The Implementation of Feedforward Backpropagation Algorithm for Digit Handwritten Recognition in a Xilinx Spartan-3

Panca Mudji Rahardjo, Moch. Rif'an dan Nanang Sulistyanto

***Abstract*—This research is aimed to implement feedforward backpropagation algorithm for digit handwritten recognition in an FPGA, Xilinx Spartan 3. This research is expected to give a contribution such as the feedforward algorithm design in VLSI technology based on FPGA, the practice module of Xilinx Spartan-3 development board and further research in artificial neural network and FPGA field in Electronics Laboratory.**

The feedforward backpropagation algorithm is used to recognize 10 objects. The feedforward backpropagation network consists of two layers, 36 input unit which is the feature vector of object, 10 hidden neurons, and 10 output unit. The first layer activation function is tansig and second layer activation function is purelin.

The multipliers use 18 bits. The proposed design fits into the smallest Xilinx FPGAs3.

***Index Terms*—feedforward backpropagation network, digit handwritten recognition, FPGA, Spartan-3.**

I. Introduction

In recent year, many researcher make a pattern and character recognition machine. Some researcher interested in handwritten character recognition, either separated or junctioned. This research focus is to recognize the separated handwritten, applying in post code, bank cheque, computer vision etc. Successive of character recognition system depends on result of the input segmentation process into single character. This segmentation result will be recognized detail. The problem facing in handwritten recognition is very complex, such as vary the handwritten model, pen for writing, etc [1][2].

The general requirement for a commercial digit recognition system is as follow [1]: (a) the recognition system is an author independent, (b) the recognition system is able to recognize a digit with the arbitrary size, (c) the recognition system robust to noise and background pattern, (d) the recognition system has very low error and rejected rate, (e) system is able to operate at high speed for a commercial applications.

One of the handwritten recognition method with high successive rate uses MLP (multi-layer perceptron) neural network with backpropagation training algorithm [2][3]. Each individual network classifies the different digit class.

Firstly, the computation algorithm being used in digital image and signal processing, multimedia, wireless communication, cryptography application and networking, is realized by using a software in a digital signal processor (DSP) or a general purpose processor (GPP). But, the development in VLSI (very large scale integration) technology, the hardware implementation will be an attractive alternative. Computation rate can be achieved by hardware computation intensively and parallel in algorithm [5].

Currently, field programmable gate arrays (FPGA) has raised as a choice platform for an efficient hardware implementation of computation intensive algorithm. Especially, if the design needs a high performance, a designer can utilizes high performance and high density FPGA to substitute an expensive multi core DSP system. An FPGA possibly performs in high parallel and achieves the higher speed than GPP. This is an advanced result of FPGA embedded. An FPGA has an advantage in hardware speed and software flexibility. An FPGA also offers a cheap and high rate programmable silicon.

Based on the advantage of backpropagation training algorithm and FPGA as above description, this research will implement the backpropagation feedforward algorithm in an FPGA Spartan-3.

The methodology in this research is as follow: Character data mining. The required data are handwritten digit from some writers vary in size and pen ink color. These data are scanned in 300 dpi resolution and grayscale.

Data preprocessing uses Matlab 7.0 software. The preprocessing are size normalization into 20×20 pixels and transition feature extraction. This transition feature vector becomes the FPGA's input.

Learning and setting the multi layer perceptron with Backpropagation algorithm uses Matlab 7.0 software. Learning is accomplished to set the network's weights and bias at it's minimal error. The network architecture refers to [2] and [4], the input vector of 36, hidden layer

Panca Muji Rahardjo. Author is with the Electrical Engineering Departement of Universitas Brawijaya, Malang, Indonesia (corresponding author provide phone 0341-665144; email panca@ub.ac.id)

Moch. Rif'an. Author is with the Electrical Engineering Departement of Universitas Brawijaya, Malang, Indonesia

Nanang Sulistyanto. Author is with the Electrical Engineering Departement of Universitas Brawijaya, Malang, Indonesia

of 10 and output layer of 10. The first layer activation function is tansig and the second layer activation function is purelin.

II. Basic Theory

A. System of Digit Handwritten Recognition

The character recognizer in this research is a multi layer net which is trained using back-propagation network. The character recognizer block diagram is shown in Figure 1. Before feeding into the character recognizer, a segmented character is normalized its size, then its feature is counted to make an input vector.

Character size normalization is performed due to the writer character size is different from each other. The character size is important thing for feature extraction. The normalized character size in this research is 20 × 20 pixels [2][4].

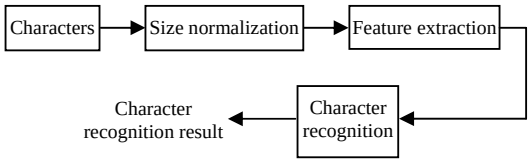


Figure 1. Digit recognition block diagram

B. Feature Extraction [2][4]

Feature extraction is performed to reduce the input vector size. Feature extraction in this research is transition feature. This feature counts location and amount of transition, from background to foreground along horizontal row and vertical column of a character image. The transition counting is performed from left to right, right to left, up-down and down-up. Because of constant dimension, a feature is needed for input vector to feed into neural network.

The amount of maximum transition, M, is counted for each row and column. When the transition is larger than M, then the first transition M is used only, others is ignored (Figure 2 with M is 3). If the transition is smaller than M, than the exceed value is zero. Assumed there are n transitions at each row/column at coordinate (x_i,y_i) for i = 1,2,…,n, the transition feature extraction algorithm is shown below:

```
For i=1:min(n,M),
        If  line_is_a_row then
                p_i=x_i;
                d=w;
        else
                p_i=y_i;
                d=h
        end;
        t_i=1-(p_i/d)
end;
if n<M then
        for i-n-1 : M
                t_i=0;
        end;
end;
```

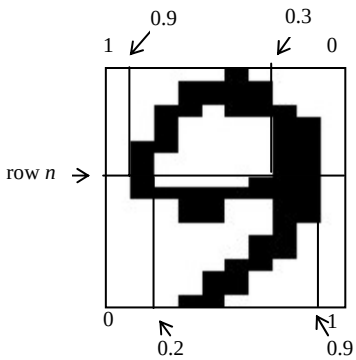


Figure 2 Computation of a row transition feature

After the transition is counted, then each column and row of the transition matrix is counted its mean value to get the transition feature mean matrix of $M \times M$ dimension.

This transition feature matrix is the feed-forward net's input. With this matrix, a 20×20 pixels character image is coded into 36 out of 400 column vector.

C. Back-Propagation Neural Net

The simple back-propagation learning method is gradient descent to minimize the output total square error. Applications using such nets can be found in virtually every field that uses neural nets for problems that involve mapping a given set of inputs to a specified set of target outputs (that is, nets that use supervised training) [6].

A multi layer neural network with one hidden layer (the Z units) is shown in Figure 3. The output units (the Y units) and the hidden units also may have biases (as shown). The bias on a typical output unit Y_k is denoted by w_{0k}; the bias on a typical hidden unit Z_j is denoted by v_{0j}. These bias terms act like weights on connections from units whose output is always 1.

After training, a backpropagation neural net is applied by using only the feed-forward phase of training algorithm. The application procedure is as follows [6]:

Step 0. Initialize weights (from training algorithm).

Step 1. For each input vector, do Steps 2 – 4:

Step 2. For i = 1,….,n : set activation of input unit x_i;

Step 3. For j = 1,….,p;

$$z_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \;;$$

$$z_j = f(z_in_j)$$

Step 4. For k = 1,….,m;

$$y_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \;;$$

$$y_k = f(y_in_k)$$

The above application procedure will be implement-

ted in an FPGA, in this research.

D. *FPGA Spartan-3 development board*

The implementation of a feedforward back-propagation algorithm in an FPGA Spartan-3 development board needs the basic concept knowledge of its operation and structure. FPGA Spartan-3 belongs to the fifth generation of Xilinx family. An FPGA consists of programmable logic block array which connected into themselves by vertical and horizontal connection line [5].

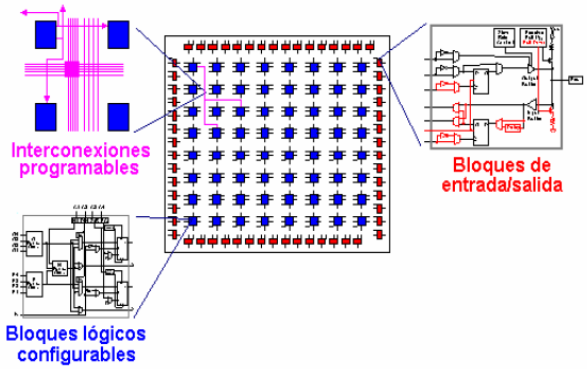


Figure 4 Basic architecture of FPGA [5]

An FPGA has characteristic as follows :

- Low power consumption.
- High speed.
- More endurance.
- Fast development time.
- Simple in methodology.
- Simple in equipment.

An FPGA Spartan-3 is ideal for low cost application and high volume. It is designed instead of fix logic gate structure and for application special IC (ASIC).

A Spartan-3 device has regular and flexible architecture consist of five programmable basic functional element : configurable logic blocks (CLBs), surrounded by input/output block (IOB), block RAM, multipliers (18×18) and digital clock managers (DCMs).

III. Result and Analysis

With 200 set data for learning stage, the network is trained in 6000 epoch. The learning mean square error (MSE) graphic is shown in Figure 5.

Based on Matlab computation, the bias of network are:

v_{01} = 0.04; v_{02} = 0.032; v_{03} = 0.14; v_{04} = 0.11; v_{05} = 0.21; v_{06} = 0.04; v_{07} = 0.022; v_{08} = 0.013; v_{09} = 0.01; v_{010} = 0.02; etc.

A part of VHDL code is shown in Figure 6.

```
--Description : feedforward backpropagation algorithm.
--Engineer    : Panca Mudji Rahardjo
--Date        : October 20, 2010

Library IEEE;
Use IEEE.STD_LOGIC_1164.ALL;
Use IEEE.STD_LOGIC_ARITH.ALL;
Use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

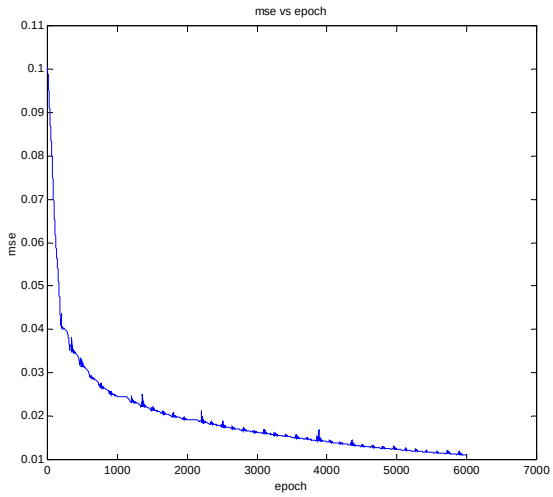


Figure 5 The learning graphic

```
entity backprop is
  Port (input : in integer range 0 to 255;
        clock,enter : in std_logic;
        sel0,sel1,sel2,sel3 : out std_logic;
        output: out std_logic_vector(6 downto 0));
end backprop;

architecture Behavioral of backprop is

signal x1,x2,x3, x4, x5, x6, x7, x8, x9, x10: integer range 0 to 255;
signal x11,x12,x13, x14, x15, x16, x17, x18, x19, x20: integer range 0 to 255;
signal x21,x22,x23, x24, x25, x26, x27, x28, x29, x30: integer range 0 to 255;
signal x31,x32,x33, x34, x35, x36: integer range 0 to 255;
signal data_ke,next_data: integer range 1 to 3 :=1;
signal proseslah: integer range 0 to 1 :=0;
signal state,nextstate: integer range 1 to 4 :=1;
signal clkdiv: std_logic_vector(31 downto 0)
:="00000000000000000000000000000000";
signal new_clk : std_logic;

.............
................

constant v01: integer :=40;
 constant v02: integer :=32;
constant v03: integer :=140
constant v04: integer :=110;
constant v05: integer :=210;

...........
...........

begin

data_masukan:
process (data_ke)
  begin
    if proseslah/=1 then
  case data_ke is
  when 1 =>
    x1<=input;
    next_data<=2;
  when 2 =>
    x2<=input;
    next_data<=3;
  when 3 =>
    x3<=input;
    next_data<=1;
  end case;
............
end if;
  end process;
```

```
process (enter)
  begin
  if enter='1' and enter'event then
    if data_ke=3 then
      proseslah<=1;
    else
      data_ke<=next_data;
    end if;
  end if;
end process;

process (x1,x2,x3,clock)
      begin
        if proseslah=1 then
      if  clock='1' and clock'event then
        a11 <= x1 * w111;
        a12 <= x1 * w112;
        a13 <= x1 * w113;
        a14 <= x1 * w114;
        a15 <= x1 * w115;

        a21 <= x2 * w121;
        a22 <= x2 * w122;
        a23 <= x2 * w123;
        a24 <= x2 * w124;
        a25 <= x2 * w125;

.........
.........
       end if;
       end if;

      end process;

recognition_result:
process (o1,o2,o3,clock)
  begin
    if o1>o2 and o1>o3 then
      output <= "1001111"; --1
    else if o2>o1 and o2>o3 then
      output <= "0010010"; --2
    else if o3>o1 and o3>o2 then
      output <= "0000110"; --3
    else output <= "1111110"; --
................
................
    end if;
    end if;
    end if;
end process;

display_to_7segment:
process (state)
  begin
    case state is
    when 1 => sel0<='0';sel1<='1';sel2<='1';sel3<='1';nextstate<=2;
    when 2 => sel0<='1';sel1<='0';sel2<='1';sel3<='1';nextstate<=3;
    when 3 => sel0<='1';sel1<='1';sel2<='0';sel3<='1';nextstate<=4;
    when 4 => sel0<='1';sel1<='1';sel2<='1';sel3<='0';nextstate<=1;
    end case;
  end process;

process (clock)
begin
  if clock='1' and clock'event then
    clkdiv<=clkdiv+1;
    end if;
end process;

new_clk<=clkdiv(26);  --display scanning every 1 second (50 MHz/2^26)

process (new_clk)
begin
  if new_clk='1' and new_clk'event then
    state<=nextstate;
    end if;
end process;
end Behavioral;
```

Figure 6 VHDL code

The RTL schematic and realization in Spartan-3 development board are shown in Figure 7 and 8 respectively.

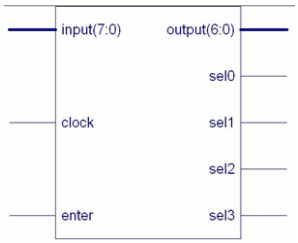
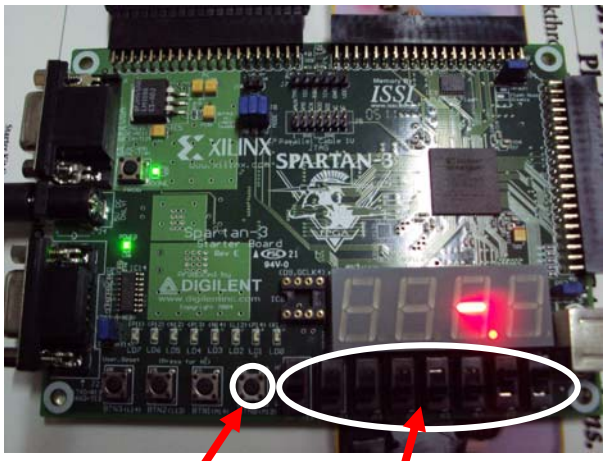
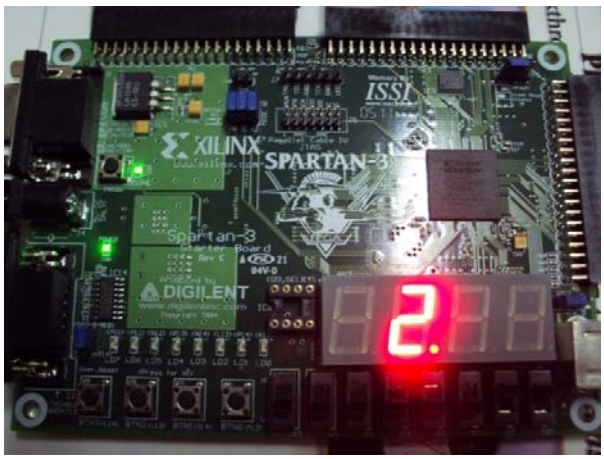


Figure 7 RTL schematic of the implementation



ENTER (a) 8 bit inputs



(b)

Figure 8 (a) Implementation in Spartan-3 development board
(b) seven segment display when it recognize class '2'

Device utilization summary is shown in Table 1.

Table 1 Device utilization summary

Logic Utilization	Used	Available	Utilization
Total Number Slice Registers:	3,050	3,840	79.43%
Number used as Flip Flops:	2,003		
Number used as Latches:	1,047		
Number of 4 input LUTs:	3,100	3,840	80.73%
Number of occupied Slices:	1,672	1,920	87.08%

Number of Slices containing only related logic	1,672	1,672	100%
Number of bonded IOBs:	21	173	12%
Number of MULT18x18s:	12	12	100%
Number of GCLKs:	3	8	37%

The recognition summary is shown in Figure 9.

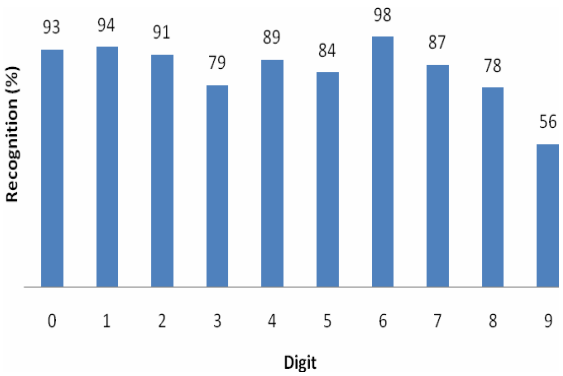


Figure 9 Recognition rate

IV. Conclusion

A feedforward backpropagation network has been implemented in an FPGA Spartan-3. The algorithm requires 36 inputs of 8 bits. 36 inputs are the transition feature of a handwritten digit image. Due to the implementation has one input of 8 bits, the entering process of 36 inputs use shift register that it is activated by an 'enter' button.

Acknowledgement

I would like to thank the Engineering Faculty of Universitas Brawijaya for the financial support within the frame of the research projects 'DIPA' No. 15/H10.6/PG/2010.

References

[1] Kimura, F. Shridhart, M. '*Handwritten Numerical Recognition Based on Multiple Algorithms*'. Pattern Recognition, Vol. 24, No. 10. pp. 969-983, 1991

[2] Panca Mudji Rahardjo. "*Penerapan Jaringan Perambatan-Balik untuk Pengenalan Kode Pos Tulisan Tangan*". Thesis. Universitas Gadjah Mada. 2001.

[3] Chaaban, I, "*Applying Domain Knowledge to The Recognition of Handwritten ZIP Codes*", Thesis. Indiana University South Bend. 2007.

[4] Panca Mudji Rahardjo, Herman Tolle. "*Penerapan Jaringan Perambatan Balik untuk Sistem Deteksi Kesalahan pada Pengenalan Angka Tulisan Tangan*". Jurnal Ilmu-Ilmu Teknik. Lembaga Penelitian Universitas Brawijaya. Vol. 17, No. 1, April 2005, hal. 85.

[5] Syed M. Qasim, Shuja A. Abbasi, Bandar A. Almashary. "*Advanced FPGA Architectures for Efficient Implementation of Computation Intensive Algorithms: A State-of-the-Art Review*". MASAUM Journal of Computing, Volume 1 Issue 2, September 2009.

[6] Laurenne Fausett. "*Fundamentals of Neural Network*". Prentice Hall. 1994.

[7] Stefan Sjoholm, Lennart Lindh. "*VHDL for Designers*". Prentice Hall. 1997.

[8] Karen Parnell, Nick Mehta. "*Programmable Logic Design Quick Start Handbook, CoolRunner-II RealDigital CPLD XC9500XL*". Xilinx. August 2003.

Panca Mudji Rahardjo was born in Sumenep, Indonesia in 1970. In 1988, he took college one year in Institut Teknologi Sepuluh Nopember Surabaya, at Diploma of Civil-Hydro Technique. He obtained the B.Eng. Degree in Electronics Engineering, Universitas Brawijaya, Indonesia, in 1995 and M.Eng. Degree in Electronics and Signal Engineering, Universitas Gadjah Mada, Indonesia, in 2001. He joined as Assistant Engineer of Production Engineering Dept. at PT. Asahi Electronics Indonesia, a Telephone Answering Device manufacturer, in '96-'98; as Engineering Assistant of Engineering Dept. at PT. Tokyo Pigeon Indonesia, an Audio Mechanism manufacturer, in '98-'99; and as Lecturer of Electrical Engineering Dept. at Universitas Brawijaya, since 2002. He served as the Head of Electronics Laboratory at Universitas Brawijaya since 2007.

His current research interests include Instrumentation System Design, Pattern Recognition and Digital System Design.