Example Set-ups: Synchrony and Learning

Synchrony (Lab 5) and Learning (Lab 9) data

Synchrony Lab set-up and software
Learning Lab set-up and software

Microcircuit
Corrected microcircuit

We use parts of this microcircuit to implement lab experiments.

For the Synchrony Lab, have several requirements:
- excitatory neurons inactive
- constant current to each interneuron
- interneurons set to inhibit globally (or just themselves)

Synchrony Lab Set-up & Biases

Synchrony Lab configured microcircuit

We turn off the excitatory neurons by setting their excitatory synaptic input strengths to zero ($V_{magamp} = 0.000V$).

We only use the interneurons, one of their four fast excitatory inputs, and their inhibitory synapse.

We drive each neuron with constant current by driving a fast excitatory synapse faster than its pulse width, which we set to be as long as possible ($V_{leakamp} = 0.000V$). The interneurons’ excitatory synaptic input strength controls the constant current ($V_{magampint} = 0.510V$).

We set the excitatory synapse diffusion to zero ($V_{re} = 0.750V$), so each neuron receives its own separate constant current.

We set the inhibitory diffusor to spread current globally (synchrony, $V_{ri} = 1.750V$) or to restrict it to the neuron that generates it (control, $V_{ri} = 0.750V$).
Synchrony Lab Software

The software has several roles:
- Initialize the program and board, including biases, scanning to neuron, and sending membrane potential
- Transmitting spike data to the chip continuously to maintain the constant current to each neuron
- Receiving spike data from the chip continuously as well as saving and plotting the data

```
data out example
for y=0:15
for x=0:15
excite fast synapse (y,x);
wait dt;
end
end
transmit;
```
Synchrony Lab Software components

The software has four primary parts:
- Initialize, which creates variables, starts the USB link, and sets the chip's biases
- Main loop, which manages the data flow and plotting
- Transmit thread, which continuously sends data to the chip
- Receive thread, which continuously acquires data from the chip

Learning Lab Neuron Set-up & Biases

Learning Lab configured microcircuit

We turn off the inhibitory neurons by setting their excitatory synaptic input strengths to zero ($V_{\text{magampaint}} = 0.000V$).

We drive each neuron with constant current by driving a fast excitatory synapse with the same period as its pulse width, which we set to be 0.25ms ($V_{\text{leakampa}} = 0.300V$). When we want a noisy current, we drive each neuron at the same rate but randomly drop spikes. We increase the neurons' excitatory synaptic input strength ($V_{\text{magampa}}$) such that its mean current is constant even when we drop spikes.

We set the excitatory synapse diffusion to zero ($V_{\text{rc}} = 0.750V$), so each neuron receives its own independent current.

We set the neurons' M-type potassium current to limit each one to one spike per theta cycle.

We vary the neurons' leak conductance ($V_{\text{leak}}$) to emulate an 8.75Hz theta inhibition.
Learning Lab Board Set-up

Learning Lab look-up table

The Learning Lab uses a RAM Chip to implement a look-up table.

Connections were generated randomly with each neuron sending and receiving (most) connections from other neurons within a distance of 5 nodes away.

When an neuron spikes the CPLD looks in the RAM to determine which 21 STDP synapses (on other neurons) it should stimulate.
Learning Lab Software

The Learning Lab requires 4 programs.

- **writesram.exe** loads the connections into the look-up table
  a pregenerated file determines each connection

- **ltd.exe** depresses all synapses
  it uses the scanner to shift through all 21,504 synapses, writing their states

- **readsyn.exe** downloads the synapses’ states
  it uses the scanner to read the synapses' states

- **memory.exe** is similar to the Synchrony Lab’s software; however it performs some additional functions:
  - its initialization block turns on the look-up table
  - it keeps track of time to implement the theta inhibition
    every 2ms it modifies the value of $G_{\text{leak}}$, varying it sinusoidally
  - it randomly decides to stimulate each synapse every 0.25ms, generating a noisy current for each neuron
  - it selects a fraction of the 100 neurons stimulated when recalling a pattern