Behavioural Simulation and Synthesis of Biological Neuron Systems using VHDL

J.A. Bailey¹, P.R. Wilson¹, A.D. Brown¹ and J.E. Chad²
School of Electronics and Computer Science¹ &
School of Biological Sciences²
University of Southampton,
SO17 1BJ,
U.K.
Introduction
The Nervous System
VHDL Neuron Model
VHDL Neuron Network Model
Synthesis
Introduction

- Biologists & Engineers
- Investigate Neuron Structures
- Biological experiments
  - Live Tissue
  - Can’t Establish Connectivity
- Behavioural Modelling
  - Simulate behaviour
  - Determine network characteristics

Stained Rat Cortical Neurons [1]

Motivation

- Biologically realistic simulation
- Efficient Models

- Reduce run time
  - Hardware acceleration
  - Real Time Simulation
  - Virtual Animal/Nervous System

- Reusable Libraries
  - Easily configurable
Introduction

The Nervous System

VHDL Neuron Model

VHDL Neuron Network Model

Synthesis
Introduction

The Nervous System

VHDL Neuron Model

VHDL Neuron Network Model

Synthesis
History of the VHDL Model

- **Cell Automata Model**
  - Enric Claverol, 2000

- **System C Neuron Model**
  - Sankalp Modi, BMAS, 2004

- **VHDL Biologically realistic neuron model**
  - Julian Bailey, BMAS 2007
Model Overview

Synapses → Dendritic Tree → Soma

Threshold Block

Burst Block

Oscillator

Axon Hillock

Axon

Ref

On

Off
More Model Details

- **Neuron Library - LibNeuron**
  - Contains all Sub Components

- **Three top level entities**
  - Neuron 1
  - Neuron 2 (Oscillator Activated Neuron)
  - Synapse

- **Each Configurable using Generics**
Sub Component – Threshold Block

Initial Reset State
Index = 0
Sum = 0

Sum = Sum + SynWeight[index]

Increment index
No

Index >= # of Synapses?

Yes
Increment Index
No

Index >= Max # of Synapses?

Sum >= Ex Thld

No

Sum >= In Thld

Yes

AbvExThld = ‘1’
BellInThld = ‘0’

AbvExThld = ‘0’
BellInThld = ‘0’

AbvExThld = ‘0’
BellInThld = ‘1’
Sub Component – Burst Block

Start

STATE: OFF
Output: ‘0’
if Excite = ‘1’
Start AP Counter

STATE: Fire
Output: ‘1’
If Inhibit signal = ‘1’ then
Burst Counter = 0

STATE: Refractory
Output: ‘0’
Subtract 1 from Burst Counter,
If Inhibit signal = ‘1’ Burst Counter = 0

Counter Finished?
No
Yes

Burst Counter = 0?
No
Yes
Neuron 1

*“Standard” Neuron*

*Activated by Synapses*

Example Parameters: - Threshold 3, AP Time 1 ms, Ref. Time 2 ms, Burst 5.
Neuron 2

- Neuron to drive network
- Activated periodically by oscillator

Example Parameters: - Period 16 ms, Phase 2 ms, AP Time 1 ms, Ref. Time 2 ms, Burst 3.
Connect Neurons

Model Delays & Activation Duration

Can be activated once already active

Example Parameters:- Delay 1ms, Duration 1ms, Weighting 1.
Inside the Synapse

Transmission Delay

Initial State

Start Signal

No

Yes

Counter Available

No

Yes

Start a Delay Counter

Duration Timer

Initial State

Delay Finished

No

Yes

Counter Available

No

Yes

Start a Dur. Counter

Weighting Accumulator

Reset State

Output = 0

Idle State

Delay Finished

No

Yes

Decrease Output By Synaptic Weight

Increase Output By Synaptic Weight
Introduction

The Nervous System

VHDL Neuron Model

VHDL Neuron Network Model

Synthesis
Model Verification

⇒ Nematode, C Elegans

⇒ 302 Neurons

⇒ Extensively studied

⇒ Connections partially known

- Yellow: Muscle
- Blue: Cross Inhibition
- Light Brown: Forward Interneuron
- Red: Backward Interneuron
LibElegans VHDL Library

- Animal Library
  - ElegansLoco

- Specifies Generics
  - Creates Types
  - Repeated Pattern “Loco Unit”

![Diagram of neural network with symbols for muscle, cross inhibition, forward interneuron, and backward interneuron]
C Elegans
Introduction

The Nervous System

VHDL Neuron Model

VHDL Neuron Network Model

Synthesis
Synthesis

- Previous work
  - C Elegans Design Size (200k FG, 85k DFF’s)

- Optimisation
  - 2 Types of Neuron
  - User Definable Length Counters
  - Up Counters only
  - Disable Pins on all blocks

- Design fits on many more devices
  - 60,506 FG’s & 48,891 DFF’s

- Hardware acceleration

Synthesis Example

A section of C Elegans (Mini Elegans)

- 33 Neurons, 50 Synapses
- Only goes forwards
Hardware Acceleration

➡️ Traditional Simulations
  ➡️ Hours -> Days
  ➡️ Example: Mini Elegans (6 Sec) – 3 hours 20 mins
  ➡️ Example: C Elegans (15 Sec) – 32 hours 12 mins

➡️ In Hardware
  ➡️ Real-Time
  ➡️ Example: Mini Elegans (6 Sec) – 6 Seconds! (2000x Faster)
  ➡️ Near Future : C Elegans (15 sec) – 15 Seconds! (7728x Faster)

➡️ However...
  ➡️ Limited by Current FPGA Technology
  ➡️ Large Scale Multi-Processor Hardware Simulation Frameworks
     ➡️ Spinnaker - Univ. Southampton, UK & Univ. Manchester, UK
Summary

→ Synthesizable Neuron Library

→ Post-Synthesis Verification
  → Compared against previous work
  → C Elegans

→ Post Synthesis design
  → Reasonably sized
  → Download onto FPGA
  → Hardware acceleration
  → Virtual Animal/Nervous System on FPGA!!!
Thanks For Listening!

Any Questions?