TECHNISCHE
UNIVERSITÄT
DRESDEN

Faculty of Electrical and Computer Engineering
Chair of Highly-Parallel VLSI System and Neuro-Microelectronics

SPIKING NEURAL NETWORKS

OBERSEMINAR INFORMATIONSTECHNIK

eingereicht von:	Wu, Binyi
Matrikel-Nr.:	4571474
Studiengang:	Electrotechnik
Betreuer:	Dipl.-Ing. Jörg Schreiter
Abgabedatum:	21.03.2017

1. Introduction

In my work, I created a small neural network with VHDL, then generate spike and show by simulation how the neural network reacts to it. My neural network has only 3 integrate-and-fire(I&F) neurons with adaptive Synapses. It is easier to identify the changes in neurons. In the next section, I will detail my work.

2. Spiking Neural Network

2.1 Adaptive Synapse

The key properties of adaptive synapse are the ability to exhibit short-term plasticity and long-term- plasticity. The adaptive synapse in my work is base on the following circuit **[1]** showing in Fig.1, which is proposed by Indiveri, G., Chicca, E., and Douglas, R. .

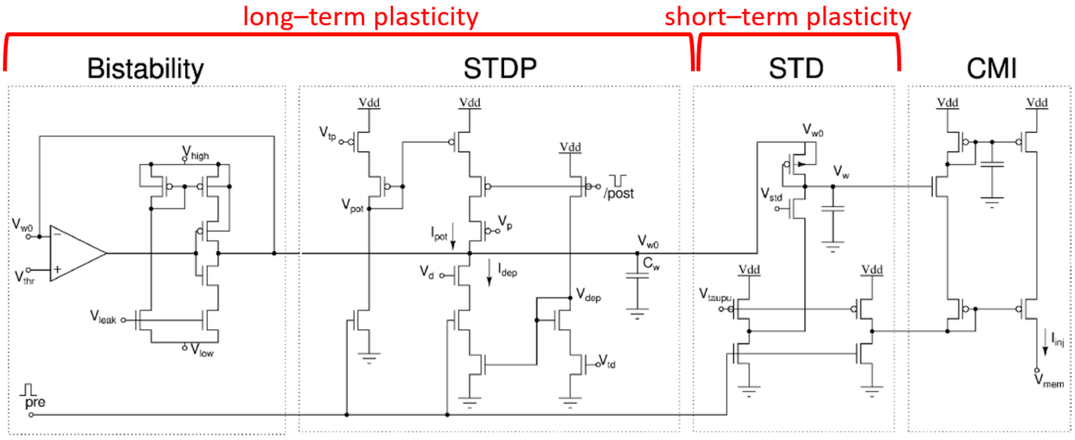


Fig.1. Adaptive Synapse Circuit

2.1.1 Long-Term Plasticity

The long-term plasticity (long-term memory) produces a long-term changes in synaptic weight according to the timing of pre-spike and post-spike. In other words, long-term plasticity is spiking-timing dependent. In my work, If the pre-spike arrives before the post-spike within Δt (200us), the synaptic weight increase Δw. If the pre-spike arrives after the post-spike within Δt (200us), the synaptic weight decrease Δw. If the pre-spike and post arrive simultaneously, there is no weight modification. The range of weight is between 0 and 1. Δw a fixed step in my work and is summarized by following equations (α is constant):

$$\Delta\mathrm{w} = \begin{cases} (1-\mathrm{w}) \times \dfrac{\partial}{\partial t}(e^{\frac{t}{\tau}})\Big|_{t=0} \times \alpha, & \Delta\mathrm{t} > 0 \\ (-\mathrm{w}) \times \dfrac{\partial}{\partial t}(e^{\frac{t}{\tau}})\Big|_{t=0} \times \alpha, & \Delta\mathrm{t} < 0 \end{cases}$$

$$\Delta\mathrm{w} = \begin{cases} (1-\mathrm{w}) \times \dfrac{\alpha}{\tau}, & \Delta\mathrm{t} > 0 \\ (-\mathrm{w}) \times \dfrac{\alpha}{\tau}, & \Delta\mathrm{t} < 0 \end{cases}$$

After the modification, the synaptic weight is summarized by following equations:

$$W = \begin{cases} w + (1 - w) \times \frac{\alpha}{\tau}, & \Delta t > 0 \\ w + (-w) \times \frac{\alpha}{\tau}, & \Delta t < 0 \end{cases}$$

2.1.2 Short-Term Plasticity

The short-term plasticity produces dynamic modulation of the synaptic weight by the pre-spike. In my model, the synaptic weight is always decreased $\Delta w'$ when every pre-spike comes and after a while the synaptic weight will restore. We also call short-term-plasticity a short-term-memory.

$$\Delta w' = (-w) \times \frac{\alpha}{3\tau}$$

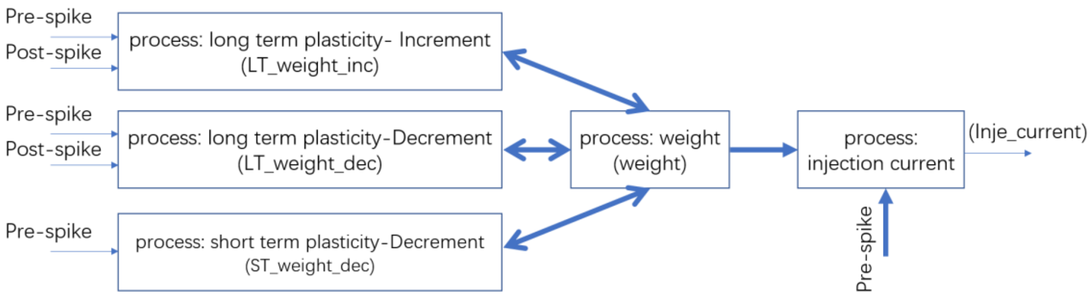
2.1.3 Postsynaptic Current

In my work, I set the the postsynaptic current, which injected into the neuron, equal to the synaptic weight.

$$I_{inje} = W$$

2.1.4 Architecture of adaptive Synapse (VHDL)

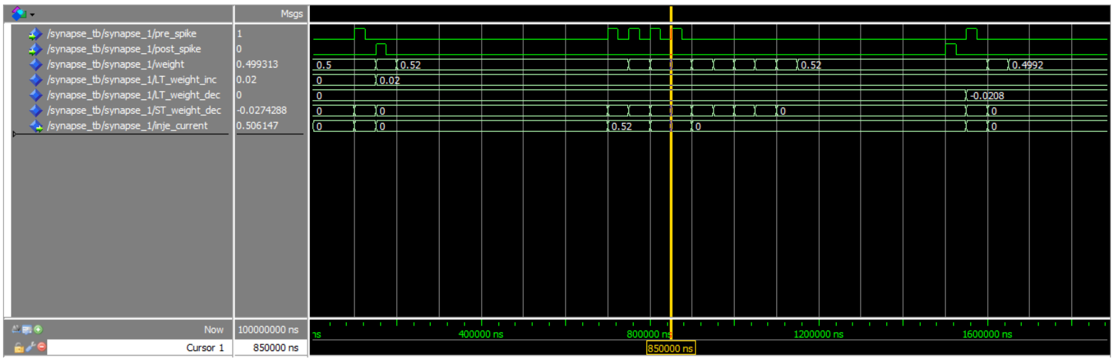
Process:long-term-plasticity-Increment produces a positive signal “LT_weight_inc”.
Process:long-term-plasticity-Decrement produces a negative signal “LT_weight_dec”.
Process:short-term-plasticity-Decrement produces a negative signal “LT_weight_dec”.
These three signal will be send to the process:weight , and then the variable “weight” will be renewed. In addition to this, the signal “weight” will also feedback to these three processes.
Process:injection-current will produce signal ”inje_current” according to “weight” when a pre-spike comes.


Pre-spike
Post-spike
process: long term plasticity- Increment (LT_weight_inc)
Pre-spike
Post-spike
process: long term plasticity-Decrement (LT_weight_dec)
Pre-spike
process: short term plasticity-Decrement (ST_weight_dec)
process: weight (weight)
process: injection current
(Inje_current)
Pre-spike

2.1.5 Simulation

weight = 0.5 (Initial value)
LT_weight_inc : Increment of weight in long-term plasticity
LT_weight_dec : Decrement of weight in long-term plasticity
ST_weight_dec : Decrement of weight in short-term plasticity
Inje_current : Current injected into neuron

When the pre-spike is arrived before post-spike within Δt, the “weight” will increase. When the pre-spike is arrived after post-spike within Δt, the “weight” will decrease. When pre-spike comes, “weight” will decrease and after a while will restore. From the waveform you can see that.



	Msgs
/synapse_tb/synapse_1/pre_spike	1
/synapse_tb/synapse_1/post_spike	0
/synapse_tb/synapse_1/weight	0.499313
/synapse_tb/synapse_1/LT_weight_inc	0.02
/synapse_tb/synapse_1/LT_weight_dec	0
/synapse_tb/synapse_1/ST_weight_dec	-0.0274288
/synapse_tb/synapse_1/inje_current	0.506147
Now	100000000 ns
Cursor 1	850000 ns

2.2 Neuron

In my work, a neuron has a simple address decoder, two adaptive synapses, and a spike generator. Fig.2 shows the structure of the neuron. This structure is based on the LIF (Leaky Integrate and Fire) neuron **[2]** (Fig. 3), which is proposed by Runchun M.Wang, Tara J. Hamilton, Jonathan C. Tapson and André van Schaik.

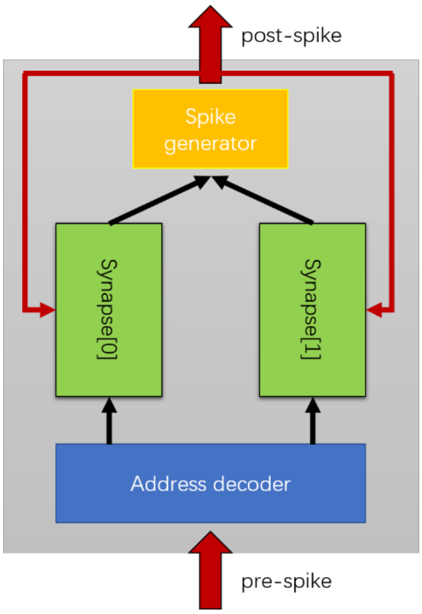


Fig.2 Structure of neuron

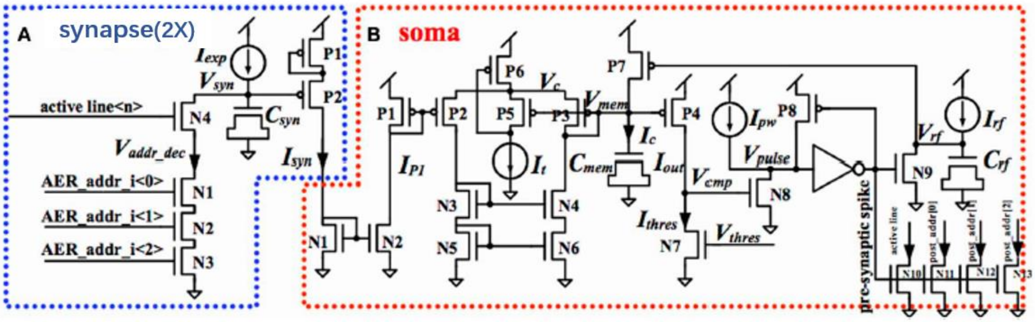


Fig. 3 Circuit diagram of the LIF neuron

2.2.1 Address Decoder

All the pre-spike signal will be put on the pre-spike-bus. Every clock address decoder will obtain the address signal from the pre-spike-bus, then execute the AND operation with his own address to get whether its pre-spike exists on the pre-spike-bus.

2.2.2 Synapse (Adaptive Synapse)

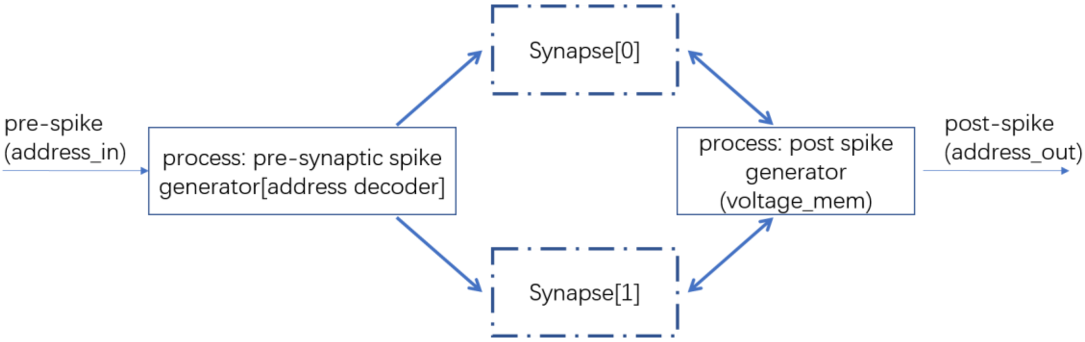
The synapse showed in the preceding figures is the adaptive synapse detailed in section 2.1. Different synapses have different address (a unique address). When the pre-spike comes, the address decoder will decode the address and send pre-spike to the corresponding synapse. Then synapse will generate injection current and send to spike generator.

2.2.3 Spike Generator

The spike generator is based on the integrate-and-fire(I&F) circuit. The injection current from synapse will injected into the spike generator (neuron soma) for integration, which stores in a capacitor C_{mem}. When the voltage of the capacitor V_{mem} exceeds a certain threshold, the spike generator will produce a post-spike and the output address. This post-spike will also feedback to synapses. Synapses will use this feedback signal to change the synaptic weight.

2.2.4 Architecture of neuron (VHDL)

Synapse[0] and Synapse[1] are the instances of the adaptive synapse, which is detailed in the previous section 2.1. The output of Synapse will charge the capacitor. The capacitor voltage “voltage_mem” becomes larger. When “voltage_mem” exceeds a certain threshold, it will produce a output “post_spike”. This output will also be feedback to the synapses.

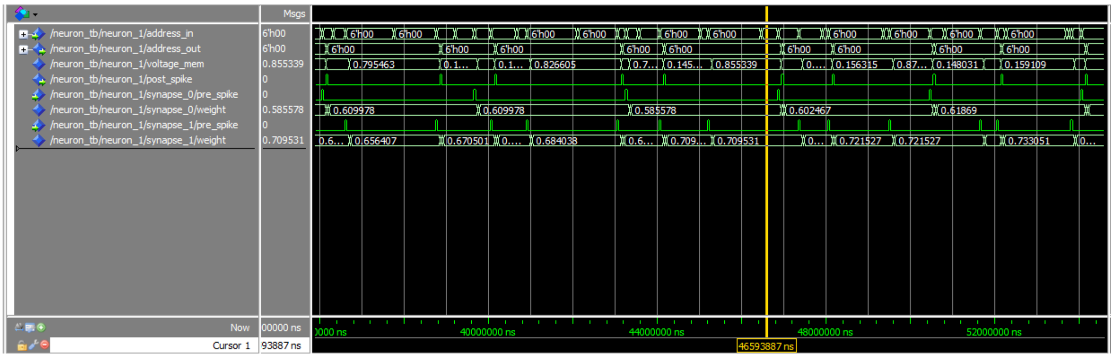

pre-spike
(address_in)
process: pre-synaptic spike generator[address decoder]
Synapse[0]
Synapse[1]
process: post spike generator (voltage_mem)
post-spike
(address_out)

2.2.4 Simulation

address_in : the input address (pre-spike)

address_out : the output address (post-spike)

voltage_mem: the voltage of the capacitor V_{mem}



	Msgs
/neuron_tb/neuron_1/address_in	6'h00
/neuron_tb/neuron_1/address_out	6'h00
/neuron_tb/neuron_1/voltage_mem	0.855339
/neuron_tb/neuron_1/post_spike	0
/neuron_tb/neuron_1/synapse_0/pre_spike	0
/neuron_tb/neuron_1/synapse_0/weight	0.585578
/neuron_tb/neuron_1/synapse_1/pre_spike	0
/neuron_tb/neuron_1/synapse_1/weight	0.709531
Now	00000 ns
Cursor 1	93887 ns

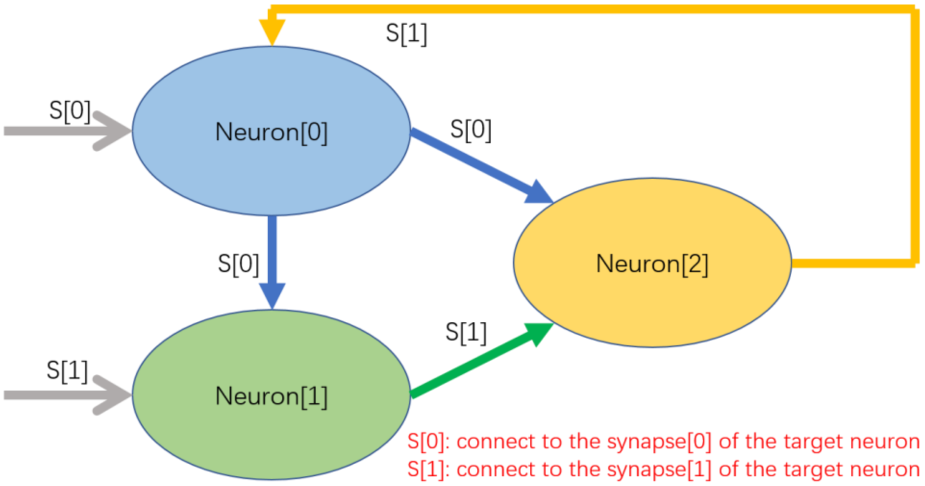
2.3 Spiking Neuron Network

In my work, I realize a neuron network with only 3 neurons. It is easier to identify and analyze the status of neurons. In addition to neurons, there is a simple address encoder.

2.3.1 Address Encoder

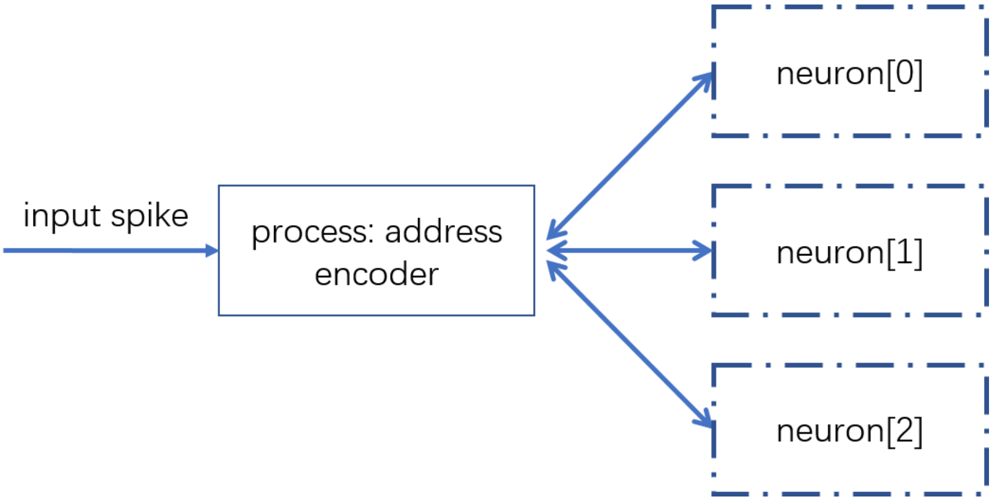
The address encoder in my neuron network is very simple. All the post-spikes will be send to the address encoder, then address encoder will execute OR operation with all the post-spike signals and send out to the pre-spike-bus.

2.3.2 Structure of Neuron Network


S[1]
S[0]
Neuron[0]
S[0]
S[0]
Neuron[2]
S[1]
S[1]
Neuron[1]
S[0]: connect to the synapse[0] of the target neuron
S[1]: connect to the synapse[1] of the target neuron

2.3.3 Architecture of neuron network (VHDL)

The neurons are the instance of the previous section 2.2. All the neurons get input signal (pre-spike and input address) from the BUS and send output signal (post-spike and output address) to the process:address_encoder. process:address_encoder will synthesize a new signal and send on the BUS.

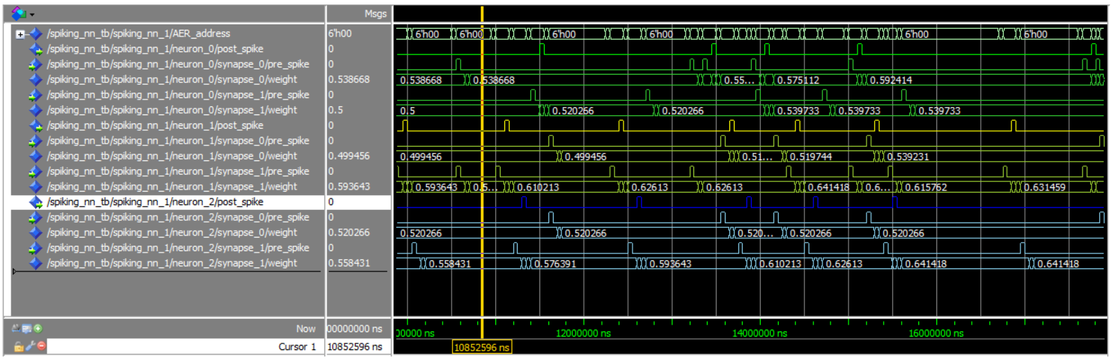


2.3.4 Simulation

The green line represents the first neuron.

The yellow line represents the second neuron

The blue line represents the third neuron



	Msgs
/spiking_nn_tb/spiking_nn_1/AER_address	6'h00
/spiking_nn_tb/spiking_nn_1/neuron_0/post_spike	0
/spiking_nn_tb/spiking_nn_1/neuron_0/synapse_0/pre_spike	0
/spiking_nn_tb/spiking_nn_1/neuron_0/synapse_0/weight	0.538668
/spiking_nn_tb/spiking_nn_1/neuron_0/synapse_1/pre_spike	0
/spiking_nn_tb/spiking_nn_1/neuron_0/synapse_1/weight	0.5
/spiking_nn_tb/spiking_nn_1/neuron_1/post_spike	0
/spiking_nn_tb/spiking_nn_1/neuron_1/synapse_0/pre_spike	0
/spiking_nn_tb/spiking_nn_1/neuron_1/synapse_0/weight	0.499456
/spiking_nn_tb/spiking_nn_1/neuron_1/synapse_1/pre_spike	0
/spiking_nn_tb/spiking_nn_1/neuron_1/synapse_1/weight	0.593643
/spiking_nn_tb/spiking_nn_1/neuron_2/post_spike	0
/spiking_nn_tb/spiking_nn_1/neuron_2/synapse_0/pre_spike	0
/spiking_nn_tb/spiking_nn_1/neuron_2/synapse_0/weight	0.520266
/spiking_nn_tb/spiking_nn_1/neuron_2/synapse_1/pre_spike	0
/spiking_nn_tb/spiking_nn_1/neuron_2/synapse_1/weight	0.558431
Now	00000000 ns
Cursor 1	10852596 ns

3. Reference

[1] Indiveri, G., Chicca, E., and Douglas, R. (2006). A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. IEEE Trans. Neural Netw. 17, 211–221.

[2] Runchun M.Wang, Tara J. Hamilton, Jonathan C. Tapson and André van Schaik. 2014. "A mixed-signal implementation of a polychronous spiking neural network with delay adaptation". Frontiers in Neuroscience. doi: 10.3389/fnins.2014.00051