SOI Implementation of Spiking Equilibrium Propagation for Real-Time Learning

Amruta Parulekar, Anubhav Bhatla, Tanmay Joshi
(20d070009) (200070008) (200070027)

I. Introduction

In deep learning tasks, input data is first processed through the artificial neural network (ANN) in a feed-forward pass, resulting in an initial prediction, which is compared to the ground truth to calculate a loss. Backpropagation is a training algorithm for ANNs in which the gradients of the loss are computed with respect to the network weights and biases while moving backwards through the network, layer by layer. The gradients indicate the direction and magnitude of changes needed in the network's parameters to reduce the loss. These changes are made using optimizing algorithms, based on a learning rate. This is repeated till the loss is sufficiently minimized. The steps in backpropagation are computationally intensive and unsuitable for real-time applications that require low latency and fast data processing. Backpropagation also relies on labeled training data to compute gradients, making it less suitable for unsupervised or self-supervised learning scenarios.

Equilibrium Propagation (EP) [1] is an alternative training algorithm that addresses these limitations of backpropagation. It treats ANNs as dynamical systems that evolve over time while backpropagation treats each training example in isolation. EP aims to reduce the need for explicit supervision and be more adaptable to unsupervised learning tasks. It improves the generalization performance of neural networks by focusing on energy-based objectives that encourage the model to learn relevant and robust features from the data. EP captures the principles of neural information processing in the brain by mimicking the way neurons reach equilibrium states during learning. It leverages energy functions for learning and inference in neural networks by finding stable equilibrium states of minimum energy that best match the input data by iteratively updating the network's activations. Feedback connections are used to transmit information about the difference between the current state and the target state like recurrent connections in biological neural networks.

In this project, we have used 45nm CMOS Technology to implement spiking equilibrium propagation. Spiking Equilibrium Propagation (SEP) is an extension of the Equilibrium Propagation (EP) learning algorithm which combines the principles of EP with spiking neural network (SNN) architectures, which are biologically inspired models of neural computation. SNNs use discrete, spike-based communication between neurons, resembling the way neurons in the brain transmit information. SEP retains the energy-based modeling concept found in EP, using an energy function to measure the compatibility between the current state and the input data, to find states of minimum energy that best represent the data. Like EP, SEP treats the neural network as a dynamical system and is particularly well-suited for tasks where precise timing and spike-based encoding are important.

II. Background

Energy-based models (EBMs) are a class of machine learning models characterized by the use of energy functions to quantify the compatibility or goodness of fit between data and model configurations.The energy function assigns an energy score to each possible configuration. Lower energy configurations are favorable and correspond to better representations of the data. During training, the energy function is learned or fine-tuned to minimize the difference between the energy of observed data and that of generated or model data. Inference in EBMs often involves finding the configuration that minimizes the energy, which can be done by gradient based methods. EBMs are particularly useful in situations where capturing dependencies between variables is crucial.

CMOS is a widely used technology for the design and fabrication of integrated circuits and digital electronic devices. CMOS technology is known for its energy efficiency, low power consumption, and high integration capabilities, making it suitable for a wide range of applications. A CMOS implementation refers to the realization of a digital or analog electronic system using CMOS technology. This involves designing and fabricating the circuit or system using CMOS transistors and components.

Real-time learning refers to the ability of a system, to continuously adapt and update its knowledge, model, understanding or behavior in a near-instantaneous fashion, based on incoming data as it becomes available. This is in contrast to offline or batch learning, where models are trained on static datasets and updated less frequently. Real-time learning systems respond quickly to new information, which is crucial for applications where timely decisions are required. Real-time learning comes with challenges related to the need for efficient algorithms, low-latency processing, and handling noisy or streaming data.

III. Objective

Our objective is to develop a hardware-based implementation of the Equilibrium Propagation (EP) learning algorithm in the form of CMOS circuits, specifically designed for real-time learning.

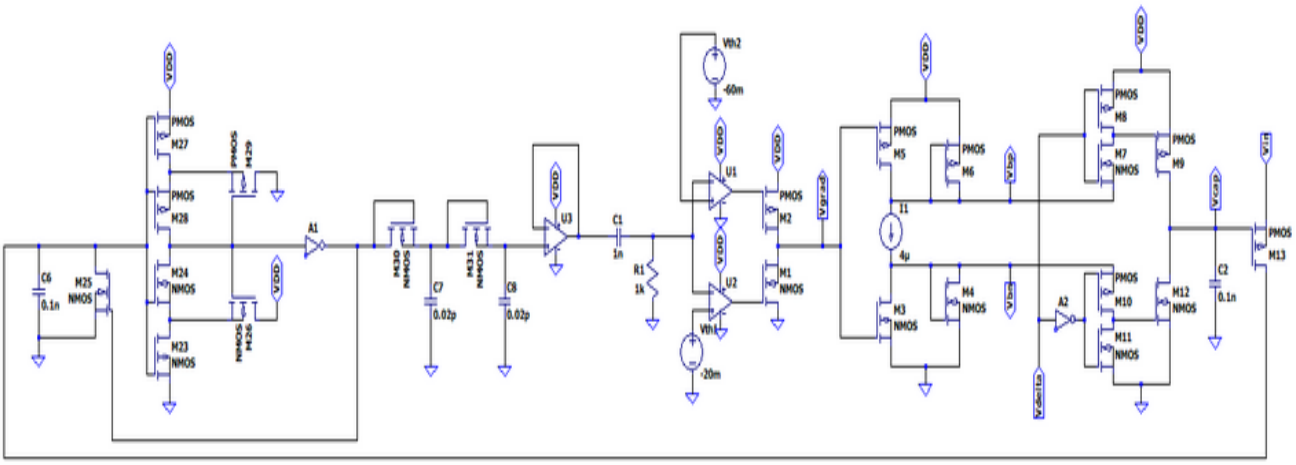

Fig. 1. Schematic design for spiking equilibrium propagation in 45nm CMOS technology.

We hypothesize that

- Using CMOS technology for the implementation of SEPs will provide higher energy efficiency and low power consumption as compared to conventional neural network hardware.
- The use of spiking neural networks will lead to a more biologically plausible learning system which will closely mimic the principles of neural information processing in the brain.
- CMOS-based SEP will be capable of real-time learning, with the system adapting rapidly to changing data inputs, due to its low latency and scalability.

IV. Experiment

Figure 1 describes the schematic for modelling spiking equilibrium propagation between a pre-synaptic neuron and a single post-synaptic neuron. We shall now study and implement this circuit by dividing it into five blocks described as follow:

A. Integrate-and-fire circuit

The integrate-and-fire (IFC) circuit mimics the spiking behaviour of a LIF post-synaptic neuron. The spikes are generated using a Schmitt trigger, which provides two different threshold voltage levels for the rising and falling edge. We have implemented the Schmitt trigger using 45nm CMOS technology. Once the input voltage reaches a threshold, the Schmitt trigger switches high, activating an nMOS to discharge the capacitor. The Schmitt trigger output remains high until the capacitor voltage goes below a certain value. This rapid switching generates spikes which imitate post-synaptic neuron spiking. Refer Figure 2 for the schematic of the integrate-and-fire circuit.

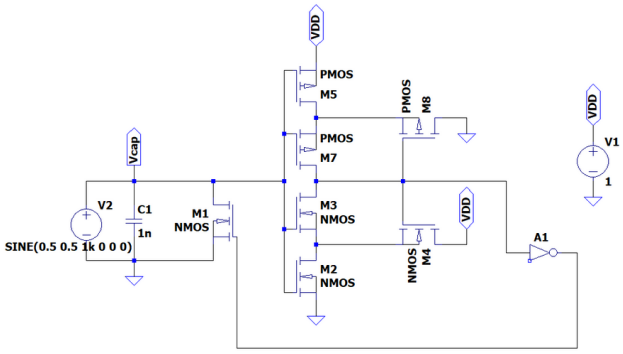

Fig. 2. IFC using a Schmitt Trigger.

B. Two-stage Low-pass Filter

The output spikes from the IFC stage need to be leaky-integrated with a two-stage low-pass filter. We use two diode-connected nMOS devices in series, with capacitors from each diode's output to ground. Using diodes ensures that the capacitor voltage increases rapidly with each spike but decays slowly between spikes. An Opamp-based buffer separates the output from the next stage. The circuit schematic is shown in Figure 3.

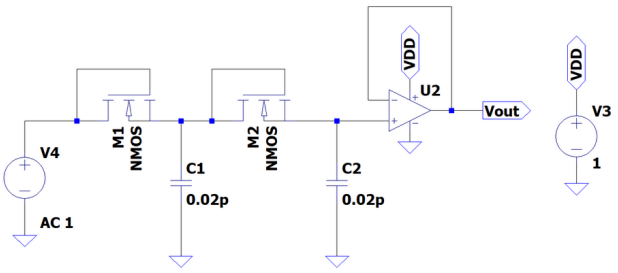

Fig. 3. Two-stage Low-pass Filter.

C. Derivative circuit with comparators

We build a derivative circuit using an RC circuit. Figure 4 shows the schematic used for implementing the derivative circuit. The output signal is proportional to the derivative of the input signal. This derivative signal is fed to two Opamp-based comparators with thresholds of +20mV and -20mV respectively. The output of the two comparators is combined to generate the V_{grad} signal.

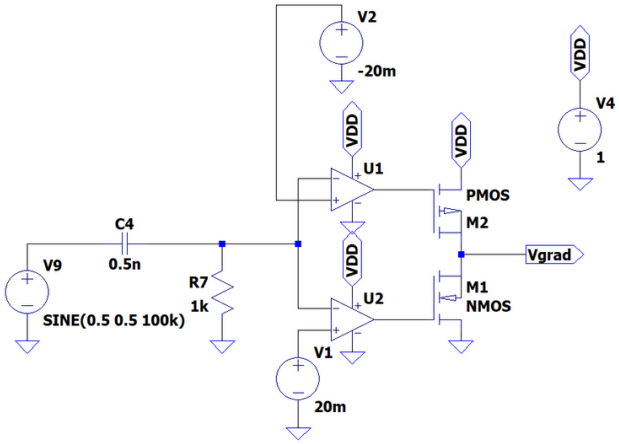

Fig. 4. Derivative circuit with comparators.

D. Bias circuit

The bias circuit takes the gradient of the spike rate as an input and outputs the bias voltages, V_{bp} and V_{bn}, which are then fed to the synapse circuit. When a low V_{grad} is asserted, M4 pulls V_{bp} to VDD. M1 is off, but the current source through M2 pulls the gate/drain voltage (V_{bn}) over GND. The opposite is true for a high V_{grad}, with the current through M3 pulling V_{bp} below VDD, and M1 pulling V_{bn} to GND. Figure 5 shows the schemtic used for the bias circuit.

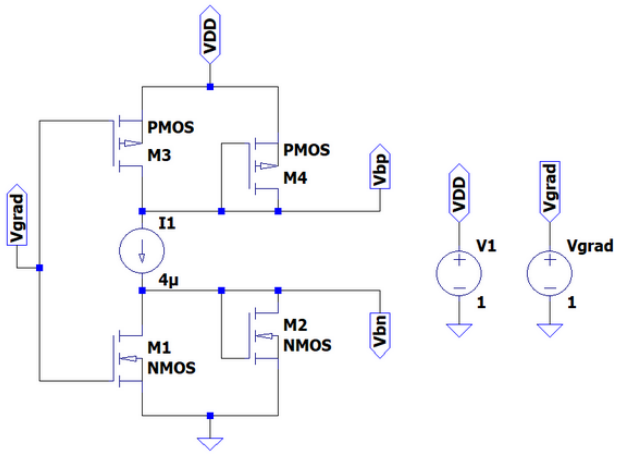

Fig. 5. Bias circuit.

E. Synapse circuit

The synapse circuit takes in inputs from both the pre-synaptic neuron (V_{IN}) and the post-synaptic neuron (V_{bp}, V_{bn}). Finally, it outputs I_{OUT} which is fed back to the post-synaptic IFC. The update signal ($V_{\Delta W}$) is also driven by the output spikes of the post-synaptic neuron. The capacitor (refer Figure 6) gets charged when V_{bp} is VDD and $V_{bn} >$ GND, and when $V_{\Delta W}$ spikes. When V_{bn} is GND and $V_{bp} <$ VDD, the pMOS conducts and the capacitor discharges. When V_{bp} is VDD and V_{bn} is at GND, neither the pMOS nor the nMOS conducts and the capacitor voltage remains steady.

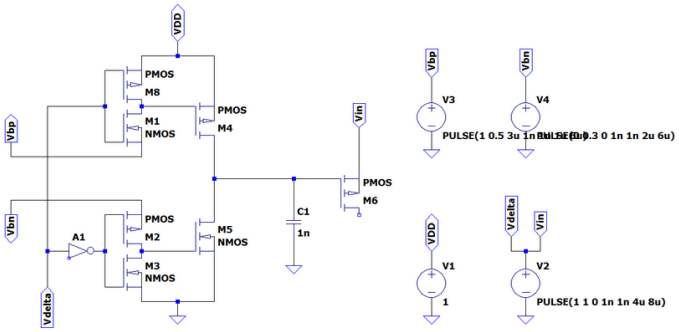

Fig. 6. Synapse circuit.

V. Results

We separately simulate all the five blocks described in the previous section. Figure 7 shows the output of the IFC for a sinusoidal input. We observe that when the input voltage goes above a certain threshold ($\approx 0.6V$), the Schmitt trigger output voltage is triggered and get VDD at the output of the inverter. Similarly, once the input voltage goes below the threshold voltage of $\approx 0.4V$, the Schmitt trigger output voltage gets triggered and we see GND at the inverter output.

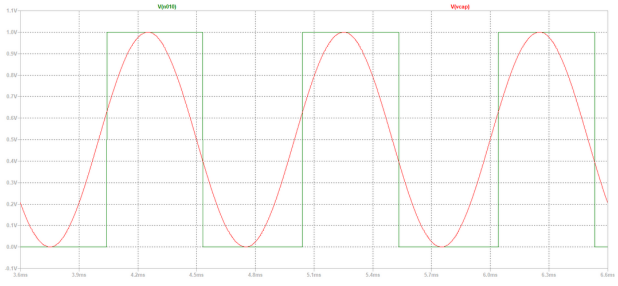

Fig. 7. Output of the integrate-and-fire circuit for a sinusoidal input.

The bode plot shown in Figure 8 shows a 3-dB frequency of $\approx 500kHz$. To test the two-stage low-pass filter, we use a 100Hz sinusoidal input signal, with 100kHz noise added to it. In Figure 9 we observe a smooth signal at the output, as expected from the low-pass filter.

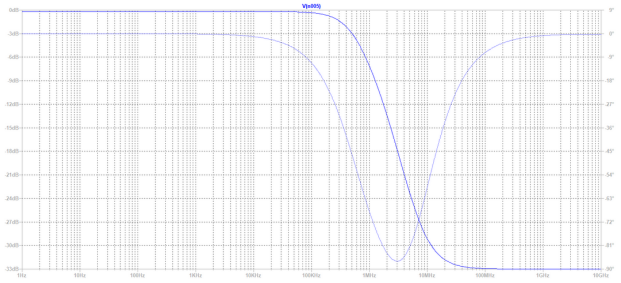

Fig. 8. Bode plot for the two-stage low-pass filter.

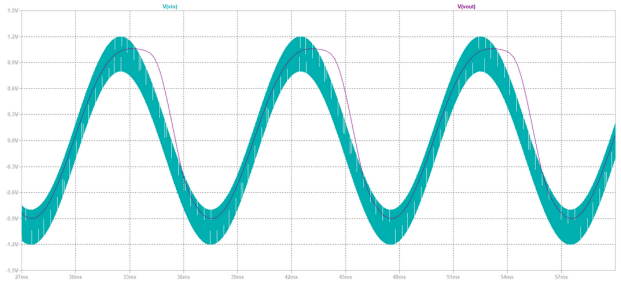

Fig. 9. Output of the two-stage low-pass filter for a noisy sinusoidal signal.

The bode plot for the derivative circuit shown in Figure 10 shows a 3-dB frequency of $\approx 300kHz$. A sinusoidal input is applied as input to the derivative circuit. The cosine waveform seen at the output of the derivative circuit is passed to the two comparators and then two comparator outputs are combined to generate the V_{grad} signal, as seen in Figure 11.

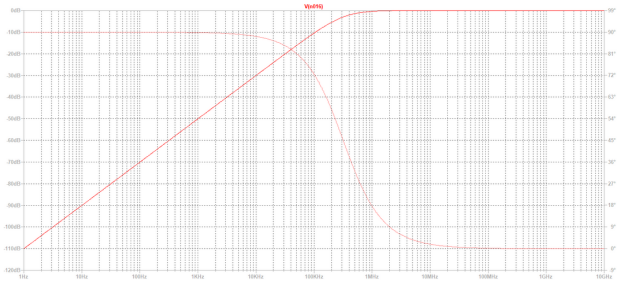

Fig. 10. Bode plot for the RC derivative circuit.

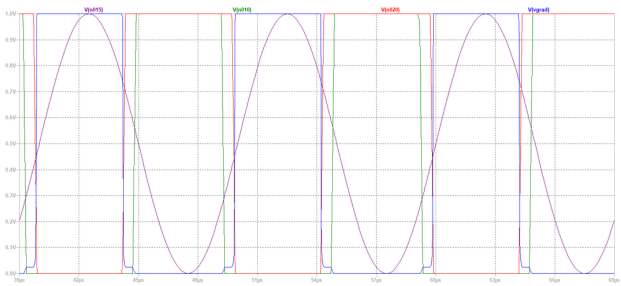

Fig. 11. Output of the derivative circuit and the comparators for a sinusoidal input.

The bias circuit is used to generate the V_{bp} and V_{bn} voltages as its outputs using the V_{grad} input voltage. As mentioned in [2], we require a reasonably high V_{bp} and V_{bn} for very small values of V_{grad}. As V_{grad} increases, V_{bn} reduces and saturates close to GND. Upon further increasing V_{grad}, V_{bp} also reduces and saturates around (VDD/2). Figure 12 shows the obtained input-output characteristics for the bias circuit.

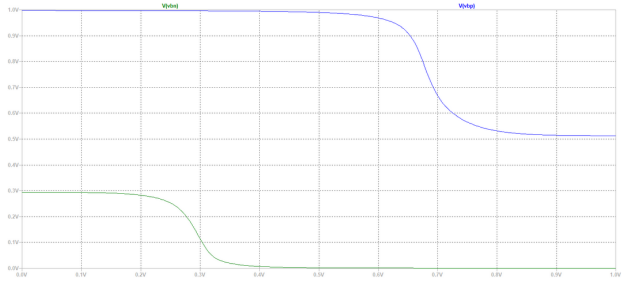

Fig. 12. DC analysis of the bias circuit as V_{grad} varies from 0 to 1V.

Figure 13 shows the change in capacitor voltage as we vary V_{bp} and V_{bn} and provide spikes at $V_{\Delta W}$. The characteristics are in line with our expectations described in the previous section.

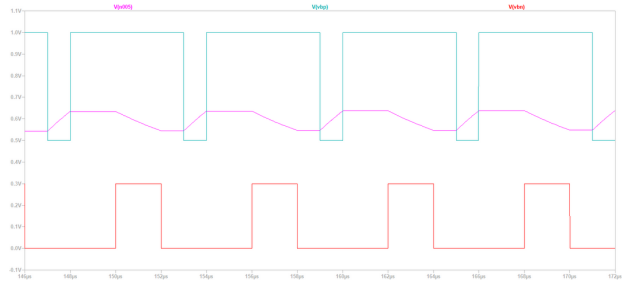

Fig. 13. Change in capacitor voltage for different values of V_{bp} and V_{bn}.

Figure 14 shows the post-synaptic neuron spikes generated at the output of the IFC circuit and its leaky integration after passing through the two-stage low-pass circuit. This is then passed through the derivative circuit to generate the output shown in Figure 15. We can also observe threshold voltages of $\pm 20mV$.

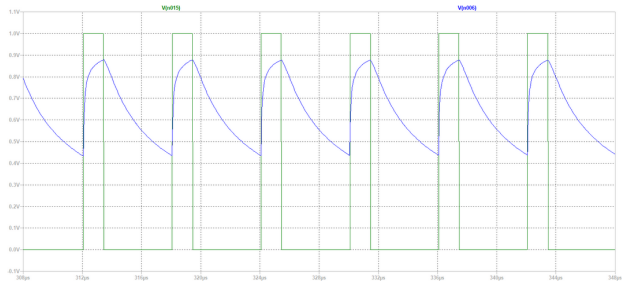

Fig. 14. Post-synaptic neuron spikes (green) and its leaky integration after passing through a two-stage low-pass filter (blue).

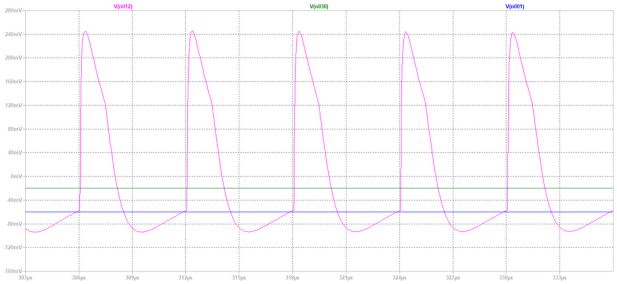

Fig. 15. The output of the differentiator circuit (pink) along with the threshold voltages (green and blue).

Figure 16 shows the final capacitor voltage obtained at the output of the synapse circuit. The bias voltages and the V_{in} pre-synaptic neuron spikes have also been show. We can observe that the capacitor voltage trend is in line with the previously observed trend for the synapse circuit.

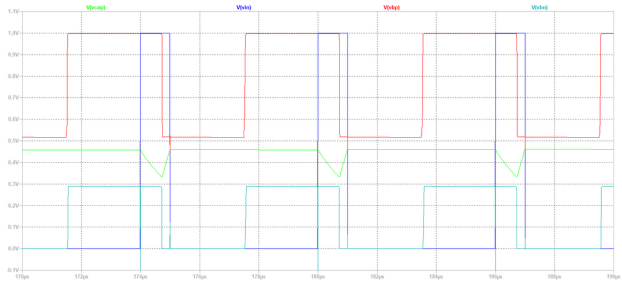

Fig. 16. The capacitor voltage (green) observed for a spiking pre-synaptic neuron input (blue). The corresponding bias voltages have also been shown.

VI. Conclusion

Thus, we have successfully simulated the neuron and synaptic circuit of the spike equilibrium propagation hardware using 45nm CMOS technology with total power consumption of $82.65\mu W$, out which only $8.79\mu W$ is taken up by the synapse circuit. We have cascaded the integrate and fire circuit, two stage lowpass filter, derivative circuit, bias circuit and synaptic circuit such that each operates within its required region of operation. The five blocks are working individually as well as together as intended. This combined circuit gives us a low-power and energy efficient, biologically inspired algorithm for real time learning tasks. Future experiments will involve alterations of this circuit in-order to make improvements in it.

References

[1] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers in Computational Neuroscience*, vol. 11, 2017. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fncom.2017.00024

[2] B. Taylor, N. Ramos, E. Yeats, and H. Li, "CMOS implementation of spiking equilibrium propagation for real-time learning," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2022, pp. 283–286.