An ASIC-Based Artificial Neural Network Applied Real-time Speech Recognition SOPC

Lam D. Pham, Hieu M. Nguyen, Du N. N. T. Nguyen and Trang Hoang

Abstract—Artificial Neural Network (ANN) is promoted to one of major schemes applied in pattern recognition area. Indeed, many approaches to software-based platforms have proven great performance of ANN. However, developing pattern recognition systems integrating ANN hardware-based architecture has been limited not only by the silicon requirements such as frequency, area, power, or resource but also by high accuracy and real-time applications strictly. Although a considerable number of ANN hardware-based architectures have been proposed currently, they have experienced a deprivation of functions due to both small configurations and ability of reconfiguration. Consequently, achieving an effective ANN hardware-based architecture so as to adapt to not only strict accuracy, enormous configures, or silicon area but also real-time criterion in pattern recognition systems has been really challenged. To tackle these issues, this work has proposed a dynamic structure of three-layer ANN architecture being able to reconfigure for adapting to various real-time applications. What is more, a complete SOPC system integrating proposed ANN hardware has also implemented to apply Vietnamese speech recognition automatically to confirm high recognition probability around 95.2 % towards 20 Vietnamese discrete words. Moreover, experiment results on such ASIC-based architecture have witnessed maximum frequency at 250 MHz on 130nm technology as well as great ability of reconfiguration.

Index Terms—Artificial Neural Network (ANN), System on a Programmable Chip (SOPC), floating Point, booth algorithm, activation function.

1. Introduction

ALTHOUGH Moores law has been still valued to confirm the increasing of transistor density gradually, distinct applications utilizing ANN consisting of an enormous number of single neurons exceed limitation of resource. As a matter of fact, the authors of [1] and [2] presented critical issues relating to drawback of hardware resource, which is needed to be halted in future work. In order to not only configure a large ANN successfully but also adapt to silicon requirements feasibly, popular methods have attended to apply approximate models to reduce utilized resource. Particularly, almost techniques have referred to combine between fixed-point data formats and approximated activation functions in structure of single neuron showed in [3] - [6]. To be more specific, these published papers have approximated nonlinear functions such as Sigmoid and Tan-Sigmoid to forms of linear functions being able to implement hardware-based architectures feasibly. Nevertheless, the trade-off between accuracy and using resource is greatly concerned whenever complex systems integrating ANN hardware require high accuracy strictly. Obviously, an ANN comprising of a considerable number of single neurons requires enormous operations, which leads to a lack of accuracy critically with too many approximated formulas. Moreover, the training results in pattern recognition machines applying ANN algorithm has experienced a variety of weight coefficients in real format number. As a result, if fixed-point data format utilized to approximate nonlinear activation functions is also used in hold system completely, the measurement uncertainty will negatively affect performance inevitably. Obtained results as TABLE 1 confirmed the comparison among approximated methods with original Sigmoid. These obtained errors will increase gradually after data is transferred through other single neurons in network.

$$E_{eva} = \frac{\sum_{i=0}^{N} |\widehat{f}(x_i) - f(x_i)|}{N} \quad (1)$$

$$E_{max} = max|\widehat{f}(x_i) - f(x_i)| \quad (2)$$

where: $\widehat{f}(x_i)$ is result of hardware implementation while $f(x_i)$ is from expected theory

In terms of ANN application, a considerable number of published results based on SOPC systems approaching a FPGA chip have confirmed the efficiency. It is vital that dynamic SOPC systems can be feasibly reconfig-

Lam D. Pham was with the Department of Electrical and Electronics Engineering, University of Technology, Ho Chi Minh City, Vietnam (Email: lamd.pham@hcmut.edu.vn).

Hieu M. Nguyen was with the Department of Electrical and Electronics Engineering, University of Technology, Ho Chi Minh City, Vietnam (Email: nguyenmhieu@hcmut.edu.vn).

Du N. N. T. Nguyen and Trang Hoang are with the Department of Electrical and Electronics Engineering, University of Technology, Ho Chi Minh City, Vietnam (Email: www.iheartech.hcmut.edu.vn).

Manuscript received November 15, 2015; revised February 3 and April 15, 2016; accepted May 29, 2016.

Digital Object Identifier 10.31130/jst.2016.22

TABLE 1
Performances of approximated methods

Activation Function	Maximum Error (E_{max})†	Mean Error (E_{eva})†
M.Jame [6]	0.0822	0.0596
Zhang [5]	0.0215	0.0076
Allipi [3]	0.0189	0.0087
Plan [4]	0.0189	0.0058

† E_{eva} and E_{max} are mean absolute error (MEA) and maximum absolute error, respectively

TABLE 2
Artificial Neural Network configurations

Previous Works	ANN Configuration	Technology
Dhirajkumar S. Jinde [10]	Three layers: 2-6-2	FGPA
Esraa Z. Mohammed [11]	Two layers: 3-1	FPGA
Alexander Gomperts [12]	Four layers: 10-6-3-2	FPGA
K. Venkata Ramanaiah [13]	Two layers: 2-16	130 nm
Panca M. Rahardjo [14]	Three layers: 36-10-10	FPGA
Suhap Sahin [15]	Three layers: 2-3-1	FPGA
Hassne Faiedh [16]	Three layers: 2-12-1	350 nm

ured to generate larger ANN to obtain the superior performances. Specially, a proposed system referred from [7] showed complete speech recognition SOPC configuring a large three-layer ANN, while another SOPC illustrated in [8] presented smaller ANN configuration. Nonetheless, these solutions basing on reconfigured SOPC systems dynamically have not solved real time issue completely. Particularly, the research described in [9] also pointed out the trade-off between software and hardware implementations related to timing consumption.

As regards ANN configuration, detailed information given by TABLE 2 illustrates concerning statistics towards published ANN hardware architectures implemented on different platforms. Despite of adapting to real-time requirement, almost those reveal limited resource with maximum configuration at three layers (36:10:10) as in [14]. Based on TABLE 2, most of researches have approached to FPGA-based designs to be able to setup various ANN configurations flexibly, whereas ASIC-based designs have experienced the deprivation of results mainly caused of the dynamical setting ability.

From such circumstances, an effective dynamic ANN architecture is proposed in this work so as to balance these issues. First and foremost, floating-point data format identified by IEEE 754 standard is applied for all formulas in complete design to ensure high accuracy. Thus, ASIC-based ANN architecture has experienced effective combination between single neuron and controller so as to perform not only ability of flexible reconfiguration by input ports but also silicon requirement and real-time specification. Furthermore, word-level speech recognition SOPC integrating the proposed ANN hardware is also implemented to confirm the higher efficiency in comparison to other researches.

The rest of paper is organized follows. Section II introduces applied techniques integrated in proposed ANN hardware such as *IEEE 754 standard*, *Booth* algorithm as well as applied ASIC design flow. Next, Section III proposes ANN architecture integrated in complete SOPC system for speech recognition application and Section IV describes the experiment results to witness achieved successes. Finally, Section V consults the paper and future work.

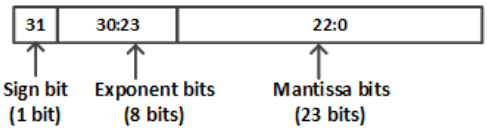

Fig. 1. IEEE 754 floating point format.

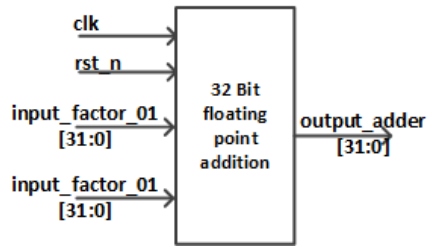

Fig. 2. Interface of 32 bit floating point addition.

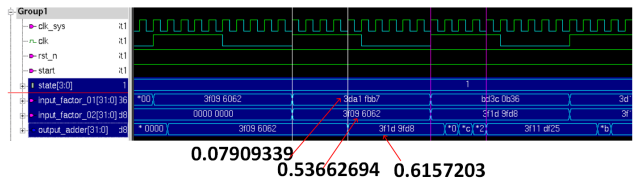

Fig. 3. Simulation of 32 bit floating point addition.

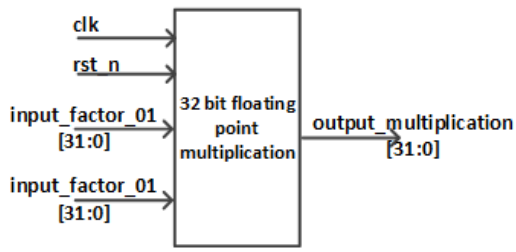

Fig. 4. Interface of 32 bit floating point multiplication.

2. Applied Techniques

2.1. IEEE 754 floating point format

Floating point data format is widely applied to variety of CPUs and FPUs in computer systems shown in [17] and [18] so to enhance accuracy mainly. Floating point number following IEEE 754 standard experiences total of 32 bit comprising of 1 sign bit, 8 exponent bits and 23 mantissa bits as Fig.1 In this work, such format is applied for all mathematical expressions such as addition as well as multiplication. From Fig.2 to Fig.5, those figures present interfaces of 32 bit floating point addition and multiplication and corresponding waveforms collected by experiment simulation.

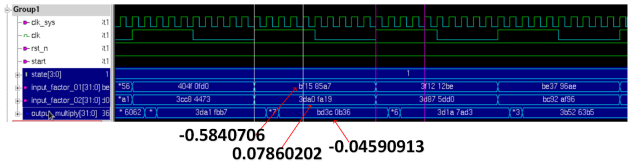

Fig. 5. Simulation of 32 bit floating point multiplication.

As regards addition and multiplication architecture, a 23 bit binary carry-save adder (CSA) is also required to perform 23 mantissa bit addition, while Booth algorithm is applied for calculating 23 bit multiplication instead of normally array multiplication. Based on applied IEEE 754 floating point format, the high accuracy is ensured with a wide range of real numbers.

2.2. Booth algorithm

Booth algorithm invented by Andrew Donald Booth is a multiplication algorithm for large signed binary numbers so to decrease calculation steps as well as enhance speech of calculation. As regards multiplication of real numbers basing on IEEE 754 floating point format, if normal array multiplication is applied for 23 mantissa bits, it is fact that issue of critical path through synthesis step of ASIC design flow is inevitable and produces negative effect for expecting high frequency. Indeed, Booth algorithm presented by Puneet Paruthi [19] or Shaifali [20] confirmed the high performance compared with array multiplication. Moreover, together with Gokul Govindus survey, it also confirmed the interpretation of Booth algorithm for matrix multiplication application on FPGA based architecture [21]. As a result, Booth algorithm is approached in this work to enhance speech and to achieve maximum frequency as well as possible.

2.3. ASIC design flow

In order to achieve ANN architecture adapting to industrial standards, complete design followed ASIC design flow is presented in Fig.6. Based on such figure, all steps are supported by Synopsys tools and Verilog hardware language is utilized to design at register transfer level (RTL). However, pre-layout static timing analysis (Pre-layout STA) is not called mainly because design for test (DFT) is not added into proposed design. Besides, complete design with different levels such as RTL design, pre-layout gate level netlist and post-layout gate level netlist is always verified to ensure matching functions and expected frequency correctly. The target library accompanying ASIC design flow applied for ANN architecture is 130nm. Noticeably, at RTL verification phase, not only is VCS tool used to simulate but practical ANN architecture on FPGA (Cyclone II-Altera) is also evaluated to confirm matching functions together.

3. Proposed Architecture

3.1. Single neuron architecture

Based on electrical pulse behavior of biological neuron, the most popular mathematical model for an artificial

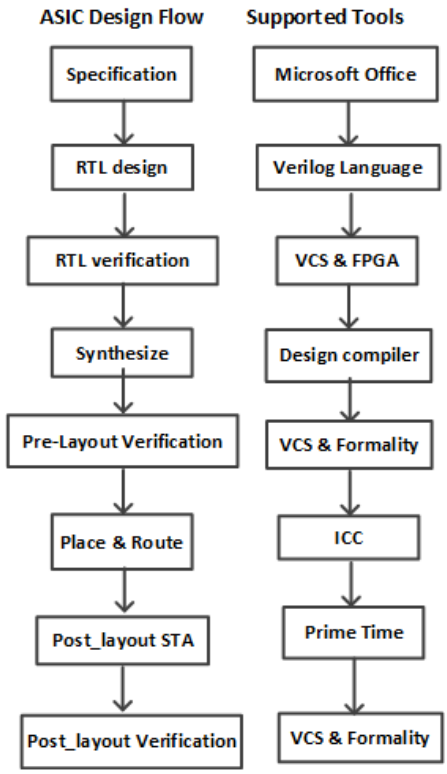

Fig. 6. ASIC design flow and supported Synopsys tools.

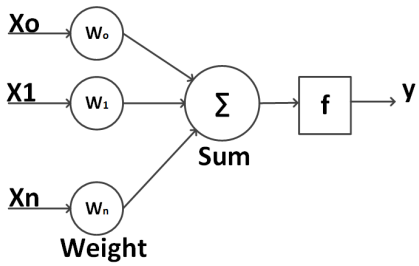

Fig. 7. Artificial neural mathematical model.

neuron widely approached in almost pattern recognition machines is identified as Fig. 7, where W_{ki} and X are weight coefficients and input vectors known as extracted features of sound in speech recognition application or features in other ones, f is the activation function, and y_{ki} is the output of single neuron i at layer k. The relation between inputs and output is illustrated in (3)

$$y_k = f(W_{ki} \times X) \tag{3}$$

$$with \quad X = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} ; \quad W_{ki} = \begin{bmatrix} w_0 \\ w_1 \\ \dots \\ w_n \end{bmatrix}$$

Based on mathematic model, an ANN includes many nodes known as single neuron, which comprises simple components named as Weight Sum (WS) and Activation Function (AF) structures. Particularly, total sum of all multiplications of input vectors and weight coefficients is executed by WS structure, while activation function is calculated on AF structure as Fig.8. In

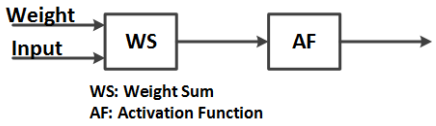

Fig. 8. Single neuron architecture.

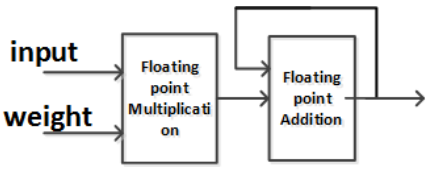

Fig. 9. WS architecture.

WS structure, combination between one floating point addition and one floating point multiplication is presented in Fig.9 to reduce total of resource. As regards active functions, currently Sigmoid and Tan-sigmoid functions as (4) and (5), role of AF, are made effort to transform to linear functions to be able to implement hardware feasibly.

$$f(x) = \frac{1}{1+e^{-x}} \tag{4}$$

$$f(x) = \frac{2}{1+e^{-2x}} - 1 \tag{5}$$

Normally, in order to implement hardware architecture of these non-linear functions, fixed-point format and approximated methods are applied to reduce the area of AF designs but still to adapt to low error in comparison to original function. However, the critical issue is that while AF structure is only triggered one time per single neuron, WS structure requires an enormous additions and multiplications if single neuron receives a hundred of inputs. Consequently, high accuracy ANN applied in speech recognition is negatively affected by using fixed-point technique. Moreover, in larger ANN with many layers, it is vital that AF is called many times, which results in low measurement uncertainty. Indeed, the error of previous layer will be propagated to next layers and increase such error in final results inevitably.

In this work, an innovation of Amins method applying combination between IEEE 754 floating point data format and Booth algorithm is proposed to satisfy not only the high accuracy requirement but also speed of calculation. More particularly, the Sigmoid is approximated to linear function ($y = ax + b$) comprising of multiplication and addition with real number as (6).

$$f(x) = \begin{cases} 1 & (x \geq 5) \\ 0.0141217245x + 0.924396007 & (3.4 \leq x < 5) \\ 0.04963536x + 0.800778912 & (2.3 \leq x < 3.6) \\ 0.136783431x + 0.606760369 & (1 \leq x < 2.3) \\ 0.231058579x + 0.504540715 & (0 \leq x < 1) \\ 1 - f(|x|) & (x < 0) \end{cases} \tag{6}$$

Basing on Sigmoid approximation as (6), this activation function can be transferred to hardware architecture feasibly as Fig.10, in which the floating point multiplication and addition are proposed as basic structures.

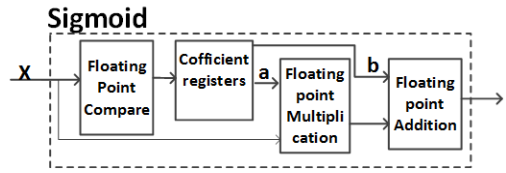

Fig. 10. Sigmoid hardware architecture .

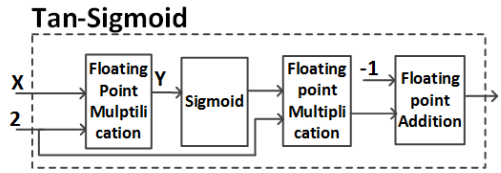

Fig. 11. Tan-Sigmoid hardware architecture.

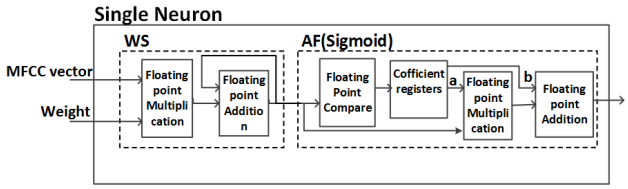

Fig. 12. Single neuron hardware architecture with Sigmoid function.

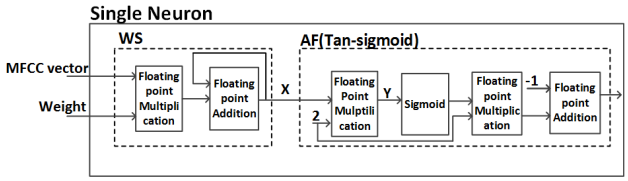

Fig. 13. Single neuron hardware architecture with Tan-sigmoid function.

Besides, coefficient register takes the role of comparison to decide a, b coefficients.

According to the formula of Tan-sigmoid as (5), it is vital that this function can be transferred to hardware architecture as Fig. 4 following an explanation (7).

$$f(x) = \frac{2}{1+e^{-2x}} - 1 = 2 \times \frac{1}{1+e^{-y}} + (-1) \tag{7}$$

As a result, hardware architecture of single neuron is combination between WS structure and AF structure as Fig.12 and Fig.13, respectively.

3.2. Artifical Neural network architecture

Regarding to ANN configuration in Automatic Speech Recognition (ASR) applications, it experiences that almost approaches have referred to software-based systems in order to configure expected ANN dynamically. Moving on to ASIC-based ANN architecture, configurations are always fixed to adapt for exact targets. Therefore, ability of reusing ANN configuration or expanding applications has become one of the critical issues. Towards large ANN configuration in hardware design, a single neuron is called as instance many times to meet timing requirement with parallel process, which results in unexpected large area. Besides, reconfiguration of ANN basing on calling instance needs an

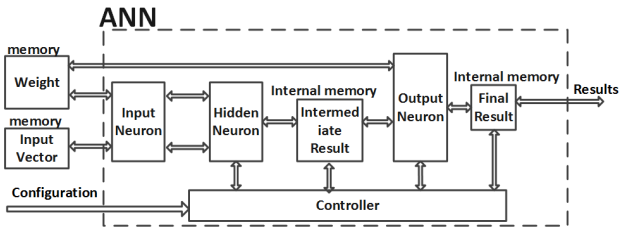

Fig. 14. The proposed Artificial Neural Network configuration.

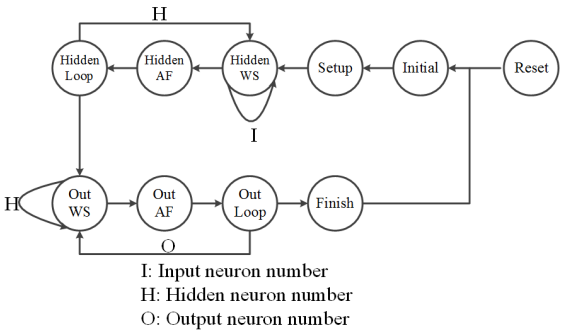

Fig. 15. State machine of proposed Artificial Neural Network .

effective compiler tool together with changing design in Register Transfer Level (RTL) inconveniently. In this research, a suitable solution in which it performs ability of reconfiguration though parameters feasibly while still adapt to resource requirement is proposed. Particularly, three-layer ANN hardware as Fig.14 is introduced with the ability of reconfiguration through input ports. According to Fig.14, there is only one neuron taking the role of all different ones which belong to same layer. Therefore, function of single neuron at one layer will be recalled many times to execute the same function to cover all neurons in same layer instead of calling many instance of single neuron. As the result, during a single neuron takes the role of all neurons in one layer, the output of this neuron should be stored for next calculation in next layers. Hence, internal memories utilized to store intermediate results are integral. Furthermore, a complex controller is also requested to control the reading or writing processes of memories harmoniously. The detailed information describing how to control the ANN is specified as the below Fig.15, TABLE 3 and TABLE 4.

Based on the Fig.15, WS structure at hidden layer is called until finishing all inputs and weight coefficients. Output of WS structure is transferred to AF structure to finish all calculation of one hidden neuron. At hidden loop state, this will check how many single neurons are calculated. If there is not enough H neurons, WS structure at hidden layer is recalled again. These steps are similar to output layer.

Consequently, combination between accuracy single router and complex controller comprising of many loop states not only achieves to build up the large ANN to adapt the targets of ASR system but also confirm the ability of reusing successfully. Because such dynamic ANN can be also reconfigured feasibly through exact

TABLE 3
ANN state machine description.

State	Description
Reset	Reset state.
Initial	Initial the ANN functions.
Setup	Receive and store the setup parameters: Input neuron number (I) Hidden neuron number (H) Out neuron number (O)
Hidden WS	Execute WS in hidden neuron.
Hidden AF	Execute AF in hidden neuron.
Hidden Loop	Check loop cycle at hidden neuron.
Out WS	Execute WS in output neuron.
Out AF	Execute AF in output neuron.
Out Loop	Check loop cycle at output neuron.
Finish	Finish ANN function.

TABLE 4
Transferred condition of ANN state machine.

State	Next State	Conditions
Reset	Initial	Reset is inactive.
Initial	Setup	ANN is triggered.
Setup	Hidden WS	After loading the ANN parameters.
Hidden WS	Hidden AF	After finishing WS at hidden neuron.
Hidden AF	Hidden Loop	After finishing AF at hidden neuron.
Hidden Loop	Hidden WS	If all neurons in hidden layer have not calculated yet.
Hidden Loop	Out WS	After finishing all neurons in hidden layer.
Out NWS	Out AF	After finishing WS at output neuron.
Out AF	Out Loop	After finishing AF at output neuron.
Out Loop	Out WS	If all neurons in output layer have not calculated yet.
Out Loop	Finish	After finishing all neurons in output layer
Finish	Initial	After storing all final results in internal memory.

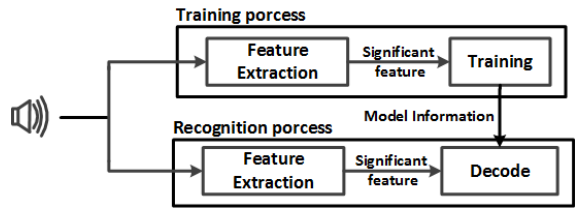

Fig. 16. State machine of proposed Artificial Neural Network .

input ports in certain systems to setup configuration such I, H and O, unchanging design at RTL phase make designer flexibly and promote ability of reusing in industrial environment.

3.3. Complete speech recognition system

The description given by the below Fig.16 illustrates a typical speech recognition scheme comprising of training and recognition processes respectively. To be able to develop high performance speech recognition system

TABLE 5
Mel Frequency Cepstral Coefficient configuration.

The sample number per one frame	160
Overlap proportion	50%
FFT technique	256
Cepstrum coefficient number	12
Energy coefficient number	1
Total members of MFCC vector	26

successfully, popular algorithms such Artificial Neural Network (ANN), Hidden Markov Model (HMM), Dynamic Time Wrapping (DTW), or Vector Quantity (VQ) significantly affecting not only training process but also recognition process are proposed. However, ANN is applied in this work, which results in choosing a popular training algorithm named Back Propagation. It is vital that almost researches have approached training process on software-based systems such performance computers mainly because of long time requirement and complex training algorithms as shown in [1] and [2]. In this work, the training process is also implemented on software by Matlab in advance.

One of considerable aspects relating to the performance of ASR system is applied feature extracting method. According to Fig.16, before solving in decode or training structure, sound sample is pre-solved by feature extraction structure to obtain the most particular features. At the present, there are many solutions for feature extraction task such as Principal Component Analysis (PCA), Linear Prediction Coefficients (LPC), or Mel-Frequency Cepstrum Coefficients (MFCC) applying to speech recognition machines; nevertheless, MFCC applied in this paper has been more popular than others. As a result, this work presents a MFCC configuration as TABLE 5, in which such parameters affecting quality of feature extraction seriously are detailed.

Concerning to combination between MFCC and ANN, there is an issue that total MFCC vectors are too large to configure the input neurons of input layer. Furthermore, MFCC vector number is dynamic and depended on the length of the sound sample. Own investigation has acknowledged that an average of four MFCC vectors corresponding to 104 input neurons of input layer confirmed the higher probability compared with other MFCC vector configurations.

In order to estimate the performance of proposed ANN hardware architecture as well as complete ASR system, a SOPC configuration built on Quatus II tool of Altera company integrates proposed ANN hardware. From such circumstances, the below Fig. 17 describes proposed SOPC system in detail.

According to Fig.17, an ASR system is built on SOPC utility, in which the MFCC structure is implemented by software on Nios CPU core while recognition core approaching ANN is implemented by Verilog language named ANN decoder. First and foremost, set of weight coefficients obtained from training process on Matlab is loaded in to internal memory through Control Panel utility supported by Quatus II.

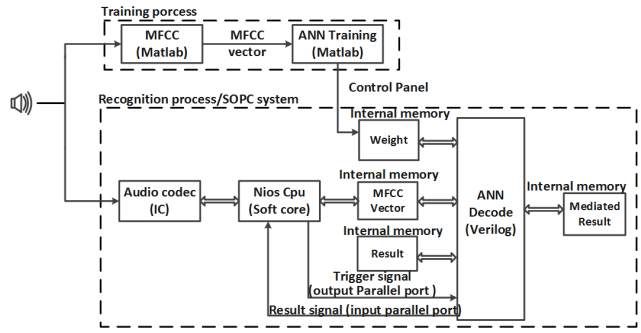

Fig. 17. The proposed speech recognition system.

TABLE 6
Performances of approximated methods

Activation Function	Maximum Error (E_{max})†	Mean Error (E_{eva})†
M.Jame [6]	0.0822	0.0596
Zhang [5]	0.0215	0.0076
Allipi [3]	0.0189	0.0087
Plan [4]	0.0189	0.0058
Proposed Sigmoid Func.	**0.0124**	**0.0023**

† E_{eva} and E_{max} are mean absolute error (MEA) and maximum absolute error, respectively

As regards real-time recognition process implemented on SOPC system, an audio codec is integrated to record human sound from micro phone. Next, the sound sample is filtered to reduce the sample number to collect the value samples for recognition process before extracting particular features by MFCC method. In order to improve efficiency in cutting silence signals, a combination between Zero-Crossing Rate (ZCR) and Zero Threshold Energy (ZTE) is proposed. After MFCC vectors obtained from feature extracting process are stored to internal memory, the ANN decoder is triggered to begin the recognition task. During time of decoding, the MFCC vectors as well as weight values are read from internal memories and transferred to ANN decoder block until to finish process. Finally, a feedback from ANN decoder containing recognition result is transferred to Nios CPU to confirm final result to user though LCD device.

4. Experiment Results

4.1. Approximated method and single neuron

First and foremost, the accuracy of proposed approximate Sigmoid and Tan-Sigmoid functions are compared to other methods. Because of using floating-point format for all operators conducted in NWS structures and AF structures in complete ANN hardware, the description given by the TABLE 7 confirms the highest accuracy through maximum error E_{max} and mean absolute error E_{ave}. Nonetheless, it is also experienced disadvantage of resource in comparison to others as TABLE 8.

Although the results from TABLE 6 present low mean absolute errors for all solutions, only AF is performed. Using fix format towards weight coefficients and input vectors poses a risk to be able to achieve low final error not only for one neuron but also for complete

TABLE 7
Performances of approximated methods

Activation Function	Slices	LUTs	DSPs	BRAMs
M.Jame [6]	-	-	-	-
Zhang [5]	93	86	1	0
Allipi [3]	127	218	0	0
Plan [4]	109	138	0	0
Proposed Sigmoid Func.	**1420**	**2609**	**0**	**0**

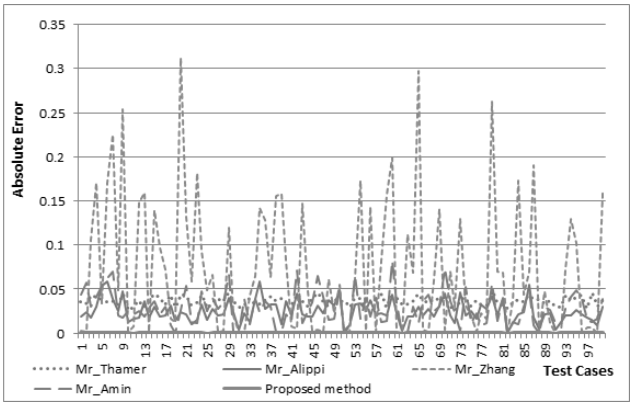

Fig. 18. Absolute error comparisons for single neuron integrating Sigmoid.

ANN because of measurement uncertainty on WS. In order to have comprehensive perspective and deeper comparisons between proposed neuron architecture and other approaches, a random inputs comprising of input vectors and weight coefficients are generated firstly. Thus, such inputs are transferred to different single neuron architectures, in which approximate functions as [3], [4], [5] and [6] are applied to estimate differences compared with the original mathematical models. Obviously, experiments have been implemented with 2000 times of random input generations, in which 104 pairs of weight coefficients and input vectors for every random process are generated. Every 20 times, average of absolute error as (1) is calculated and presented, which results in presented total of 100 values. The range of weight coefficients and input vectors are constrains from 1 to -1 mainly because almost input vectors and weight coefficients which are extracted from MFCC and training process are belonged this range.

Firstly, single neuron, in which Sigmoid function is applied as AF, is verified to confirm measurement uncertainties among 5 approximated methods in comparison to original mathematical model as Fig. 18, Fig 19 and Fig. 20, in which our proposed method is experimented on hardware implementation in RTL design while others are based on the Matlab software..

According to Fig. 18, [5] experience much higher absolute errors with maximum index at 0.31 in comparison to other methods while [3], [4] and [6] show slight difference. In more detail, Fig.19 comparing absolute error between [3], [4] and [5] shows maximum indexes at 0.06, 0.08 and 0.045, respectively. Proposed method witnesses a maximum absolute error at 0.0015, which is nearly equal to original method and much lower than

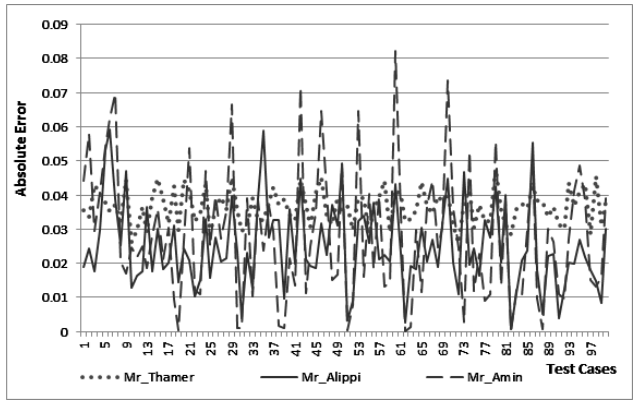

Fig. 19. Absolute error comparison between [5] and proposed method for single neuron integrating Sigmoid.

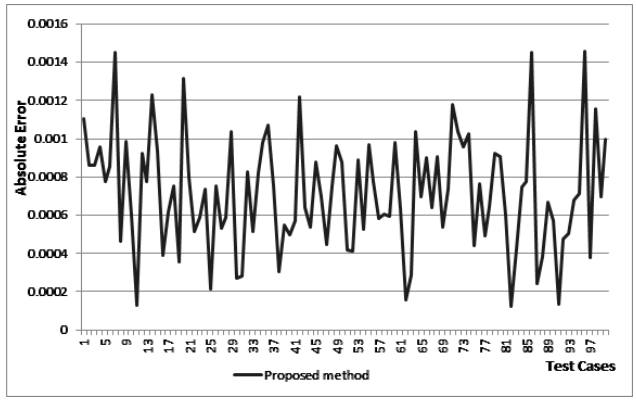

Fig. 20. Absolute error comparison between [4] and proposed method for single neuron integrating Sigmoid.

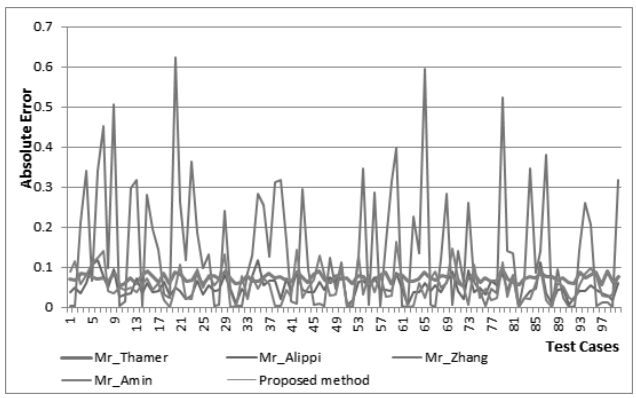

Fig. 21. Absolute error comparisons for single neuron integrating Tan-Sigmoid.

other methods as Fig.20.

Next, Sigmoid function is replaced by Tan-sigmoid function while the previous set of inputs is kept. The compared results are presented as Fig. 21.

Following Fig. 21, the absolute error presents the similar result in comparison to Sigmoid experiments, in which proposed method experiences the least absolute error.

The detailed information given by TABLE 6 and TABLE 8 reveals that although the maximum errors of approximated methods are not different much, absolute

TABLE 8
Mean absolute error comparisons

Single Neuron Integrating	Jame	Allipi	Zhang	Plan	This work
Sigmoid	0.0364	0.0249	0.0719	0.027	**0.0007**
Tan-Sigmoid	0.0729	0.0498	0.1438	0.0558	**0.0014**

TABLE 9
20 Vietnamese discrete words

1	Khong	6	Nam	11	Bat	16	Dung
2	Mot	7	Sau	12	Tat	17	Toi
3	Hai	8	Bay	13	Dong	18	Lui
4	Ba	9	Tam	14	Mo	19	Den
5	Bon	10	Chin	15	Di	20	Quat

TABLE 10
Proposed ANN configuration.

Layer number	3
Input neuron number	104
Hidden neuron number	60
Activation function for hidden neuron	Sigmoid
Output neuron number	20
Activation function for hidden neuron	Tan-Sigmoid

errors through full calculations of single neuron integrated in enormous ANN has experienced considerate number, in which proposed method witnesses the minimum number in comparison to others. It is concluded that high accuracy ANN is concerned not only by AF but also WS.

4.2. Complete Neural Network

In order to perform both neural network and full SOPC, firstly 20 Vietnamese discrete words as TABLE 9 are presented and total of 4000 sound samples are recorded with 200 sound samples per one word. Thus, the first 2000 samples are utilized for training process and the other samples are for recognition process. Noticeably, the words are selected to apply simple control or approaching to blind people

As regards ANN configuration, enormous configurations have been simulated to confirm the final configuration with 104 input neurons, 60 hidden neurons and 20 output neurons being the most suitable as TABLE 9. To be more specific, the input neuron number is simulated with 4 values comprising of 78, 104, 130 and 156, in which these values are average of 3, 4, 5 or 6 MFCC vectors over total MFCC vectors. As regards hidden neurons and output neurons, while arrange from 50 to 300, in which step is set 10, is analyzed to confirm the hidden neuron number, the output neuron number is fixed 20 being suitable to 20 discrete words for our application.

From above configurations, experiments are simulated and results are collected as Fig.22. Basing on results of recognition probability as Fig.22, almost probabilities upper 94% concentrate on the big configurations (156; 50-300; 20) while the lowest indexes experience at

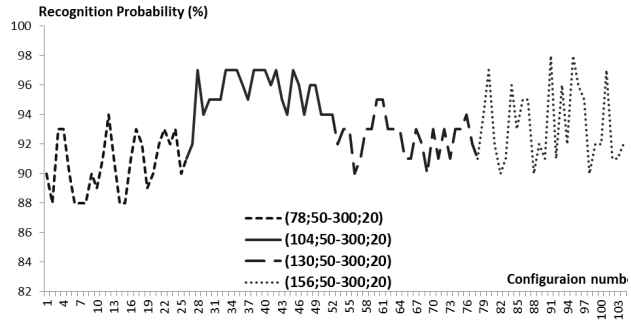

Fig. 22. Recognition Probability of different ANN configurations.

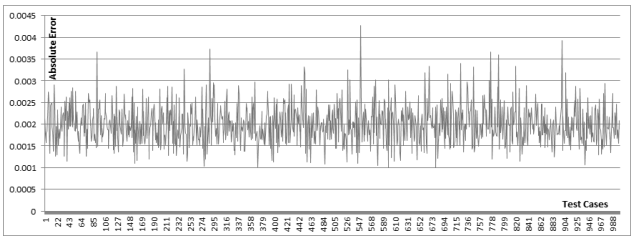

Fig. 23. The mean absolute error between ANN configuration (104; 60; 20) and original mathematical model.

low configurations (78; 50-300; 20) with maximum value at nearly 93%.Although range of configures (104; 50-300; 20) presents intermediate configurations, it confirms the high recognition probability compared with other ranges, maximum and minimum indexes at 97% and 92%, respectively. As a result, final configuration (104; 60; 20) belonging range of (104; 50-300; 20) which confirms the minimum configuration is selected to apply recognition process.

In order to estimate the accuracy of proposed ANN configuration (104; 60; 20) in comparison to mathematical model, set of 1000 test cases, in which 104 pairs of input vector and weight coefficients corresponds every case, is reused to do experiments. According to every case, 20 output values are compared with original mathematical model to obtain the mean absolute error. Obviously, the result survey is shown as Fig.23 to confirm high performance with low measurement uncertainty in comparison between hardware simulation of proposed ANN and original mathematical model only under 0.0035.

Moreover, another statistics concerning utilized internal memories as well as number of clock cycle are also confirmed to estimate trade-off among silicon requirements. Increasing ANN configuration, specially hidden neuron number from 20 to 250 while keeping instance input neuron number and output neuron number at 104 and 20, confirms the rising of both internal memory requirement and clock cycle as below Fig.24 and Fig.25.

According to Fig.24, increasing the total neuron number basing on setup parameter means that larger internal memories are required. As the result, Fig. 24 points out that maximum number of neuron (104; 250; 20) requires approximately 1Mbit. It confirms that abil-

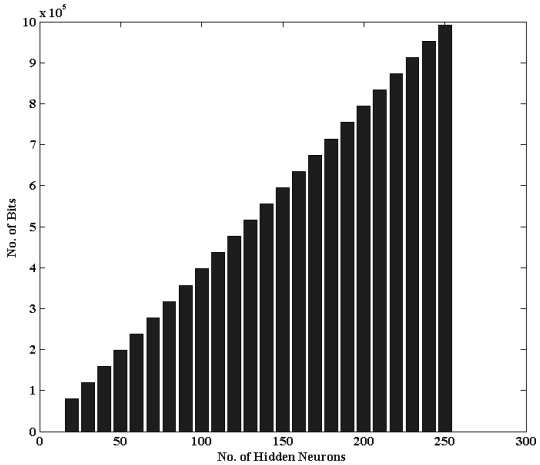

Fig. 24. Experiment results of internal memory capacity.

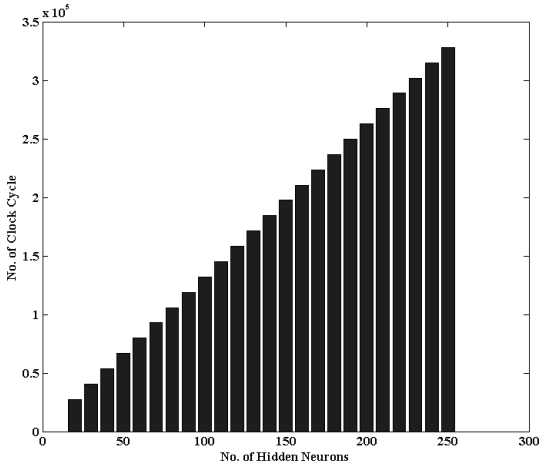

Fig. 25. Experiment results of clock cycle number.

ity of increasing proposed ANN due to current capacity of integrated internal memory up to more than 1Mbit feasibly. The obtained formula of integral memory capacity as (7) from proposed hardware architecture shows that the number of hidden neuron affects such capacity much, which is reasonable to decide the least number but still achieved the high recognition probability where I, H, and O are input neuron number, neuron number in hidden layer, neuron number in output layer, respectively.

$$MemoryCapacity = (I \times H + H \times O) * 32 \quad (8)$$

where I, H, and O are input neuron number, neuron number in hidden layer, neuron number in output layer, respectively.

However, setting input ports to configure ANN and utilizing internal memory techniques face to a trouble of real-time requirement strictly. Actually, the loop of calculation need more time whenever increasing the neuron number, specially the rising of hidden neuron number. So, certain calculation of clock cycle number is investigated to confirm ability of adapting real-time requirement. Following Fig.25, nearly 350000 clock cycle is required to perform largest ANN configuration comprising of 374 single neurons in (104; 250; 20) configuration; However, with obtained maximum frequency at 250MHz, the total consuming time for maximum ANN configuration (104; 250; 20) confirms only 1.4ms, which satisfies real-time requirement feasibly.

Besides, a referred formula as (9) to calculate the clock cycle is also collected from waveform simulations, where I, H, and O are input neuron number, neuron number in hidden layer, neuron number in output layer respectively

$$No.ofClock = 2 + (67 + 10 \times I) \times H + (67 + 10 \times H) \times O \quad (9)$$

Next, complete ANN is synthesized on 130nm technology to confirm the silicon performance. Hence, the description given by the TABLE 11 presents physical results compared with another ANN configuration presented in [13] following ASIC-based design on similar

TABLE 11
ANN physical performances on 130nm technology

	Proposed ANN (104-60-20)	**ANN [13] (16-4-16)**
Library type	**Typical**	Low Power
Frequency(MHz)	**250 MHz**	220 MHz
Combinational area	**100066.821977**	-
Buf/Inv area	**11859.733757**	-
Non-combinational area	**40812.284439**	-
Total cell area	**140879.106416**	-
Gate number (NAND)	**21026**	500

130nm technology. Based on the TABLE 10, the achieved results have confirmed the maximum frequency at 250MHz more than 220 MHz in [13]. Although the number of gates in proposed ANN is larger than one in [13], the proposed architecture presents much large configuration and ability of dynamic setting to achieve larger configuration in comparison to fix configuration (16-4-16) in [13].

4.3. Full SOPC system integrating proposed ANN hardware

Finally, the complete SOPC integrating proposed ANN hardware is built on Quatus II of Altera and experimented on DEII kit integrated Cyclone II FPGA chip family to perform ability of recognition. Obviously, the quality of speech recognition system depends on not only proposed ANN but also MFCC configuration. That is reasonable to implement necessary experiments to make decisions related to MFCC parameters such as overlap rate, FFT number or Mel filter number recommended in TABLE 5. The below detailed information collected from reports of synthesis process on Quatus II not only presents the final synthesized results of SOPC system but also confirms the effective recognition performance with set of 2000 samples for recognition process which is simulated on hardware before.

Based on TABLE 12, it is vital that although proposed ANN hardware is more advantage in timing implement, the final result of SOPC system has expended

TABLE 12
Speech recognition system

Family	Cyclone II
Device	EP2C35F672C6
Total combinational functions	17,466 / 33,216 (53 %)
Dedicated logic registers	13,627 / 33,216 (41 %)
Total logic elements	21,788 / 33,216 (66 %)
Total registers	13695
Total pins	120 / 475 (25 %)
Total virtual pins	0
Total memory bit cells	387,375 / 483,840 (80 %)
Embedded Multiplier 9-bit	53 / 70 (76 %)
Total PLLs	2 / 4 (50 %)
Frequency	50 MHz
Person number	14
Word samples/person	2000
Discrete word number	20
Average of Recognition Probability	95.2 %
Real Time recognition (one sample)	0,8s

TABLE 13
Comparison of speech recognition systems

	Proposed Systems	**SOPC [22]**
Implement method	**SOPC system**	Software system
ANN configuration	**104-60-20**	140-75-40
Feature extracting	**MFCC**	MFCC
Discrete word number	**20**	14
Recognition Probability	**95.2%**	97.98%

TABLE 14
Detailed information of proposed ANN architecture

Layer number	3 (input, hidden, output)
Data format	IEEE 754 floating point data format
Design flow	ASIC-based design
Supported tool chain	Synopsys
Library	130 nm
Using ready library	No
Maximum frequency	250 Mhz
Area	21026 (NAND)
Timing Consumption †	0.1 ms - 0.4 s
Memory Consumption †	0.2 Mbit (7K x 32bit)- 320Mbit (10 M x 32 bit)
Efficiency	Recognition Probability : 95.2 %

† Requirements for configurations from (104:60:20) - (1000;5000;1000)

more time to complete all tasks including MFCC function, audio controller or collect recognition results. This is obviously recognized that the Nios II performance has experienced drawback. To compare with another system as [22], the description presented by TABLE 13 classifies the comparisons in many different aspects. Although [22] confirms a higher index of speech recognition, larger configuration and less word number are experienced in comparison to proposed SOPC

From above experiments, TABLE 14 collects significant information to perform proposed ANN, in which real-time and memory criteria are based on the formulas (8) and (9)

5. Conclusion

In this paper, a real-time speech recognition system integrating a reconfigured three-layer ANN as hardware component is proposed to confirm the effective performance compared with other schemes. The promoted ANN hardware architecture not only solves the accuracy and real-time issues required on handle device but also adapts silicon requirements such high frequency, reused ability as well as resource limitation. The obtained experimental results validated on both FPGA and 130 nm technology also experienced the expanded ability to larger ANN later. As regards future works, a complete hardware-based ASR system is our target to achieve powerful integrated circuit. Moreover, other applications integrating proposed ANN will be also implemented to estimate the ANN hardware performance obviously.

Acknowledgments

This work is funded by the Ministry of Science and Technology, State-level key program, Research for application and development of ITC, code KC.01.23/11-15

References

[1] B. Markovic, J. Galic, S. T. Jovicie, *Application of Neural Networks in Whispered Speech Recognition*, TELFOR, 2012, pp. 728-731.

[2] I. Sahin, I. Koyuncu, *Design and Implementation of Neural Networks Neurons with RadBas, LogSig, and TanSigActivation Functions on FPGA*, Electronics & Electrical Engineering, 2012.

[3] C. Alippi, G. StortiGajani, *Simple approximation of Sigmoidal functions: realistic design of digital neural networks capable of learning*, Proc. IEEE Int. Symp. on Circuits and Systems, 1991, pp. 1505-1508.

[4] H. Amin, *Piecewise linear approximation applied to nonlinear function of a neural network*, Circuits, Devices and Systems, 1997, pp. 313-317.

[5] M. Zhang, *Sigmoid generators for neural computing using piecewise approximations*, IEEE Trans. on Computers, 1996, pp. 1045-1059.

[6] Thamer M. Jame, Ban M. Khammas, *Implementation of a Sigmoid activation function for neural network using FPGA*, Scientific Conference of Al-Mamoon University College, 2012.

[7] Shing-Tai Pan, Chih-Chin Lai, Bo-Yu Tsai, *The implementation of speech recognition systems on fpga-based embedded systems with soc architecture*, Int. Jour. of Innovative Computing, Information and Control, No. 10, 2011.

[8] Khamlich S. Eddine, Hamdoun Abdellatif, Atouf Issam, Madiafi Mohammed, *Serial Hardware Implementation of The MFCC and MLP Architecture on FPGA Circuit*, Int. Jour. of Engineering and Technology (IJET), Vol. 5, No. 4, 2013.

[9] Boussaa Mohamed, Atouf Issam, Atibi Mohamed, Bennis Abdellatif, *Comparison of Hardware and NIOS II Based Software Implementation of MLP on The FPGA Plateform*, Jour. of Theoretical and Applied Information Technology, Vol.72, No.3, 2015.

[10] Dhirajkumar S. Jinde, Samrat S. Thorat, *Neural Network Implementation Using FPGAs*, Int. Jour. of Computer Science and Information Technologies (IJCSIT), 2014, pp. 3431-3433.

[11] Esraa Z. Mohammed and Haitham K. Ali, *Hardware Implementation of Artificial Neural Network Using Field Programmable Gate Array*, Int. Jour. of Computer Theory and Engineering, Vol. 5, No. 5, 2013.

[12] A. Gomperts, A. Ukil, F. Zurfluh, *Development and Implementation of Parameterized FPGA Based General Purpose Neural Networks for Online Applications*, IEEE Trans. on Industrial Informatics, 2011, pp. 79-89.

[13] K. Venkata Ramanaiah and C. Prasanna Raj, *ASIC Implementation of Neural Network Based Image Compression*, Int. Jour. of Computer Theory and Engineering, Vol. 3, No. 4, 2011.

[14] Panca M. Rahardjo, Moch. Rifan dan Nanang Sulistyanto, *The Implementation of Feedforward Backpropagation Algorithm for Digit Handwritten Recognition in a Xilinx Spartan-3*, Journal EECCIS, 2012, pp. 17-21.
[15] Suhap Sahin, Yasar Becerikli, and Suleyman Yazici, *Neural Network Implementation in Hardware Using FPGAs*, ICONIP 2006, pp. 1105-1112.
[16] Hassene Faiedh, Chokri Souani, Kholdoun Torki and Kamel Besbes, *Digital Hardware Implementation of a Neural System Used for Nonlinear Adaptive Prediction*, ICONIP 2006 Jour. of Computer Science, 2006, pp.355-362.
[17] Andrew Thall, *Extended-Precision Floating-Point Numbers for GPU Computation*, ACM SIGGRAPH 52, 2006
[18] Damak A., Krid M., Masmoudi D.S, *Neural Network Based Edge Detection with Pulse Mode Operations and Floating Point Format Precision*, Design and Technology of Integrated Systems in Nanoscale Era, March 2008, page 1-5, 25-27.
[19] Puneet Paruthi, Tanvi Kumar, Himanshu Singh, *Simulation of IEEE 754 Standard Double Precision Multiplier using Booth Techniques*, International Journal of Engineering Research and Applications (IJERA),Vol.2, September- October 2012, pp.1761-1766.
[20] Shaifali, Ms.Sakshi, *Comparison of IEEE-754 Standard Single Precision Floating Point Multipliers*, International Journal of Emerging Trends in Electrical and Electronics (IJETEE), Vol.1, March 2013, pp. 900-904
[21] Gokul Govindu, Ling Zhuo, Seonil Choi, Prasanna, Viktor Prasanna, *Analysis of High-performance Floating-point Arithmetic on FPGAs*, Parallel and Distributed Processing Symposium, April 2004
[22] Dorde T. Grozdic, Branko Markovic, Jovan Galic, Slobodan T. Jovicic, *Application of Neural Networks in Whispered Speech Recognition*, Telfor Journal, Vol. 5, No.2, 2013.

Lam D. Pham was born in VungTau city, Vietnam. He received the Bachelor of Engineering, and Master of Science degree in Electronics-Telecommunication Engineering from Ho Chi Minh City University of Technology in 2009 and 2012, respectively. During 2009-2012, he joined Renensas VietNam Company as system level design engineer. Currently, he is lecturer at Faculty of Electricals-Electronics Engineering, Ho Chi Minh City University of Technology. His research focuses on Network on Chip, Speech Recognizer, IC architecture.

Hieu M. Nguyen received the B.S. degrees in Electronics and Telecommunication Engineering from Ho Chi Minh City University of Technology in 2014. During 2013 and 2014, he joined Integrated Circuit Design Research and Education Center where he studied about Analog and RF integrated circuit design. He also received Award of Best Student in Analog IC Design for the design of 24-Bit Delta Sigma ADC. He presently works as Teaching and Research Assistant at Department of Electronics Engineering, Faculty of ElectricalsElectronics Engineering, Ho Chi Minh City University of Technology. His current research focus is mainly on low power, high speed, high performance analog, mix-signal and RF integrated circuit design

Du N. N. T. Nguyen has been senior student in Electronics and Telecommunication Engineering from Ho Chi Minh City University of Technology currently (HCMUT). During 2012 up to now, he has worked as member of IC design laboratory in Department of Electronics Engineering, HCMUT. His current project concentrates on Artificial Neural Network and real time applications.

Trang Hoang was born in NhaTrang city, Vietnam. He received the Bachelor of Engineering, and Master of Science degree in Electronics-Telecommunication Engineering from Ho Chi Minh City University of Technology in 2002 and 2004, respectively. He received the Ph.D. degree in Microelectronics- MEMS from CEA-LETI and University Joseph Fourier, France, in 2009. From 2009-2010, he did the postdoctorate research in Orange Lab- France Telecom. Since 2010, he is lecturer at Faculty of Electricals-Electronics Engineering, Ho Chi Minh City University of Technology. His field of research interest is in the domain of FPGA implementation, Speech Recognizer, IC architecture, MEMS, fabrication.