Volume 54, Number 4, 2007 **ACTA TECHNICA NAPOCENSIS**
Electronics and Telecommunications

MODEL-BASED DESIGN AND FPGA IMPLEMENTATION OF CONTROL SYSTEMS

Gabriel OLTEAN*, Botond Sandor KIREI*, Irina DORNEAN*, Ioana OLTEAN**
**Bases of Electronics Department, Technical University of Cluj-Napoca*
26-28 G. Baritiu str., Cluj-Napoca, Romania, Tel: +40264401470; Fax: +40264591340;
** Grup Școlar Electrotehnic „Edmond Nicolau", Cluj-Napoca
Gabriel.Oltean@bel.utcluj.ro

Abstract

FPGAs have become a very promising solution for the realization of digital control systems. In this paper an efficient model-based design methodology for FPGA implementation of a control system is presented. The Matlab/Simulink environment is used here for modeling, simulation and tuning a temperature control system based on PID controller. Verilog HDL language describes the digital control system to be implemented on a FPGA development board. Results prove that the proposed methodology shortens the design cycle, because the whole system can be fully simulated and adjusted before its final design and implementation.

Keywords: *PID controller, Verilog, FPGA*
.

MODEL-BASED DESIGN AND FPGA IMPLEMENTATION OF CONTROL SYSTEMS

Gabriel OLTEAN*, Botond Sandor KIREI*, Irina DORNEAN*, Ioana OLTEAN**
*Bases of Electronics Department, Technical University of Cluj-Napoca
26-28 G. Baritiu str., Cluj-Napoca, Romania, Tel: +40264401470; Fax: +40264591340;
** Grup Şcolar Electrotehnic „Edmond Nicolau", Cluj-Napoca
Gabriel.Oltean@bel.utcluj.ro

Abstract

FPGAs have become a very promising solution for the realization of digital control systems. In this paper an efficient model-based design methodology for FPGA implementation of a control system is presented. The Matlab/Simulink environment is used here for modeling, simulation and tuning a temperature control system based on PID controller. Verilog HDL language describes the digital control system to be implemented on a FPGA development board. Results prove that the proposed methodology shortens the design cycle, because the whole system can be fully simulated and adjusted before its final design and implementation.

Keywords: *PID controller, Verilog, FPGA*

1 INTRODUCTION

The PID (Proportional, Integral and Derivative) controller has a wide variety of applications in the technical world. Due to its performance it is widely used in dynamic systems in many different areas such as aerospace, process control, manufacturing, robotics, automation, transportation systems or home appliance products [1], [2]. The implementation of the control systems is possible on several hardware platforms (microcontroller, FPGA) and its development on digital hardware is eased by HDL languages [3] and FPGA technology. These controllers offer advantages such as high-speed computation, complex functionality, real-time processing capabilities, and low power consumption [4].

Several papers dealing with digital PID controller implementation using FPGAs are reported in the literature in the past few years. They address different aspects: a design methodology involving Matlab/Simulink modeling [2], a modular FPGA-based approach to design a temperature control system [1], an efficient implementation using a Distributed Arithmetic based scheme [4], a PID controller core suitable to be introduced into a System-on-Programmable Chip design [5]. Also, computational intelligence techniques as fuzzy logic [6], [7], [8], [9] and genetic algorithms [10] are involved in the design and implementation of FPGA based control system.

Model-based design provides a valuable technique for creating embedded control systems. It supposes a hierarchical design process in which the entire design is initially defined at a conceptual level and detail is added as necessary to deliver the needed functionality [12]. The model is used to define specifications, evaluate design and system performance, automatically generate code, perform hardware-in-the-loop testing and create a software-based test harness for testing production hardware. This approach can substantially reduce development time by rapidly leading to complete and functional proof-of-concept designs and enabling rapid design iterations and parameter optimization through a unified design, simulation, and test environment.

This paper presents a model-based design methodology for FPGA implementation of a control system. Our approach starts with the development and simulation of the entire system in Matlab/Simulink environment. The architecture of the digital system is build based on this model and after a bit width analysis. The FPGA implementation uses the Verilog HDL language to describe the digital control system.

The organization of the paper is given as follows. In Section II the methodology for model-based design of a control system is presented, including a flow chart and explanations for each step. The control system, composed by a process (oven) and a PID controller is described in Section III. Section IV shows the Simulink models of the controlled process and of the PID controller together with simulation results. The architecture of the implemented control system and a brief description of its sub-modules can be found in Section V. Experimental results and synthesis report is presented in Section VI. Conclusions and further development possibilities are discussed in Section VII.

2 PROPOSED METODOLOGY

The implementation of a control system is always a challenging task. Fig. 1 presents the flow chart of the proposed methodology.

The starting point of the implementation is represented by

the system specifications. Based on this information a Simulink model is developed. This Simulink model is analyzed and tested until the simulation result matches the expected performance.
Using this Simulink model the digital system's architecture is created by performing the bit width analysis. The architecture can be easily translated to an HDL language and the code for the core of the control system simulated. When the simulation results matched the expectations, the FPGA implementation is made. This step represents a routine task; the interface with the control system core is made.

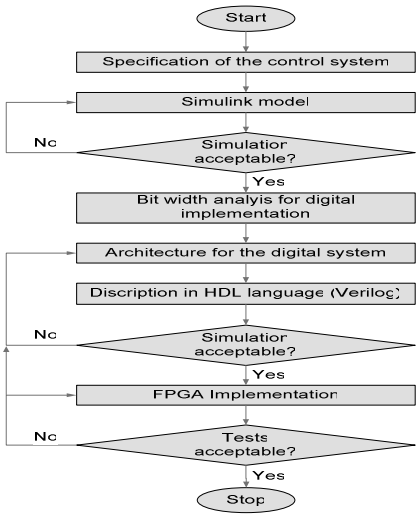

Figure 1. Flow chart for the applied methodology.

3 THE CONTROL SYSTEM

The block diagram of a feedback control system is shown in Fig. 2 [4]. The u_C is the set point signal, y is the feedback signal, e is the error signal and u is the control input signal of the process.

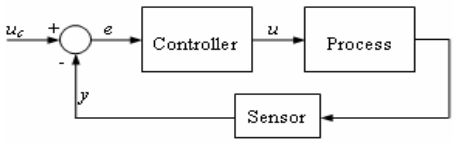

Figure 2. Block diagram of a control system

Based on this schematic, the paper provides the implementation of a temperature control system using a PID controller. The process under control is an oven that is warmed up by means of an electrical heater. For our temperature control system the u_c signal is the set point temperature (denoted T_s), the feedback signal y is the controlled oven temperature (*Tout)*, the error is *e=Ts-Tout*, and the control input of the oven is the necessary power for the heater, (denoted *power)*. To model the heat generation and transfer, the equivalent circuit [13] depicted in Fig. 3 is used. The current source *power* (thermal power) represents the power provided to the heating element.

The electrical heater has its heat capacity $C_h=500[J/°C]$ and its thermal resistance $R_{h0}=0.143[°C/W]$. The heat capacity of the oven is $C_0=1000[J/°C]$. The oven looses

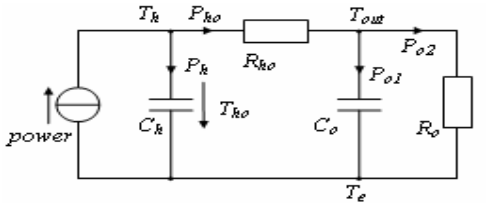

Figure 3. Model of the oven

heat to the environment at the temperature T_e through the thermal resistance $R_0=0.1[°C/W]$ of its insulation.
The simplest form of a PID control algorithm is given as:

$$u(t) = P\left[e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}\right] \quad (1)$$

where P is the controller gain, T_i the integral time constant and T_d the derivative time constant. Equation (1) can also be written in the following form:

$$u(t) = Pe(t) + I\int_0^t e(t)dt + D\frac{de(t)}{dt} \quad (2)$$

using $I = \frac{P}{T_i}$ and $D = PT_d$.

The tuning of the controller consists of determining the values of these three parameters (P, I, D) with the aim of satisfying different control specifications such as set point following, load disturbance attenuation, robustness to model uncertainties and rejection of measurement noise [10]. According with [11], P constant affects the rise time, the overshoot and the steady-state error; I constant affects the rise time, the overshoot, the settling time and the steady-state error, while D constant affects the overshoot and the settling time.

4 THE SIMULINK MODEL

According with our previously presented model-based design method we proceed with the development of the discrete Simulink model of the temperature control system. All the internal signals were declared as signed or unsigned integer with round integer calculation toward zero. The block diagram of the Simulink model is presented in Fig. 4.

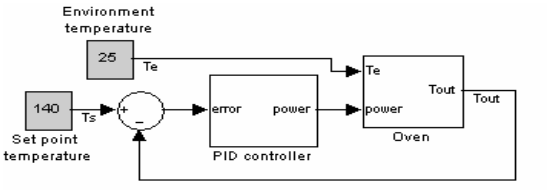

Figure 4. Simulink model of the temperature control system

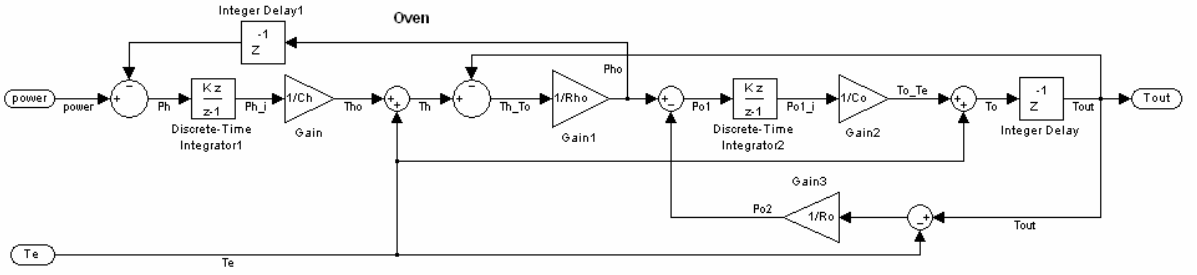

Figure 5. Simulink model of the oven

The user may set the environment temperature *Te* in the range [10; 30][^{o}C] and the set point temperature in the range [100; 250][^{o}C]. The oven block is modeled according with the circuit in Fig. 3. The Simulink model of the oven is detailed in Fig. 5. It has two inputs: *power* - the power for the electrical heater and *Te* - the environment temperature and one output *Tout* – the temperature inside the oven.

By way of example, the time behavior of the heat capacitance C_h is modeled by means of Discrete-Time Integrator1 and Gain blocks. P_h is the equivalent thermal current flowing through the capacitor and T_{ho} is the equivalent thermal voltage drop across the capacitor. These two blocks implement the equation of the equivalent capacitor:

$$T_{ho}(t) = \frac{1}{C_h}\int_0^t p_h(t)dt \qquad (3)$$

The Integer Delay blocks are used in order to get rid of the algebraic loop containing blocks with discrete-valued outputs that can not be solved in the Simulink environment.

The Simulink model of the PID controller is presented in Fig. 6. The input of the model is the *error* signal. The proportional term is implemented by the *Product2* block, the integral term by the *Discrete-Time* integrator, *Product* and *Shift Arithmetic* blocks and the derivative term by the *Product1* and *Discrete Derivative* blocks. The controller has a saturated *power* output signal in the range [0; 5000][W]. The PID controller was tuned considering the environment temperature T_e=*25°C* and the set point temperature T_s=*140°C* using the CDHW method [13]. Because the integral constant *I* has an under unity value, there was necessary to multiply this value and use instead the constant I_s=*1024 I* to obtain enough computation accuracy.

After integral component calculation we have to scale down the result by using the *Shift Arithmetic* block (shift right by 10 bits) to divide it by 1024.

In the Simulink environment the sample time was set to 1s for all blocks.

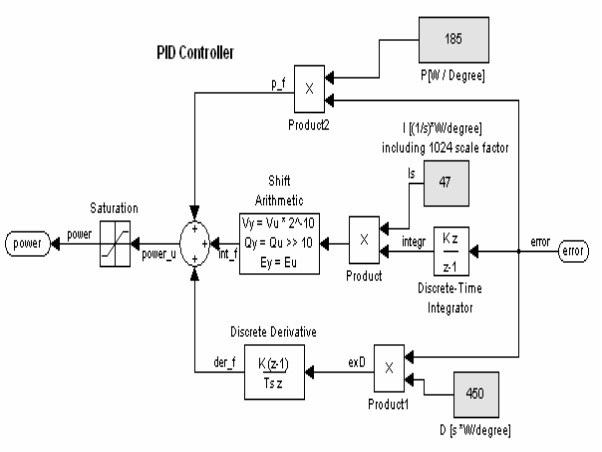

Figure 6. Simulink model of the PID controller

The results simulating the operation of the control system are presented in Fig. 7. The rise time is approximately 400s and the overshooting is 4.29%. The temperature increases up to 146^{o}C, being followed by a temperature decrease down to 138 ^{o}C. Starting with 714s, *Tout* presents some very small oscillations between 139^{o}C and 140^{o}C. The steady-state error becomes zero after 24549s. The *p_f, int_f* and *der_f* signals represent the proportional, integral and derivative components of the PID controller. The output of the controller, *power* takes the maximum possible value (5000W) in the beginning of the heating process. The power starts to decrease after 281s, when T_{out}=*116°C*. One can observe the oscillations in the *power* signal due to the derivative and proportional components. Due to 1°C resolution for the oven temperature and for the error, each time the temperature changes, power suffer some step variations. It can be noticed the continuous increase of the integral component that finally set the optimal value of the necessary heating power (1150°W). From this point forward (24548s) the temperature remains constant at 140°C. The heat wasted by the oven to the environment is fully compensated by the heat generated by the electrical heater.

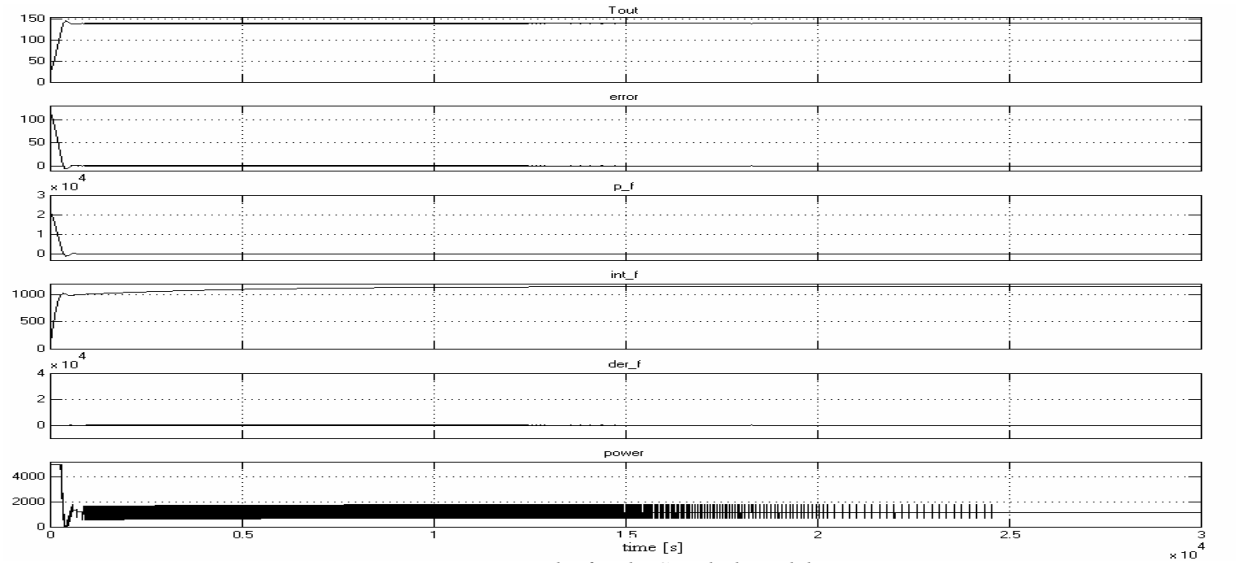

Figure 7. Results for the Simulink model

5 DESIGN OF THE DIGITAL SYSTEM

Although the Simulink model uses discrete simulations and integer values further effort has to be done for the implementation. The most important step is the bit-width analysis performed on the Simulink model's signals. In the model all signals are represented as signed integer values on 32s bit using 2's complement representation, in order to avoid the combinations which appear in the case of '-' sign. In the HDL code the signals width is reduced to the necessary value for sign-magnitude representation, resulting in the reduction of routing and logic. It is necessary to do that because, if in the Simulink representation it is possible to work only with 32, 16, 8 bits signals, in the Verilog implementation the signals can have different widths (smaller or greater). There are some modules (one of them is the controller interface) that do not worth to be modeled, so these modules are left for the designer's routine. On Fig. 8 the control system's architecture is depicted.

The core named Temperature Control includes the sub-blocks: Oven behavior and PID control. The testing of the control system can be done by emulating the ambiance. The emulation is made in the Ambience Emulation block. The proper control is implemented in the PID control block (see Fig. 8).

The necessary signals for the Temperature Control block are shown in Fig. 9. The clock signals for PID control block are generated by the *Clock signal management* module, beside of CE signal used by other modules. The *Debouncer* block is the interface for the buttons placed on the development board and its primary function is to take out the eventual glitches generated by the mechanical contacts. The *Temperature selection* block holds the set point temperature value and provides this

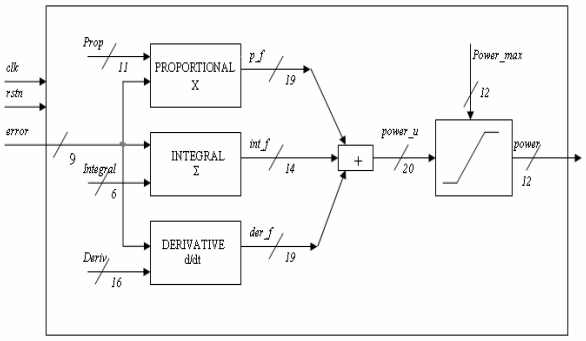

Figure 8. HDL PID control block diagram

number to the *Seven segment display* block. It provides the necessary signals for the multiplexed 4 digit seven segment displays located on the board.

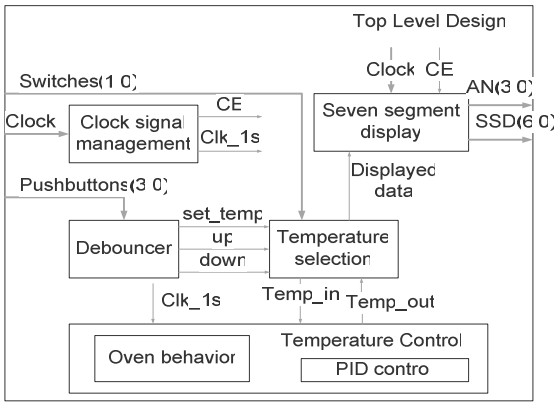

Figure 9. The architecture of the Control System.

6 RESULTS

The implementation and testing was done on the Nexys [14] development board, equipped with XC3S1000 FPGA form the Spartan-3 family manufactured by Xilinx. The board is equipped with 4 pushbuttons, 4 switches, a 4 digit seven segment display. Aria and speed optimized synthesis results for the control system are presented in Table 1. Most of the logic is occupied by the interface modules.

Optimi-zation effort	Slice Flip Flops	4 input LUTs	Bonded IOBs	MULT 18X18s
Aria	214 (1%)	1050 (6%)	16 (9%)	5 (20%)
Speed	218 (1%)	1058 (6%)	16 (9%)	5 (20%)

Table 1. Synthesis result.

The experimental results obtained for the final digital temperature control system are presented in Fig. 9. Also the transient regime in the beginning phase of the heating process is detailed (zoomed in) in Fig. 10. The accuracy of the digital system design and implementation is fully confirmed by these results that are almost identical with the ones offered by the Simulink model (see section IV).

The rise time for the temperature is approximately 273s (T_{out}=*126°C*). The overshoot is 5%, the temperature increasing up to 147°C in 385s. The response of the system also presents an undershoot of -2.14%, the temperature decreasing down to 137°C in 550s. Then the temperature increases slowly reaching the set point value (140°C) after 2412s from the beginning of the heating process. From this point forward, *Tout* presents very small oscillations between 139°C and 140°C. Considering the resolution of the temperature calculation (1°C) with round integer calculation toward zero (specific to digital systems), we can consider that the system enters its steady-state regime. A temperature of 139°C appears even in the case where the real temperature is 139.99°C due to the round integer calculation. Anyway a constant zero steady-state error computed in our digital temperature control system appears after 26508s. The output of our controller, *power* takes the maximum possible value (5000W) in the beginning of the heating process. The power starts to decrease after 249s, when T_{out}=*116°C*. One can observe the oscillations in the *power* signal due to the derivative and proportional components. As a result of the computation resolution for the oven temperature and for the error, each time the temperature changes, power suffers some step variations, both in the transient regime and in the steady-state regime up to the point when the oven temperature remains constant to its setting point value. The optimal value for the *power,* given by the integral component of the PID controller is *1150°W*. The heat wasted by the oven to the environment is now fully compensated by the heat generated by the electrical heater.

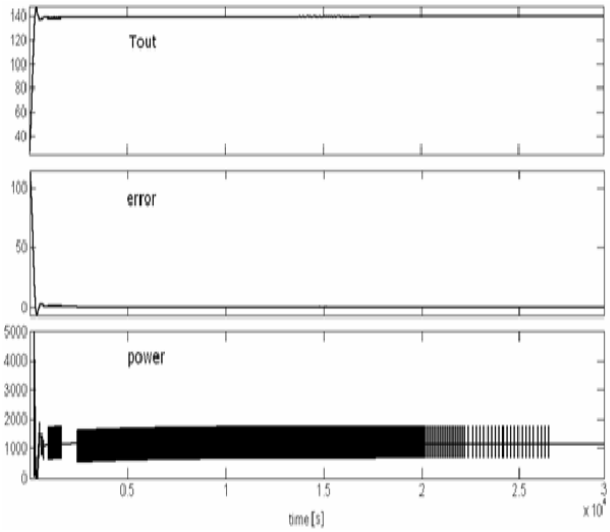

Figure 10. Response of the digital temperature control system

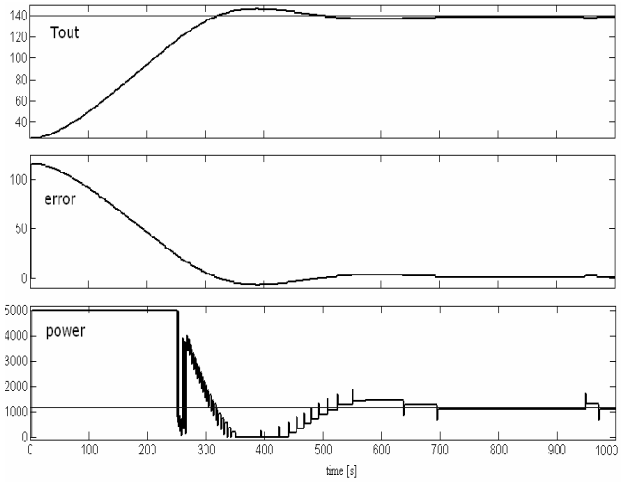

Figure 11. Response of the digital temperature control system – transient regime

7 CONCLUSIONS

In this paper an efficient model-based design methodology for FPGA implementation of a control system was presented. The methodology exploits the advantages conferred by the availability of a very accurate model before final design and implementation. This way the control system can be fully adjusted and tested in the simulation environment without any effort, cost or time consumption specific to hardware systems development. The experimental results proved the validity of the

proposed methodology.

However there are some aspects needing further research work. A fixed-point numerical representation has to be used in order to increase the resolution and by the way of consequence the accuracy. To improve the performances of the control some auto-tuning mechanisms should be integrated; computational intelligence techniques (fuzzy logic or genetic algorithms) are appropriate candidates for this task.

8 REFERENCES

[1] Y.F. Chan, M. Moallem, W. Wang, "Efficient Design and Implementation of Modular FPGA-Based PID Controllers", *Industrial Electronics, Ieee Transactions 0n*, Volume 54, pg. 1898 – 1906, 2007.

[2] J. Lima, R. Menotti, J. M. P. Cardoso, E. Marques, "A Methodology to Design FPGA-based PID controllers", *Systems, Man and Cybernetics, IEEE International Conference on*, Volume 3, pp. 2577 – 2583, 2006.

[3] B. S. Kirei, I. Dornean, A. Fazakas, M. Topa, "Comparing Verilog and VHDL", *MicroCAD 2007* International Scientific Conference, Proceedings of , pp. 35-40, 2007.

[4] Y.F. Chan, M. Moallem, W. Wang, "Efficient Implementation of PID Control Algorithm using FPGA Technology", *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, Volume 5, pg. 4885 – 4890, 2004.

[5] J. Lazaro, A. Astarloa, J. Arias, U. Bidarte, A. Zuloaga, "Simulink/Modelsim Simulable VHDL PID Core for Industrial SoPC Multiaxis Controllers*", IEEE Industrial Electronics, IECON 2006* - 32nd Annual Conference on, pg. 3007 – 3011, 2006.

[6] V. Tipsuwanpom, T. Runghimmawan, S. Intajag, V. Krongratana, "Fuzzy logic PID controller based on FPGA for process control", *Industrial Electronics, 2004 IEEE International Symposium on*, Volume 2, pg. 1495- 1500, 2004.

[7] G. Kaur, A. Chauhan, P.V. Subramanyam, **"**Fuzzy Logic Based Temperature Controller", *Granular Computing, 2005 IEEE International Conference on***,** Volume 2,pg. 492- 495, 2005

[8] F.M. Yasin, A. Tio, M.S. Islam, M.I. Reaz, M.S. Sulaiman, "The hardware design of temperature controller based on fuzzy logic for industrial application employing FPGA", Microelectronics*, 2004. ICM 2004 Proceedings. The 16th International Conference on*, pg. 157- 160, 2004

[9] S. Alcantara, C. Pedret, R. Vilanova and R. Moreno, "An undergraduate laboratory course on fuzzy controller implementation in FPGAs", *Control & Automation, 15th Mediterranean Conference on, 2007*.

[10] A. Visioli, "Tuning of PID controllers with fuzzy logic**",** *Control Theory and Applications, IEE Proceedings of*, Volume 148, pg. 1-8, 2001.

[11] J. Zhong, "PID Controller Tuning: A Short Tutorial", Mechanical engineering, Purdue University, 2006, http://www.daimi.au.dk/~plolk/project/PIDcontrollertuning.pdf/

[12] A. Mulpur, Z. Galijasevic, "Model-Based Design Shortens Development Time of Motor Control Applications", Embedded Technology Journal, Techfocus Media, August, 2006, http://www.embeddedtechjournal.com.

[13] C.D.H. 13iams, "Feedback and Temperature Control", School of Physics, University of Exeter, http://newton.ex.ac.uk/teaching/CDHW/Feedback/

[14] Digilent Nexys Board Reference Manual, www.digilentinc.com/Data/Products/NEXYS/Nexys_rm.pdf