iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** III **Month of publication:** March 2017

DOI: http://doi.org/10.22214/ijraset.2017.3076

www.ijraset.com
IC Value: 45.98

Volume 5 Issue III, March 2017
ISSN: 2321-9653

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

FPGA Based Platform for Spiking Neural Network

Devanshi Raval[1], Leuva Bhumika[2]
[1]Student ME(EC) , [2]Asst. Prof. of H.G.C.E. (EC)
Hasmukh Goswami College of Engineering, Vahelal, Ahmedabad, India

Abstract: Neuromorphic engineers are studying the nervous system and trying to emulate its function and organization in their computational and robotics systems. They are hoping to match the human brain in vision, hearing, pattern recognition and learning tasks. Our goal is to create Field Programmable Gate Array (FPGA) platforms of large scale spiking neural networks to allow the testing of certain hypotheses related to neuroscience theories. Virtualization is also very important concept and performance/price of spiking neural network. We implement general purpose spiking neural network platform using FPGA and observe performance and performance/price tradeoffs.

Keywords: Spiking neuron, Neural network, FPGA, Virtualization, Processing node, Hardware, Performance /Price

I. INTRODUCTION

The brain is complex and important organ of human body which can deal with a large quantity of streams of input information and give response. Scientists of the field of Artificial Intelligence (AI) want to imitate such a machine so it can be used for all kinds of processing tasks like image processing, speech processing, etc. The brain consists of a large number of nerve cells which called neurons. These neurons connect to each other and create network is called neural network. So that network is biological neurons. Scientists of the field of Artificial Intelligence (AI) try to model biological neural network and create a platform, so they can check their hypothesis related to brain nerves system. Here we will consider the Spiking Neuron model, which is 3rd generation of neural network.

A. What are Spiking Neural Network?

Spiking neural networks fall into third generation of neural network models. In a neural simulation increasing the level of realism. In addition to neuronal and synaptic state that SNNs also incorporate the concept of time into their operating model. SNN represent a special class of artificial neural networks (ANN), where neuron models communicate by sequences of spikes. Networks composed of spiking neurons are able to process substantial amount of data using a relatively small number of spikes. Due to their functional similarity of that biological neurons, spiking models provide powerful tools for analysis of elementary processes in the brain its including neural information processing, plasticity & learning. "The third generation of neural networks" establishing that the intricate workings of spiking neurons could support the computations required for general function approximation, just like standard artificial neural networks based on analog (rate) neurons.

B. Spiking Neuron Models

The general process of spiking signal transmission is well known, and illustrated in Figure 1 action potentials travel along axons and activate synapses. These synapses release a neurotransmitter that quickly diffuses to the post-synaptic neuron. In the post-synaptic neuron its neurotransmitters affect the neuron's membrane potential. Excitatory Postsynaptic Potentials (EPSPs) increase the membrane potential (depolarize), and without new inputs, this excitation then leaks away with a typical time constant. Its membrane potential rises quickly with each incoming spike, and then slowly decays again (inset). If several spikes arrive in a short time window the membrane potential crosses a threshold & a spike is fired down the axon. Inhibitory Postsynaptic Potentials (IPSPs) decrease the membrane potential (hyper polarization).When sufficient EPSPs arrive at a neuron, the membrane potential may depolarize enough to reach a certain threshold, and the neuron generates a spike itself, resetting its membrane potential. The thus generated spike then travels on to other neurons. The above is a stark simplification, and real neurons display many different spiking behaviors some respond to input only after a delay, others respond with a burst of spikes. To explain their behaviors, detailed models were developed with the Hodgkin-Huxley model the most famous. The models are typically phrased as dynamical systems of various complexities and it does include models like the Leaky-Integrate-and-Fire (LIF) model. The

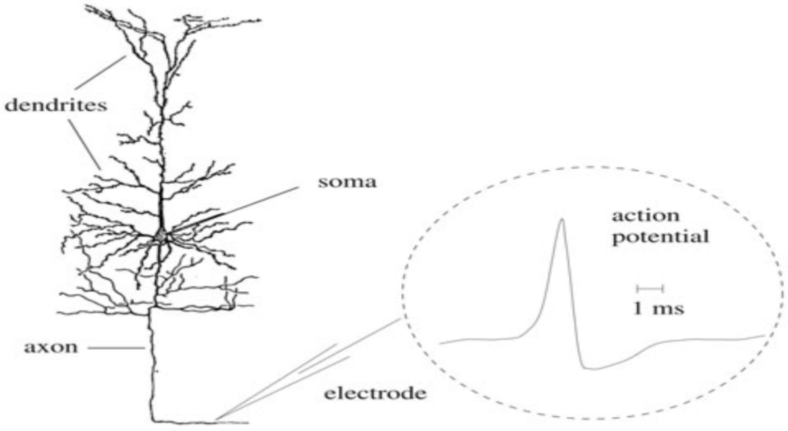


Figure 1: Spiking neurons of real neurons communicate with each other via sequences of pulses – spikes.

Quadratic-Integrate-and-Fire model and more complicated models, representing different trade-offs between the neuron scientific realism and computational complexity. An alternative to dynamical systems models are called Spike Response Models (SRMs). In SRMs, the membrane potential is not computed through differential equations but as a sum of integral kernels. In principle, SRMs are equivalent to many classes of dynamical systems models, but offer a different insight into the processes in spiking neurons. In this view, a neuron include its incoming synapses acts mathematically as filter or operator in that it maps incoming sets of spike trains into an output spike train. Both synapse and dendrite are known to actively contribute to neural computation but compared to spiking neurons, much less is known. To capture such extended neural computation, spiking neuron models would have to be complemented by more detailed models of synapses and dendrites.

II. LITERATURE SURVEY

Paper-1: An FPGA design framework for large-scale spiking neural networks.
Authors: Runchun Wang, Tara Julia Hamilton, Jonathan Tapson, André van Schaik
Publication/year: IEEE, the MARCS Institute, University of Western Sydney, Sydney, NSW, Australia-2014.
Concept: In this Paper, FPGA design framework for large scale spiking neural networks, Understanding of mathematical model of Neuron and Hardware utilization versus number of neurons. The proposed FPGA design framework is based on a reconfigurable neural layer, which is implemented using a time-multiplexing approach to achieve up to 200,000 virtual neurons with one physical neuron using only a fraction of the hardware resources in commercial-off-the shelf FPGA. Rather than using a mathematical computational model, the physical neuron was efficiently implemented with a conductance-based model, of which the parameters were randomized between neurons to emulate the variance in biological neurons. Besides these building blocks the proposed time-multiplexed reconfigurable neural layer has an address buffer, which will generate a fixed random weight for each connection on the fly for incoming spikes for neuron.
Conclusion: In this paper presented an FPGA design framework, which can implement large-scale spiking neural networks in a highly efficient way. But utilization of the hardware resources on the FPGA for one neural layer and the neural network that has high LUTs and BRAM.

Paper-2: Performance/price estimates for cortex-scale hardware: A design space exploration
Authors: Mazad S. Zaveri, Dan Hammerstrom1
Publication/year: 2010, Department of Electrical and Computer Engineering, Maseeh College of Engineering and Computer Science, Portland State University, Portland, OR, United States
Concept: In this paper, we review on concept of virtualization. Virtualization is useful for understanding & investigating the performance/price trade-off and other trade-offs related to the hardware design space. A design space exploration is a necessary part of the study of hardware architectures for large-scale computational models for intelligent computation including AI, bio-inspired & neural models. A methodical exploration is needed to identify potentially interesting regions in the design space & to assess the relative performance/price points of these implementations. The specific results suggest that hybrid nanotechnology such as CMOL is a promising candidate to implement very large-scale spiking neural systems, providing a more efficient utilization of the density

and storage benefits of emerging nano-scale technologies. In general, we believe that the study of such hypothetical designs/architectures will guide the neuromorphic hardware community towards building large-scale systems & their help guide research trends in intelligence computing, and computer engineering.

III. PROBLEM STATEMENT

Spiking neural network is highly parallel structure we required for testing platform for Artificial Intelligence related hypothesis so we taking flexible FPGA platform for implementing maximum number of neurons using minimum hardware utilization.

IV. IMPLEMENTATION

A. Hardware Utilization

The human brain consists of an estimated 2 billion neurons. All neurons having personal microcontroller or processing unit. All neurons are connected with each other. At a time only 1 or 2% neurons work. So output of the cortex neuron structure of that Hardware Utilization is approx zero or minimum. So we required virtualization.

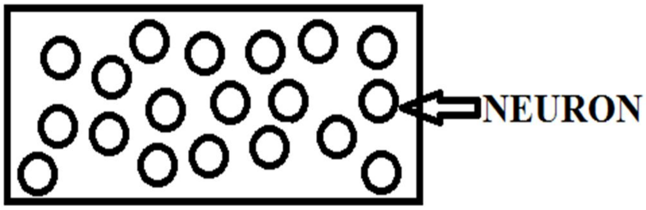


FIG 2: Cortex Neuron Structure

Figure 3 shows that cortex neuron cluster , so that struture create different different cluster that’s called virtualization. Virtualization is nothing but the Multiplexing.All the cluster’s having personal microcontroller or processing unit. That clusters are connected with each other and give the process between them is parallel. Cluster’s neurons are give that process is sequencial. So that process get more time but our Hardware is maximum utilize.

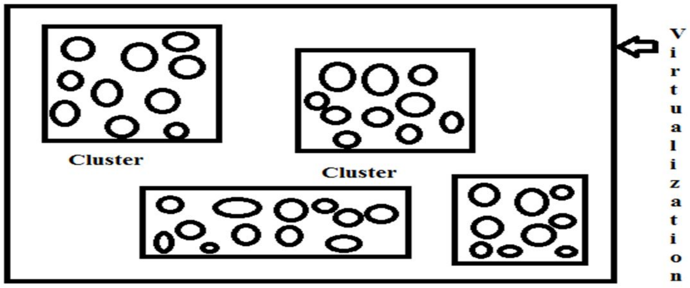


FIG 3: Cortex Neuron Cluster

B. Hardware Virtualization

When studying various designs/architectures for implementing biologically inspired computational models, one of the most important decisions, which has a significant impact on the performance/price of the implementation, is the degree of ‘‘virtualization’’ used in the hardware mapping. An important issue in implementing biologically inspired models as neuron model in silicon is the degree of virtualization utilized by the underlying hardware. Virtualization is defined as ‘‘the degree of time-multiplexing of the components of computation & communication via hardware resources’’. The concept of virtualization implies ‘‘architecture assessment methodology’’ for studying a range of hardware implementations and the various trade-offs and involved in implementing neural computational models. Such trade-offs like encompass computations, communication, multiplexing, programmable vs. non-programmable, sequential vs. parallel, general purpose vs. application-specific, fine-grained vs. coarse grained, data precision, timings etc. In fact, virtualization has a much big usage than simple ‘‘time-multiplexing’’ & therefore, allows us to think about all the various dimensions of the architecture space. Since neural models consist of fine grained networks, very fine grained parallel implementation is generally possible and providing a wide range of implementation options. Virtualization allows us to study design trade-offs over that range. Fine-grained hardware implementations may not always give that the best performance/price. A typical ‘‘hardware virtualization spectrum’’ for the neural models we are studying is shown in Fig. 4. It is essentially a software and hardware design space for implementing massively concurrent algorithms. As we move from left to right, the time-multiplexing of hardware resources decreases & parallelism is increases. And as we move from top to bottom, the programmability as reconfiguration ability of hardware varies from flexible to non-flexible. ‘‘Flexible’’ is a subjective term so that is not formally defined here. We use it to represent the kinds of ‘‘programmability’’ that is generally possible with more highly multiplexed hardware so best option for Hardware Utilization is FPGA. The right bottom corner represents a design in which the algorithm is directly mapped into silicon (for example, analog designs/architectures in region 8), such designs achieve maximum parallelism and performance, but have minimum virtualization, and minimum flexibility. An absolute minimum virtualization implementation would have a multiplier at each synapse and an accumulator at each branch in the dendrite tree.

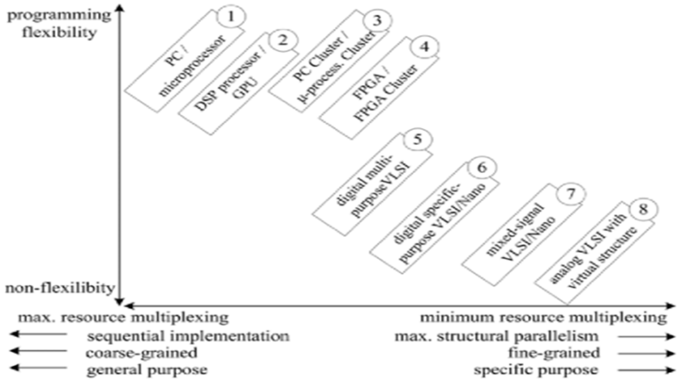

programming flexibility
non-flexilibity
1
PC / microprocessor
2
DSP processor / GPU
3
PC Cluster / µ-process. Cluster
4
FPGA / FPGA Cluster
5
digital multi-purposeVLSI
6
digital specific-purpose VLSI/Nano
7
mixed-signal VLSI/Nano
8
analog VLSI with virtual structure
max. resource multiplexing
sequential implementation
coarse-grained
general purpose
minimum resource multiplexing
max. structural parallelism
fine-grained
specific purpose

FIG: 4 Hardware virtualization spectrums. Numbered of boxes are correspond to various hardware design options & that referred to as region 1 though region 8.

Moreover, depending on the dynamics of the neural model network such a design could be inefficient with large amounts of hardware sitting idle most of the time. Previous work has shown that, with many possible future hardware technologies, only semi-virtualized hardware designs (regions 5–7) will scale[1]$_{NN}$ well. It should be noted that efficiency, which mostly effects cost as end in itself is not our only goal since it must be traded off with execution time & power density goals as well. Most neural system implementations are hybrid hardware & software designs which fall into regions 1 and 3. Designs from region 3 may not scale$_{NN}$ well as compared to ASIC designs from region 5 through 7, particularly in terms of area/volume, while computational resources and timing are the limiting factors in scaling$_{NN}$ the designs from regions 1, 2 and 4. An analog design from region 8 is fully parallel at the computational level. However, the expense of metal lines and the huge bandwidth discrepancy, intermediate and long range communication tends to be digital and multiplexed. So virtualization of communication resources for neural network is generally assumed by analog designs when scaling to very large systems. For most neural models and algorithms that designs in regions 5 through 7 are generally left unexplored or at best studied by only a few. Hence, these are regions that warrant further investigation. The input-activity on each neuron is distributed in the range of very low or zero input activity to high input activity. So average activity of that neuron being 1%. Hence, as the degree of virtualization for the hardware implementations is varied by changing the available hardware resources and various performance/price trade-offs are possible. Most of the work on spiking neural models can be categorized according to the hardware virtualization spectrum for a survey of existing work.

C. Digital Processing Node

Fig. 5shows a simple block diagram of a single digital Processing Node (PN), which includes some of the major arithmetic and memory components required for implementing the neuron model specified. The basic functioning of a PN is now summarized for each presynaptic input-event in the Input Events Memory (IEM) and the PSP value is obtained from the PSP-LUT (look-up table), based on the event-timing. The global event-index is translated to a local-index by a logical address transformation, which is then fed to the Connection Memory, which, in turn, points to the list of neurons forming a synapse with this particular input. For each synapse the weight and the PSP value is multiplied and then added to the previous membrane potential value. The new value is written to the Membrane Potential Memory (MPM) since that PN could be emulating multiple neurons. The membrane-potential is

[1] The subscript ‘NN’ refers to the scaling aspects of neuron/bio-inspired computational models (software or hardware), i.e. when ‘small-scale’ computational models are expanded to emulate the ‘large-scale’ human cortex, what are the resultant effects in terms of timing, computational resources, hardware area/volume, data/memory size, model complexity, etc? From the hardware perspective, we are more interesting in timing, silicon area, power density, and efficient utilization.

compared to a threshold and if exceeded an output event as spike along with the event-time is added to the Output Events Memory (OEM). Each record in the IEM and OEM has the format: event-index, event time[2]; and each record in the MPM have the format membrane potential value, refractory-time. The operations performed by the PN can be separated into various stages and pipelined, increasing the performance and utilization of the hardware the time required to service one synapse (t_{syn}) is then roughly proportional to the slowest pipe stage. The representation of events and the communication of these events between the PNs are assumed to be based on digital AER message transfer. The Source Memory (SM) of a PN has an address listing of which (internal[3]and external) neurons connect to the neurons within that PN.

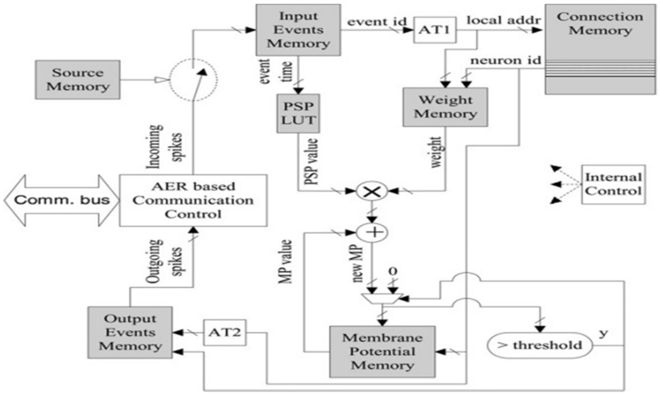


FIG: 5 Simplified block diagram of a digital processing node (PN).

External events/messages are received through the shared communication buses and by the AER communication controller of the PN an event is included in the IEM if its particular event address is present in the SM. Internal events (from the OEM of the PN) are included in the IEM (of the same PN) if their particular event addresses are present in the SM (of the same PN) that process occurs concurrently when each PN is broadcasting its events (from the OEM) to other (external) PNs.

Assume a simple model for the neuron given by,

$$yi(t) = \sum\forall j\ wij\varepsilon ij(t - tj)$$

Where, yi is the output of neuron i, εij(t − tj) is the postsynaptic potential at input j, and wij is the synaptic weight linking input j to neuron i.

D. Register Transfer Level of Computation Part of PN

See in figure 6, it is the RTL of computation part of PN. Main element of computation block is memory. In this RTL, there are seven memory blocks. These memory sizes are dependent on number of neurons in PN, Weight bit, post synaptic potential (PSP) bit, No of Synapses per PN, Time bit, No of PNs in cortex, Event Id, active neuron in PN, Number of active Synapses per PN. Here event id contains PN Id and Neuron Id who generate an event and time when it generates. Here serial number is given to each memory which is same as given to in RTL. Among seven memories only two memory, Post Synaptic Potential (PSP) and weight memory are constructed by block RAM (BRAM). All other memories are constructed by distributed RAM. Operation occurs in SNN is divided in to two parts: communication and computation. In communication operation, one PN broadcast its events which generated by its active neurons than second PN broadcast its events and so on. When one PN broadcast its events, other PNs receive these events. In computation operation, all PNs compute membrane potential of input event related neurons. All PNs compute membrane potential and check that membrane potential is greater than threshold value or not. If membrane potential of neuron is greater than threshold, it generates an event which store into output event memory. During computation operation, According to incoming event id from IEM, connection memory gives information about start and stop address of weight memory.

[2] Many neural systems are assumed to run in ‘‘real time’’ and do not have or need explicit event time information. For virtualized systems, the degree to which one can get away with this simplification depends heavily on the degree of virtualization. We are assuming here some basic event timing capability, since we are looking at a range of virtualization from one neuron per PN, which can probably leave out event timing information to all neurons on a single PN, which definitely needs a simulated event time.

[3] With respect to a PN, internal means local; external means non-local.

Here weight memory has information about weight of synapses and neuron id of present PN. From difference of current and event time, we Post Synaptic Potential (PSP) memory give value of post synaptic potential. Here multiplier gives product of weight and PSP value. According to neuron id of present PN, membrane potential memory gives membrane potential of that neuron.

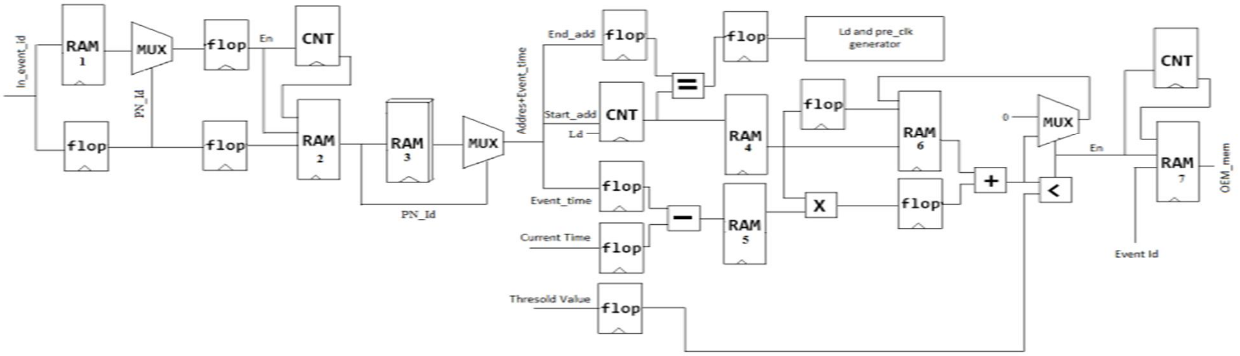
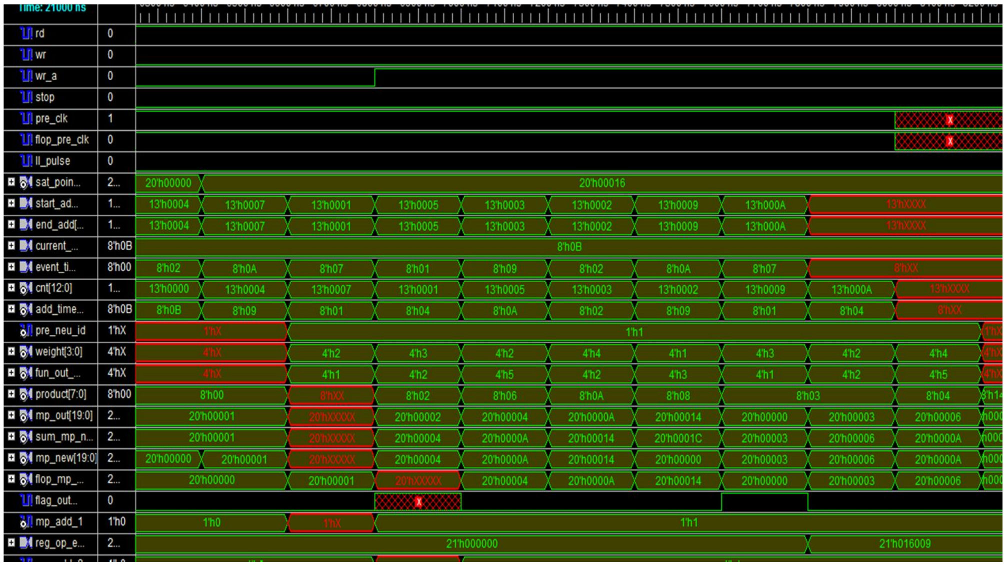


FIG: 6 Register Transfer Level (RTL) of computation part of PN

Next block is adder which add product term and membrane potential of neuron. If sum is higher than threshold value, new membrane potential of that neuron will be zero otherwise new membrane potential will be sum of old membrane potential and product term. If sum is higher than threshold value, event will be generated. This event contain PN id, neuron id which generate an event, and time when event generate. This output event stores into output event memory (OEM).

E. Simulation Result



V. INTRODUCTION TO FPGAS

Field programmable Gate Arrays (FPGAs) are pre-fabricated silicon devices that can be electrically programmed in the field to become almost any kind of digital circuit or system. For low to medium volume productions, FPGAs provide cheaper solution and faster time to market as compared to Application Specific Integrated Circuits(ASIC) which normally require a lot of resources in terms of time and money to obtain first device. FPGAs on the other hand take less than a minute to configure and their cost anywhere around a few hundred dollars to a few thousand dollars. Also for varying requirements of that the portion of FPGA can be partially reconfigured while the rest of an FPGA is still running. Any future updates in the final product can be easily upgraded by simply downloading a new application bit stream. The main advantage of FPGAs i.e. flexibility is also the major cause of its draw back. Flexible nature of FPGAs makes them significantly larger, slower, and more power consuming than their ASIC counter parts. These disadvantages arise largely because of the programmable routing interconnects of FPGAs which comprises of almost 90%of total area of FPGAs. But despite these disadvantages FPGAs present a compelling alternative for digital system implementation due to their less time to market and low volume cost.

A. Normally FPGAs Comprise of

1) Programmable logic blocks which implement logic functions.
2) Programmable routing that connects these logic functions.
3) I/O blocks that are connected to logic blocks through routing interconnect and that make off-chip connections.

VI. EXPECTED OBSERVATION RESULT

Table: Expected Observation Result

Layer	Number of 4 input LUTs	Number of BRAMs
1	5713 out of 9312	12 out of 20

VII. CONCLUSION

After completion of design of SNN, we observed effect of virtualization in SNN. There is proportional relation between virtualization and hardware utilization efficiency. We also observe that hardware utilization of PN depend on number of synapse and number of neuron per PN.

REFERENCES

[1] Spiking Neural Networks: Learning, Applications, and Analysis by Hesham H. Amin (Author), Paperback: 132 pages,Publisher: LAP LAMBERT Academic Publishing (July 1, 2011)Language: English, ISBN-10: 3845405155, ISBN-13: 978-3845405155.

[2] Neural Information Processing: Research and Development by Jagath Chandana Rajapakse, LipoWang,Springer, 06-Dec-2012.

[3] Introduction to Artificial Neural Networks Paperback – 20 Sep 2003by S. SivanandamPaperback: 236 pages,Publisher: Sangam Books Ltd; First edition (20 September 2003)Language: English,ISBN-10: 8125914250,ISBN-13: 978-8125914259.

[4] Verilog Digital System Design –RT Level Synthesis, Testbench and Verification, Zainalabedin Navabi, Ph.D. Professor of Electrical and Computer Engineering Northeastern University Boston, Massachusetts, Second Edition McGraw-Hill, New York Chicago San Francisco Lisbon London Madrid Mexico City, Sydney Toronto.

[5] A VHDL Primer ,Third Edition, J.Bhasker, Bell Laboratories, Lucent Technologies Allentown, PA, Pearson Education.

[6] Runchun Wang, Tara Julia Hamilton, Jonathan Tapson, André van Schaik,"An FPGA design framework for large-scale spiking neural networks", IEEE, The MARCS Institute, University of Western Sydney, Sydney, NSW,Australia-2014.

[7] Marco Aurelio Nuno-Maganda, Miguel Arias-Estrada, Cesar Torres-Huitzil," High performance hardware implementation Of spike prop learning potential and tradeoffs",IEEE, National Institute for Astrophysics, Optics and Electronics (INAOE)-2007

[8] M.J.Pearson, C.Melhuish, A.G.Pipe, M.Nibouche, I.Gilhesphy, K.Gurney, B.Mitchinson,"Design and FPGA implementation of an embedded real-time biologically plausible spiking neural network processor",IEEE,IAS laboratory ,University of the West of England Coldharbour Lane, Bristol-2005.