



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Hardware Implementation of ANN and SNN Model- A Survey

SoniChaturvedi¹, Dr. A.A.Khurshid², Dr. S.S. Dorle³

Research Scholar, Electronics Engg., GHRCE, Nagpur, India

Professor and Head, Electronics Engg., RCOEM, India

Professor and Head, Electronics Engg., GHRCE, Nagpur, India

Abstract: For Validation of any algorithm based results recently the advance research is heading in the direction of Hardware Implementation. Whether it is a digital system or a Neural Network system or may be any real time application. With the advent of Nanotechnology(SOC) any design with large number of gates can be squeezed into a tiny chip area with minimum power requirements, which leads to integration of innumerable applications on to a single device so as to design any electronic consumer product initiated in the era of digital convergence. With the introduction of Reconfigurable computing technology since mid of 80's, FPGA's have gained momentum for implementation of Random Logic, Prototyping, Reconfigurable Hardware and special purpose computation engines. Similar to FPGA, ASIC may also be implemented though a separate category, has many applications in the area of signal processing and networks. One has many choices for selecting either of these reconfigurable techniques based on Speed, Gate Density, Development, Prototyping and simulation time, cost.

Keywords: ANN; SNN; FPGA; ASIC; Netlist; Power optimization;

I. Introduction

FPGAs are recognised as being a suitable configurable platform for high-speed spiking neural network simulations, due to FPGA fabric's highly reconfigurable nature. However, designing an FPGA based simulator takes significant amounts of time and hardware design expertise are required. It is desirable that a fast emulation platform provides user-friendly configure interface for general users. FPGA is a two-dimensional array of logic blocks and flip-flops with the interconnection between the logic blocks and function of each block. FPGAs were introduced as an alternative to custom ICs for implementing logic: SSI/MSI components (within around 10x of custom ICs) and computer aided design (CAD) tools circuits could be implemented in a short amount of time (no physical layout process, no mask making, no IC manufacturing).

FPGAs are not appropriately suited for efficient SNN implementation as they attempt to map biological neuron and synaptic computations onto general arrays of configurable digital logic blocks which are not optimized in area-size or power consumptions for dense network realizations. This FPGA interconnect challenge is a significant limiting factor in the suitability of FPGAs for SNN implementation. Researchers have investigated several routing optimizations and topologies in attempts to improve the FPGA routing latency and performance for generic architecture.

Application specific integrated circuits (ASICs) are non-standard integrated circuits designed for a specific use or application. Generally this design used for a product having large production run. This contains large part of the electronics on single integrated circuits. Cost of this design is high, and so therefore they reserved for high volume products. By using this design we get exact requirements of the product and all design contained in one integrated circuit so number of components can be reduced. They are widely used in high volume products like cell phones or other similar applications, often for consumer products where volumes are higher, or for business products that are widely used. Generally, the process of designing an ASIC can be divided into two phases called the front end which is the starting point of the design process, and the back end where the physical implementation is carried out. Concerning digital logic, the front end design starts with a behavioral description of the ASIC using a high-level hardware description language, such as Verilog or VHDL4. The following digital circuit synthesis involves the translation of the behavioral description of a circuit into a gate-level netlist comprising generic logic gates and storage elements. For the physical implementation, these generic elements need to be mapped to a target technology. This technology consists of a library of standard logic cells (AND, OR, etc.) and storage elements (flip-flops, latches) that have been designed for a specific fabrication process. The development and manufacture of an ASIC design including the ASIC layout is a very expensive process. In order to reduce the costs, there are different levels of customization that can be used. There are three levels of ASIC that can be used.

Gate Array: This type of ASIC is the least customisable. In this the silicon layers are standard but the metallization layers allowing the interconnections between different areas on the chip are customisable. This type of ASIC is ideal where a large number of standard functions are required.

Standard cell: In this type of ASIC, the mask is a custom design, and the silicon is made up from library components. This provides a high degree of flexibility.

Full custom design: This type of ASIC involves the design from down to transistor level so this is most flexible. The ASIC layout can be tailored to the exact requirements of the circuit. The costs are very much higher and it takes much longer to develop, but it gives the highest degree of flexibility. The risks are also higher because the whole design is untested and not built up from library elements.

Artificial neural networks have been applied in pattern recognition, forecasting, problems intelligent control and so on. among themost popular neural network models we could mention the feed-forward neuralnetwork trained with the back-propagation algorithm. this type of neural network cannot reach an optimum performance in non-linear problems if it is not well designed. The selection of a learning algorithm andparameters such as number of neurons, number of layers, and the transfer functions,determine the accuracy of the network in complex architectures to solve the problem.Artificial neural networks are inspired by the biological systems. Artificial neurons and neural networks try to imitate the working mechanisms of their biological counterparts. Learning can be an optimisation process. Biological neural learning get by the modification of the synaptic strength. Artificial neural networks learn in the same way. The modification rules for artificial neural networks can be derived by applying mathematical optimisation methods. Learning tasks of artificial neural networks can be reformulated as function approximation tasks. Neural networks can be considered as nonlinear function where the parameters of the networks should be found by applying optimisation methods.

Spiking neural network comprises of computation neurons which is based on the spike processing. This systems found by neuroscientist.In SNN we use spikes as a input which is encoded from digital values. This comes under third generation neural network model. For travelling spikes, neuron must go through the weights and delays. The biological neural structures are nothing but the spiking neural network.Artificial neuron networks (ANN) only based on weights but in case of SNN it also based on delay factor. Delay plays very important role in spiking neural network because according to length spike delay from one neuron to successive neuron is different.

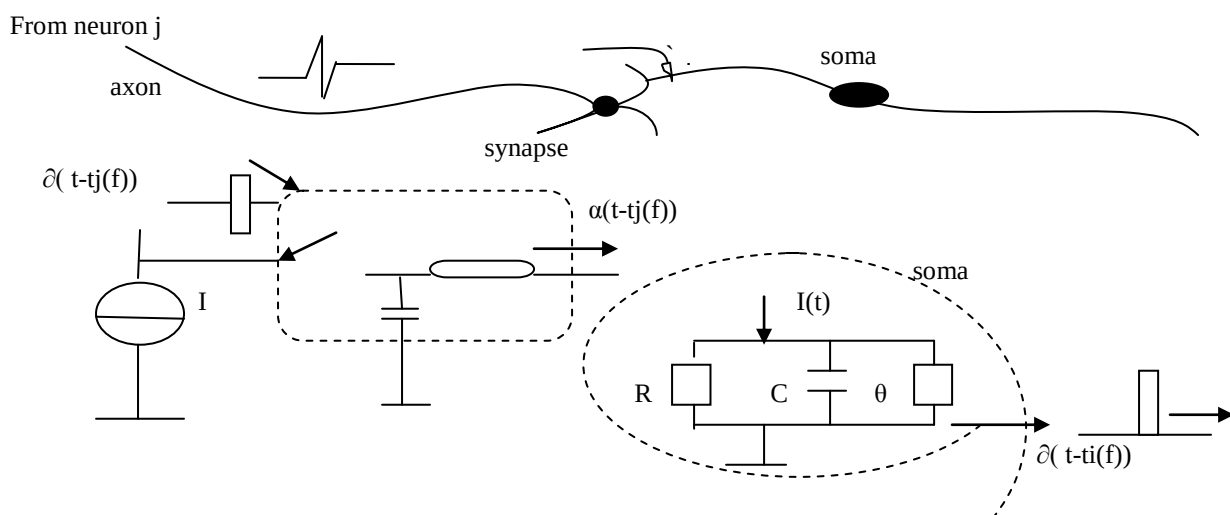


Fig. 1. Leaky Integrate and Fire Neuron Model

Spiking networks have temporal functionality. We used Integrate-and-Fire (IAF) neurons. Each neuron has membrane potential (m) as a forward connection to, and resets m . Membrane potential is a way of implementing memory that varies through time. A large domain of applications would benefit from the stronger inference capabilities including speech recognition, computer vision, textual and image content recognition, robotic control, and data mining. The brain utilizes a large collection of slow neurons operating in parallel to achieve very powerful cognitive capabilities. There has been a strong interest amongst researchers to develop large parallel implementations of cortical models on the scale of animal or human brains. At this scale, the models have the potential to provide much stronger inference capabilities than current generation computing algorithms. Several research groups are cortical column based models on high performance computing clusters and groups are examining large scale implementations of neuron based models.

Integrate-and-fire (I&F) neuron is the first model used in computational neuroscience. It is class 1 excitable. It is an integrator. It can also capable of firing tonic spikes with constant frequency. To prove analytical results I&F

is worst models for simulations. current $I(t)$ charges the RC circuit. Voltage $u(t)$ is compared to a threshold. If $u(t)$ exceeds threshold then output pulse at $(t - t_i(f))$ is generated.

II. Hardware Implementation of ANN

ANN models adapt to problems according to a given learning algorithm on changing the strength of the interconnections between computational elements. However, constrained interconnection structures have a limit on such ability. ANNs with circuit structure adaptation are well suited for Field programmable hardware devices. The hardware implementation is better comparing with software approach because it can take advantage of parallelism and natural features of neural network (NN). Many researchers have made great efforts on the realization of NN using FPGA technique. Since FPGA is a digital device that have programmable properties. The different alternatives used to implement, artificial neural networks in FPGAs includes the neuron's multiplier implementation. Using the Boolean function few researchers have also developed the algorithm for hardware implementation of compact neural network to describe the operation of perceptron which is nothing but an artificial neurons. Number of redundant gates can be reduced by conversion of this model into logic gate structure and hardware optimisation. Choices of various tools like, QuartusII, Active – HDL etc. is available to generate an optimised VHDL model. It helps to combine ANN design and simulation software packages with the hardware design so that the activation functions of the neurones can be extended.

III. Hardware Implementation of SNN

In this review paper the importance of large scale implementations of spiking neural networks on FPGA's has been discussed. We discuss the challenges and possible solution for the importance of low cost embedded solution using closer abstraction of real neurons to improve the performance in an adaptable, real-time environment. The programmable hardware is used for the adaptable requirements of neural network. Here we focus on using Izhikevich spiking neuron model due to its accuracy and efficiency. The development of modularized processing element for evaluation of large number of spiking neurons in a pipelined manner is done to explore the feasibility of FPGA's for large scale simulations. The utilisation of character recognition algorithm is done over 9000 neurons. The results shown by the researcher indicate that FPGAs are suitable for large scale cortical simulations using the Izhikevich spiking neuron model.

Recent neuropsychological research has begun to reveal that neurons encode information in the timing of spikes. Spiking neural network simulations are a flexible and powerful method for investigating the behaviour of neuronal systems. Simulation of the spiking neural networks in software is unable to rapidly generate output spikes in large-scale of neural network. An alternative approach, hardware implementation of such system, provides the possibility to generate independent spikes precisely and simultaneously output spike waves in real time, under the premise that spiking neural network can take full advantage of hardware inherent parallelism. We introduce a configurable FPGA-oriented hardware platform for spiking neural network simulation in this work. We aim to use this platform to combine the speed of dedicated hardware with the programmability of software so that it might allow neuroscientists to put together sophisticated computation experiments of their own models.

References

- [1] A. Rosado-Muñoz, A.B. Fijałkowski, M. Bataller-Mompeán, J. Guerrero-Martínez, "FPGA implementation of Spiking Neural Networks supported by a Software Design Environment", Preprints of the 18th IFAC World Congress Milano (Italy), August 28 September 2, 2011.
- [2] Vinita Dutt, Sunil Dutt "Hand Written Character Recognition Using Artificial Neural Network" 2011.
- [3] Ankur Gupta and Lyle N. Long "Character Recognition Using Spiking Neural Network" presented at IEEE Neural Network Conference, Orlando, FL, August 2007. [5] [4] Eugene M. Izhikevich "Simple Model of Spiking Neuron", IEEE Transaction on Neural Networks, Volume 14, No.6, November 2003.
- [4] Yihua Liao "Neural Networks in Hardware: A Survey".
- [5] Jilles Vreeken "Spiking neural networks, an introduction".
- [6] Wikipedia "Spiking neural network". [10] Prof. Dr. Karlheinz Meier, Prof. Dr. René Schüffny "VLSI Implementation of a Spiking Neural Network".
- [7] Roberto A. Vázquez "Izhikevich Neuron Model and its Application in Pattern Recognition".
- [8] Ankit Sharma, Dipti R Chaudhary "Character Recognition using Neural Network", International Journal of Engineering Trends and Technology (IJETTE) - Volume 4 Issue 4-April 2013J.