock enable, reset and the set_point is of “std_logic_vector (11 downto 0)” type: It is a periodic signal written in VHDL as a state machine.
- Outputs: The control is "std_logic_vector (11 downto 0)" vector.

The VHDL System program has four inputs and output:

- Inputs: Clock (frequency 50 MHz), clock enable, reset and the control is of “std_logic_vector (11 downto 0)”.
- Outputs: The output is "std_logic_vector (11 downto 0)" vector.

The choice of 12 bits is due to the use of the DAC for displaying the output on the oscilloscope.

5 Simulation and Implementation Results

In this section, we present the simulations of the programs carried out on Simulink and Xilinx ISE 10.1 with Xilinx ISE simulator as well as the interpretation results.

5.1 Simulation Results By Simulink

The resulting curves of the Matlab simulation are illustrated in Fig. 3 and Fig. 4.

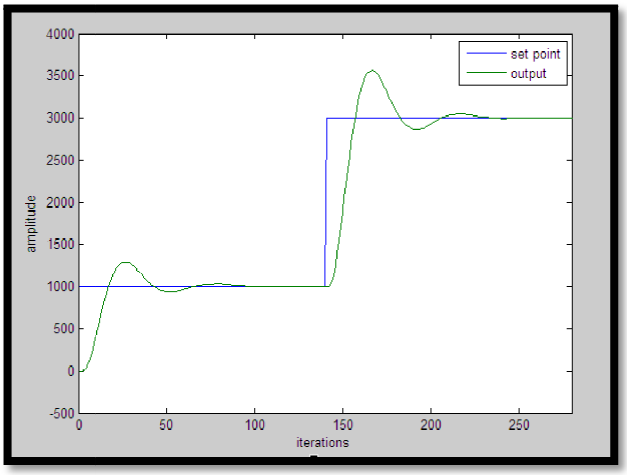


Fig. 3. Set point and output signals of the matlab

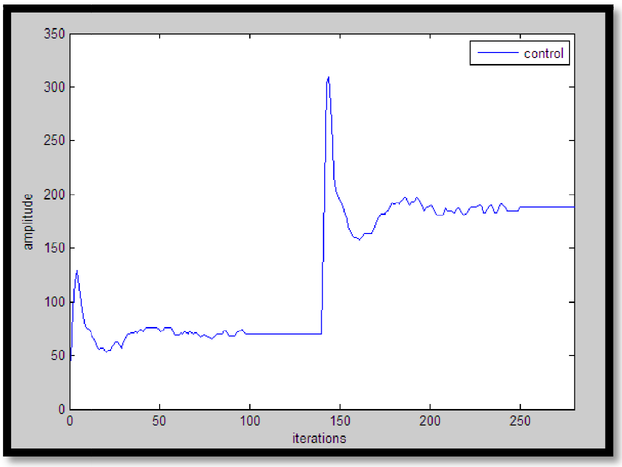


Fig. 4. Matlab control signal of PID controller

5.2 Simulation By The ISE Simulator

The simulation using the ISE simulator is shown in Fig. 5. The set point varies between 1000 and 3000.

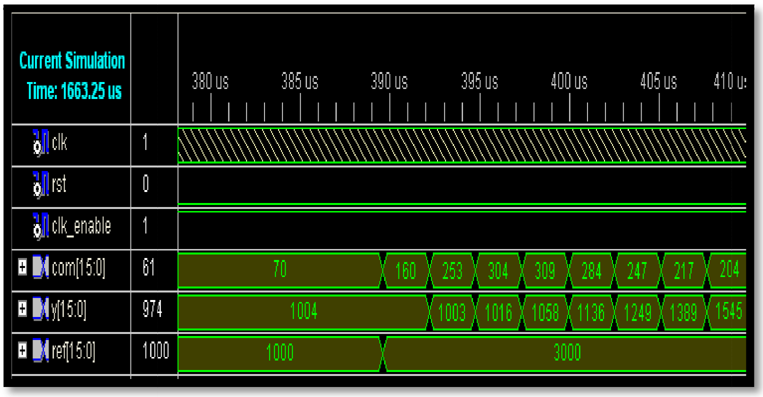


Fig. 5. Simulation of PID control system by the ISE simulator

The clock frequency is fixed at 50 MHz (20ns) which is equal to the clock frequency of the XC3S700A Starter Kit.

The output reaches the set point value after 20 iterations.

We note that the value of the output is stabilized with the first set point value (1000), when there is a change in the set point value, the value of the output starts to increase until it reaches the second value set point (3000).

5.3 Digital-To-Analog Converter

As shown in Fig. 6, the FPGA uses a Serial Peripheral Interface (SPI) to communicate digital values to each of the four DAC channels. The bus SPI is a full-duplex, synchronous, character-oriented channel employing a simple four-wire interface [4]. A bus master (the FPGA in this example) transmits serial data (SPI_MOSI) to the selected bus slave (the DAC in this example) and drives the bus clock signal (SPI_SCK). Meanwhile, the bus slave transmits the serial data (SPI_MISO) back to the bus master.

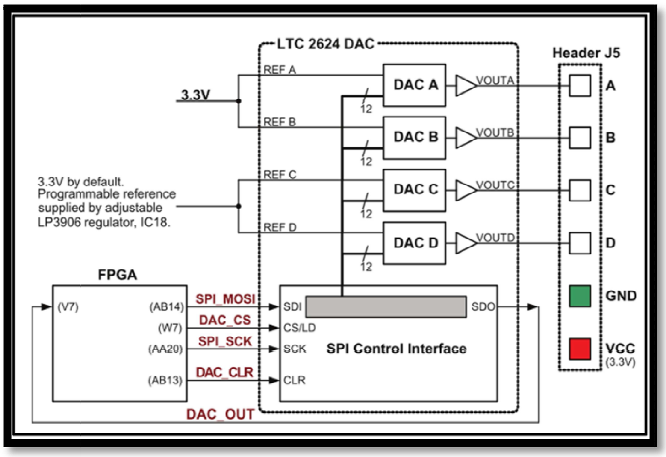


Fig. 6. Digital-to-Analog Connection Schematics

Sending data is synchronous with DAC_SCK which operates at a maximum frequency of 50 MHz, it is programmed in a state machine that is illustrated by the following transition diagram:

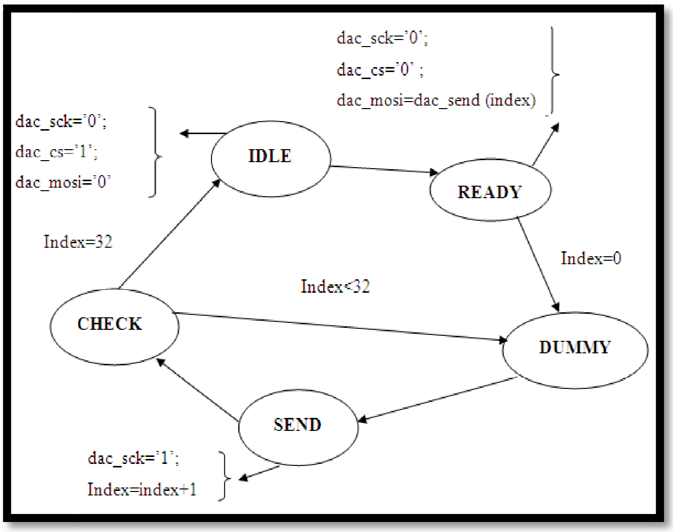


Fig. 7. State diagram of the VHDL program DAC

During the state (idle), the frame that contains the data we want to convert by setting DAC_CS at the highest level. Then we start to read the frame during the state (ready). After this task, and during the state (send), we start to send the frame that contains 32-bit when the clock DAC_SCK is set to high level as it is prepared to the state (dummy). Finally, after checking the transmission of the frame that ensures the state (check), we put DAC_CS high in order to start sending a new frame.

After converting the binary data in decimal, the analog value output is calculated by the formula (4).

$$Vdac = \frac{data[11:0]}{4096} \times Vref \tag{4}$$

With $Vref$ = 3.3 V

5.4 Implementation Results

After the execution of the synthesis process [3], the Xilinx ISE software generates the RTL diagram illustrated in Fig. 8, as well as statistics on the number of used hardware resources which are depicted in Fig. 9.

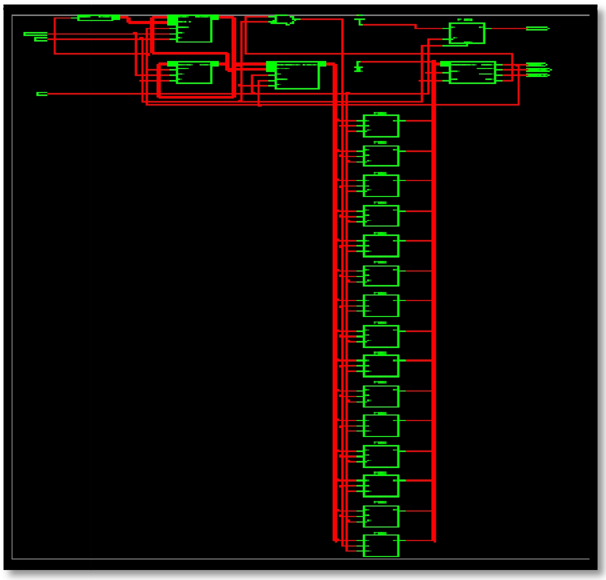
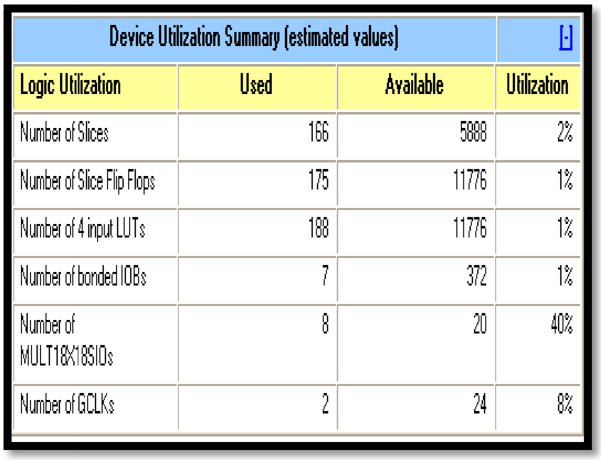


Fig. 8. RTL schematic design "PID, system and DAC"



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	166	5888	2%
Number of Slice Flip Flops	175	11776	1%
Number of 4 input LUTs	188	11776	1%
Number of bonded IOBs	7	372	1%
Number of MULT18X18SIOs	8	20	40%
Number of GCLKs	2	24	8%

Fig. 9. Statistics on the material resources used to implement the "PID, system and DAC" design

Then it executes the implementation process (translate, Map, Place & Route). After that, the corresponding bit stream file is generated [3].

Finally, we implement the VHDL program "PID, system and DAC" on Spartan 3A (Fig. 10) the map using the Xilinx ISE 10.1 software and JTAG cable.

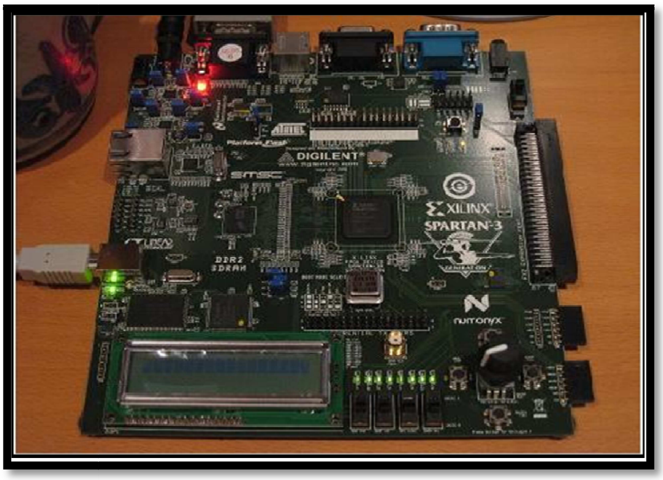
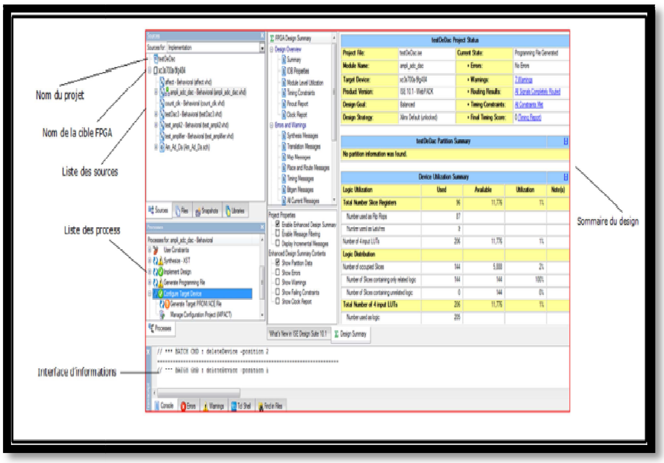


Fig. 10. Circuit Xilinx XC3S700A Spartan 3A Starter Kit



testDeDac Project Status			
Project File:	testDeDac.ise	Current State:	Programming File Generated
Module Name:	ampl_adc_dac	• Errors:	No Errors
Target Device:	xc3s700a-4fg484	• Warnings:	7 Warnings
Product Version:	ISE 10.1 - WebPACK	• Routing Results:	All Signals Completely Routed
Design Goal:	Balanced	• Timing Constraints:	All Constraints Met
Design Strategy:	Xilinx Default (unlocked)	• Final Timing Score:	0 (Timing Report)

testDeDac Partition Summary

No partition information was found.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Total Number Slice Registers	96	11,776	1%	
Number used as Flip Flops	87			
Number used as Latches	9			
Number of 4 input LUTs	206	11,776	1%	
Logic Distribution				
Number of occupied Slices	144	5,888	2%	
Number of Slices containing only related logic	144	144	100%	
Number of Slices containing unrelated logic	0	144	0%	
Total Number of 4 input LUTs	206	11,776	1%	
Number used as logic	205			

Fig. 11. The Xilinx ISE 10.1 software

The VHDL program "PID, system and DAC" is implemented in Spartan 3A map using the Xilinx ISE 10.1 software (Fig. 11) and JTAG cable. The output signal on channel 2 and the set point signal on the chain 1 are then displayed together on the screen of the oscilloscope. The result of the implementation is depicted in Fig. 12. The output signal on channel 2 and the signal control of the chain 1 are then displayed together on the screen of the oscilloscope. The result of the implementation is shown in Fig. 13.

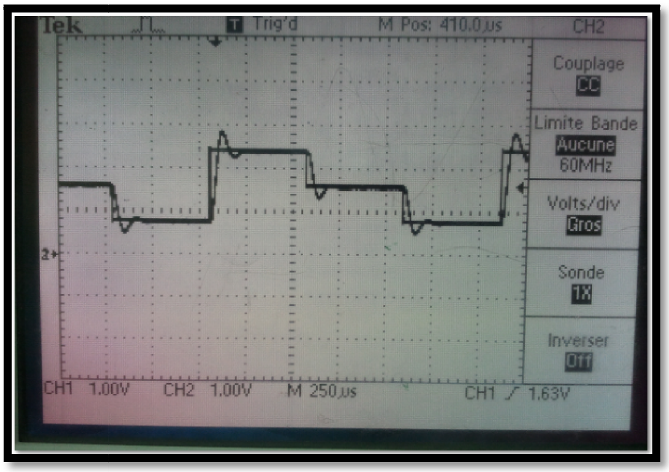


Fig. 12. Result of the implementation of the oscilloscope set point and output

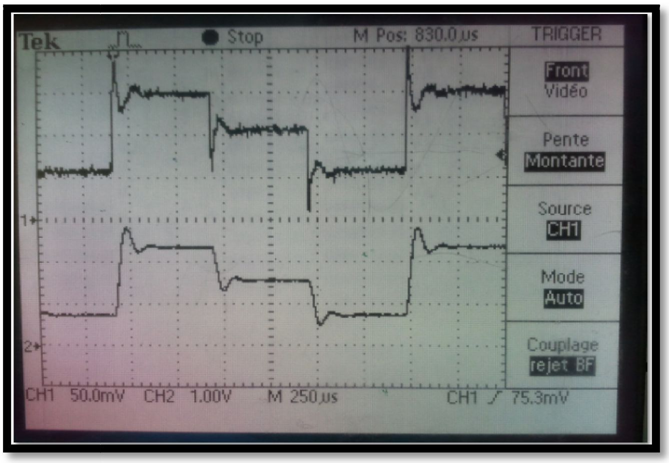


Fig. 13. Result of the implementation of the oscilloscope control and output

5.5 Interpretation Results

Simulation results and implementation, obtained by Matlab and ISE show that the speed and the right followed reference trajectory by the output signal.

The difference between the results of simulation and implementation on FPGA is due to the difference interpretation of integer type used in VHDL program and real type used in the Matlab program.

Implementing PID controllers on FPGA features speed, accuracy and power over other techniques of digital implementation.

6 Conclusion

In the current investigation, an implementation of PID controller on a map XC3S700A, FPGA-based, is performed by writing a synthesizable VHDL integer program. A comparison of VHDL signals to those obtained by Matlab is carried also out. The use of integer type provides good results because it solves the overflow problems during the computations.

As a future extension of the current investigation, we will try to control a real system using a FPGA board.

References

[1] M.W. NAOUAR, « Commande numérique à base de composants FPGA d'une machine synchrone », algorithmes de contrôle du courant, Thèse de Doctorat, Ecole Nationale d'Ingénieurs de Tunis et l'Université de Cergy Pontoise, Tunis, 2007.

[2] D.L. Perry: "VHDL", McGraw-Hill, 2004.

[3] E. Monmasson and Y. A. Chapuis, "Contributions of FPGAs to the control of Electrical System, A Review," IEEE Trans. Ind. Electron. Society Newsletter, vol.49, no.4, pp.8-15, December. 2002.

[4] Spartan-3A/3AN FPGA Starter Kit Board User Guide, Xilinx, 2008.

[5] A. Kheriji Abbes, F. Bouani and Mr. Ksouri. "A Microcontroller Implementation of Model Predictive Control". ICCESSE International Conference on Computer Electrical and Systems Sciences and Engineering, Paris, France, June 2011.

[6] L. Samet, N. Masmoudi, M.W. Kharrat, L. Kamoun: A Digital Pid Controller for Real Time and Multi Loop Control, 5ème Colloque d'Informatique Industrielle CII'98 8,9 et 10 février 1998,Djerba Tunisie.

[7] L. Samet, "Etude de l'intégration électronique en technologie FPGA d'un algorithme de contrôle de processus: le PID" Thèse Docteur Ingénieur, ENIS-TUNISIE, décembre 1996.

[8] Mohamed Abdelati, the Islamic University of Gaza, Gaza, Palestine:" FPGA-Based PID Controller Implementation".

[9] Y. F. Chan, M. Moallem, W. Wang, "Efficient implementation of PID control algorithm using FPGA technology", Proceedings of the 43ed IEEE Conference on Decision and Control, V5, PP. 4885-4890, Bahamas 2004.

[10] G. Palomo, K. Hilton, and J. Rossiter, "Predictive control implementation in a plc using the iec 1131.3 programming standard," in Proceedings of American Control Conference, 2009.

[11] A. Voda and S. Gentil, Regulateur PID analogique et numeriques. Technique de l'ingenieur, 1999.

[12] Franklin G.F., J.D. Powell and M.L., Workman. 1990. Digital Control of Dynamic System. MA: Addison-Wesley Publishing Company.

[13] Anthony Cataldo, "Low-priced FPGA options set to expand " Electronic Engineering Times Journal, N 1361, PP 38-45, USA 2005.