Behavioral Simulation of Biological Neuron Systems using VHDL and VHDL-AMS

Julian A. Bailey, Peter R. Wilson, Andrew D. Brown
School of Electronics and Computer Science,
University of Southampton, UK
prw@ecs.soton.ac.uk

John Chad
School of Neuroscience,
University of Southampton,
UK

ABSTRACT

The investigation of neuron structures is an incredibly difficult and complex task that yields relatively low rewards in terms of information from biological forms (either animals or tissue). The structures and connectivity of even the simplest invertebrates are almost impossible to establish with standard laboratory techniques, and even when this is possible it is generally time consuming, complex and expensive. Recent work has shown how a simplified behavioural approach to modelling neurons can allow "virtual" experiments to be carried out that map the behaviour of a simulated structure onto a hypothetical biological one, with correlation of behaviour rather than underlying connectivity. The problems with such approaches are numerous. The first is the difficulty of simulating realistic aggregates efficiently, the second is making sense of the results and finally, it would be helpful to have an implementation that could be synthesised to hardware for acceleration. In this paper we present a VHDL implementation of Neuron models that allow large aggregates to be simulated. The models are demonstrated using a system level VHDL and VHDL-AMS model of the C. Elegans locomotory system.

1. INTRODUCTION

1.1. Biological Neurons

Neurons are body cells specialized for signal transmission and signal processing. Figure 1 shows the typical structural characteristics of a neuron. It has a cell body (or soma) and root-like extensions called neurites. Amongst the neurites, one major outgoing trunk is the axon, and the others are dendrites. The signal processing capabilities of a neuron is its ability to vary its intrinsic electrical potential (membrane potential) through special electro-physical and chemical processes. A single neuron receives signals from many other neurons, (typically in order of 10,000 for mammals) at specialized sites on the cell body or on the dendrites, known as synapses.
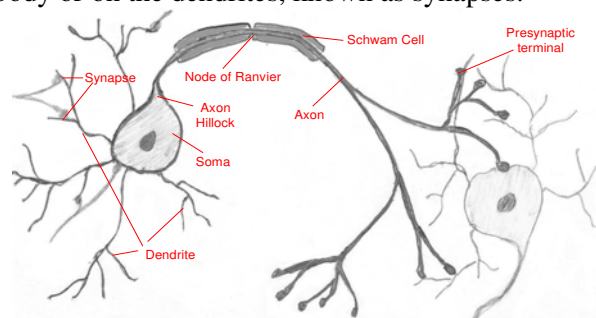

Fig 1 Diagram of a generic neuron

Synapses receive signals from a pre-synaptic neuron and alter the state of the postsynaptic neuron (the receiver neuron) and eventually trigger the generation of an electric pulse, the action potential (a spike), in the postsynaptic neuron. This action potential is initiated at the rooting region of the axon, the axon-hillock, and it subsequently travels along the axon sending information signal to the other parts of the nervous system.

1.2. Neuron Models

Models of neurons can be created at various level of abstraction ranging from sub-molecular level to network level. The pioneering Hodgkins–Huxley model [3] and other compartmental models based on it [4-6] model variation in cell membrane voltage using ion channels kinetics. Experiments have shown that the membrane voltage variation during the generation of an action potential is generally in a form of a spike (a short pulse - figure 2); and the shape of this pulse in neurons is rather stereotype and mathematically predictable. Models such as "*Integrate and Fire*" are built with an assumption that the timing of a spike is the information carrier and not the shape of the spike [8-10].
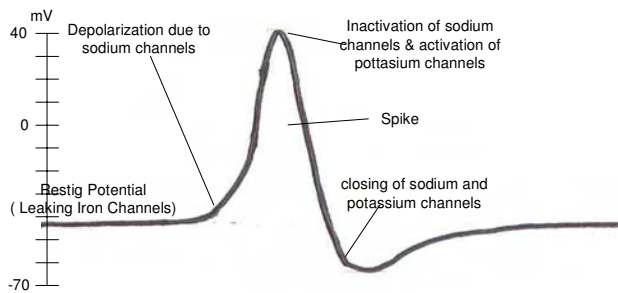
153

Fig 2 Typical Action Potential

This bio-physical approach is suitable for electro-physicists to develop quantitative description based on experimental results, but these models are computationally expensive and unsuitable for the simulation of large aggregate of neurons.

An alternative approach is to develop highly abstract neuron models encapsulating the essential functionality of a neuron relevant for network behaviour in order to develop understanding of network population dynamics. Binary neuron models (McCulloch and Pitts [11]), the Perceptron model (Rosenblatt [12]) and the Spiking-rate model [7] represent this end of the spectrum in neuron modelling and they are widely used in artificial neural networks. These models are computationally efficient but they are generally too abstract to be used in biologically realistic simulations.

An intermediate approach uses cell automata models make use of state automaton to capture the functionality. Various types of behaviour can be captured at an abstract level using this technique. They are computationally much more efficient than the traditional biophysical models and they have been successfully used for biological neuron simulation. [7] and this type of approach was applied in this work.

1.3. Simulators

The simulation of biological neurons is computationally expensive and designing a practical simulation platform is an area of an active research. Traditionally the biophysical models have been designed using continuous simulation. However, the evolution of highly computationally efficient discrete simulation techniques has made this approach more attractive choice for larger networks.

Several computer software packages are available for simulation of a biological neuron network. Most of them are based on continuous simulation and biophysical models, and therefore they are computationally expensive. Only two packages use event-driven discrete simulation: SPIKE/NEURON, Nishwitz-Gluender. They are, however, not optimised for large networks. The simulation system developed in [7] has successfully used event driven simulation with abstract modelling techniques for biologically realistic neuron systems and it has achieved very good performance for neuron systems of about 10^5 neurons [7 p112]. Previously we have implemented a large scale network model using System-C, and while this has been quite effective in modelling systems, there have been some problems with this approach. The first has been the memory inefficiency inherent in System-C, and the second has been the software approach. The VHDL models in this paper have been developed to operate potentially on a hardware platform, and to interact with other specialist hardware modules and in addition to interface with VHDL-AMS models of the muscle system.

2. Neuron Model

The neuron is a complex entity, it receives multiple inputs from synapses and although it has a single output, this output can be mapped to the inputs of many synapses. A Neuron can be easily broken down into blocks, each of which can be modelled separately. The Neurons in the model implemented here have been resolved into three separate components:

2.1. Threshold Block

This is responsible for determining when the membrane voltage goes above the excitatory threshold or below the inhibitory threshold. The variable w_{sum} tracks the current membrane voltage due to synaptic activity from the multiple input synapses When the value of w_{sum} is equal to or above the excitatory threshold th_e, an *on* signal is passed to the burst block to indicate that it should begin firing a burst of action potentials. In a similar fashion, when the value of w_{sum} is equal to or below the inhibitory threshold th_i then an off signal is sent to the burst block, terminating any current bursts.

2.2. Oscillator Block

This is required to make a neuron that fires regularly with a predefined period. This block is only active in neurons that drive network

behaviour. Neurons with this block active are not usually used as postsynaptic targets.

2.3. Burst Block

This is responsible for producing the action potentials from the neuron and bursts of action potentials. Action potentials are fired with a duration defined by t_{ap} followed by a refractory period t_{ref}. The length of a burst is defined by the parameter n_{burst}, if this number is -1 then the burst is of infinite length and can only be terminated by an inhibitory signal from the threshold block. A more detailed description of the theoretical basis of the model is available in the previous work by Claverol [7] and Modi [13]

3. Synapse Model

The synapse model is considerably simpler than the neuron model due to the single input and single output nature of this block. The synapse block receives a single input from the burst block of a neuron and has a single output to the threshold block of a different neuron. The block operates as follows:

1. First the receipt of an action potential triggers a timer which models the delay t_{del} of the neurotransmitter crossing the synapse from pre- to post-synaptic neuron.

2. After t_{del} has passed, the synapse increments its output by an amount defined by the value of w_{syn} which represents the synaptic weight. The synapse increments the output by this amount for a period of time defined by the parameter t_{dur} which is the duration.

3. Finally, once t_{dur} has passed the synapse decrements the output by w_{syn}.

It is entirely possible that if the delay and duration parameters of the synapse are longer than the interval between action potentials that the synapse could become activated whilst it is already active. This has to result in the increment of the output by w_{syn} again. The challenge is to keep track of these changes and make sure that the increment and decrement events of the output appear as if they have been happening independently from each other. Once again a more detailed description of the model is available in the previous work by Claverol [7] and Modi [13]

4. VHDL Neuron Network Models

4.1. Introduction

VHDL is a powerful language and allows the creation of standard sets of tools for the creation of network designs. The Neuron and Synapse models were compiled into a library which allows the referencing each model entity as components where the various model parameters could be specified.

Connectivity is specified by defining the various signals between neurons and synapses. Outputs from neurons are connected to synapses and synapses are connected to the input of another neuron. This simple regular connectivity makes it simple to generate your own networks.

4.2. Neuron Library in VHDL

Fundamental to the implementation of networks of neurons using VHDL is the creation of a standard library for general purpose use. One of the key aspects in using VHDL is the ability to mix logic, real and *time* based signals. The entity for the basic Neuron model has the following basic signal interface:

```
signal gamma : in
    real_vector(
        (NumberSynapses-1)
        downto
        0
    );
signal alpha : out std_logic;
signal Tosc  : in  time;
signal Tph   : in  time;
signal reset : in  std_logic
```

The flexibility inherent in the model is based around the configurable synapse input (**gamma**), and the output pulse train is defined using the logic signal **alpha**. In addition, rather than fixed generics, the temporal signals **Tosc** and **Tph** are defined. Finally, the neuron can be "reset" using a logic signal (**reset**). While this is perhaps biologically unrealistic, at least it allows an initial condition to be defined.

The architecture of the Neuron contains three main elements – a threshold function, a burst generator and an oscillator. These are all natural to be implemented in VHDL.

4.3. C. Elegans Locomotory Neuron Model

The C. Elegans Locomotory Neuron System is implemented using the VHDL library of Neurons and Synapses described in the previous section. The key element from a *systems* perspective is that the model can be reset and directionally controlled (allowing interface to sensory neurons or a "higher level" abstract model) and interfaced to biologically realistic muscle models. The

interface is therefore completely defined using standard logic signals:

```
signal Reset  : in  std_logic;
signal FwdBwd : in  std_logic;

-- Dorsal Motor Neuron Outputs
signal MSCD0: out std_logic;
     …
signal MSCD9: out std_logic;

-- Ventral Motor Neuron Outputs
signal MSCV0: out std_logic;
     …
signal MSCV9: out std_logic
```

Using this interface it is straightforward to drive the model and interface to more broadly based biological test scenarios.

5. C. Elegans Locomotory System

C. Elegans is a free living nematode which has a generation time of about 3.5 days. The nematode can grow to a length of 1.3mm long and a diameter of 80μm if there is a sufficient supply of food available [15]. C. Elegans is easy to culture in the lab on bacterial lawns grown on an agar substrate. They normally inhabit the interstices between damp soil particles or in rotting vegetation.

The locomotion of C.Elegans is achieved by dorsal-ventral movement of the body which produces a sinusoidal wave which propagates along the length of the body. Body movements are produced by four strips of muscle cells that run in four quadrants between longitudinal ridges inside the body cavity.

The nervous system of C.Elegans is constructed from 302 neurons which were mapped using electron microscopy, laser ablation and mutant studies. A complete map of the neurons in the nervous system was published in 1986 by White *et al*. This provides us with a useful map of the neurons in the body but provides little definitive information about how the neurons are connected and what types of synapses are formed.

The subset of neurons which make up the locomotory system only numbers approximately 80 neurons [18-19]. These neurons enable the worm to move in a forward and backward direction. Coiling and omega types turns [20] are also normal locomotory patterns for this nematode.

6. C Elegans Locomotory Model

A C.Elegans locomotory model was conceived in 2000 by Enric Claverol [7] based on data from White *et al* [15]. There is limited electrophysiological data available about C. Elegans [21], therefore models of the locomotory system are based upon a mixture of anatomical data from White *et al* [15] and analysis of video recordings of the animals behaviour.
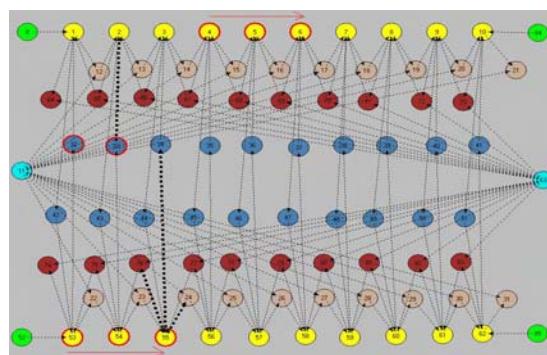

Fig 3 : C Elegans Locomotory Model

The picture shown in Figure 3 was generated by a specially designed user interface developed in-house to allow straightforward design of the networks for the simulator without having to generate the VHDL test bench manually.

The figure depicts the different classes of neuron in the locomotory system. For simplicity the original names for the neurons given by White *et al* [15] have been used. The model has also been adjusted so that it looks more symmetrical which makes it easier to understand and separate the behaviour of each subset of neurons.

The model includes eight classes of neurons (VA, VB, VD, DA, DB,, DD, and AVA & AVB) and two groups of muscle cells (MSCD and MSCV). Excluding the AVA and AVB classes, all other neurons are preceded by a V or D which defines which side of the body the neuron is on. The layout of the neurons is such that the layout on one side of the body is the mirror image of that on the other.

The neurons AVA, VA and DA are responsible for backward locomotion where as AVB, VB and DA are used for forward locomotion [22]. The neurons VD and DD are responsible for cross-lateral inhibition. MSCV and MSCD represent ventral and dorsal muscle cells respectively.

There are four extra neurons (NRD,NRV, TSD and TSV) which represent signal sources encapsulating activity that comes from groups of neurons in the head or tail sections of the nematode.

The nematode was simulated during forward, backward and whilst reversing direction. Results were compared with those obtained in Claverol [7] and Modi [13].
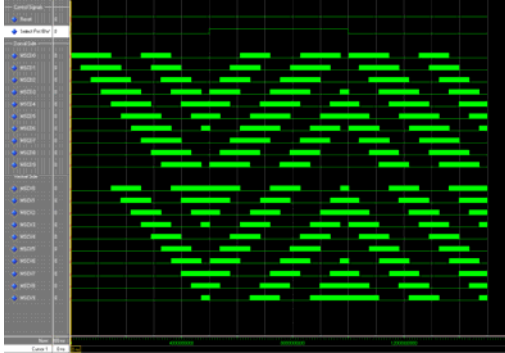

Fig 4: VHDL Simulation of C Elegans Locomotion

Figure 4 shows the waveform trace of the electrical activity in the muscle cells. The trace shown shows the waves of action potentials propagating from the head to the tail on one side of the worm and then on the other side with a phase lag. Midway through simulation the parameters of the model were changed in the VHDL test bench and the worm reversed direction with the muscle activity now propagating in the opposite direction. Again the parameters are switched to demonstrate the ability of the model to cope with arbitrary changes in direction. This behaviour agrees with the activity shown in Claverol [7] and Modi [13].

7. Interfacing the motor neurons to muscle using VHDL-AMS

In order to begin to understand the behaviour of the nematode in the context of the biological environment we need to extend the scope of the VHDL to the analog domain – and this is where the VHDL-AMS models are ideal. If we consider a simple version of the muscle structure of the C Elegans using 10 dorsal and 10 ventral muscle segments, we can create muscle models and also use our VHDL control network of neurons to drive the muscle system. Figure 5 shows the outline of the simple muscle structure. Clearly each model can be as simple or complex as we require, but the basic idea is appropriate to see the operation of the structure in a physical context.
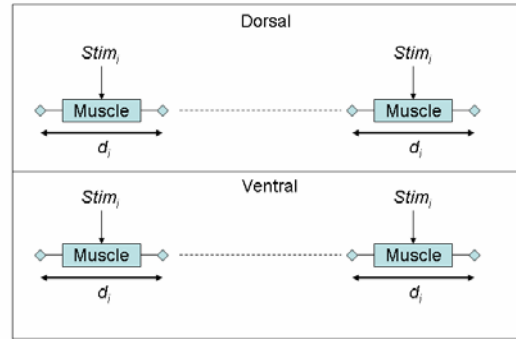

Fig 5 : Muscle Structure – VHDL-AMS

The muscle models are implemented in VHDL-AMS[1] using a translational mechanical basis, with a simple spring-like approach. A damping model is used to give the 1ms physical time constant in real muscle, and the basic movement can be observed in this model. A simplified version of the model is given for brevity below:

```
Library ieee;
Use ieee.mechanical_systems.all;

Entity muscle is
    Port (
            Din : IN std_logic;
            P : translational_pos;
            M : translational_pos
    );
End entity muscle;

Architecture basic of muscle is
    Quantity pos across
    force through p to m;
    Quantity p_in;
    Signal d_p_in : real;
Begin
    Process (din)
    Begin
      If (din='1') then
        D_p_in = 0.1 * len;
      Else
        D_p_in = len;
      End if;
    End process;

    D_p_in == pos'dot/0.628 + pos;
    Break on din;
End architecture basic;
```

As can be seen in figure 6, the model contracts based on the digital stimulus (din) and the contraction is limited with a first order filter response.

Using this basic muscle model, and the complete Neuron System Model in VHDL, it

---

[1] This model has been implemented in VHDL-AMS and Saber (MAST)

is possible to now simulate the complete C Elegans Locomotory System in an integrated simulation. The advantages of this are now obvious. With the interface to the "real" environment established using the VHDL-AMS modelling approach, the system can be tested under a variety of standard "Lab" conditions and compared against observed behaviour in a wet lab.
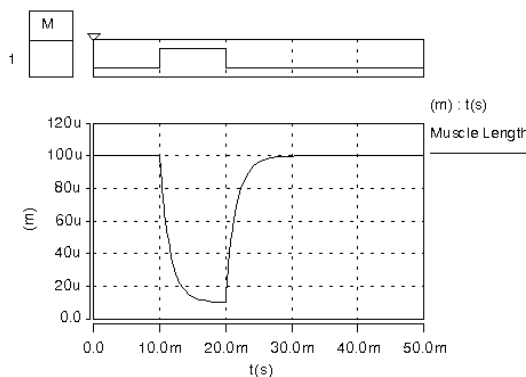

Figure 6 : Simulation of C Elegans Muscle

8. Conclusions

This paper has shown how a network model for neurons and synapses can be implemented in VHDL and demonstrates the locomotory mechanism in a standard laboratory animal (C Elegans) in a simple and powerful manner. The extension of the model to include simple mechanical models offers the potential of integrating not only the neuron behaviour but also the biological system feedback from chemical and environmental stimuli.

This last aspect is the subject of ongoing research at the University of Southampton.

9. References

[1] Shastri, Lokendra, "The role of temporal coding in the processing of relational information in the mind-brain", *Proceedings of the International Joint Conference on Neural Networks*, Vol. 2, 2003, pp1379-1384

[2] Chen, Fang, Xu, Jinghua; Gu, Fanji; Yu, Xuhong; Meng, Xin; Qiu, Zhicheng, "Dynamic process of information transmission complexity in human brains, *Biological Cybernetics*, v 83, n 4, 2000, p 355-366

[3] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve", *Journal of Physiology*, Vol. 117, pp500-544, 1952.

[4] Rinzel and Rail W. "Transient response in a dendritic neuron model for current injected at one branch", *Biophysics Journal*, Vol. 14, pp759-790, 1974.

[5] Rail W., "Core conductor theory and cable properties of neurons.", *Handbook of Physiology.*, pp39-97. American Physiological Society, 1977.

[6] Rail W., "Electrophysiology of a dendritic model.", *Biophysics. Journal.*, Vol. 2, pp145-167, 1962

[7] Enric T. Claverol , "An event-driven approach to biologically realistic simulation of neural aggregates", PhD thesis, University of Southampton, September 2000

[8] Christodoulou G., Bugmann G., and Clarkson T.G. The temporal noisy-leaky integrator neuron model. In Beale R. and Plumbley M.D., editors, Recent advances in neural networks. Prentice Hall, 1993.

[9] Smith L.S. A one-dimensional frequency map implemented using a network of integrate-and-fire neurons. In Proceedings of the 8th International Conference on Artificial Neural Networks, pages 991-996. Springer, 1998.

[10] Smith L.S., Nischwitz A., and Cairns D.E. Synchronization of integrate-and-fire neurons with delayed inhibitory lateral connections. In Marinaro M. and Morasso P.G., editors, Proceedings of ICANN94, pages 142-145. Springer-Verlag, 1994.

[11] McCulloch W.S. and Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull, of Math. Biophysics, 5:115-133, 1943.

[12] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychol. Rev., 65:384-408, 1958. Hopfield J.J. Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Natl. A cad. Sci.,81:3088-3092, 1984.

[13] Modi S.S, "Design of SystemC Framework for Simulation of Biological Neuron System", MSc Project Report, University of Southampton, November 2003

[14] http://www.systemc.org

[15] White J.G., Southgate E., Thomson J.N., and Brenner S. "The structure of the nervous system of Caenorhabditis elegans." Phil. Trans. R. Soc. Lond. /Biol],314:1-340, 1986.

[16] Ware R.R., Clark D., Crossland K., and Russell R.L. "The nerve ring of the nematode C. elegans: sensory input and motor output." J. Corn p. Neuro 1.,162:71-110, 1975.

[17] http://www.ee.eng.hawaii.edu/~msmith/ASICs/HTML/Verilog/LRM/HTML/15/ch15.2.htm

[18] White J.G., Southgate E., Thomson J.N., and Brenner S. "Structure of the ventral nerve cord of caenorhabditis-elegans." Phil. Trans. R. Soc. Lond./Biol, 275(938):327–348, 1976.

[19] Wicks S.R. and Rankin C.H. "The integration of antagonistic reflexes revealed by laser ablation of identified neurons determines habituation kinetics of the caenorhabditis elegans tap withdrawl response.", J. Comp. Physiol. A, 179(5):675-85, 1996

[20] Suzuki M., Tsuji T. Ohtake H. "A model of motor control of the nematode C. Elegans with neuronal circuits, Artificial Intelligence in Medcine.", 35:75-86, 2005

[21] Goodman M.B., Hall D.H., Avery L., Lockery S.R.,"Active currents regulate sensitivity and dynamic range in C. Elegans neurons", Neuron, 20(4):763-72, 1998.

[22] Niebur E. and Erdos P, "Theory of the locomotion of nematodes: control of the somatic motor neurons by interneurons.", Math. Biosci., 118(1):51-82.