ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

Hardware Accelerator for General Purpose Artificial Neural Network

S. R. Ganorkar, Padmaraj A. Jain

Abstract*— the goal of this work is to present the simpler reduced instruction set computing (RISC) Super Harvard architecture. This architecture will support general purpose computing as well as artificial neural network (ANN) acceleration. The multiply and accumulate (MAC) operation is frequently used to compute a generic artificial neuron. The design optimizes the frequently used MAC Operation. The architecture is presented using Very High Speed Integrated Circuits Hardware Description Language (VHDL). The architecture is implemented using Xilinx ISE 14.7 IDE for Xilinx Spartan 6 series FPGA family. The simulation results are analyzed in terms of: operating frequency, resource utilization, instruction throughput.

***Index Terms*—ANN, MAC, RISC, VHDL.**

I. INTRODUCTION

ANN has wide variety of applications in solving problems in areas of pattern recognition, image processing and medical diagnostic. Advantage of ANN over conventional computing is its ability to use existing computational power to provide approximate solution. Whereas in case of conventional computing a modification in algorithm is required to achieve same. ANN is parallel distributed information processing system [1]. Maximum computational throughput can be achieved by implementing it on parallel hardware architecture. Increase in throughput allows adding more accuracy to the solution. Parallel hardware architecture increases the chip area consumption and eventually increases the cost of solution. A balance need to be achieved between desire accuracy and parallelism required to attain the accuracy. ANN can be implemented by balancing between software and hardware. There are multiple ways the hardware part can be designed. One way is to implement a neuron or multiple identical neurons with fixed neuron design parameters. Advantage of such system is the increase in throughput. Disadvantages of such system are more than its advantages. Such system may not be suitable for general purpose computation. As design parameters of neurons are fixed flexibility to use the system in case of different parameters or different type of neural network is lost. For general purpose computing another general purpose computational unit is required. In this paper, we are proposing a design which tries to balance between the parallelism and accuracy required for given neural network. As this design target the computationally intense part in most types of the neural network; it is flexible to use in computing different types of neural network. We have tested the design simulation by implementing it using VHDL for FPGA.

II. ARTIFICIAL NEURAL NETWORK (ANN)

A. Introduction to ANN

Artificial neural networks are inspired from the nature's neural network design in human brain. The natural neural network design is adaptive; it can process incomplete information; it does not have any predefined algorithm and can reach to an approximate solution. The problem solving approach using general purpose computation analyses the task and derives a suitable algorithm. If successful, the result is immediately available. Neural networks can solve problems which are difficult to describe in an analytical manner. Only disadvantage is: prior to usage, the network must be trained.

B. ANN Architecture

Basic architecture of ANN can be described using example of multi-layer Perceptron. It consists of several layers of processing elements (PE's) or artificial neurons [1]. It has minimum three layers of artificial neurons:

- Input layer: Real time inputs are connected to this layer.
- Hidden Layers: One or more hidden layers are connected to input and output layer.
- Output Layer: Final layer processing output from hidden layers.

An input vector is presented to the neural network which determines an output. The comparison between the computed and desired output provides an output error [1].

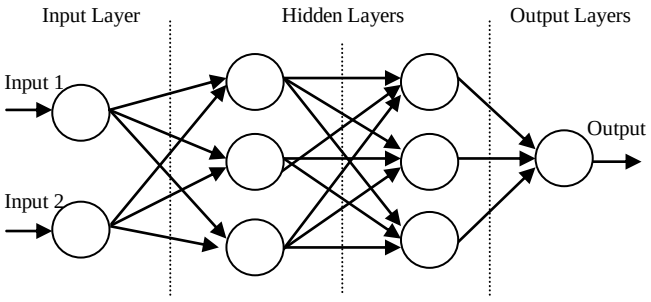

Fig. 1 Multi-layer Perceptron Network.

The error is used by a learning algorithm to adapt the network parameters. Fig.1 shows diagrammatic representation of multi-layer Perceptron with one input layer, two hidden layers and one output layers. Each circle represents neuron or PE.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

C. *Mathematical Model of Artificial Neuron*

General neuron of ANN can be described in mathematical form as shown in (1).

$$Output_n = f(\sum_{i=0}^{N}(w_i \times x_i)) \qquad (1)$$

Where,

n = Index of neuron,
f = Activation function,
N = number of input or weights,
w = weights,
x = inputs

The model of neuron in ANN based on above equation is shown in Fig.2

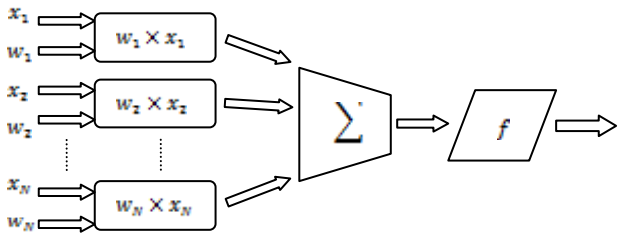

Fig. 2 Mathematical Model of neuron

III. ANN HARDWARE ACCELERATOR

We are able to observe that there is inherent spatial parallelism in the execution of the neuron's products, called the intra-neural parallelism. We notice that a neuron inside a layer is independent from the others within the same layer. We can even observe that, there is dependency among the neurons from a layer and neurons from previous layer. Output of previous layer is input to the next layer. Each Neuron also has mathematical operation of MAC, where multiple multiplication and addition operation can be performed in parallel. The design proposed in this paper explores parallelism in MAC operation and the property of the neuron layer where input is fixed. Parallelism in multiplication operation is addressed by implementing four two input multipliers and four two input adder with an accumulator. In case of computing single layer, input is fixed and weights change from neuron to neuron. Input is captured in input registers which are constant for one layer.

A. *Hardware Design*

Proposed design is Super Harvard RISC architecture. It has separate data and code memory. It uses three stage pipelines to ensure one instruction per clock throughput. The important components of proposed architecture which support parallelism in MAC operation are:

- 4 x 32-bit Multipliers
- 2 input, 4 x 32-bit / 4 input, 1 x 64-bit Adder
- 16 x 32-bit input registers (I0 – I15)
- 4 x 32-bit weight registers (W0 – W3)
- 1 x 32-bit address register
- 8 x 32-bit / 4 x 64-bit output registers

Input register file is a special purpose register file of 16 x 32-bit registers. At a time only 4 x 32-bit register can be accessed. Selection of particular 4x32-bit registers out of 16x32-bit registers is performed by using register group selection logic. Such selection allows storing 16 inputs and using them as global registers in group. Weight register file is a general purpose register file of 4x32-bit registers. These registers are accessible simultaneously unlike the input registers. One address register is also included in the architecture to load the input registers and weight registers with values from memory. Different modes of multiplier and adder and respective input and output configuration are showed in I and II:

I: Multiplier operations with different modes

Mode	Input registers	Weight registers	Output registers
Multiplier 1	I0 / I1 / I2 / I3	W0	O0-O1
Multiplier 2	I4 / I5 / I6 / I7	W1	O2-O3
Multiplier 3	I8 / I9 / I10 / I11	W2	O4-O5
Multiplier 4	I12 / I13 / I14 / I15	W3	O6-O7

One more mode is also supported where all multiplier with same input and output register configuration work parallel at same instance.

II: Adder operations with different modes

Mode	Input registers	Weight registers	Output registers
Adder 1	I0 / I1 / I2 / I3	W0	O0-O1
Adder 2	I4 / I5 / I6 / I7	W1	O2-O3
Adder 3	I8 / I9 / I10 / I11	W2	O4-O5
Adder 4	I12 / I13 / I14 / I15	W3	O6-O7

One more mode is also supported where all multiplier with same input and output register configuration work parallel at same instance. A special mode of adder with multiplier allows performing four MAC operations in single clock cycle. With all these capabilities of ALU in the architecture Fig. 3 shows the block diagram of the ALU. Special care has been taken to capture the output as 64-bit in case of 32-bit mode to save the precision of output. Similar care is taken in MAC operation and output is stored as 128-bit. Output registers are selected based on type of math operation and mode of input. Such selection of output registers makes sure integrity of output and output is not corrupted by other parallel operations.

III shows details of Operations supported by this architecture:

III: ANN Accelerator Operations.

Sr. No.	Operation	Src 1	Src 2	Dest
1	Move	In	-	In
2	Move	Wn	-	Wn
3	Move	#immediate	-	Address

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

				register
Sr. No.	**Operation**	**Src 1**	**Src 2**	**Dest**
4	Load (/repeat)	Memory	-	In
5	Load (/repeat)	Memory	-	Wt
6	Multiply	I0/I1/I2/I3	W0	O0-O1
7	Multiply	I4/I5/I6/I7	W1	O2-O3
8	Multiply	I8/I9/I10/I11	W2	O4-O5
9	Multiply	I12/I13/I14/I15	W3	O6-O7
10	Multiply	I0/I1/I2/I3	W0	O0-O1
		I4/I5/I6/I7	W1	O2-O3
		I8/I9/I10/I11	W2	O4-O5
		I12/I13/I14/I15	W3	O6-O7
11	Add	I0/I1/I2/I3	W0	O0-O1
12	Add	I4/I5/I6/I7	W1	O2-O3
13	Add	I8/I9/I10/I11	W2	O4-O5
14	Add	I12/I13/I14/I15	W3	O6-O7
15	Add	I0/I1/I2/I3	W0	O0-O1
		I4/I5/I6/I7	W1	O2-O3
		I8/I9/I10/I11	W2	O4-O5
		I12/I13/I14/I15	W3	O6-O7
16	MAC(/repeat)	I0/I1/I2/I3	W0	O0-O3
		I4/I5/I6/I7	W1	
		I8/I9/I10/I11	W2	
		I12/I13/I14/I15	W3	

IV. IMPLEMENTATION RESULTS

IV: Device information

Parameter	Value
Family	Xilinx Spartan 6
Device	XC6SLX4
VHDL Source analysis standard	VHDL-93

V: Device utilization summary

Parameter	Value
Number of slice registers	1490
Number of slice LUT's	725
Number of occupied Slices	405
Number of MUXCYs used	112
Number of LUT Flip Flop pairs used	1550
Number of bonded IOBs	35
Number of BUFG/BUFGMUXs	1

VI: Timing information

Parameter	Value
Minimum Period	17.367 ns
Maximum Frequency	57.582 Hz

VII: Maximum Math Operation throughput

Parameter	Value per clock
Add	4
Multiply	4
MAC	4

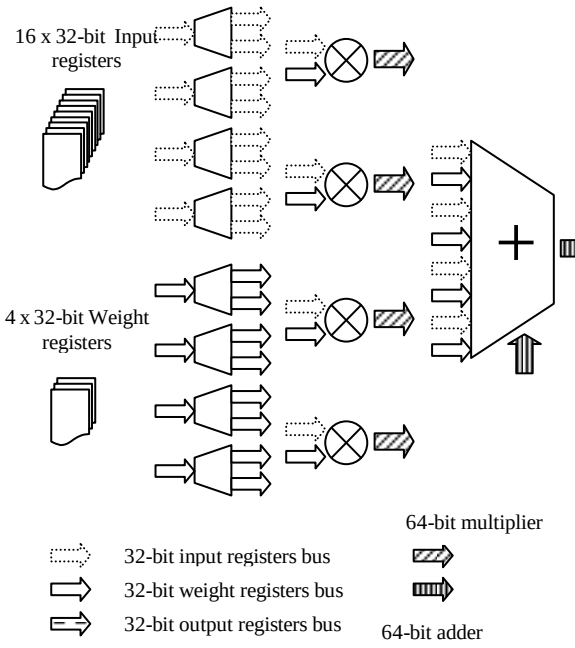

Fig. 3 Block diagram of ALU

REFERENCES

[1] Sandrin Ntoune Ntoune, R. and Bahoura, M., "FPGA-implementation of pipelined neural network for power amplifier modeling," New Circuits and Systems Conference (NEWCAS), 2012 IEEE 10th International, pp. 109-112, June 2012.

[2] Chandrashekhar Kalbande and Anil Bavaskar, "Implementation of FPGA-Based General Purpose Artificial Neural Network," ITSI Transactions on Electrical and Electronics Engineering (ITSI-TEEE), vol. 1, issue 3, 2013.

[3] Hariprasath, S. and Prabakar, T. N., "FPGA implementation of multilayer feed forward neural network architecture using VHDL," Computing, Communication and Applications (ICCCA), 2012 International Conference, pp. 1-6, Feb 2012.

[4] Granado J.M., Vega M.A., Perez R., Sanchez J.M. and Gomez J.A., "Using FPGAs to Implement Artificial Neural Networks," Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference, pp. 934 – 937, Dec 2006.

[5] Suhap Sahin, Yasar Becerikli, and Suleyman Yazici, "Neural Network Implementation in Hardware Using FPGAs," ICONIP 2006, Part III, pp. 1105-1112, 2006.

[6] Krips M., Lammert T. and Kummert A., "FPGA implementation of a neural network for a real-time hand tracking system," Electronic Design, Test and Applications, 2002, IEEE International Workshop, Jan 2002.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 10, April 2014

AUHTOR'S PROFILE

Prof. S. R. Ganorkar joined Sinhgad Technical Education Society in September 2002. He completed M.E. (Advanced Electronics) and Ph. D. (E&TC). He is working as Professor having total experience of 25 years including 13 years of industry. Area of research is biomedical signal processing and soft computing. He got outstanding academic performance award twice. He completed BCUD research project of University of Pune on Iris recognition: An emerging biometric technology. His findings got published in 19 international journals, 16 international conferences and 41 national conferences clearly reflecting the passion towards the domain area.

In addition to teaching, he believes in playing multiple roles to bring out various facets of personality and has successfully executed various roles as coordinator for Ph.D., NBA, Purchase and Student and teacher training program. He worked as a Nodal officer (Procurement) under TEQIP-II & IEEE branch counselor.

Padmaraj A. Jain is currently pursuing M.E. in Digital systems (Electronics Dept.) from Sinhgad College of Engineering, Pune University, Pune. He has 8 years of industry experience in field of embedded systems, low power graphics processing unit (GPU) and respective applications.