

A Cellular Structure for Online Routing of Digital Spiking Neuron Axons and Dendrites on FPGAs

Hooman Shayani¹, Peter Bentley¹, and Andy M. Tyrrell²

¹ Dept. of Computer Science, UCL, London WC1E 6BT, UK

² Department of Electronics, University of York, York, UK

Abstract. As a step towards creating evolutionary developmental neural networks on FPGAs, a bio-inspired cellular structure suitable for online routing of axons and dendrites on FPGAs based on a new digital spiking neuron model (introduced previously by the authors) is proposed here. This structure is designed to allow changing the routing of the dendrites and axons and formation/elimination of synapses on the fly by dynamic partial reconfiguration of the LUTs. The feasibility and techniques for implementing this structure on a Xilinx Virtex-5 FPGA are also studied.

Keywords. FPGA, Dendrite and Axon Growth, Synapse Formation, Evolvable Hardware, Cellular, Digital Spiking Neuron Model.

1 Introduction

The idea of creating a small adaptable, robust, fault-tolerant and intelligent brain in silicon has been around for more than a decade. However, the realization of this dream may require a better understanding of how real brains work. As new peculiarities of the brain unfold every day (e.g. [1]), the need for an alternative approach becomes more clear. Inspired by nature, many researchers resorted to use evolutionary computing to create such artificial brains.

Numerous different methods to evolve artificial neural networks [2] have been introduced, which were more or less successful in creating intelligent systems. However, oversimplification of natural processes in these approaches may have impaired the system as including some of these complexities leads to higher performance and emergence of new capabilities. The higher computation power of spiking neural networks compared to the traditional neural networks [3,4] and the intrinsic ability of recurrent spiking neural networks to process temporal patterns [5] both indicate that including a spiking mechanism with all of its complexities is rewarding. The positive impacts of introducing developmental processes on emergence of robustness [6], scalability [7], regeneration, and fault-tolerance [8] in neural networks all imply that incorporating some of these complexities may improve the final performance of the system.

Inclusion of these complexities, on the other hand, requires a huge amount of computational power that makes an evolutionary approach intractable. For example evolving a developmental spiking neural network involves iterative nested

loops of evolution, development, simulation and learning over a diverse training set from a problem class at differing timescales. This is particularly problematic in an experimental setting where there are dozens of parameters to tune and many different techniques to investigate. It even seems impossible given that nature has accomplished the equivalent to this through billions of years of evolution, employing huge number of processing elements, optimizing each and every system and process from scratch. The only ray of hope is to adopt the right combination of natural processes and imitate this subset of nature at a sufficient level of detail. Evolvable hardware [9] may enable us to exploit computational resources at a lower level, leading to fine-grained system interactions, low-level parallelism, and a biologically more plausible approach compared to traditional evolutionary computation.

Two major subtleties of this kind are the particular structure of the brain and the malleability of its structure in response to new problems and changes in the environment. Evidence suggests that structural plasticity [10] and wiring delays [11] play major roles in the brain and the placement and wiring of the neurons are optimized for the high interconnectivity in the brain [12]. However, the existing evolvable hardware neural network models (*e.g.* [7,13]) are not capable of regeneration and dendrite growth on FPGAs. They are typically either constricted in terms of number of inputs per neuron or impose constraints on the patterns of connectivity and/or placement on the actual chip mostly due to implementation issues. They also do not allow heterogeneous networks with flexible parametric neurons and learning rules as important bio-plausible features.

Following a nature-inspired approach, here we propose a cellular structure for online routing of axons and dendrites on FPGAs (Field-Programmable Gate Arrays), based on a new digital spiking neuron model introduced by the authors [14], aiming at: flexibility and evolvability, development-friendliness, high simulation and learning speeds, parallelism, and bio-plausibility, while having hardware implementation in mind. This work is another step towards creating adaptable and bio-plausible FPGA-based spiking neural networks using evolutionary, developmental and learning processes.

In the next section, the relevant literature is reviewed. In section 3, the digital spiking neuron model that is used in this work and its advantages are summarized. The new cellular structure is introduced in section 4 along with a comprehensive example. The feasibility of implementation of this cellular structure in Xilinx Virtex-5 FPGAs and possible reconfiguration techniques are discussed in section 5. The paper is concluded and possible future directions are mentioned in section 6.

2 Background

Cellular Computing is a vast field of study based on computation using locally-connected multi-dimensional arrays of simple processing elements. It includes (but is not limited to) the study of Cellular Automata (CA), Cellular Neural

Networks (CNN) [15], other cellular complex systems, evolving them and applying them to different problems such as image processing, pattern recognition and vision [16], control, random number generation, etc. The cellular nature of FPGAs (Field-Programmable Gate Arrays) [17] lends itself to high-speed parallel implementation of cellular systems.

The seminal work of Thomson [18] with a cellular structure on Xilinx XC6264 not only revived the field of intrinsic evolvable hardware, but also showed the power of evolutionary cellular systems on FPGAs. Different cellular developmental systems for FPGAs have been designed by Haddow and Tufte, Liu, Miller and Tyrrell, and many others that are reviewed in [7]. Upegui and Sanchez [19] used a cellular structure to evolve random boolean networks on a Xilinx Virtex FPGA and introduced a new method for dynamic reconfiguration of Virtex FPGAs based on direct bitstream manipulation. In another work, Upegui *et al.* evolved a 3-layer recurrent spiking neural network on Xilinx Spartan FPGA. However, the number of neurons and synapses, general architecture of the network and the neuron parameters were fixed during the evolution. Cellular systems have been used for development (and simulation) of neural networks and sought to be implemented in FPGA by de Garis *et al.* [20].

One of the interesting studies on evolving developmental neural networks on FPGA is the work of Roggen [7]. He also presented a comprehensive review of the developmental systems in evolvable hardware and introduced a new classification criteria for developmental systems noting that most of the advantages of developmental systems lie in cellular online developmental systems implemented in hardware. He introduced such a cellular development system for evolvable hardware and used it to evolve neural networks for pattern recognition and robot navigation. However the connectivity patterns of the neurons and the neuron model were limited to 6 fixed patterns and a simplistic leaky integrate and fire soma model.

3 Digital Neuron Model

The digital spiking neuron model used as the basis of this work is explained in detail in [14]. Here, we summarise its general design and advantages. To improve the performance of evolution, a developmental digital neuron model should be as flexible as possible, for any constraint may impair evolvability. Evolution must be able to modify everything from network topology and dendrite structures to learning rules, neuro-coding, membrane decay constants, and other cell parameters and processes.

This model is suitable for evolutionary development of heterogeneous spiking neural networks on FPGAs in many different ways. First, it uses a parametrically flexible and relatively bio-plausible soma model, which has the potential to be upgraded to more plausible models if hardware budget permits. Secondly, it provides the means for adding a local (thus parallel) learning process such as STDP in each synapse. Moreover, it is relatively fast and occupies acceptable

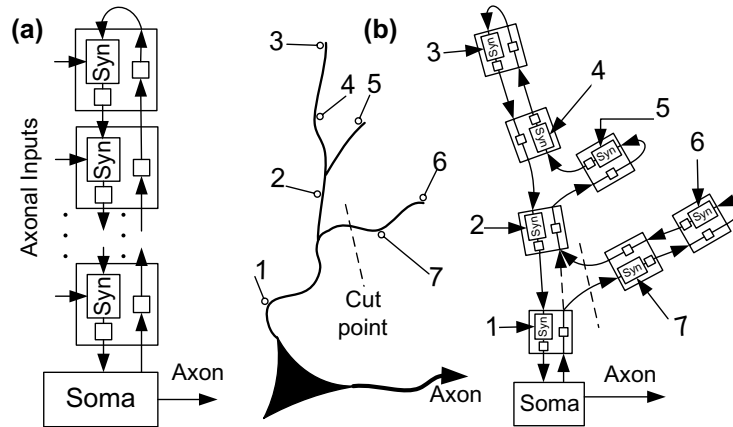


Fig. 1. (a) General Architecture of the digital neuron model. (b) Example of the dendrite structure and its adaptability (*Syn* and *small blocks* represent synapse units and pipeline D flip-flops respectively).

area on the FPGA. But more importantly, it allows us to develop adaptable dendrite and axon branches in a cellular structure as is proposed here.

3.1 General Architecture

In this model, each digital neuron consists of a set of synapse units and a soma unit connected in a daisy chain architecture shown in figure 1(a). The axonal input of each synapse is connected to the axon of the pre-synaptic neuron. This architecture creates a 2-way communication channel in the dendrite and allows the development of different dendrite structures as demonstrated in the example of figure 1(b). The dendritic lines and DFFs (D flip-flops) that connect the units form a loop (dendritic loop) that conveys data packets. The soma unit sends a packet containing the current membrane potential on its dendritic output. Synapse units process the packets. If a synapse unit receives a pre-synaptic action potential it adds (or subtracts) its synaptic weight to the first arriving packet. Therefore, the soma unit receives the sum of membrane potential and post-synaptic currents in its dendritic input. After processing this packet, the soma unit sends another packet with the updated membrane potential. Serial arithmetic is used in all the units to create pipelined parallel processing inside each neuron, meaning that neighbouring units process different bits of the same packet at the same time.

This architecture minimizes the number of local and global connections, which leads to a significant relaxation of constraints imposed upon the network architecture as limited routing resources is the major constraint in optimal utilization of FPGA functional resources. Each unit needs only a global clock signal to work. Other global signals can also be added for global supervised learning mechanisms. The user is free to trim (add) dendrite branchlets at any point

simply by cutting few connections and bypassing (inserting) the root unit of the branchlet as shown by the dashed lines in figure 1(b). This is done through a set of multiplexers as explained in detail in section 4. This flexibility is vital for a developmental model that needs on-line growth and modification.

3.2 The Synapse Unit

The synapse unit comprises a serial adder, a shift register containing the synaptic weight, and a control unit. It simply redirects its Dendritic Input (DI) to its Dendritic Output (DO) when no spike has arrived. When the control unit detects a spike on the aXonal Input (XI), it waits for the next packet to arrive. Then it enables the shift register and the adder until the whole packet is processed. The weight value loops back into the weight register. A learning block can be simply inserted into the feedback loop of the weight register in order to realize a local unsupervised learning mechanism like STDP. This learning block can access the post-synaptic potential and the pre-synaptic input.

3.3 The Soma Unit

Most of the hardware models are based on the Leaky Integrate and Fire (LIF) [21,22] or simplified LIF neuron models [7,13]. However, the neuron model used here is based on a Piecewise-Linear Approximation of Quadratic Integrate and Fire neuron model (PLAQIF) [14], which is biologically more plausible compared to the popular LIF model. This soma model has the parametric flexibility needed for evolving heterogeneous networks and the user can change its parameters by reconfiguring LUTs and shift registers on the fly. It occupies almost the same area as an equivalent LIF model. It generates bio-plausible action potentials in the dendrites and sends out logic '1' pulses (spikes) on its axonal output. The design and benefits of this soma model are discussed in [14] in more detail.

4 Cellular Structure

Biological brains are mainly composed of neurons and glial cells [23]. Glial cells provide support and nutrition for neurons and act as “glue” between them and recently were suspected to be involved in the synapse formation and axon and dendrite development [23,24]. Here, we use the word “glial cells” to mean non-neuron cells that provide the means for routing dendrites and axons, and formation of synapses at their intersections.

The proposed cellular structure consists of a grid of glial cells with neuron soma cells embedded in the middle of them. A hexagonal or even a multilayer 3D grid structure is also possible on new FPGAs. However, the limited resources of the FPGA logic blocks makes a simple 2D grid simpler and more feasible. To keep the regularity of the cellular structure, it is desirable that soma and glial cells be of the same size. Nevertheless, as functionality of soma cells requires more hardware resources than glial cells, they are two times larger than glial cells and

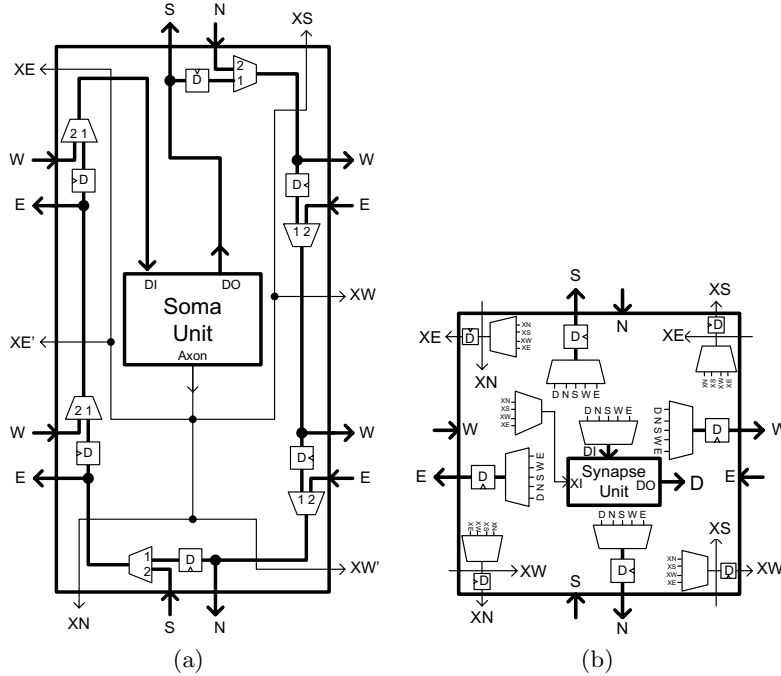


Fig. 2. (a) Internal Architecture of soma cell. (b) Internal architecture of glial cell.

fit into two vertically adjacent grid cells. The vertical option is preferred as it minimises the signal delay between neighbouring cells on the actual chip (see section 5). Each glial cell receives an axonal and a dendritic input signal from each side and has an axonal and a dendritic output on each side. Soma cells have six of those signals as they are in contact with six neighbouring glial cells.

4.1 Soma Cells

Each soma cell consists of a soma unit, six reconfigurable multiplexers and six pipeline D flip-flops (DFF). Reconfigurable multiplexers (from now on we refer to them as MUX) are basically FPGA LUTs (look-up tables) or multiplexers that are dynamically configured to work as many-to-one switch boxes. Using LUTs for this purpose makes it possible to use difference-based dynamic partial reconfiguration [25] for changing the routings [19]. The internal architecture of the soma cell is shown in figure 4. The axon output of the soma unit is connected to all the six axonal outputs of the soma cell (XN, XS, XW, XW', XE, XE'). This way axons can project out of the soma cell in any direction before branching into branchlets, increasing the flexibility of the system. When there is no dendrite growth, DFFs and MUXs form the dendritic loop right inside the soma cell by switching all MUXs to their first input. A soma cell can start growing a dendrite branch on any of its edges by switching the corresponding MUX to its second input. Therefore, a soma cell can project up to six dendrite branches directly from the cell body before any

division into dendritic branchlets. This adds to the flexibility of the routing while resembles to dendrite growth of the biological neurons.

4.2 Glial Cells

Figure 4 shows the internal architecture of the glial cells. Each glial cell consists of a synapse unit, ten MUXs, and eight DFFs for routing axons and dendrites. On each side of a glial cell, there is one axonal output coming from a pipeline DFF connected to a MUX. Each axonal MUX can switch to any of four axonal inputs on the edges of the glial cell (XN, XS, XW, XE). This way, it is possible to route up to four axons through a glial cell as explained later in the example of section 4.3.

A similar circuit is employed for the dendrite routing. However, each MUX in the dendritic circuit has a fifth input, which is connected to the dendritic output of the synapse unit (DO). The dendritic input of the synapse unit (DI) comes from another MUX that can switch to any of the dendritic inputs on the edges of the glial cell (N, S, W, E). Therefore, the synapse unit can be inserted into any of the dendritic loops routed through the glial cell. The axonal input of the synapse unit can also be connected to any of the four axonal inputs of the glial cell using a 4-to-1 MUX. Therefore, it is possible to form a synapse between any dendrite and axon routed through a glial cell in three simple steps: 1. Switch the axonal MUX of the synapse unit. 2. Copy the configuration of the corresponding dendritic MUX to the dendritic MUX of the synapse unit. 3. Switch the corresponding dendritic MUX to synapse dendritic output (DO). The pipeline DFFs in the routing circuit improve clock frequency and allow evolution to optimize dendritic and axonal delays by changing the length and path of each branch. Similarly, a reverse procedure can be use to eliminate a synapse.

The only limitation is that there is only one synapse unit available in each glial cell. The other option is to assign more hardware resources to glial cells and have two (or even more) synapse units in each glial cell. By increasing the number of synapse units, fan-in of the dendritic MUXs increases (to 6 inputs for 2 synapse units) and hardware resources to implement them grow exponentially. For efficient use of the hardware resources there should be an appropriate ratio of functional resources to routing resources in each cell. Although up to four different dendrites can project into a glial cell, the practical average number of the dendrites passing through a cell will be less than two in practice. Therefore, one synapse unit per glial cell seems reasonable.

4.3 Example

Figure 3(a) shows a symbolic view of the example network. It consists of three soma cells in a 6x4 grid of glial cells. Figure 4 shows the active circuit elements of the same network. The bottom soma in the E2 and F2 cells, projected three dendrites and one axon. On the bottom edge, there is no dendrite thus the bottom MUX is switched to input 1 to bypass the external circuit and use a pipeline DFF instead. On the bottom-left edge a very short dendrite is projected

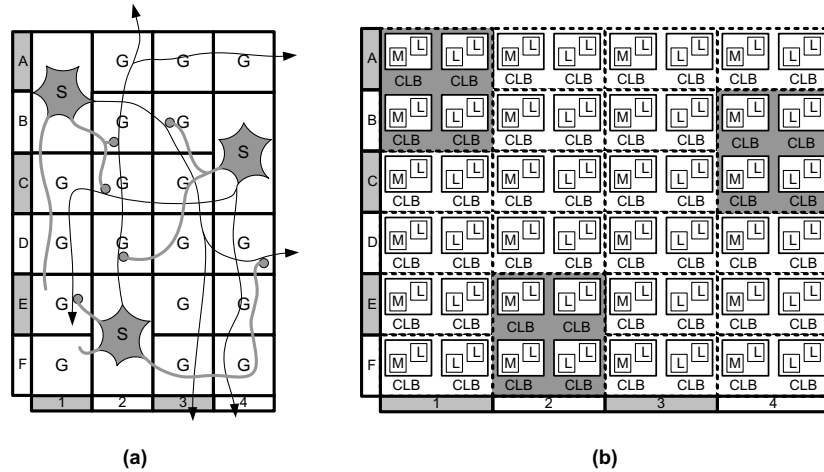


Fig. 3. (a) Symbolic view of the example network in a 4x6 grid. (b) Assignment of FPGA CLBs to glial and soma cells.

into the F1 cell. Therefore, the bottom-left MUX is switched to input 2. In the F1 cell the dendrite is looped back without forming any synapse by switching the corresponding MUX to input E. The dendritic loop is continued on the top-left edge of the soma cell with another short projection, this time forming a synapse with an axon coming from above. The other projection of the soma cell on its bottom-right edge has passed through a number of MUXs in different glial cells and formed a synapse with another axon in D4. The dendritic loop of this neuron contains 12 FFs, 17 MUXs and 2 synapse units. Its axon is gone through 3 MUXs and FFs upwards into A2 and then divided into two axons extending outwards. Routing of the projections from the other two neurons can be also tracked in a similar manner. In C3, for instance, a dendrite is divided into two branches. In B2, another dendrite formed a synapse as it extended into C2.

5 Virtex-5 Feasibility Study

Here we discuss the feasibility of implementing this cellular structure in Virtex-5 family of FPGAs. Two horizontally adjacent CLBs (Configurable Logic Blocks) are assigned to each grid cell. This is because synapse and soma unit designs make extensive use of Virtex-5 32-bit shift registers and only one out of four slices in two horizontally adjacent CLBs is a SLICEM capable of implementing shift registers [26]. As soma cells need more hardware resources they occupy a square block of four CLBs on the FPGA. This is because assigning 4 CLBs in a row to soma complicates the partial reconfiguration process (due to columnar nature of the FPGA fabric) and leads to employing long-range routing lines of the FPGA for inter-cellular connectivity. These lines are limited in number and

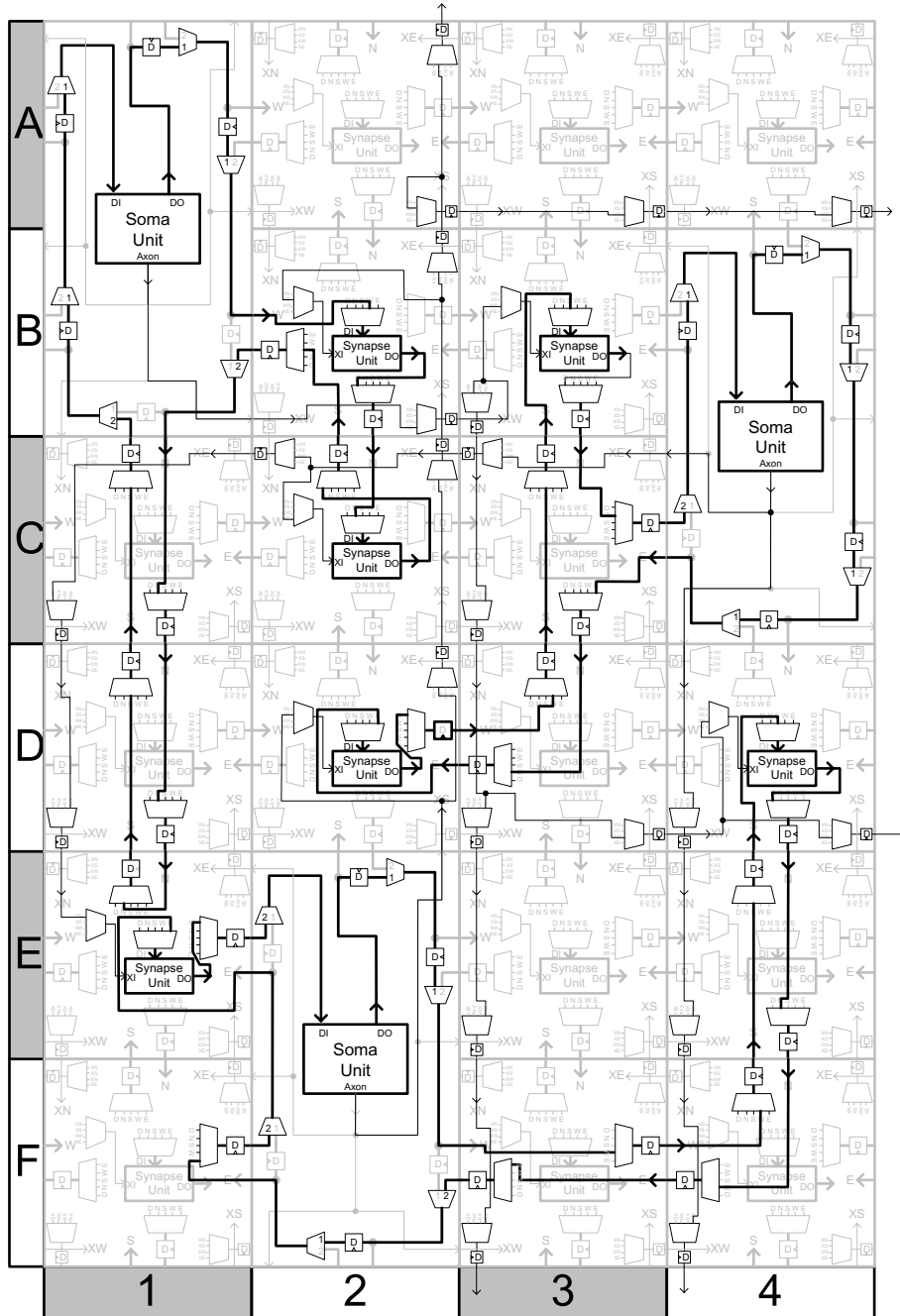


Fig. 4. Schematic diagram of the active circuits of the example network

have higher signal delays. Figure 3(b) shows how cellular structure of the above example can be implemented in the Virtex-5 CLBs.

VHDL and ISE 9.2i design tools were used for implementation of a sample cellular structure in a LX50T Virtex-5 FPGA. Implementing and floor-planning of the the soma and glial cells on the chip revealed that it is possible to pack the soma and glial cells in 2 and 4 CLBs respectively. Every two 5-to-1 MUXs with the same set of inputs are implemented in a 6-input LUT configured as two 5-input LUTs. This way, the whole routing circuit of a glial cell is implemented with six LUTs and eight DFFs, which is less than the available resources in the right CLB of the glial cell. The synapse unit takes almost all of the left CLB of the glial cell. The routing circuit of the soma cell is implemented using six DFFs and three LUTs, each configured as two 2-input LUTs. The rest of the hardware resources were more than enough for implementing the soma unit. Therefore, the extra hardware resources in each cell were reserved for future improvements (*e.g.* synaptic plasticity or upgrading to a more bio-plausible soma model).

It is possible to pack 1800 glial cells in an entry-level Virtex-5 FPGA (LX50T) and 12960 glial cells in the largest Virtex-5 chip (LX330T). With a 1/10 soma to glial cell ratio (each soma cell surrounded by a layer of glial cells), it is possible to implement networks with 150 and 1080 neurons with up to 1500 and 10800 synapses in LX50T and LX330T chips respectively. This might not be the optimum ratio but this is left to evolution to tune the ratio and placement of the cells in order to optimize the resources and performance.

5.1 Reconfiguration

The cellular structure is designed to exploit the dynamic partial reconfiguration feature of Virtex-5 FPGAs. Here, feasibility of reconfiguration methods for placement of the neurons at the first stage of the development and runtime modification to parameters and wiring of the neural network are studied. The reconfiguration can be carried out in three main steps:

At the first step, the whole area on the FPGA that is assigned to the neural network is configured as glial cells. This is simply done through the standard flow configuring the device with a bitstream generated from HDL. However, glial cells need to be defined as hard macros so that the exact locations of all MUXs (LUTs) and cell ports be fixed and known. In the second phase, soma cells are reconfigured instead of glial cells in the required places using merge dynamic reconfiguration technique [27]. Soma cell should be defined as a hard macro again with its ports carefully matched with the ports of the neighbouring glial cells. The merge reconfiguration technique [27] allows to vertically relocate a module with an arbitrary shape and size (2x2 CLBs in this case). Therefore, a relocatable soma bitstream should be created for each grid column (2 CLBs wide). In the final phase, soma and synapse parameters and axon and dendrite routings are modified by runtime difference-based dynamic partial reconfiguration of LUTs [25,28,19] provided that all the parameters and routings are based on LUT contents and the exact locations of all these LUTs on the FPGA are known (by using hard macros). Therefore, it will be possible to grow dendrites and axons and form/eliminate synapses on the fly.

6 Conclusions

A new cellular structure for routing of axons and dendrites on FPGAs based on the new digital spiking neuron model [14] was proposed and its feasibility on Virtex-5 FPGA was studied. The feasibility of online adaptation of routing and synapse formation/elimination through runtime partial reconfiguration of LUTs was shown. Using this design, it would be practically possible to develop neural networks as large as 150 and 1080 neurons and up to 1500 and 10800 synapses on entry-level and high-end Virtex-5 chips respectively. Despite the limited resources when implemented in current FPGAs, this architecture will still allow systems such as time series prediction, robot navigation, and gesture recognition to be implemented successfully.

The next step toward creating an evolutionary developmental neural network would be to design an online developmental process that configures the LUTs with dendrite growth and axon guidance mechanisms based on gene expression, protein diffusion and availability of routing resources at each site. It would be also very interesting to see if local information in each glial cell can be used to create activity dependant synapse formation and elimination.

References

1. Hausser, M., Mel, B.: Dendrites: bug or feature? *Current Opinion in Neurobiology* 13(3), 372–383 (2003)
2. Yao, X.: Evolving artificial neural networks. *Proceedings of the IEEE* 87(9), 1423–1447 (1999)
3. Maass, W.: On the Computational Power of Noisy Spiking Neurons. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.) *Advances in Neural Information Processing Systems*, vol. 8, pp. 211–217. The MIT Press, Cambridge (1996)
4. Maass, W.: Noisy Spiking Neurons with Temporal Coding have more Computational Power than Sigmoidal Neurons. In: Mozer, M.C., Jordan, M.I., Petsche, T. (eds.) *Advances in Neural Information Processing Systems*, vol. 9, p. 211. The MIT Press, Cambridge (1997)
5. Maass, W., Natschlagel, Markram: Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *NEURCOMP: Neural Computation* 14, 2531–2560 (2002)
6. Hampton, A.N., Adami, C.: Evolution of Robust Developmental Neural Networks. In: *Artificial Life, IX.* (ed.) *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems* (2004)
7. Roggen, D., Federici, D., Floreano, D.: Evolutionary morphogenesis for multicellular systems. *Genetic Programming and Evolvable Machines* 8(1), 61–96 (2007)
8. Federici, D.: A regenerating spiking neural network. *Neural Networks* 18(5-6), 746–754 (2005)
9. Gordon, T., Bentley, P.: Evolving hardware. In: *Handbook of Nature-Inspired And Innovative Computing*, pp. 387–432. Springer, Heidelberg (2006)
10. Chklovskii, D.B., Mel, B.W., Svoboda, K.: Cortical rewiring and information storage. In: *Nature*, October 2004, vol. 431(7010), pp. 782–788 (2004)
11. Chklovskii, D.B., Schikorski, T., Stevens, C.F.: Wiring Optimization in Cortical Circuits. *Neuron*. 34(3), 341–347 (2002)

12. Chklovskii, D.B.: Synaptic Connectivity and Neuronal Morphology: Two Sides of the Same Coin. *Neuron*. 43(5), 609–617 (2004)
13. Upegui, A., Pena-Reyes, C.A., Sanchez, E.: An FPGA platform for on-line topology exploration of spiking neural networks. *Microprocessors and Microsystems* 29(5), 211–223 (2005)
14. Shayani, H., Bentley, P.J., Tyrrell, A.M.: Hardware Implementation of a Bioplausible Neuron Model for Evolution and Growth of Spiking Neural Networks on FPGA. In: Second NASA/ESA Conference on Adaptive Hardware and Systems, 2008. AHS 2007 (2008); Third NASA/ESA Conference on submitted to Adaptive Hardware and Systems, 2008. AHS 2008
15. Chua, L., Yang, L.: Cellular neural networks: theory. *IEEE Transactions on Circuits and Systems* 35(10), 1257–1272 (1988)
16. Chua, L., Roska, T.: *Cellular Neural Networks and Visual Computing: Foundation and Applications*. Cambridge University Press, Cambridge (2002)
17. Rodriguez-Andina, J., Moure, M., Valdes, M.: Features, Design Tools, and Application Domains of FPGAs. *IEEE Transactions On Industrial Electronics* 54(4), 1810 (2007)
18. Thompson, A.: An Evolved Circuit, Intrinsic in Silicon, Entwined with Physics. In: Higuchi, T., Iwata, M., Weixin, L. (eds.) ICES 1996. LNCS, vol. 1259, pp. 390–405. Springer, Heidelberg (1997)
19. Upegui, A., Sanchez, E.: Evolving Hardware with Self-reconfigurable Connectivity in Xilinx FPGAs. In: First NASA/ESA Conference on Adaptive Hardware and Systems. AHS 2006, pp. 153–162 (2006)
20. Gers, F., de Garis, H., Korkin, M.: CoDi-1Bit: A simplified cellular automata based neuron model. *Artificial Evolution* 1363, 315–333 (1998)
21. Izhikevich, E.M.: *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting (Computational Neuroscience)*. MIT Press, Cambridge (2007)
22. Gerstner, W., Kistler, W.: *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, Cambridge (2002)
23. Laming, P., Syková, E.: *Glia! Cells: Their Role in Behaviour*. Cambridge University Press, Cambridge (1998)
24. Pfrieger, F.W.: Role of glia in synapse development. *Current Opinion in Neurobiology* 12(5), 486–490 (October 2002)
25. Xilinx: *Difference-Based Partial Reconfiguration* (February 2007)
26. Xilinx: *Virtex-5 User Guide* (February 2008)
27. Sedcole, P., Blodget, B., Becker, T., Anderson, J., Lysaght, P.: Modular dynamic reconfiguration in Virtex FPGAs. *IEE Proceedings of Computers and Digital Techniques* 153(3), 157–164 (2006)
28. Upegui, A., Sanchez, E.: Evolving Hardware by Dynamically Reconfiguring Xilinx FPGAs. *Evolvable Systems: From Biology to Hardware* 3637, 56–65 (2005)