FPGA-BASE DIGITAL NEUROMORPHIC PROCESSOR AND VARIOUS PARALLEL ARCHITECTURES

[1]P.Rahul Reddy,Research Scholar, [2]Dr.Albert Raj A, Supervisor
Faculty of Electronics and Communication Engineering
Anna University, Chennai, 600 025

Abstract

This work proposes an efficient design methodology based on a new theory of measuring a performance of the computational complex recurrent reservoirs. A reconfigurable reservoir pre-processor with task-dependent powered of a gate is proposed to develop the energy effectiveness. Furthermore, we enable this LSM processor to perform the firing activity based power gating for each particular task. The other architecture is developed for a feed-forward SNN with STDP learning rule, which performs the neuron dynamics in parallel. Meanwhile, both of the proposed architectures examine current application of estimated compute in neuromorphic systems, or demonstrate reduction of energy consumption without introducing significant learning performance degradation.

Keywords: FPGA, Spiking Neuron Network, Neuromorphic Processor

1. Introduction

FPGAs offers the more flexible or reconfigurable for quick prototype or hardware accelerate the software algorithm. To facilitates the applications of SNNs in implanted system & develop process accelerating for bulky data set, there has a several attempt to implemented software algorithm here FPGA [1]- [2]. Mean while, due to their much shorter development period compared with ASIC designs, the FPGAs are widely used in the data centers of companies such as Microsoft and Amazon. Therefore, the digital neuromorphic architectures in this section are also based on the FPGA platform.

2. Two-layer Spiking Neuron Network

the LSM learning processor mentioned earlier, we also propose a parallel neuromorphic learning system for a 2-layer spiking neural network with global inhibition, which is tuned by the STDP learning rule. To demonstrate the performance, we using the propose architecture to solving a hand written digital recognize problems by image for MNIST, a popular people domains datasets of handwriting digit's by the 28x28 decision. MNIST involve 60,000 image for trains or 10,000 image to recognitions. Every 28x28 images is converter onto a model by 28x28 pixel, which is uses to generates the other inputs spike to the inputs layers of the spikes neuralnetwork. In orders the obtained acceptability performances of these particularly testing benches, is instantiates the spike neural networks by 784 excitatory neuron input layers or 800 exciting neuron on a outcome layers, when illustrate with Fig. 1. Present is also 6 inhibit neuron in the input layers or 1 inhibitor neurons in the outcomes layers. The purpose of this global inhibition is to recognize the winner-take-all (WTA) mechanism inside each layer.

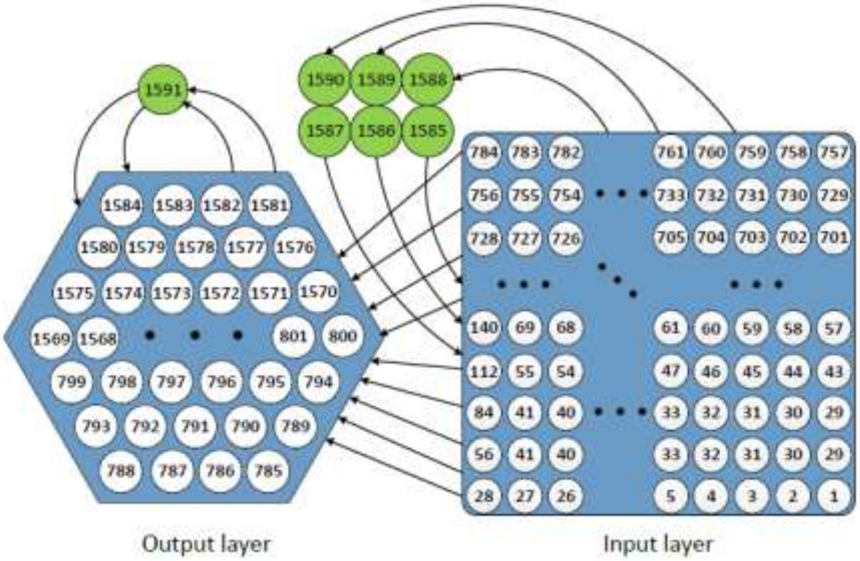


Figure 1: The labeled 1-784 neurons is excitatory from the input, as the neuron label 785-1584 is a exciting neuron on outcomes layers. The inhibitory neuron with other neurons.

3. Serial Baselines of Neuromorphic Processor Architectures

To describes in detailed the propose neuromorphic PC architectures for the 2-layers spikes neural networks. Several critical issues those are memory organizations, efficient parallel process or the applications of Fig. 2 demonstrate the basic lines structural design of the propose neuromorphic PC. A syntactical parameter are *W*, *A*+ or *A*− is store at blocks of RAM. The synaptic weight is reads outs of the BRAMs sequential & A casing potential which is record the Neuron Units were updating one by one.

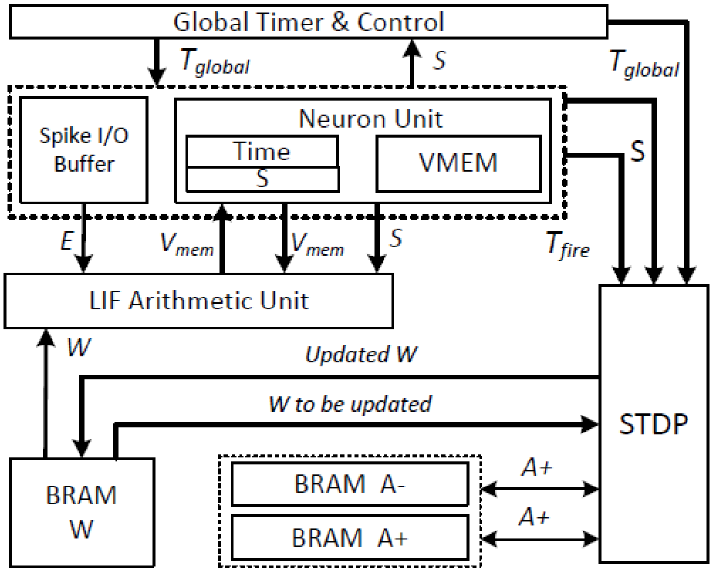


Figure 2: The diagrams of a serial blocks based lines architectures not by a parallel compute. A synaptic weight W is storing at a singleport blocks RAM, or syntactical parameter $A+$ or $A-$is storing on other 2 blocks of RAM.

Fig. 3 show detail designing of Neuron Units (NU) or the LIF Arithmetical Units (LAU). The NU involve 3 most require register file is storing a membrance potential (V_{mem}'s), the fired time ($T_{f\ ire}$'s) or the fired activity flag (S's) of all the neuron. In the NOS, the LAU firstly read out the V_{mem} or the S's for the NU, A syntactical weight for BRAM or another inputs spike for the spike I/O buffering, or write updates V_{mem} return to NU.

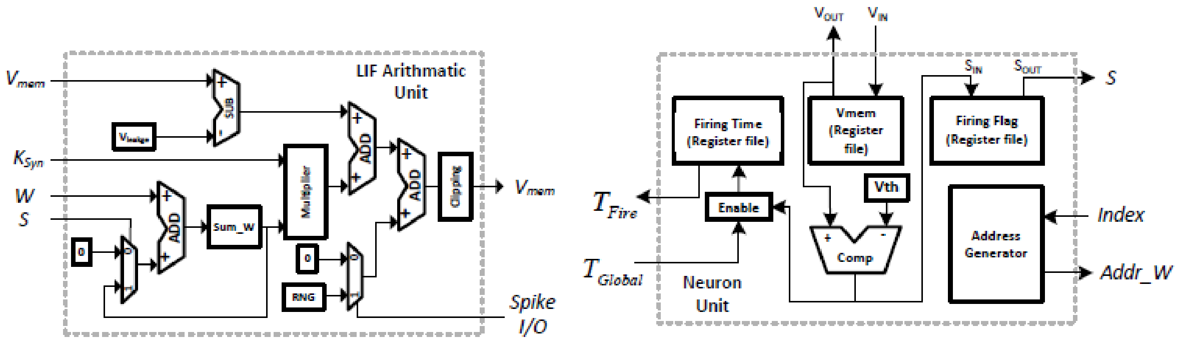


Figure 3: A propose LIF Arithmetical Units (LAU) or NeuronUnit (NU). LAU is uses are updated a membranes potential of whole neuron. NU is use to storing the membranes potentials, each neuron fire times or firing activity flags. Synaptic weight is store on the BRAM.

Fig. 4 show the designing detail of the propose STDP units, which is using the updated the synaptic weight base on the differences of fired time among the presynaptic The post synaptic Neuron and neuron Assume there is N_{output} neuron on the outcomes layers or N_{input} neuron on input layers, All numbered of the artificial synapse is exist update $N_{output} \times N_{input}$. Every plastic synapses is associate by 2 parameter A_{+} & A_{-} which might depends of a present state of synapses. To changing of syntactical weights W is calculate by this 2 parameter in the LOS. The exponential functions using updated the A+ or A− is realize a precomputing value find at tables.

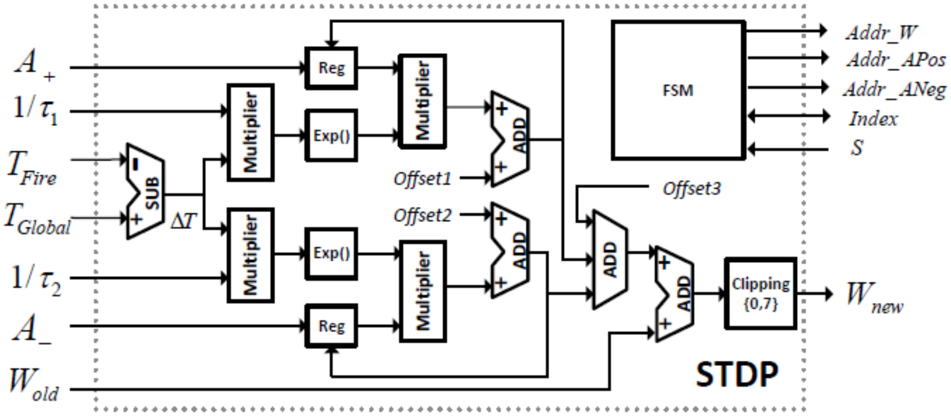


Figure 4: The propose STDP units which are use to updating the synoptically weight. $T_{f\ i}$ or S is obtain by the neuron units. W , A_{+} &A_{-} is of the BRAM.

To propose neuromorphic process have two operating in modes, namely, the preparation modes or the recognize modes. The recognition modes are simpler than the train modes because the synoptically weight needs doesn’t update through recognitions. The NU, LAU or the BRAM of a syntactical weight is reuse on the recognition modes, which lead to visible reduces the area overhead as no addition of function blocks are adding.

4. Propose Parallel Architecture & Memory Organization

Various parallel architecture proposing. In Figure. 4.19, W (j, i) represent a weights of synapses of the jth neurons to ith neurons. Assuming that there is N input layers neuron or M outcome layers of neuron. The neuron on the input layers is label as of 1- N , or the neuron in the outcome layers is label for N + 1 to N + M. Fig. 5 show a parallel architectures should support Kway parallel process due to NOS wherever K membranes potential is updating simultaneously. The process might take lots of clock cycle to completing since used for every updates many pre-synoptically weight needs to be reading out in sequences.

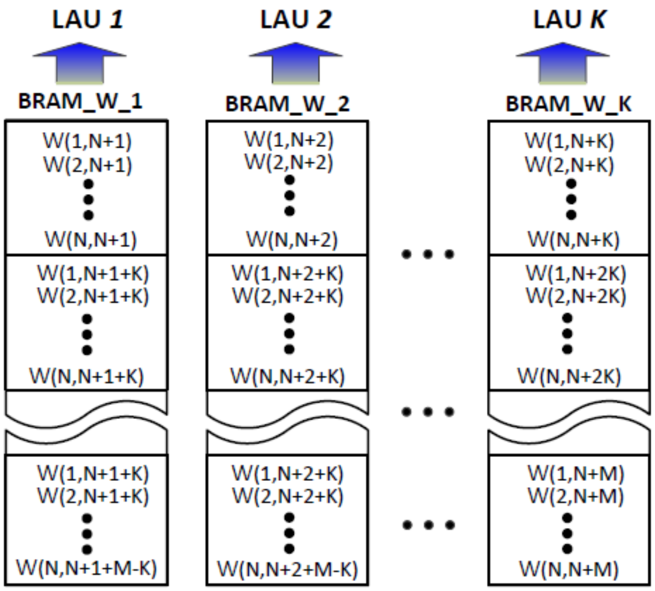


Figure 5: Parallel process scheme for *N* excited neuron at incoming layers or *M* excitatory neuron are at outcome layers: instantaneous update of *K* membrane potential by synaptic weight store at *K* parallel blocks RAMs.

The parallel architectures illustrate by Fig. 6. The weighted synapses inputs layers to outcome layers is store on *K* blocked RAMs. The weight associate by each outcome layers neurons whole similar blocks RAM. Ideal, The work loaded of the NOS is more balance, every LAU perform A V_{mem} updating of *M/K* excitatory neuron at outcome layers or *K* LAUs working in parallel. The V_{mem} updated another neuron are parallelize by similar ways. While All capability of registers of file (V_{mem}, S and $T_{f\ ire}$) inside the neurons units (NU) remain the similar, multiples of data port is create the NU to enables the *K* LAU are accessing data at insight of NU on parallel.

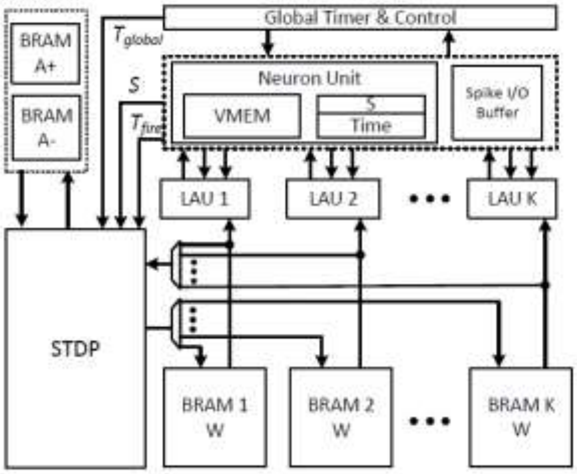


Figure 6: The propose diagonal neuromorphic process should develop *K* - way parallel process base in LIP. *K* blocks RAM is use to stored synaptic weight, or *K* LAUs worked at parallel to updates of K membranes potential by similar timing.

5. Experimental Results

5.1 Design Platform

By using XILINX synthesis tool the proposed neuromorphic process is design in Verilog HDL or synthesized . A Xilinx ML605 Evaluate Boards, make the uses of a FPGA Virtex 6 cores, have to be employee in order to developing or our designs are test.

The Whole experiment platforms of these neuromorphic systems are shows at Figure. 7. The Matrix laboratory programs in PC convert the trained pattern to spikes sequence or sending to a Xilinx's ML605 estimate boards thoughts on UART (universalinduction receivers/transmitters) cables. Formerly the train finish, result (i.e. outcome spike or synoptically weight) is sent return to PC by similar cables. The propose FPGA-base neuromorphic processing are compose 3 main component: Neuron Units, an array of LIF (Leakage- Integrates-or-Firing) Arithmetical Units, or STDP Units. The syntactical weight of synthetic synapse (i.e. the snaps as to excited layers to outcome layers) is store on the block RAM (BRAMs) on the FPGA chips. The read /write interface synapse by realizing BRAMs through access. We follow the usual FPGA designing flows to performs functional simulations, logics separation, placing & route, or generating the configurations bit streams. According to the time study conduct since parts of synthetic flows, are propose neuromorphic process is capable to runs with 133.288 MHz We employee a MMCM (Mixing Modes Clock Managers) blocks generating the real clocks rates that is 120MHz. The propose design is synthesize in a hierarchical/bottoms- up manners, to allows straightforward reusing of basic line build the block of a Neuron Units, LIF Arithmetical Units or STDP Units amongst target architecture variant. In arrange to propose architecture are communicating by Mat- lab programs runs on PC, we also implements a UART(Universal induction Receivers/Transmitters) to supports series communiqué. The Energy consume of every architectures obtain with use XPower Analyze, which offer detail power study of the design of Xilinx's FPGA.

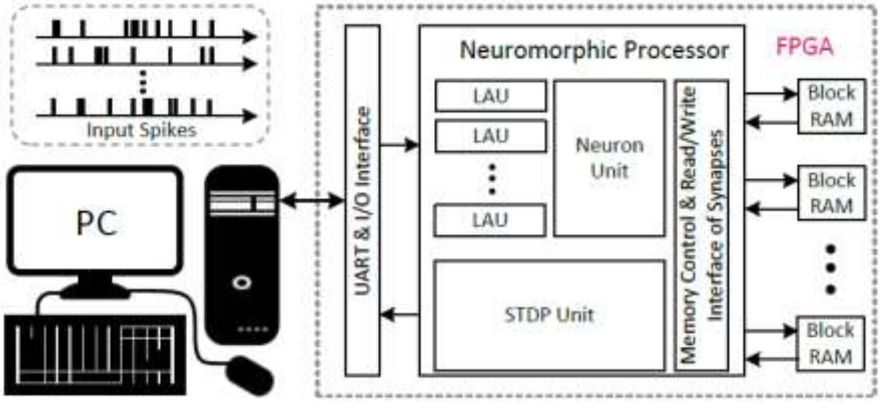


Figure 7: Top-level representation of the propose neuromorphic process runs at Xilinx ML605 estimate board, by the synaptic weight store on blocks RAM. The communication among PC or FPGA is realize with an UART cables.

5.2 Tradeoffs between Power, Energy and Hardware Overheads

Table 4.7 list the power consume or little using of every builder block on the basicline serial neuromorphic process designs. The membrane potential of all neuron is update one by one. The summary of the synaptic weight of every is realize with a single accumulators, those are consistent by Fig. 1. The slice utilize of every builder blocks is obtain behind places or routes. The power of the building block is obtain of a XPower An-analyzing (XPA) or Xilinx Powers Estimate (XPE), 2 commercial tools are analysis powers. While the clocked frequencies are 120MHz neuromorphic process, the real switch frequencies of every build blocks much lessthan of a major systems clocks. So, the Neuron Units have lower energy consume the numbers of using of flip-flops are largely. The designing of a information of basic line designs with not or by the approximate multiplier is shows at Tables 1(a) or Table 1(b), respective. As compare those variant of base line series designs, we simply see approximately adoption multiplies and help to reducing both power consumptions or slice uses.

	Slices LUTs	Slice FFs	Powerunits(mW)
LIF Arithmetic	626	96	2.98
Neuron Unit	69,265	50,688	1.07
STDP Unit	2,352	199	10.44
Glue Logic	68	16	0.55
Block RAM	2,508,800 of 4nibbles for *W*		0.39
Block RAM	5,017,600 bits for *A*+		0.48
Block RAM	5,017,600 bits for *A*−		0.48

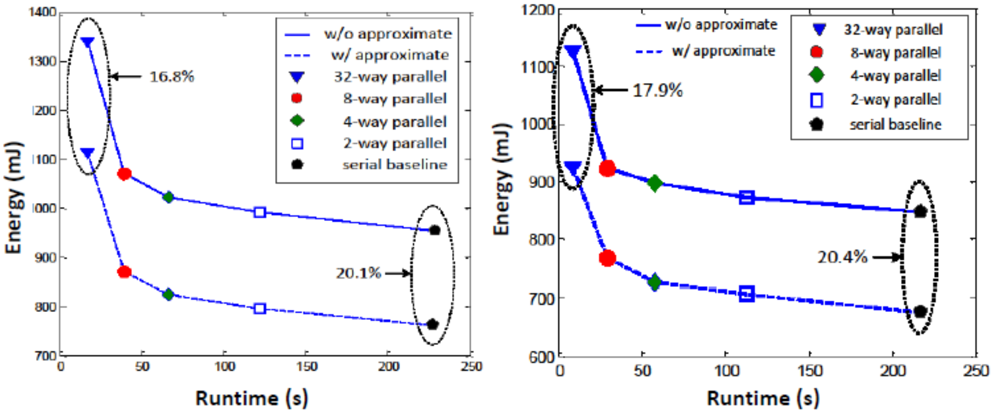
(a) The basic lines of designs by standard booth multiplier

	Slices LUTs	Slice FFs	Powerof units(mW)
LIF Arithmetic	516	83	2.21
Neuron Unit	69,265	50,688	1.07
STDP Unit	1817	134	7.71
Glue Logic	68	16	0.55
Block RAM	2,508,800 bits for *W*		0.39
Block RAM	5,017,600 bits for *A*+		0.48
Blocks RAM	5,017,600 bits for *A*−		0.48

(b) The baseline design with approximate multipliers

Table 1: Powered or resources utilize of the builder blocks component of the baselines architectures, which accumulate the pre-syntactical weight serial of every outcomes of layers neurons. All the neuron is process one with one in a sequential manner.

Fig. 8 compare this design by respective to runtimes or energy consume. A degree of parallel increase, the runtimes get shorter or less however the energies consume go up. These are since addition resources or powered overhead is introduce to supports parallel process. The energy improving introduce with the approximate multiplier can reaching up to 20.1% for the serial designs.



(a) Training

(b) Recognition

Figure 8: Compare the various designing in relations of runtime or energy consume. The soiled curves represent the design uses standard booth multiplier. The dashed curves represent the design use the approximate multiplier. (a) is for the training modes, while (b) is for the recognition mode.

6. Conclusion

In these working, currently on FPGA-base digitally neuromorphic process or various diagonal architecture. The propose architecture successful addressing some critical issue affecting the efficiently parallel in membranes energy consume, on-chip storing of synoptically weight, & integrate to estimate a arithmetical unit. The trades-off among through put, hardware costs or powers overhead of unusual configuration has be carefully investigate. A promise trains accelerate 13.5x recognition to speedup of 25.8x is achieve with a parallel designs whose degrees of parallelism is 32. The whole trains are speedup provide with the 32-ways of diagonal designs runs with 120 MHz more the series software model run at 2.2 GHz CPU are 59.4x. Up to 20% reduce the power consumptions is achieve once use the estimated multiplier in our baseline process designs, as to maintain appealing most of similar levels to recognitions presentation of hand written digital recognize.

7.References

1. D. Anguita, A. Ghio, L. Oneto, S. RIDELLA. *In-sample and out-of-sample model selection and error estimation for support vector machines* IEEE Trans. on Neural Networks vol.23, no.9, pp.1390,1406, Sept. 2012

2. D.Anguita, S.Ridella, F.Rivieccio, R.Zunino. *Hyperparameter design criteria for support vector classifiers* Neurocomputing Vol 55, No. 1-2, pp. 109-134, 2003.

3. [82] Kaibo Duan, S. Sathiya Keerthi, and Aun Neow Poo. *Evaluation of simple performance measures for tuning SVM hyperparameters.* Neurocomputing 51 (2003): 41-59.

4. Intel Open Source Technology Center, PowerTop 2.0, 2007.

5. Uzilov, Andrew V., Joshua M. Keegan, and David H. Mathews. *Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change.* BMC Bioinformatics 7.1 (2006): 173.

6. [85] Chang, Chih-Chung, and Chih-Jen Lin. *LIBSVM: a library for support vector machines.* ACM Trans. on Intelligent Systems and Technology (TIST) 2.3 (2011): 27.

7. Upegui, Andres, Carlos Andrs Pea-Reyes, and Eduardo Sanchez et al. *An FPGA platform for on-line topology exploration of spiking neural networks*. Microprocessors and *M*icrosystems 29.5 (2005): 211-223.

8. Pearson, Martin J., et al. *Implementing spiking neural networks for real-time signal-processing and control applications: a model-validated FPGA approach*. Neural Networks, IEEE Transactions on 18.5 (2007): 1472-1487.

9. Rice, Kenneth L., et al. *FPGA implementation of Izhikevich spiking neural networks for character recognition*. Reconfigurable Computing and FPGAs, 2009. ReConFig'09. International Conference on. IEEE, 2009.

10. Thomas, David B., and Wayne Luk. *FPGA accelerated simulation of biolog- ically plausible spiking neural networks*. Field Programmable Custom Com- puting Machines, 2009. FCCM'09. 17th IEEE Symposium on. IEEE, 2009.

11. Y. Zhang, P. Li, Y. Jin, and Y. Choe. *A digital liquid state machine with biologically inspired learning and its application to speech recognition* IEEE Trans. on Neural Networks and Learning Systems 2015