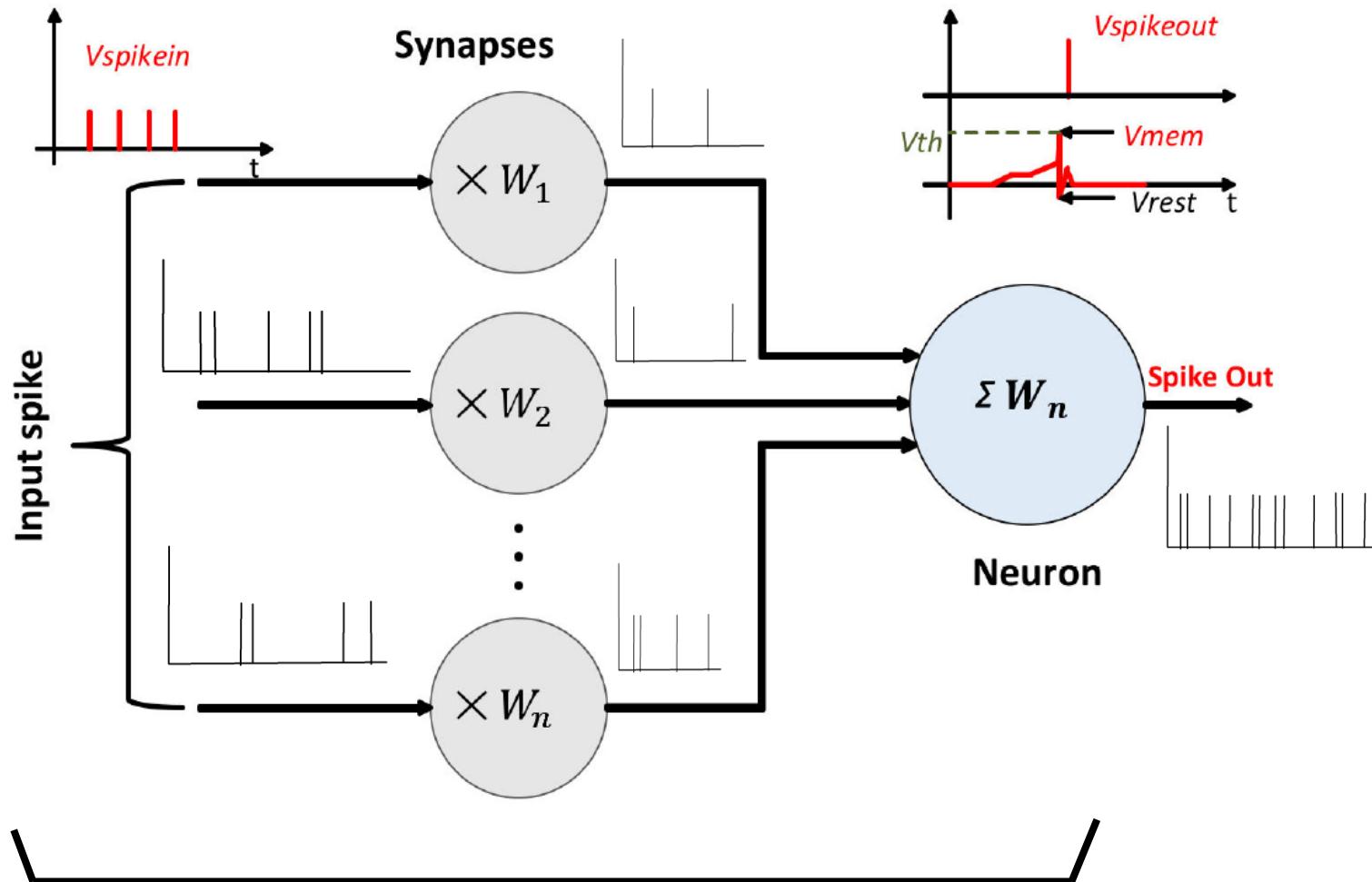


Spiking Neuron Model in CMOS GPDK180 – Charge Pump based Synapse and Schmitt Trigger based LIF Spike Generator

Nick Iliev

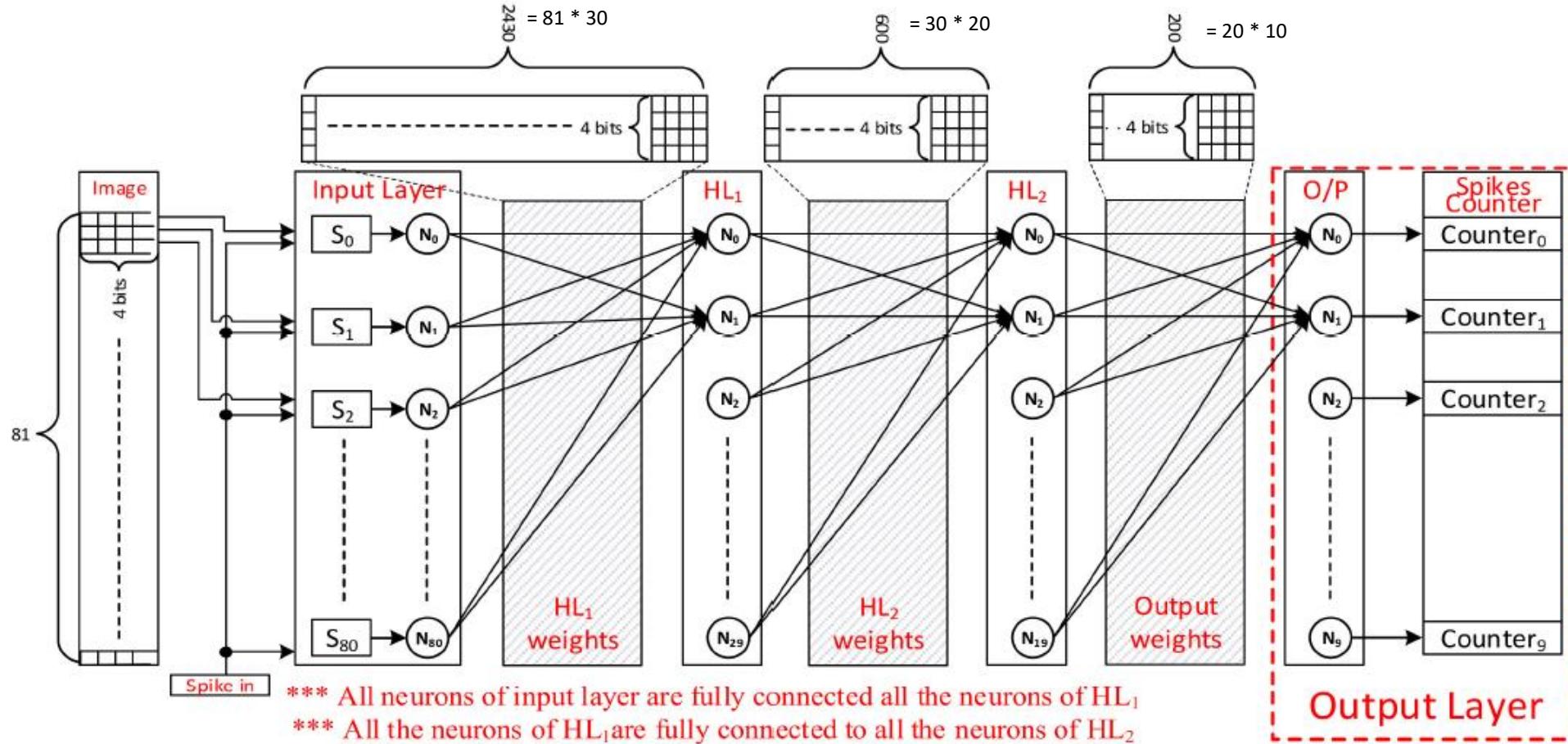
Nov 30 2021

Basic Model of a Spiking Neuron : a single neuron connected to multiple input synapses (input weights)



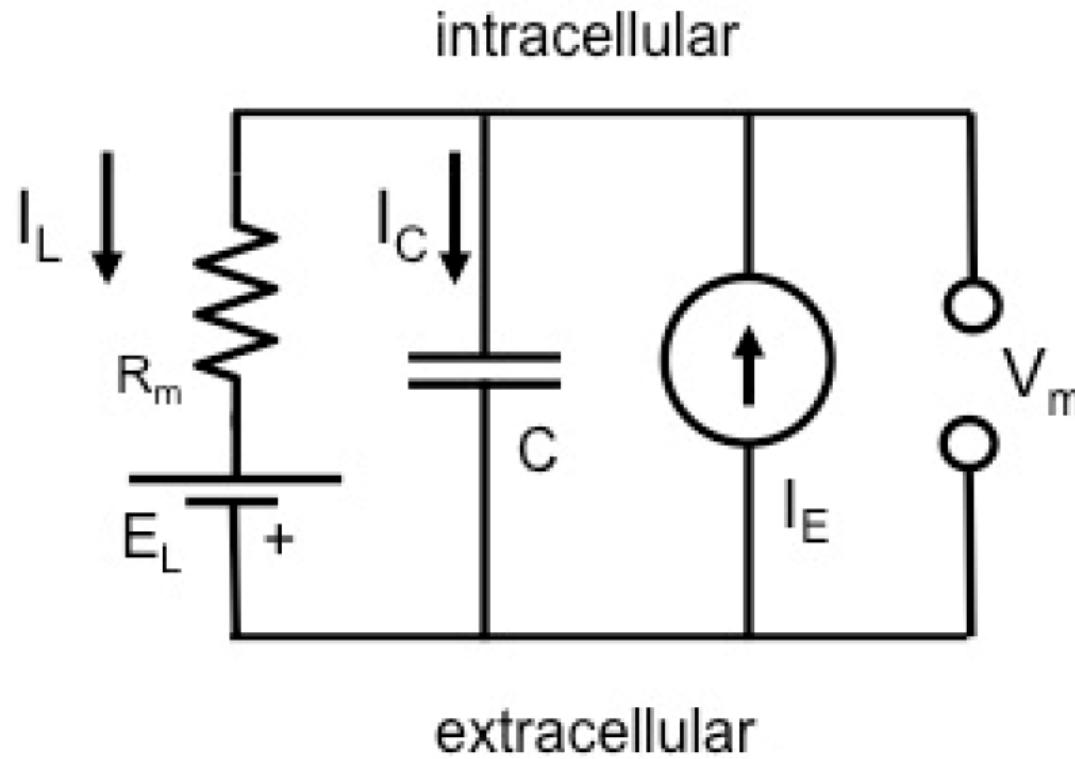
This is N_x on the next slide

Example Application : 4-layer Fully-Connected (MLP) Classifier of MNIST digits (81x4 pixels per input image for a digit)



Reference : M.S. Asghar et al. 2021 “A Low-Power Spiking Neural Network Chip Based on a Compact LIF Neuron and Binary Exponential Charge Injector Synapse Circuits”, No float (fixed-point) math (mult/add), no ReLu (activation function)

RC Circuit Model for a Leaky Integrate and Fire (LIF) Spiking Neuron



Equivalent RC circuit of a neuron's membrane : capacitor C (C_{mem}) models a cell's membrane; the membrane, capacitor C , provides electrical insulation between the intracellular fluid and the extracellular fluid; C has a capacitance proportional to the cell's surface area A ; Votage difference across membrane = V_m

Input electrical stimulation = I_E ; membrane's resistance = R_m ; leakage current = I_L ;
membrane resting potential = E_L

Time-domain Dynamics of the above RC model, for injected current I_E

Starting with Kirchhoff's current law :

$$C_m \frac{dV_m}{dt} + \frac{V_m - E_L}{R_m} = I_E$$

which can be written as :

$$V_m(t) + \tau_m \frac{dV_m}{dt} = E_L + R_m I_E \quad \text{With } \tau_m = R_m C_m$$

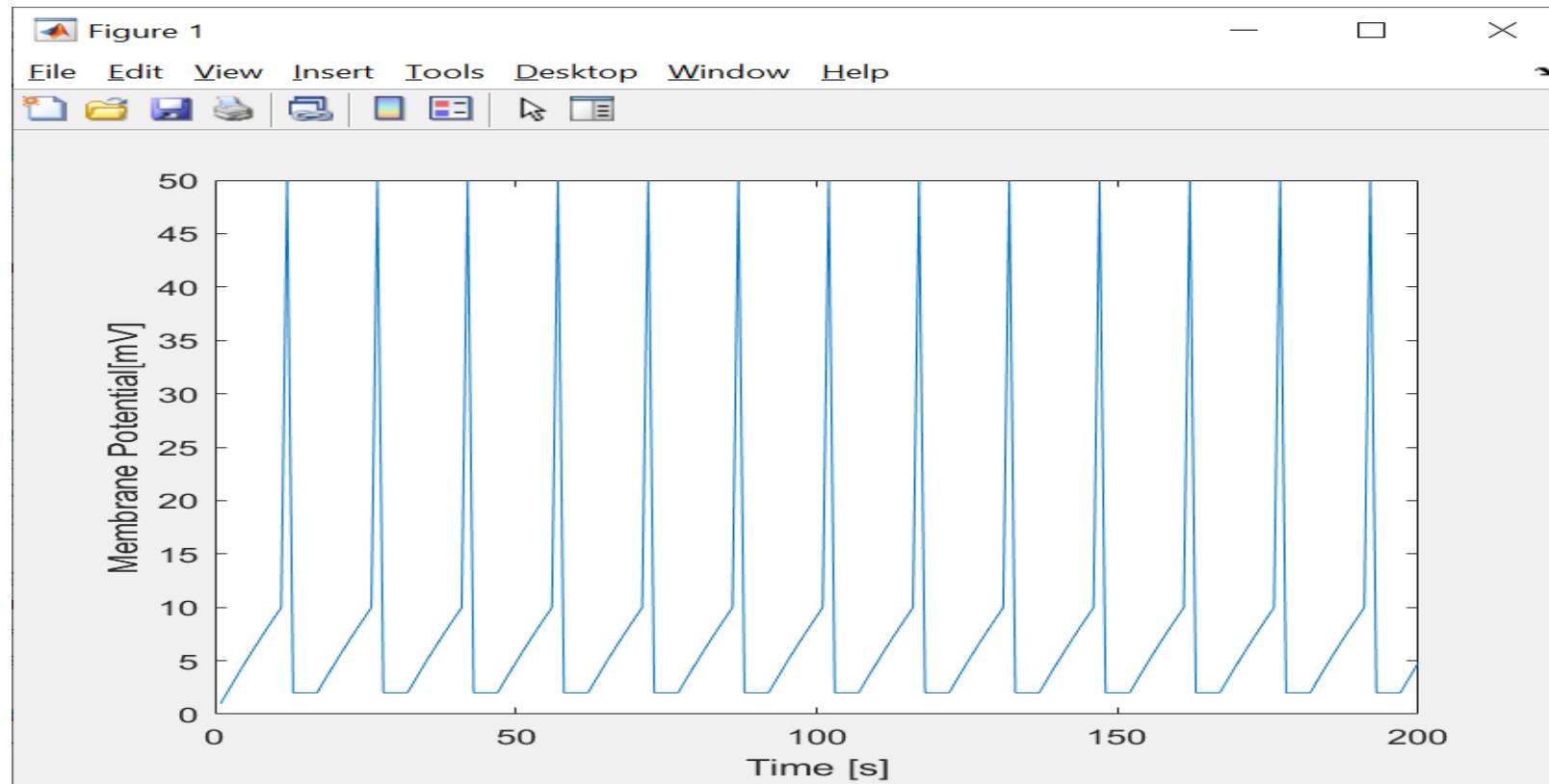
And we get the difference equation :

$$V_m(t + \Delta t) = V_m(t) + \frac{\Delta t (E_L - V_m(t) + R_m I_E(t))}{\tau_m}$$

replacing dV_m as $V_m(t + \Delta t) - V_m(t)$ and dt by Δt

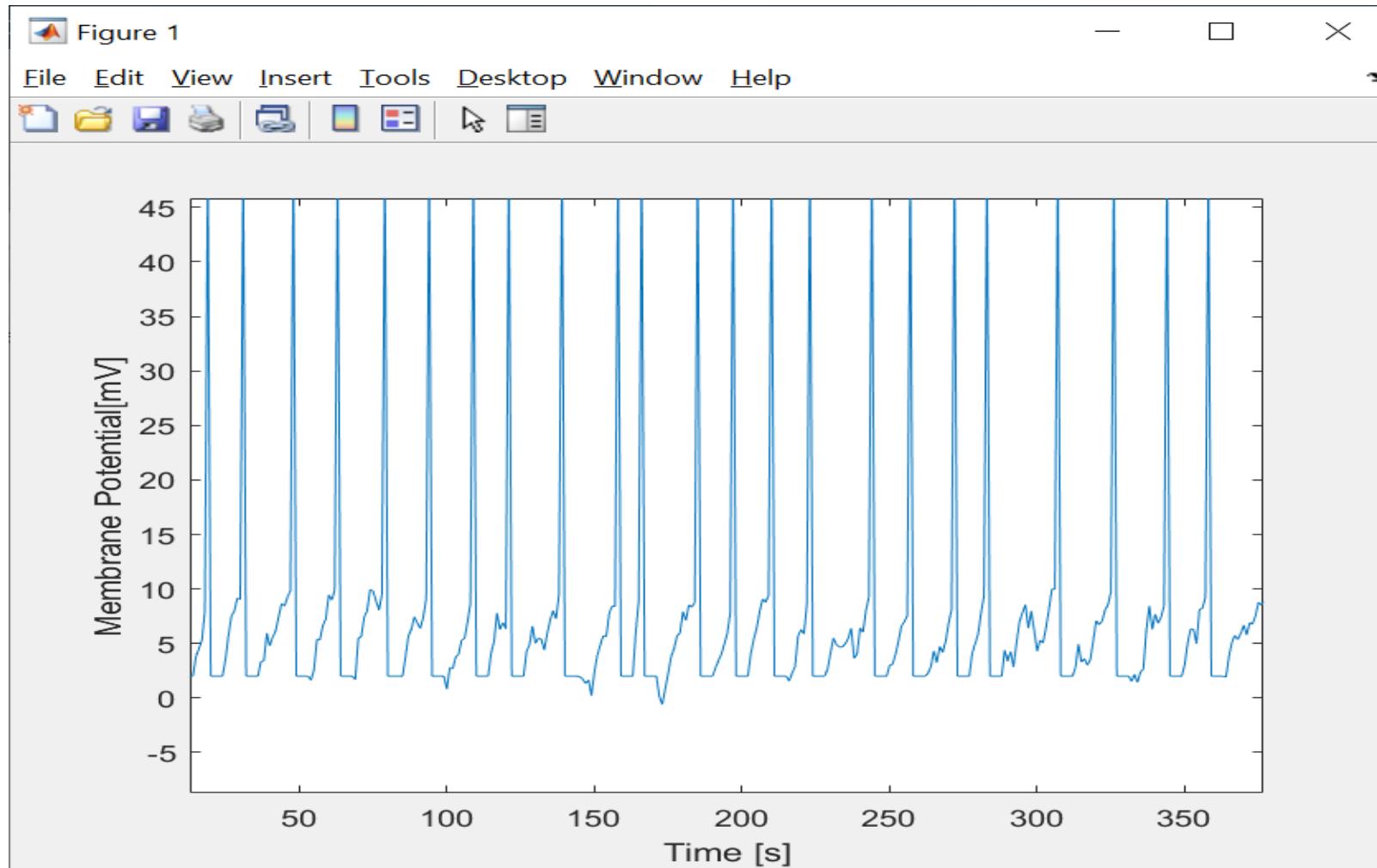
Here Δt is the integration time-step, the time elapsed from t to $t + \Delta t$.

Matlab simulation of the above difference equation shows the periodic Spikes Generated for $V_m(t)$ (Y axis in the plot)

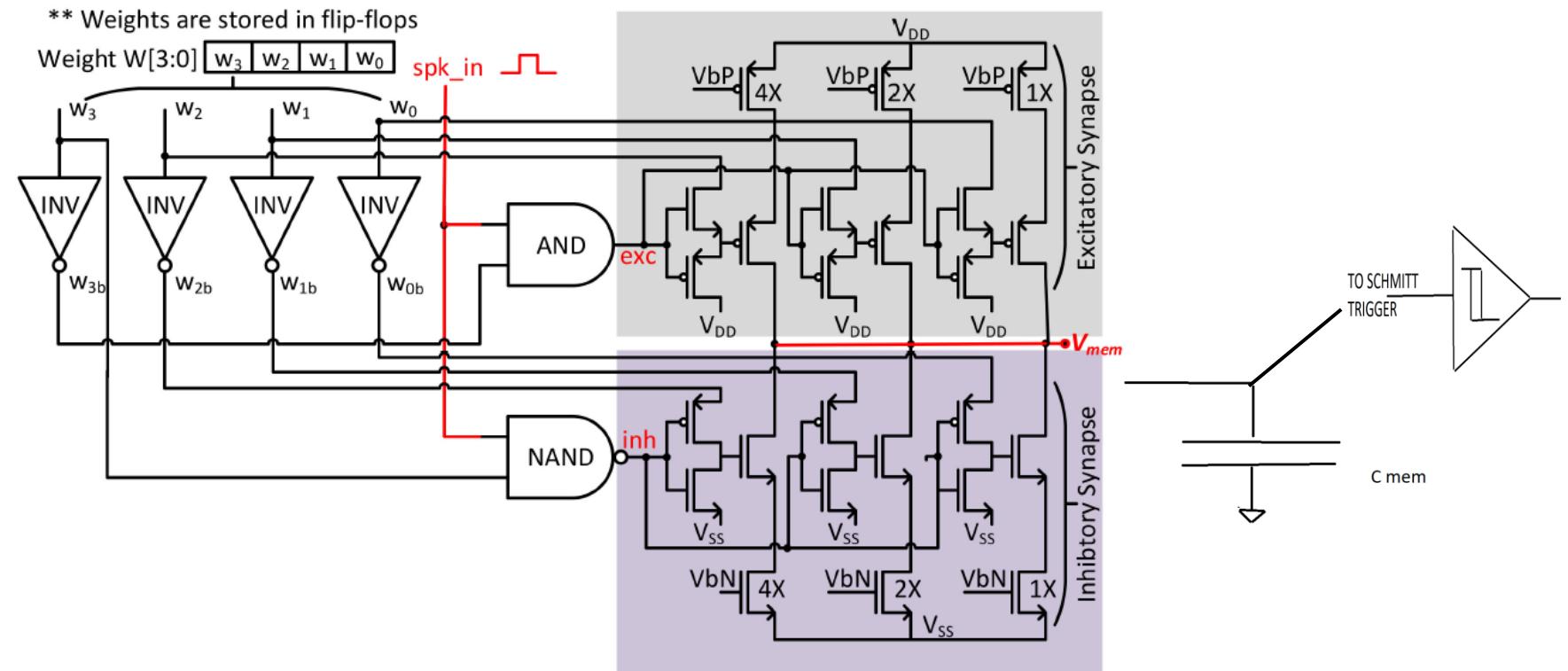


In the plot, input current I_E is constant at 1 nA ; $C = 1 \text{ nF}$, $R_m = 50 \text{ Mohms}$; $E_L = 2 \text{ mV}$; $V_{\text{threshold}} = 10 \text{ mV}$

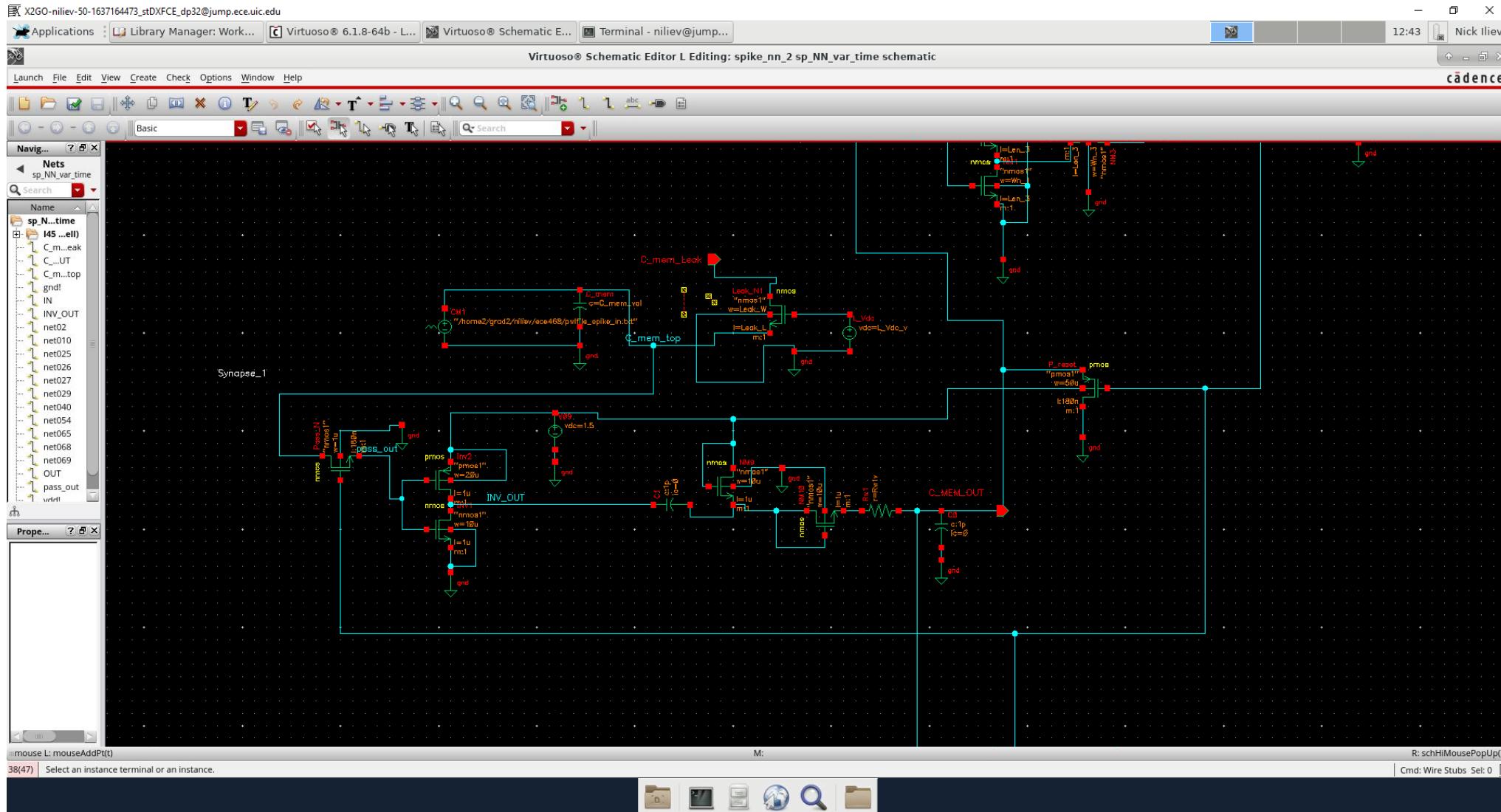
Irregular (non-periodic) Spikes for $V_m(t)$ can be generated by simply adding random noise to input current I_E as shown below (biological neurons also have noisy input currents) ; or random V threshold changes



Recent CMOS circuit implementation of 1 synapse with a 4-bits weight (signed) is proposed by Asghar et al. 2021. This 4-bits weight has to be learned during off-line training. Device count is 24 devices for this synapse, for the 1X,2X,4X Excitatory/Inhibitory circuits for charging/discharging C_{mem} (V_{mem} in the figure) ; 16 devices are needed for the signed 4-bit weight + logic; 40 devices total, for 1 synapse



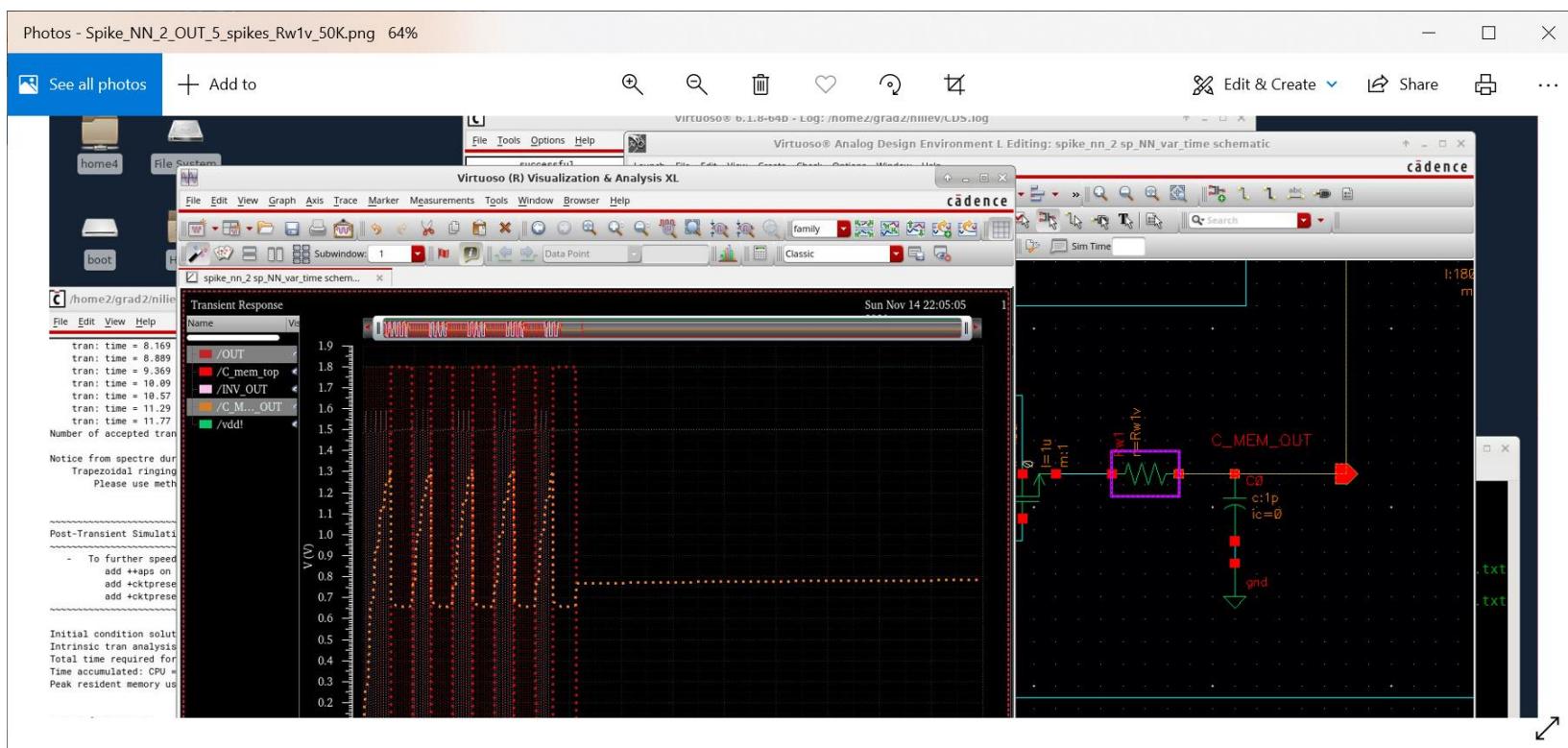
My idea : keep a 4-bits positive (unsigned) weight only, and use a charge pump in the synapse for charging C mem ; devices NM9, NM10 and caps C1, C0, act as a charge pump, charging from approx. 600 mV to 1.2 V (doubling)



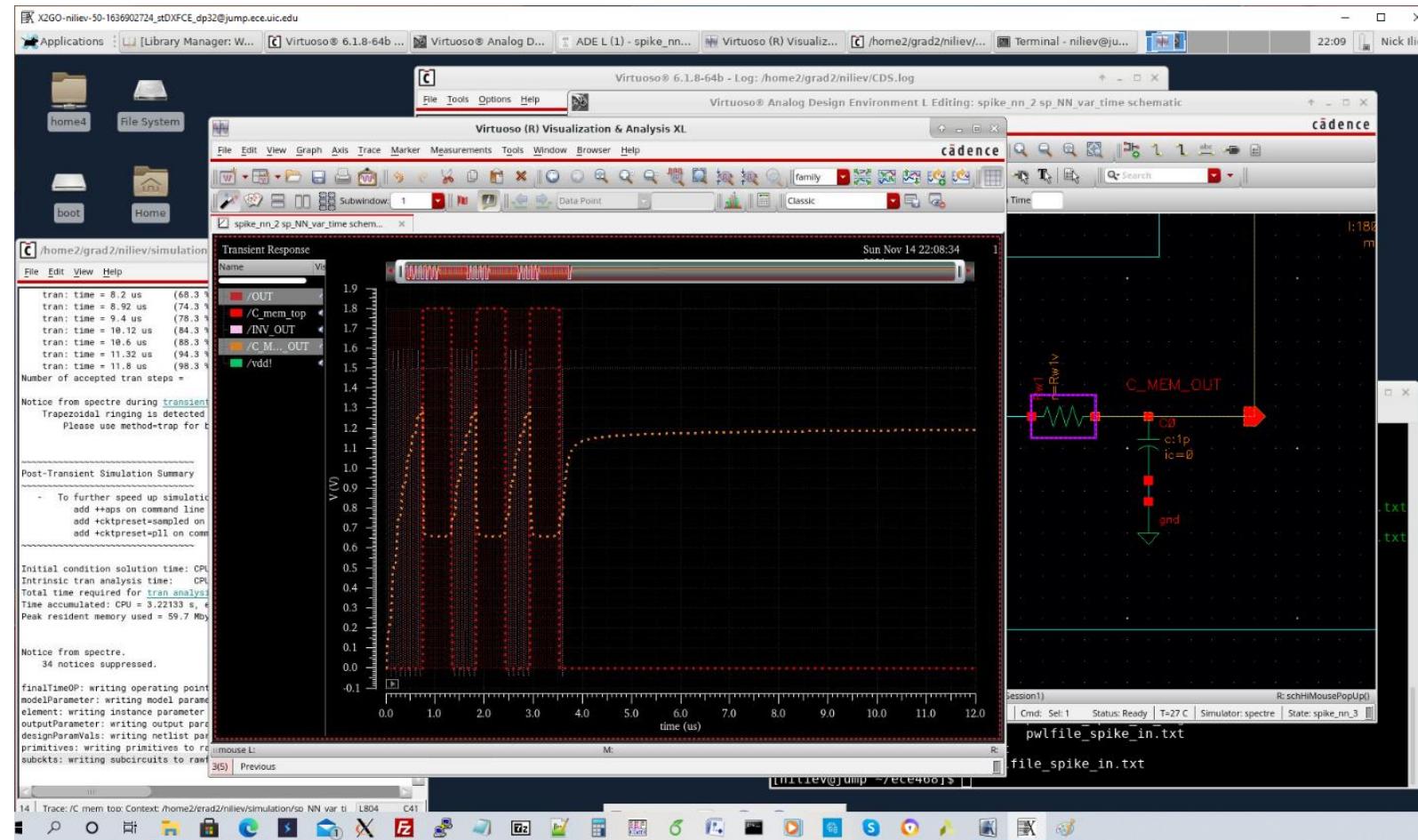
My idea : input periodic spikes come into Synapse_1 via devices Pass_N and Inv2; resistor R_{w1} forms an RC circuit with C_0 (C_{MEM_OUT} , the membrane capacitor).

The resulting RC time constant causes the following Schmitt trigger to trigger at specific times, for each RC value;

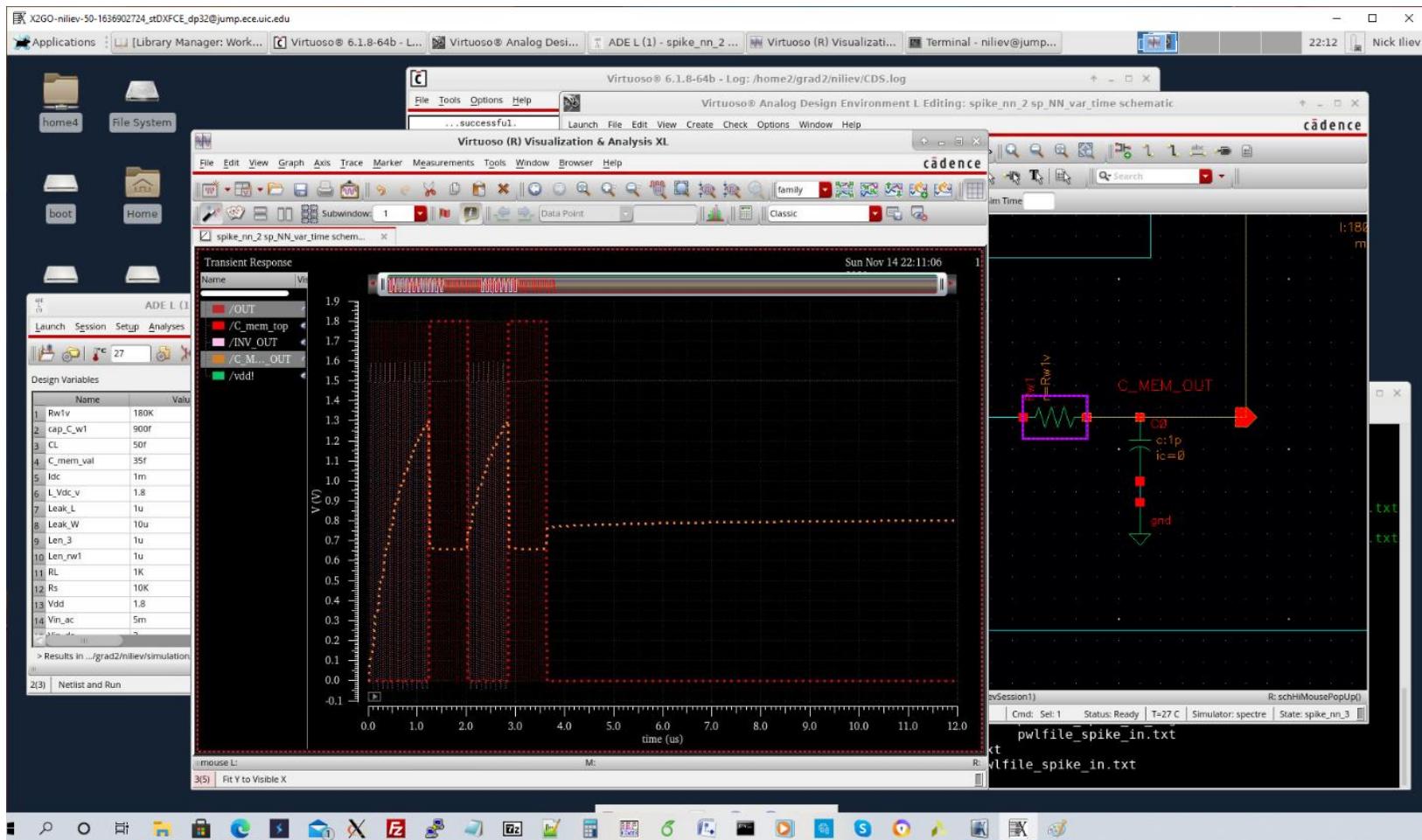
For example, with $R_{w1} = 50$ KOhms, (C_0 (C_{MEM_OUT}) is always 1pF) the following 5 spikes are produced at the output ; the C_{MEM_OUT} rising waveform is shown as well, 5 rising waveforms are generated for 5 output spikes :



For $R_{W1} = 90 \text{ KOhms}$, ($C_0 (C_{MEM_OUT})$ is always 1pF) the following 3 spikes are produced at the output ; the C_{MEM_OUT} rising waveform is shown as well, 3 rising waveforms are generated for 3 output spikes :



For $R_{W1} = 180 \text{ KOhms}$, ($C_0 (C_{MEM_OUT})$ is always 1pF) the following 2 spikes are produced at the output ; the C_{MEM_OUT} rising waveform is shown as well, 2 rising waveforms are generated for 2 output spikes :



My idea : the charge-pump based synapse for charging C mem takes 5 devices, instead of 24 devices in the above Reference Asghar_2021. The 4-bit positive weight for the synapse is responsible for switching in the desired R_{w1} resistance. With 4 bits, a total of 16 resistance values can be implemented.

For example

0000 = 10 KOhms

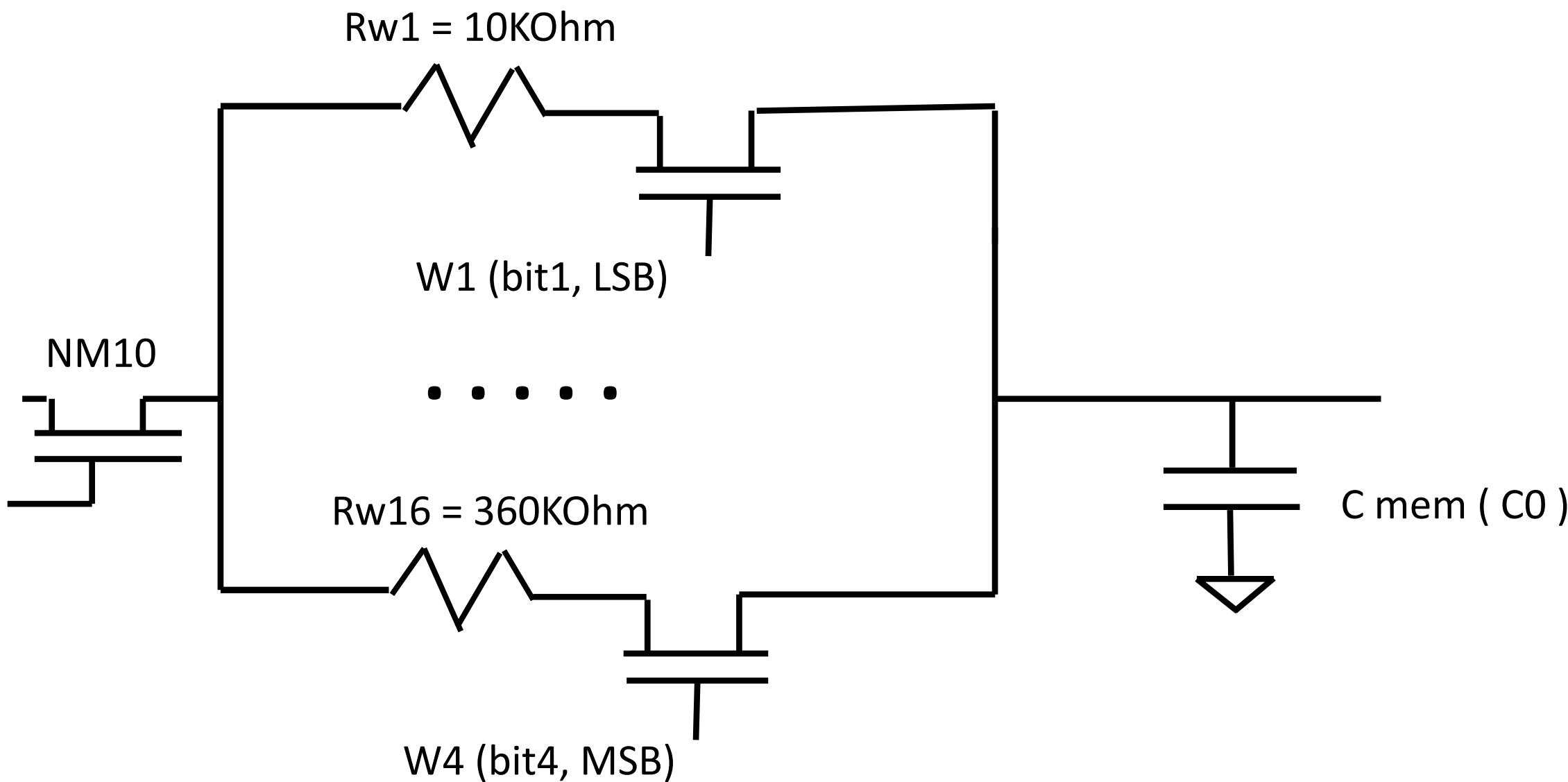
0001 = 20 KOhms

.....

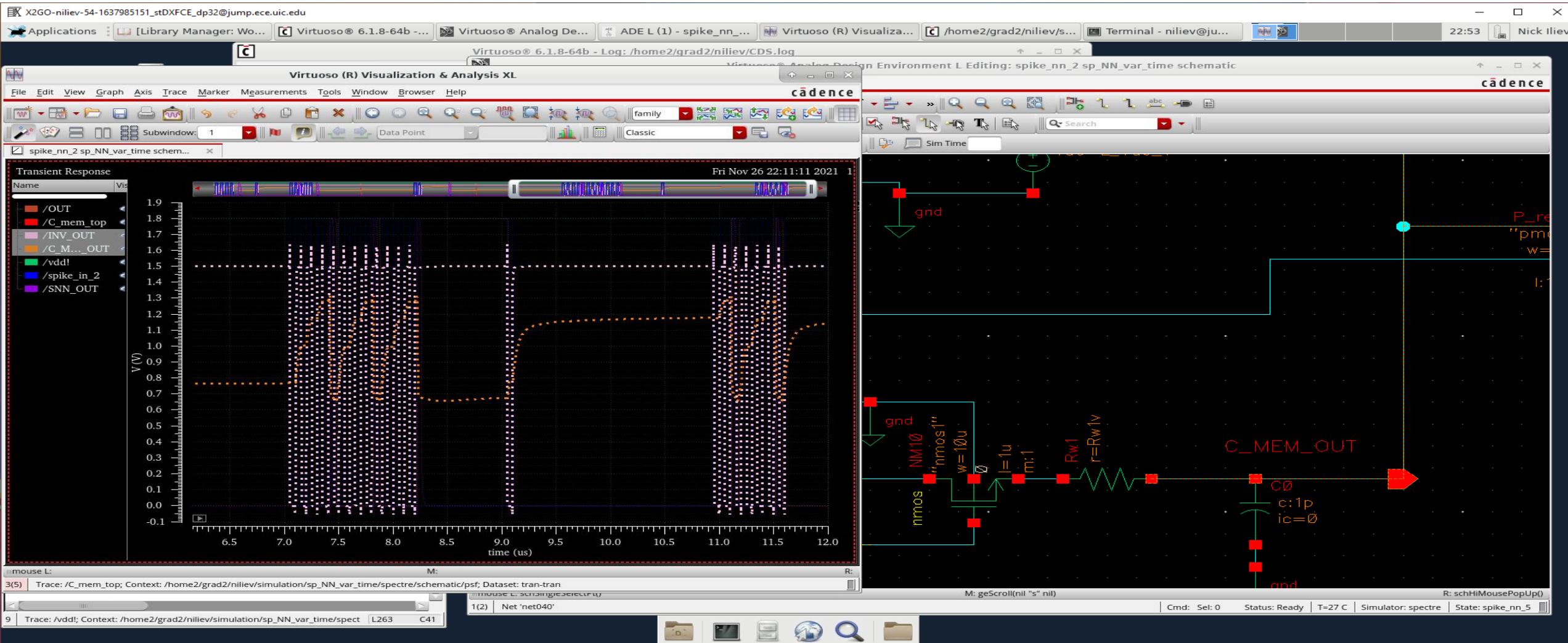
1111 = 360 KOhms

Optimal resistance values (4-bit weight) for each synapse can be derived during off-line training. 16 devices will be required to select one (or several) R_{w1} values (from 16 possible) in a synapse; this is then a total of $5+16 = 21$ devices for my new charge-pump based synapse vs $16+24=40$ devices for the above Reference.

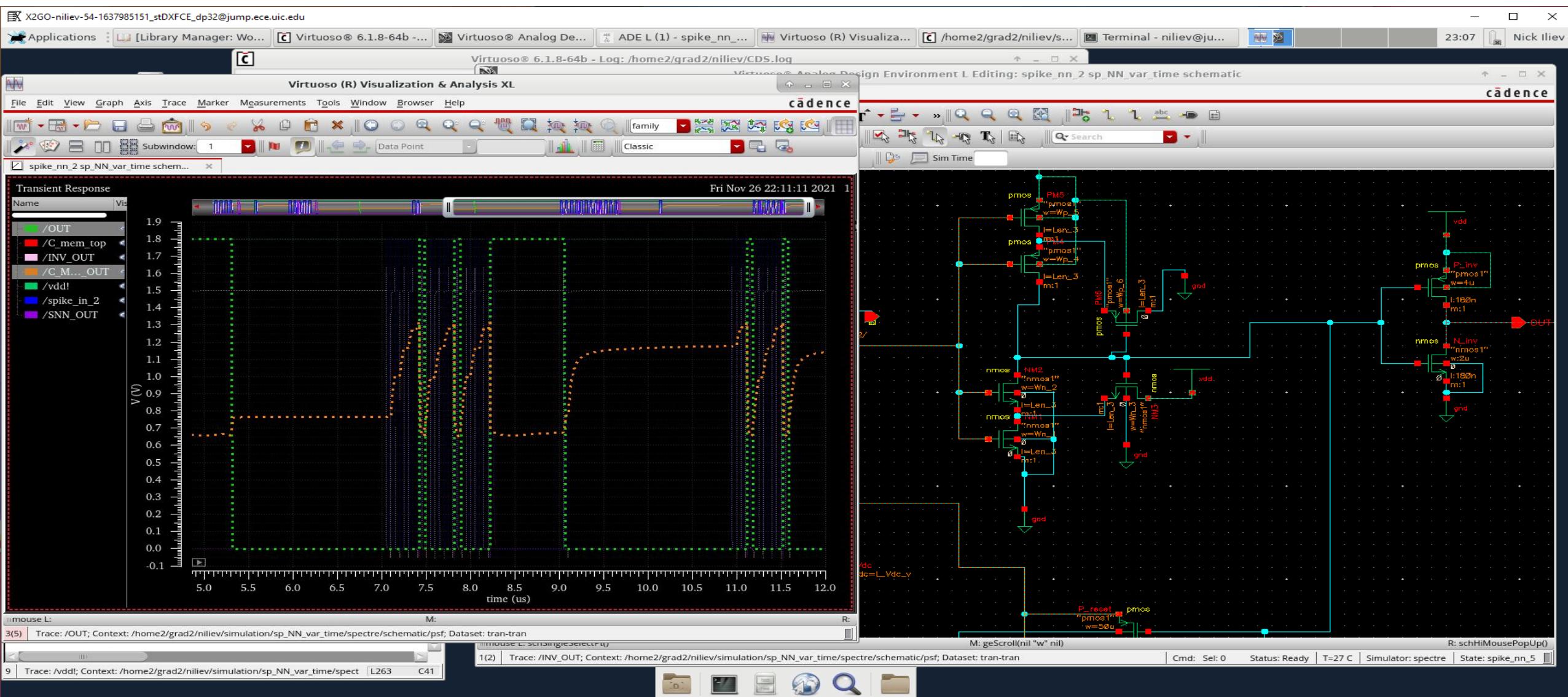
My idea : proposed 4-bit weight switch for Rw_1 (or multiple Rw 's) resistor selection for C mem



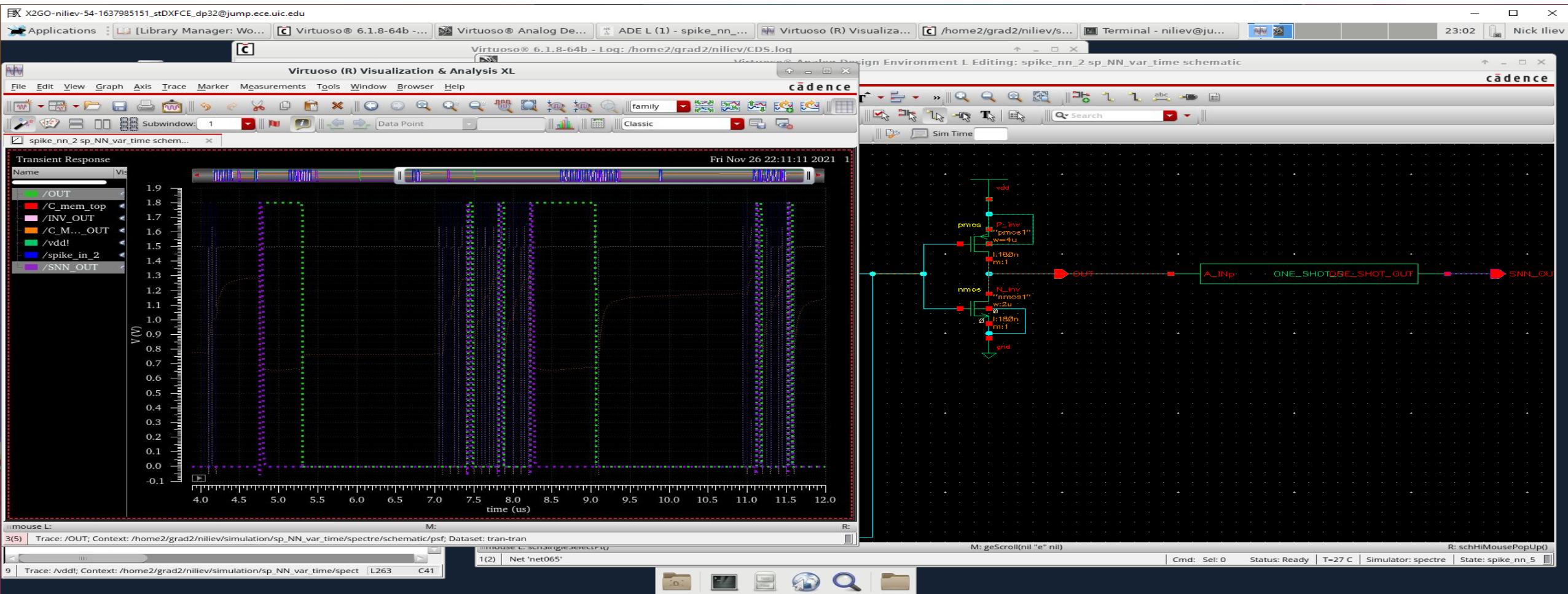
My idea : When input spikes are Irregular (non-periodic), the Charge pump capacitors C1 and C mem may charge at different rates resulting in different RC time constants for the $Rw_1 - C$ mem RC circuit. This is shown in the next plot, white is the irregular input spikes ; brown is the resulting C mem voltage



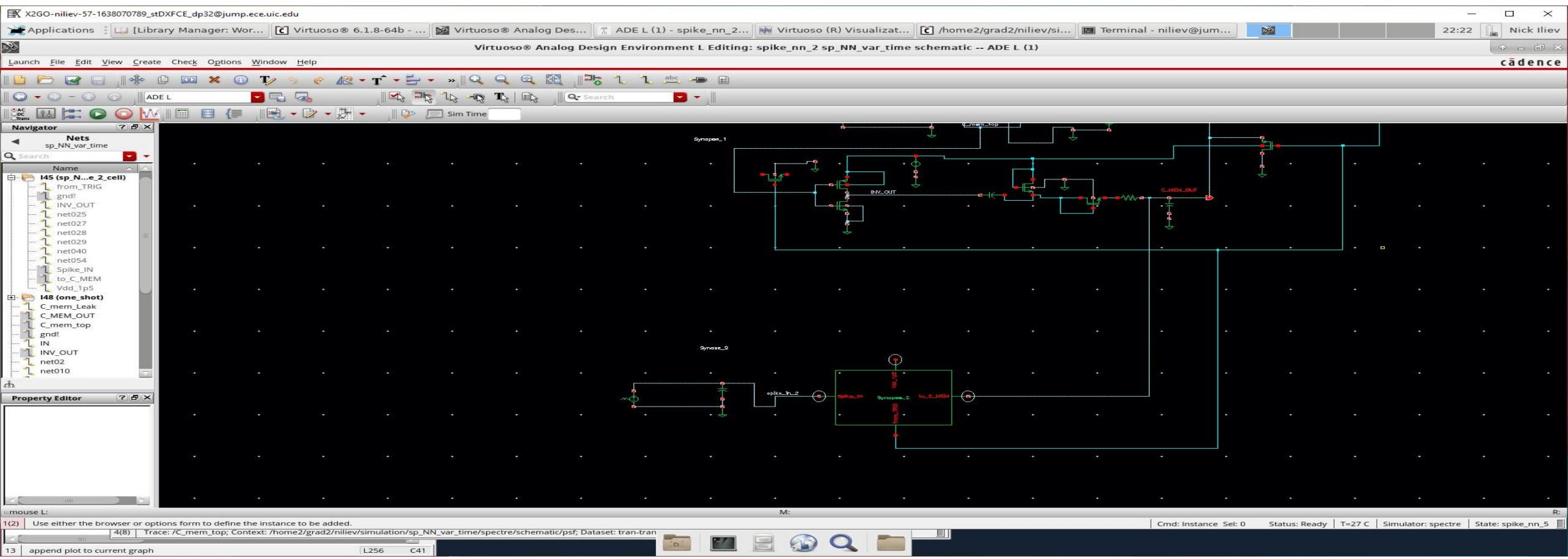
My idea : Irregular input spikes - the Schmitt trigger output pulses will therefore have variable widths (from 10s of nsec to 10s of usec); the trace below shows a case of this, green are the output pulses



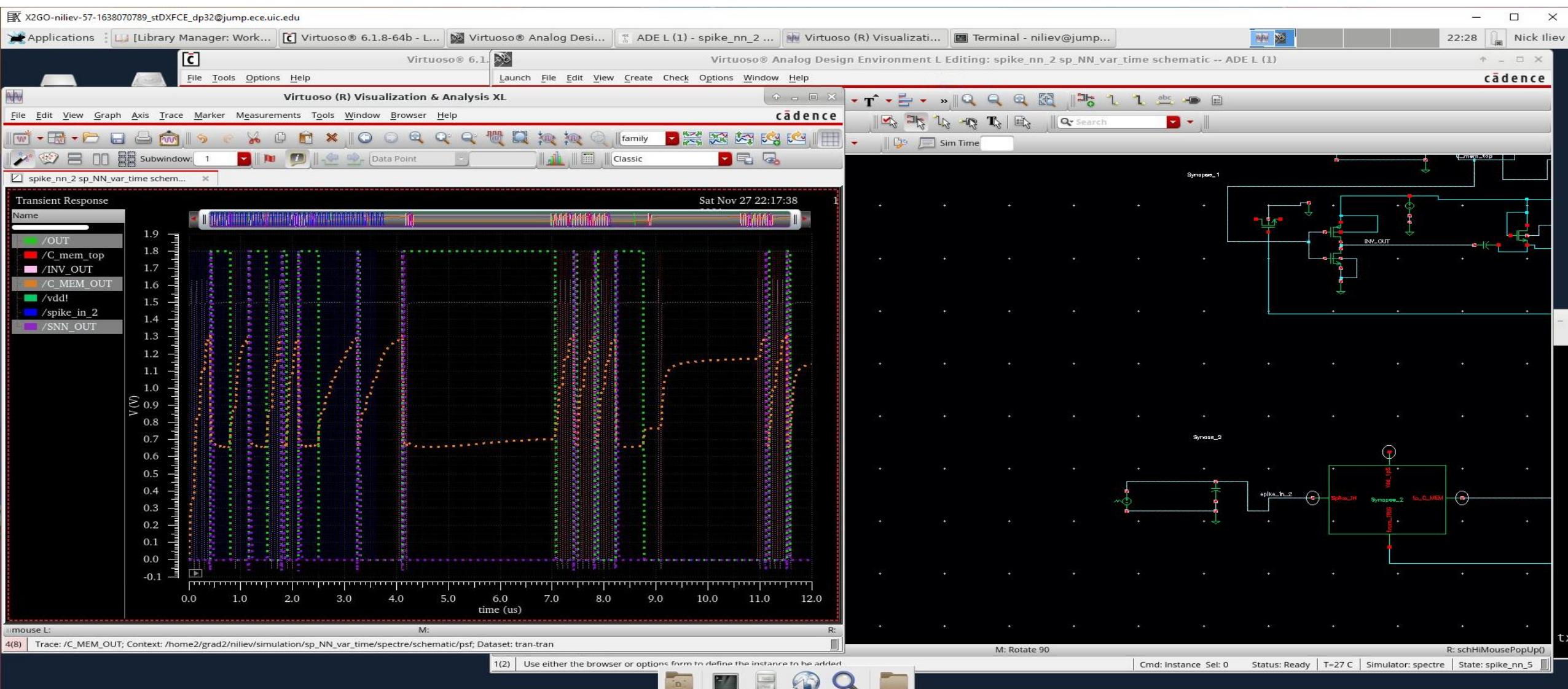
My idea : Irregular input spikes – as a solution to Schmitt trigger's variable pulses, I propose to add a One-Shot circuit, driven by the Schmitt trigger output; the One-Shot output pulses have a rising edge _aligned_ with the Schmitt trigger output rising edge, and have a constant width of 50 nsec (or other desirable width); the trace below shows the One-Shot pulses in purple, aligned with the green Schmitt trigger pulse rising edges.



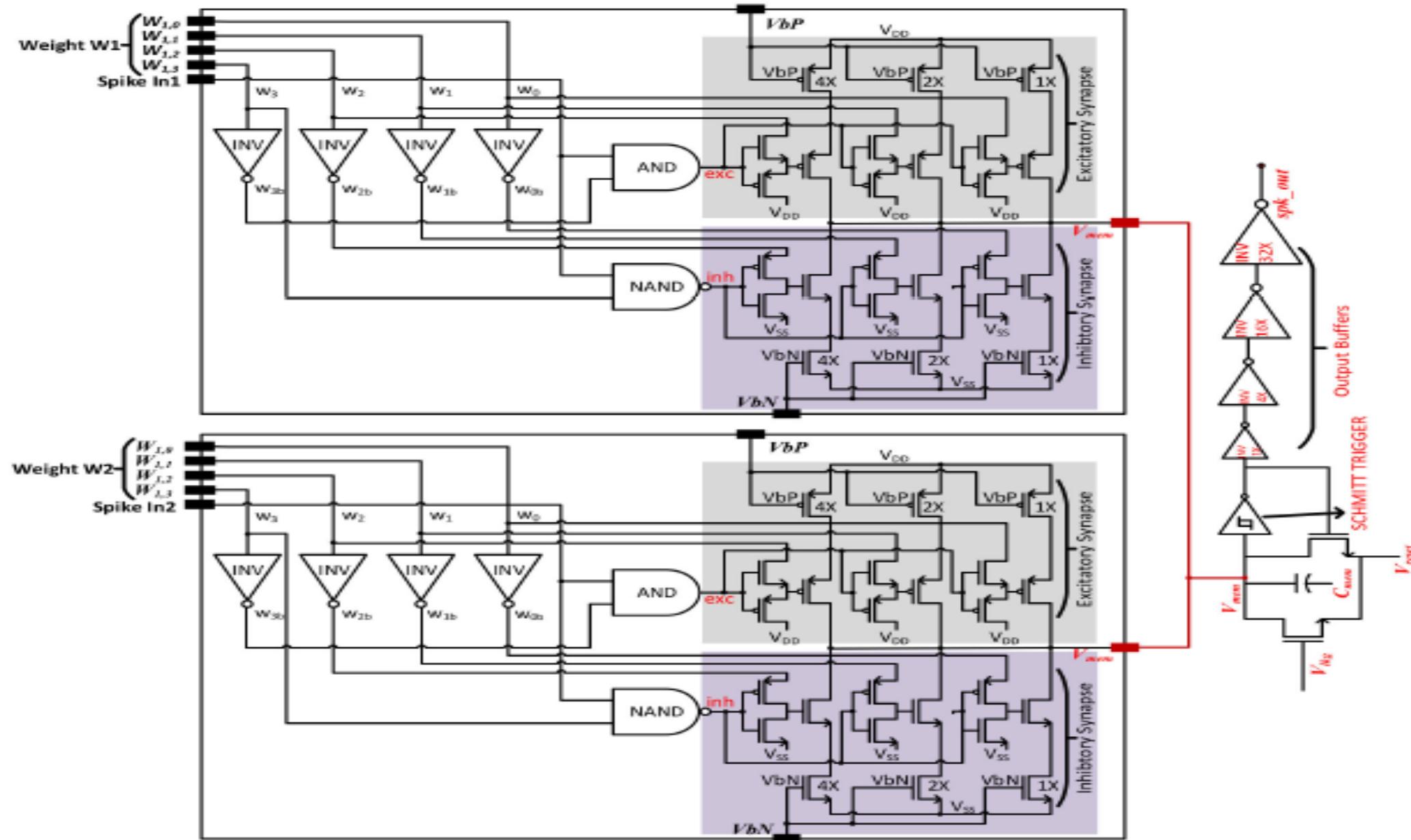
My idea : Scaling to 2 (or more) Synapses; the symbol Synapse_2 below, is identical to the proposed charge-pump based circuit for Synapse_1. Synapse_2 has its own Spike_IN input port, it's own output port to_C_MEM to contribute to the charging of C mem. Synapse_2 also has its own programmable resistor Rw1v_syn2 for its own RC constant with Cmem. It also has a feedback connection to the Schmitt trigger output, for gating Spike_IN (not used).



My idea : The above 2 Synapse Spiking Neuron circuit has been simulated and outputs 50 nsec pulses as expected; Synapse_1 has $R_w1=50\text{KOhms}$, Synapse_2 has $R_w1_v_{syn2}=180\text{KOhms}$. Identical charge pumps are used in both synapses. 42 devices for 2 synapses.



In comparison, reference Asghar_2021, uses 80 devices for 2 synapses :



Up-scaling to an MNIST 4-layer (81-30-20-10) fully-connected classifier :

total number of synapses is :

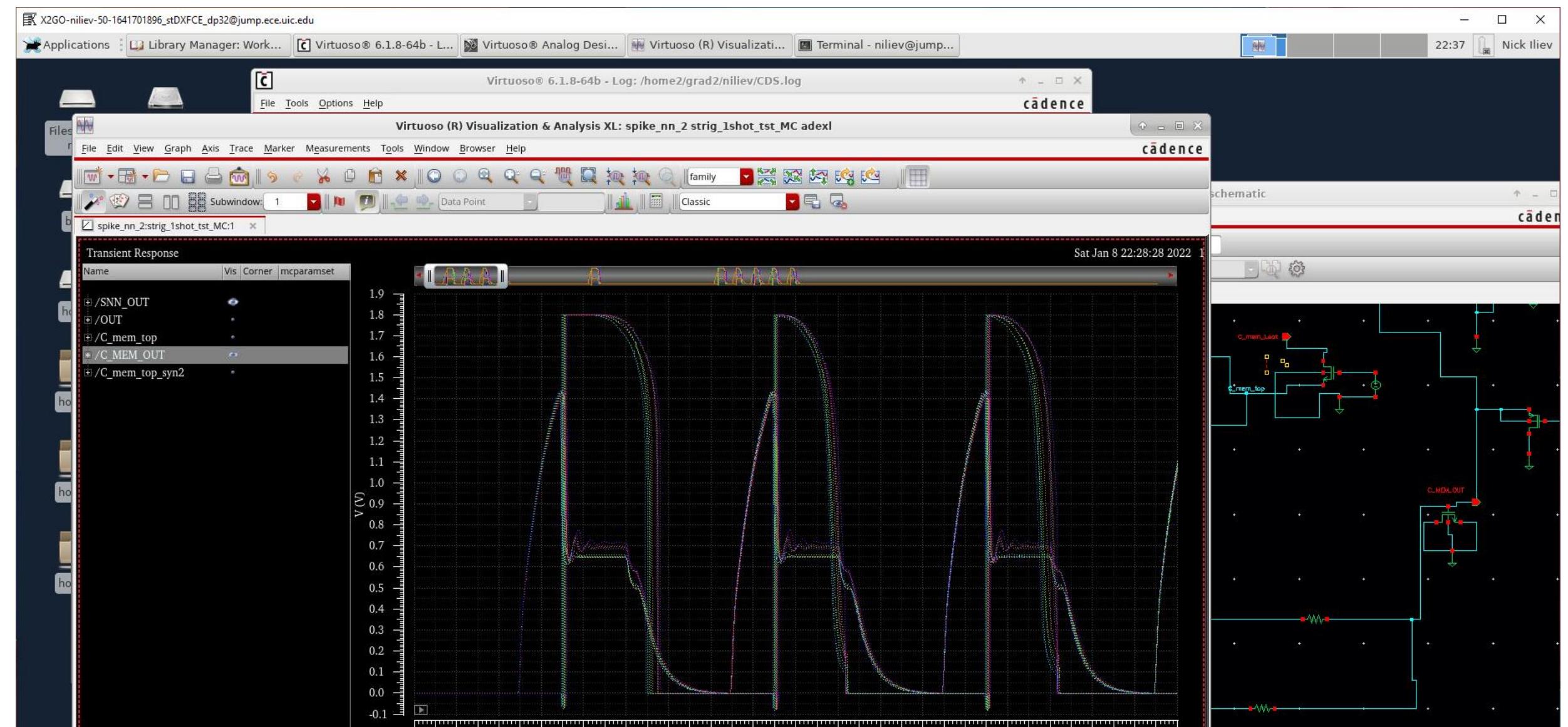
$$81 + 81*30 + 30*20 + 20*10 = 3,311 ;$$

Total devices for all synapses : $3,311*40 = 132,440$ (reference Asghar_2021)

Total devices for all synapses : $3,311*21 = 69,531$ (my idea, charge-pump based synapse + positive-only weights) : ***a 47.5 % reduction in device count from the original 132,440***

Monte Carlo (PVT corners) simulations – 1

string_1shot_ADE_XL_SNN_OUT_12corners_good_spikes_2_Synapses_Rw1v_10K_Rw2v_50K_C_MEM_OUT



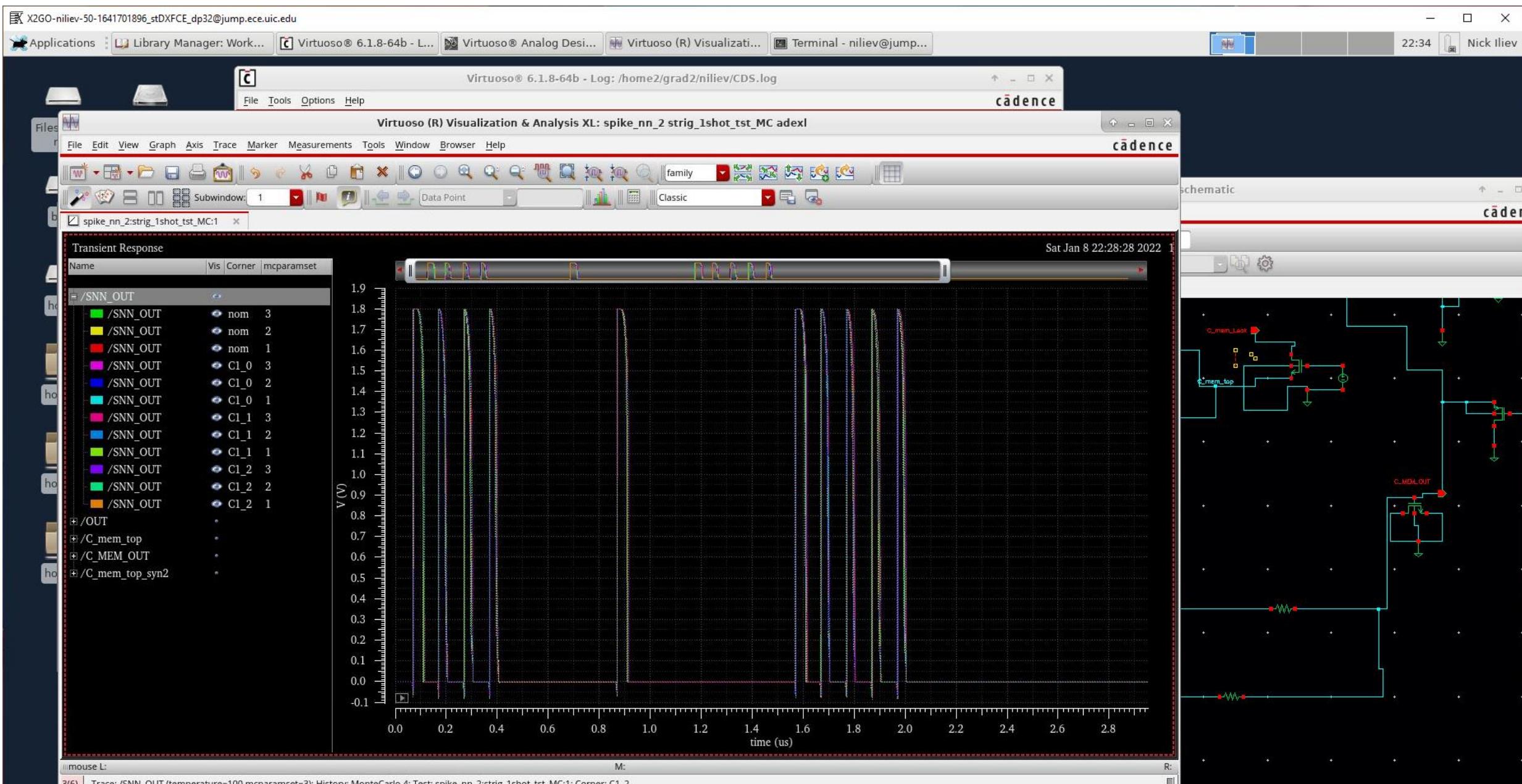
Monte Carlo (PVT corners) simulations – 2

string_1shot_ADE_XL_SNN_OUT_12corners_good_spikes_2_Synapses_Rw1v_10K_Rw2v_50K.png_zoom



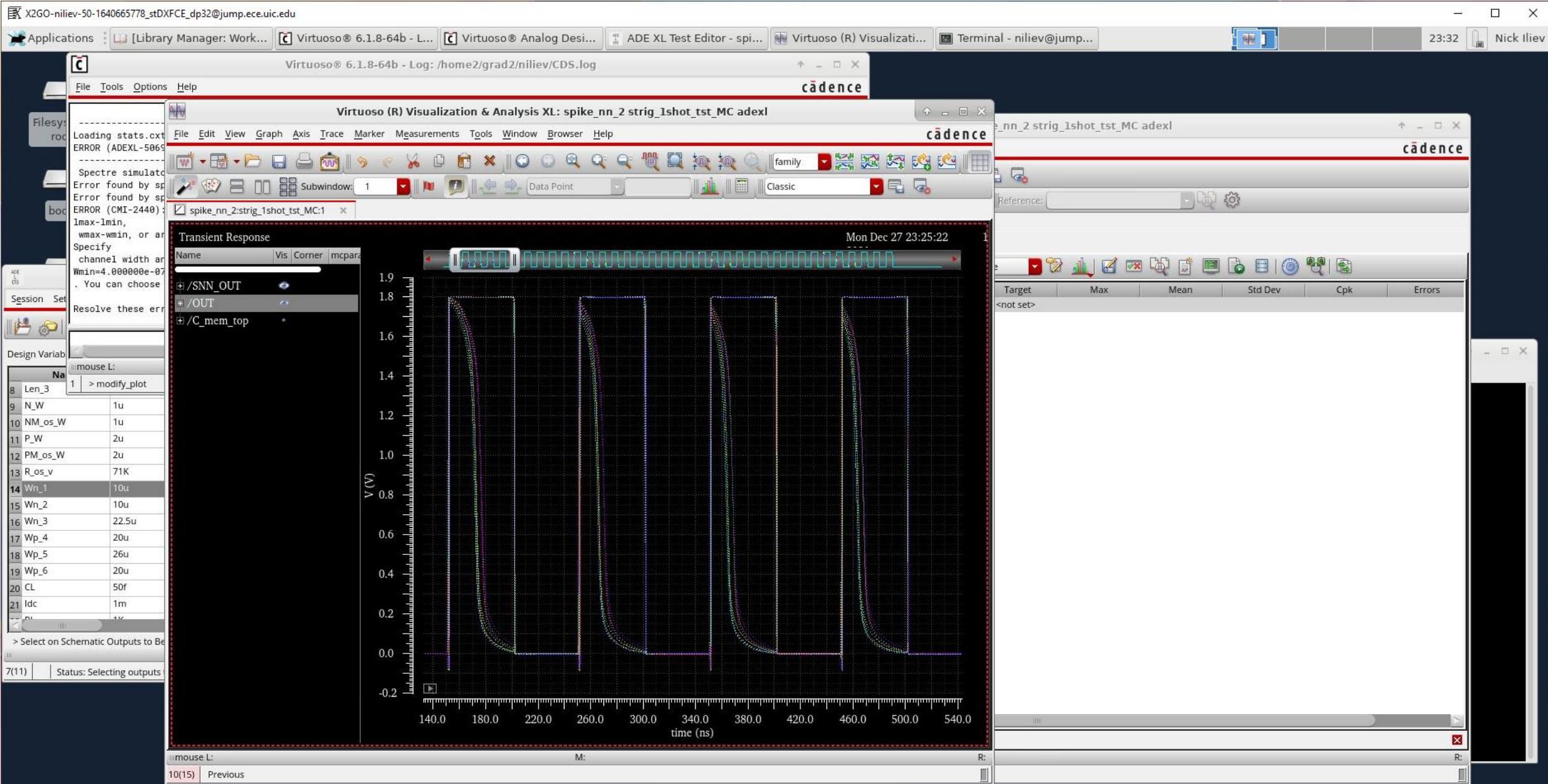
Monte Carlo (PVT corners) simulations – 3

string_1shot_ADE_XL_SNN_OUT_12corners_good_spikes_2_Synapses_Rw1v_10K_Rw2v_50K



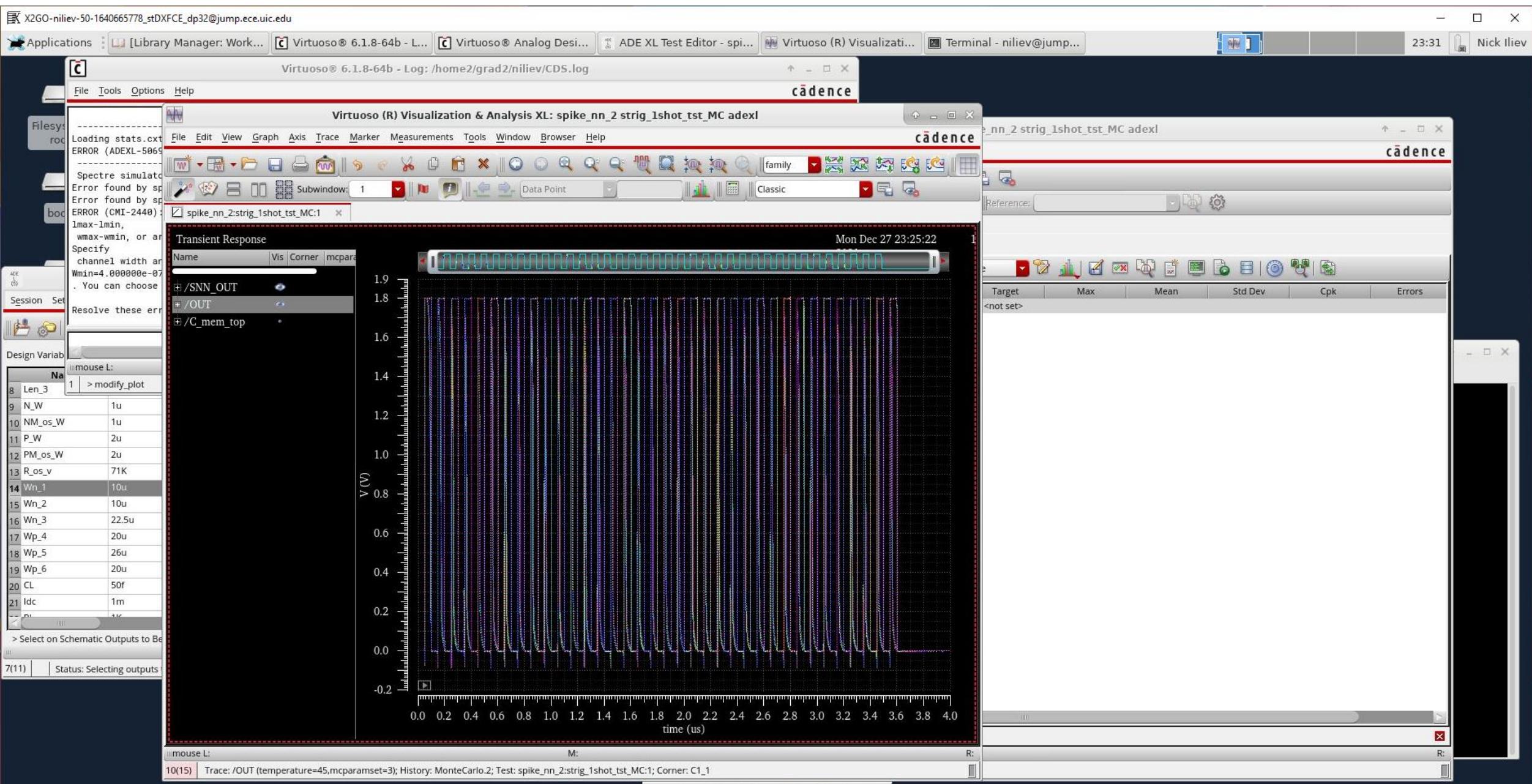
Monte Carlo (PVT corners) simulations – 4

strig_1shot_ADE_XL_SNN_OUT_12corners_zoom_in_spike_width_approx_15nsec



Monte Carlo (PVT corners) simulations – 5

strig_1shot_ADE_XL_SNN_OUT_12corners_good_spikes_in_4us



Conclusion :

A novel charge-pump based synapse circuit (for a spiking-neuron) has been proposed, taking 21 devices to implement (5 for charge-pump + input gating, 16 for selecting desired R_w value and RC time constant using 4-bit positive weights). This is a considerable saving from the recent Asghar_2021 synapse circuit, which takes 40 devices per synapse, and uses 4-bit positive and negative weights. Initial simulations of the proposed 2 synapse SNN model with regular or irregular spiking inputs looks promising.

Future work :

A validation of the proposed synapse + spiking-neuron circuit has to be implemented using an MNIST MLP 4-layer (81-30-20-10) model, as described in Asghar_2021. The 4-bit positive synapse weights in my proposed model have to be learned with off-line training and then applied during the test inferencing phase.

Future work : Comparison with other Spiking Neuron MNIST classifier based on all-positive weights, as reported in

“A Minimal Spiking Neural Network to Rapidly Train and Classify Handwritten Digits in Binary and 10-Digit Tasks”, by A. Tavanaei 2015

This work uses a 3-layer MLP for MNIST, an additional inhibitory neuron, and all-positive weights base on STDP training

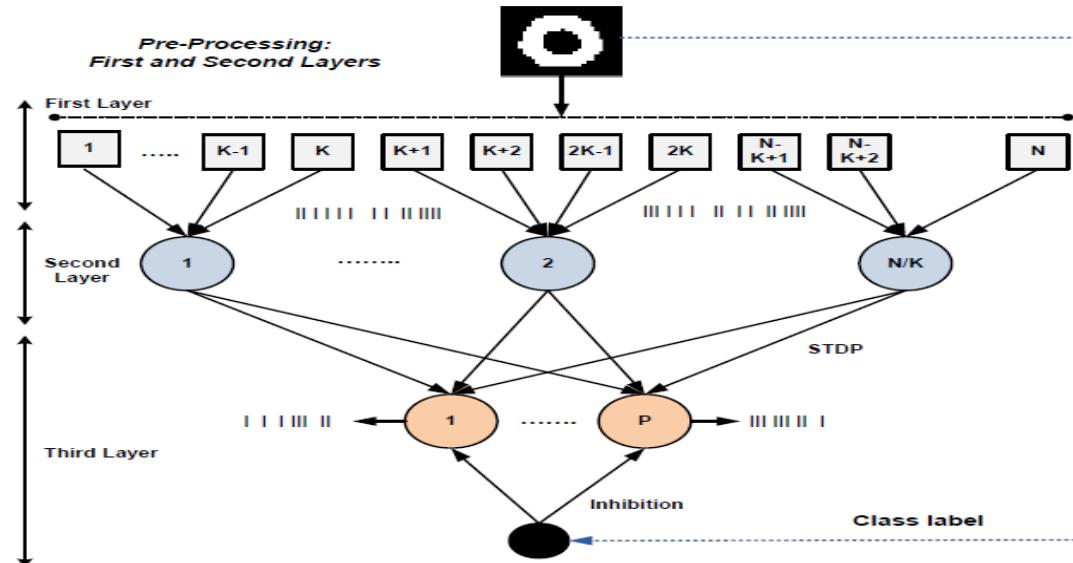


Fig. 2. Supervised SNN architecture containing spike transcription layer, spike train segmentation, STDP learning, output pattern firing, and inhibitory neuron. N : number of rows. K : number of adjacent rows connected to one neuron. P : number of classes. Black circle inhibits all output neurons except the one designated by the class label

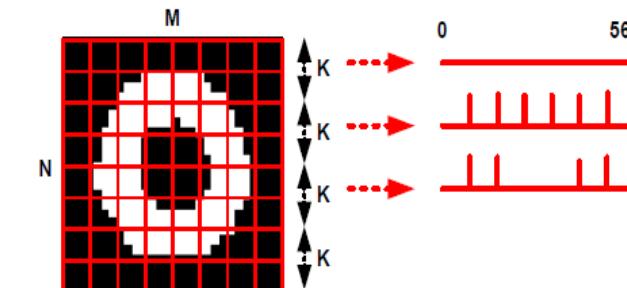


Fig. 3. A digit image with $N \times M$ pixels divided into N/K segments, consisting of k rows per segment

Future work : Comparison with other Spiking Neuron MNIST classifier based on all-positive weights, as in Tavanaei_2015

TABLE V. SYNAPTIC WEIGHTS OF 14 SYNAPSES OF 10 DIGITS 0-9. THE FIRST TWO SYNAPTIC WEIGHTS AND THE LAST ONE ARE SMALLER THAN OTHER SYNAPSES BECAUSE THEY MOSTLY CONVEY BACKGROUND INFORMATION. THUS, RELATIVELY MUCH WEAK SYNAPSES PROVIDE A FAST METHOD OF BACKGROUND ELIMINATION IN THE ROW ORDER

Syn \ Digit	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0.008	0.008	0.041	0.028	0.033	0.070	0.109	0.137	0.158	0.155	0.132	0.105	0.008	0.008
2	0.009	0.010	0.062	0.147	0.100	0.034	0.026	0.051	0.147	0.190	0.142	0.063	0.009	0.009
3	0.009	0.009	0.130	0.190	0.083	0.045	0.076	0.062	0.018	0.026	0.122	0.186	0.033	0.010
4	0.010	0.010	0.010	0.028	0.082	0.138	0.187	0.208	0.159	0.055	0.036	0.038	0.029	0.010
5	0.008	0.008	0.020	0.065	0.092	0.117	0.137	0.152	0.051	0.105	0.144	0.088	0.008	0.008
6	0.009	0.033	0.051	0.043	0.051	0.071	0.132	0.165	0.171	0.151	0.097	0.011	0.009	0.009
7	0.010	0.010	0.010	0.079	0.199	0.114	0.069	0.045	0.051	0.044	0.066	0.091	0.151	0.063
8	0.006	0.006	0.032	0.084	0.114	0.127	0.113	0.055	0.089	0.113	0.113	0.112	0.029	0.006
9	0.008	0.008	0.008	0.059	0.133	0.154	0.152	0.155	0.052	0.027	0.041	0.072	0.124	0.008
0	0.008	0.008	0.021	0.047	0.083	0.092	0.122	0.126	0.114	0.146	0.156	0.061	0.008	0.008

Detailed Monte-Carlo simulation analysis will be done with the proposed SNN model to characterize its performance across process+temperature variations.