FPGA Realization of PID Controller Based on BP Neural Network

Shanyong Xu*, Yourui Huang, Liguo Qu, Chaoli Tang
Institute of Electrical and Information Engineering Anhui University of Science and Technology
Anhui province, China
*Corresponding author, e-mail: xsyong326@163.com

Abstract

Through coordinately work of input layer module, hidden layer module, output layer module and back propagation calculation module to complete the whole system of the closed-loop calculation, this is a process of FPGA realization of PID controller based on BP neural network. Operation of the entire system with FPGA internal clock coordination is unified coordination, just provide clock and reset signals can automatically tune parameters of PID controller.

Keywords*: BP neural network, PID controller, FPGA realization, parameter tuning*

1. Introduction

FPGA means Field Programmable Gate Array, the boundaries between software and hardware is redistricted with FPGA rise, however, the requirements for large capacity, low voltage and low power consumption of FPGA products is becoming high with the improvement of people's living standards. This means the difficulty of the design of the FPGA is increasing. At present, engineers are keen to combine intelligence with FPGA and all kinds of intelligent products came into being [1]. DSP Builder is a development tool of digital signal processing (DSP), it provides an interface between the QuartusII and the Matlab/Simulink. When designing a DSP system in the programmable logic device (PLD), development tools which support advanced algorithms and hardware description language (HDL) are required [2-3]. Matlab and Simulink have the ability of algorithm development, simulation and confirmation, DSP Builder combines these tools with the development tools of Altera then provides a whole DSP development platform. This article firstly gives the hardware graph of the whole system then detailedly describes the design of modules. Lastly, in the end of the article there gives a simulation example, and the results of simulation prove that the design strategy is effective.

2. Hardware Structure

Figure 1 is a hardware structure diagram of PID controller based on BP neural network, initial module, input layer computation module, hidden layer computation module, output layer computation module, hidden layer weight value adjustment module, input layer weights adjustment module, control module and memory module together consist of parameter self-setting controller in FPGA [4-6]. Because the entire closed loop control is implemented in FPGA, so put output yout and error in input layer calculation module (not directly displayed in Figure 1) in design. In which memory module 1 to save the hidden layer input values, memory module 2 to save the hidden layer output values, memory module 3 to save the output data, memory module 4 to save the hidden layer weights data, memory module 5 to save the input layer weights data.

After a system has been power-on, firstly system gives out reset signal, when system received a reset signal and detected reset is ending, the control module will control all modules to run, and reset all control signals then obtain a group of PID control parameters through the calculation of the input layer [7-9], hidden layer and output layer. Following continued to adjust the weights of BP network, firstly adjusts the hidden layer weight then adjust the output layer

Received April 27, 2013; Revised July 11, 2013; Accepted July 22, 2013

weigh, lastly return to input layer and begin the next cycle computation. The number of cycles is set by the system, in this article the number of cycles is 2000.

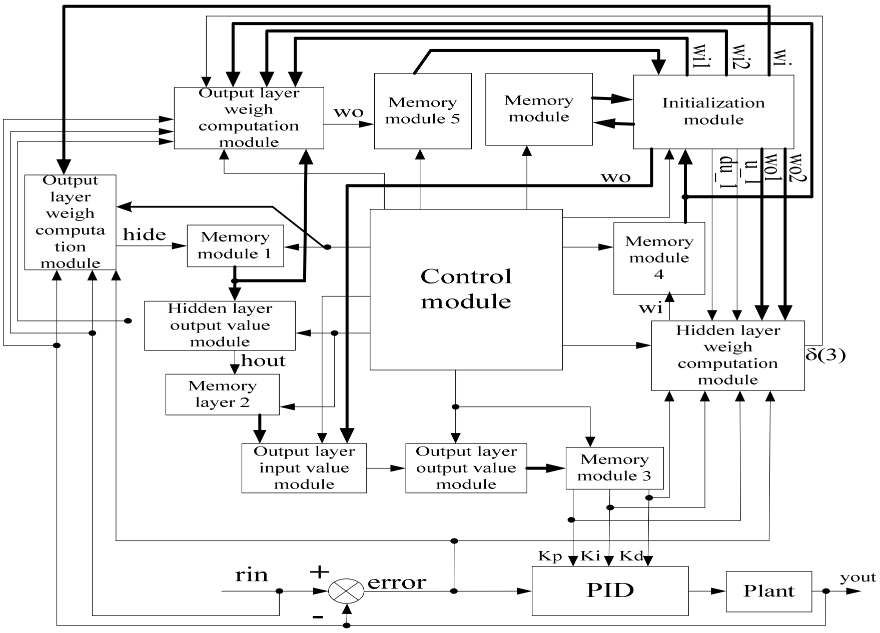

Output layer weigh computation module
wo
Memory module 5
Memory module
Initialization module
wi1
wi2
wi
wo
du_1
u_1
wo1
wo2
Output layer weigh computa tion module
hide
Memory module 1
Control module
Memory module 4
wi
Hidden layer output value module
hout
Memory layer 2
Hidden layer weigh computation module
δ(3)
Output layer input value module
Output layer output value module
Memory module 3
Kp
Ki
Kd
rin
+
-
error
PID
Plant
yout

Figure 1. Hardware structure diagram of PID controller based on BP neural network

In the above diagram, control module is the core of the whole system, controls the running of all other modules and provides stimulating clock and the loop control module for all other modules [10-11]. Calculations of the overall system are complex, it is unpractical if use only a module. So this section design many modules, in order to make full use of the advantage of DSP Builder, in this paper most design is finished by DSP Builder.

3. Design of Function Module

Because of the space of an article is limited, so there only detailed introduction the design of the input layer module and hidden layer module:

3.1. Design and Implementation of the Input Layer Module

The four data of the input layer, in addition to r(k) and constant '1' is known, y(k) and e(k) is variable that need to calculate. In the input layer module, It also calculated x(1), x(2) and x(3) at the same time. Which, $x(1) = e(k) - e(k-1)$; $x(2) = e(k)$; $x(3) = e(k) - 2e(k-1) + e(k-2)$.

The design principle diagram of the module as below [12]:

In order to ensure accuracy, data are adopted fixed-point arithmetic, combined with the data characteristics of the network design, Module put to use 24-bit binary, four-bit represented the integer part and 20-bit represented the fractional part. Input module contains multiplication module, division module and addition module. For division module, in order to improve the accuracy of the data, it firstly left shift 30-bit the input dividend. Equivalently, it firstly expanded 30 times of the dividend, then, it right shift 30-bit the output.

3.2. Design and Implementation of the Hidden Layer Module

The calculation of the hidden layer module have two parts which include the calculation of hidden input value calculation and the calculation of hidden output value [13].

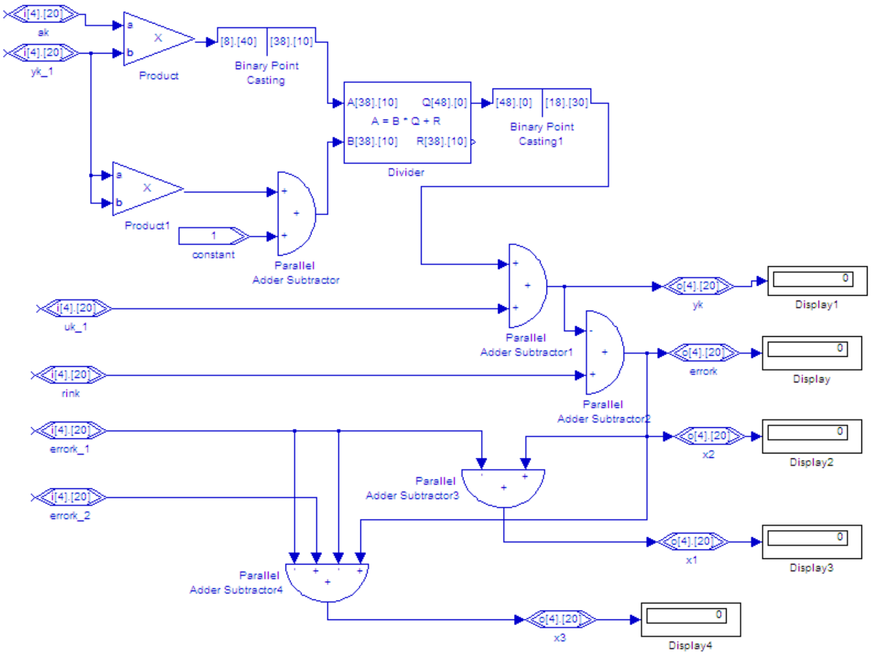


Figure 2. Design of input values module

3.2.1. Input Values Calculation Module

Calculation of hidden layer input values is the process that four inputs of the input layer multiplied the corresponding weight to obtain five input values of hidden layer input. This calculation of five input values is similar, so in order to enhance the ratio of resource utilization, it is necessary to calculate five hidden layer input values by using the circulation method. Specific design modules as shown in the following figure:

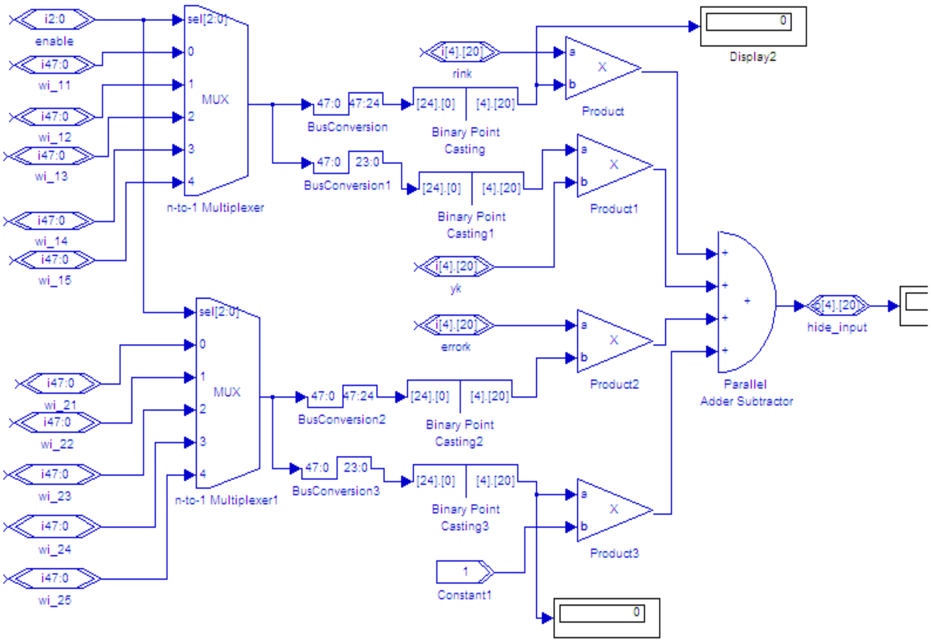


Figure 3. Design of hidden layer input values module

In module design, there is a channel selection module, it provides input terminal for five times circulation of hidden layer input values. For weight storage, it can be implemented by using RAM storage module, in the process of error back-propagation, there need to store the data. Due to the number of input layer, hidden layer and output layers is not much, so to save data by defining middle signal. However, to a complex system, when the number of each number is enough much, there needs use memory module to save data. When read and save data, it is necessary to provide an address, and then provide a trigger signal to ensure that read and store data correctly. In module design, every weight is represented by 24-bit binary, four-bit represented the integer part and 20-bit represented the fractional part, each signal save two weight data.

3.2.2. Output Values Calculation Module

Calculation of hidden layer output is the process that uses the positive and a negative symmetrical sigmoid function to obtain the corresponding output value [14].

The design of hidden layer output is shown as follows:

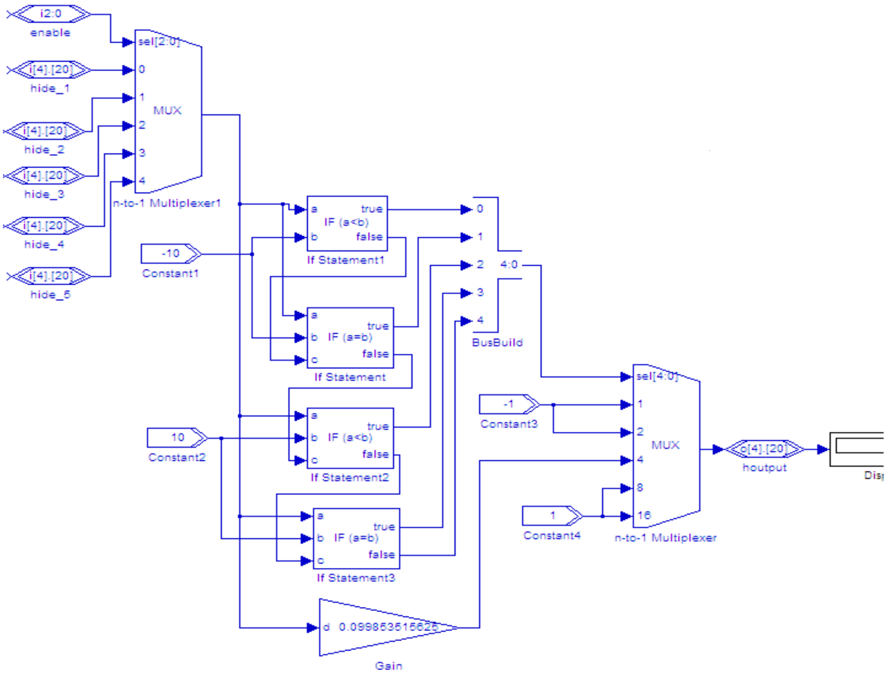


Figure 4. Design of hidden layer output values module

The function of a hidden layer is piecewise function

$$f(x)=\begin{cases}1 & x\geq 1\\ 0.95x & -1<x<1\\ -1 & x\leq -1\end{cases} \tag{1}$$

The above mentioned function replaced positive and negative symmetrical Sigmoid function, it simplified the calculation, So there can realize piecewise function by using IF sentence. IF sentence has three input terminal that is two comparisons input terminal a, b and one cascade input terminal c (the first IF sentence does not contain input terminal c). Hidden layer input values module and output values module cycle five times, and save the input and output value of the hidden layer. Similarly, use the input values of hidden layer to calculate input value of output layer then to obtain the output value of output layer that is parameters of PID controller.

4. Brief Introduction of PID Control Module

For the open-loop PID controller, the design of the calculation control module is very simple. The whole open-loop control module state machine is shown as follows:

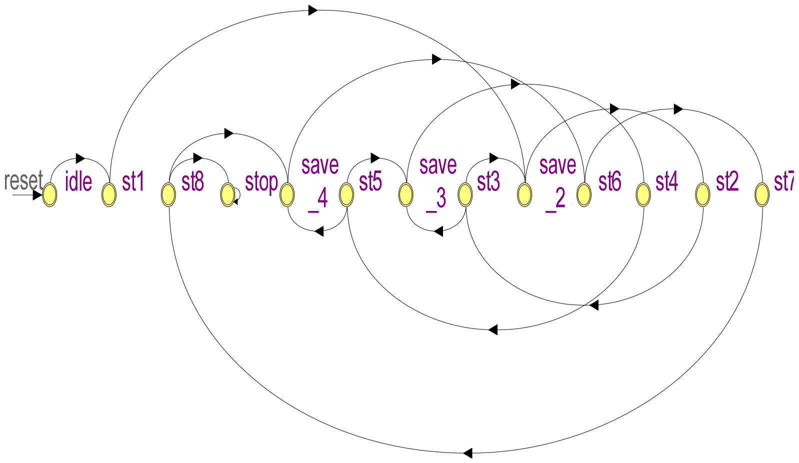


Figure 5. Structure graph of open-loop PID controller state machine

In the above figure, state idle and state stop is a reset state and an end state. There are 11 work state, respectively is: State st1 to state st8 and State save_2 to state save_4.

Three output values of BP neural network are parameters of PID controller. A calculation process of open-loop PID controller is the part of the whole closed-loop controller. The whole closed-loop control module state machine is shown as follows [15]:

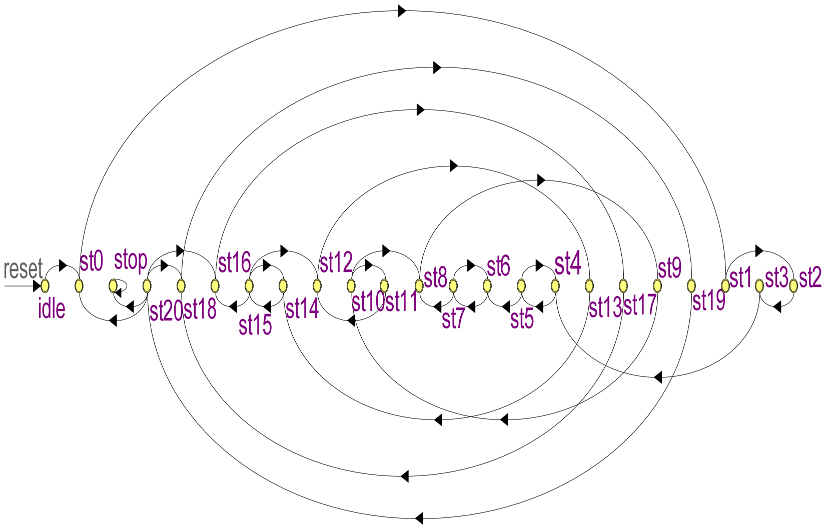


Figure 6. Structure graph of closed-loop PID controller state machine

In the above figure, state idle and state stop is a reset state and an end state. State st0 to state st20 are all internal work state, they can be divided four parts that respectively are computation state of PID controller parameters (state st3~state st10), adjustment state of hidden layer weight (state 11~state 15), adjustment state of input layer weight (state 16~state 20) and cycle assignment state (state st0~state st2).

5. Simulation Experiment and Results Analysis

Design simulation experiment based on hidden layer module, it is detailed prescribed as follows [16-18]:

5.1. Simulation of Hidden Layer Input Values

The following figure is Oscillograph of hidden layer input values module, it can be seen when the control signal appears to raise edge, output an input value hide_input of the hidden layer, ena_2 control five data output of the hidden layer. When the value of ena_2 is ‘000’ then rink, yk, errork and constant ‘1’ multiply corresponding high 24 bits of weight wi_11, low 24 bits of weight wi_11, high 24 bits of weight wi_21 and low 24 bits of weight, add the results then the sum is the first input value of the hidden layer, similarly, when the value of ena_2 is ‘001’, obtain the second the input value of the hidden layer.

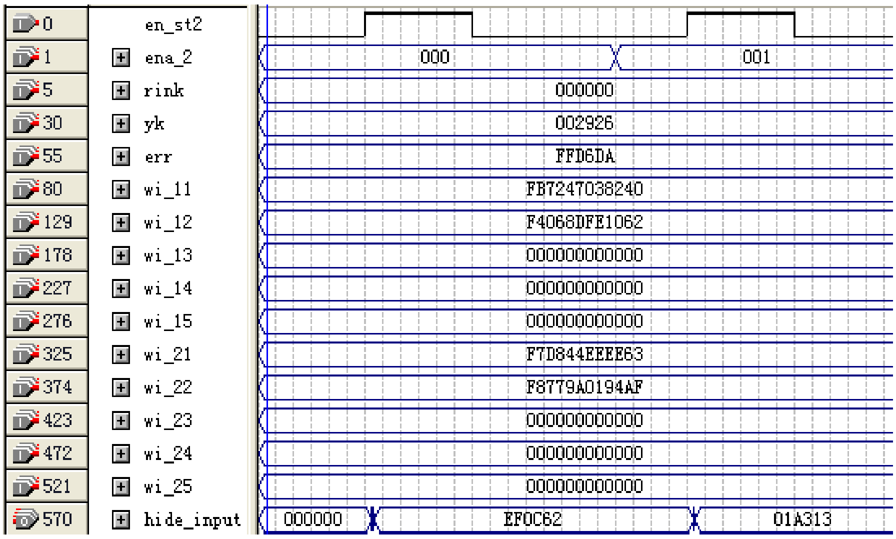


Figure 7. Oscillograph of hidden layer input values module

5.2. Simulation of Hidden Layer Output Values

The following figure is Oscillograph of hidden layer output values module, it can be seen when control signal en_st3 appears to rise edge, according to the different value of the ena_2 signal the output signal houtput (hidden output value) will get different data. When the value of ena_2 is ‘000’ then the system will input the value of hide_1 into piecewise function to get the first output value houtput of hidden layer.

$$f(x)=\begin{cases} 1 & x\geq 1 \\ 0.95x & -1<x<1 \\ -1 & x\leq -1 \end{cases} \tag{2}$$

Similarly, when the value of ena_2 is ‘001’ get the second output value of the hidden layer, like this circulation five times.

In following figure, when the value of ena_2 is ‘000’ then output the corresponding value of hide_1. If the value of hide_1 is ‘180000’ more than 1 then the output value is ‘100000’ and decimal value is 1. when the value of ena_2 is ‘001’ then output the corresponding value of hide_2. If the value of ena_2 is ‘E70000’ less than negative 1 then the output value is ‘F00000’ and decimal value is -1. when the value of ena_2 is ‘010’ then output the corresponding value of hide_3. If the value of ena_2 is ‘080000’ decimal value is 0.5 then the output value is ‘079980’ and decimal value is 0.4749755859375, the actual value should be 0.475 and error is 0.0000244140625.

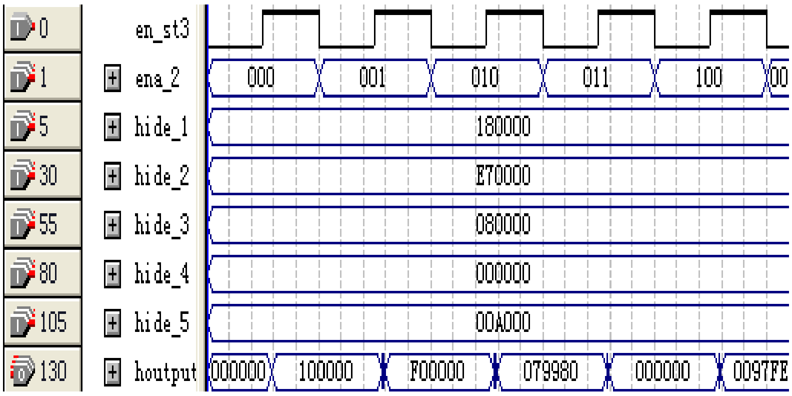


Figure 8. Oscillograph of hidden layer output values module

The second and following pages should begin 1.0 inch (2.54 cm) from the top edge. On all pages, the bottom margin should be 1-3/16 inches (2.86 cm) from the bottom edge of the page for 8.5 x 11-inch paper; for A4 paper, approximately 1-5/8 inches (4.13 cm) from the bottom edge of the page.

5.3. Closed Loop Simulation of PID Controller

The following figure is Oscillograph of closed loop simulation of PID controller, the value of system cycle control signal k is 1990 to 2000, the 2000th circulation get the value of Kp, Ki, Kd, respectively are 028075 H, 087130 H, 02 E554H, transform into the decimal respectively are: Kp=0.1563616, Ki=0.52763, Kd=0.1810133. Use the procedures of MATLAB simulation software and get Kp, Ki, Kd, the values respectively are: Kp = 0.1543, Ki = 0.52433, Kd = 0.17803. After comparison, the errors respectively are 0.00206, 0.0033 and 0.002983.

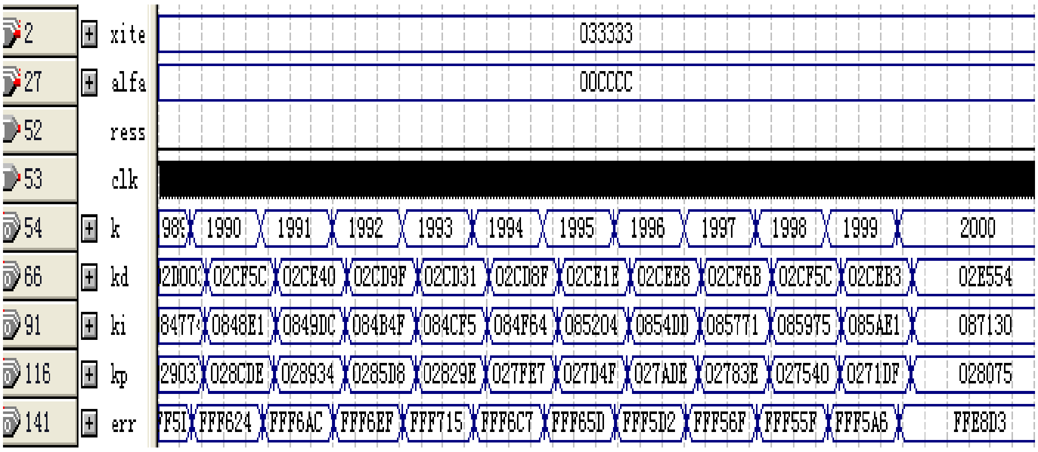
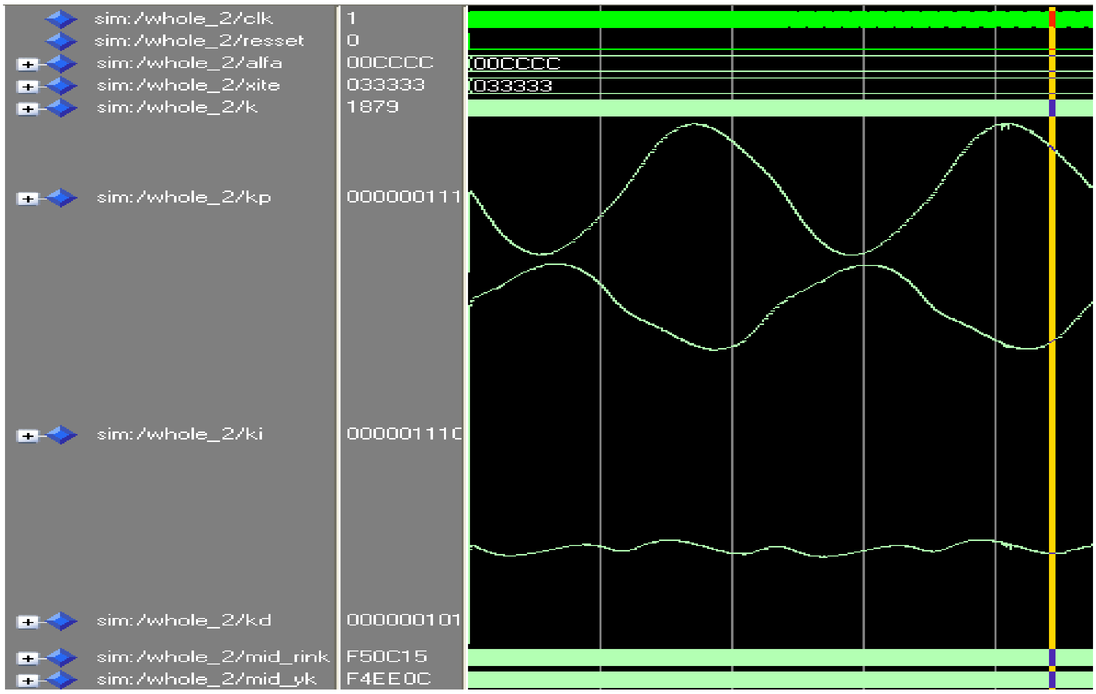

2 xite 033333
27 alfa 00CCCC
52 ress
53 clk
54 k 198 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000
66 kd 2D00 02CF5C 02CE40 02CD9F 02CD31 02CD8F 02CE1E 02CEE8 02CF6B 02CF5C 02CEB3 02E554
91 ki 8477 0848E1 0849DC 084B4F 084CF5 084F64 085204 0854DD 085771 085975 085AE1 087130
116 kp 2903 028CDE 028934 0285D8 02829E 027FE7 027D4F 027ADE 02783E 027540 0271DF 028075
141 err FF5I FFF624 FFF6AC FFF6EF FFF715 FFF6C7 FFF65D FFF5D2 FFF56F FFF55F FFF5A6 FFE8D3

Figure 9. Oscillograph of PID controller

Model sim is a simple, powerful logic simulation tools. Use Modelsim can get the analogue waveform of signal and to analyze the result is intuitive. Simulation VHDL documents of Quartus II by using Modelsim, the Oscillograph are shown as follows:



sim:/whole_2/clk	1
sim:/whole_2/resset	0
sim:/whole_2/alfa	00CCCC
sim:/whole_2/xite	033333
sim:/whole_2/k	1879
sim:/whole_2/kp	000000111
sim:/whole_2/ki	00000111C
sim:/whole_2/kd	000000101
sim:/whole_2/mid_rink	F50C15
sim:/whole_2/mid_yk	F4EE0C

Figure 10. Oscillograph of PID controller parameters by using Model simulation

6. Conclusion

This paper researched FPGA realization of PID controller based on BP neural network, the system module is divided and designed. The division of modules is very important to design. The article combines simulation technology to complete function confirmation based on feasibility and correctness of the whole system. The use of the simulation software verified the function and timing sequence, ensured the actual performance of the system, and guaranteed orderly work of each module under the clock coordination. In this paper, the programming language is VHDL language, it not only has completed the computation process from the input layer to output layer but also has completed the calculation of PID controlled object samples value. It ensures that the design module is universal and reliability and reduce the workload of design.

Acknowledgments.

This work is supported by National Natural Science Foundation of China (61073101, 51274011), supported by New Century Excellent Talents in University (No.NCET-10-0002), supported by Anhui Provincial Natural Science Foundation (No.1108085J03), supported by a grant from Natural Science Research Project of Anhui Province (KJ2012A097).

References

[1] Qu liguo, Sun xia, Huang yourui. FPGA implementation of QoS multicast routing algorithm of mine internet of things perception layer based on ant colony algorithm. *Advances in Information Sciences and Service Sciences*. 2012; 4(18): 124-131.

[2] T Sutikno, NRN Idris, AZ Jidin, A Jidin. A Model of FPGA-based Direct Torque Controller. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(2): 747-753.

[3] Min Lin, Yinli Fang. High-level synthesis and digital system design of VHDL. *Publish House of Electronics Industry*. 2002.

[4] Li yang, Wang qin, Li zhancai, Wan yong. The neural network hardware realization method based on FPGA. *Journal of University of Science and Technology Beijing*. 2007; 29(1): 90-95.

[5] Chen zhaoming, Bai xianglin, Gong xiaohong. Based on FPGA digital PID controller design. *Chongqing university of science and technology journal (natural science edition)*. 2012; 12(4): 149-151.

[6] Yaolin Cheng. Analysis of FPGA system design method. *Electronic Technology*. 2005; 19(1): 90-93.

[7] T Sutikno, AZ Jidin, A Jidin, NRN Idris. Simplified VHDL Coding of Modified Non-Restoring Square Root Calculator. *International Journal of Reconfigurable and Embedded Systems.* 2012; 1(1): 37-42.

[8] CC Shi and GS Zhang. *A New Method of PID Control Based on Improved BP Neural Network.* In proceedings of Chinese Control Conference. Harbin, China. 2006: 1167-1171.

[9] YJ Chen, Du Plessis. *Neural Network Implementation on a FPGA.* Proceedings of IEEE Africon. 2002; 1: 337-342.

[10] Hui Lv, Jing He, Gang Wang. Generation algorithms of pseudo random sequence nature polynomial. *Computer Engineering*. 2004; 30(16): 108-109.

[11] Wallace Tang. Hardware implementation of genetic algorithms using FPGA. *IEEE*. 2004; 1 (1): 549-552.

[12] Ying Chen, Yong-jie Ma, Wen-xia Yun. Application of Improved Genetic Algorithm in PID Controller Parameters Optimization. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(3): 1524-1530.

[13] Hanaa T El-Madany, Faten H Fahmy, Ninet MA El-Rahman, Hassen T Dorrah. Design of FPGA Based Neural Network Controller for Earth Station Power System. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(2): 281-290.

[14] Yin, Jialin. Research on the optimization PID parameters based on genetic algorithm. *Journal of Convergence Information Technology*. 2012; 7(11): 81-88.

[15] Shi Xiabo, Lin Weixing. PID control based on an improved cooperative particle swarm-bacterial hybrid optimization algorithm for the induction motor. *Advances in Information Sciences and Service Sciences.* 2012; 4(21): 517-524.

[16] Libao Su, Kezhu Song, Yanfang Wang. FPGA realization and research of Pseudo random generator. Journal of Circuits and Systems. 2003; 8(3): 121-123.

[17] Liu Jinkun. *Advanced PID control and MATLAB simulation*. Beijing: Electronic Industry Press. 2011.

[18] Ge zexue, Sun zhiqiang. *Neural network theory and MATLABR2007 realized*. Beijing: Electronic Industry Press. 2007.