



ASIC Implementation for Improved Character Recognition and Classification using SNN Model

Mrs. S. Chaturvedi¹, Dr .A. A. Khurshid² Nirja. Karlewar³

¹Assistant Prof. & Head, Electronics, P.I.E.T., Nagpur (M.S.), India

²Prof. & Head, Electronics Dept., R.C.O.E.M., Nagpur (M.S.), India

³Research Scholar, Electronics, P.I.E.T., Nagpur (M.S.), India

Abstract: This paper depicts how Spiking neural network model is used for character recognition and classification. Here we adapt to the technique of using ASIC for large scale simulations of the Izhikevich model and use RTL Clock gating approach for reducing the dynamic power. In the current work Izhikevich model is designed & their performance parameters is measured. This Izhikevich model is recognized and classifies different characters. Here we describe how a spiking neural network model can be implemented on ASIC with 90 nm Process. The Izhikevich spiking neuron model is best suited for large scale cortical simulations due to its accuracy, efficiency, power and simulation time.

Keywords: Application specific integrated circuits (ASIC); Spiking neural network (SNN); Artificial neuron network (ANN); ordinary differential equations (ODE); Izhikevich; Pattern recognition; Pattern classification.

I. Introduction

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. 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.

II. Clock gating

A significant fraction of the dynamic power in a chip is in the distribution network of the clock. Up to 50% or even more of the dynamic power can be spent in the clock buffers. This results makes intuitive sense since these buffers have the highest toggle rate in the system, there are lots of them, and they often have a high drive strength to minimize clock delay. In addition, the flops receiving the clock dissipate some dynamic power even if the input & output remain the same. Modern design tools support automatic clock gating; they can identify circuits where clock gating can be inserted without changing the function of the logic.

The most common way to reduce this power is to turn clocks off when they are not required. This approach is known as clock gating. In the original RTL, The register is updated or not depending on a variable (EN). The same result can be achieved by gating the clock based on the same variable. If the registers involved are single bits, then a small savings occurs. If they are, say, 32 bit registers, then one clock gating cell can gate the clock to all 32 registers (and any buffers in their clock trees). This can result in considerable power savings. Much of the power savings was due to the fact that the clock gating cells were placed early in the clock path.

III. Artificial Neural Network (ANN)

Artificial Neural Network is a system closely modeled on the human brain. An artificial neuron is an information-processing unit that is fundamental to the operation of a neural network. ANNs are the neural network consisting of simulated neurons. The neurons in the network are artificial hence the name Artificial Neural Network. The network processing is parallel in nature; hence the network does not fail. Artificial Neural Network (ANN) is non-linear data driven self adaptive approach as opposed to the traditional model based method. They are powerful tools for modelling, especially when the underlying data relationship is unknown.

An ANN is defined as data processing system consisting of a large number of simple highly connected processing elements (artificial neurons) in architecture inspired by the structure of cerebral cortex of brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron

to the input of another. ANN can learn and identify correlated patterns between input data set and corresponding target values.

IV. Spiking Neural Network (SNN)

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.

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. The brain utilizes a large collection of slow neurons operating in parallel to achieve very powerful cognitive capabilities. For the design of neural circuits there are several researchers are examining the use of Memristors. Izhikevich [5] has shown that the integrate-and-fire spiking neuron model is not very biologically accurate and is also unable to reproduce the spiking behavior of many neurons. Both the integrate-and-fire model and the Izhikevich model require 13 FLOPs (Floating Point Operations) per neuron simulation, while the Hodgkin- Huxley model requires 256 FLOPs. Therefore, for large scale simulations the Izhikevich model is significantly more desirable.

Recent studies have implemented the Izhikevich neuron model instead of the integrate-and-fire model on FPGAs. In this we explore the feasibility of using FPGAs for large scale simulations of the Izhikevich model. We utilized a character recognition algorithm based on the Izhikevich spiking neuron model

V. Recognition System

Recognition and acquire knowledge through sender perception are very much related. Pattern recognition consists of recognizing a pattern using a machine (computer). It can be defined in several ways

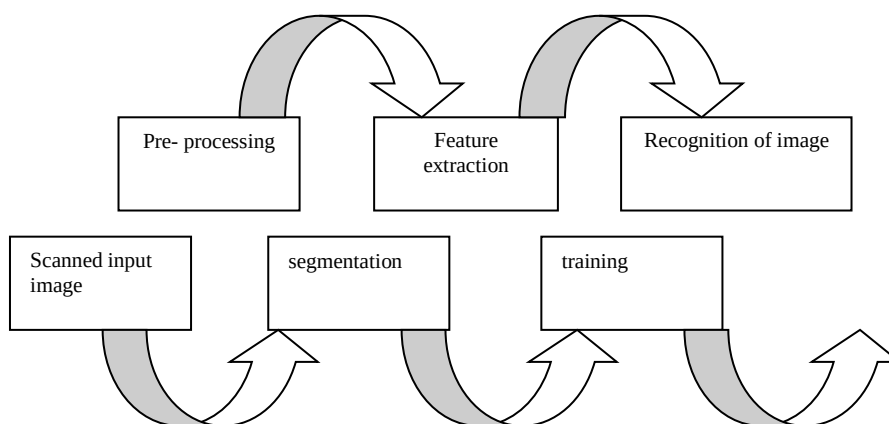


Figure 1 Diagrammatic Representation of Pattern Recognition System

The goal of pattern recognition is the classification of objects into a number of categories or classes. Depending on the application, these objects can be images, signal waveforms, or any type of measurements that need to be classified. Pattern recognition has a long history, but before the 1960s, it was mostly the output of theoretical research in the area of statistic. As with everything else, the advent of computers increased the demand for practical applications of pattern recognition, which in turn set new demands for further theoretical developments. As our society involves from the industrial to its post industrial phase, automation in industrial production and the need for information handling and retrieval are becoming increasingly important. This trend has pushed pattern recognition to the high edge of today's Engineering applications and research. Recognition is an integral part in most machine intelligence systems build for decision-making.

COGNITION: - To become acquainted with, to come to know the act, or the process of knowing an entity (the process of knowing).

Recognition = Re + cognition

VI.Character recognition in SNN

The SNN architecture uses a computationally very simple and efficient spiking neural model, in which early spikes, received by a neuron, are stronger weighted than later ones. The model performs a very fast image processing and was inspired by the neural processing of the human eye so, it is known that neural image recognition involves several succeeding layers of neurons, these experiments suggested that only very few spikes could be involved in the neural chain of image processing. face recognition tasks, reporting encouraging experimental results tested by a mathematical definitions of these neurons

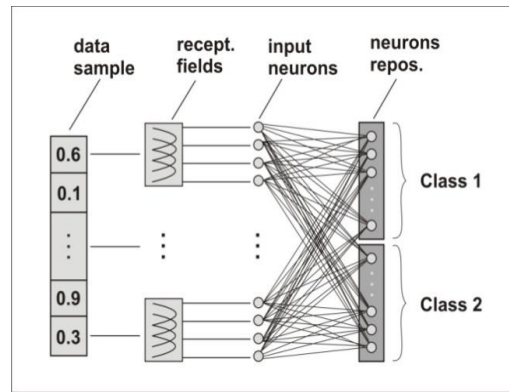


Figure 2 Character recognition in SNN

The neural model is given by the dynamics of the post-synaptic potential (PSP) of a neuron i .

0 if fired

Else 1

$$PSP_i(t) = \begin{cases} \sum_{j|f(j)<t} w_{ji} m_i^{order(j)} & \text{min} \\ 0 & \text{if fired} \end{cases}$$

Where w_{ji} is nothing but the weight of a pre-synaptic neuron j , $f(j)$ is the firing time of j , and m_i (0,1) a parameter of the model, namely the modulation factor. Function $order(j)$ represents the rank of the spike emitted by neuron j . For example a rank $order(j) = 0$ would be assigned, if neuron j is the first among all pre-synaptic neurons that emits a spike. In a similar fashion the spikes of all pre-synaptic neurons are ranked and then used in the computation of PSP_i . A neuron i fires a spike when its potential has reached a certain threshold θ . After emitting a spike the potential is reset to $PSP_i = 0$. Each neuron is allowed to emit only a single spike at most. The threshold $\theta = c \cdot PSP_{max}$ is set to a fraction c (0,1) of the maximal potential PSP_{max} possible by a neuron. In Figure 1 the change of the PSP for this neural model is presented, when a series of input spikes are presented to the different synapses of this neuron.

During the presentation of training samples the method successively creates a repository of trained output neurons. For every training sample a new neuron is trained and after that compared to that which already stored in the repository. If a trained neuron is found to be too similar (in case of its weight vector) to that which is in the repository (according to a specified similarity threshold s), the neuron will be merged with the most similar one. In other case the trained neuron is added to the repository as a new output neuron. Another approach is the population encoding which distributes a single input value to multiple neurons and they may cause the excitation and firing of several responding neurons. Receptive fields allow the encoding of continuous values by using a collection of neurons with overlapping sensitivity profiles. For every input variable is encoded independently by a group of M one dimensional receptive fields.

VII. Inserting Clock Gating in the RTL generated Izhikevich spiking neuron model using Synopsys Power Compiles

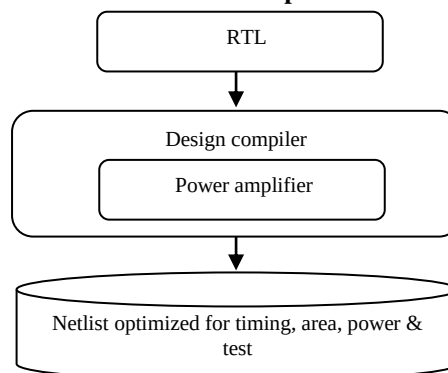


figure 3 Design flow for power synthesis within Design Compiler

Steps to insert clock gating logic to synthesize the design with the clock

- The first step is to understand the RTL design.
- The next step is to utilize the compile _ultra -gate Clock command to compile design.

While this method, the compiler have to execute the following tcl commands:

Optional setting:

- read_neuronmodel.v

- create_clock -period 16 -name CLK
- compile_ultra -gate_clock -scan
- insert_dft
- report_clock_gating
- report_power

VIII. Power Optimization of the generated Izhikevich spiking neuron model using the design Compiler

synchronous load-enable registers applies Clock gating ,which are nothing but the groups of flip-flops that share the same clock and synchronous control signals and that are inferred from the same HDL variable. Synchronous load-enable, synchronous set, synchronous reset, and synchronous toggle includes in the synchronous control signal. By use of feedback loops the registers are implemented by Design Compiler. Sometimes, through multiple cycles these registers maintain the same logic value and unnecessarily use power. By eliminating the unnecessary activity associated with reloading register banks clock gating saves power. Designs that benefit most from clock gating are those with low-throughput data paths.

VIII. Experimental Result

Post synthesis Results	
Area	164301 squm
Power	3/2 mw

IX. Layout with IC Compiler

The layout of the Izh SoC is initiated by the help of IC Compiler after the gate level netlist is effectively created by the Design Compiler,

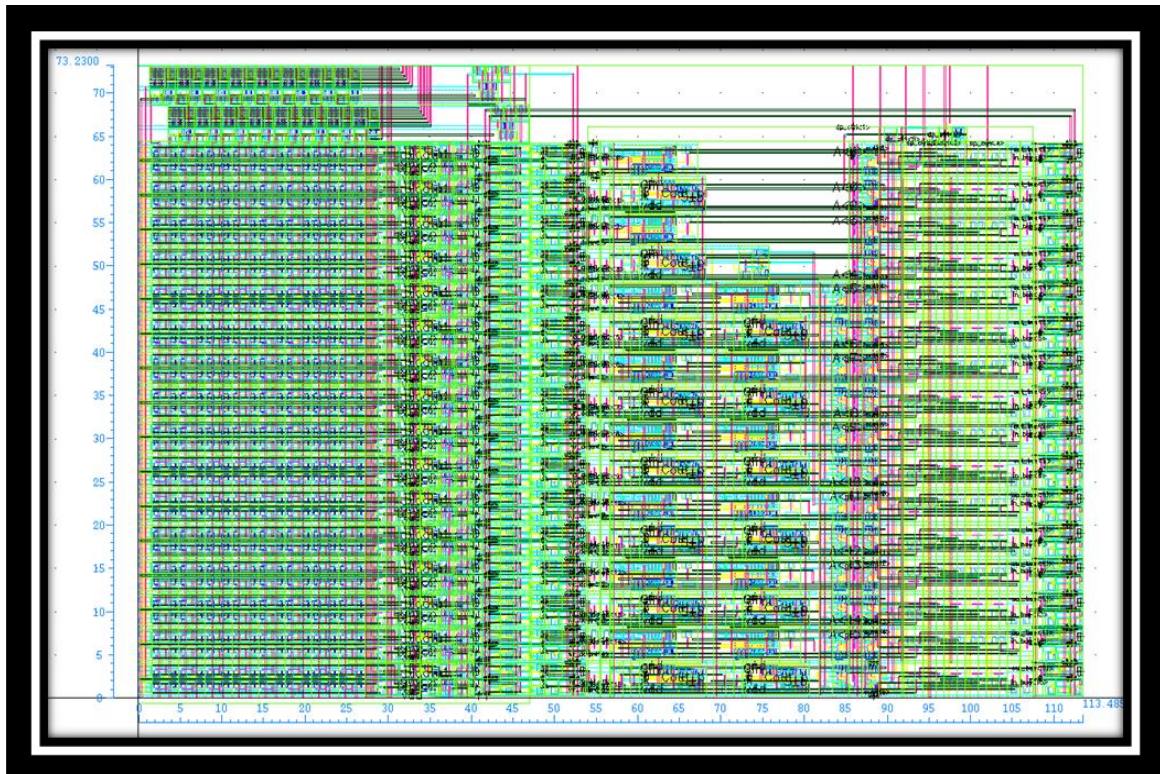


Figure 4 Layout of the generated Izhikevich spiking neuron model

Steps to set up the IC Compiler Environment:

Step 1: The initial step is to setup the suitable and correct link and target libraries. Afterwards the Milkyway reference library is supposed to be created which serves as the internal specific database arrangement which utilized by IC Compiler to accumulate all the physical and logical information about a design to enable accurate RC extraction results, the TLUPLUs file is required.

Step 2: The TLUPlus file puts into a table all the RC coefficients in a specific binary form. Second essential key is the tech2itf file which is supposed to match the layer and via names in the Milkyway technology file. Basically all via names get matched with the names with the ITF file (Interconnect Technology Format)

References

- [1] A. Rosado-Munoz, A.B. Fijalkowski, M. Bataller-Mompean, J. Guerrero-Martinez, "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] Eugene M. Izhikevich "Which Model To Use for Cortical Spiking Neurons" IEEE Transaction on Neural Networks, Volume 15, No.5, September 2004.
- [3] Jilles Vreeken "Spiking neural networks, an introduction".
- [4] Prof. Dr. Karlheinz Meier, Prof. Dr. Rene Schuffny "VLSI Implementation of a Spiking Neural Network"
- [5] Eugene M. Izhikevich "Simple Model of Spiking Neuron", IEEE Transaction on Neural Networks, Volume 14, No. 6, November 2003.
- [6] C. Panchev, S. Wermter, and H. Chen, "Spike –timing dependent competitive learning of integrate-and-fire neurons with active dendrites," in Proceedings of the International Conference on Artificial Neural Networks, August 2002, pp. 896-901.
- [7] Wysoski, S. G., Benuskova, L., and Kasabov N. (2008). Science Discret Neurocomputing, Fast and adaptive network of spiking neurons for multi-view visual pattern recognition, 71,pp 2563-2575.
- [8] Roberto A. Vazquez "Izhikevich Neuron Model and its Application in Pattern Recognition".
- [9] Ankit Sharma, Dipti R. Chaudhary "Character Recognition using Neural Network", International Journal of Engineering Trends and Technology (UETTE)- Volume 4 Issue 4-April 2013J.
- [10] Yihua Liao "Neural Networks in Hardware: A Survey".
- [11] Vinita Dutt, Sunil Dutt "Hand Written Character Recognition Using Artificial Neural Network"2011.