Si elegans: FPGA Hardware Emulation of *C. elegans* Nematode Nervous System

Pedro Machado[*], John Wade[*], and T.M. McGinnity[*, †]

[*]Intelligent Systems Research Centre (ISRC), University of Ulster, Magee Campus, Derry, Northern Ireland
e-mail: p.machado@ulster.ac.uk, jj.wade@ulster.ac.uk, tm.mcginnity@ulster.ac.uk

†. School of Science and Technology, Nottingham Trent University, Nottingham, United Kingdom
e-mail: martin.mcginnity@ntu.ac.uk

***Abstract* — For many decades neuroscience researchers have been interested in harnessing the computational power of the mammalian nervous system. However, the vast complexity of such a nervous system has made it very difficult to fully understand basic functions such as movement, touch and learning. More recently the nervous system of the *C. elegans* nematode has been widely studied and there now exists a vast wealth of biological knowledge about its nervous structure, function and connectivity. The *Si elegans* project aims to develop a Hardware Neural Network (HNN) to accurately replicate the *C. elegans* nervous system behavior to enable neuroscientists to better understand these basic functions. Replication of the *C. elegans* biological system requires powerful computing technologies, based on parallel processing, for real-time computation. The *Si elegans* project will use FPGAs due to their advanced programmable features that allow reconfigurability, high performance parallel processing and relatively low price per programmable logic element. Furthermore, the project will deliver an open-access framework that will be available via a Web Portal to neuroscientists, biologists, clinicians and engineers. In this paper an overview of the complete hardware system required to fully realize *Si elegans* is presented along with an early small scale implementation of the hardware system.**

Keywords: *Si elegans, Field Programmable Gate Array (FPGA), Hardware Neural Network (HNN), Spiking Neural Networks (SNN), C. elegans.*

I. INTRODUCTION

Caenorhabditis elegans (*C. elegans*) is one of the simplest and well characterized biological neural systems (BNS). It includes 959 cells, 135 muscle cells (95 body wall muscles, 30 innervated and 10 non-innervated internal muscles), 302 neurons and approximately 8000 synaptic connections. Moreover, this data and the entire connectome, a map of neural interconnectivity, is fully detailed and publicly available through the Worm Atlas [1]-[4].

Even though the *C. elegans* neural system has a very small number of neurons and connections (when compared with the human nervous system) it exhibits intelligent behaviours such as touch sensitivity, reproduction, defecation, thermotaxis, chemosensory, and 3 types of locomotion [5]. Furthermore, the nematode performs associative/non-associative learning that persist over several hours [6].

The simplicity of the *C. elegans* nervous system has gained much interest from the neuroscience community in recent years with a view to gaining greater insights into basic neural functions. However, current von-Neumann computer architectures are still unable to faithfully and quickly emulate this relatively simple nervous system. Such detailed and accurate simulation is only possible with a rich, powerful and biological real time computing environment composed of a high performance computational framework. To this end, the *Si elegans* project will deliver a parallel computing framework, consisting of Field Programmable Gate Arrays (FPGAs) that accurately simulates the complexity of the *C. elegans*' behaviour. While focused on *C. elegans*, the environment developed will be extensible to other neuromorphic studies. This framework will be made available to the worldwide scientific community through an open-access web portal enabling scientists to develop, implement and analyse realistic simulations.

FPGA devices were selected due to their advanced programmable features that allow reconfigurability, high performance parallel processing and relatively low price per programmable logic element [7].

In Section II a brief review is presented which details previous implementations of Neural Networks (NNs) using FPGAs. This is followed by an overview of the complete *Si elegans* hardware framework in Section III. The *Si elegans* project is currently at an early stage and in Section IV a small scale prototype of the *Si elegans* project is described and some preliminary results are presented in Section V. Finally, Section VI draws conclusion to the paper and describes future work.

II. FPGA NEURAL NETWORK BACKGROUND

Biological neural systems have been widely studied, by researchers and scientists, over the last number of decades. Neuroscience's interest in BNS and human brain functions was complemented by the development of artificial neural networks (ANN) that are characterized by parallelism, modularity and dynamic adaptation. These 3 characteristics

978-1-4799-5937-2/14/$31.00 ©2014 IEEE

make FPGAs the ideal devices to be used for ANN implementation. [8]. However, most existing ANN applications in commercial use are normally developed as software [9] reducing the emulation accuracy and increasing the simulation time.

In software, ANNs are generally modelled as a simple network of neuron like elements which emulate the phenomenological behaviour of Neural Networks (NNs). This helps reduce the simulation time required but reduces the biophysical realism of the system. Hardware implementations of ANN (HNN) take advantage of the truly parallel and distributed processing capabilities of a biological nervous system and therefore address this trade-off between realism and simulation time. Over the last 2 decades FPGAs have been used in the following HNN applications [9]:

- Pattern recognition;
- Image segmentation;
- Intelligent video analytics;
- Autonomous robotics;
- Sensorless control.

NN topology in ANNs is normally specified in terms of the Activation Function (AF), learning algorithm, number and type of inputs/outputs, number of neurons and synaptic interconnections and number of layers. Hardware implementations issues also include data representation, weight storage, precision bits, connection types, on/off-chip learning/training and on/off chip transfer function [9].

Most HNN implementations to date emulate multiple-neurons on a single FPGA device [10]-[13]. Alongside FPGA implementations of neural structures, the SpiNNaker project [14] has recently implemented a very large scale HNN utilising specialised ARM chips comprising 18 dedicated cores. The main drawback of these hardware implementations is that the neural models utilized lack the high level of biological fidelity required to faithfully reproduce realistic neural behaviour. Furthermore, large complex neural structures require communication pipelining techniques that reduces the biological accuracy of the system as the communication is no longer carried out in a truly parallel manner.

To address these issues, it is proposed that the *Si elegans* architecture will utilize a single FPGA per neuron topology similar to work previously described [15], [16]. This approach allows for greater biophysically realistic descriptions of neural and synaptic models with a fully parallel synaptic connection scheme. Furthermore, *Si elegans* differs from previous single FPGA per neuron systems by allowing the user to select neuron models from the neuron model library and parameterise the selected neuron. Neurons in the library are represented in VHDL format. Currently the library consist of 2 simple neuron models, namely the Integrate and Fire (IF) given by [17]:

$$I(t) = C_m \frac{dv}{dt} \tag{1}$$

and the LIF given by [17]:

$$\tau_m \frac{dv}{dt} = -v(t) + R_m I(t) \tag{2}$$

where v is the membrane potential, R_m is the membrane resistance, $I(t)$ is the sum of all synaptic currents at time t, C_m is the membrane capacitance and τ_m is the membrane time constant. These 2 neural models have been described in VHDL and can be freely and arbitrarily parameterized. Ongoing work is focused on VHDL translations of more biophysical realistic neural models such as Hodgkin and Huxley Model [18], FitzHugh-Nagumo Model [19], [20], Morris-Lecar Model [21] and Izhikevich Model [22].

III. *SI ELEGANS* OVERVIEW

The *Si elegans* project is currently at an early developmental stage. In this section an overview of the full *Si elegans* framework is presented.

The project aims to develop a powerful framework capable of performing realistic simulations of the *C. elegans* nervous BNS. Furthermore, it seeks to develop an easy-to-use framework that is freely available for members of the scientific and academic community who are interested in modelling information processing it the *C. elegans* nematode.

An overview of the full *Si elegans* framework architecture is depicted in Fig. 1.

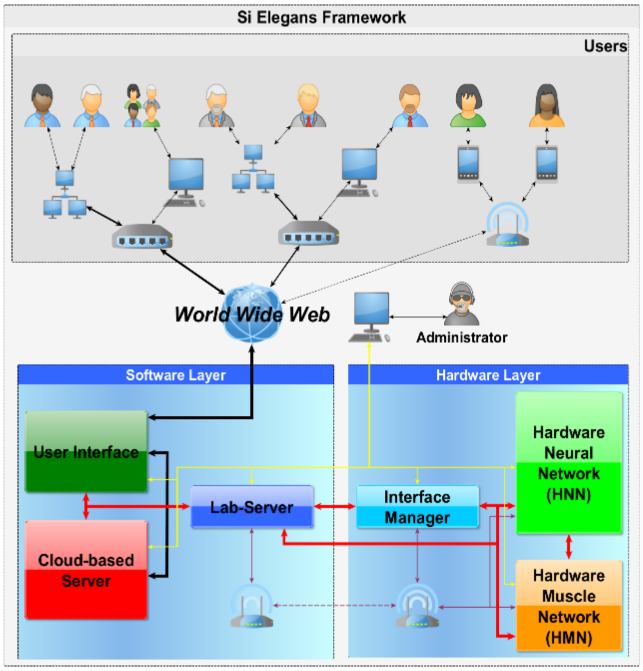


Fig.1 *Si elegans* framework architecture.

Users are assumed to include neuroscientists, biologists, computational intelligence and intelligent systems researchers interested in studying the *C. elegans*' BNS.

It is anticipated that users connect to *Si elegans* platform via a Web Portal using a variety of computational devices. Users can activate *Si elegans* using the User Interface (UI).

The UI provides an advanced graphical Hardware Neuron Network (HNN) and Hardware Muscle Network (HMN) definition environment where Users can:

- Select existing parameterisable neuron models from the hardware neuron model library;
- Create their own neuron models and networks using a range of user friendly software tools (which automate the creation of hardware neuron descriptions, hardware synthesis, place and route of neuron models to selected FPGA technology and creation of FPGA bit configuration files, while hiding these processes from the user);
- Configure the HNN neuron parameters;
- Configure the User Interface's simulation parameters (stimulation pattern etc.);
- Configure the HMN parameters;
- Configure the remote *Si elegans* hardware framework;
- Prepare specific *Si elegans* simulation experiments;
- Schedule experiments, interact with the Virtual Arena during simulations and with the *Si elegans* internal model elements;
- Download and analyse simulation results.

The UI will also provide a dynamic environment both for emulating the worm's physical sensory input interactions with the world, and for observing the resulting behaviour of the nematode in a 3D cinematic virtual environment (Virtual Arena).

The *Si elegans* framework can be divided into a software layer and a hardware layer. The software layer has the following components:

- User Interface – neuron model creation, model distribution and network selection/definition, trace selection, visualization and definition of the simulation environment and collaborative tools;
- Cloud-based Server – social platform services, experiment scheduling and management service, databases, model verification and experiment results an variable retrieval service;
- Lab Server – simulation controller, physics engine, result upload service.

The main aim of the work presented in this paper is the development of the hardware layer, which is shown diagrammatically in Fig. 2. The Hardware layer is composed of the following system components:

- Interface Manager (IM) – configures all the neuron FPGAs with the simulation parameters and neuron model configuration sent from the Software layer. Also manages the simulation, handles spikes from the neuron FPGAs and sends the collated simulations results to the server;
- Hardware Neural Network (HNN) – computes the simulations according to parameters, neuron models and synaptic models previously uploaded by the IM.
- Hardware Muscle Network (HMN) – computes muscle activation based on HNN spikes and the muscle models previously uploaded by the IM.

The complete *Si elegans* hardware system is composed of 330 tightly coupled FPGA boards, arranged in a set of conventional racks which will utilize a novel light based synaptic connection scheme.

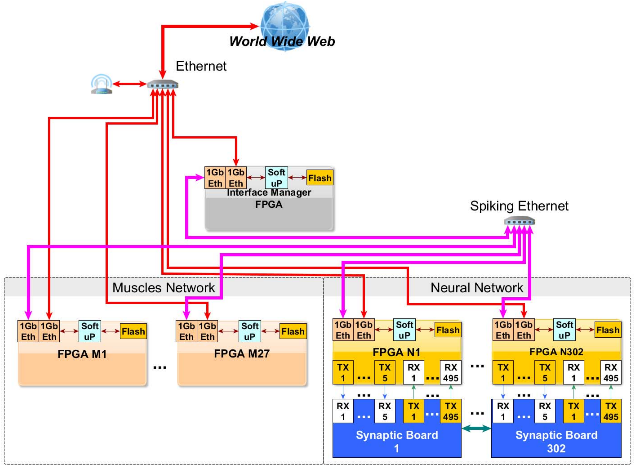


Fig. 2. Si elegans interconnectivity infrastructure.

IV. Small Scale *Si elegans* Hardware Layer

The first step in the hardware development cycle was to implement a small scale prototype system for concept validation. This was comprised of a HNN, consisting of 8 Terasic DE4 FPGA boards and an extra DE4 FPGA board was used as an Interface Manager.

All 9 FPGA boards were interconnected using a specifically designed interconnect board that connected the FPGAs via the 40 pin General Propose Input Output (GPIO) of each FPGA board. These connections were made using standard 40 pin ribbon cable. Each 40 pin GPIO had 36 single-ended I/Os available that were used to connect each single neuron to the other neurons and to the interface manager.

A server computer was connected to the IM through an RS-232 cable. The RS-232 cable was used to send simulation parameters and network configuration from the server computer to the IM and to receive collated simulation results from the IM. Fig. 3 represents the 8 HNN architecture.

A. Hardware layer

The small scale Hardware layer is currently composed of the following system components:

- Interface Manager (IM);
- Server – sends the simulation and network configuration to the IM and receives the simulation results;
- Hardware Neural Network (HNN);

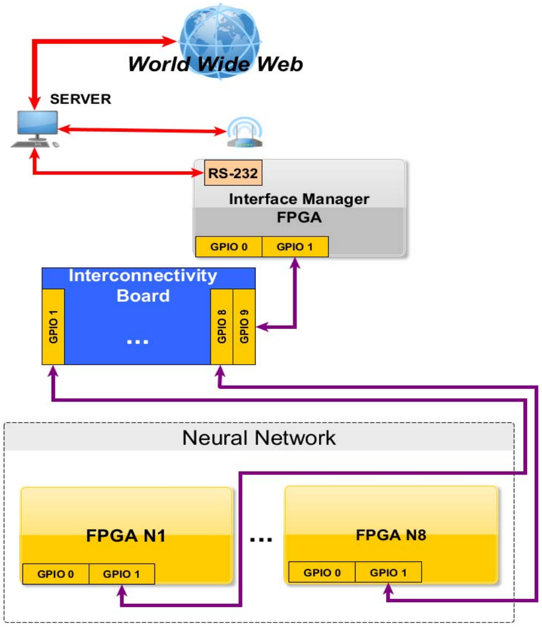


Fig. 3. 8 HNN architecture.

1) *Interface Manager*

The IM is used to manage the simulations. When a new simulation starts it is required to configure all the neuron FPGAs, define time steps, simulation duration and apply/remove stimulus to specific neurons. A Biological Clock controller was also developed to ensure that all the neurons are synchronized in the same biological clock cycle even if some neuron FPGAs have models much more complex than others.

The IM shares 3 channels that are used to exchange data with the neuron FPGAs as showed in Fig. 4.

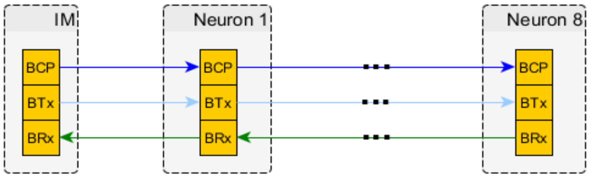


Fig. 4. Connection channels between the IM and the Neuron FPGAs.

Each channel is used to transmit/receive the following data type:

- Biological Clock Pulse (BCP) – channel used to transmit one pulse per timestamp. These pulses are used to inform the neurons that a new timestamp has started;
- Transmit data (BTx) – channel used to broadcast data from the IM across all the neurons;
- Receive data (BRx) – channel used to receive spikes from neurons.

2) *Server*

The server is connected to the IM through the COM port (RS-232). The COM port is configured to work with a baudrate of 115.2 kHz. The simulation software is installed and running on the server and is used to generate simulation and network configurations that are posteriorly uploaded to the IM.

The COM port is also used to receive the simulation results that are uploaded by the IM. Those results are then presented to the user.

3) *HNN*

The HNN is composed of the 8 neuron FPGAs. Each neuron FPGA has available two neuron models described in VHDL. Before each simulation the IM sends the configuration parameters to each neuron FPGA. These parameters include the neuron model for each neuron FPGA, synaptic weights, thresholds, etc.

When a simulation is running the IM sends biological clock pulses (BCP). Each time that a BCP is received, the neuron FPGA checks if there was any spike arriving from the pre synaptic inputs and multiplies those spikes by the synaptic weights which generates the synaptic current *I*. That result is then used for computing changes in the model membrane potential.

After the computation has finished each neuron sends a 2 bit packet, through the BRx channel, to the IM. The first bit is used as start bit and the second bit is set to 1 if there was a spike otherwise 0. Each neuron has up to 125ms to send the confirmation. If the confirmation is not received then an error is triggered by a watchdog that is running on the IM.

When the simulation finishes the IM sends a command through the BTx channel alerting the neuron FPGAs that the BCP that was received was the last pulse. The BTx channel is also used by the IM to apply/remove stimulus to selected neurons.

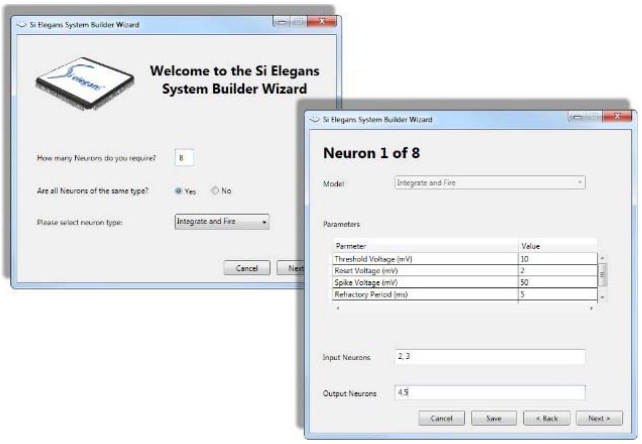
B. *Software layer*

To enable testing of the 8 HNN described above a simple software testbed was developed (*Si elegans* System Builder) which allows the user to describe the desired network configuration to be carried out on the FPGAs. The software is presented in a simple 'wizard' format which asks the user a series of questions about their configuration requirements (Fig. 5).

Currently the user can implement a network with a maximum of 8 neurons chosen from a pre-defined library of neural models (at present, Integrate and Fire or Leaky Integrate and Fire). Furthermore, the user can change any of the available model parameters and can implement the desired network interconnection. When the user is finished designing their network with the wizard two files are created:

- Network Design File: saves the configuration of the network for future use in a simple XML style file;
- Configuration File: Hex data describing the network configuration which can be uploaded to the FPGAs via the Interface Manager.

When the necessary files have been created the user can upload the configuration to the FPGAs and start a simulation. At present, the uploading functionality and simulation control is complete.



Si Elegans System Builder Wizard

Welcome to the Si Elegans System Builder Wizard

How many Neurons do you require? 8

Are all Neurons of the same type? ● Yes ○ No

Please select neuron type: Integrate and Fire

Si Elegans System Builder Wizard

Neuron 1 of 8

Model: Integrate and Fire

Parameters

Parmeter	Value
Threshold Voltage (mV)	10
Reset Voltage (mV)	2
Spike Voltage (mV)	50
Refractory Period (ms)	5

Input Neurons: 2, 3

Output Neurons: 4,5

Fig. 5. Screen shot of the *Si elegans* System Builder Wizard.

V. Results

To test the small scale system developed in section IV, VHDL descriptions of two well-known neural models were developed:

- Integrate and Fire (eqn. 1) [17];
- Leaky Integrate and Fire (eqn. 2) [17].

In these implementations, if the membrane potential v is greater than the firing threshold (v_{th}) then a spike is generated and v is clamped at v_{reset} for set period of time (v_{ref}) thereby implementing the refractory period of the neuron.

Simulation results of these neural models can be seen in Fig. 6 and Fig. 7 where each model has been tested individually using Mentor Graphics Questa Sim 10.1d. In each test case 2 synaptic input spike trains with a frequency of 250Hz and 100Hz respectively were generated to stimulate the neuron. Furthermore, the following parameters were used: C_m = 1nF; R_m = 40MΩ; v_{th} = 10mV; v_{reset} = 2mV; v_{ref} = 5ms; weight = 1.

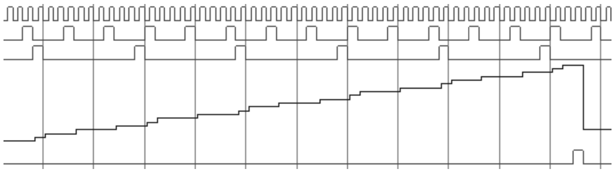


Fig. 6. Simulation results of the Integrate and Fire neuron model on Mentor Graphics Questa Sim 10.1d. Note: the first row is the biological clock, rows 2 and 3 represent synaptic inputs, row 4 represents the neuron membrane voltage and row 5 represents the output spikes of the neuron.

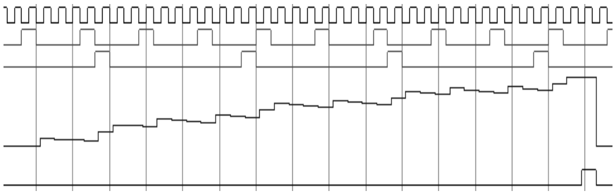


Fig. 7. Simulation results of the Leaky Integrate and Fire model on Mentor Graphics Questa Sim 10.1d. Note: the first row is the biological clock, rows 2 and 3 represent synaptic inputs, row 4 is the neuron membrane voltage and row 5 represents the output spikes of the neuron.

A series of experiments was then performed on the small scale FPGA hardware using the neural network configuration described in Fig. 8, which was developed in the *Si elegans* System Builder. Various configurations of neurons were examined, to ensure that the system could handle different models at the same time.

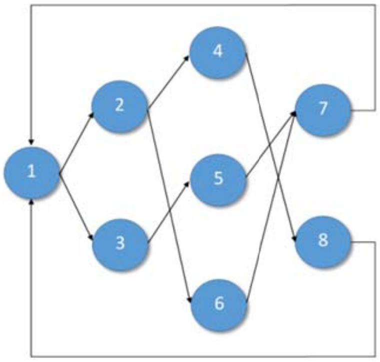


Fig. 8. Neural network topology.

The system builder was used to upload the configuration hex stream through the COM port to the IM which relayed the configuration setup to the neuron boards. After uploading the configuration hex streams the IM started the simulation. The simulations ran for 10000 Biological Clock Cycles (BCC) with a time step of 1ms; total simulation time = 10s.

During the simulations a random stimulus was applied to neuron 1 to ensure that the neuron spiked periodically. Results were sent, by the interface Manager, to the Server each time one or more neurons spiked. Typical results generated by a simulation can be seen in Fig. 9.

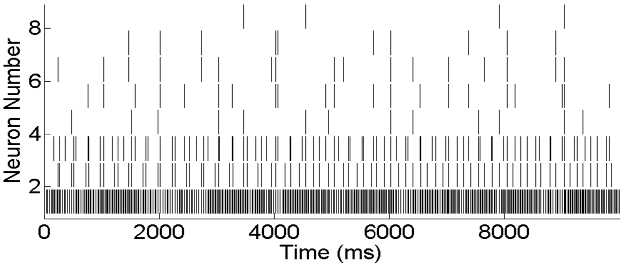


Fig. 9. Typical simulation results.

Note that neuron 1 fires the fastest as this is the neuron which is stimulated by a random stimulus, all other neural activity is a result of neuron 1's activity propagating through the network. As expected, as the information propagates through the network each successive layer's firing rate decreases. Note that the firing rate of neuron 1 is artificially high; this was necessary to cause subsequent neural layers to fire.

Finally, the 8 neurons HNN was also used to develop the communications protocol that will be used in the final system. Furthermore, the use of the COM port helped the authors to validate the data payload protocol that is exchanged between devices, however this type of communication is slower when compared with an Ethernet connection. In the near future the COM port will be substituted by an Ethernet connection which will result in a communications speed increase.

VI. Conclusions and Future Work

Currently there are still many unanswered questions in the field of neuroscience with regards to neural function. This is in part due to the size and complexity of the human brain. Moreover, simulation of neural networks comprised of biophysical realistic models of neurons requires prohibitively long simulation times. Therefore large scale simulations generally utilize phenomenological models which do not capture the rich dynamics of biophysical models. The *Si elegans* project is a multi-platform environment which aims to emulate faithfully the small yet extremely complex nervous system of the *C. elegans* nematode. It will also provide an environment which is freely accessible to neuroscientists, enabling them to easily explore the different neural behaviours and functions of the *C. elegans* worm. Furthermore, the framework will be scalable, allowing neuroscientists to define new neural models and connectomes.

In this paper early work on the hardware architecture of the *Si elegans* framework was described. The preliminary results showed the concept to be viable but highlighted the need for a custom made FPGA board solution to realise a system capable of emulating the full nervous system of the *C. elegans* nematode and which would also be scalable in the future for larger neural system emulation. As a result of the required custom based solution the cost of the hardware will be very high (~500000 euro). However, this custom hardware will provide a very flexible neural architecture permitting neuroscientists the ability to perform neural simulations in real-time using models with a very high degree of biological fidelity. To the best of the authors' knowledge this high level speed and biological realism, in terms of model accuracy and parallelism, has never before been possible with current FPGA or software neural implementations. The chosen high performance FPGA boards will have the power to address effectively the computational complexity associated with processing a single neuron, even with the anticipated high degree of biological realism.

The next stage of this work will be to integrate the optical based synaptic interconnect boards developed by our partners in Istituto Italiano di Technologia (IIT). During this stage all wired synaptic connectivity will be removed and the system will be retested by re-running all simulations previously carried out. This will ensure that developed system is capable of driving and communicating correctly with the new synaptic interconnect boards.

Current ongoing work is also focused on developing a method to allow the user to visualise and analyse results presented by the *Si elegans* hardware system. Note that this is intended as an interim developmental and test tool, and is not intended to replace the full virtual arena being developed by project partners Vicomtech and National University Ireland Galway (NUIG).

Fig. 10 describes the envisaged layout of this part of the test software. This software will enable the user to interact with the hardware in several ways.

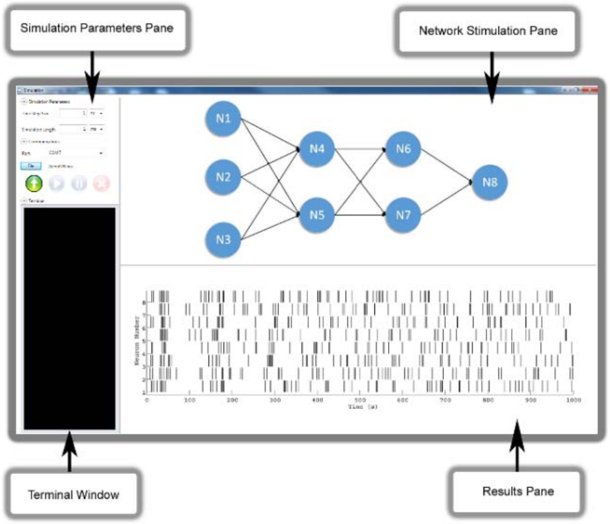


Fig. 10. Envisaged layout of simulation test software.

Firstly, in the Simulation Parameters Pane the user can specify the time step of the simulation, the required simulation length, the COM port used to connect to the hardware and the file to be uploaded. Below this there will be a Terminal Window which will feed information back about the upload success and simulation data such as neural stimulation that has been applied and simulation progress. The Network Stimulation Pane will display a graphical representation of the configuration network and will allow the user to interact with the simulation by applying stimulus currents to any of the neurons at any time throughout the simulation. Finally, the Results Pane will feed back simulation results to the user in real time and will display information such as neuron firing times and synaptic weights. The main aim of this software is to provide the Intelligent Systems Research Centre (ISRC) with a full software test bed for testing the developed FPGA neural emulation platform. Preliminary results

Furthermore, the software will also enable the authors to fully develop and test a communication protocol between the software and hardware layer which can then be used with the UI.

Finally, the small scale system will be developed to full scale with custom made FPGA boards and integrated into the complete *Si elegans* system with other system components developed by our project partners NUIG, IIT and Vicomtech. NUIG are currently focused on implementing a module which allows users to define new neural models with a high degree of biological realism using various computational languages which are then automatically translated to HDL for use with the *Si elegans* hardware. Therfore, a user who is not an expert in FPGA technology is completely removed from the computational complexity of the underlying architecture. As previously discussed, IIT are currently developing novel synaptic connection modules which will allow wireless connectivity of the Neuronal harware. The software layer UI and virtual arena, which grants the user access to the framework and provides simulation analysis tools to the user, is currently under development by Vicomtech and NUIG.

ACKNOWLEDGEMENTS

The research leading to these results has been partially supported by the *Si elegans* project, which has received funding from the European Community's 7th Framework Programme under the Neuro BioInspired Systems Project Grant agreement 601215;

REFERENCES

[1] WormAtlas, "Anatomy of Caenorhabditis elegans," 1 May 2014. [Online]. Available: http://www.wormatlas.org/index.html.

[2] T. B. Achacoso and W. S. Yamamoto, *Ay's Neuroanatomy of C. Elegans for Computation*, USA: Boca Raton: CRC Press, 1991.

[3] K. Oshio, Y. Iwasaki, S. Morita, Y. Osana, S. Gomi, E. Akiyama, K. Omata, K. Oka, and K. Kawamura, "Database of synaptic connectivity of c. elegans for computation," 1 May 2014. [Online]. Available: http://ims.dse.ibaraki.ac.jp/ccep/.

[4] L. Varshney, B. Chen, E. Paniagua, D. Hall, and D. Chklovskii, "Structural Properties of the Caenorhabditis elegans Neuronal Network," *PLoS Comput Biol,* DOI: 10.1371/journal.pcbi.1001066, 2011.

[5] N. Chatterjee and S. Sinha, "Understanding the mind of a worm: Hierarchical network structure," *Progress in Brain Research,* vol. 168, pp. 145-153, 2007.

[6] O. Hobert, "Behavioral plasticity in c. elegans: Paradigms, circuits, genes," *Journal of Neurobiology,* vol. 54, no. 1, pp. 203 - 223, 2003.

[7] S. Jung and S. S. Kim, "Hardware Implementation of a Real-Time Neural Network Controller With a DSP and an FPGA for Nonlinear Systems," *Industrial Electronics, IEEE Transactions on*, vol. 54, no. 1, pp. 265 - 271, 2007.

[8] J. Zhu and P. Sutton, "FPGA Implementation of Neural Networks - A Survey of a Decade of Progress," in *Proc. 13th International Conference on Field-Programmable Logic and Applications (FPL 2003)*, Lisbon, Portugal, 2013.

[9] J. Misra and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress," *Neurocomputing,* no. 74, pp. 239 - 255, 2010.

[10] B. Glackin, T. M. McGinnity, L. P. Maguire, Q. X. Wu, and A. Belatreche, "A Novel Approach for the Implementation of Large Scale Spiking Neural Networks on FPGA Hardware," *Computational Intelligence and Bioinspired Systems,* vol. 3512, pp. 552-563, 2005.

[11] C. Ang, A. McEwan, A. van Schaik, C. Jin, and P. Leong, "FPGA implementation of biologically-inspired auto-associative memory," *Electronics Letters* , vol. 48, no. 3, pp. 148 - 149 , 2012.

[12] T. Iakymchuk, A. Rosado, J. Frances, and M. Batallre, "Fast spiking neural network architecture for low-cost FPGA devices," in *Proc. 7th International Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC),* pp. 1 - 6 , 2012.

[13] S. Pande, F. Morgan, T. B. Seamus Cawley, G. Smit, B. McGinley, S. Carrillo, J. Harkin, and L. McDaid, "Modular Neural Tile Architecture for Compact Embedded Hardware Spiking Neural Network," *Neural Processing Letters,* vol. 38, no. 2, pp. 131-153, 2013.

[14] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker System Architecture," *Computers*, *IEEE Transactions on,* vol. 62, no. 12, pp. 2454-2467, 2013.

[15] K. Mohamad, M. Mahmud, F. Adnan, and W. Abdullah, "Design of single neuron on FPGA," *2012 IEEE Symposium on Humanities, Science and Engineering Research (SHUSER),* pp. 133 - 136 , 2012.

[16] V. Salapura, M. Gschwind, and O. Maischberger, "A fast FPGA implementation of a general purpose neuron," in *Proc. Field-Programmable Logic Architectures, Synthesis and Applications*, Prague, Czech Republic, Springer Berlin Heidelberg, 1994, pp. 175-182.

[17] W. Gerstner and W. M. Kistler, Spiking Neuron Models, Cambridge, U.K.: Cambridge University Press, 2002.

[18] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology,* vol. 117, no. 4, pp. 500–544, 1952.

[19] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane.," *Biophysical Journal,* vol. 1, no. 6, pp. 445-466, 1961.

[20] J. Nagumo, S. Arimoto, and S. Yoshizawa, "An Active Pulse Transmission Line Simulating Nerve Axon," *Proceedings of the IRE,* vol. 50 , no. 10, pp. 2061 - 2070, 1962.

[21] C. Morris and H. Lecar, "Voltage oscillations in the barnacle giant muscle fiber," *Biophysical Journal,* vol. 35, no. 1, pp. 193–213, 1981.

[22] E. Izhikevich, "Simple model of spiking neurons," *Neural Networks, IEEE Transactions on,* vol. 14, no. 6, pp. 1569 - 1572, 2003.