

# Implementation of ANN architecture in FPGA for recognizing the numbered music notation

## Abstract

In this paper, we have managed to develop a Number Note recognition system using Artificial Neural Network (ANN). This system is implemented on FPGA Altera Cyclone IV DE0 Nano EP4CE22F17C6. Our developed system has a maximum frequency of 57.16MHz with the clock cycle required for 1 epoch (one training process per sample input) is 216 clock cycles.

**Keywords:** artificial neural network, FPGA, recognition system, numbered music notation

Volume 2 Issue 3 - 2018

Baskoro Widyarachmanto,<sup>1</sup> Putut Dewantoro,<sup>1</sup> Luthfi Naufan Yamin,<sup>1</sup> Syifaul Fuada,<sup>2</sup> Trio Adiono<sup>2</sup>

<sup>1</sup>School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Indonesia

<sup>2</sup>University Center of Excellence on Microelectronics, Institut Teknologi Bandung, Indonesia

**Correspondence:** Trio Adiono, University Center of Excellence on Microelectronics, Institut Teknologi Bandung, Indonesia, Tel +62-22-2506280, Email tadeono@stei.itb.ac.id

**Received:** July 27, 2018 | **Published:** November 16, 2018

## Introduction

Learning music is one of positive activity<sup>1</sup> especially for the kids development because it can give the experience to make them in fun condition and training the brain intelligence stimulus as well. Many kinds of instruments can be utilized to learn the music in which the most favorite way is by reading the music sheet. However, the problem is whether the kids know what kind of sound that the number note produce.

It is important for kids to know so, as they need to play the right note of the song they play. Hence in this project, we aim to develop a system that could recognize the number note on a music sheet with the system producing the respecting sound of the note it recognized. We used the method of Artificial Neural Network (ANN) in which it is as a tool to classify an object based on its class. It consists of multiple layers of simple processing elements called neuron. The mathematical model of network including the concepts of inputs, weights, summing function, activation function and outputs. The ANN helps to decide the type of learning for adjustments of weights with change in parameters. Finally the completed ANN implementation and training. The overall evaluation of the implementation is in an FPGA Altera Cyclone IV DE0 Nano EP4CE22F17C6.

## Architecture description

In this work, we follow the guideline of the LSI Design 2017 Competition that available.<sup>2</sup> The SoC design guides<sup>2</sup> are very complete from the calculations (include the Neuron, Cost Function, Gradient Descent, and Delta Value) till system block of ANN. We edited the source code in line with our system specification. As stated in the research background, we used the ANN to recognize number notes. We decided to limit our system into 3 outputs, which are 1, 2, and 3 in which the detailed of these outputs classification can be seen.<sup>2</sup>

## Pre-processing

ANN is a machine learning tools that needs complex calculation.

Hence, these tools need long time to perform or large areas for component. In this system we have handwritten digit images as an input that has 28x28 pixels. With this input, we need large memory for the input, weight, bias, etc. Because this image has big resolutions, therefore we need pre-processing to make the input much smaller other than before.

For images that captures by camera for testing the result then, the image will resized to 28x28 pixel before processed using Principal Component Analysis (PCA). This process is needed to make the PCA output closed to the train images. Therefore, this processed will reduce the dimension for the ANN and will increase the speed of calculation and decrease the area that needed to perform the ANN.

## Forward propagation

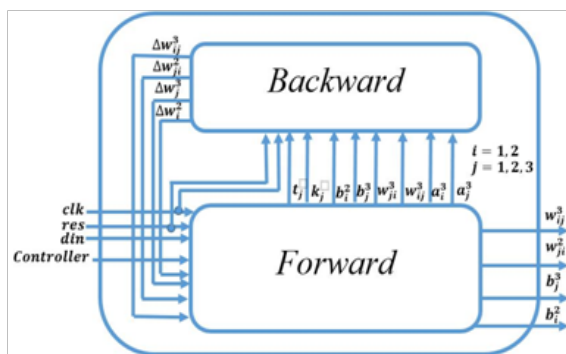
After the images was pre-processed then now is the time to make them an input of the ANN. The first thing to do before training is the feed forward phase. We will see whether the input and the supervised is true or false. The output of this feed forward phase is a binary number. The amount of this phase input is 15, the hidden layer is 5, and the output is 3 as mentioned in the specification.

## Back propagation

This process is used to do a gradient descent so the weight and bias can be updated to make true classification on forward propagation and make the error (cost function) convergent to 0.

## Top level

Top level is a program that combine forward propagation block and backward propagation block and we used controller to select the process. So, there are no overlapping with that two process and the system can run synchronously (Figure 1). Integration between backward block with forward block, the detailed of architecture of each block can be seen in.<sup>2</sup>



**Figure 1** There are no overlapping with that two process and the system can run synchronously.

### Appeal points and originality

**Number note application:** As character recognition is one of the basic recognition in image processing, we thought about what kind of application we could use this on. Hence, as we already mentioned in the introduction, we decided to use it on number note recognition so it would produce the sound of the note and help children learn music.

**PCA:** Using an image as an input to the ANN would bring a numerous problem if not pre-processed. If the image value is used as the input per pixel, it would demand a huge area to the design, while also complicating the design even more. Thus, as we also have mentioned before, we use the PCA method to reduce the area requirement of our system and make it implementable in an FPGA.

## Results and implementation

### Synthesis result

In this work, we only show the performance test of developed system including synthesis, simulation in RTL, and analysis of critical path. The result of synthesis is summarized in the Table 1 which is based on the FPGA Altera Cyclone IV DE0 Nano EP4CE22F17C6.

**Table 1** Synthesis result

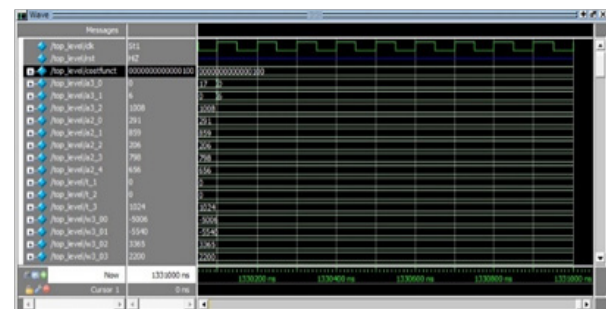
Resources	Used	Available	%
Number of Logic Elements	13,213	22,320	59
Number of Registers	4,905	22,320	22
Number of Pins	18	154	12
Number of Memory bits	352	608,256	<1
Embedded Multiplier 9-bit elements	104	132	79
Maximum Frequency on "Slow 1200 mV 85C Model"	57.16MHz		

In this work, we focus on smaller areas, with the trade-off of a slower calculation process. Even though the area is relatively small, the processing itself is not too slow. To process all the input samples from forward to the end of the backward propagation (1 epoch), our design manage to process it with 216 clock cycles.

### Simulation result

The values of the weights and biases have also been updated to the most accurate values, though the modelsim version that we used

to simulate couldn't show the fractional number we set into decimal so it was shown as an integer instead (Figure 2).



**Figure 2** Shows the simulation result of the circuit for the whole process of the ANN training. It is shown that the cost function has reached 0000000000000100, or 0.00390625 if converted into decimal.

### Simulation result

#### Critical path and analysis

Our implementation requires 216 clock cycles to complete one epoch. At the maximum clock settings of 57.16MHz, we have the critical path value at 17.49ns. Hence, we have managed to build a system where one epoch finishes at 3.777μs. The training of Neural Network would require a large number of epochs, which could lead to a slow training process. However, with one epoch only requiring 3.777μs, we could manage a high number of epoch in training with not as much time consumed. Even with 100,000 epochs, it would take about 377.77ms, which is still considerably fast, considering we chose to focus on making the area smaller.

#### Source-code

The Verilog code of this work can be downloaded in the link as follow:

<https://drive.google.com/file/d/171xtJnacYY3j00qE813Rh1k171bWjBvu/view?usp=sharing>

## Conclusion and future work

The developed system has been simulated and has given a converging value of cost function, signing the system has ran successfully. In the future work, we would like to implement the whole system completely with Zybo FPGA board which has a similar area constraints to the board we use. Later, we will design the systolic array processor to obtain more high-throughput.

## Acknowledgements

This paper was compiled from the final report of VLSI course in School of Electrical Engineering and Informatics, Institut Teknologi Bandung under supervision of Dr. Trio Adiono.

## Conflict of interest

The author declares there is no conflict of interest.

## References

1. M Tervaniemi. Music in learning and relearning: The life-span approach. *Psychomusicology Music, Mind, and Brain*. 2017;27(3):223–226.
2. *Guide of LSI Design and Contest*. 2017.