OPTIMIZED STRUCTURE OF DIGITAL PID CONTROLLER BASED ON FPGA [+]

Raaed F. Hassan [*]

Abstract:

The work presented in this paper is concerned with design an optimized structure of a digital Proportional – Integral – Derivative (PID) controller to be implemented using Field Programmable Gate Array (FPGA).

The analog PID controller was firstly considered to control a 3rd order plant, the optimum parameters of this controller is estimated using signal constraint block provided with Simulink. The discrete time PID controller is developed from analog counterpart using trapezoidal integration to approximate integral action and a backward difference for derivative action. From the time response point of view, simulation result shows that the discrete time controller has the same characteristics of the analog controller.

The digital prototype of the discrete time PID controller was performed using Very high speed integrated circuit Hardware Description Language (VHDL) in order to implement based on FPGA. Results show the improvement of control system from the processing time and design flexibility points of view.

المستخلص:

يتركز العمل المقدم في هذا البحث على تصميم تركيب محسن للمسيطر الثلاثي الرقمي لغرض بناءه باستخدام رقاقات مجال مصفوفة البوابات المبرمجة (FPGA).

في البداية أخذ بنظر الاعتبار المسيطر الثلاثي التناظري للسيطرة على منظومة من الدرجة الثالثة ، وتم احتساب عناصر المسيطر المثلى باستخدام صندوق تحديد الاشارة الموجود في السميلنك. تم انشاء المسيطر المتقطع زمنيا باستخدام المسيطر التناظري ونظريات التحليلات العددية لتقريب التكامل والمشتقة الموجودة في المسيطر التناظري. نتائج المحاكاة أظهرت بان المسيطر المتقطع زمنيا يمتلك نفس المواصفات للمسيطر التناظري .

تم انشاء النموذج الرقمي للمسيطر المتقطع زمنيا بواسطة لغة تصميم الكيانات المادية (VHDL) وذلك لغرض بناؤه باستخدام رقاقات مجال مصفوفة البوابات المبرمجة. أظهرت النتائج تحسن منظومة السيطرة من ناحية زمن المعالجة ومرونة التصميم.

Introduction:

[+] **Received on 15/11/2006 , Accepted on 22/5/2008**

[*] **Lecturer / College of Electrical and Electronic Techniques**

Proportional – Integral – Derivative (PID) controller has been used successfully for regulating processes in industry for more than 60 years. The number of applications requiring high precision control is increasing; up to 95% of these applications are solved through the PID controller [1, 2]. The controllers for the above applications are usually implemented as discrete controllers, and require high precision and high processing speed, over 50 Msa/s [3].

Implementation of any complex digital controller must be done by means of some form of computer. Typical microcontrollers, while cheap, don't normally provide enough processing power to effectively perform all but the simplest calculations in real – time. Digital signal processors, on the other hand, are designed to implement complex algorithm quickly. The major drawback of DSP's, however, is the cost [4]. Field Programmable Gate Array (FPGA) is a median between these two extremes of performance and cost. The aim of this paper is to design a new executing method for a complex PID algorithm to be implemented using FPGA.

A General PID Controller Algorithm:

In the study of continuous – time control systems it was found that if proportional control is employed, a steady – state error was necessary in order to have a steady – state output. It was also found that if an integrator replaces the proportional controller, the steady – state error can be made zero for steady output. Often the introduction of the integrator into loop will create instability or, at best, poor dynamic character which manifests itself as overshoot and excessive ringing in the output. Several compromises are possible, one of which uses an actuating signal which has one component proportional to the error signal and the other which is proportional to the integral of the error. This combination also reduces the steady – state error to zero and yields acceptable dynamics. If further improvement in dynamics is required, a differentiator which is sensitive to error rate can also be included in parallel with two other devices [5]. These improvements yield which is known a PID controller with the following mathematical model:

$$G_c(s) = Kp + \frac{Ki}{s} + \frac{Kd\ s}{0.1s + 1} \qquad (1)$$

where:
Kp: is the proportional gain;
Ki: is the integral gain;
Kd: is the derivative gain.

This continuous – time control scheme is shown in Figure (1) with 3rd order continuous – time plant.

The first task is to find the optimum values of Kp, Ki and Kd for a given plant. This task is performed using signal constraint block provided with Simulink as shown in Figure (2).

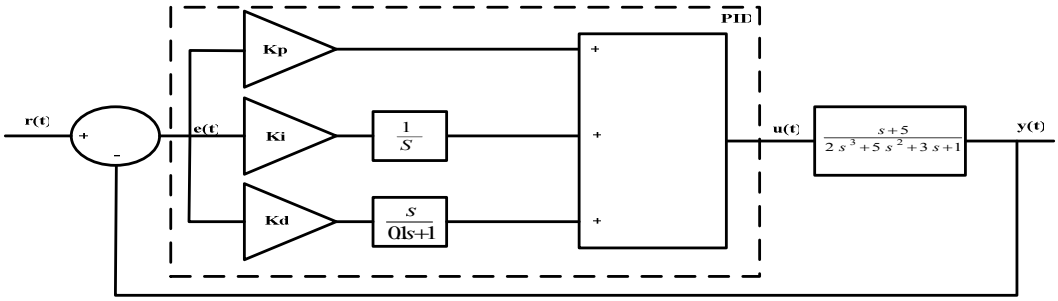

Fig. (1): Continuous – time control system

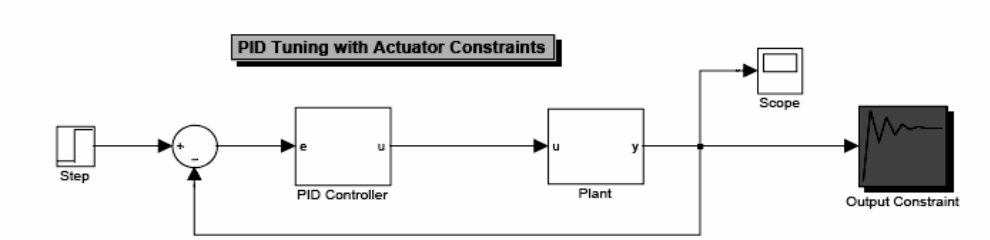

Fig. (2): Simulink model for closed loop control system.

Simulation result is shown in Figure (3) and the values of the three terms will be:

Kp	Ki	Kd
0.3174	0.1171	0.3625

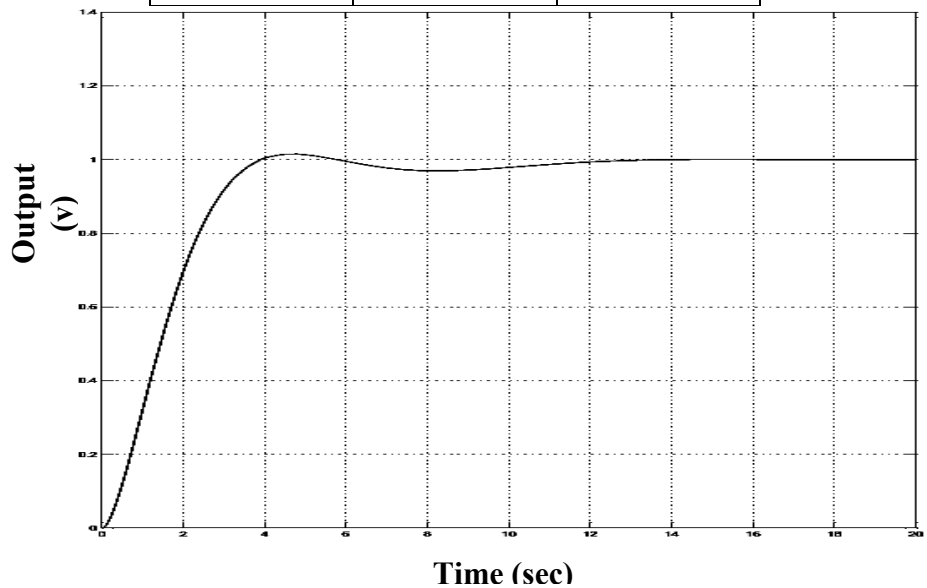

Fig. (3): Time response of control system to unit step input.

Design of Discrete – Time PID controller:

Following is a block diagram of the system under consideration:

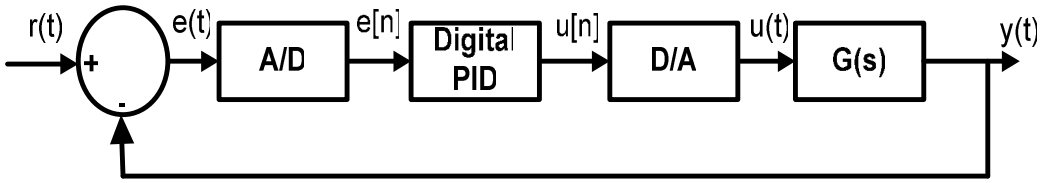

Fig. (4): Digital PID Controller in Control system

In figure (4), r(t) is the continuous input signal (or set point), y(t) is the continuous output signal, e(t) is the continuous error between r(t) and y(t), e[n] is the digital error signal, u[n] is the digital control signal, u(t) is the continuous control signal, and G(s) is the plant under consideration.

A digital PID controller has been implemented according to the discrete – time counterpart who has the following form [6]:

$$D(Z) = Kp + Ki\,\frac{T}{2}\,\frac{Z+1}{Z-1} + Kd\,\frac{Z-1}{TZ}$$

$$= \frac{ao + a1Z^{-1} + a2Z^{-2}}{1 - Z^{-1}} \qquad (2)$$

with

$$ao = Kp + \frac{KiT}{2} + \frac{Kd}{T}$$

$$a1 = -Kp + \frac{KiT}{2} - \frac{2Kd}{T}$$

$$a2 = \frac{Kd}{T}$$

where T is the sampling period .

Equation (2) approximates the integral action in equation (1) using trapezoidal discretization method and differentiation action using backward difference.

For Kp = 0.3174, Ki = 0.1171, Kd = 0.3625 and T= 0.05 msec (according to minimum conversion time of ADC/DAC). The discrete – time PID controller transfer function according to equation 2 will be:

$$D(Z) = 7250\ (\frac{1-2Z^{-1}+Z^{-2}}{1-Z^{-1}}) \qquad (3)$$

Simulation result of a designed discrete – time PID controller with a 3rd order plant is shown in Figure (5).

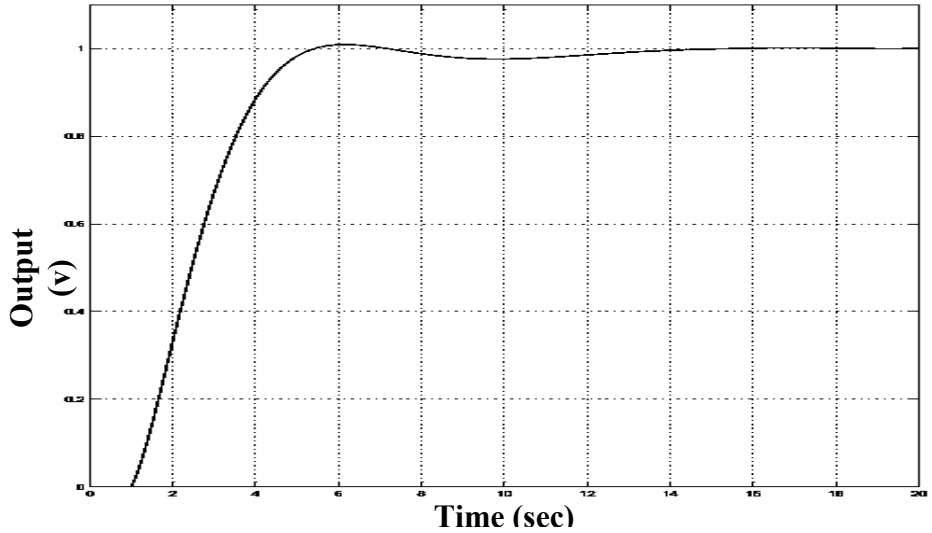

Fig. (5): Time response of control system with digital PID

Digital PID based on FPGA:

The development of the digital PID based on FPGA is achieved by decomposing the discrete transfer function of equation (3) as illustrated below:

$$D(Z) = 7250 * D_n(Z) * D_d(Z) \quad (4)$$

where:

$$D_n(Z) = \frac{S(Z)}{e(Z)}$$

$$\frac{S(Z)}{e(Z)} = 1 - 2Z^{-1} + Z^{-2} \quad (5)$$

The difference equation resulting from discrete transfer function in equation (5) is:

$$S[k] = e[k] - 2\,e[k-1] + e[k-2] \quad (6)$$

and :-

$$D_d(Z) = \frac{u(Z)}{S(Z)} = \frac{1}{1 - Z^{-1}} \quad (7)$$

$$u[k] = u[k-1] + S[k] \quad (8)$$

The resulting representation of digital PID controller core is shown in figure (6) which is described under VHDL.

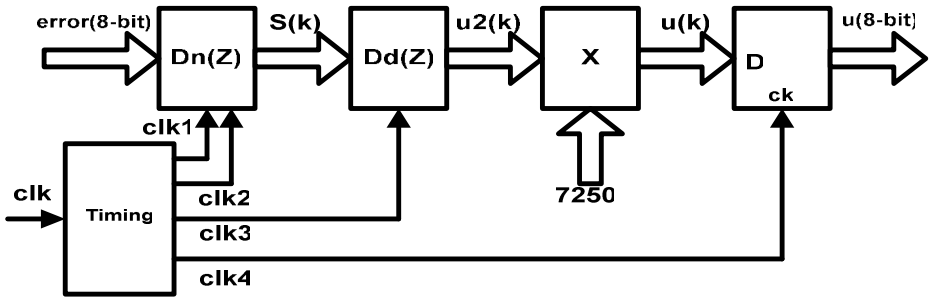

Fig. (6): Representation of digital PID block diagram

The core representation of equation (6) is shown in figure (7), which is described under VHDL with project Navigator ISE4.1.

Multiplication by the constant (7250) was performed by shift and add with VHDL statement shown below

x8<= ("0000000"&x9&'0')+(x9&"00000000");

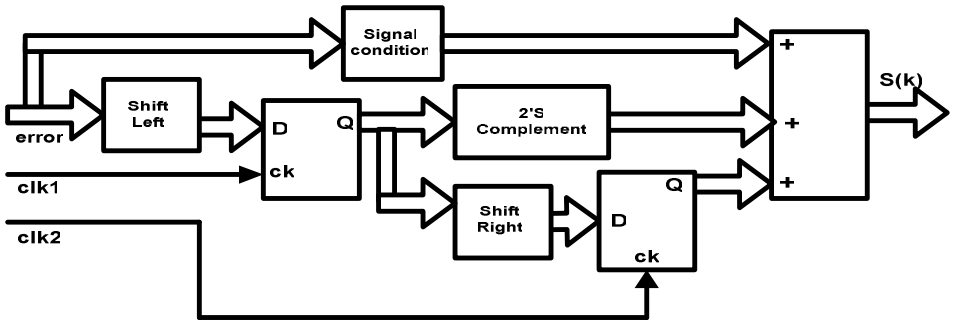

Fig. (7): Details representation of Dn(Z) block

From figure (7) the following remarks are arises which represent an optimized structure of a digital PID controller:

- Multiplication by 2 is performed by shift left operation which can be represented by the following statement:

  x1<= e&'0';

- Z^{-1} is performed by D – type Flip Flop which triggered every master clock cycle. The D – type Flip Flop was designed with behavioral mode by the following statements:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity dff7 is
   Port ( din : in std_logic_vector(7 downto 0);
          clk : in std_logic;
          dout : out std_logic_vector(7 downto 0));
end dff7;
architecture Behavioral of dff7 is
begin
```

```
process (clk)
begin
  if clk'event and clk='1' then  --CLK rising edge
    dout <= din;
  end if;
end process;
end Behavioral;
```

- Z^{-2} is performed by another D – type Flip Flop which triggered every two master clock cycle.
- Minus sign is performed by 2's complement operation which can be represented by the following statement:

```
x31<=not x2+"000000001";
```

The core representation of equation (7) is shown in figure (8), which is described under VHDL with project Navigator ISE4.1.

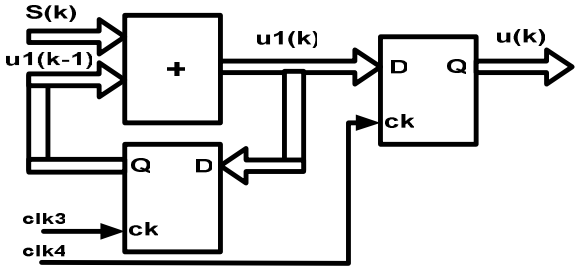

Fig. (8): Details representation of Dd(Z) block

The design was implemented using VertixE (xcv 300e -8 fg 256) FPGA device with the following implementation report:

Device utilization summary:

Number of External GCLKIOBs	1 out of 4	25%
Number of External IOBs	16 out of 176	9%
Number of LOCed External IOBs	0 out of 16	0%
Number of SLICEs	33 out of 3072	1%
Number of GCLKs	1 out of 4	25%

Simulation Results:

Simulation process was performed using ModelsimXE 5.5b software package which is compatible with the project Navigator ISE4.1. In order to verify the correctness of data flow, simulation of each unit in the digital PID controller architecture was performed by implementing a test bench waveform file(.tbw), in which the input data and the timing constrains are specified in a way stated in [7,8].

In figure (9) simulation result of the functional behavior of the digital PID controller is shown. It is specified in figure (6) which produces an output after (1.5 μsec) from receiving error signal for a maximum master clock frequency (100 MHz). This time of producing output can't be applicable in real time because of the limitations of ADC/DAC conversion time. Therefore, the master clock frequency was reduced to 0.3 MHz which forces the controller to produce an output after 50 μsec from receiving error signal as shown in figure (10).

A comparison between error signal produced by the control system with conventional PID controller and that produced with digital PID controller is shown in figure (11). From figure (11) it can be notice that the difference between two signals

in transient stage is due to the delay produced by the calculation time of digital PID controller.

Conclusion:

A digital PID controller was successfully designed and tested with 3rd order plant based on FPGA. Design of digital controller based on FPGA using VHDL provides a flexible method for functional design of different sub blocks without restricted by the software library; beside it provides full information of the behavior of the system in real time applications. Although the design of digital PID controller based on FPGA didn't utilize fully the FPGA's facilities. This is due to the fact that the FPGA is a very fast calculation device but it is bounded with the limitation of ADC/DAC conversion time.

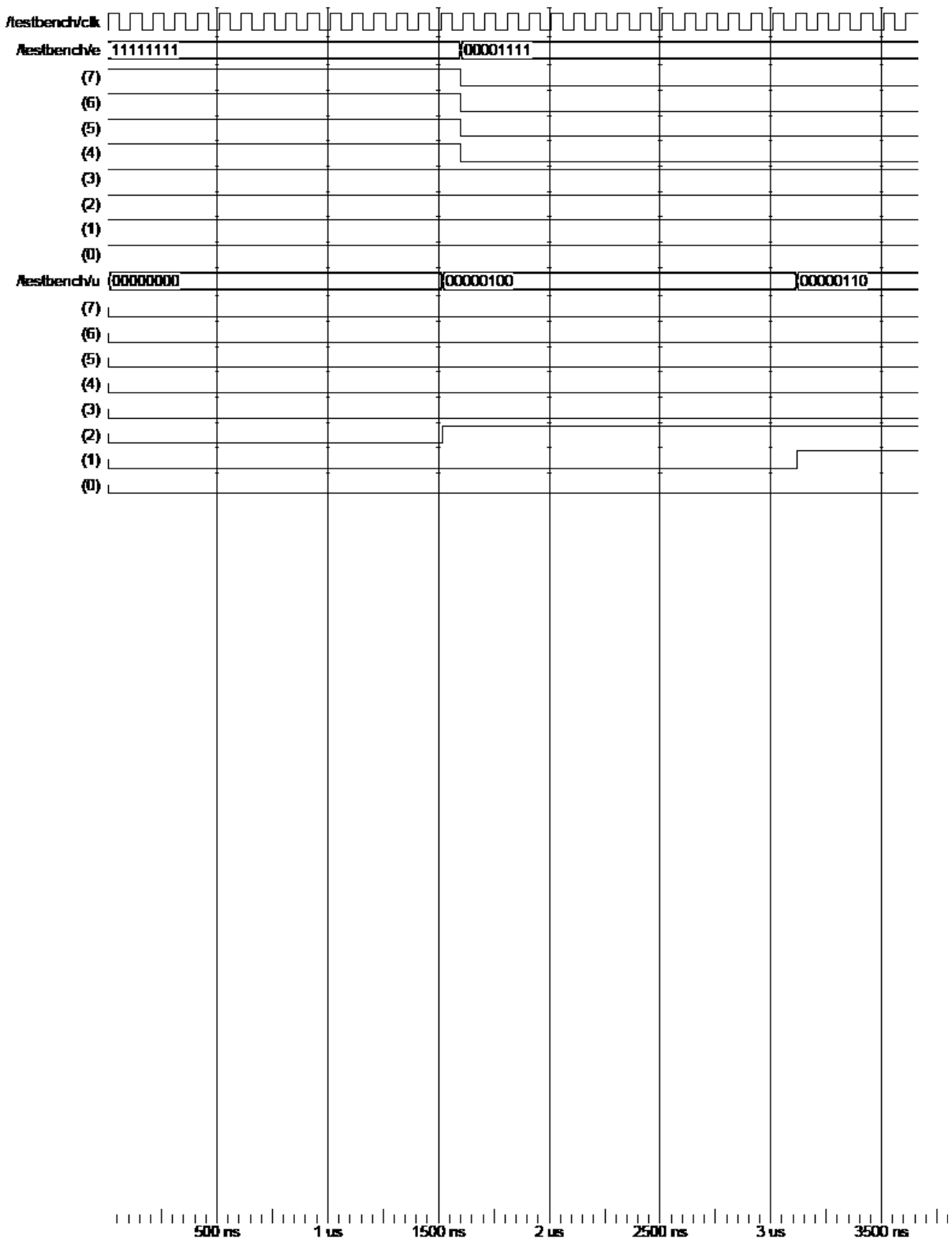

Fig. (9): Simulation results of FPGA based digital PID controller (100 MHz master clock frequency).

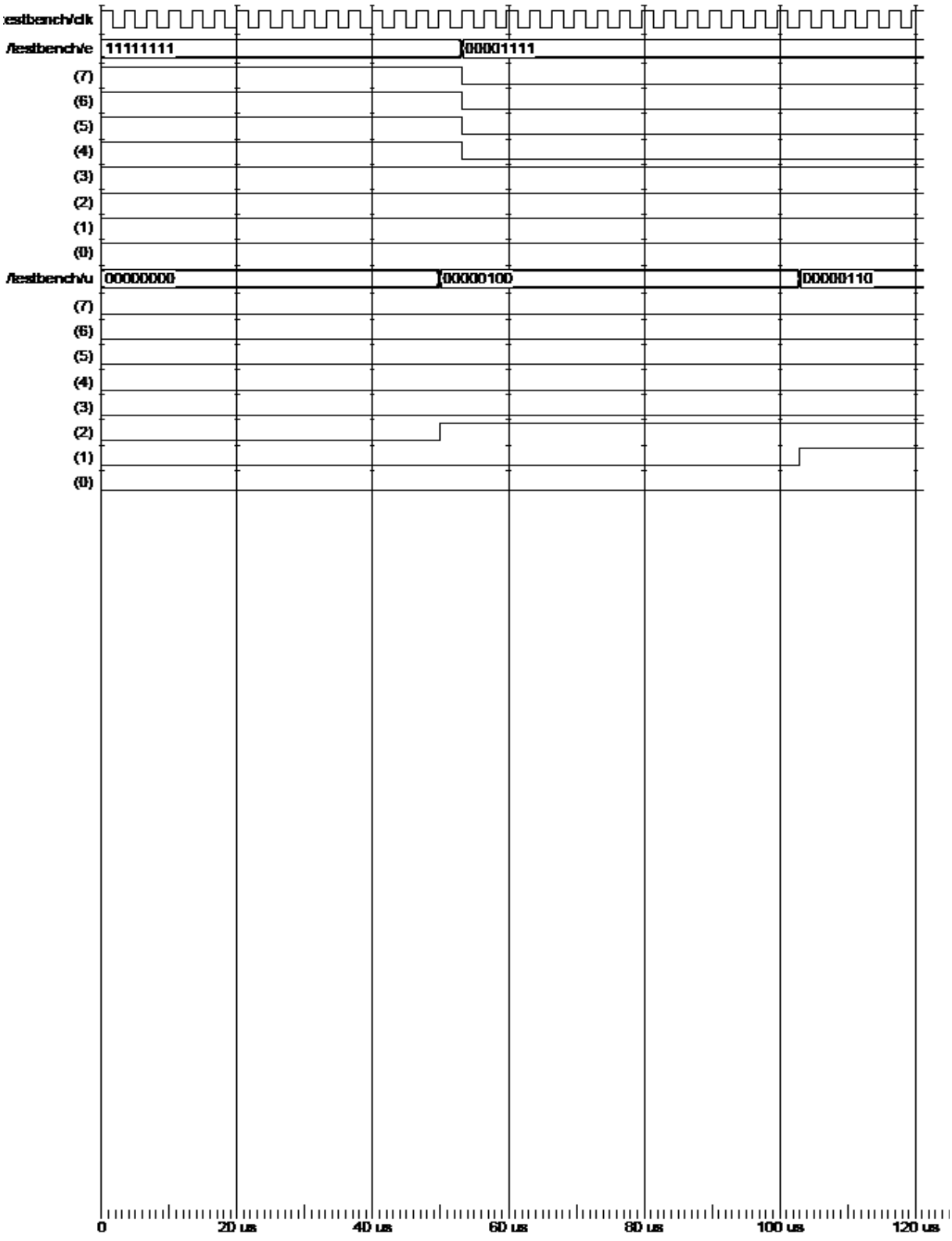

Fig. (10): Simulation results of FPGA based digital PID controller (0.3 MHz master clock frequency).

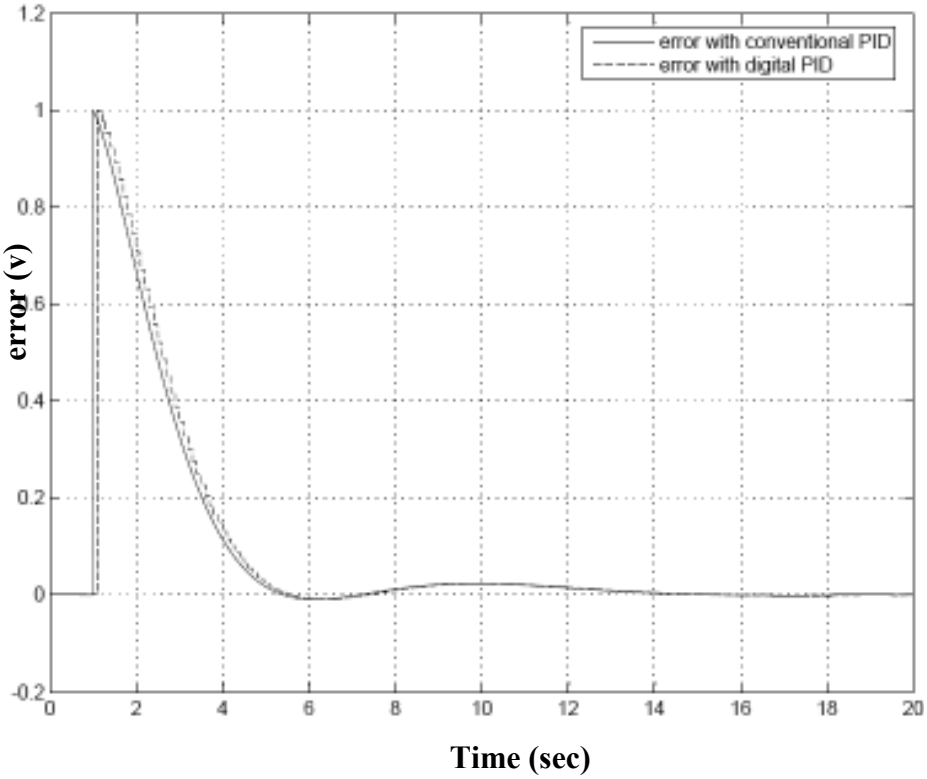

Fig. (11): Simulation results of error signals

References:

1. R. Kurfess, H. Jenkins, " Ultra – High precision control ", in the control Handbook (CRC Press) 1996.

2. K.J. Astrom and T. Hagglund, "PID control", in the control Handbook (CRC Press) 1996.

3. L. Parrilla, A. Garcia and A.Luoris, "Implementation of High Performance PID controllers using RNS and Field – Programmable Devices ", IFAC workshop on Digital control, 2000, 628 – 631.

4. P. Leisher, C. Meyers, "FPGA Implementation of a PID controller with DC motor Application", Bardley university, senior project 2002.

5. Richard C. Dorf and Robert H. Bishop, "Modern Control Systems ", Addison Wesley Longman, Inc. 1998.

6. C. L. Phillips and H. T. Nagle, "Digital Control Analysis and Design ", Prentice Hall, Inc., 1995.

7. XESS Corp.," Introduction to WebPACK 4.1 for FPGAs"2001.

8. Xilinx Development System, "ISE 4 Tutorial".Xess Corporation, October 2001 (www.xess.com).