

Outlines

- **Introduction on reviewed paper**
 - Liquid state machine (LSM)
 - Spike timing-dependent plasticity (STDP)
- **Liquid State Machine Architecture**
 - Overall architecture
 - Reservoir Unit
 - Training Unit
- **Digital spiking neuron implementation**
 - Reservoir neuron
 - Unsupervised STDP learning engine
 - Output neuron
 - Supervised STDP learning engine
- **Improvement over the reviewed work**
 - Hardware optimization
 - Resource overhead reduction
- **Benchmark and Result**
 - Training Setup and Benchmark
 - Result and Discussion

Introduction

- Reviewed Paper:
 - Energy-efficient FPGA based SNN accelerators with supervised & unsupervised STDP^[1]

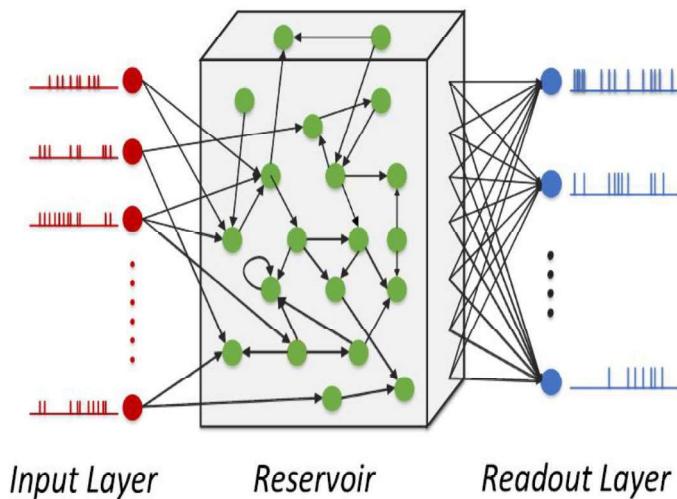


Figure: Liquid State Machine ^[1]

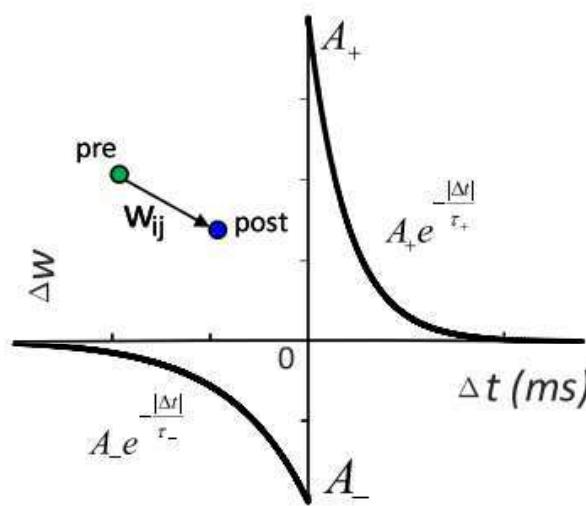
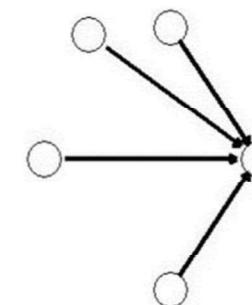


Figure: Standard STDP curve ^[1]

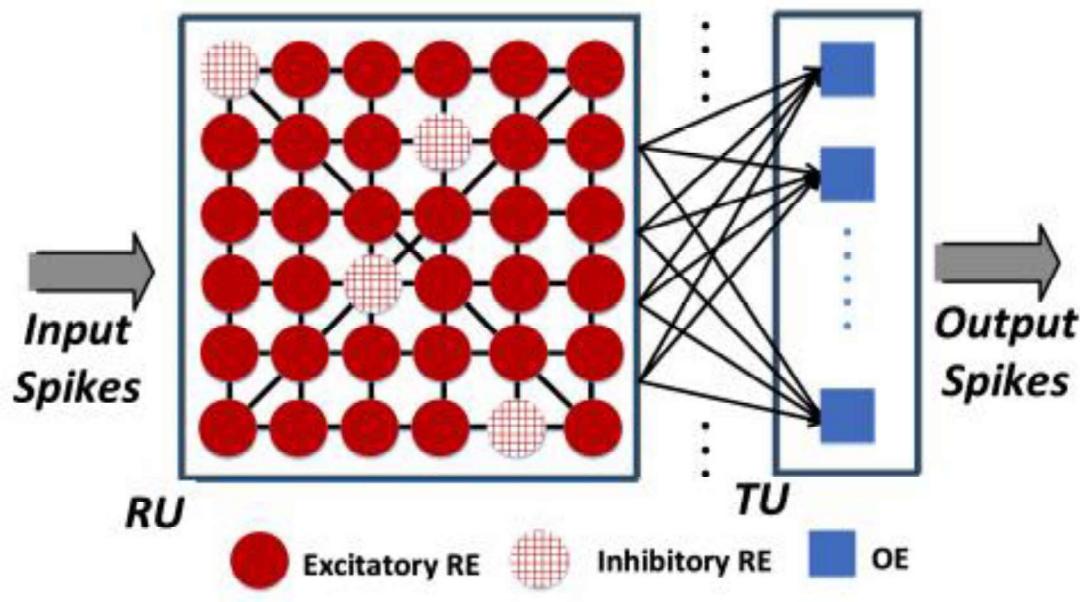
$$\Delta w^+ = A_+(w) \cdot e^{-\frac{|\Delta t|}{\tau_+}} \quad \text{if } \Delta t > 0,$$
$$\Delta w^- = A_-(w) \cdot e^{-\frac{|\Delta t|}{\tau_-}} \quad \text{if } \Delta t < 0,$$



MakeAGIF.com

[1] Liu, Yu, Sai Sourabh Yenamachintala, and Peng Li. "Energy-efficient FPGA spiking neural accelerators with supervised and unsupervised spike-timing-dependent-plasticity." *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 15.3 (2019): 1-19.

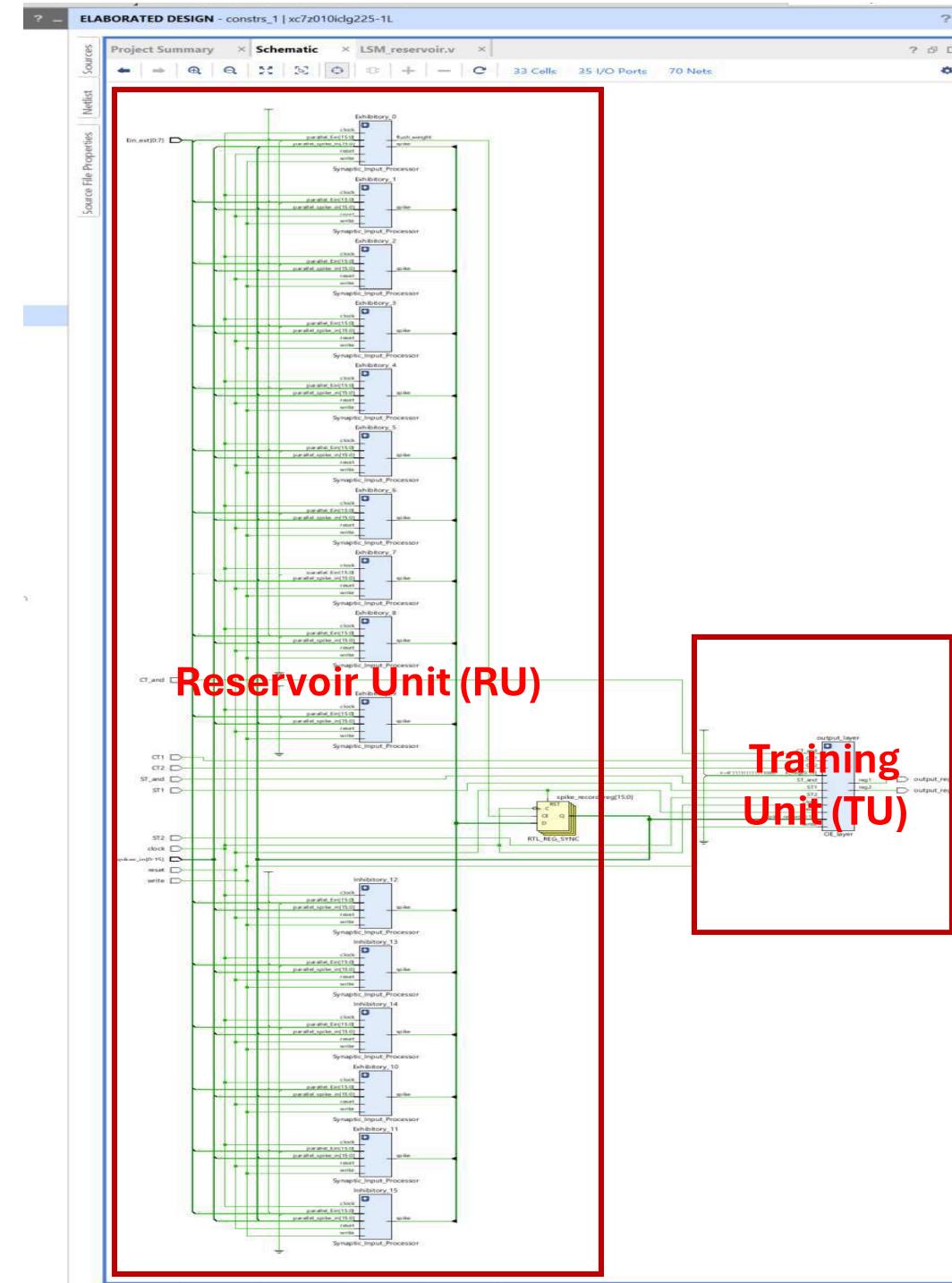
Overall LSM Architecture



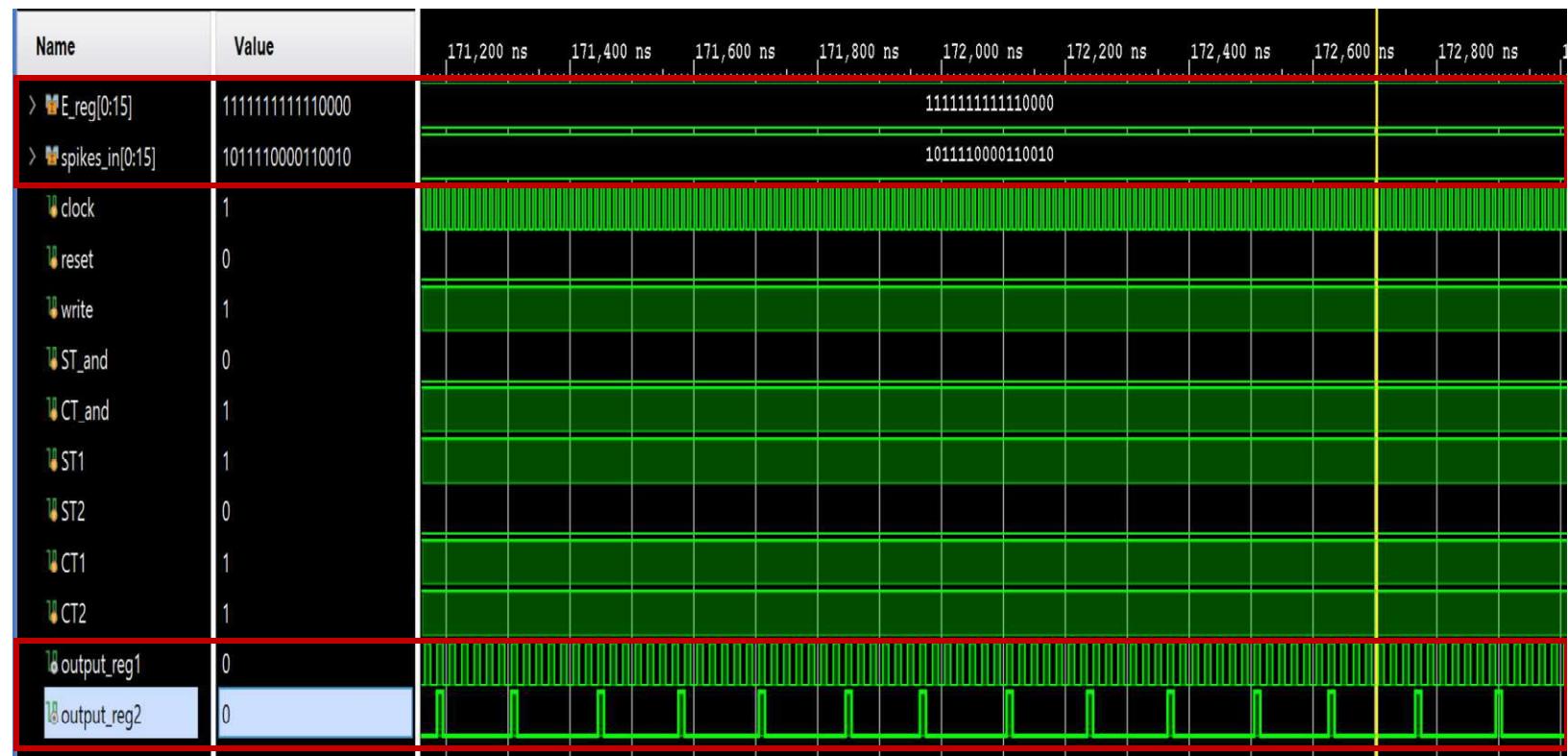
Operation mechanism

- The input spike enters the reservoir unit through a predefined crossbar interface.
- Output of the reservoir unit (RU) is sent to the output training unit (TU) and spike parts are fed back to the reservoir element as a recurrent response.

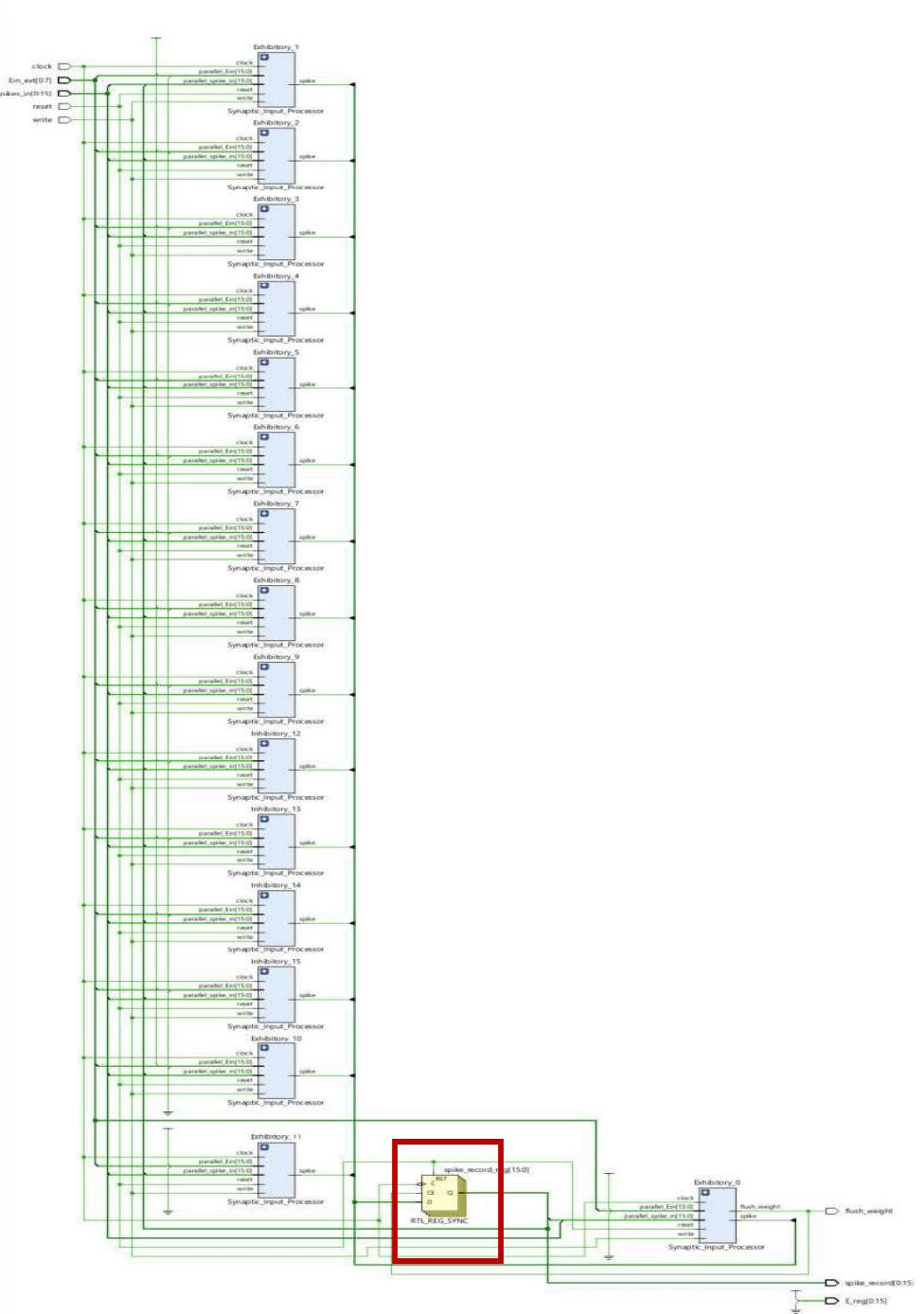
LSM Schematic



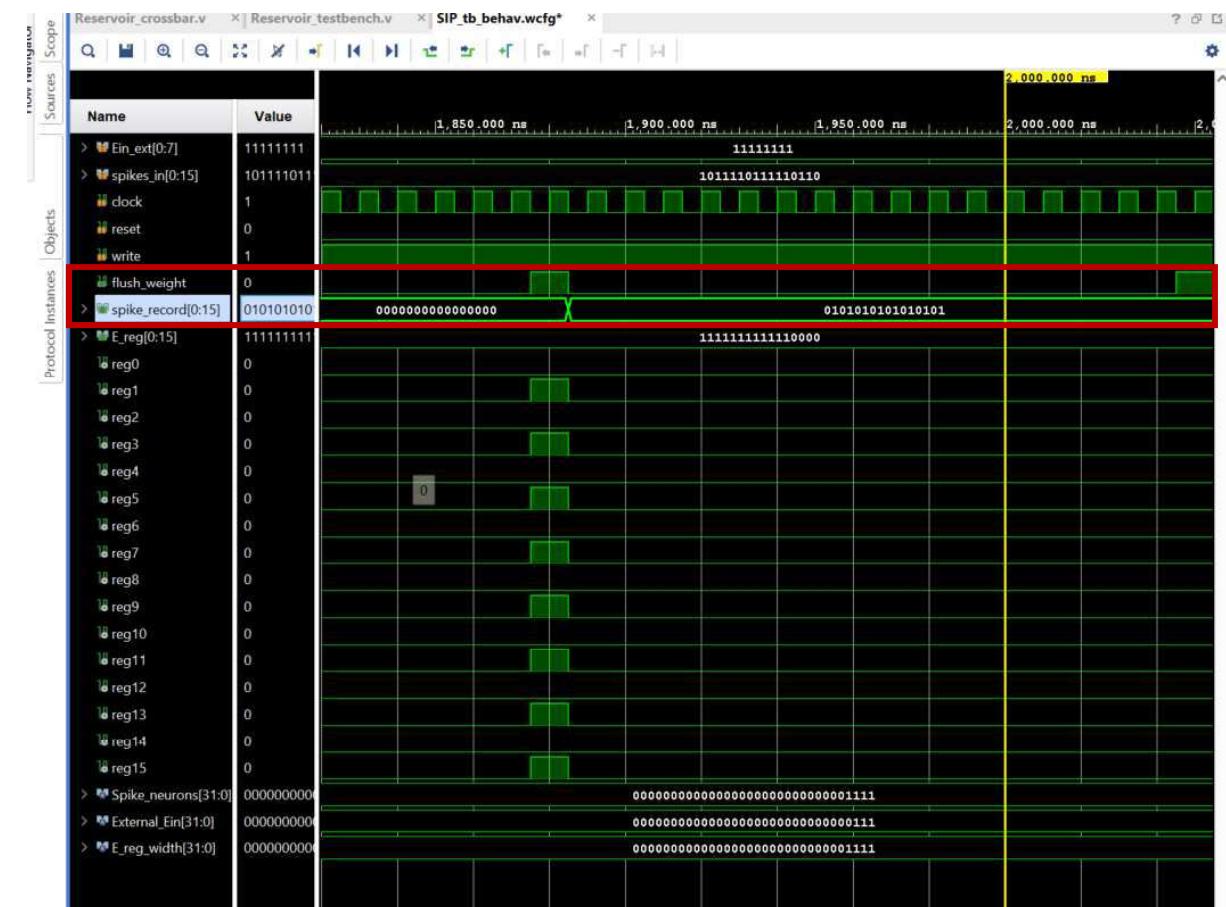
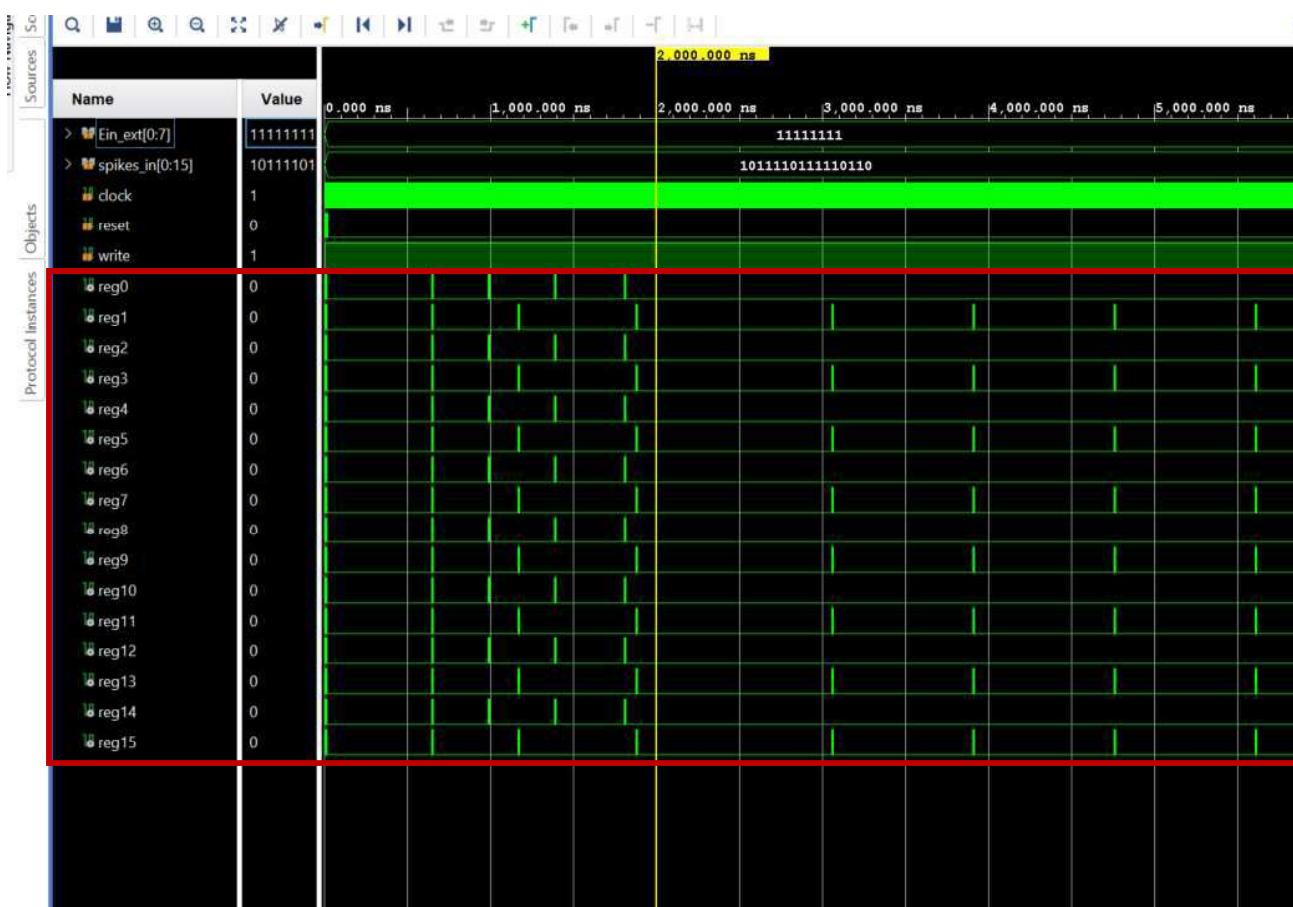
LSM Simulation



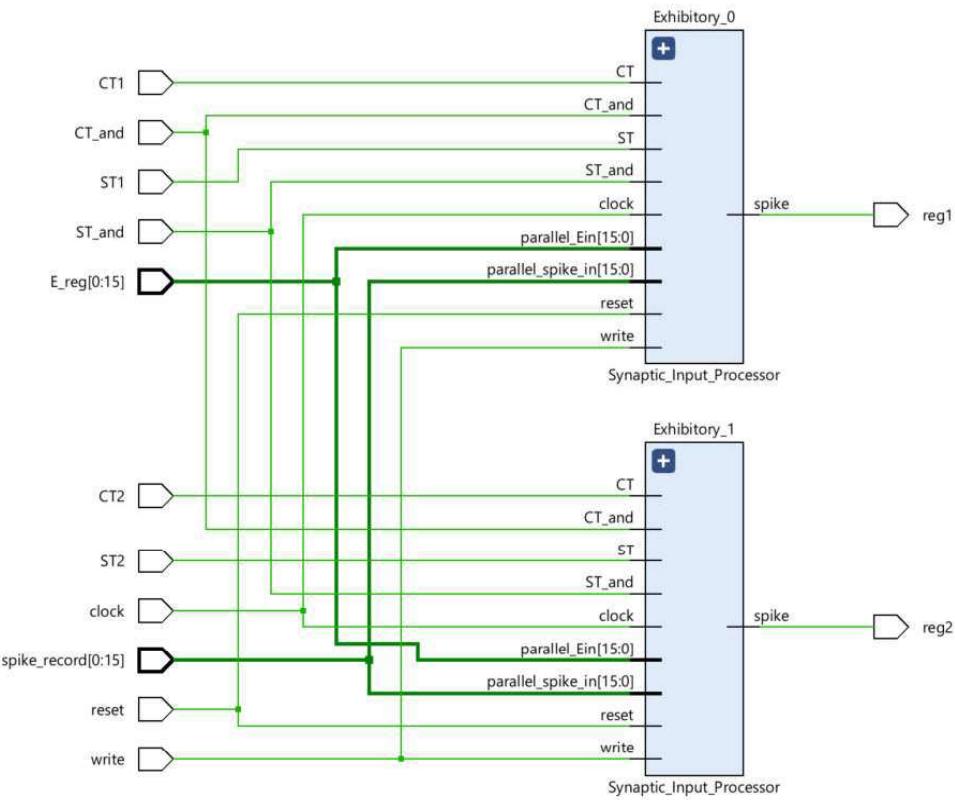
Reservoir Unit Schematic



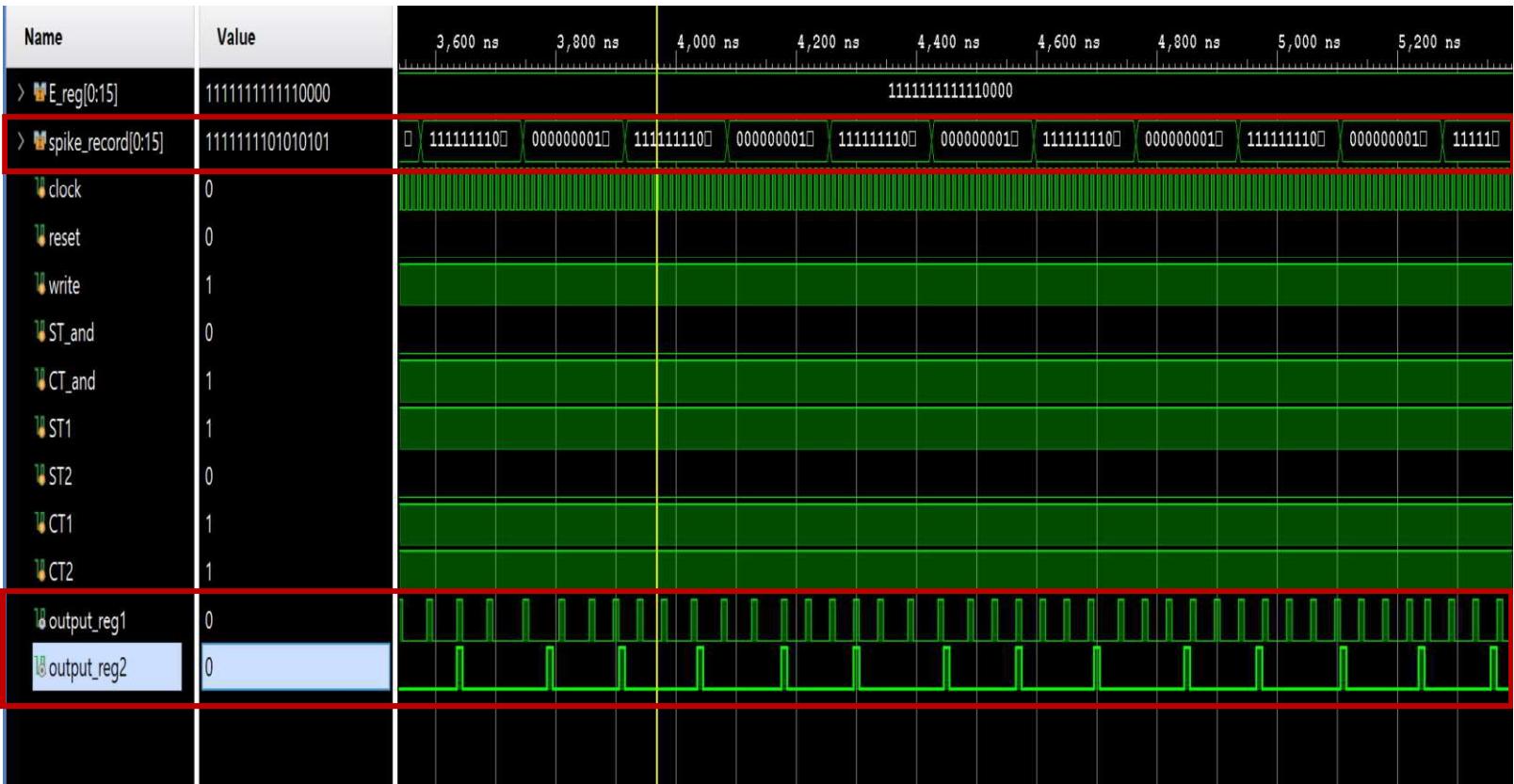
Reservoir Unit Simulation



Training Unit Schematic

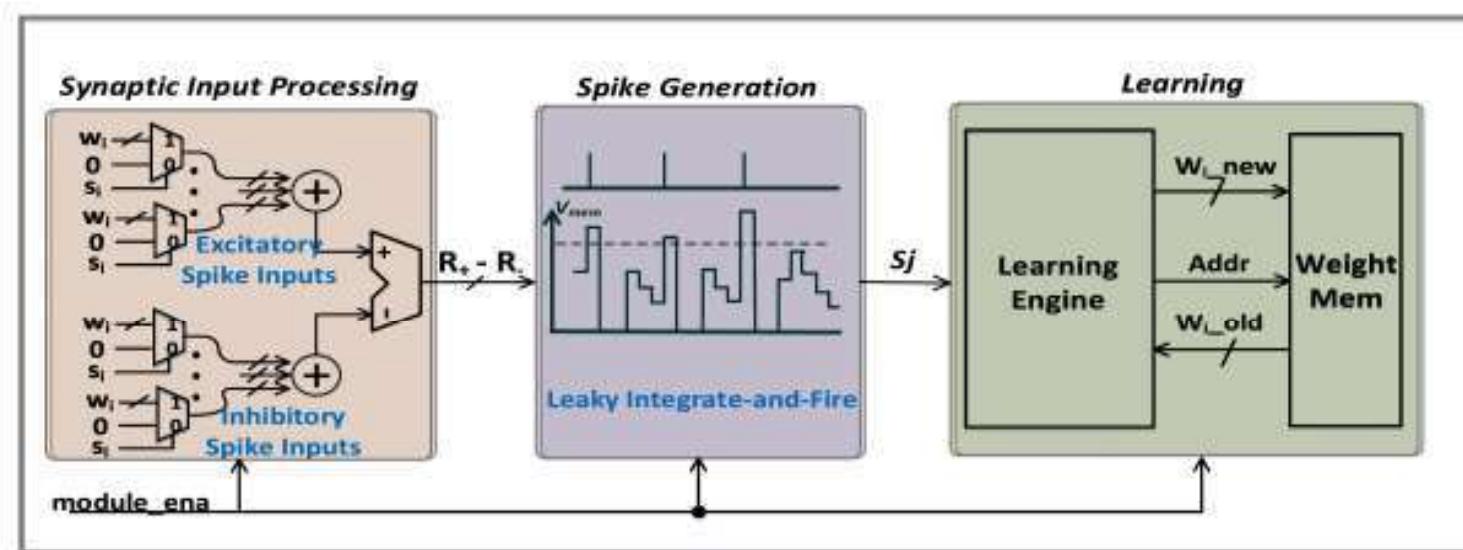


Training Unit Simulation

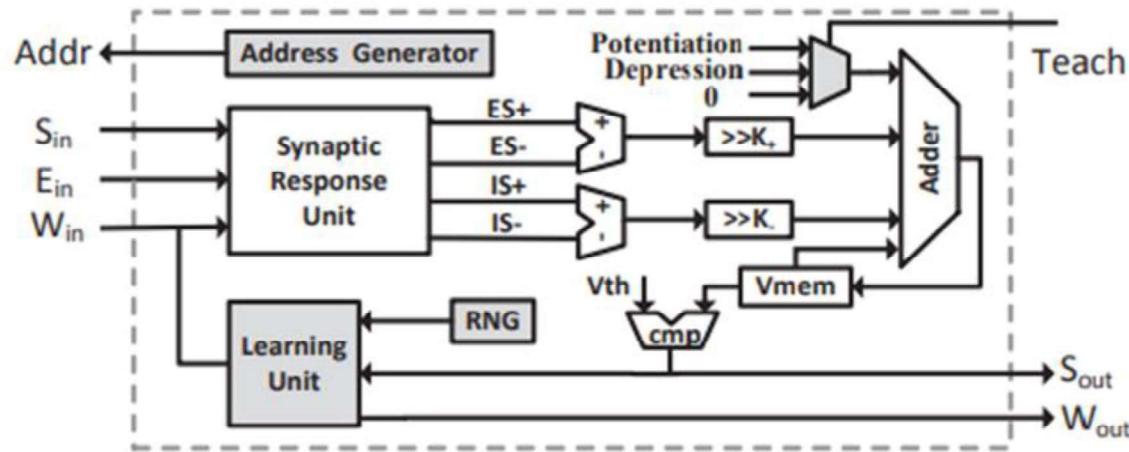


Implementation of Spiking Neuron

- Synaptic input processing module
- Spike generation module
- Learning module



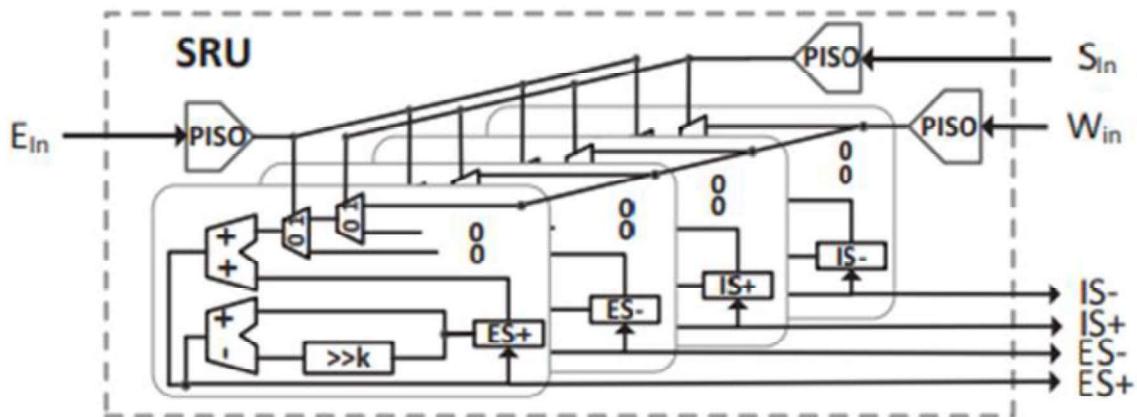
Implementation of digital neuron



$$V_{mem}(t) = V_{mem}(t-1) - \frac{V_{mem}(t-1)}{\tau} + R_+ - R_-$$

$$R_+ = \frac{ES_+ - ES_-}{\tau_{ES_+} - \tau_{ES_-}}, \quad R_- = \frac{IS_+ - IS_-}{\tau_{IS_+} - \tau_{IS_-}}$$

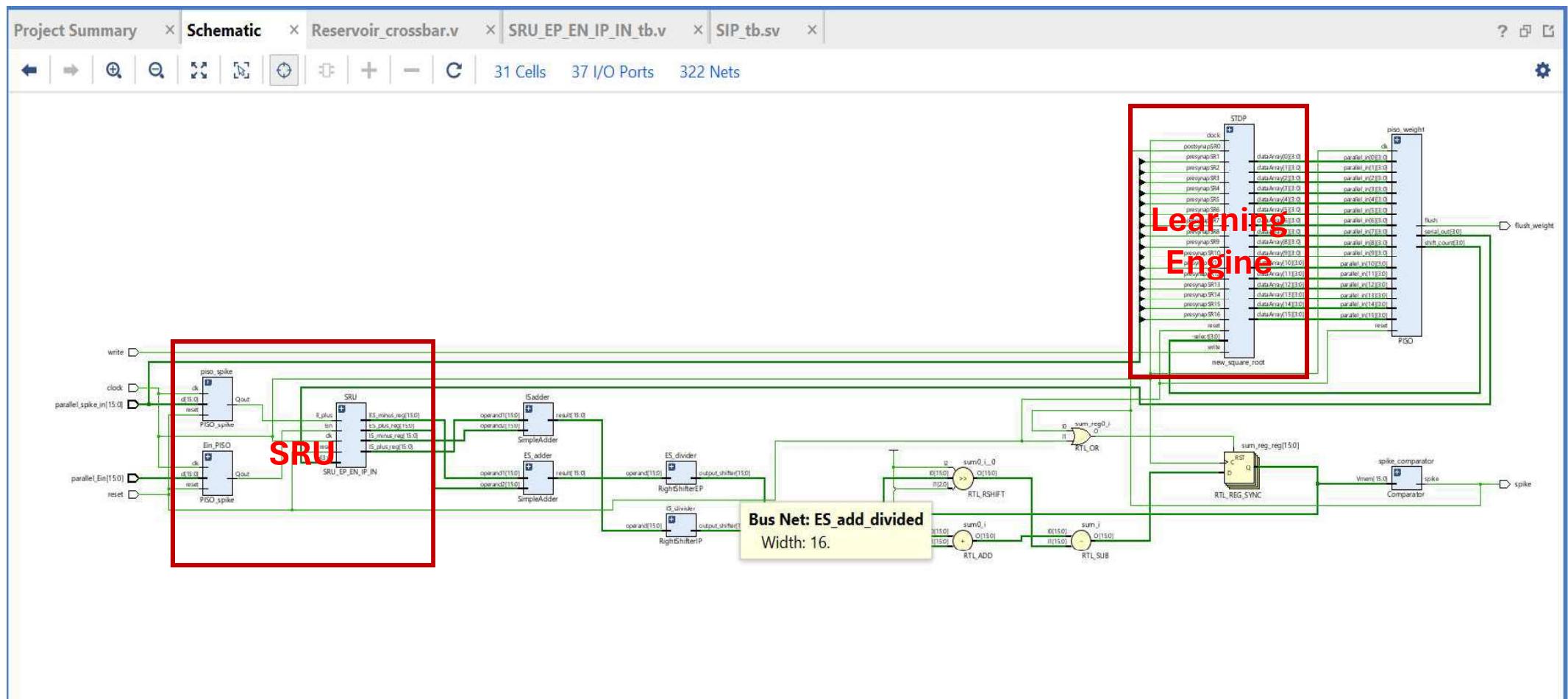
- Full Digital Neuron Architecture



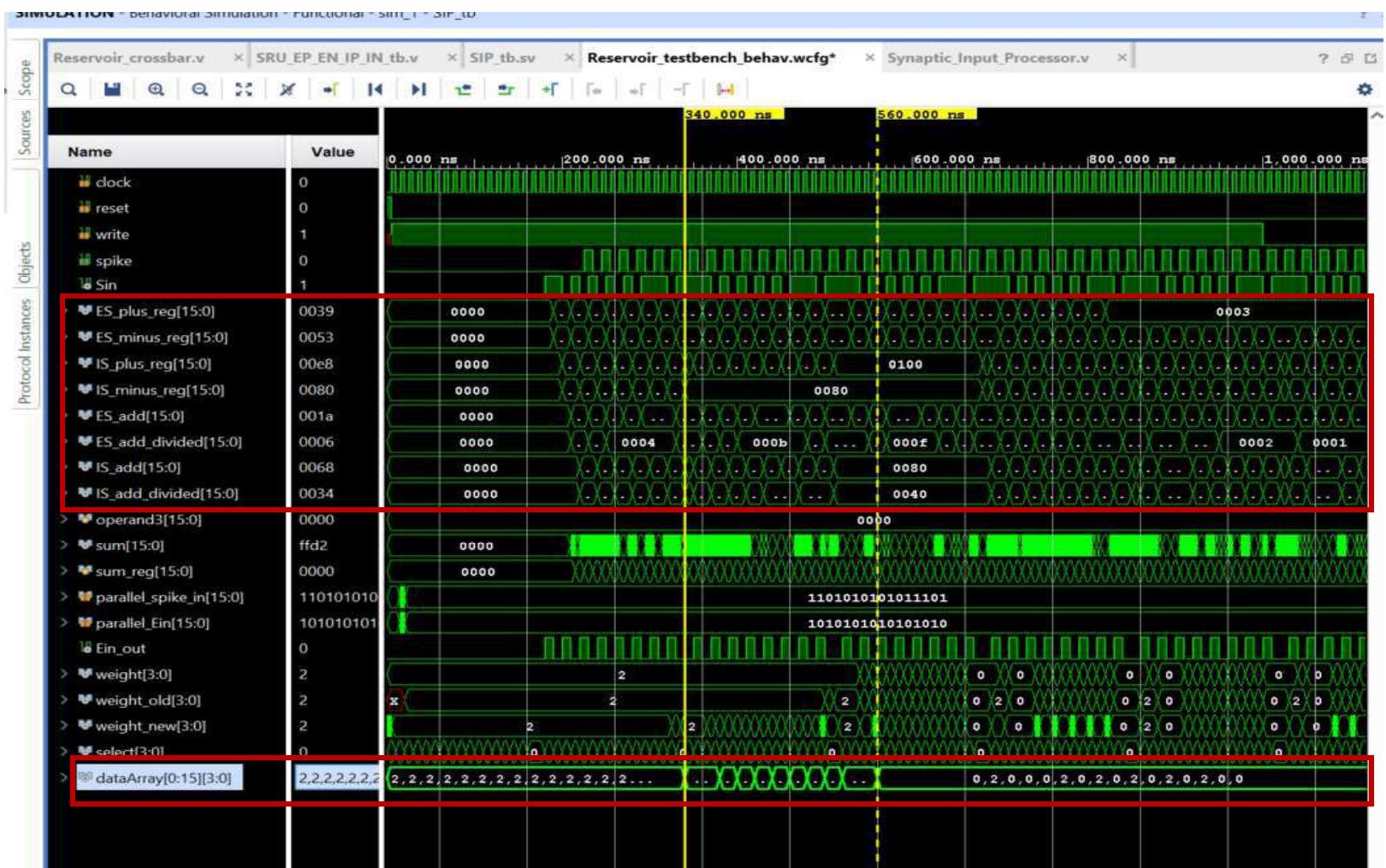
$$\begin{cases} ES_+(t) = ES_+(t-1)(1 - 1/\tau_{ES_+}) + \sum w_i \cdot E_+(i) \\ ES_-(t) = ES_-(t-1)(1 - 1/\tau_{ES_-}) + \sum w_i \cdot E_-(i) \\ IS_+(t) = IS_+(t-1)(1 - 1/\tau_{IS_+}) + \sum w_i \cdot E_-(i) \\ IS_-(t) = IS_-(t-1)(1 - 1/\tau_{IS_-}) + \sum w_i \cdot E_-(i) \end{cases}$$

- Synaptic Response Unit

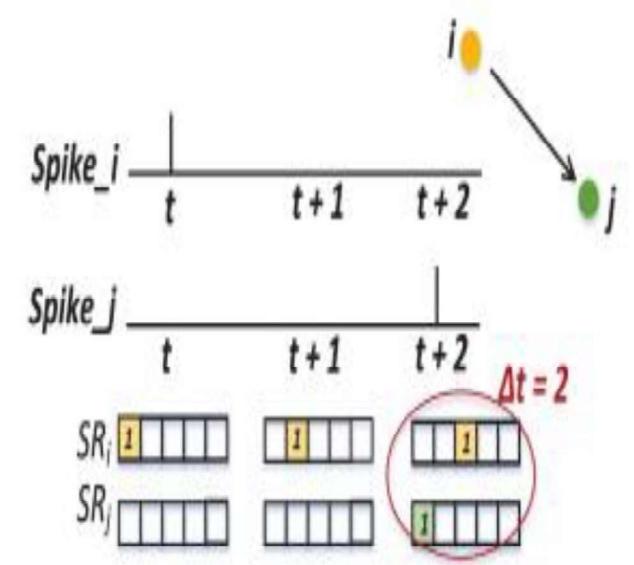
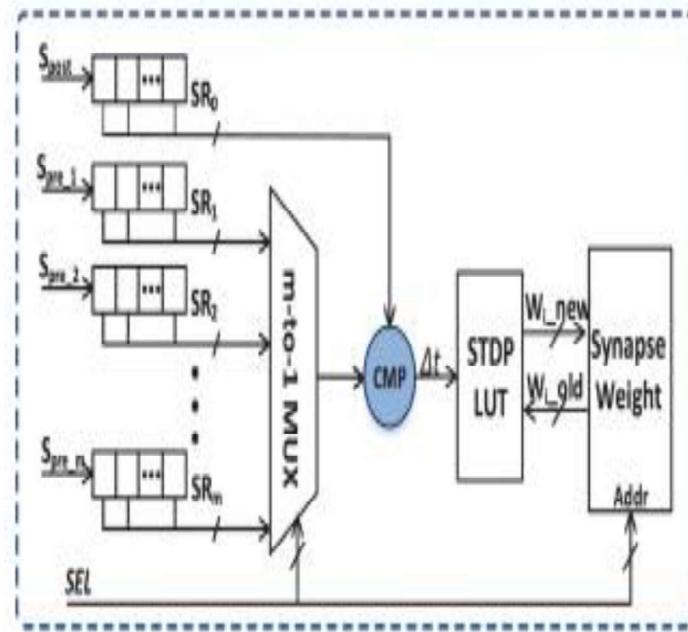
Reservoir Neuron Schematic



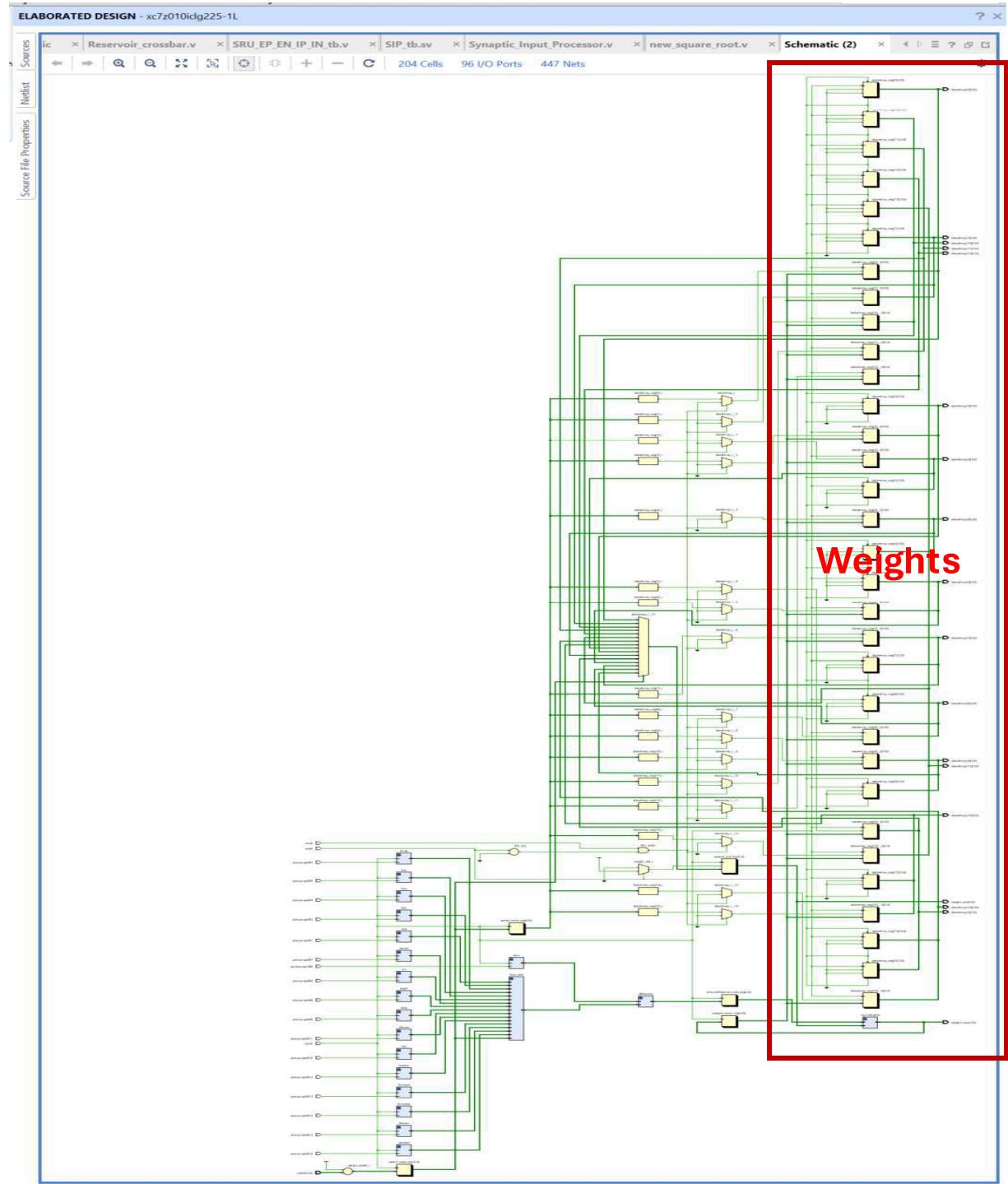
Reservoir Neuron Simulation



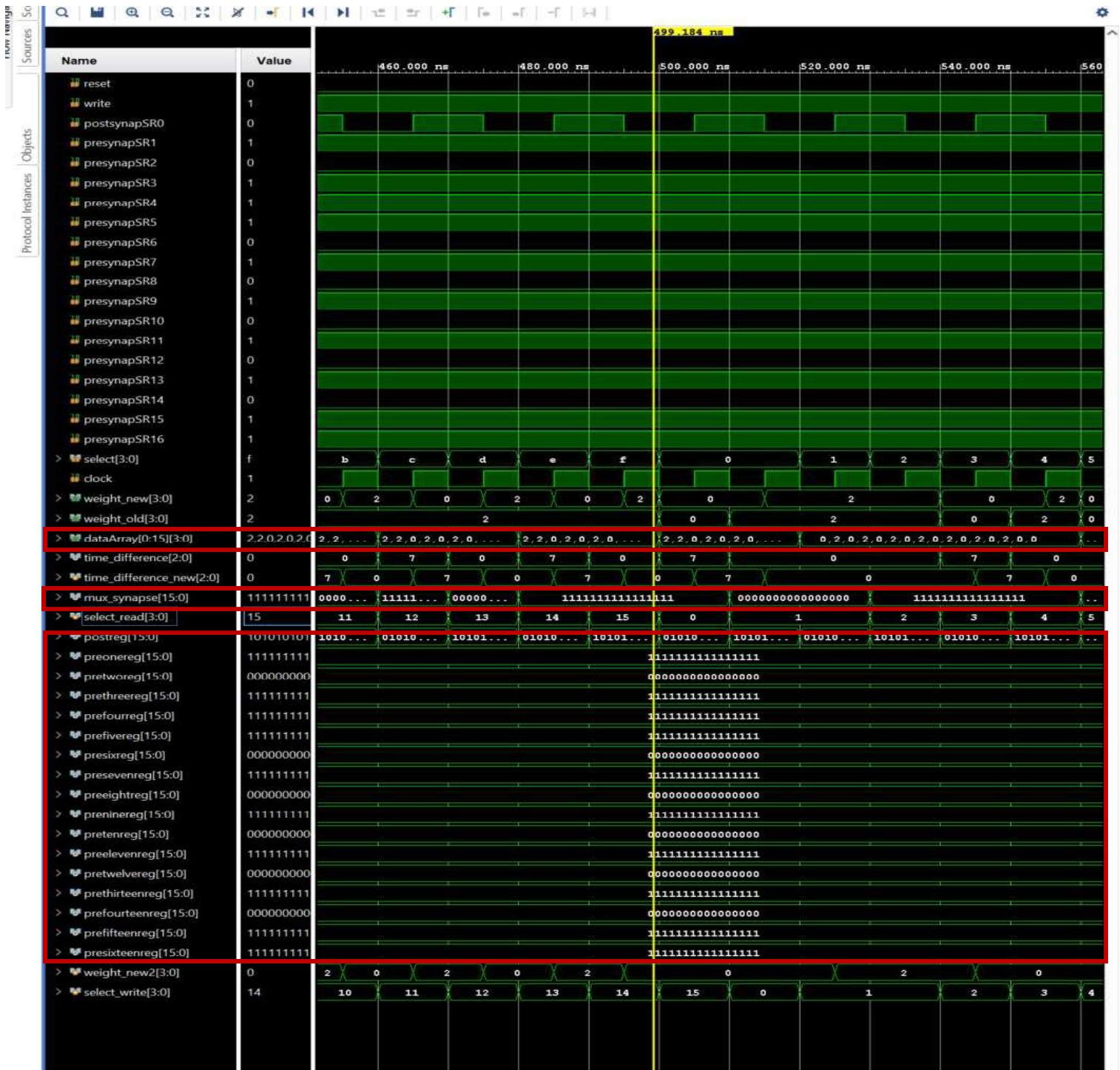
Learning Engine of Reservoir Neuron



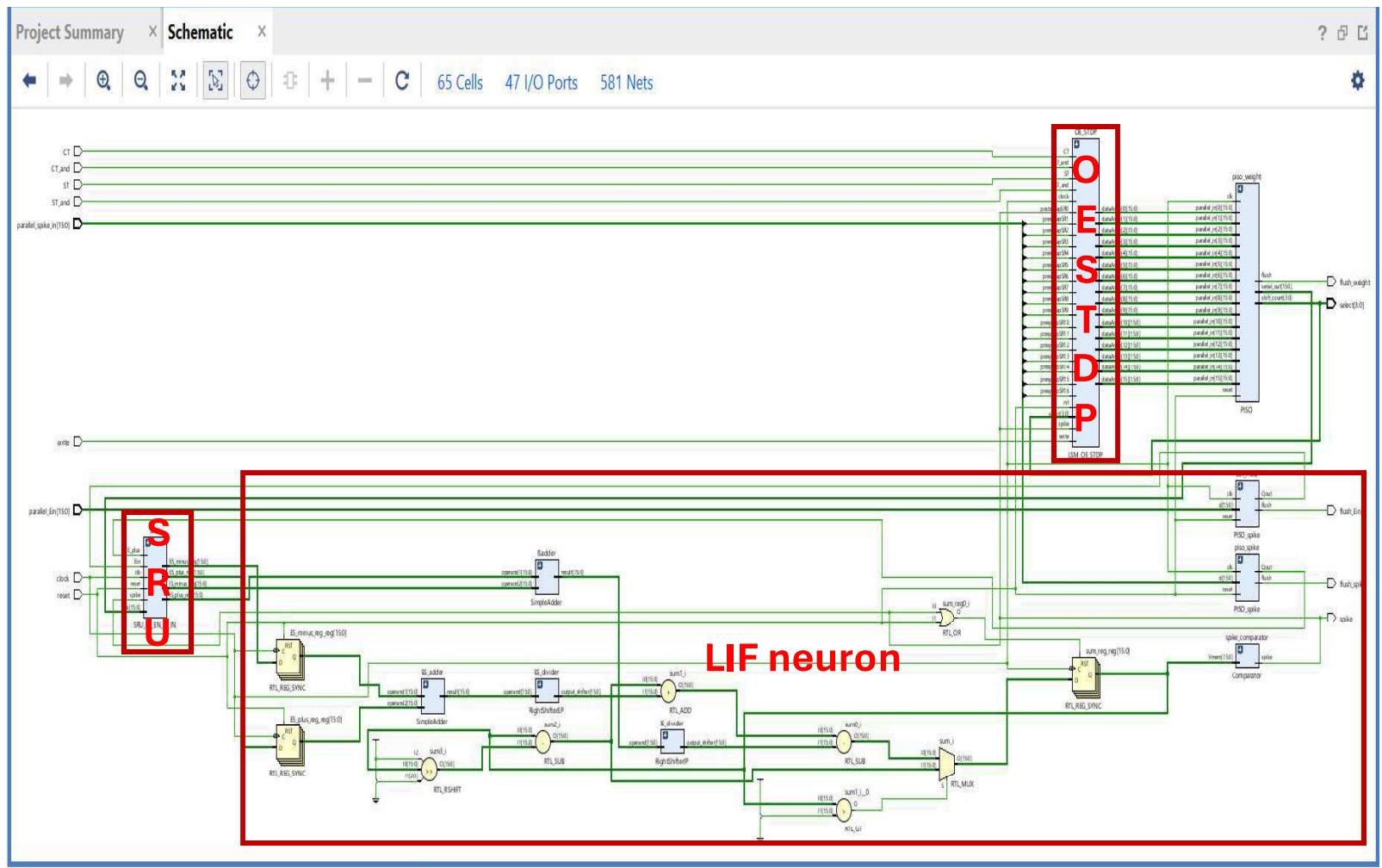
Learning Engine of Reservoir Neuron Schematic



Learning Engine of Reservoir Neuron Simulation



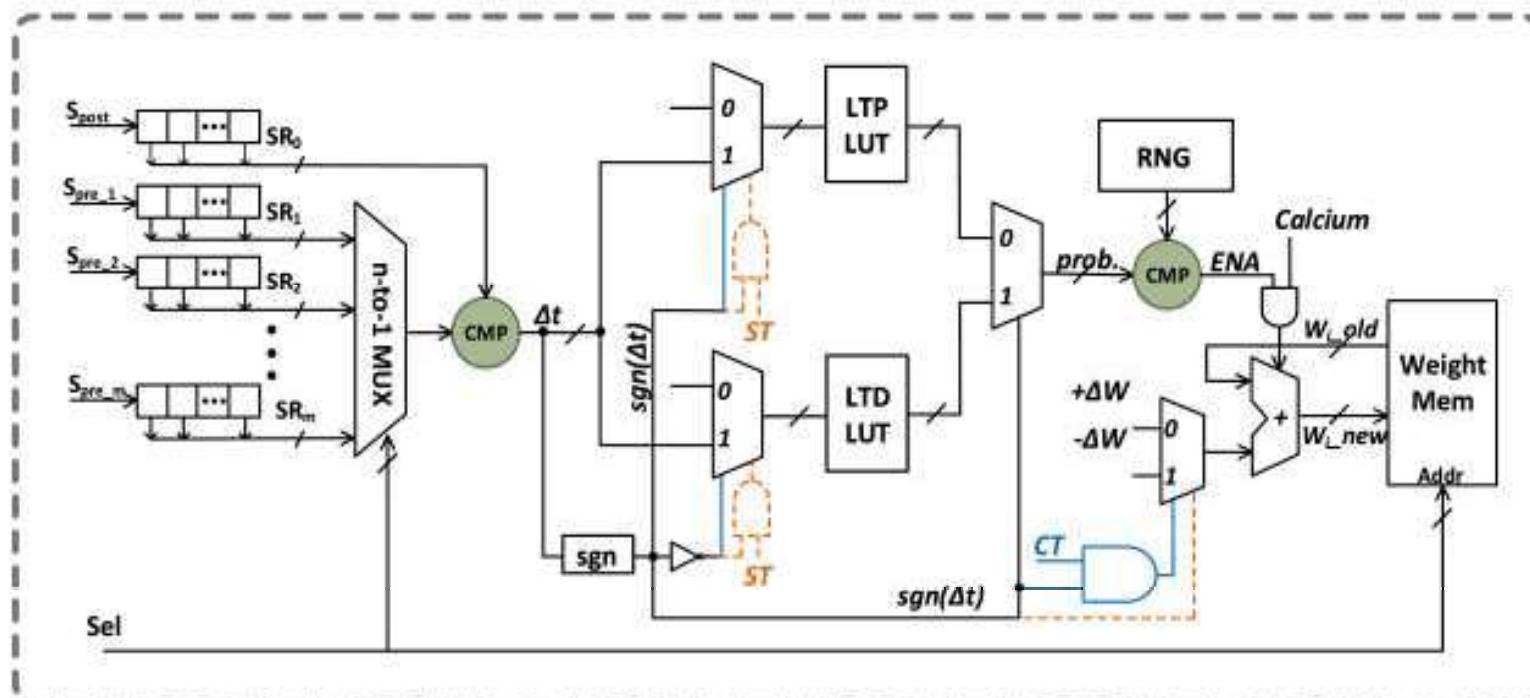
Implementation of Output neuron Schematic



Output Neuron Simulation

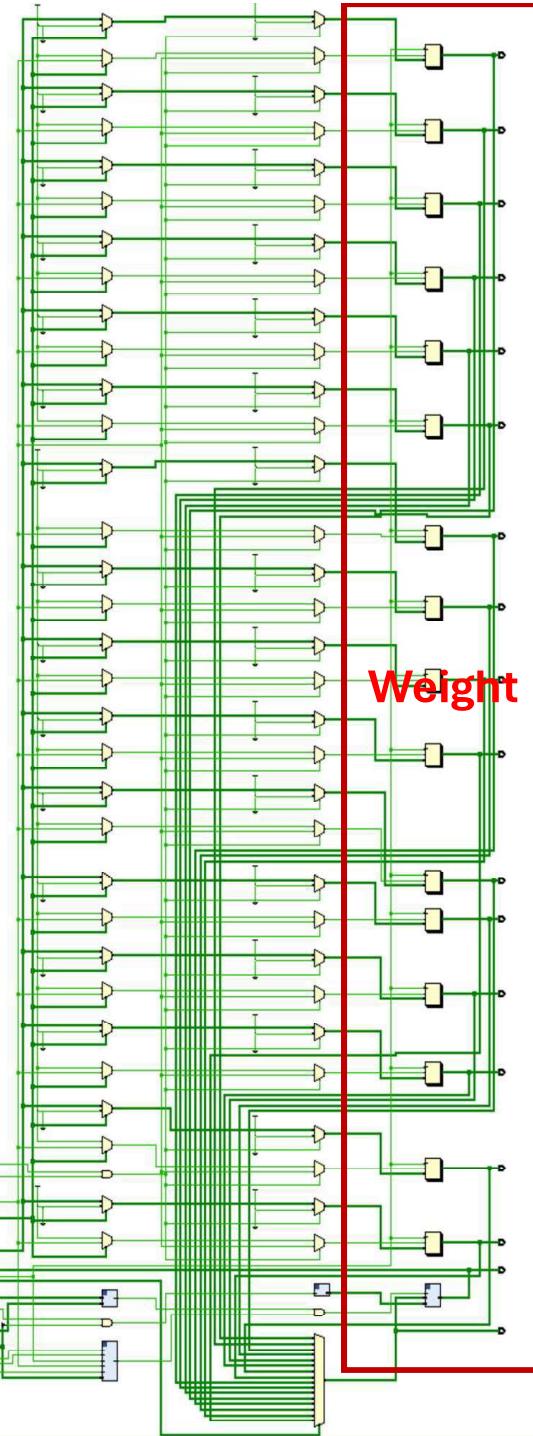
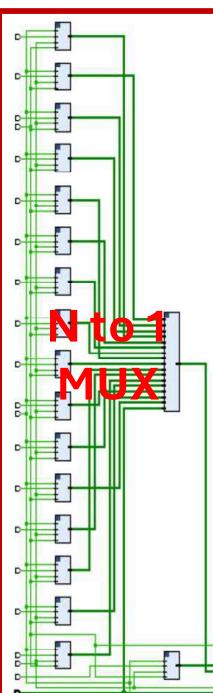


Output Neuron Learning Engine Implementation

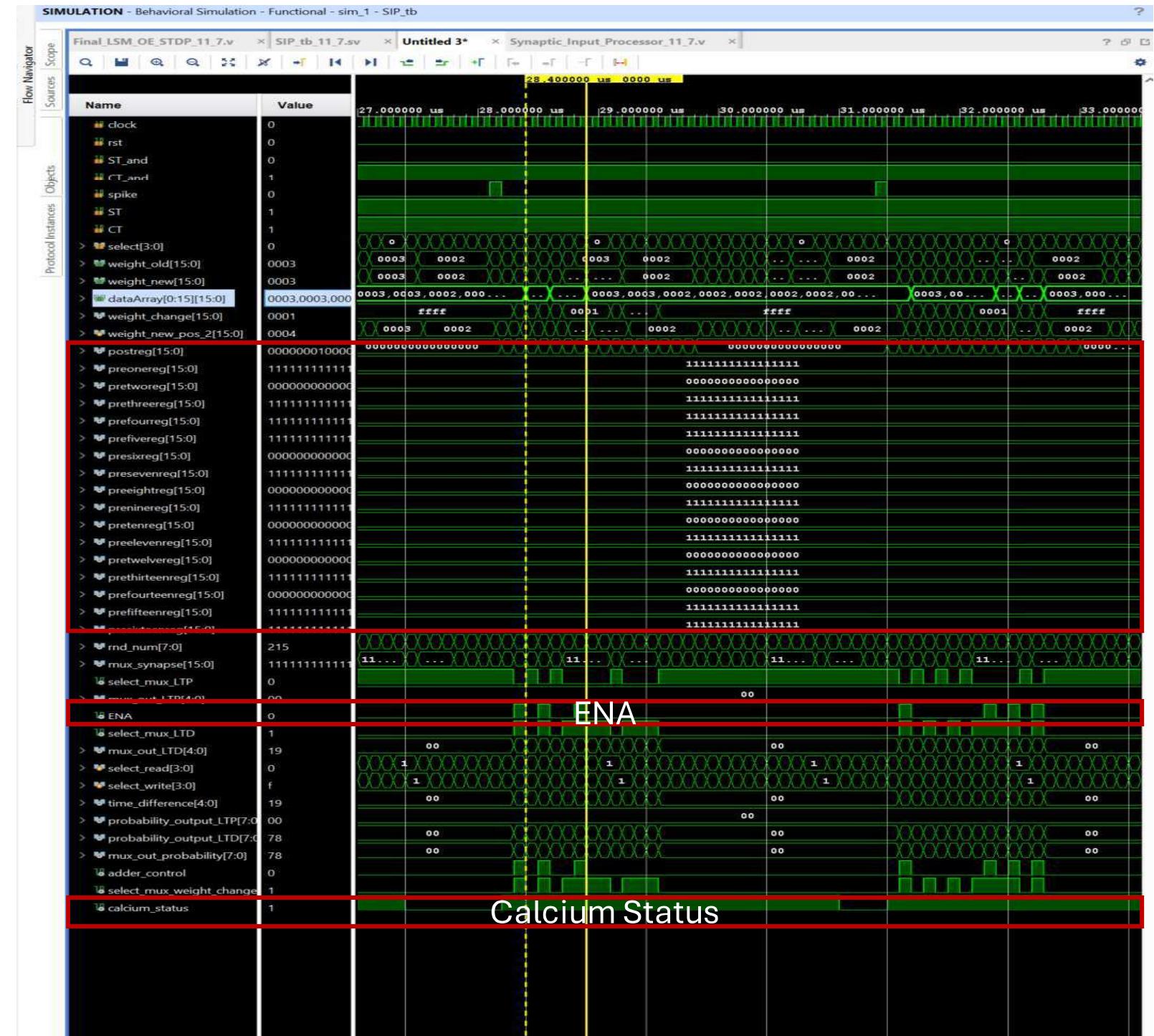


- Sparsification stage,
 - CT is always 1
 - ST is 1 for desired and 0 for the other
- Classification stage,
 - ST is always 1
 - CT is 1 for desired and 0 for the other

Output Neuron - Learning Engine Schematic



Output Neuron - Learning Engine Simulation



Benchmark and Result

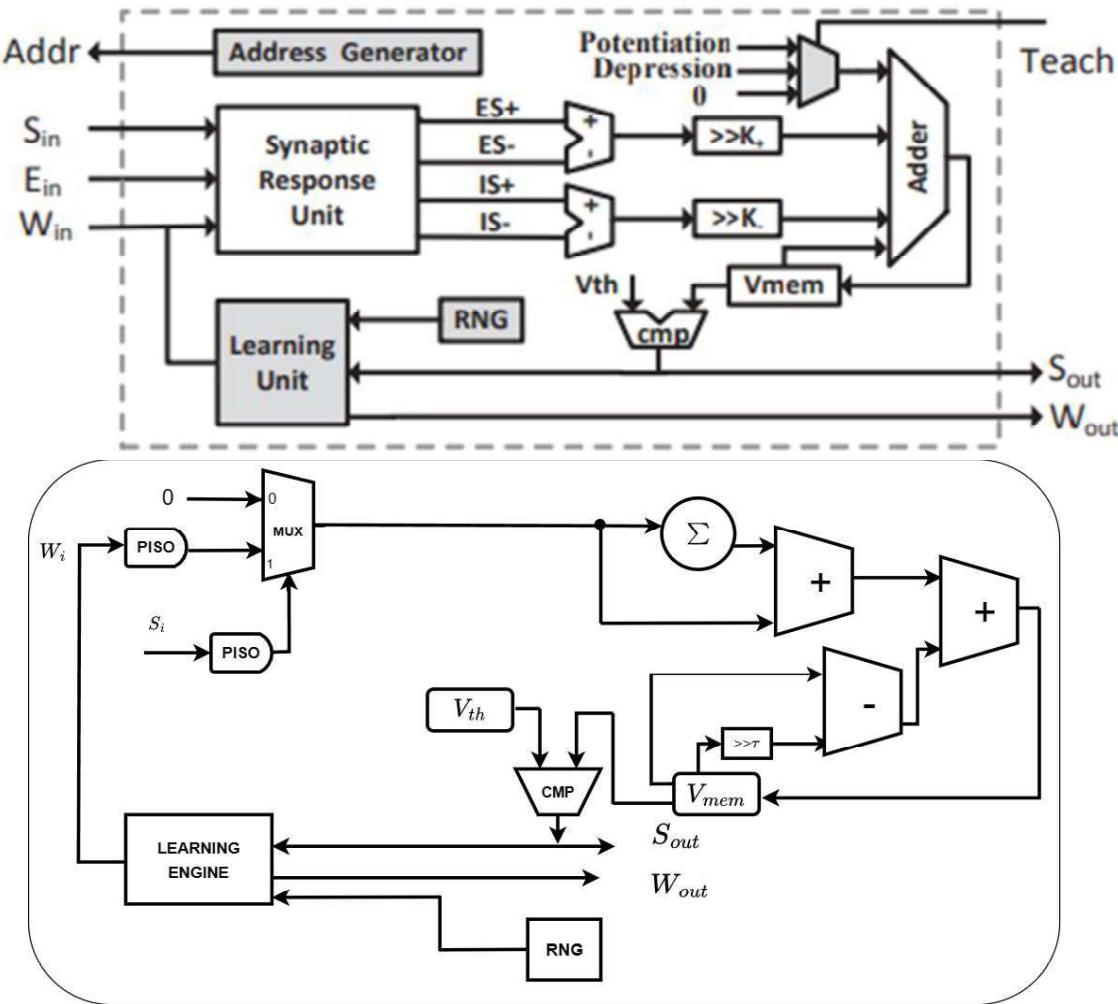
- Benchmark and Data Processing
 - Spectrum Sensing dataset from RWTH Achen University^[2]
 - Input contains one feature data of **received energy signal**
 - Central Frequency of channel (**Fc = 3750Mhz**)
 - Bandwidth (**Bw = 1500Mhz**)
 - Frequency resolution(**Fr = 200Khz**)
 - Total samples (7500)
 - Train data(6000)(80%)
 - Test data(1500)(20%)
 - Encoding scheme – Rate encoding

[2]Wang, L., Hu, J., Jiang, R., & Chen, Z. (2024). A Deep Long-Term Joint Temporal–Spectral Network for Spectrum Prediction. *Sensors*, 24(5), 1498.

Performance Analysis

SNR	Model	2 TX-RX	4 TX-RX	6 TX-RX
-10 dB	LSM	68.39%	98.86%	99.71%
-20 dB	LSM	64.53%	88.35%	95.79%

Schematic for optimized design



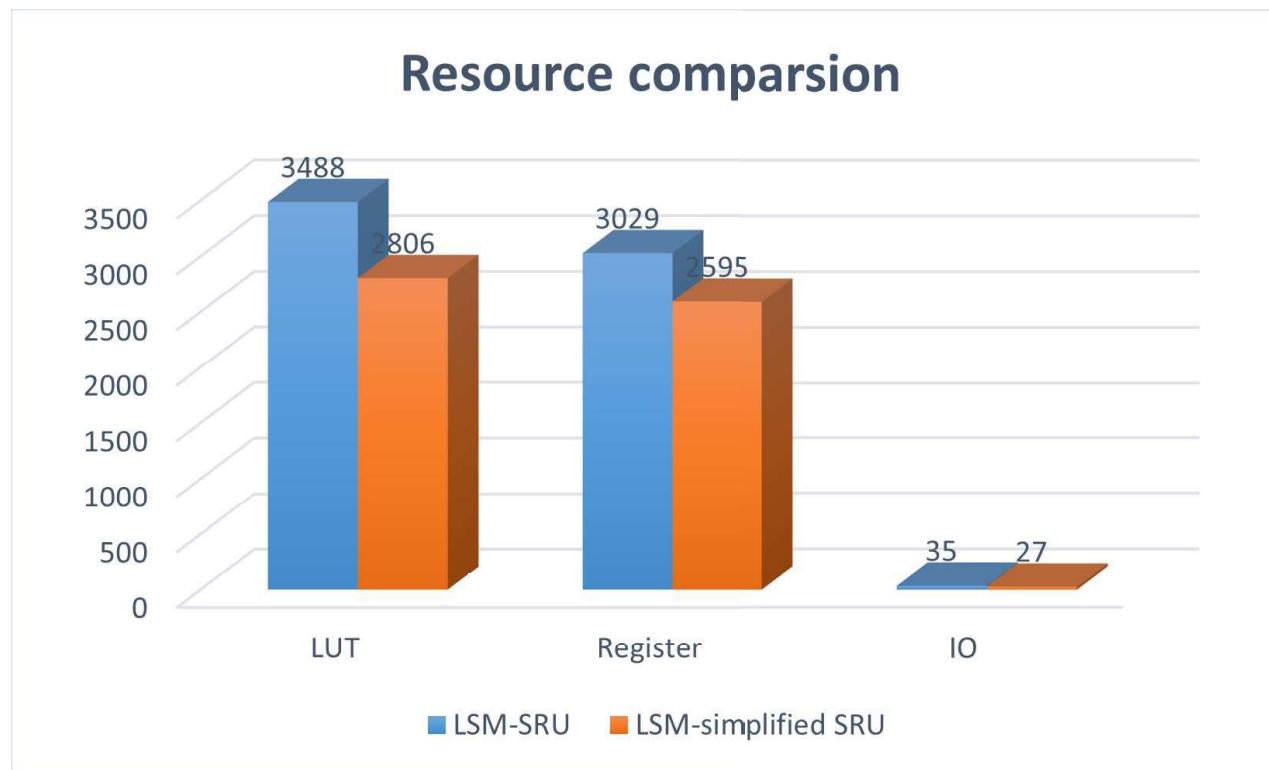
$$V_{mem}(t) = V_{mem}(t-1) - \frac{V_{mem}(t-1)}{\tau} + R_+ - R_-$$

$$R_+ = \frac{ES_+ - ES_-}{\tau_{ES_+} - \tau_{ES_-}}, \quad R_- = \frac{IS_+ - IS_-}{\tau_{IS_+} - \tau_{IS_-}}$$

$$V_{mem}(t) = V_{mem}(t-1) - V_{mem}(t-1)/\tau + \sum \text{weight}$$

[3] C. Lee, G. Srinivasan, P. Panda and K. Roy, "Deep Spiking Convolutional Neural Network Trained With Unsupervised Spike-Timing-Dependent Plasticity," in IEEE Transactions on Cognitive and Developmental Systems, vol. 11, no. 3, pp. 384-394, Sept. 2019, doi: 10.1109/TCDS.2018.2833071.

LSM hardware resource comparison



Resource utilization for a reservoir neuron with a SRU and simplified SRU

Name	^ 1	Slice LUTs (53200)	Slice Registers (106400)	F7 Muxes (26600)	F8 Muxes (13300)	Bonded IOB (200)	BUFGCTRL (32)
LSM_reservoir		2806	2595	94	36	27	1
Exhibitory_0 (Synaptic_Input_Processor)		178	204	14	6	0	0
new_square (new_square_root_106)	>	125	93	8	3	0	0
new_SRU (new_SRU_simplified_105)		15	15	0	0	0	0
piso_spike (PISO_spike_107)		13	22	0	0	0	0
piso_weight (PISO_108)		23	58	6	3	0	0

LSM_reservoir	3488	3029	94	36	35	1
Exhibitory_0 (Synaptic_Input_Processor)	326	271	14	6	0	0
Ein_PISO (PISO_spike_124)	12	18	0	0	0	0
ES_adder (SimpleAdder_123)	0	0	0	0	0	0
ISadder (SimpleAdder_125)	10	0	0	0	0	0
new_square (new_square_root_127)	126	93	8	3	0	0
piso_spike (PISO_spike_128)	13	22	0	0	0	0
piso_weight (PISO_129)	18	58	6	3	0	0
SRU (SRU_EP_EN_IP_IN_126)	146	64	0	0	0	0

Summary

- Implemented and verified the whole LSM in RTL with spectrum sensing dataset
- Optimized the digital neuron
 - Replaced synaptic response unit with a simplified design
 - Around 50% less power consumption and 20% less resource utilization

Future improvement

- Reducing the hardware utilization
- Running in complex dataset
- Prepare the design for regression problem
- Compare with another type of reservoir neural network, Echo state network

References

- [1] Liu, Yu, Sai Sourabh Yenamachintala, and Peng Li. "Energy-efficient FPGA spiking neural accelerators with supervised and unsupervised spike-timing-dependent-plasticity." ACM Journal on Emerging Technologies in Computing Systems (JETC) 15.3 (2019): 1-19.
- [2] Wang, L., Hu, J., Jiang, R., & Chen, Z. (2024). A Deep Long-Term Joint Temporal–Spectral Network for Spectrum Prediction. *Sensors*, 24(5), 1498.
- [3] C. Lee, G. Srinivasan, P. Panda and K. Roy, "Deep Spiking Convolutional Neural Network Trained With Unsupervised Spike-Timing-Dependent Plasticity," in IEEE Transactions on Cognitive and Developmental Systems, vol. 11, no. 3, pp. 384-394, Sept. 2019, doi: 10.1109/TCDS.2018.2833071.