

---

# **WebInterfaceDocs Documentation**

***Release 0.0.1***

**Nikesh Lama**

**Jun 28, 2018**



<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview of the platform . . . . .	1
1.2	Requirements . . . . .	2
1.3	Disclaimer . . . . .	3
1.4	Bug reports . . . . .	3
<b>2</b>	<b>Platform Features</b>	<b>5</b>
2.1	User Account . . . . .	5
2.2	Simulation Parameters . . . . .	5
2.3	Neuron Models . . . . .	7
2.3.1	Integrate and Fire . . . . .	7
2.3.2	Leaky Integrate and Fire . . . . .	7
2.3.3	Izhikevich . . . . .	8
2.3.4	Custom Models . . . . .	8
2.4	Topology . . . . .	9
2.4.1	Architecture . . . . .	9
2.4.2	Interconnections . . . . .	9
2.5	Stimulation . . . . .	13
2.5.1	Step current . . . . .	13
2.5.2	Frequency based . . . . .	13
2.5.3	Biphasic stimulation . . . . .	13
2.6	Results . . . . .	13
2.7	Examples and Tutorials . . . . .	13
<b>3</b>	<b>Developers Documentation</b>	<b>15</b>
<b>4</b>	<b>Contact</b>	<b>17</b>



The Field Programmable Gate-Arrays(FPGAs) cluster offers a powerful hardware platform capable of performing realistic simulations of Biological Neural Systems(BNS). However, making use of such powerful platform is not easy and not all users have enough understanding and expertise to exploit such hardware. Having a user friendly software platform to interact with the FPGAs in a much higher abstraction level will allow more researchers to make use of such powerful hardware without having to spend a lot of time actually understanding and programming the platform. Such user interface takes away a certain level of abstraction and makes it intuitive and easy to generate desired biologically inspired neural network with desired neuron models, synapse models, network size, network topology and desired stimulation configuration for the neural network. Here, we discuss web based user interface that we developed to interface with the FPGA cluster that allows users to easily and intuitively create desired network to be run on the FPGA cluster.

---

**Note:** So far only biologically inspired spiking neuron models are implemented.

---

## 1.1 Overview of the platform

The simulator platform is mainly composed of three distinct functional layers - software layer (Web Interface), intermediate layer (Interface Manager(IM)) and hardware layer (FPGA cluster). Software layer is the user interface which acts as front end part of the platform that allows users to interact with the platform.

The web interface is linked to the interface manager (IM) which works as an intermediate node between web interface and the FPGA clusters and manages flow of packets to and from both FPGAs and web interface. IM receives the simulation initialisation files, which contains all the information about the simulation in the xml format, from the web interface generated by the users which are then sent to the appropriate FPGA nodes for simulation. Once the simulation has completed, the results packets are sent by the FPGA clusters to the IM with correct information of simulation and user for correct redirection. IM then handles the incoming packets from the FPGA cluster and send it to the users.

The simplified version of the workflow can be shown as below:

Some of the key features of the web interface are listed below:

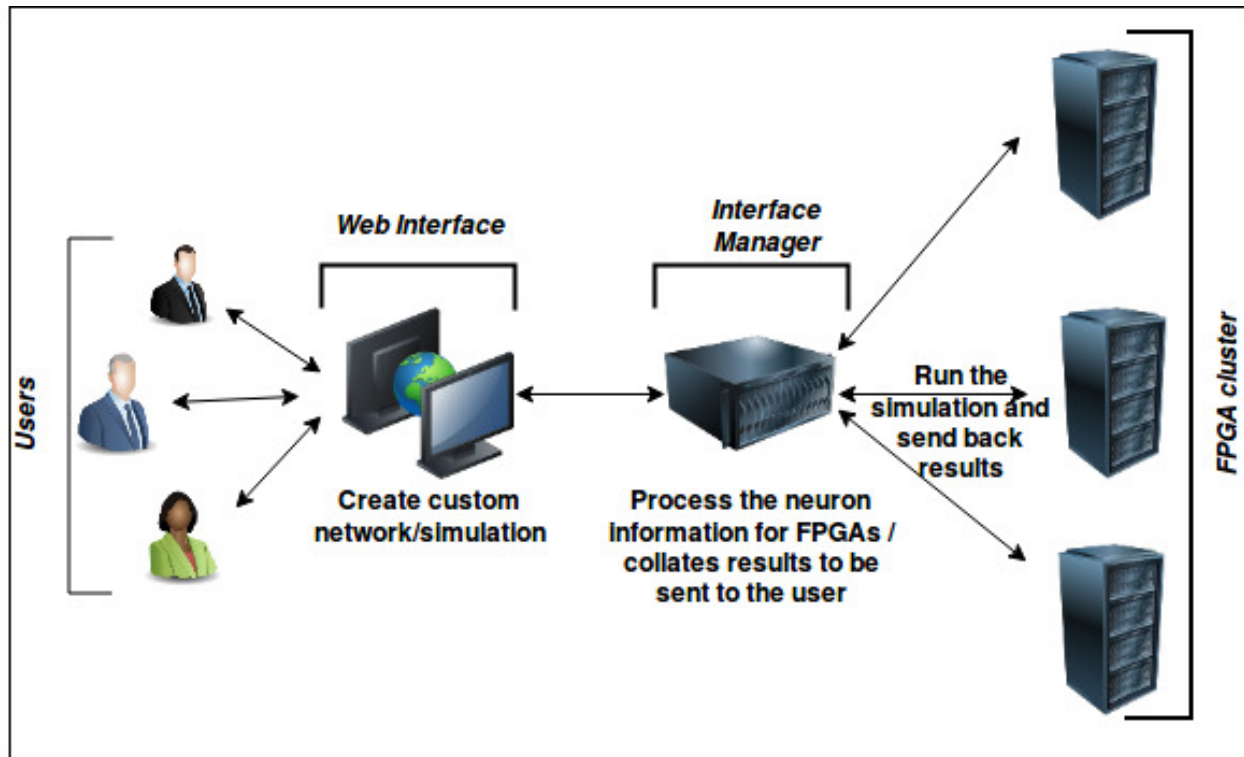


Fig. 1: Figure1: Simplified work flow of the whole process

- Custom network of spiking neurons with parameterisable simulation and models with custom stimulation parameters.
- Parameterisable neuron models from the model library(Current Models: Integrate and Fire, Leaky Integrate and Fire, Izhekevich).
- User specification of neurons with same or different neuron models. If you want same models with different parameters then you can also do that.
- Selection of layered or non-layered network. Layered network would be a feedforward network where the first layer is the input layer and last layer is the output.
- Selection of same or different neuron models on each layer. All the neurons in a single layer will have the same neuron model.
- Topology selection; fully interconnected, random or probabilistic(30% connection at the moment)
- Simulation parameter configuration, such as simulation time, timestep size, watchdog period etc.
- Stimulation parameter configuration such as start and end time and voltage. *[Only step stimulation is implemented at the moment]*
- Visualisation of the spike train obtained after the simulation as raster plot.

## 1.2 Requirements

The following requirements should be met:

- Mozilla or Google chrome browser (Recommended).

- CPU requirements

## 1.3 Disclaimer

- The contents of the platform remains a sole Intellectual property of Nottingham Trent University.

## 1.4 Bug reports

Did you find a bug? Open an issue on [github](#) . Thank you.





---

## Platform Features

---

This part of the documentation is focused to provide a comprehensive details of the interface for users to make it easier to understand capabilities and limitations for the desired simulation. This section covers all the components of the interface and how they all come together during network creation with different possibilities for different types of neuron models, network type, stimulation and capabilities to control how the simulation runs.

The interface employs configuration options presented in the form of questionnaires for easy and straightforward configuration. We will cover different aspects of the interface followed with an exemplary tutorials on how to use the platform.

The simulation workflow is simplified as a flow of events in a sequential order.

### 2.1 User Account

All the users must create an account before proceeding with the simulation for a simple reason that all the configurations and results are to be stored in a corresponding user folder.

Once the account has been successfully created, users can access all the previous simulations or current simulations. The users can also start, stop or abort simulations at any time.

### 2.2 Simulation Parameters

The first step of the network creation is defining the simulation parameters. Here, we define basic parameters for the simulation such as timestamp size, no.of neurons, units for simulation, how long to run the simulation etc.

The user has to choose whether to have all the neurons with the same model or have a combination of different models. In case of different model option, the user has to choose how many of the neurons will use a different model.

## Registration Form

**First Name**

**Last Name**

**Username**

**Password**

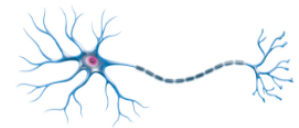
**Confirm Password**

**E-Mail**

SUBMIT



# Neuromorphic Simulator



[Home](#)
[Project](#)
[Publication](#)
[System Builder](#)
[Manage Your Account](#)
[How To Use](#)
[About Us](#)

nikeshlama2018 logged in.
 [Logout](#)

Your user name is: nikeshlama2018.

Your registered email is: testing@testing.com.

In this page the user should be able to access and manage their data (username, password and email) and initialisation files. The user should also be able to delete files, download files and reset the counter for the files.

List of configured simulation which are currently active

[Click here to view your simulation and results files](#)

Simulation Number	Simulation Configured	Simulation Status	Action for Simulation		
3	2017-10-27 13:39:40	Finished	--	--	--
4	2017-10-27 14:04:34	Finished	--	--	--

Parameter	Description
Timestamp	Time bin size
No.of Neurons	Total no of neurons
Are all neurons same model?	Same models for all the neurons or use different models
How many different models?	If different model is chosen
Simulation units	either in milliseconds or seconds
Simulation time	How long to run the simulation in units selected earlier
Watchdog time	Time in ms to wait for a packet

## 2.3 Neuron Models

The simulator covers most widely used neuron models from a simplest Integrate and Fire Model to a rather complex Hodgkin-Huxley(in progress) models. All the models are parametrisable with easy interface to change the parameters. Also, a network can have a combination of these models. For examples, for a layered feedforward network, input layer can be comprised of one model and the next layer be another and so on.

---

**Note:** No two models can be in the layer for a layered network

---

### 2.3.1 Integrate and Fire

Integrate and Fire(*InF*) is one of the simplest and earliest neuron model. The model is given by,

$$I(t) = C_m \frac{dV_m(t)}{dt}$$

which is simply the time derivative of the law of capacitance,  $Q = CV$ . With input current applied, the membrane potential increases with time until it reaches a threshold voltage  $V_{th}$  at which point a spike (delta function) occurs and the voltage is reset back to the resting potential. One of the major shortcoming of this model is the fact that there is no sense of time-dependent memory meaning the voltage is retained if it is below threshold. In real neurons, there is diffusion of ions which reduces the voltage.

There are four parameters for this neuron model:

Parameter	Description
Absolute Refractory period	Inactive period after a spike
Threshold voltage	Voltage at which a neuron emits a spike
Reset voltage	Membrane voltage neuron drop to after a spike
Action potential	Initial membrane potential

### 2.3.2 Leaky Integrate and Fire

The memory problem present in Integrate and Fire model is mitigated in Leaky Integrate and Fire model by introducing a 'leakage' of membrane potential, reflecting diffusion of ions that takes place through the membrane to return the membrane potential to a resting potential value. The model can be described with,

$$I(t) = \frac{V_m(t)}{R_m} + C_m \frac{dV_m(t)}{dt}$$

where  $R_m$  is the membrane resistance. The leakage term  $\frac{V_m(t)}{R_m}$  is introduced which governs the leakage of ions resulting the voltage to return back to resting potential when no input current is applied.

Parameter	Description
Absolute Refractory period	Inactive period after a spike
Threshold voltage	Voltage at which a neuron emits a spike
Reset voltage	Membrane voltage neuron drop to after a spike
Capacitance	Membrane capacitance
Resistance	Membrane resistance which governs current leakage
Action potential	Initial membrane potential

### 2.3.3 Izhikevich

Integrate and fire and Leaky integrate and fire models are simplest neurons models which replicates some dynamics of the neurons but they are not biologically plausible models. For a more biologically plausible models many neurons models are proposed but they get complex resulting in large computational requirements which again is not very useful to model a network of such neurons. Izhikevich neuron model falls in the middle, encompassing biological plausibility and simple and efficient enough to model computationally.

Izhikevich model reduces a complex biologically plausible Hodgkin and Huxley model to a two-dimensional(2-D) system of ordinary differential equations of the form,

$$\begin{aligned}v' &= 0.04v^2 + 5v + 140 - u + I \\u' &= a(bv - u)\end{aligned}$$

with auxiliary after-spike resetting

$$\text{if } v \geq V_{th}, \text{ then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases}$$

Here,  $v$  represents the membrane potential of the neuron and  $u$  represents a membrane recovery variable, which accounts of ionic (in)activation. When the membrane potential reaches a threshold,  $V_{th}$ , the membrane voltage is reset back to the resting potential. Please refer to [Izhikevich paper](#) for detailed information.

The parameters that are available for users to configure from the interface are:

Parameter	Description
a	Describes time scale of the recovery variable $u$
b	Describes sensitivity of the recovery variable $u$
c	Describes after-spike reset value of $v$
d	Describes after-spike reset of the recovery variable $u$
Threshold voltage	Threshold potential for spike
Initial action potential	Starting membrane potential
Initial membrane recovery	Initial membrane recovery value
Action potential	Initial action potential
Membrane recovery	Membrane recovery value( $u$ )

### 2.3.4 Custom Models

—Under Development—

## 2.4 Topology

In terms of the topology, we are looking into an architecture of the network and interconnections among neurons. The architecture of the simulated neuronal network can be divided as layered and non layered. Layered networks that can be implemented with this platform is a feedforward network. No recurrency is possible at the moment for a layered network. Different neuron models can be implemented for each layer, for eg. input layer can be of Leaky Integrate and Fire neurons and layer 2 can be Izhikevich or any combination of the available neuron models.

Within layered and non-layered networks, how the neurons are connected can also be altered. Manual connection is not a suitable choice since it is labourous and takes very long time to configure one by one. Hence, three different options for automatic interconnections are presented. Please see [Interconnections](#) for more details about the options.

### 2.4.1 Architecture

The first step of the network creation is specifying whether to create a layered or non layered network.

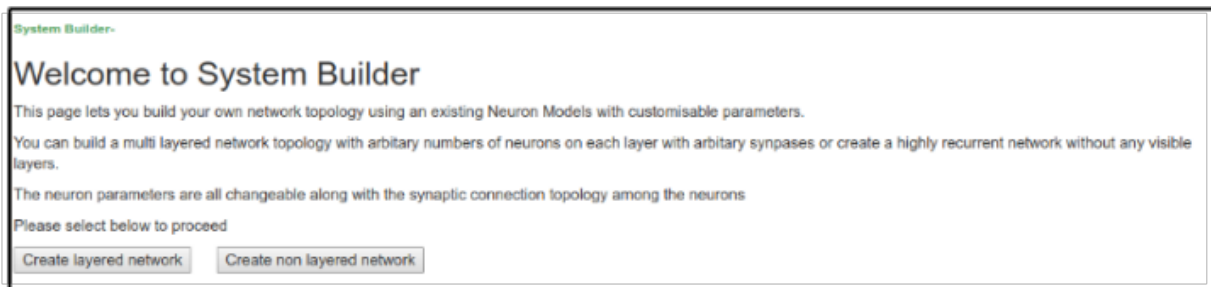


Fig. 1: Figure: First step of network creation

### Layered

The figure below shows a typical graphical representation of what a layered network is. The first layer in the layered network is the input layer where stimulus are applied and the last layer is the readout layer. The interconnections among neurons between the layers is described in [Interconnections](#).

Number of layers need to be specified. Neuron are numbered in an increasing order from the first layer to the last. The menu guides to select number of neurons in each layer along with neuron models for each layer if different neuron model option is selected.

### Non-layered

In contrast to layered network, non layered network has no apparent layered structured. Any two neurons can be connected to each other. The user has full control over which neurons can be assigned as input neurons and output neurons. Recurrency is also possible in this type of non layered topology. The topology can be illustrated as shown in the figure below.

### 2.4.2 Interconnections

In addition to layer topology, interconnections among different neurons is also configurable with the web interface.

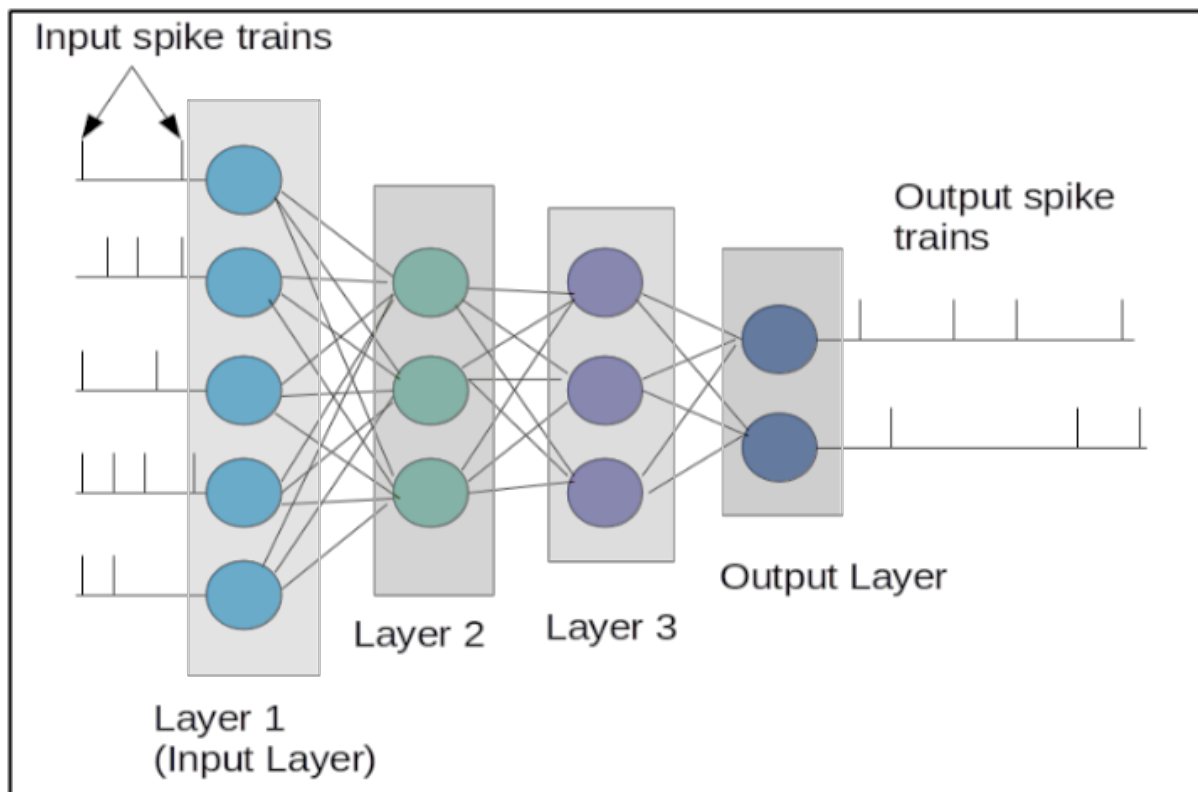


Fig. 2: Figure: Graphical illustration of the layered network

System Builder->Simulation Parameters

## You have selected to build a layered network

This page lets you build your own network topology using an existing Neuron Models with customisable parameters.

**Please specify the number of layers**

Number of Layers :

Are all neurons using the same model:

Timestamp(in ms):  (Fixed value)

Simulation units:

Simulation time:

Watchdog (ms):

System Builder->Simulation Parameters->Layer Configuration

## You have selected to build a layered network

You have selected to create a network with 5 layers with different neurons

Number of Neurons in layer 1 ( <b>Input layer</b> ):	<input type="text" value="5"/>	Neuron model: <input type="button" value="Integrate and fire"/>
Number of Neurons in layer 2 :	<input type="text" value="10"/>	Neuron model: <input type="button" value="Leaky integrate and fire"/>
Number of Neurons in layer 3 :	<input type="text" value="10"/>	Neuron model: <input type="button" value="Izhikevich"/>
Number of Neurons in layer 4 :	<input type="text" value="5"/>	Neuron model: <input type="button" value="Integrate and fire"/>
Number of Neurons in layer 5 ( <b>Output layer</b> ):	<input type="text" value="3"/>	Neuron model: <input type="button" value="Integrate and fire"/> <input type="button" value="Integrate and fire"/> <input type="button" value="Leaky integrate and fire"/> <input type="button" value="Izhikevich"/>

<--Lets assign neuron parameters

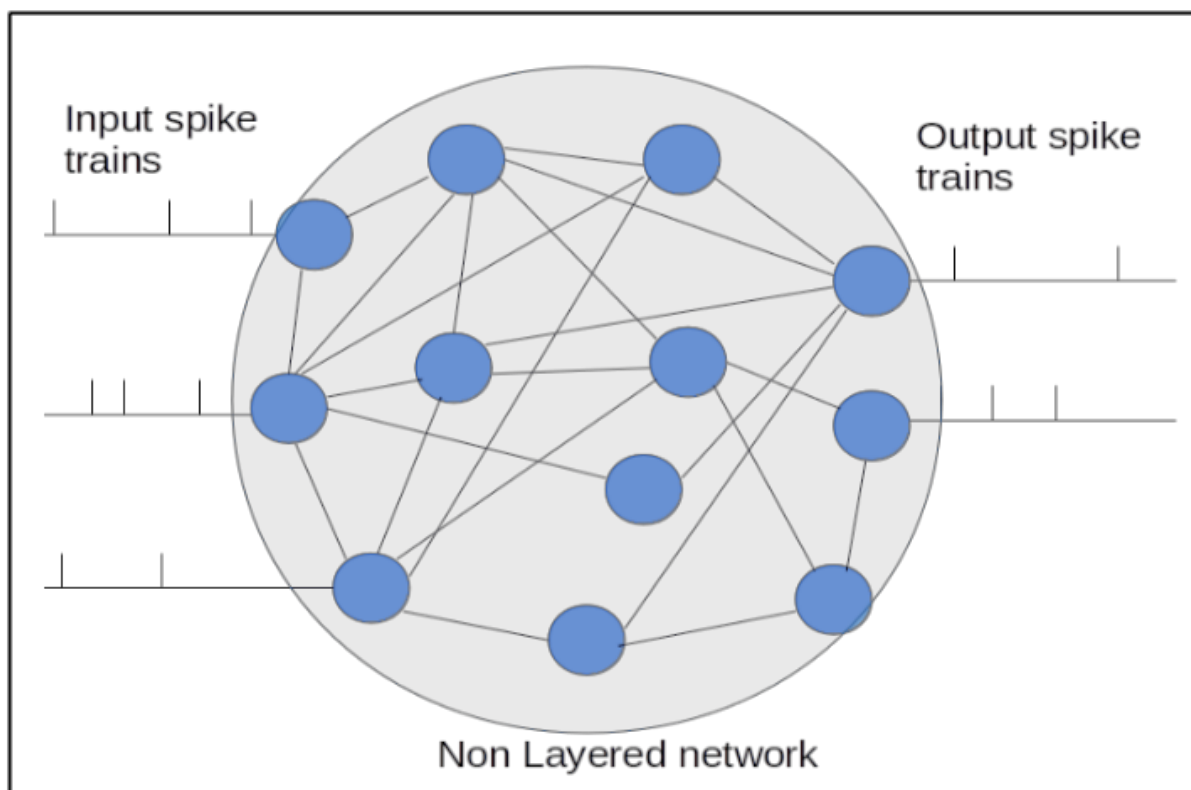


Fig. 3: Figure: Graphical illustration of the non-layered network



**Fully interconnected network**

**Randomly connected network**

**Probabilistic network**

## **2.5 Stimulation**

External sensory stimuli are transduced into electrical signals which results a specific network response. Hence, ability to apply stimulation is very important for any kind of spiking neural network simulators. Here, we present available options for the stimulation and plans for further stimulation options.

### **2.5.1 Step current**

### **2.5.2 Frequency based**

### **2.5.3 Biphasic stimulation**

## **2.6 Results**

Deals with results collation and further analysis

## **2.7 Examples and Tutorials**

This section will shows the features of the web app. —to be completed——



## CHAPTER 3

---

### Developers Documentation

---

Developers page—in progress. . . .

This section will include details on the codebase.



## CHAPTER 4

---

### Contact

---

If any questions please contact Nikesh Lama @ [nikesh.lama2015@my.ntu.ac.uk](mailto:nikesh.lama2015@my.ntu.ac.uk)