NASA/ESA Conference on Adaptive Hardware and Systems

Dynamic Routing on the Ubichip: Toward Synaptogenetic Neural Networks

Andres Upegui, Yann Thoma, Andres Perez-Uribe, Eduardo Sanchez
REDS - HEIG-VD
Yverdon-les-Bains, Switzerland
Email: (andres.upegui yann.thoma andres.perez-uribe eduardo.sanchez)@heig-vd.ch

***Abstract*— The *ubichip* is a bio-inspired reconfigurable circuit developed in the framework of the european project Perplexus. The *ubichip* offers special reconfigurability capabilities, being the dynamic routing one of them. This paper describes how to exploit the dynamic routing capabilities of the *ubichip* in order to implement synaptogenetic neural networks. We present two techniques for dynamically generating the network topology, we describe their implementation in the *ubichip*, and we analyse the resulting topology. This work constitutes a first step toward neural circuits exhibiting more realistic neural plasticity features.**

I. Introduction

The Perplexus project [1] aims to develop a scalable hardware platform made of custom reconfigurable devices endowed with bio-inspired capabilities. This platform will enable the simulation of large-scale complex systems and the study of emergent complex behaviors in a virtually unbounded wireless network of computing modules. At the heart of these computing modules, we will use a *ubichip* [2], a custom reconfigurable electronic device capable of implementing bio-inspired mechanisms such as growth, learning, and evolution. These bio-inspired mechanisms will be possible thanks to reconfigurability mechanisms like dynamic routing, distributed self-reconfiguration, and a simplified connectivity. Such an infrastructure will provide several advantages compared to classical software simulations: speed-up, an inherent real-time interaction with the environment, self-organization capabilities, simulation in the presence of uncertainty, and distributed multi-scale simulations.

The *ubichip* offers thus an interesting set of reconfigurability mechanisms for supporting networks featuring different types of neural plasticity. Different approaches have been proposed for automatically generating neural networks' topologies. Evolutionary artificial networks [3], for instance, generate a network from a description contained in a genome. Each link of the network is somehow coded in the genome. Being an effective approach for computing purposes, this approach results unplausible in biological neural systems given that the human brain features around 10^{14} connections, far more than can be coded in the 3×10^{9} nucleotides contained in the human ADN. Another common approach for generating network's topologies is growing and pruning algorithms [4]. Such algorithms add or remove neurons to or from a network, according to its computing or generalization capabilities. This approach results also useful for computing purposes but it is not biologically plausible given the topological dependency on a specific task performance.

Ontogenetic (or developmental) neural models arise as another approach for building neural networks. From some initial construction rules and some initial conditions, the network is incrementally built under a constant interaction with the environment. In [5], for instance, Cangelosi uses a genotype for encoding the construction rules of a neural network. This approach results more biologically plausible given the undirect correlation between the individual's genotype and phenotype. An individuals phenotype is not directly derived from its genotype, but it is also influenced by the environment stimuli during the individual's life-time. This phenotypic development is also driven by physical constraints that allow to bound the brain's size, the number of dendrites per neuron, an axon's length, and the resulting neural structure. Physical constraints are very rarely taken into account by incremental network building models.

Unlike most of the artificial neural networks used for computing purposes, brain network's topology exhibits a quite intrincated, but still well organizing pattern. How such a network with billions of neurons interacting through electric spikes manage to exhibit higher level and quite complicated processes such as learning and consciousness remains a mystery for scientists. However, neuroscientists have found several clues about the underlying mechanisms that allows such higher level processes to arise. One of these clues suggests that network's topology plays a fundamental role in this process [6], making brain's topology and plasticity one of the key phenomena to drawn inspiration from in order to model and understand such type of systems.

This paper presents two approaches for exploiting the *ubichip*'s reconfigurable capabilities, more specifically the dynamic routing, in order to implement synaptogenetic artificial neural circuits. The synaptogenetic model presented in this paper allows a network to be developed in function of the interaction with the environment, more precisely, in function of the input's stimuli. We drawn inspiration from some biological processes in order to propose our configurable circuit architecture. It must be noted that this paper focuses on the network topology generation and not on its ability to solve a task. To the best of our knowledge, this is the first reported neural circuit featuring synaptic plasticity.

This paper is structured as follows: In sections II and III

978-0-7695-3166-3/08 $25.00 © 2008 IEEE
DOI 10.1109/AHS.2008.21

we introduce the *ubichip*'s architecture, and we describe the dynamic routing mechanisms offered by it. Section IV gives a short introduction to complex systems' topologies and neural circuits. Then, section V describes our two models for generating neural topologies: a random and an activity-driven approach. Section VI describes the implemented networks and the results obtained from them. Finally, section VII concludes.

II. UBICHIP

The *ubichip* is a custom reconfigurable electronic device capable of implementing bio-inspired mechanisms such as growth, learning, and evolution. These bio-inspired mechanisms are possible thanks to reconfigurability mechanisms like dynamic routing, distributed self-reconfiguration, and a simplified connectivity.

Recent work in this field is the POEtic tissue [7], a reconfigurable hardware platform for rapidly prototyping bio-inspired systems that employs POE principles [8], which has been developed in the framework of the European project POEtic. The POEtic chip has been specifically designed to ease the development of bio-inspired applications. The limitations exhibited by the POEtic tissue suggest several architectural and configurability features to be improved. These improvements may lead us to a reconfigurable platform better suited for supporting the bio-inspired principles that we want our devices to mimic.

The *ubichip* is mainly composed of three reconfigurable layers. The first one is an array of *ubicells*, the reconfigurable logic elements used for computation purposes. A *ubicell* is composed of four 4-input look-up tables (LUT) and four flip-flops (DFFs). These *ubicells* can be configured in different modes like counter, FSM, shift-register, 64-bit LFSR, adder, subtractor, etc. An *ubicell* can also implement a simple 4-bit processing element being part of a SIMD multiprocessing platform, and n *ubicells* can be merged to create a $4n$-bit processor. In this last mode, an on-chip centralized sequencer is responsible of decoding the instructions for the multi-processor management.

The second layer contains dynamic routing units that permit the *ubicells* to dynamically connect to any part of the circuit. Based on identifiers and a concept of sources and targets trying to reach a correspondent with the same ID, it looks quite similar to the system described in [9], while having enhancements on different aspects. It will be described in more detail in section III.

Finally, the third layer is made of self-replicating units that allow part of the circuit to self-replicate somewhere else on the chip, without any external intervention. This truly new feature can be very useful for cellular systems such as neural networks with changing topologies. A neuron could for instance decide to self-replicate if it has a high level of activation, or die by self-destruction if it is unused, in order to leave resources for another neuron. More details about this mechanism can be found in [10].

These three layers are interconnected for allowing the *ubicell* layer to control the two top layers. The *ubicell* layer can thus implement a circuit able to control the dynamic routing and the self-replication layers. For this, we grouped units from the three different layers for forming what we call a macrocell. A macrocell contains thus four *ubicells* connected to a routing unit and a self-reconfiguration unit as described in the schema of figure 1.

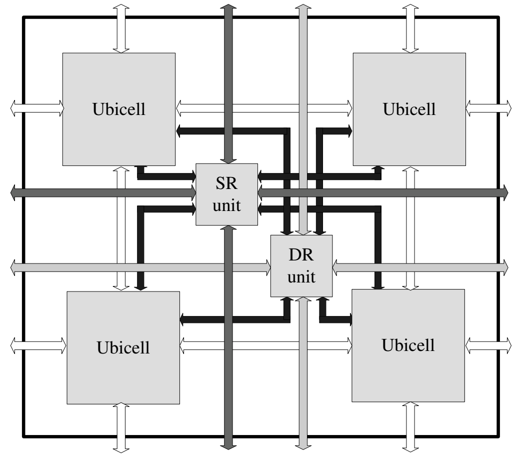

Fig. 1. Macrocell architecture

Because of the scope of this paper, we will focus on the dynamic routing layer description, we will describe its implementation, its features, and its utilisation.

III. DYNAMIC ROUTING

Growing and developing cellular systems require the ability of creating and destroying paths at runtime, in order to connect newly created cells. Epigenetic systems such as growing neural networks would also need to build connectivity during the lifetime of the network. For this purpose, the *ubichip* provides hardware mechanisms to handle such dynamic topologies.

Considering the typical high silicon overhead due to routing matrices, specially high for dynamic routing, we chose a solution requiring the less amount of logic as possible, while being flexible enough to deal with the changing topology of the network. One of the simplest physical realization consists in the wormhole routing concept [11]. However, the hardware overhead of this kind of algorithms is not suitable for the granularity of the reconfigurable array. Therefore, we will implement a dynamic routing algorithm, by improving the routing implemented in the POEtic chip [12]. The risk of congestion will be reduced by means of four features:

1) The new algorithm will better exploit the existing paths by reusing the created paths instead of fully routing each signal independently from the source to the target.
2) An 8-neighborhood will allow a dramatic reduction of congestion risk compared to the amount of logic required.
3) Underlying *ubicells* can trigger a path destruction, in order to remove unused connections.

4) *Ubicells* can also modify the address of routing units, in order to reconnect to other units different from the one initially specified.

And finally, while in POEtic the circuit execution was frozen during a routing process, in the *ubichip* the creation of a new path will let the system run without interruption.

The basic idea of the routing algorithm is to construct paths between sources and targets by dynamically configuring multiplexers, and by letting the data follow the same path for each pair of source and target. A phase of path creation executes a breadth-first search distributed algorithm, looking for the shortest path. Sources and targets can decide to connect to their corresponding unit at any time by launching a routing process. Unlike the POEtic chip, where the system structure was predefined and the dynamic aspect was only present in the network creation, the *ubichip* allows to destroy paths and reconnect to different cells, making dynamic the structure definition itself.

Special routing units are implemented in hardware. They are composed of the multiplexers needed for data routing, the corresponding registers required to store the multiplexers configuration, and a finite state machine to handle the routing processes. The routing units are connected to the logic units in order to allow the underlying implemented system (neurons for instance) to manage the creation of new connections.

A routing process, dealing with the construction of a new path, is decomposed in 5 main phases:

1) When a source or a target initiates a connection, it activates a global signal. If different logic units start such a process at the same time, priority is given to the most bottom-left unit.
2) After a master is identified, it sends serially its ID.
3) At the end of the address broadcasting, all sources and targets have compared the data with their own address and know if they are involved in the current process.
4) A breadth-first search algorithm then searches for the shortest path.
5) When the target is found, a backward signal allows for the configuration of the multiplexers present on the path, and the process ends up with a new path.

These dynamic routing mechanisms, along with the computational capabilities offered by the *ubicells*, will allow us to tackle the modeling of neural circuits exhibiting intricate and dynamic topologies.

IV. Complex Systems and Neural Circuits

Mammals' brain is a complex system composed of millions of neurons interconnected by an intrincated network. The topology of such network and the developing mechanisms that allows to form it, remain a challenging study field for neuroscientists. Several studies have attempted to find characteristic patterns in such connectivity in order to model these neural circuits. These studies have found that neural interconnectivity is neither completely regular nor fully random, but it exhibits an intricate organization [13].

Complex systems are systems composed of a set of parts that exhibit behavioral properties which are not obvious to predict from the behavioral description of the forming parts [14]. Complex systems are characterized by emerging behaviors, non-obvious and rich interactions among parts, a hierarchical multi-scale nature, and in most cases the system itself evolves over time. The increasing interest in understanding and modeling such systems is because of their abundance on real life. Some examples of complex systems are multi-cellular organisms, energy distribution infrastructures, social networks, economics, climate, and of course, the brain.

A very important aspect on these systems is their topology: the way in which the parts or nodes are interconnected. The simplest topology used for modeling complex systems is a regular array, where each node is connected to neighbor nodes forming a n-dimensional regular grid. Cellular automata are a good example for these type of systems: an automaton's next state is determined by its own and its surrounding neighbors' states.

Another common modeling approach is random network structures. Nodes are randomly linked, independently of their positions or their previous connections. Some examples of such models are random boolean networks and echo state networks, which are well known for exhibiting very interesting dynamical properties. For a long time, they have been used as models for several real complex systems. The degree distribution (the distribution of the number of connections per node) of random networks exhibit a Poisson distribution.

Somewhere between these regular and random networks, we find the so-called small world networks [14]. A small world topology is characterized by connections with neighbor nodes mixed with a some degree of randomness, exhibiting a high degree of local clusterization. Small world networks are mainly characterized by the short path that connects every two nodes, and have been shown to be very robust when erasing nodes. The best known example of a small world network is the Milgram's experiment [15], which concluded that every two persons in the world are connected to each other by a maximum of six social links.

Another type of network is the scale-free network. These networks posses the particular characteristic of exhibiting a power law degree distribution. This distribution implies the existence of hubs in the network. These hubs are highly connected nodes that always allow to facilitate short paths among nodes. These networks remain as robust as the standard small world network when the removed node is not a hub. However, when the removed node is a hub the network connectivity gets very affected. A good example of this type of network are the flight connections: the general impact on flight connections for closing the airport of Paris (a hub) is not comparable to that of Geneva (a node).

Recent studies have concluded that neural circuits exhibit small world connectionism [13]. In part, this explains the brain robustness in presence of neural death. Additionally, the brain structure exhibits a constant plasticity, that implies that brain's topology is not static, but dynamic. Neurons and synapses are constantly being created and destroyed at time scales of minutes or hours. During early mamalian development this plasticity is very high, for achieving a certain stability in the adult stages. However, neural and synaptic birth and death remain active during the whole individual's life-time.

V. Synaptogenetic Hardware Model

The *ubichip* arises thus as a promising hardware substrate for implementing changing-topology electronic networks, more precisely in our case, synaptogenetic artificial neural networks. Our model considers, in a first stage, the initial existence of a set of unconnected 2-input neurons, where dendrites (inputs) and axons (outputs) are connected to dynamic routing units which are previously configured to act as targets and sources respectively. The connectivity pattern will be further generated during the neural network life-time.

Since in this work we focus on network's topology generation, and not in the ability of such network for solving a task, we use a very simplified neuron model whose implementation on the *ubichip* requires only four macrocells. The neuron model is defined by a stochastic activation function $y = f(i_1 + i_2)$ where, i_1 and i_2 are the two inputs of the neuron, and $f(x)$ is an activation function that if $x = 2$, $f(x) = 1$; if $x = 0$, $f(x) = 0$; and if $x = 1$, $f(x) = 1$ with a probability of 50% and $f(x) = 0$ otherwise. In other words, if both inputs are active the neuron fires, if no input is active the neuron does not fire, and if a single input is active the neuron has a probability of 50% of firing. This stochastic function is easily implemented in the *ubichip* thanks to the LFSR configuration mode of the *ubicell*. Figure 2 shows a screenshot of the dynamic routing layer in the UbiManager design tool [16]. In this screenshot we can identify four routing units: the two top units are configured as targets and act as the neuron's dendrites, the bottom left unit is configured as source and acts as the neuron's axon, and the bottom right unit is not used.

An important aspect to consider in the network generation model, is the eventual impossibility of creating a connection between two existing nodes because of routing congestion.

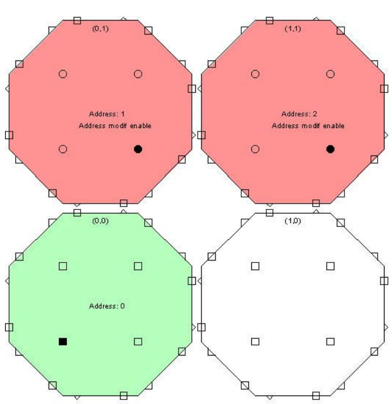

Fig. 2. Inputs and output of a neuron connected to dynamic routing units

More than a limitation of the implemented model, it constitutes an interesting feature. Physical routing constraints are present in both: biological and our artificial network, and this similarity will certainly make, in both cases, more likely to create connections between close neurons than between remote neurons, generating specific clustering patterns. However, the size of the models presented in this paper are not still large enough for exhibiting such clustering.

In this paper we present two approaches for generating the network's topology. The first is a random process where each dendrite attempts with a given probability to connect to a random axon, and the second one is an activity-driven approach where the probabilities of an axon to get connected are function of the neuron's activity.

A. Random Synaptogenesis

As described in section III a connection between a source and a target is created when both have the same address, and one of them triggers a process of dynamic routing. The order of creating a connection comes from the *ubicell* layer, which is the layer that contains the computation logic. The *ubicell* layer can also modify the address assigned to a routing unit.

For implementing the random synaptogenesis network a set of unconnected neurons is initially configured in the *ubichip*. Each neuron has a unique axon address and dendrites' address can be randomly modified at each clock cycle with a certain probability. Each dendrite has also a certain probability to attempt a connection (create a synapse) with the axon corresponding to its current address. If the synapse is successfully created, it remains and the dendrite does not attempt to connect any more. If it is not successful it continues retrying new connections until a synapse is created.

There are several reasons for a synapse creation to not being successful. The first is the case where another dendrite is trying to connect; the creation of a path requires several clock cycles and only one can be done at a time. The second reason is the non-existence of an axon with such address; addresses are coded in 8 bits and if the number of neurons is less than 2^8, one can attempt connections to non-existing axons. The third reason is routing congestion; at the end of the synaptogenetic process it can happen that the created network does not allow the connection between two existing nodes because of the unavailability of routing paths.

B. Activity-Driven Synaptogenesis

The principle behind the activity-driven synaptogenetic approach is the fact that more active neurons are more likely to get connected that less active neurons. This principle is inspired in a phenomenon called ocular dominance plasticity [17]. This phenomenon has been observed in an experiment called monocular deprivation, where a single eye of a kitten is occluded during a critical period of early life. It has been observed that synaptic connections are mainly connected to the non-occluded eye, and the occluded eye loses ends up weakly connected to cortical cells. It has been also observed that when uncovering the occluded eye the cortical circuit is reconnected in order to compensate the dominance of the non-occluded eye. Of course, the recovery is not complete and it is possible only if the uncovering is done before a certain age.

Our implementation aims to inspire from this principle for creating the synaptogenetic network. The network has two inputs which may be stimulated with different firing rates. The most excited neurons will have more chances to get connected than the less excited neurons. On the same way, the neurons that get connected to the input neurons will be activated by them and will increase also their possibility of getting connected. In this paper we present a first stage of the complete implementation, where we excite both inputs at different firing rates and we analyse the network topology generated from it.

The implementation on the *ubichip* uses the same neuron model described before, but with a different synaptogenetic process. A set of unconnected neurons is initially configured in the *ubichip*. Each dendrite has a unique address. Axons' address can be randomly modified, with a certain probability, at each firing of its respective neuron. The more a neuron fires, the higher will be the probability of modifying its address. And the more an axon changes its address, more probabilities it will have to get connected to different dendrites. As in the random network, dendrites are constantly attempting to get connected to a constant address, and if the synapse is successfully created, the connection remains and the dendrite does not attempt connections any more.

VI. Experimental Setup and Results

We implemented both types of synaptogenesis on the *ubichip* architecture using the UbiManager design tool [16]. In both cases we implemented an array of 62 neurons (8×8 leaving some place for generating the input stimuli) using an *ubichip* description of 32×32 *ubicells*, the equivalent of 16×16 macrocells.

A. Random synaptogenesis

In the case of the random synaptogenesis, each dendrite at every clock cycle has a probability of 6.25% ($1/16$) of attempting a connection and a probability of 37.5% ($6/16$) of attempting to modify its current address. These parameters where arbitrarily chosen.

After a certain number of clock cycles, a mean of 6940 with a standard deviation of 458, the whole network is created, exhibiting the degree distribution illustrated in figure 3. This figure corresponds to the average distribution on 10 runs. In the degree distribution, each column shows the number of neurons featuring a number of *x* connections. One can see, from the figure, that every neuron has at least 2 connections since each neuron has 2 input dendrites that are all connected at the end, and an axons connections can range from 0 to 8 connections. One can also conclude from the figure that the degree distribution roughly approaches to a Poisson distribution, which is the characteristic degree distribution of random networks [14].

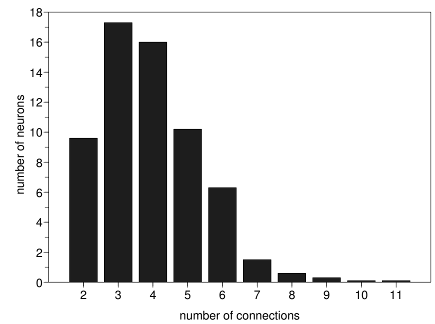

Fig. 3. Degree distribution of the random synaptogenesis

Figure 4 shows an example of a resulting random network viewed in the ubimanager. For the sake of clarity, the network in the figure is smaller than the network reported in this paper. It has 16 neurons (arranged 4×4), without specific input neurons.

In figure 5, we can also observe an example of one of the generated network topologies. The observed network structure correspond to a standard random network.

B. Activity-driven synaptogenesis

In the activity-driven synaptogenesis, each dendrite at each clock cycle has also a probability of 6.25% of attempting a connection, and each axon has a probability of attempting to modify its address of 50% at each axon firing. Two neurons are used as input neurons, they receive an external stimuli (generated by a counter and some extra circuitry) that mimics the activity of external sensors (or eyes in the case of a kitten). The more excited input neuron receives a stimulus every 16 clock cycles through one dendrite and the second dendrite receives no stimulus. The less excited input neuron receives a

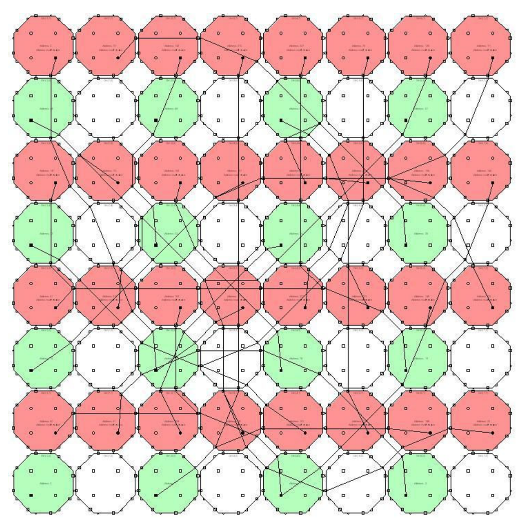

Fig. 4. Example of a randomly generated network 4×4 on the *ubichip*

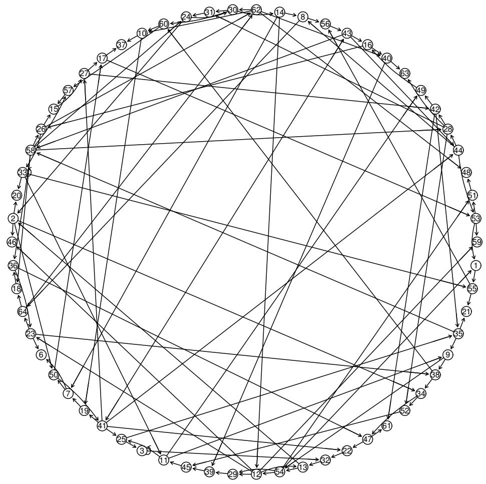

Fig. 5. Random synaptogenesis topology

stimulus every 256 clock cycles through one dendrite and the second dendrite receives no stimulus.

After a number of clock cycles the complete network is created, the mean number of clock cycles being 29482 with a standard deviation of 3415. An example of the obtained network is depicted in figure 6. The input neurons of the network are in the bottom center of the figure.

The degree distribution of the created network is depicted in figure 7. This figure corresponds to the average distribution of 14 runs. This degree distribution suggests at first view a Power law, which is the characteristic distribution of scale-free networks [14]. One can intuitively explain the structure of this network as some few hubs being highly connected, which should be the more excited input neuron and the first neurons getting connected to it, and most of the neurons weakly connected. However, we stongly belive that this Power law distribution is achieved because of the reduced number of links per node, and that increasing the connectivity of the network we will end up with a Poisson degree distribution. This hipotesis is very important because real neural topologies exhibit this type of degree distribution, and it is supported by the work presented in [18], where they have shown that the random adding of links to a scale-free network results in a Poisson degree distribution. Proving this hipotesis is the main raison for pursuing our research toward more highly connected networks.

The activity-driven synaptogenesis requires much more clock cycles than the random synaptogenesis approach. This difference is because the random approach allows any axon to get connected at any time since the beginning, and in the activity-driven approach, only stimulated neurons can have a valid axon address, so at the beginning only the two input neurons can get connected.

In figure 8, we can observe an example of one of the network topology generated by the activity-driven synaptogenesis. In this network, we can observe some highly connected nodes or hubs. For instance, the node 6 in the figure is the more excited of the input neurons. It has 11 outputs and the two inputs (not shown in the figure) come from external inputs. We can also identify the node 16, which was initially stimulated by the node 6, acting as a node with 11 connections. On the other hand, it can be identified many nodes (more than in the random network) featuring a low connectivity with only 2 or 3 connections.

Concerning the two different stimulation rates, we can also identify in the figure 8 the differences between the networks created from each input. The network inputs are the nodes 6 and 3, being respectively the highly and the lowly stimulated inputs. Figure 9 shows the network created from the node 6, the highly stimulated node. The figure only shows the nodes

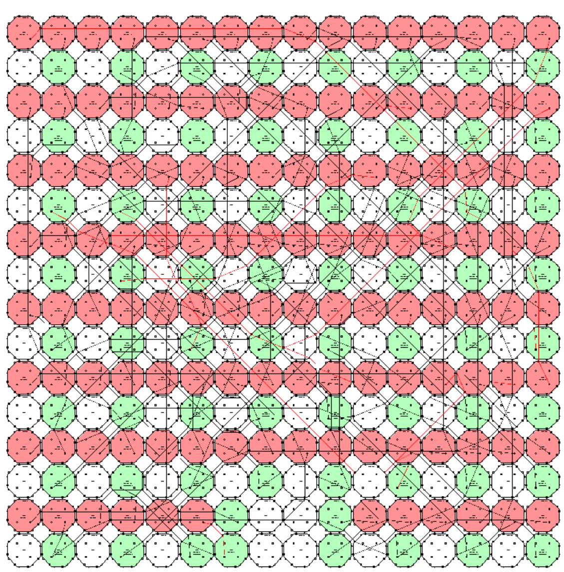

Fig. 6. Activity-driven network on the *ubichip*

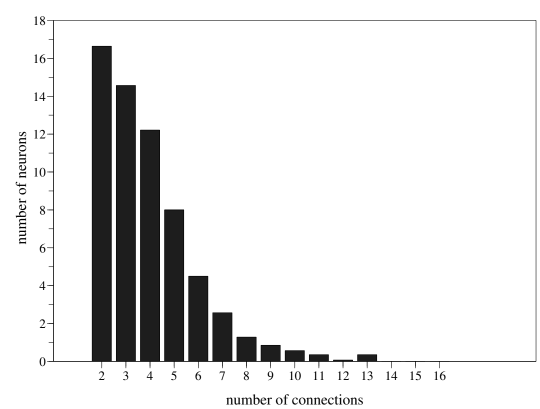

Fig. 7. Degree distribution of the activity-driven synaptogenesis

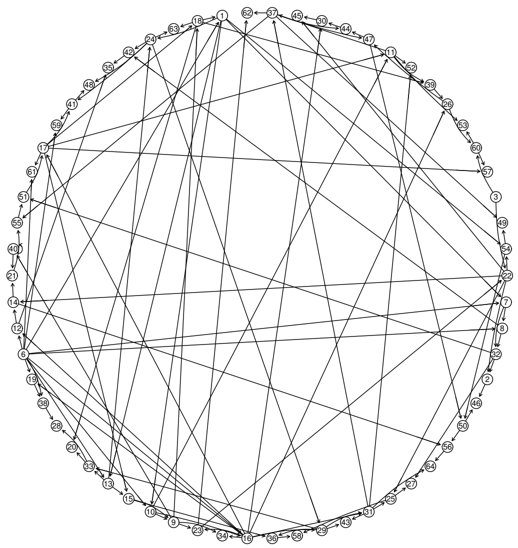

Fig. 8. Activity-driven synaptogenesis topology

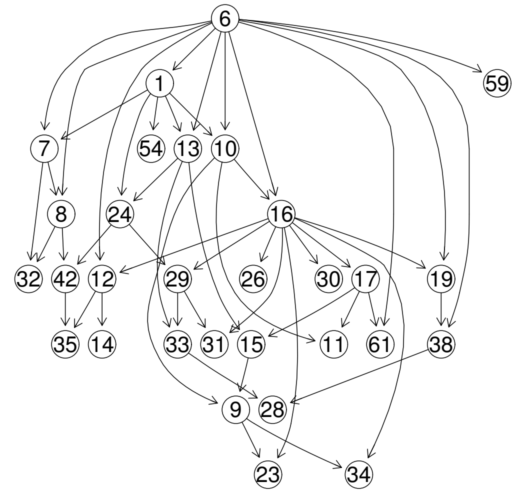

Fig. 9. Network created from the more stimulated input

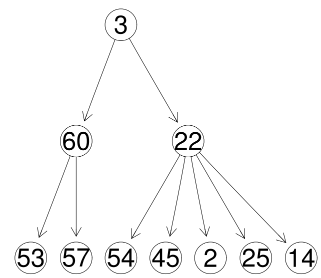

Fig. 10. Network created from the less stimulated input

that can be attained by a maximum of two links from the input neuron. This input can attain 29 neurons through a maximum of 2-edges paths; this corresponds to almost the half of the nodes of the full network. Figure 10 shows the network created from the node 3, the poorly stimulated node. In the same two steps it can only attain 9 nodes, and only 2 in a singly step. This second input is less connected to the network and, as in the experiment with the kitten described in subsection V-B, the network connectivity is more developed toward the more stimulated input (the non-occluded eye) than toward the less stimulated one (the occluded eye).

VII. Conclusions and Future Work

This paper presented two models, with their respective implementation, for implementing synaptogenetic neural networks by exploiting the dynamic routing capabilities of the *ubichip*. The networks obtained by the activity-driven model, exhibits complex topologies featuring some similarities with real neural topologies, i.e. being neither fully random nor completely regular. This structure is the result not only of the activity-driven plasticity, but of the physical constraints imposed by the substrate (*ubichip*), in a similar way to brain-like structure development in organisms. The possibility of mixing some level of randomness with some level of organization should drive future efforts in this research line.

The dynamic routing mechanisms offered by the *ubichip* represent an important implementation tool for designing neural circuits exhibiting some level of synaptic plasticity. Future work will exploit the capability of pruning existing synapses, which will increase the richness of the phenomenon being modeled. This will also permit to go further on the ocular occlusion experiment, where a recovery of the occluded eye has been observed when uncovering it.

Future work will also exploit other reconfigurable capabilities of the *ubichip* like self-replication, which will allow to implement neurogenetic circuits. In this case an initial neuron with self-replicating capabilities will be able to fill the available *ubicells* on the circuit with a number of neurons. These neurons will further need to create connections between them, allowing to think about more complete models of neurogenetic and synaptogenetic neural circuits.

Acknowledgment

This project is funded by the Future and Emerging Technologies programme IST-STREP of the European Community, under grant IST-034632 (PERPLEXUS). The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication.

References

[1] E. Sanchez, A. Perez-Uribe, A. Upegui, Y. Thoma, J. Moreno, A. Villa, H. Volken, A. Napieralski, G. Sassatelli, and E. Lavarec, "PERPLEXUS: Pervasive computing framework for modeling complex virtually-unbounded systems," in *AHS 2007 - Proceedings of the 2nd NASA/ESA Conference on Adaptive Hardware and Systems*, T. Arslan et al, Ed. Los Alamitos, CA, USA: IEEE Computer Society, aug 2007, pp. 600–605.

[2] A. Upegui, Y. Thoma, E. Sanchez, A. Perez-Uribe, J. Moreno, and J. Madrenas, "The Perplexus bio-inspired reconfigurable circuit," in *Proceedings of the 2nd NASA/ESA Conference on Adaptive Hardware and Systems*, T. Arslan et al, Ed. Los Alamitos, CA, USA: IEEE Computer Society, aug 2007, pp. 600–605.
[3] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.
[4] R. Reed, "Pruning algorithms - a survey," *IEEE Transactions on Neural Networks*, vol. 4, no. 5, pp. 740–747, 1993.
[5] A. Cangelosi, S. Nolfi, and D. Parisi, "Cell division and migration in a 'genotype' for neural networks," *Computation in Neural Systems*, no. 5, pp. 497–515, 1994.
[6] S. Grossberg, "The link between brain learning, attention, and consciousness," *Consciousness and Cognition*, no. 8, pp. 1–44, 1999.
[7] Y. Thoma, G. Tempesti, E. Sanchez, and J. M. Moreno Arostegui, "POEtic: an electronic tissue for bio-inspired cellular applications," *Biosystems*, vol. 76, no. 1-3, pp. 191–200, 2004.
[8] M. Sipper, E. Sanchez, D. Mange, M. Tomassini, A. Perez-Uribe, and A. Stauffer, "A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 83–97, 1997.
[9] Y. Thoma and E. Sanchez, "An adaptive FPGA and its distributed routing," in *Proc. ReCoSoc '05 Reconfigurable Communication-centric SoC*, Montpellier - France, Jun. 2005, pp. 43–51.
[10] Y. Thoma, A. Upegui, A. Perez-Uribe, and E. Sanchez, "Self-replication mechanism by means of self-reconfiguration," in *20th International Conference on Architecture of Computing Systems 2007 (ARCS '07), Workshop proceedings*, M. Platzner, K.-E. Grosspietsch, C. Hochberger, and A. Koch, Eds. VDE Verlag, mar 2007, pp. 105–112.
[11] L. M. Ni and P. K. McKinley, "A survey of wormhole routing techniques in direct networks," *Computer*, vol. 26, no. 2, pp. 62–76, 1993.
[12] Y. Thoma, E. Sanchez, J. M. Moreno Arostegui, and G. Tempesti, "A dynamic routing algorithm for a bio-inspired reconfigurable circuit," in *Field-Programmable Logic and Applications, LNCS*, vol. 2778. Springer-Verlag, 2003, pp. 681–690.
[13] O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag, "Organization, development and function of complex brain networks." *Trends Cogn Sci*, vol. 8, no. 9, pp. 418–425, September 2004.
[14] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 4–5, pp. 175–308, 2006.
[15] S. Milgram, "The small world problem," *Psychology Today*, vol. 1, pp. 61–67, May 1967.
[16] Y. Thoma and A. Upegui, "Ubimanager: a software tool for managing ubichips," in *AHS 2008 - Proceedings of the 3rd NASA/ESA Conference on Adaptive Hardware and Systems*. Los Alamitos, CA, USA: IEEE Computer Society, 2008. In this volume.
[17] D. H. Hubel and T. N. Wiesel, "The period of susceptibility to the physiological effects of unilateral eye closure in kittens," *J Physiol*, vol. 206, no. 2, pp. 419–436, 1970.
[18] J. Scholz, M. Dejori, M. Stetter, and M. Greiner, "Noisy scale-free networks," *Physica A Statistical Mechanics and its Applications*, vol. 350, pp. 622–642, May 2005.