ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2 Issue Special 1 Jan-March 2014
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

Implementation of Hyperbolic Tangent Activation Function in VLSI

[I]P. Motcha Mary Rathi Pushpa, [II]K. Manimala
[I]M.E VLSI Design Student, Dr. SACOE, Tiruchendur, Tamil Nadu, India
[II]Professor, Dept. of ECE, Dr. SACOE, Tiruchendur, Tamil Nadu, India

Abstract
This work presents the implementation of Artificial Neural Network (ANN) chip, which can be designed to implement certain functions. Usually designing of neural networks is done using software tools in the computer system. The neural networks designed off-line are fixed and are very slow especially for portable real time applications. In order to overcome this disadvantage, neural network is to be implemented on-chip with the neural network. Efficient implementation of the activation function is important in the hardware design of artificial neural networks. In this project, a simple and efficient approximation for digital hardware implementation of the hyperbolic tangent function is presented. The proposed technique employs a hybrid method which is the combination of linear method and bit level mapping and found to perform better than other conventional method.

Keywords
Hyperbolic tangent, neural networks, nonlinear activation function, hybrid method VLSI implementation

I. Introduction

Artificial Neural networks (ANNs) have a wide range of applications in analog and digital signal processing. ANNs were typically implemented only in software until recently, wherein their hardware implementation has become important [1]; this is largely due to the performance gains of hardware systems compared to software implementations.

Special care must be taken into account when designing every computational element in ANN hardware design. Hardware implementation of neural networks has been used in applications such as pattern recognition [2], optical character recognition [3], test of analog circuits [4], real-time surface discrimination [5].

The main building blocks needed for hardware implementation of neural networks are multiplier, adder, and nonlinear activation function. A lot of research has been done in digital implementation of multipliers and adders, which can be readily used leaving the nonlinear activation function which is used at the output of every neuron as the most complex building block. To implement the neuron, various nonlinear activation functions, such as threshold, sigmoid, and hyperbolic tangent can be used. Hyperbolic tangent and sigmoid are mostly used because their differentiable nature makes them compatible with back propagation algorithm.

The hyperbolic tangent function is given by equation (1)

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad (1)$$

Because of the exponentiation and division terms present in hyperbolic tangent activation function, it is hard to realize the hardware implementation of these functions directly.

To solve the implementation problem, approximation methods are generally applied. These methods are based on piecewise linear approximation (PWL) where the function is divided into segments and linear approximations used in each segment [6], piecewise nonlinear approximation is similar to the PWL method with the difference that nonlinear approximation is used in each segment [7], lookup table (LUT) in which input range is divided to equal sub-ranges and each sub-range is approximated by a value stored in LUT and is used in [8] to implement the hyperbolic tangent, bit-level mapping approximates output based on a direct bit-level mapping of input. This method can be implemented using purely combinational circuits, and is used in [9] to implement the sigmoid function, and hybrid methods use a combination of the aforementioned methods. Examples include [10], which have used a combination of PWL and LUT methods to implement hyperbolic tangent activation function.

The approximation error present in all methods affects the neural network performance. Research performed in [11] shows that the nonlinear activation function implementation with higher accuracy improves the learning and generalization capabilities of neural networks. However, implementations with higher accuracy require more silicon area and decrease the network operation speed. Therefore, having nonlinear activation function hardware structures with lower area and higher speed for a specified accuracy becomes a key issue.

In this paper, a new hybrid architecture, which is based on linear approximation in combination with bit-level mapping is proposed. The proposed approximation scheme divides the input range to three different regions using different strategy in each region. A mathematical analysis of the proposed approximation scheme in each region is provided.

The rest of this paper is organized as follows. The hyperbolic tangent approximation scheme is discussed in 1. The proposed structure based on the mathematical analysis done is explained in Section II. Results are discussed in section III. Finally, conclusions are drawn in Section IV.

A. Hyperbolic Tangent Approximation Scheme

The mathematical analysis of the approximation scheme used for hardware implementation of hyperbolic tangent function is provided. The mathematical analysis in this and the following sections uses the basic properties of hyperbolic tangent function.

Hyperbolic tangent is an odd function

$$\tanh(-x) = -\tanh(x) \quad (2)$$

Using this property, only the absolute value of input is processed and the input sign is directly passed to the output.

The Taylor series expansion of hyperbolic tangent is as follows:

$$\tanh(x) = x - x^3/3 + 2x^5/15 - 17x^7/315 + \quad (3)$$

For small values of x, the higher order terms become small and can be ignored. Therefore, the hyperbolic tangent passes the small input values to output

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

$$\lim \tanh(x) = x \qquad x \to 0 \qquad (4)$$

The output variation for large values of input is low

$$\frac{d\tanh(x)}{dx} = 0 \qquad x \to \infty \qquad (5)$$

Considering the two last properties, input range is divided to three regions. Region I in which the output is approximately equal to input is named pass region while because of low variation of output in region III, it is named saturation region. Region II includes the rest of input range, named processing region. Different regions of hyperbolic tangent function are shown in Fig.1.

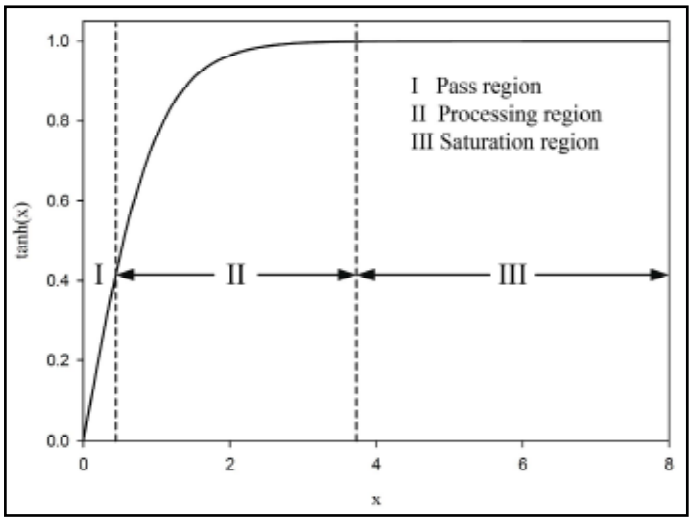

Fig. 1: Different regions of hyperbolic tangent function

(i). Output Approximation in the Pass Region

The input and output of hyperbolic tangent function are represented as signed-magnitude notation. Therefore, considering the first basic property of the hyperbolic tangent function, input sign bit is directly passed to the output sign bit and only the absolute value of input is processed.

In the pass region, output is approximated by passing the input to the output, which means that a linear approximation is used in this region. The inputs in the pass region include the values close to the origin, which are represented by fractional part of the input.

The output is given by equation (6).

$$y_k = \begin{cases} x_k & -N_f \le k \le -1 \\ 0, & -N_{out} \le k < -N_f \end{cases} \qquad (6)$$

where N_{out} is the number of bits used for representation of absolute value of output, N_f is number of bits for fractional part of the input. Based on (6), the fractional part of input is shifted to left by $N_{out} - N_f$ bits and then is passed to the output.

(ii). Output Approximation in the Processing Region

Processing region is the region between the pass and saturation region.Before going through the proposed approximation scheme in this region, a new parameter named N_{one} is introduced. This parameter is an indicator of position of the first occurrence of one in binary input, when scanned from left. Therefore, based on this parameter, the input range can be shown in equation (7)

$$2^{N_{one}} \le x < 2^{N_{one}+1} \qquad (7)$$

This input range is divided into equal sub-ranges. The number of these sub-ranges, N, is based on the equation (8)

$$N = 2^{(N_{one}+N_f-i)}$$

$$0 \le i \le N_{one} + N_f. \qquad (8)$$

Based on the value of N, sub-ranges within the input range are given in equation (9)

$$2^{N_{one}(1+\frac{j}{N})} \le x < 2^{N_{one}(1+\frac{j+1}{N})}$$

$$0 \le j < N. \qquad (9)$$

To have an approximation value close to all outputs corresponding to an input sub-range, the average value of outputs is considered as the approximation value as given in equation (10)

$$\sum_{k=0}^{\frac{2^{N_{one}+N_f}}{N}} \frac{\tanh\left(2^{N_{one}\left(1+\frac{j}{N}\right)} + k \times 2^{-N_f}\right)}{\frac{2^{N_{one}+N_f}}{N}} \qquad (10)$$

The total number of sub-ranges is found based on the fact that the difference between all values inside a sub-range and the approximation value found using (10) should be less than maximum allowable approximation error in this region.

For inputs in the processing region, a bit-level input mapping is required. The number of bit-level mapping blocks required is equal to the number of input range sin this region. For each input range in the processing region, $\log^N_2$ bits after N_{one} bit of input should be mapped to output bits using the bit-level mapping. Using $\log^N_2$bits after N_{one} bit covers all sub-ranges. The output of each sub-range is calculated using (10) The bit-level mapping can be implemented using a combinational circuit.

(iii). Output Approximation in the saturation Region

The hyperbolic tangent function reaches its maximum value in the saturation region, while at the same time output variation in this region is low. Therefore, all output values in this region are approximated by the maximum value representable by the output bits and is equal to $1 - 2^{N_{out}}$.

II. Proposed Structure

Block diagram of the proposed structure is shown in Fig. 2.The hardware is composed of two main blocks, including hyperbolic tangent approximation and output assignment.

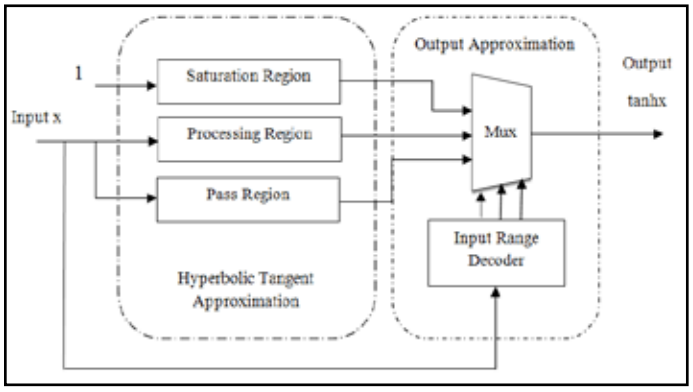

Fig. 2: Block Diagram of the Proposed Structure

A. Hyperbolic Tangent Approximation

This block is composed of three main blocks to approximate the hyperbolic tangent function in all three regions, including

saturation, processing, and pass region. General arithmetic operations in each region can be described as follows.

1. Pass Region
In pass region, fractional part of input is passed to the output. Based on (6), a shift to left by Nout−N f bits before passing the input to output is required.

2. Processing Region
For inputs in the processing region, a bit-level input mapping is required. The number of bit-level mapping blocks required is equal to the number of input range sin this region. For each input range in the processing region, logN2 bits after None bit of input should be mapped to output bits using the bit-level mapping. Using logN2bits after None bit covers all sub-ranges. The output of each sub-range is calculated using (10). The bit-level mapping can be implemented using a combinational circuit.

3. Saturation Region Approximation
In saturation region, hyperbolic tangent function is approximated by the maximum value representable by output bits, and can be realized by setting all output bits to one.

B. Output Assignment
The input range decoder detects the None introduced previously, which is set by input range and region of operation, respectively. Depending on the input range, a multiplexer issued to obtain the appropriate output value.

III. Result and Discussion
The hyperbolic tangent activation function implemented in VLSI using hybrid method is divided into three regions pass region, processing region and the saturation region.
The pass region is approximated using linear approximation and the result is shown in Fig. 3.

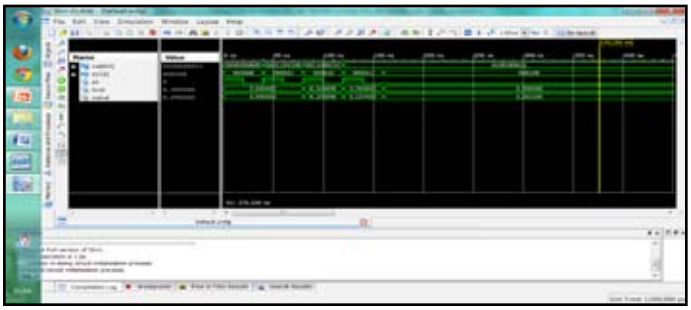

Fig. 3: Simulation result of pass region

The processing region approximated using bit level mapping is shown in Fig. 4.

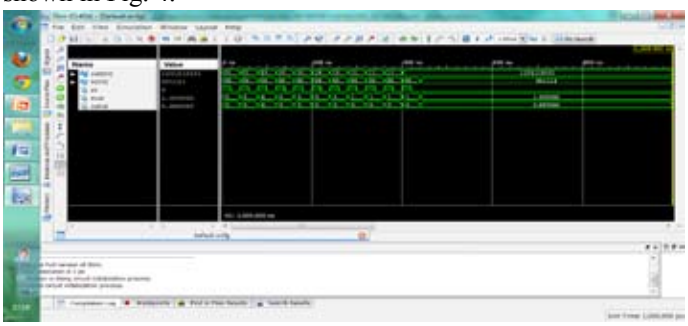

Fig. 4: Simulation result of processing region

The saturation region result which indicated low variation in output is shown in Fig. 5.

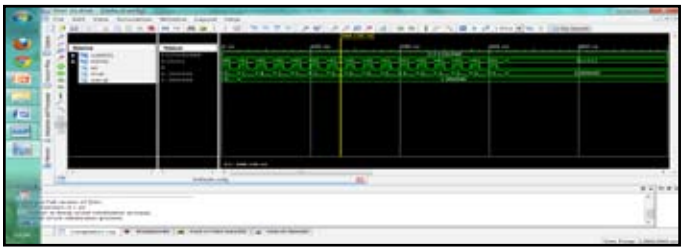

Fig. 5: Simulation result of saturation region

The simulation result of hyperbolic tangent approximation is shown in figure 6.

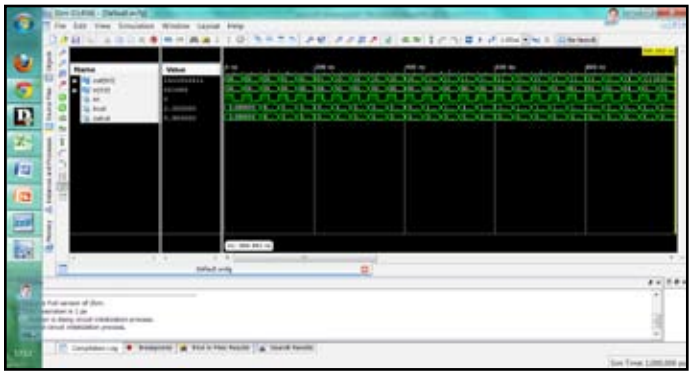

Fig. 6: Simulation result of hyperbolic tangent approximation

IV. Conclusion
The proposed approximation scheme is based on a mathematical analysis considering maximum allowable error as a design parameter. Based on the proposed approximation scheme, hybrid method for hardware implementation of hyperbolic tangent activation function was presented. Hybrid method is the combination of linear approximation and bit level mapping. The proposed structure is used for implementing a back propagation network.

V. AcknowledgEment
P. MOTCHA MARY RATHI PUSHPA would like to thank Dr. K. MANIMALA HOD of ECE Department, Dr. SACOE, Tiruchendur. who is guiding throughout the project and supporting me in giving technical ideas about the paper and motivating me to complete the work effectively and successfully.

References
[1] *A.R. Omondi, and J.C. Rajapakse, "Neural networks in FPGAs". Proc. 9th Int. Conf. on Neural Information Processing (ICONIP), Singapore, 18-22 November 2002, vol. 2, pp. 954-959.*
[2] *V. Koch and R. Goodman, "Analogy VLSI neural network with digital perturbative learning," IEEE Trans. Circuits Syst. II, Exp. Briefs vol. 49,no. 5, pp. 359–368, May 2002.*
[3] *D. Kim, H. Kim, H. Kim, G. Han, and D. Chung, "A SIMD neural networkprocessor for image processing," Adman. Neural Newt., vol. 3497, pp. 665–672, Jun. 2005.*
[4] *D. Maluku, H.-G. Stratigopoulos, and Y. Makris, "An analog VLSI multilayer perceptron and its application toward built-in self-test in analog circuits," in Proc. IEEE 16th Int. Online Test. Sump. Conf., Jul. 2010, pp. 71–76.*
[5] *L. Gated, H. Tap-Beteille, and M. Lescure, "Real-time surface*

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

discrimination using an analog neural network implemented in a phase-shift laserrangefinder," IEEE Sensors J., vol. 7, no. 10, pp. 1381–1387, Oct. 2007.

[6] A. Armato, L. Fanucci, E. Scilingo, and D. D. Rossi, "Low-error digital hardware implementation of artificial neuron activation functions and their derivative," Microprocess. Microsyst., vol. 35, no. 6, pp. 557–567, 2011.

[7] M. Zhang, S. Vassiliadis, and J. Delgado-Frias, "Sigmoid generators for neural computing using piecewise approximations," IEEE Trans. Comput., vol. 45, no. 9, pp. 1045–1049, Sep. 1996.

[8] K. Leboeuf, A. Namin, R. Muscedere, H. Wu, and M. Ahmadi, "High speed VLSI implementation of the hyperbolic tangent sigmoid function," in Proc. 3rd Int. Conf. Converg. Hybrid Inf. Technol. Nov. 2008, pp. 1070–1073

[9] M. Tommiska, "Efficient digital implementation of the sigmoid function for reprogrammable logic," IEEE Comput. Digital Tech., vol. 150, no. 6 pp. 403–411, Nov. 2003.

[10] A. Namin, K. Leboeuf, R. Muscedere, H. Wu, and M. Ahmadi, "Efficient hardware implementation of the hyperbolic tangent sigmoid function,"in Proc. IEEE Int. Sump. Circuits Syst., May 2009, pp. 2117–2120

[11] K. Basterretxea, J. Tarela, I. D. Campo, and G. Bosque, "An experimentalstudy on nonlinear function computation for neural/fuzzy hardwaredesign," IEEE Trans. Neural Newt., vol. 18, no. 1, pp. 266–283, Jan. 2007.

P. Motcha Mary Rathi Pushpa has done her B.E in Electronics and Communication Engineering from Francis Xavier Engineering College, Tirunelveli, India in the year 2012 and currently pursuing her M.E in VLSI Design from Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, India. Her areas of interest include VLSI Design, Digital Electronics and Neural Networks.

She has 17 years teaching experience. She is currently an Professor at the Electronics and Communication Engineering Department of Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, India. She has completed PhD in the area of Data Mining.