HARDWARE AND SOFTWARE IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK IN ALTERA DE1-SOC

LIM CHUN MING

UNIVERSITI SAINS MALAYSIA

2017

HARDWARE AND SOFTWARE IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK IN ALTERA DE1-SOC

by

LIM CHUN MING

Thesis submitted in partial fulfillment of the

requirements for the degree of

Bachelor of Engineering (Electronic Engineering)

JUNE 2017

ACKNOWLEDGEMENTS

This dissertation is dedicated to everyone in the field of artificial neural network in embedded system who embarks the journey of expanding the collection of knowledge and transcendent passion for artificial neural network in embedded system.

Firstly, I want to thank my final year project and research supervisor Assc. Prof. Dr. Zaini Abdul Halim for her patient guidance and advices throughout the time of doing my final year project. Her continuous support and precious advices are one of the factors in successful completion of my project and are highly appreciated. I feel proud to be one of the student of my supervisor.

Besides, I would also like to thank the PhD student, Tan Earn Tzeh for his precious advices and support for my final year project. He is always willing to share his experiences and knowledges to me and giving me advices about the final year project and research.

In addition, I want to thank my precious course mates that study together for 4 years in my university life. They are always willing to spend their time with me, sitting together exchanging knowledges, discussing about the problems faced during the time of doing final year project.

Lastly, I would like to thank my family for giving me the love and support. Without their support, I would not have successfully completed my final year project and thesis.

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURES

LIST OF ABBREVIATIONS

ALM	Adaptive Logic Module
ALU	Arithmetic Logic Unit
ANN	Artificial Neural Network
ASIC	Application Specific Integrated Circuit
CPU	Central Processing Unit (processor)
DSP	Digital Signal Processing
FMAX	Maximum Frequency
FPGA	Field Programmable Gate Array
GUI	Graphic User Interfaces
HDL	Hardware Description Language
HPS	Hard Processor System
LED	Light Emitting Diode
LUT	Look Up Table
MAC	Multiply-Accumulate Circuit
MLP	Multilayer Perceptron
MSE	Mean Squared Error
PC	Personal Computer
ROM	Read Only Memory

RAM	Read and Write Memory
SOC	System On Chip
VHDL	Very High-Speed Integrated Circuit HDL
VT	Ventricular Tachycardia
VF	Ventricular Fibrillation

PERLAKSANAAN PERKAKASAN DAN PERISIAN RANGKAIAN NEURAL TIRUAN DALAM ALTERA DE1-SOC

ABSTRAK

Rangkaian neural tiruan banyak digunakan dalam banyak aplikasi dan mula digunakan dalam sistem terbenam. Baru-baru ini, platform yang baru seperti Altera DE1-SOC yang mempunyai kedua-dua pemproses and FPGA telah diperkenalkan di pasaran. Apabila menggunakan platform ini, rangkaian neural tiruan boleh dilaksanakan samada di dalam pemproses menggunakan perisian atau di dalam FPGA menggunakan perlaksanaan perkakasan. Analisis perlu dijalankan untuk mengetahui samada pemproses atau FPGA lebih sesuai untuk melaksanakan rangkaian neural tiruan. Dalam projek ini, rangka kerja tentang pelaksanaan ANN pada pemproses dan FPGA dalam Altera DE1-SOC telah dibuat dan prestasi pelaksanaan tersebut tentang ketepatan, masa pelaksanaan dan penggunaan sumber telah dikaji. Beberapa model multilayer perceptron(MLP) dengan bilangan input, bilangan neuron tersembunyi dan jenis fungsi pengaktifan yang berbeza telah dilatihkan di MATLAB dulu, barulah dilaksanakan pada kedua-dua pemproses dan FPGA. Eksperimen juga telah dijalankan untuk mengukur prestasi model-model ini dalam pemproses and FPGA. Setelah membandingkan keluaran dengan ANN pada MATLAB dan mengirakan purata kuasa dua ralat (MSE), keputusan menunjukkan ANN pada pemproses mempunyai 100% ketepatan dan ANN pada FPGA mempunyai minimum 7.3×10^{-6} MSE. Manakala, ANN pada FPGA 20 kali lebih laju berbanding dengan ANN pada pemproses. Oleh itu, sekiranya satu sistem memerlukan ketepatan yang tinggi, ANN dicadang untuk dilaksanakan pada pemproses. Manakala, sekiranya masa pelaksanaan sangat penting, ANN dicadang untuk dilaksanakan pada FPGA.

HARDWARE AND SOFTWARE IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK IN ALTERA DE1-SOC

ABSTRACT

Artificial neural network (ANN) has been widely used in many applications and has been started to be implemented in embedded system. Recently new platform like Altera DE1-SOC that contains both processor and FPGA had been introduced. When using this type of platform, artificial neural network can be either implemented in processor using software implementation or in FPGA using hardware implementation. Analysis should be done to see whether processor or FPGA is a better choice for the ANN. In this project, framework for implementation of ANN in processor and FPGA of Altera DE1-SOC has been developed and the efficiency of implementation of ANN in processor and FPGA in terms of accuracy, execution time and resources utilization has been studied. Several multilayer perceptron (MLP) models with different number of inputs, number of hidden neurons and types of activation function have first been trained in MATLAB and after that, these trained models have been implemented in both processor and FPGA of Altera DE1-SOC. Experiments have been carried out to test and measure the performance of these MLP models in processor and FPGA. After comparing output result with ANN that run in MATLAB and computing the mean squared error (MSE), results show that the ANN in processor has 100% accuracy and ANN in FPGA has minimum MSE of 7.3×10^{-6}. While ANN in FPGA is 20 times faster than ANN in processor. Therefore, if accuracy is main priority and execution time is not so important in a system, ANN is suggested to be implemented in processor. However, if execution time of ANN must be fast like less than microsecond in a system, ANN is suggested to be implemented in FPGA.

CHAPTER 1

INTRODUCTION

1.1 Research Background

Artificial neural network (ANN) is a computational model that is based on the interconnection of the neuron in the nervous system of human brain. ANN can be trained using learning algorithm and data set to solve problems. ANN can be used for pattern recognition, classification, prediction etc.

Today, artificial neural network has been widely used in various field of application like photovoltaic system [1], [2], medical diagnosis system [3], stock market prediction [4] etc. Most of the ANN implementations are PC based simulation software model [5], [6] which is relatively slow compared to hardware implementation type of ANN [7]. Besides, many of the ANN implementation in medical diagnosis system is not portable [8], [9]. Thus, more research should be done to study the hardware implementation of ANN model to increase the portable neural network solution in the market.

When it comes to portable embedded platform, CPU, ASIC, FPGA are available and recently, hybrid CPU-FPGA platform which integrate processor together with FPGA has been introduced in the market. In October 11, 2011 Altera Corporation introduces a new SOC FPGAs device that integrating a dual core ARM Cortex-A9 MPCore Processor with 28-nm Cyclone V and Arria V FPGA fabric [10]. High-level management functionality of processor and real-time, high flexibility, extreme data processing FPGA becomes one of the powerful embedded platform.

Since hybrid CPU-FPGA platform has both processor and FPGA, sometimes one may have to consider which one to use for the application. FPGA with parallel feature has higher data processing rate, but design with high performance that use up a lot of resources may have a higher cost. [11]

It is important to have a detail analysis on the performance of application like artificial neural network on processor and FPGA based on several aspects like accuracy, execution time, resource utilization or cost to justify whether implementation on processor or FPGA is better for the artificial neural network.

In this project, implementation of multilayer perceptron neural network (MLP) in Altera DE1-SOC platform will be presented. Five MLP models with different structure, for example different number of input feature, different number of hidden neuron, and different type of activation function will first be trained in MATLAB and after that, trained models will be implemented on FPGA and Arm Cortex A9 of DE1-SOC separately. Performance of each model on FPGA and processor will be evaluated and analyzed based on accuracy, execution time and resource utilization. Through this project, people may gain a better and thorough understanding on the effectiveness of implementation of ANN models in processor and FPGA. This study will give a clear comparison of performance on the implementation of ANN models on processor and FPGA in hybrid CPU-FPGA platform like Altera DE1-SOC.

1.2 Problem Statement

Recently, new hybrid CPU-FPGA platform like Altera De1-SOC that integrates Arm Cortex A9 with FPGA fabric has been introduced in the market. When implementing application or system for example artificial neural network in this platform that has both processor and FPGA, some may wonder whether artificial neural network should be

implemented in processor or FPGA. Implementation of ANN model in FPGA may have high data processing rate but the cost might be high if the design takes up huge area of hardware resources. ANN model is easy to be developed in processor and lower cost but the execution time of ANN model might be longer. A detail analysis should be done to see whether CPU or FPGA is a better choice of implementation for artificial neural network in hybrid platform Altera DE1-SOC.

1.3 Objective of Research

1) To develop a framework for implementation of artificial neural network in Altera DE1-SOC board.

2) To study the efficiency of implementation of artificial neural network in processor and FPGA in terms of accuracy, execution time and resources utilization.

1.4 Scope of project

In this project, multilayer perceptron neural network will be implemented in processor and FPGA of Altera DE1-SOC board. Five multilayer perceptron models with different number of inputs, different number of hidden neurons and different type of activation function will first be trained in MATLAB and trained neural network will be implemented in processor using C programming language and will be implemented in FPGA using Verilog hardware description language. After that, experiments will be carried out to test and measure the performance in terms of accuracy, execution time and resources utilization for several MLP models implemented in processor and in FPGA. After that, the result of the experiments will be analyzed.

1.5 Thesis Outline

This thesis consists of 5 chapters. Chapter 1 is introduction that discusses about the research background, main motivation of doing this research, problem statement, objective of the research and scope of the project.

Chapter 2 is literature review that reviews about the previous related work done on implementation of artificial neural network in FPGA and in microprocessor. Some basic knowledge related to the project is also being discussed.

Chapter 3 is methodology. It discusses about the project implementation flow, design flow for software and FPGA, structure of multilayer perceptron, model of neuron and the way of implementing it, type of activation function and the way of implementing it. Besides, experimental procedure regarding testing and measuring the performance of several MLP models in terms of execution time, resources utilization and accuracy is also being discussed.

Chapter 4 presents the result and analysis of the experiments designed in previous chapter. The performance of several MLP models in processor and FPGA in terms of accuracy, execution time and resources utilization will be presented. Analysis regarding the results will also be discussed.

Chapter 5 is conclusion that presents the summary of the work and results of the project. Future works of this project is also included.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

Recently, researchers have started to do research on hardware or software implementation of artificial neural network in portable embedded platform. Many methods of implementation and type of ANN have been studied for the past few years. In this chapter, some fundamental knowledge related to project has been discussed. Besides, previous work done on implementation of artificial neural network on embedded platform like FPGA or microprocessor also has been reviewed in this chapter also.

2.2 Artificial Neural Network

Artificial neural network is a computation model that consists of many interconnected nodes or neuron inspired by the way of how biological neural network in brain process information. [12] Each neuron in the artificial neural network is a processing element. Artificial neural network can be trained to solve problems and used for pattern recognition, classification, prediction etc. The most common form of artificial neural network is multilayer perceptron as shown in Figure 2.1. Multilayer perceptron is a feedforward neural network where data flows only in one direction from input layer through hidden layer and to output layer. Each neuron in layer is connected to all neurons in next layer with certain weights.

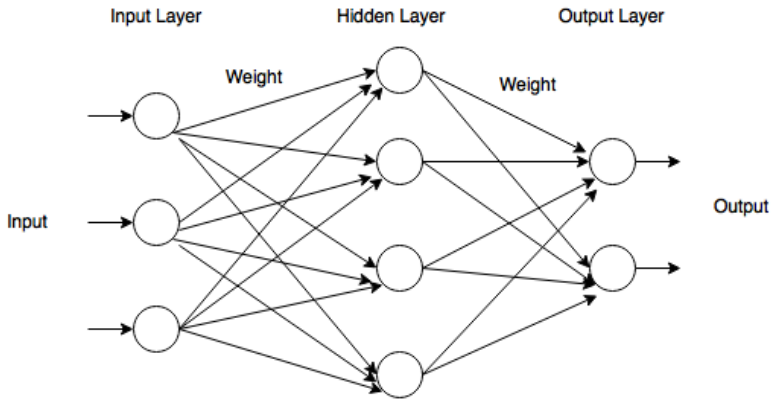

Figure 2.1: Multilayer Perceptron Model

The most common used training algorithm for artificial network is backpropagation algorithm where output is compared with target data and error function is computed. This error is then feedback through the network and weight is adjusted based on the error [13]. The algorithm adjusts weight to reduce the error function. The process continues until the error becomes small.

2.3 Processor (CPU)

Processor or Central Processing Unit (CPU) is an electronic circuitry that executes the instructions of a program in sequential manner by performing arithmetic, logic and control operations. CPU basically contains arithmetic logic unit (ALU) that performs arithmetic and logic operations, register that store the result of ALU, and control unit that control the fetching and execution of instruction.

2.4 Field Programmable Gate Array (FPGA)

Field programmable gate array is a semiconductor device that is designed to be configured by designer or customer to a desired application with certain functionality requirements. [14] FPGAs contain an array of programmable logic blocks and reconfigurable interconnects that allows the logic blocks to be wired together to perform complex combinational function. Hardware description language (HDL) is used to specify the configuration of the logic block. Figure 2.2 shows the structure of FPGA fabric.

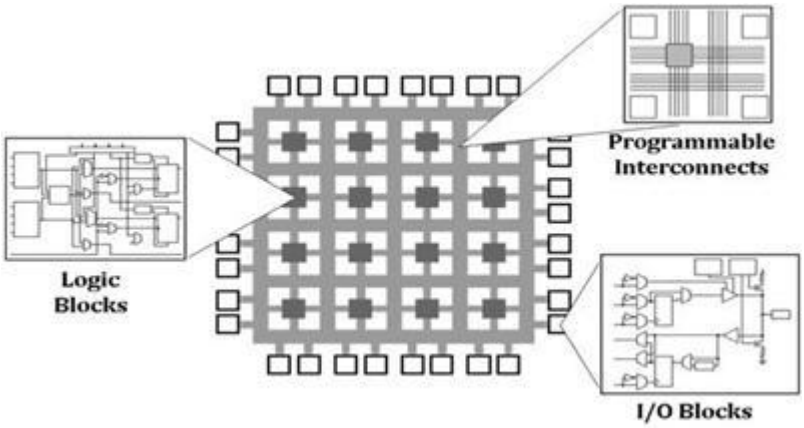

Figure 2.2: FPGA fabric structure [15]

One of the important features of FPGAs is the ability of parallel processing which is different to processor that runs program in sequential manner. Flexibility is another feature of FPGA since the functionality of the FPGA can be changed every time the device is powered up. Change can be made by downloading new configuration file into the device.

2.5 CPU-FPGA Hybrid Platform

Recently, a new hybrid platform that integrating processor with FPGA fabric had been released. Altera DE1-SOC is one of the CPU-FPGA hybrid platform that integrates ARM-based hard processor system with FPGA fabric using high-bandwidth interconnect. [16] Hard processor system consists of dual-core ARM Cortex-A9 embedded cores, peripherals and memory interfaces. Besides, Altera DE1-SOC board also contains Ethernet networking, video capabilities and DDR3 memory. With this hybrid platform, user can implement design in processor only by using software, or in FPGA by using hardware description language, or in both processor and FPGA by using hardware software co-design method.

2.6 Hard Processor vs Soft Processor

Hard processor is a type of processor system where all components have been fixed in the board and cannot be reconfigured after manufactured. [17] While soft processor refers to the processor system that created using resources in field programmable gate array (FPGA) and can be customized and reconfigured based on desire specification. Soft processor has better flexibility than hard processor because of the ability to reconfigure. [18] However, the hard processor has lower power consumption and better performance compared to soft processor.

2.7 Floating Point vs Fixed Point Number Representation

Floating point is a type of number representation that used for fractional number. [19] Floating point is based on scientific notation where it consists of mantissa and exponent. The decimal point in floating point format can 'float' and not fixed. While fixed point is alternative way to represent fractional number where the decimal point is fixed. The number of digits or bits before and after the decimal point is fixed. Floating point has better dynamic range and precision than fixed point. [20] However, fixed point implementation is less costly than floating point implementation in hardware.

2.8 Related Works

K.V.Ramanaiah, et al.[7] had proposed a parallel multilayer perceptron neural network architecture that implemented on Xilinx Virtex family FPGA device. The proposed hardware architecture was being implemented in two phases. First phase includes the training of neural network in MATLAB program. In the learning stage, the weight and bias were updated based on the backpropagation training algorithm. Second phase was hardware implementation of the trained neural network on the Xilinx FPGA device. The size of the neural network was 16-4-16 that means first layer with 16 input features, hidden layer with 4 neurons and output layer with 16 neurons. For the multiplication part of the architecture, Wallace tree based arithmetic was used. The activation function used in the hidden layer is a tangent sigmoid, while the activation function used in output layer is a pure linear function. For implementation of the tangent sigmoid function in FPGA, look up table (LUT) method was used and this LUT was stored in ROM. After experimental analysis, the architecture on FPGA was found to be operated at the

maximum speed of 138MHz, occupying 3% of the resources and consuming a total power of 55.285mW.

Another researchers Andres Orozco-Duque, et al. [21] had presented the implementation of artificial neural network on a 32bit ARM Cortex M4 microcontroller and on a Xilinx FPGA Spartan 6 for real time detection of ventricular tachycardia (VT) and ventricular fibrillation (VF). The implemented neural network structure had 4 layers: 1 input layer with 4 neurons ,2 hidden layer each with 3 neurons and 1 output layer with one neuron. The activation function used at the hidden layers was sigmoid function and the activation function used at the output layer was pure linear function. The ANN model was first trained in the MATLAB program using backpropagation algorithm. After that, the trained ANN model was implemented in FPGA using Xilinx System Generator. ROM block is used to implement the step by step approximation for sigmoid function. For microcontroller part, the neural network was implemented on C. Andres Orozco-Duque, et al used CMSIS 2.0 library for the implementation of ANN. CMSIS 2.0 library consists of more than 60 DSP function that includes fixed point and single precision floating point math that is possible to help to implement the prediction algorithm of neural network with matrix operation. For the activation function of the ANN, it is not necessary to use approximation function as in FPGA as it can use "math.h" library in C. After experimental analysis, test result shows an accuracy of 99.46% and execution time of 30us in FPGA and less than 0.6ms in microcontroller. Andres Orozco-Duque, et al. mentioned that low power consumption was the advantage of using microcontroller and it was easy to implement complex mathematical function using C programming language.

A.Thilagavathy and K.Vijaya Kanth [22] had presented the digital hardware implementation of artificial neural network for image recognition. The proposed neural network was a multilayer perceptron network with 1 input layer, 1 hidden layer and 1

output layer. The proposed neural network was implemented on Xilinx Spartan 3E device by using VHDL. A.Thilagavathy and K.Vijaya Kanth had used three types of activation function in this research which were hyperbolic tangent sigmoid, symmetric saturating linear and symmetrical hard limiter. The motivation of the research was to test out three different types of activation function and compare the performance of these three types activation function in neural network. After simulation using Model Sim 5.7, results show that total execution time for multilayer perceptron network with hard limit function was 15.820ns, 15.231ns for saturating linear function and 29.598ns for hyperbolic tangent sigmoid function. In terms of resources utilization, hard limit function used the least number of resources and hyperbolic tangent sigmoid used the highest number of resources of FPGA. This research is good since it tested on 3 different types of activation function and compared their performance. But it didn't describe more detailed on the method used to implement these activation function. Different method of implementation may affect the result of the experiment.

Philippe Dondon, et al. [23] had also done an implementation of a feed-forward neural network on FPGA using VHDL language. The platform used was Zynq-7000 EPP 7Z020 Zedboard kit. The ANN model was first trained beforehand in MATLAB, after training was done, the weight and bias were extracted. Philippe Dondon, et al. used ROM to store the weight and bias and used RAM to store the input of network. Data format of 32 bit where 1 bit for the sign, 21 bit for the integer and 10 bit for the fractional part. State machine was used to control the modules. The size of the network was 3 layers where input layer consists of 3 neurons and hidden layer consists of 3 neurons and output layer consists of only 1 neuron. 2 types of solution of neural network structure had been tested. One module per neuron was used in first structure and each neuron carried out operations in parallel. Once initialization start, each of them did the multiplication and summing

process, and finally sent the result to RAM. This solution allows a high-speed advantage but might end up using a lot of FPGA resources. Second solution of structure using only 1 module per layer. Additional module was added to each neuron module to remind the neuron module about its position in the network. This type of structure consumed less amount of FPGA resources compared to the first solution but end up increasing the process time. After simulation, the first solution consumed 16 DSP components of total 220 DSP components which enable the implementation of total 55 neurons. In this research, the author did not test out different type of activation function which may be a factor of affecting the overall resources consumptions of the design.

Sami El Moukhlis, et al. [24] had proposed another FPGA implementation of artificial neural network for classification of handwritten signature. The ANN model was first trained in MATLAB using backpropagation algorithm, after that the weight and bias were extracted. The proposed ANN model was implemented on Altera DE2-70 FPGA and being described using Very High Speed Integrated Circuits Hardware Description Language (VHDL). The proposed architecture of ANN has 3 layers with the size of 20 neurons in input layer, 6 neurons in hidden layer and 1 neuron in output layer. Both hidden layer and output layer used sigmoid activation function and 16 bits MAC or multiply accumulate circuit was used. For the implementation of the sigmoid function, piecewise liner approximation method which consists of linear approximation of log sigmoid function was used instead of look up table (LUT) method. After simulation, the proposed design used up 39% of the resources and only 5% of the embedded multiplier.

In this research, Sami El Moukhlis, et al. [24] used piecewise linear approximation instead of LUT may decrease the amount of resources needed but might end up increase the execution time of the ANN model [F] since LUT is better than piecewise linear approximation in terms of execution time.

Andre L.S Braga, et al. [25] had done an interesting research on analyzing the implementation of ANN models in both FPGA by hardware description language (HDL) and in Xilinx MicroBlaze embedded soft microprocessor by software approach in terms of resources utilization, accuracy, speed and power consumption. The author highlighted several factors that will affect the performance of the implementation of ANN like data presentation, the way of implementation for activation function etc. Author had created 3 ANN models for FPGA with VHDL and 4 models for MicroBlaze soft microprocessor. First FPGA model named VHDL A was a 13 bits fixed point where 3 bit for integer part and 10 bit for fractional part with activation function being implemented in the form of look up table (LUT). VHDL B was also 13 bits fixed point with activation function implemented using combinatorial design. VHDL C was a 32 bits fixed point where 4 bits for integer part and 28 bits for fractional part with activation function being implemented using piece-wise linear approximation technique. Besides, same ANN model was also programmed in C language and implemented in MicroBlaze embedded soft microprocessor. After that experiment had been carried out. Results showed an increasing number of logic blocks used from VHDL A to VHDL C. VHDL A had used the most of the memory spaces among other designs. This is due to the presence of the look up table that consumed a lot of memory resources. MicroBlaze design had better accuracy than FPGA designs but FPGA designs were way faster than MicroBlaze designs.

The research done by Andre L.S Braga, et al. [25] was very good because he analyzed different ANN models in terms of power consumption, resources utilization, accuracy and speed. But he did not take into account of other important factors like number of input, number of hidden neuron and type of activation function into his ANN model for experimental analysis. These factors may also affect the performance of the FPGA design.

A. P. do A. Ferreira and E. N. da S. Barros [26] had proposed an architecture that could support the implementation of large ANN models by concerning the area reduction. The proposed architecture had a size of 256 input neurons, 10 hidden neurons and 10 output neurons. Instead of using m+1 multipliers and compute each neuron individually, the proposed method used less adders for entire layer. The proposed method help reduces the resources consumption. Another important feature of this architecture is that the area versus performance parameter can be tuned or adjusted by making some changes in the circuit. All the designs were implemented by using Verilog and synthesized with Synopsys Synplify Premier and Altera Quartus 2. After simulation, the final designs consumed 30% of the ALUTs components. In this research, author had done a good job by analyzing a big architecture with the proposed method. But in the proposed method, author only focus on the structure of multiply accumulation unit but didn't take into account of the type of activation function which may be an important factor that will affect the performance of whole architecture.

OZDEMIR, A.T. and DANISMAN, K. [27] had done a comparative study on two artificial neural network architectures. First model was a 32-bit floating point FPGA based model. Sigmoid function was used at the hidden and output layer. Second model was a 16-bit fixed point FPGA based model. Sigmoid function was used at the hidden layer while pure linear function was used at the output layer. Piecewise linear approximation method was used to implement sigmoid function. Both architecture had a sized of 8 input neurons, 2 hidden neurons and 1 output neuron. First model achieved accuracy of 97.66% and second model achieved accuracy of 96.54%. However, resource utilization of second model was lower than the first model. Piecewise linear approximation approach for sigmoid function consumed less hardware resources than the real sigmoid activation function block in the first model. Classification time for the first

model is 1.07us and 1.01us for second model. Second model was faster because of the 16-bit fixed point numerical representation since the routing in FPGA was highly reduced.

Boussaa Mohamed, et al. [28] had done a comparison of Nios2 software based and hardware based of implementation of multilayer perceptron neural network on FPGA Altera DE2-70 in terms of execution time and hardware resources utilization. NIOS 2 is a 32-bit soft core microprocessor that being built using logic of the FPGA resources. The hardware implementation in FPGA was being described using VHDL. The implementation of sum, multiplication, division and exponential used mega functions block described by VHDL. These mega functions block were intellectual property proposed by developer of QUARTUS. For the software implementation part, microprocessor NIOS 2 was being implemented first using QUARTUS and SOPC builder. After that the multilayer perceptron model can just being programmed using NIOS 2 EDS that allows the use of C, C++ and assembler language. After all implementation in hardware and software part done, these implementations had been tested. Architecture size used was 4 input neurons, 3 hidden neurons, and 2 output neurons. All modes received same synaptic weights and input vectors. Results showed that execution time of software model was 1s and 1.9us for the hardware model. Although hardware model was way faster than software model but the resources consumed by software model was less than hardware model.

In overall, research done by Boussaa Mohamed, et al. [28] was good because a comparison on software model of ANN in soft processor and hardware model of ANN in FPGA in terms of execution time and hardware resources utilization had been done. But only one type of size of ANN architecture which is 4-3-2 was tested. It would be better if he did several more architecture that took into account of some factors like number of input, type of activation function and etc to make it a more detailed analysis.

Table 2.1: Summary of Related Works

References	Year Published	Platform	ANN architecture	Activation function of output layer	Measurement	Features
[25] Andre L.S Braga, et al.	2010	Xilinx FPGA	1 input, hidden and output layer	Hyperbolic tangent sigmoid	Accuracy Execution time Power consumption Resources Utilization	Comparison of ANN in FPGA and soft microprocessor in terms of time, accuracy, power, resources
[22] A.Thilagavathy and K.Vijaya Kanth	2013	Xilinx Spartan 3E	1 input, hidden and output layer	Hyperbolic tangent sigmoid Saturating linear Hard limiter	Execution time Resources Utilization	Several activation functions were implemented and performance was compared
[27] OZDEMIR, A.T. and DANISMAN, K.	2013	FPGA	8-2-1	Sigmoid Pure linear	Execution time Resources Utilization	2 type of FPGA based ANN was compared
[7] K.V.Ramanaiah, et al.	2014	Xilinx Virtex FPGA	16-4-6	Pure linear	Execution time Resources Utilization	ANN with large size was implemented
[28] Boussaa Mohamed, et al.	2015	FPGA Altera DE2	4-3-2	Not reported	Execution time Resources Utilization	Comparison of ANN in FPGA and soft processor NIOS 2 in terms of time and resources

2.9 Summary

In overall, there were quite some of the researchers did research on the implementation of artificial neural network on FPGA or embedded soft microprocessor. Most of the findings showed that many of the researchers only focus on the FPGA implementation of ANN and they did not analyze and compare the performance between implementation in processor and FPGA. Although researcher Andre L.S Braga [25] did have some analysis and compare the performance of implementation of ANN model on FPGA and implementation of ANN model using software approach on embedded soft microprocessor in terms of resources utilization, execution time and accuracy, however there is still some improvement that can be made. The improvement is a more detailed analysis that include some other important factors like number of input, number of hidden neurons and type of activation function in testing of performance. Besides, there is still lack of detailed analysis of ANN on hybrid platform like Altera DE1-SOC that consists of hard processor and FPGA. It would be better if several ANN models that includes previously mentioned factors can be tested on processor and FPGA of hybrid platform like Altera DE1-SOC.

CHAPTER 3

METHODOLOGY

3.1 Overview

In this project, artificial neural network (ANN) will be implemented on FPGA using Verilog and will be implemented on processor using C programming language. The platform that being used is the Altera DE1-SOC that contains both hard processor system (HPS) and FPGA fabric. The ANN that will be implemented is a multilayer perceptron neural network (MLP) with 1 input layer, 1 hidden layer and 1 output layer. 5 MLP models with different activation function, number of input features and number of hidden neurons will be first trained in MATLAB and after that the trained network will be developed and implemented on FPGA and HPS. After that experiments will be carried out and the performance of these MLP models on FPGA and HPS in terms of execution time, resources utilization and accuracy will be measured and analyzed. With these collected and analyzed data, the performance of implementation of artificial neural network on FPGA and processor can be easily compared. Detail process that include implementation of ANN and experimental procedure will be explained in this chapter. Figure 3.1 shows the overall project implementation flow.

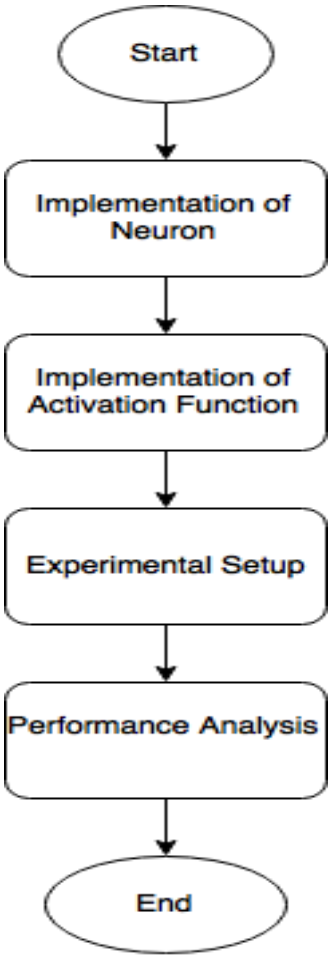

Figure 3.1: Project Implementation Flow

3.2 Implementation Platform

In this project, Altera DE1-SOC board will be used for ANN implementation. Altera DE1-SOC is a hybrid platform that integrates ARM based hard processor system that consists of processor, memory interfaces and peripherals with industry leading FPGA fabric using high bandwidth interconnects.

3.2.1 Hard Processor System

Processor of hard processor system (HPS) is 800Mhz dual-core ARM Cortex-A9 MPCore processor with 1GB DDR3 SDRAM using 32bit data bus. Processor executes program or software in a sequential manner. Figure 3.2 shows the software design flow. At first, design or architecture of ANN will be defined and then it will be described by coding using C/ C++ programming language. After coding is done, it will be compiled, debugged and will be ready to be uploaded to processor for execution. Linux is used as operating system for ARM processor.

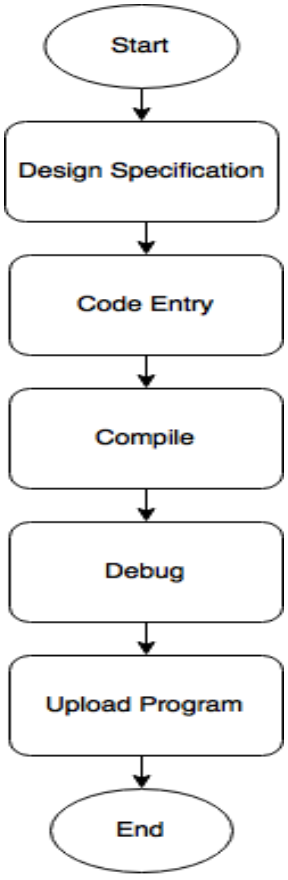

Figure 3.2: Software Design Flow

3.2.2 Field Programmable Gate Array (FPGA)

FPGA is Altera Cyclone 5 SE 5CSEMA5F31C6N device with 64MB SDRAM using 16bit data bus. FPGA is able to process data in parallel. The FPGA hardware design flow is shown in Figure 3.3. At first, design or architecture of ANN will be defined and then it will be described using Verilog hardware description language in software ModelSim Altera. After coding is done, functional verification will be carried out to check whether the design is being correctly described. If the design is correct, logic will be synthesized for the design. Place and route is the process of interconnecting all the logic blocks.

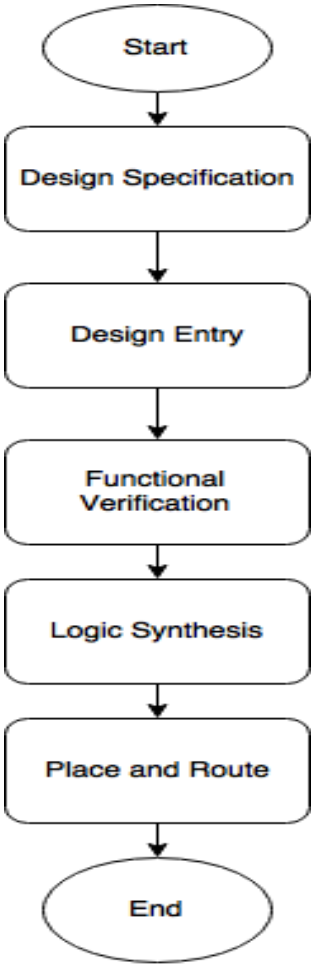

Figure 3.3: FPGA Hardware Design Flow

3.3 Implementation of neuron

Artificial neural network used in this project is a multilayer perceptron with 3 layers: input layer, hidden layer, output layer. Multilayer perceptron is a type of feedforward neural network where information flows through layers in one direction only. Unit in each layer is called neuron. Neuron in layer is connected with certain weights to all neurons in next layer. All neurons except the input layer neuron are considered as processing elements. Figure 3.4 shows the structure of a multilayer perceptron neural network.

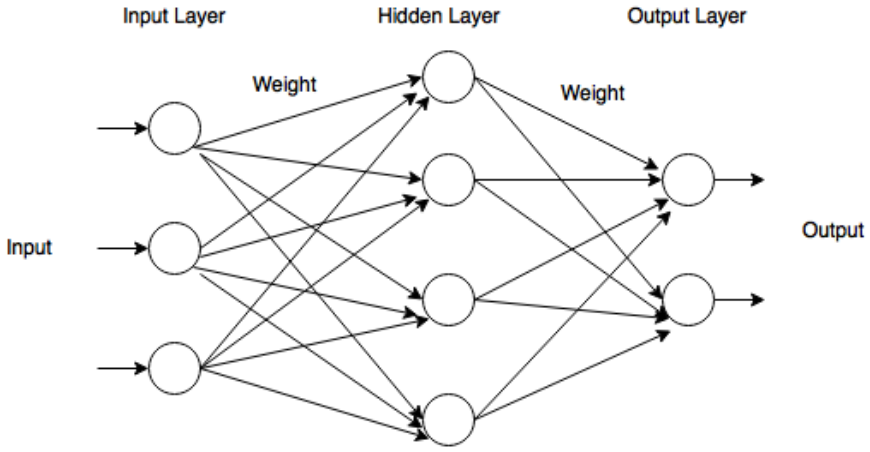

Figure 3.4: Multilayer perceptron models

Figure 3.5 shows the structure of artificial neuron where inputs of the neuron will be multiplied by respective weights. After that all products will be summed together with bias and go through the activation function. Activation function will then define the output based on the inputs.

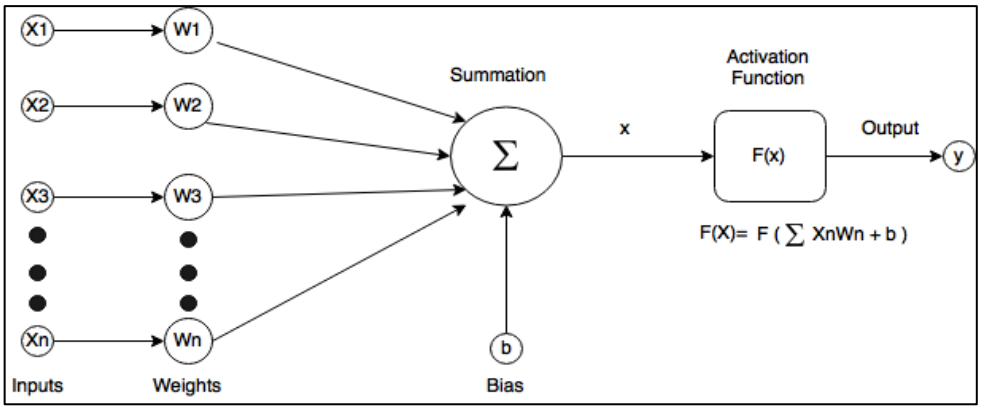

Figure 3.5: Artificial neuron model

3.3.1 Software Implementation part

For the implementation of neuron in processor, program like using a loop to keep multiplication of inputs with weights until all inputs have been multiplied can be designed. After that the sum will be added by bias and applied in activation function. Figure 3.6 shows the program flow chart for processing in neuron.

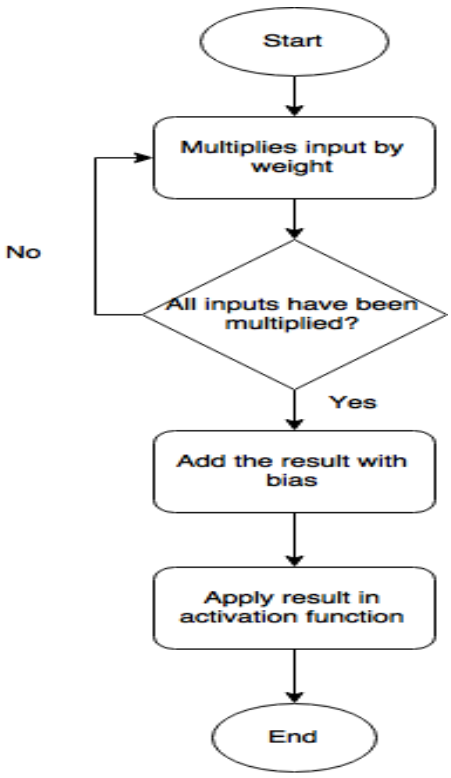

Figure 3.6: Algorithm flow chart for processing in neuron

64 bits double precision floating point will be used as data type for inputs, weights and bias in the C program of ANN. Figure 3.7 shows the total program flow chart of the ANN. Processing of neurons will be carried out one by one. After all neurons in hidden layer have been processed, processing of the neurons in output layer will be carried out one by one until getting the final output.

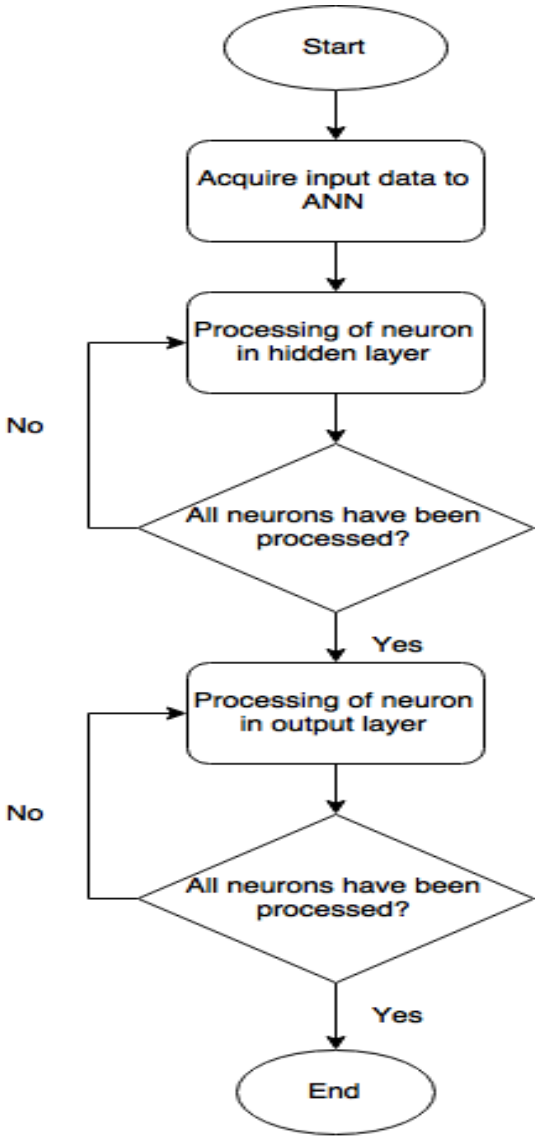

Figure 3.7: Overall program flow chart of ANN.