

# FPGA MODELLING OF NEURON FOR FUTURE ARTIFICIAL INTELLIGENCE APPLICATIONS

**Korani Ravinder<sup>1</sup>, Hajera Hasan<sup>2</sup>, Imthiazunnisa Begum<sup>3</sup>, Dr P Chandra Sekhar Reddy<sup>4</sup>**

M-Tech Phd, Assistant Professor, ECE Department, VIFCET JNTU Hyderabad<sup>1</sup>

M-Tech Student [VLSI Design] ECE Department, VIFCET JNTU Hyderabad<sup>2</sup>

M-Tech H.O.D, ECE Department, VIFCET JNTU Hyderabad<sup>3</sup>

ME M-Tech, Phd Co-ordinator JNTU Hyderabad<sup>4</sup>

**Abstract:** This paper presents a digital design of neuron architecture on field-programmable gate array (FPGA). The objective of this project is to translate data from electrochemical sensor signals and process the data with neuron structure on digital hardware. The hardware realization of neural network requires investigation of many design issues relating to signal interfacing and design of a single neuron. Analysis focuses on effect of digital design decisions such as module architecture towards data accuracy and delay. The work touches on analogue to digital interfacing, data structure and digital module design that includes adder, multiplier and multiplier accumulator (MAC). A major component of the algorithm is the design of the activation function. The chosen activation function is the hyperbolic tangent which is approximated by Taylor Series expansion. The neuron is evaluated on an Altera DE2-70 FPGA. The performances are evaluated in terms of functionality, usage of resources and timing analysis. For the data structure, it was demonstrated that increasing the fractional bits will increase the precision. The neuron functionality was demonstrated on digital platform. It was found that less delay was produced by using Carry Look Ahead design compared to Ripple Carry Adder by 25% in the MAC performance.

**Keywords:** component; formatting; style; styling; insert (key words)

## I. INTRODUCTION

Electrochemical sensors are often used to determine concentrations of various analytes in testing samples such as fluids and dissolved solid materials. Electrochemical sensors are frequently used in occupational safety, medical engineering, process measuring engineering, environmental analysis [1]. ANN is known to be able to improve electrochemical sensor signal interpretation [2]. In general, hardware realization requires a good compromise between accuracy and complexity of the processing units to allow a low cost effective device [3]. This paper describes a system realization of translating data from electrochemical

sensor for neuron to process on FPGA. Analysis on the effect of different digital module architecture towards neuron design is investigated. The structure of a neuron is split into various sub blocks and these blocks are implemented individually first and then they are integrated to form the entire neuron. The digital platform is Field Programmable Grid Array (FPGA). The approach for this project can be represented in block diagram as shown in the Fig.1. The key issue in designing this system is modular design for re-configurability. The first issue is to convert the signal from an analog to a digital form, by sampling it using an analog-to-digital converter (ADC),



which turns the analog signal into a stream of numbers. The next module is the design of mathematical operation. This includes issues relating to data structure, design of Multiplier Accumulator (MAC) and activation function implementation. The final module is displaying the result from the data that have been accumulating by the neuron. A 7-segment driver is included to enable reading and displaying the data that comes out from neuron architecture.

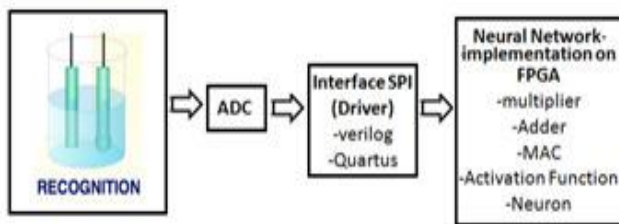


Fig.1. General flow of the project system linking applied chemical sensor to digital processing

## II. DESIGN AND METHODOLOGYS

This section presents the design of the sub-modules in implementing Fig 1. This covers the interfacing issues such as analog to digital implementation, data structure and the neuron architecture topology.

### A. Analog to Digital Interfacing

In this project the 10 bit ADC chip-MCP3001 were used to convert analog signal from electrochemical sensor to digital. The Microchip Technology Inc. MCP3001 is a successive approximation 10-bit A/D converter with onboard sample and hold circuitry. The device provides a single pseudo-differential input. Communication with the device is done using a simple serial interface compatible with the SPI protocol. SPI is an interface that allows one chip to communicate with one or more other chips and in this case is ADC chip-MCP3001 with the FPGA-Altera DE2-70 board. The SPI algorithm is required to be implemented in hardware description language (HDL) on FPGA. Fig.2. demonstrates SPI interfacing that allows one chip to communicate with one or more other chips. As shown in the figure above the wires are called SCK, MOSI, MISO and SSEL, and one of

the chip is called the SPI master, while the other the SPI slave. A clock is generated by the master, and one bit of data is transferred each time the clock toggles. Data is serialized before being transmitted, so that it fits on a single wire. There are two wires for data, one for each direction. The master and slave know beforehand the details of the communication (bit order, length of data words exchanged, etc...). The master is the one who initiates communication. Because SPI is synchronous and full-duplex, every time the 2012 IEEE Symposium on Humanities, Science and Engineering Research clock toggles, two bits are actually transmitted (one in each direction). In term of performance, SPI can easily achieve a few Mbps (mega-bits-per-seconds) [4]. For this module, the approach taken is hardware implementation of existing technique, tailored to 10-bit environment.

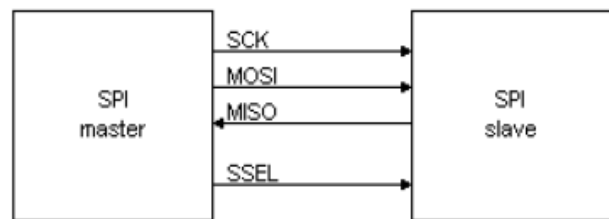


Fig.2. SPI Interfacing applied to signal

### B. Digital Design: Data Structure and Modules for Neuron

In this section, there are 2 major parts: data structure and digital modules for neuron design on FPGA. For design tools, Modelsim to simulate the design at multiple stages throughout the design process and Quartus to program the board are used. Generally, a data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently. Data structures are generally based on the ability of a computer/chip to fetch and store data at any place in its memory, specified by an address that can be manipulated by the program. For this project, the data computed from ADC will be converted into fixed-point number representation. Fixed-point DSPs use 2's complement fixed-point numbers in different Q formats. Among the major issues in data structure is the conversion



technique of fixed-point number from a Q format to an integer value so that it can be stored in memory and recognized by simulator. It is also required to keep track of the position of the binary point when manipulating fixedpoint numbers in writing verilog codes.

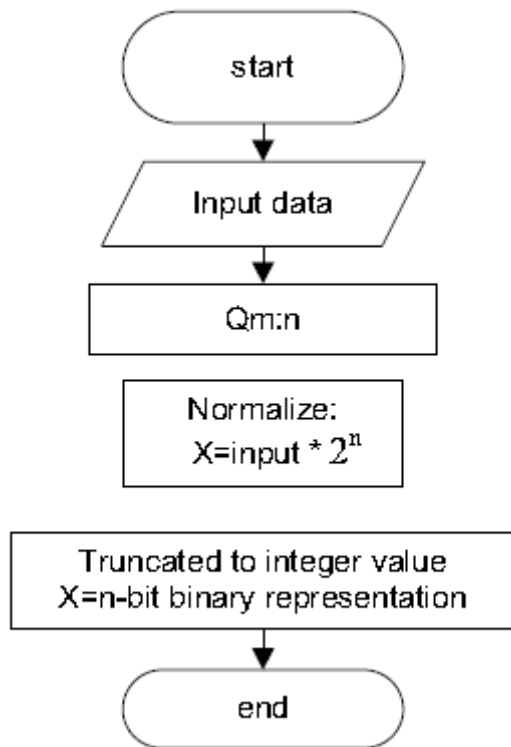


Fig.3. Flow of fixed point arithmetic conversion

The DSP (Digital Signal Processing) flows throughout the conversion to Q format representation are shown in the Fig. 3. As shown in the flowchart, a fractional number is converted to an integer value that can be recognized by a DSP assembler using the Q15 format .The number is first normalize then scaled down by 2 to the appropriate value that can be accommodated by the bits number. Finally, the value will be rounded (truncated) to integer value and be represented in binary number [5]. A neuron can be viewed as processing data in three steps; the weighting of its input values, the summation of them all and their filtering by a activation function. The Neuron can be expressed by the following equation:

$$y_j = f\left(\sum_i w_{ij} x_i - \theta_j\right) \quad [1]$$

where y is the output of the neuron, w is the synaptic weight, x is the input and  $\theta_j$  is the bias. The subscript i denotes the preceding neuron and j the neuron considered. The neuron computes the product of its inputs, with the corresponding synaptic weights, and then the results are added. The result is presented to a comparison unit designed to represent an appropriate activation function such as linear, sigmoid or hyperbolic tangent [3]. The equation is shown in block diagrams in Fig.4. For the weighted inputs to be calculated in parallel using conventional design techniques, a large number of multiplier units would be required. To avoid this, multiplier/Accumulator architecture has been selected. It takes the input serially, multiplies them with the corresponding weight and accumulates their sum in a register [6-7].

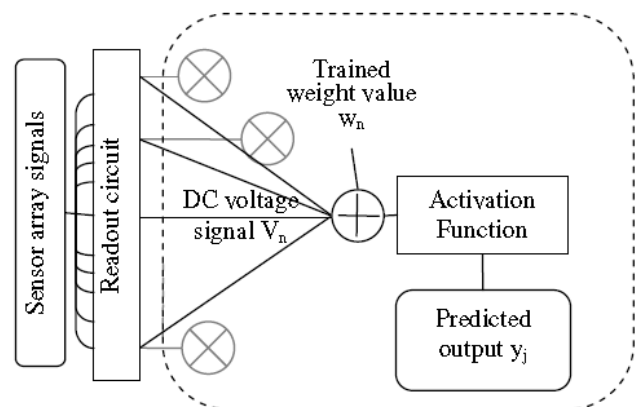


Fig.4. Structure of Neuron [7]

The block diagram and flow of the hardware implementation is shown in Fig. 5 and 6. The accumulator unit is composed of a bit-serial adder and 16 bit register. The design of multiplier accumulator consists of adder and multiplier. MAC are frequently used in general computing and are especially critical to performance of digital signal processing applications. The MAC typically operate on a digital, and usually binary, multiplier quantity and a corresponding digital multiplicand quantity and generate a binary product. The design of multiplier accumulator proposed in this project consists of adder and multiplier



that can accommodate or handle 4 channel of input (array of sensor).

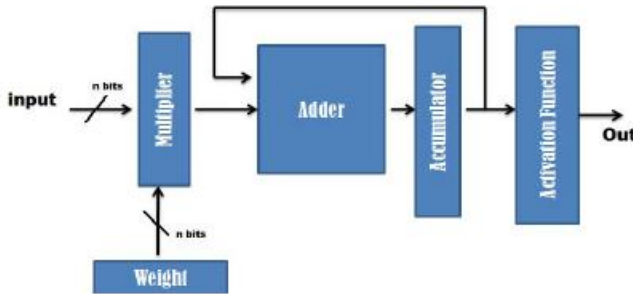


Fig.5: The flow of proposed neuron architecture

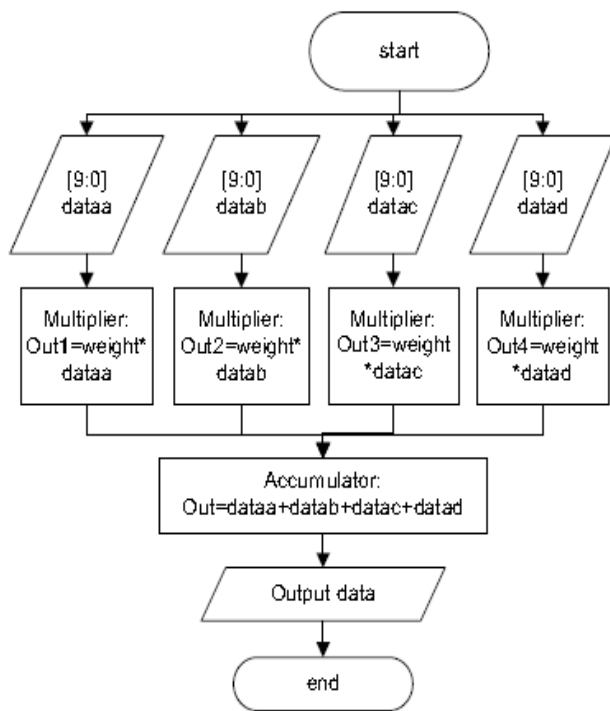


Fig.6: Signal handling of multiplier accumulator

The architecture for the MAC is shown in Fig. 7. With tree configuration as shown in Fig.12, the use of tile logic is quite uneven and less efficient than with a chain. The idea of this configuration is that the 2 value from multiplier were added separately. The partition of the computation is then added at adder4 for the final output.

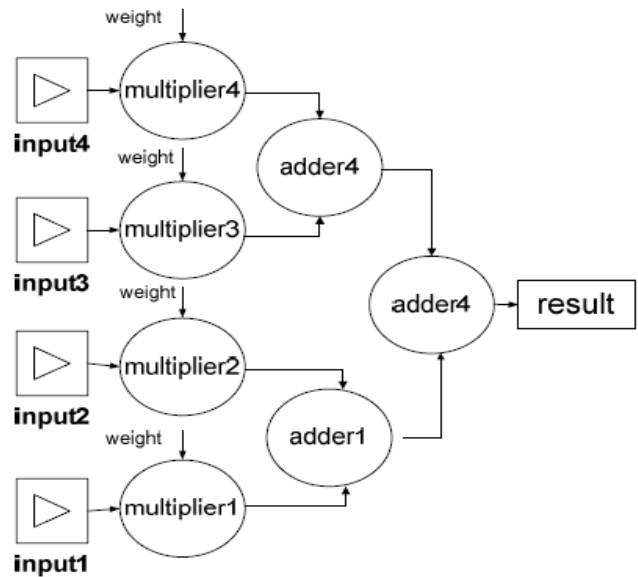


Fig.7: Tree configuration of Multiplier Accumulator

Activation function in a backpropagation network defines the way to obtain output of a neuron given the collective input from source synapses. The backpropagation algorithm requires the activation function to be continuous and differentiable. It is desirable to have an activation function with its derivative easy to compute. The mathematical algorithm for tanh approximation using Taylor's Series expansion that is used in the hardware calculation is provided by equation 2 [10-11]: The design flow is presented in Fig 8.

$$y = x - x^3/3 + 2x^5/15 + \dots \quad [2]$$

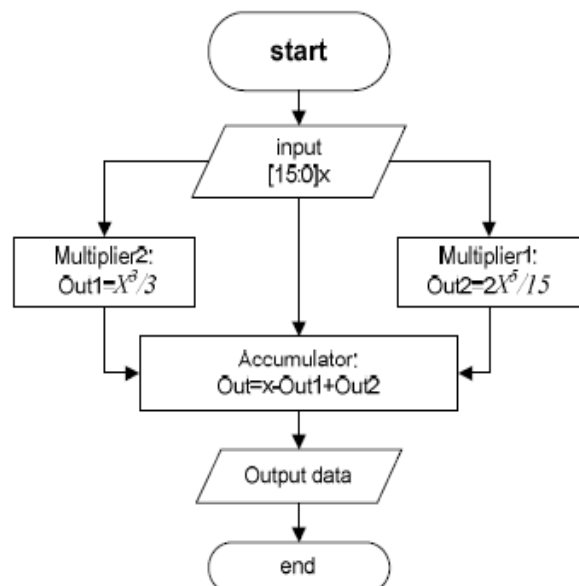


Fig.8: Design flow of activation function



### III. RESULTS AND DISCUSSION

The analysis considers clock-to-output delay which is the time to obtain a valid output at an output pin fed by a register (tCO) for all output pins and both minimum length of time that data must arrive before the active clock edge (tSU) and must be stable after the active clock edge (tH) for all input pins. The time required for an input pin signal to propagate through combinatorial logic and appear at an external output pin (tPD) is taken into consideration for any pin-to-pin combinatorial paths in the design [12]. The in-built timing analysis algorithm in Quartus is utilized to measure performances of the design by stages throughout the project. Comparison was made using booth multiplier to shift-add multiplier timing performance in the architecture shown in Fig 7. The multiplier timing performance itself is first compared as shown in Table I. The booth multiplier performs faster as shown by 18.6% as compared to shift-add multiplier for tCO. Booth multiplier performs faster compared to the basic shiftadd structure. The effect of the multiplier performance is amplified in the MAC architecture as shown in Table II with the MAC structure built with shift-add multiplier performing slower by 70.3%.

Multiplier Architecture	Timing Parameter		
	$t_{su}(ns)$	$t_{co}(ns)$	$t_h(ns)$
Booth Multiplier	5.611	7.399	0.232
Array Multiplier	5.851	9.131	0.914

Table I. Timing Performance Of Multiplier Module In Neuron

Timing parameter	Multiplier Architecture Utilized	
	Using Booth Multiplier	Using Shift-Add Multiplier
$t_{su}(ns)$	5.040	15.049
$t_{co}(ns)$	9.805	31.605
$t_h(ns)$	-0.435	17.867

Table II. Timing Performance Of Multiplier Accumulator Unit

For the activation function, two approaches were compared. Approach 1 is constructed using behavioral statements of equation (2) up to 3 terms of the expansion with continuous assignment, utilizing the inbuilt library architectures. In approach 1, whenever the value of a

variable on the right-hand side changes, the expression is re-evaluated and the value of the left-hand side is updated. Approach 2 uses RTL statements of the multiplier architecture and registers designed from previous stage. Table 3 shows that the time required for an input pin signal to propagate through combinatorial logic and appear at an external output pin for Approach 1 is 61.272ns and this is higher by 13.9ns compared to designed based on Approach 2. This is due to the fact that parallel multiplication were used in Approach 2 as compared to Approach 1.

Activation Function Architecture	$t_{pd}$
Approach 1: Behavioral Statement	61.272 ns
Approach 2: Multiplier and registers RTL Statements	47.372ns

Table Iii. Timing Performance Of Activation Function Unit

### IV. CONCLUSION

Modules for a neuron structure with hyperbolic activation on has been designed for hardware realization on digital platform. The work demonstrates that the performance of neuron architecture on FPGA depend strongly on the methodology, coding styles and also type of multiplier and adder used. The neuron implemented on FPGA with 38.832ns propagation delay and maximum fanout is 68. From the point of view neuron architecture it was found that using Tree structure for MAC Booth Multiplier gives lower delay compared to using ripple-carry adder and shift-add multiplier as its main component in the architecture. The activationfunction was achieved with better performance by 23% using array multipliers compared to behavioral statements of the mathematical expansion.

### ACKNOWLEDGMENTS

The authors would like to thank Universiti Teknologi MARA for funding the research work through the Excellence Fund Grant 600-RMI/ST/DANA 5/3/Dst(171/2011).

## REFERENCES

- [1] Wan Fazlida Hanim Abdullah., Masuri Othman, Mohd Alaudin Mohd Ali, and Md. Shabiul Islam. Improving ion-sensitive field-effect transistor selectivity with backpropagation neural network. *WSEAS Transactions of Circuits and Systems*. 9(11): 700-712,2010
- [4] 'Overview and Use of the PICmicro Serial Peripheral Interface', Microchip(TM), [page 1-9]
- [5] Poole, D., *Linear algebra : a modern introduction*. Belmont, CA, Thomson Brooks/Cole, 2005.
- [6] Polikar, R. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine* 6(3): 21-45, 2006.
- [7] A.Durg,W.V.Stoelker,J.P.Cookson,S.E.Umbaugh and R.H.Moss, "Identification of Variegating Coloring in Skin Tumors:Neural Network vs Rule Based Induction methods",*IEEE Eng. in med. and Biol.*, vol.12 pp.71-74 & 98,1993
- [8] FPGA Implementation of Artificial Neural Networks: An application on Medical Expert Systems,G.P.K Economou,E.P.Mariatos, N.M.Economopoulos, D.Lymeropoulos, and C.E. Goutis,Department of Electrical Engineering University of Patras, GR 261 10,Patras, Greece
- [2] Stradiotto, N. R., Yamanaka, H., & Zanoni, M. V. B. Electrochemical sensors: a powerful tool in analytical chemistry. *Journal of the Brazilian Chemical Society* 14: 159-173, 2003
- [3] FPGA Implementation of a Multilayer Perceptron Neural Network using VHDL,Yamina TARIGHT, Michel HUBIN,Proceedings of ICSP '98 [page1-3]
- [9] A.R Ormondi and J.C Rajapakse, FPGA Implementation of Neural Network, [page 271-296], 2006
- [10] Simon Haykin 'Neural Networks and Learning Machines, third edition, [page 40-45]
- [11] Benard Widrow, David E. Rumelhart, and Michael A. Lehr, "Neural networks: Applications in industry, business and science," *Communications of the ACM*,vol. 37, no. 3, pp. 93-105, Mar. 1994.
- [12] Altera Corporation 'Cyclone III Device Handbook, Volume 1', [chapter 5,page 1-8], July 2007