Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 55(69), Fascicola 2, 2010

High Speed Digital Controller Implemented with FPGA Board

Nanu Sorin[1], Lie Ioan[2], Belgiu George[3], Muşuroi Sorin[4]

Abstract: The purpose of the paper is to present a structure of digital control system where the programmed logic digital controller was replaced with a wired logic solution in order to drastic increase the computational speed. Two control systems were designed, implemented and experimented. The experimental results are presented within the paper.
Keywords: digital control system, wired logic, FPGA, sampling time, high speed.

I. INTRODUCTION

The classical digital control system has a block diagram as in Fig.1.

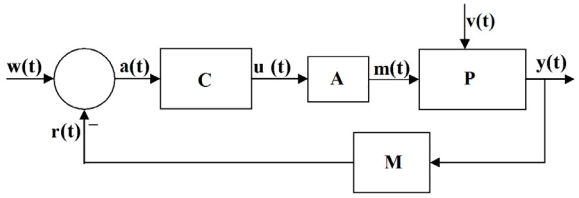

Fig.1. Control system structure

Where C-controller, P- process, A- amplifier, M-measurement element, w-reference, a-error, u-command, y-output, m-amplified command, r-feedback.
The digital controller is implemented with digital equipment, e.g. microprocessor based system, microcontroller, PLC. All these equipments are known as programmed logic devices as they execute sequentially a software module that drives the operations requested for control: data acquisition, computing according to control algorithm, signal generation towards the process. The sum of time intervals for these operations determines the sampling time t_s of the system, and consequently the characteristics of processes being controlled (fast or slow). According to Shannon's theorem, the sampling time must be at least half of the period of the signal processed or half of the smallest time constant of the process. Thus, with a greater sampling time, a slower process that can be controlled.
In this paper, the authors are presenting a solution for implementing a digital controller with a wired logic device, FPGA (Field Programmable Gate Array) that can perform all the operations much faster than a programmed logic device.

II. CONTROL SYSTEM LOOP

In order to experiment the solution, two digital control systems were built.

II.1. THE PROCESSES

The controlled processes, with u(t) as input and y(t) as output, are:

a) DC motor – (see Fig.2) which has an approximate transfer (amplifier A and measurement element M are included in the model of process):

$$H_{PA}(s) = \frac{k_A}{s} \qquad (1)$$

(k_A being a constant that incorporate electrical and mechanical constants in all blocks, the load is considered constant).

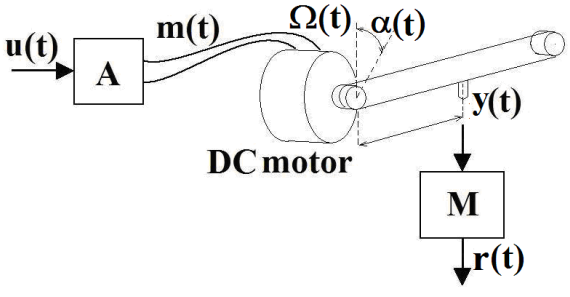

Fig.2. Physical model of an actuator with DC motor

[1] Facultatea de Automatică şi Calculatoare, Departamentul de Automatică şi Informatică Aplicată, Bd. V. Pârvan Nr. 2, 300223 Timişoara sorin.nanu@aut.upt.ro
[2] Facultatea de Electronică şi Telecomunicaţii, Departamentul de Electronică Aplicată, Bd. V. Pârvan Nr. 2, 300223 Timişoara ioan.lie@etc.upt.ro
[3] Facultatea de Mecanică, Departamentul Ingineria Materialelor si a Fabricatiei, Bd. M.Viteazu Nr. 1, 300222 Timişoara george.belgiu@eng.upt.ro
[4] Facultatea de Electrotehnică, Departamentul Inginerie Electrică, Bd. V. Pârvan Nr. 2, 300223 Timişoara, sorin.musuroi@et.upt.ro

where: A-amplifier, M- measurement module, m- amplified input, Ω(t)- angular speed, α(t)- angular position, y(t)- linear position, r(t)-measured position.
b) double integrator model – (see Fig.3) with transfer function:

$$H_{PB}(s)=\frac{1}{2s^2} \qquad (2)$$

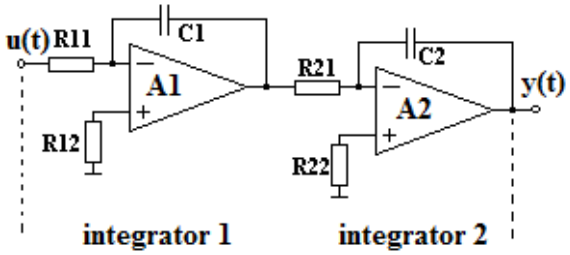

Fig.3 Double integrator model

Where: A1, A2 – operational amplifiers, R- resistor, C- capacitor. The size of R and C was selected in order to obtain a constant of $\frac{1}{2}$.

II.2. THE CONTROL LAW

In order to control the processes, two control systems, like in Fig.1, are designed. For case a), a proportional control law:

$$u(t_k)=k_{CA}\cdot a(t_k) \qquad (3).$$

In operational $H_{CA}(s)=\frac{u(s)}{a(s)}=k_{CA}$,will generate a first order closed loop control system with transfer function:

$$H_A(s)=\frac{y(s)}{w(s)}=\frac{k_{CA}\cdot k_{PA}}{s+k_{CA}\cdot k_{PA}} \qquad (4).$$

For the case b), some steps have to be followed in order to build a control system:

- Discretization of the process transfer function, with a time step of 0.3 sec, obtaining the discrete time transfer function

$$H_{PB}(z)=(1-z^{-1})\cdot Z\left\{\frac{1}{s}\cdot H_{PB}(s)\right\}=\frac{0.3^2}{4}\cdot\frac{z+1}{(z-1)^2}$$

- Selection of a first order structure for digital controller:

$$H_{CB}(z)=\frac{b_0+b_1 z^{-1}}{1-a_1 z^{-1}},$$

- Writing the discrete time transfer function for closed loop control system:

$$H(z)=\frac{y(z)}{w(z)}=\frac{H_{CB}(z)H_{PB}(z)}{1+H_{CB}(z)H_{PB}(z)}=$$

$$=\frac{l(z+1)(b_0z+b_1)}{z^3-(a_1+2-lb_0)z^2+(2a_1+1+lb_1+lb_0)z+(lb_1-a_1)}$$

where constant $l=\frac{0.3^2}{4}$.

- Obtaining of b_0, b_1, a_1 parameters of controller using the method of pole allocation. We consider a set of three poles for the three order system H(z), p_1 = 0.2, p_2 = 0.3 p_3 = 0.5, that endow the system with suitable behaviour (settling time, overshot). The controller parameters are obtained identifying the denominator of H(z) with a three order transfer polynomial having the selected solutions p_1, p_2, p_3. The result is b_0=26, b_1=-19.77, a_1=-0.415.

With this set of parameters, the time model of controller is:

$$u(t_k)=-0.415u(t_{k-1})+26a(t_k)-19.77a(t_{k-1}) \qquad (5).$$

II.3. THE IMPLEMENTATION OF CONTROLLER WITH FPGA

The functions the controller has to fulfill are:

- *Control*- process the information of reference and feedback according to control law,
- *Input*- acquire reference and feedback data from manual or remote source, respectively from process,
- *Output* – provide the result of processing towards process and issue all the input and output data to a display.

In order to reduce the computing time for control law, the digital device that was chosen to accomplish these functions, was an FPGA (Field Programmable Gate Array) circuit. The major advantage of FPGA is that all the functions are performed by hardware through a wired logic, the structure and consequently the behaviour being defined by a hardware description language (HDL).
The circuit chosen was XILINX XC3S500E. For these applications a SPARTAN-3E board with the mentioned FPGA circuit was used.
The applications are structured like in Fig.4.

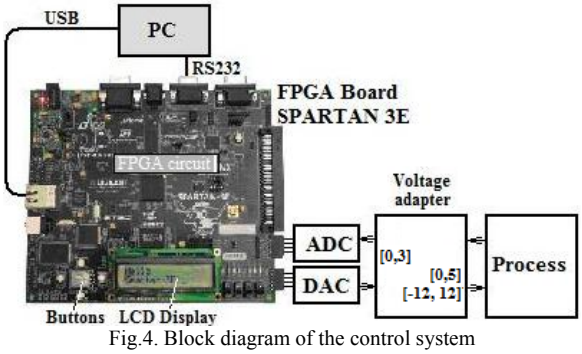

Fig.4. Block diagram of the control system

Apart from FPGA circuit, the used resources on the board are:- LCD Display, USB and RS232 serial interfaces, rotary buttons. The interface between SPARTAN board and Process consists of *ADC, DAC* and *Voltage adapter*. ADC (Analog to Digital Converter) transforms the analog signal from process into a digital signal while DAC (Digital to Analog Converter) transforms the digital signal from controller into an analog signal for process. As processes are working with signal in domain [0,5] V respectively [-12, 12]V, the Voltage adapter module

transforms these domains in [0,3]V domain the converters are working with.
The hardware inside FPGA circuit is programmed in a manner to form the structure as in Fig.5.

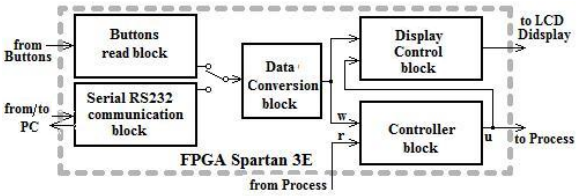

Fig.5.The structure of FPGA circuit

The *controller* function is accomplished by *Controller block*. The inputs are the reference and feedback from process, and the output is the command towards the process and to display. The *input* function is realized by

- *Button read block*, that reads reference data from buttons on board,
- *Serial RS232 communication block* that reads reference data from PC,
- *Data conversion block* that transforms data format from ASCII(from PC) into number towards the Controller block.

The *output* function is accomplished by

- *Display Control block*, that generates all the signals for LCD display,
- *Serial RS232 communication block* that sends all internal data to PC, in order to display it there,
- *Data conversion block* that transforms data format from number to ASCII in order to send appropriate signal to Display Control block.

The Controller function consists of operations:

- Taking over the reference data
- Taking over the feedback data
- Performing the computation according to control law
- Generating the command data

The signals of control block are detailed in Fig.6.
The inputs of the block are:

- master clock (Clk), the system clock of 50 MHz,
- Input of reference, coming from rotary buttons (manual reference) or from PC.
- Input of feedback from process (through ADC).

The serial data comes on SPI bus, 12 bit, natural binary coded.
The outputs of the block are:

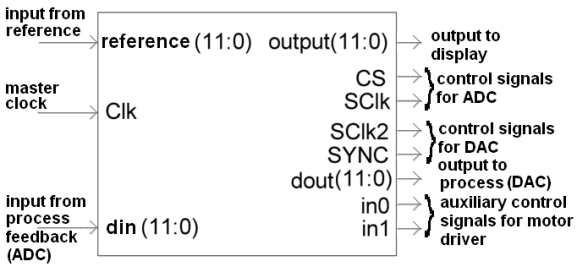

Fig.6. Signals of Controller block

- Output – serial data representing the input transferred to display,
- Dout – serial command signal towards process (through DAC),
- CS, SClk – selection and clock signals for DAC,
- SYNC, SClk2 – selection and clock signals for ADC,
- In0, in1 – auxiliary control signals for motor driver.

The controller block can be considered as Finite State Machine (FSM) with 4 states as in Fig.7.

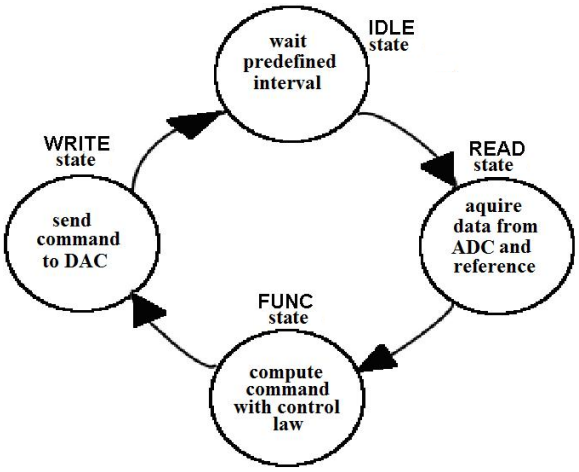

Fig.7.The Finite State Machine of Controller block

The IDLE state is the initial state of FSM. Here, a defined number of clocks are waited, in order to elapse the sampling period. The next state is READ, where the data from reference and feedback is acquired through serial line. After completing the operations, the state FUNC is activated, where the command data is computed. First, the difference $a(t_k) = w(t_k)-r(t_k)$ (see Fig.1) and then $u(t_k)$ is computed using (3) or (5) in those two cases. In (5), the previous samples from feedback are used. Within WRITE state, data is sent towards both the process and the display. The computed value $u(t_k)$ is sent towards the process through DAC on serial line, and $u(t_k)$ and $a(t_k)$ is sent to LCD display.
Even the structures of the controllers are rather complex, with the aid of optimization techniques, the reduced number of resources of FPGA were used as seen in Table 1.

Table 1. Resources used in FPGA circuit

Resources	used/available	%
IOB	24 / 232	10%
MUX	3/24	12%
MULT18x18	3/20	15%
SLICE	770/4656	16%
SLICEM	2/2328	1%

II.4. SOFTWARE

Two types of software modules were used in programming the application:

- Software for FPGA

- Software for user interface on PC

The hardware of FPGA was programmed in VHDL, using the XILINX ISE 10.1 (Integrated Software Environment), a specialized environment for FPGA. ISE has specialized modules to fulfill the achievement of FPGA project:
-defining of project (modules ECS, editor HDL, StateCAD, CORE Generator, etc),
-synthesis (module XST)
-simulation (module HDL Bencher Testbench Generator),
-implementation (editor PACE, module Floorplanner, etc),
-device configuration (module iMPACT).
The software that creates the interface on a PC with the user was written in LabVIEW.

III. THE EXPERIMENT

The test stand that contains the process (DC motor), FPGA board, PC and oscilloscope is presented in Fig.8.

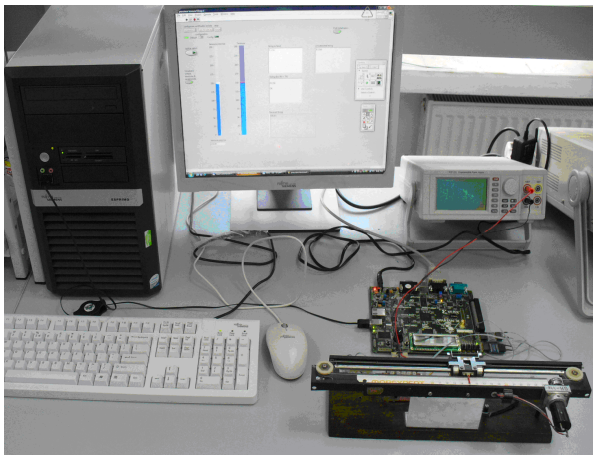

Fig. 8 Test stand

The experiment consists in building the test stands for both control systems, setting the proper configuration according to theory, changing the value of reference as a step and recording the output y(t) and control value u(t).
The user interface (Fig.9) functions are:

- to set the reference value (blue left bar in Fig.9),
- to display the actual value recorded by FPGA (blue right bar),
- to set the serial communication protocol,
- to display the data sent and received from FPGA in order to allow the check of correctness of data.

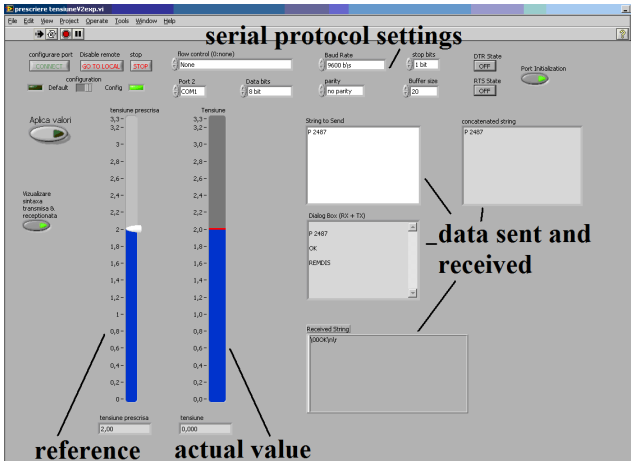

Fig.9. User interface

Data is sent and received between PC and FPGA on the serial 232 line.

III.1. THE CONTROL SYSTEM OF DC MOTOR

The modules are presented in Fig.10, the ongoing experiment was caught in Fig.11, where can be observed the coincidence of three values, reference w(t), output y(t) and recorded output by FPGA.

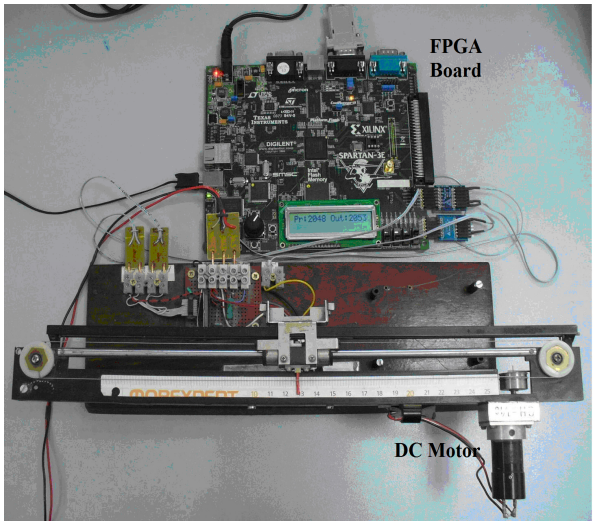

Fig.10. The modules of DC motor control

The first set of measurements was made in the following conditions:

a) change of reference w(t)
   i. k=4
   ii. k=30 (Fig.12)
   iii. k=100

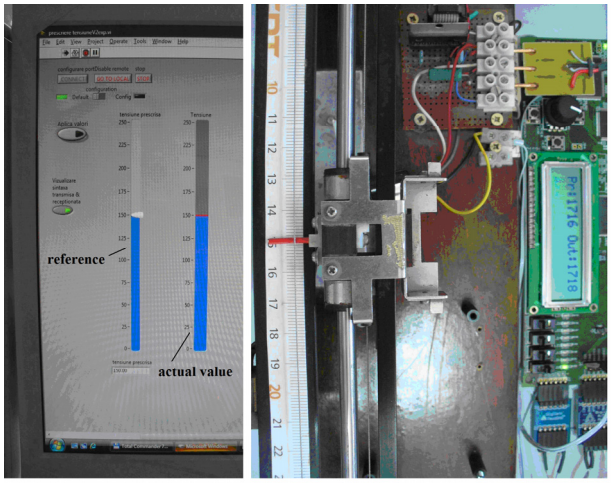

Fig.11. Application running (see the coincidence between reference, position recorded by FPGA on user interface and actual position of the motor)

In all figures, the output is the above signal, and the control is the beneath signal.

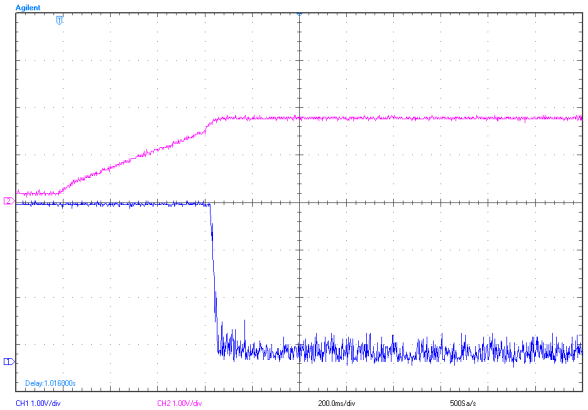

Fig.12. k=30, change in reference

b) application of perturbation (it is realized by dragging the slider with hand and then releasing)
   i. k=4 (Fig.13)
   ii. k=30 (Fig.14)
   iii. k=100 (Fig.15)

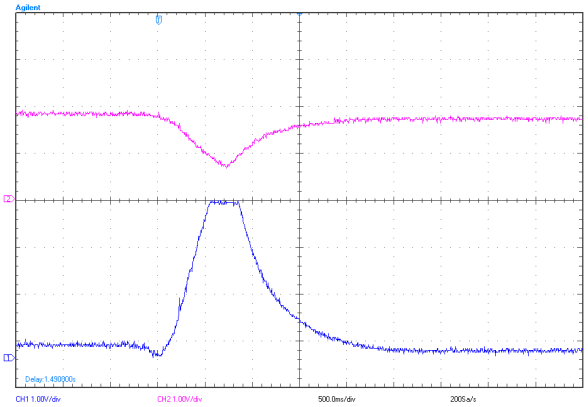

Fig.13. k=4, application of perturbation

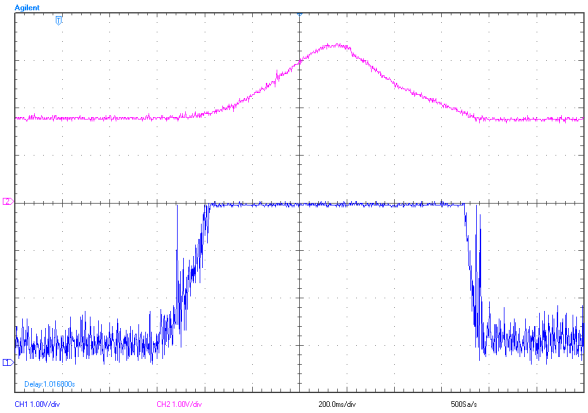

Fig.14. k=30, application of perturbation

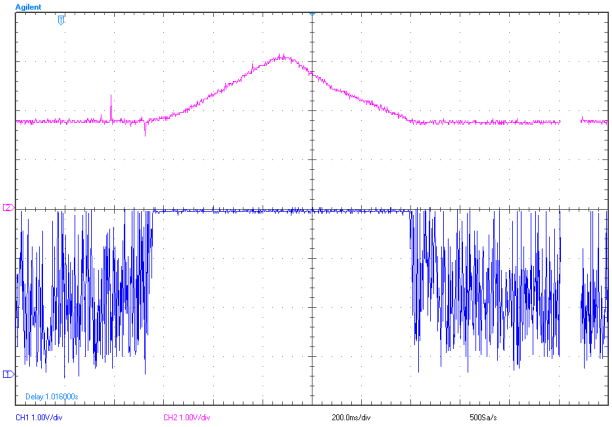

Fig.15. k=100, application of perturbation

It can be seen in the recordings the qualitative behaviour of DC motor control system. The main differences between the outputs and control values are that the greater the constant k of controller, the faster output, and the greater oscillations.

III.2. CONTROL OF DOUBLE INTEGRATOR MODULE

As the first step, the control system was simulated in Simulink, with the model in Fig.16. The results of simulation are presented in Fig.17.

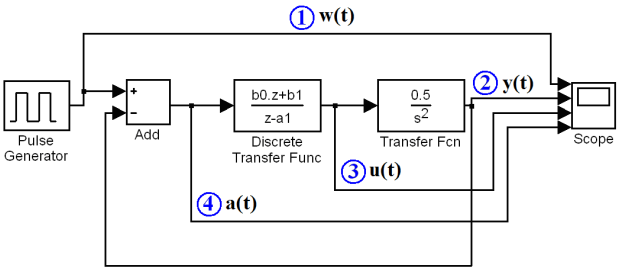

Fig.16. Simulink model of double integrator control system

The double integrator module controlled by FPGA has the response in Fig.18.

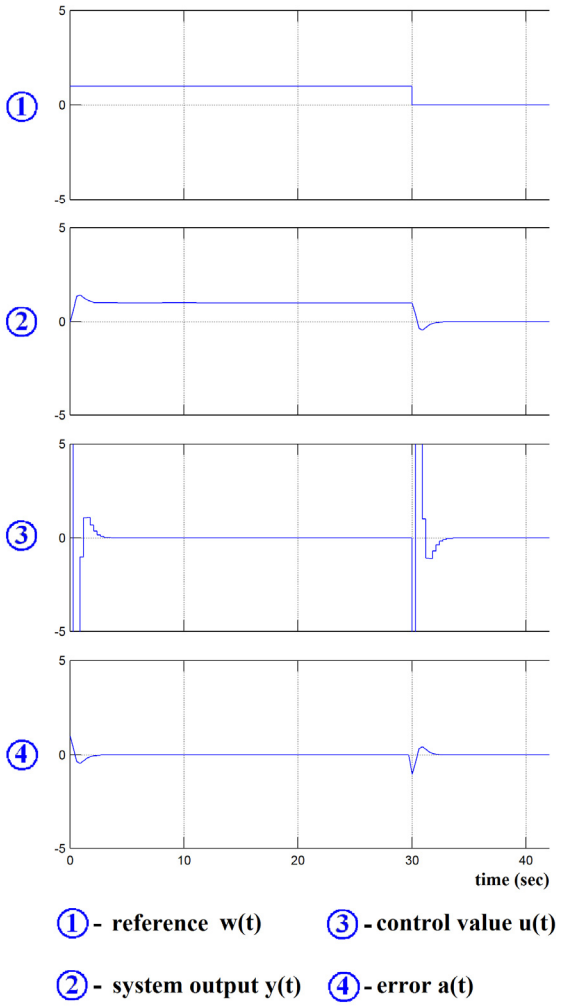

Fig.17. Results of simulation

When running the experiment, the reference was changed as a step, the output and control value had the behaviour as in Fig.18. It can be seen that there is a difference between simulated and real results, due to perturbations that occur on the signal lines.

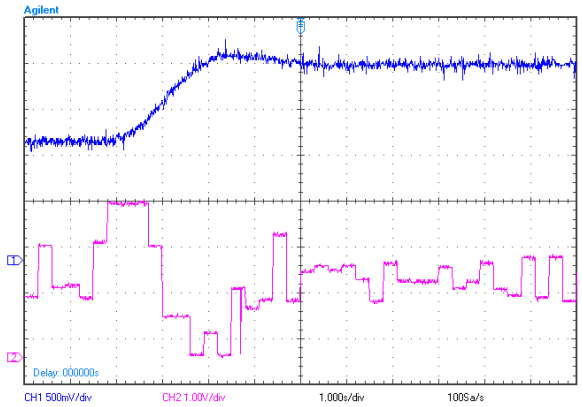

Fig.18. The output and control for double integrator module

III.3. PERFORMANCE CONSIDERATIONS

The great achievement of FPGA circuits is the speed of computing. Within these circuits, a multiplication and an addition is performed during a cycle that means a clock period. Because of the complex structure of the circuits, all other operations, like transfer, memory, multiplication, etc., do not consume computing time.

If a time computing is made for the application presented, we obtain amazing results, as:

For system oscillator 50MHz, 20ns cycle results. Computing time for DC motor control system that has proportional controller results 20ns. A great time consuming are the AD and DA converters that spend 16 µs each, sum is 32 µs. So overall time is 32,02 µs. If faster converters are used, the time will be reduced. For the second control system, the situation is similar, computing is 3x20ns=60ns. But it was calculated with a sampling time of 300ms, so, waiting states were introduced up to desired time. The 300ms sampling time can be seen on diagram in Fig.18.

With these considerations, it can be concluded that such a device implemented with an FPGA is suitable for digital control, especially where control law requires many calculations.

IV. CONCLUSIONS

In the paper were presented two control systems, designed and simulated with MATLAB Simulink. The real processes were controlled with FPGA based digital controller, and the behaviour was compared, with very good results with simulated systems. The great achievement of this solution is the drastic reduction of computing time of FPGA due to wired logic. In the presented two examples, 20ns and 60ns were the computing time for first order controllers.

REFERENCES

[1] Alan V. Oppenheim, Ronald W. Schafer, John R. Buck, Discrete-Time Signal Processing, Prentice-Hall, New Jersey, 1999

[2] Janocha, H., Pesotski, D., e.a., FPGA based Compensator of Hysteretic Actuator Nonlinearities for Highly Dynamic Applications, IEEE Transactions on Mechatronics, vol.13, no.1, feb.2008, pp.112-116.

[3] Dragomir, T.L., Elemente de Teoria Sistemelor,, vol.1, Ed. Politehnica, Timisoara, 2004.

[4] System Generator for DSP Getting Started Guide, Xilinx, 2008

[5] System Generator for DSP User Guide, Xilinx, 2008

[6] Digilent Pmod AD1 Analog To Digital Module Converter Board Reference, Manual, Digilent, 2005

[7] Digilent Pmod DA2 Digital To Analog Module Converter Board Reference Manual, Digilent, 2005

[8] ST7066U Dot Matrix LCD Controller/Driver, Sitronix, 2006