LODeNNS: A Linearly-approximated and Optimized Dendrocentric Nearest Neighbor STDP

Akwasi Akwaboah
aakwabo1@jhu.edu
Department of Electrical and Computer Engineering
Johns Hopkins University
Baltimore, Maryland, USA

Ralph Etienne-Cummings
retienne@jhu.edu
Department of Electrical and Computer Engineering
Johns Hopkins University
Baltimore, Maryland, USA

ABSTRACT

Realizing Hebbian plasticity in large-scale neuromorphic systems is essential for reconfiguring them for recognition tasks. Spike-timing-dependent plasticity, as a tool to this effect, has received a lot of attention in recent times. This phenomenon encodes weight update information as correlations between the presynaptic and postsynaptic event times, as such, it is imperative for each synapse in a silicon neural network to somehow keep its own time. We present a biologically plausible and optimized Register Transfer Level (RTL) and algorithmic approach to the Nearest-Neighbor STDP with time management handled by the postsynaptic dendrite. We adopt a time-constant based ramp approximation for ease of RTL implementation and incorporation in large-scale digital neuromorphic systems.

KEYWORDS

STDP, neuromorphic systems, spiking neural networks

ACM Reference Format:
Akwasi Akwaboah and Ralph Etienne-Cummings. 2022. LODeNNS: A Linearly-approximated and Optimized Dendrocentric Nearest Neighbor STDP. In *International Conference on Neuromorphic Systems (ICONS 2022), July 27–29, 2022, Knoxville, TN, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3546790.3546793

1 INTRODUCTION

In a giant messy soup of neuronal connections, timing is everything! Computing with spikes heavily relies on the arrival times of presynaptic events and their correlations with postsynaptic spike times.[1] A well-appreciated contributor to hebbian learning is spike-timing-dependent plasticity (STDP). This involves either the potentiation (increase) or depression (decrease) of synaptic weights based on a causal or anti-causal postsynaptic spiking with respect to the presynaptic activity. The biological basis of this arises from the modulation of the density of the solely ligand-gated AMPA receptors in response to the amounts of Ca^{2+} ions in the dendrite of the postsynaptic neuron. While these AMPA receptors, in the presence of the neurotransmitter glutamate released by the presynaptic neuron, allow the influx of excitatory Na^{+} and K^{-} ionic currents; another dendritic receptor, the voltage- and ligand-sensitve NMDA channels in addition to other ionic currents facilitate the influx of the needed Ca^{2+}ions. A causal postsynaptic spike leads to an increased intra-dendritic Ca^{2+} concentration which through a series of chemical pathways leads to either an increased postsynaptic sensitivity (via an enhanced AMPA receptor synthesis) [15] or an increased presynaptic neurotransmitter release probability (via a positive nitric oxide feedback from the post-synaptic neuron) [8]. Conversely, an acausal postsynaptic spike results in lower intra-dendritic concentration, which in turn leads to the depopulation of AMPA receptors.

Realizing *in silico* implementations of the STDP has largely been approached in two ways: the classical all-to-all spike pairing and the Nearest-Nearest (NN) approach [2, 9, 11, 13]. In the former, the contributions of all presynaptic and postsynaptic spike pairs occurring within a specified time window (typically a few tens of milliseconds) are equally considered in determining a synaptic weight update. The NN approach, on the other hand, only considers the peripheral spike pairs, triplet or quadruplet arising from two terminating post-synaptic (presynaptic) events and intermediate presynaptic (postsynaptic) events if any. Izhikevich and Desai[9] argue out the biological plausibility of the NN STDP relative to the all-to-all approach. Their justifications include the backpropagation of postsynaptic spike into the dendrites, which effectively resets membrane potential there, thus annihilating the effects of past postsynaptic spikes. Another perspective is that the immediate succeeding postsynaptic spike possibly overrides the influence of subsequent spikes as a results of desensitization of glutamate receptors or calcium saturation.

Hardware implementation of STDP requires consideration for time management to prevent time saturation or overflow, especially when the system is intended to run indefinitely. More so, optimal caching of event times offers memory- and power-saving advantages at the scale of large networks. In the digital domain, timing can be kept via a registered accumulator and in the analog domain via a pulsed-capacitor based circuit. For a large-scale neural network, one may choose to either keep a global timer with which all synapses stamp their events or several local timers for the various synapses. While the option of a global timer naively seems simple, it has the drawback of not easily determining a judicious range (or bits required) sufficient for covering the nuanced random spike patterns over the entire network. Even if, one manages to arrive at a suitably-ranged timer, computing time differences between spikes become unnecessarily slow and cumbersome in the event of uncorrelated spiking activity. Postsynaptic spikes may occur well outside of the potentiation/ depression window and computing

ICONS 2022, July 27–29, 2022, Knoxville, TN, USA

ACM ISBN 978-1-4503-9789-6/22/07…$15.00
https://doi.org/10.1145/3546790.3546793

the time difference needed for determining the weight update will require a large-bit arithmetic operation.

Related work (digital domain) include the STDP implementation using an elegant digital combinational logic-based convolutation by Cassidy *et al.*[4] and a more elaborate implementation by Belhadj *et al.* [3] involving first order approximation to exponential weight decay function. However, both effectively adopt an all-to-all STDP approach. A more recent work is that by Lammie *et al.*[10], where separate implementations of paired-, triplet-, and quadruplet-spike STDP implementation were performed. A much closely related work is the pairwise NN STDP rule adopted in Loihi[5], a digital large-scale neuromorphic system. Here, depressive weight updates can be easily determined in an event-driven manner as the weight update occur in a feedforward fashion, i.e. on presynaptic events. On the other hand, Davies *et al.*[5] articulate the difficulty of computing the potentiating half of the STDP function in an event-driven manner as it requires some form of backward routing and instead resort to a weight update on an epoch basis.

In this paper, we present LODeNNS, an optimized Register-Transfer-Level (RTL) *cum* algorithmic implementation of the NN STDP with a bounded local synaptic timekeeping useful at implementing decentralized and postsynaptic-event-driven weight update computations in a silicon neural networks. Major design optimization highlights of this work are the constraining of and minimal storage of event times to prevent expensive arithmetic and time saturation or overflows; as well as an integrated solution that allows the transition between paired-, triplet-, and quadruplet spike selection at the end of a terminating postsynaptic spike depending on the arrival times of the pre- and post- synaptic spike(s). The paper is organized as follows: §I presents the motivation for this work and related work, §II captures the theory for the adopted curve approximation strategies , §III contains implementation details is then followed by §IV, a preliminary feature extraction demonstration and §V, conclusion and future work.

2 THEORY

In order to simply realize STDP in the digital domain, the potentiation/depression time window and weight update rule must be aptly approximated as the actual curves decay exponentially with the spike time difference ($\Delta t = t_{post} - t_{pre}$). The STDP curve is mathematically defined by;

$$\Delta w(\Delta t) = \begin{cases} A_+ e^{-\frac{\Delta t}{\tau_+}} & \Delta t > 0 \\ A_- e^{\frac{\Delta t}{\tau_-}} & \Delta t < 0 \end{cases} \tag{1}$$

where $A_+ > 0$ and $A_- < 0$ are the initial/ maximum potentiation and depression intensities respectively. Whereas τ_+ and τ_- are the potentiation and depression time constants. Typical parameter values determined from experimental data from pyramidal neurons within Layer 2/3 of rat visual cortex can be found in [6]. While it is obviously easy to implement exponentials in the analog subthreshold domain, a linear/ ramp approximation is relatively convenient to implement in the digital domain with fewer arithmetic operations at a reasonable approximation error and as such we adopt that it in this study. From a time-constant based consideration for either Δw or Δt range preservation, two kinds of linear approximations can be used – tangential and chordal. Both are discussed next.

2.1 Time-Constant-based Linear Approximations: Tangential vs. Chordal

The approximation can be expressed as a piecewise linear relation shown in eq. 2;

$$\Delta \widetilde{w}_k(\Delta t) = \begin{cases} -\frac{A_+}{\tau_+}\mathbf{a}_k^T \mathbf{s}_+ & 0 < \Delta t < \alpha_k \tau_+ \\ \frac{A_-}{\tau_-}\mathbf{a}_k^T \mathbf{s}_- & -\alpha_k \tau_- < \Delta t < 0 \\ 0 & \text{elsewhere} \end{cases} \tag{2}$$

where $\mathbf{a}_k = \begin{bmatrix} \beta_k \\ m_k \end{bmatrix}$, $\mathbf{s}_+ = \begin{bmatrix} -\tau_+ \\ \Delta t \end{bmatrix}$ and $\mathbf{s}_- = \begin{bmatrix} \tau_- \\ \Delta t \end{bmatrix}$.

α_k, β_k and m_k are $k\tau$-dependent scalings for the horizontal, vertical intercept and slope factors respectively.

Tangential approximation involves the weight update line picking up the gradient of a point on the curve. A reasonable point of choice proposed here is a line tangent at $(k\tau, Ae^{-k})$, where $k \in \mathbb{Z} : k = 0, 1, 2, \cdots$. Choosing smaller k values favor the preservation of the weight update intensity at the expense of the time window and vice versa for larger k values. The general tangential linear approximation slope and vertical- and horizontal intercept factors are respectively defined by:

$$\begin{aligned} m_k &= e^{-k} \\ \beta_k &= (k+1)e^{-k} \\ \alpha_k &= \frac{\beta_k}{m_k} = k + 1 \end{aligned} \tag{3}$$

Here, $k = 1, 2, 3$ are presented as suitable approximations for Δw intensity preservation, trade-off between Δw intensity and time window, and time window preservation respectively. This can be inferred from the Figure 1. The Figure of Merit (FOM) used here is the Area-Under-Curve overlap (AUC) between the approximation and the STDP curve, shown in eq. 4, which is merely the area under the approximation (AUA).

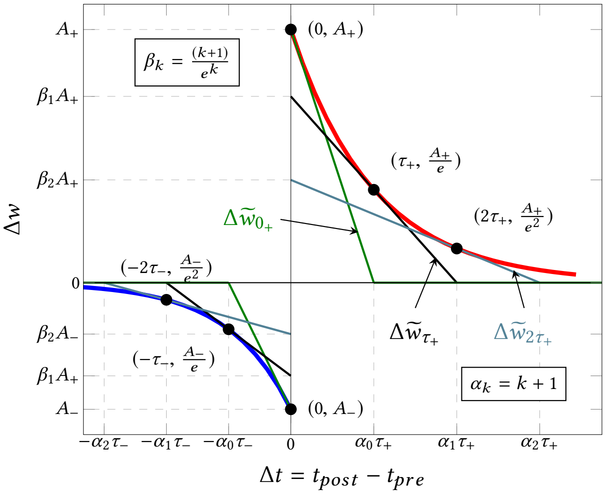

Figure 1: Tangential Linear Approximation

$$AUC = \frac{1}{2}\alpha_k \beta_k \tau |A| \tag{4}$$

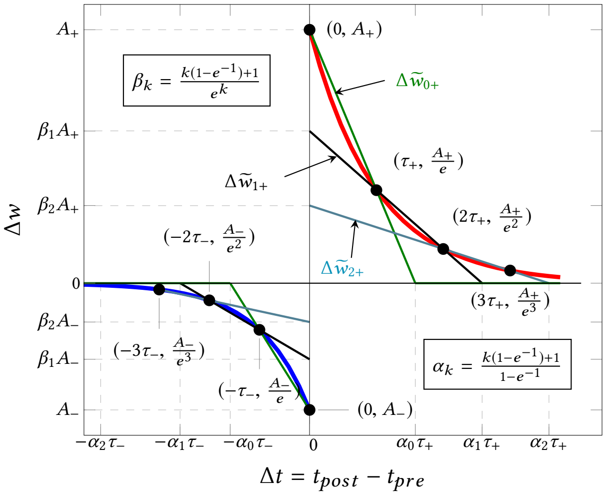

Figure 2: Chordal Linear Approximation

*AUC*s for $\Delta\widetilde{w}_0$, $\Delta\widetilde{w}_\tau$ and $\Delta\widetilde{w}_{2\tau}$ are $0.5\tau|A|$, $\frac{2}{e}\tau|A|(\approx 0.736\tau|A|)$, and $\frac{1.5}{e^2}\tau|A|(\approx 0.203\tau|A|)$ respectively and based on this $\Delta\widetilde{w}_\tau$ (i.e. $k = 1$) appears to be the preferred choice and as a matter of fact is the optimal k value for maximum AUC here.

Chordal approximation, on the other hand, involves the selection of a pair of chordal points $(k\tau, Ae^{-k})$ and $((k+p)\tau, Ae^{-(k+p)})$ for a line that similarly maximizes the *AUC* preferentially for either the Δw intensity or time window. The reasonable choice for the chordal line with curve intersection points as integer multiples of the time constant. While, the choice of the integer multipliers for the pair of points must not necessarily be consecutive, increasing the point separation increases the undesired Area-Above-Curve (AAC) that is overlapped by the approximation. The general chordal linear approximation slope, vertical- and horizontal-intercept factors are defined as:

$$\begin{aligned}
\gamma_p &= \frac{1 - e^{-p}}{p} \\
m_k &= \gamma_p e^{-k} \\
\beta_k &= (\gamma_p k + 1)e^{-k} \\
\alpha_k &= \frac{\beta_k}{m_k} = \frac{\gamma_p k + 1}{\gamma_p}
\end{aligned} \tag{5}$$

Again, we adopt $k = 1, 2, 3$ and $p = 1$ resulting in three approximation lines from three pairs of consecutive chord points as shown in Figure 2. Setting $p = 0$ (and by the L'Hôpital rule), the tangential linear approximation emerges, indicating the generality of the chordal approximation. A good trade-off here is chosen based on k that maximizes the *AUC*, which is governed by:

$$AUC = \frac{1}{2}\alpha_k\beta_k\tau|A| - \beta_k p\left[1 - \frac{p + 2(k+1)}{2\alpha_k}\right]\tau|A| \tag{6}$$

Coincidentally $k = 1$ yields the optimal approximation of the three.

2.2 Why Dendrocentric?

On the spatial front, the cable properties of a dendrite imputes a distance-dependent weighting to presynaptic inputs. The closer the synapse is to the soma, an excitatory (inhibitory) post synaptic potential, EPSP (IPSP) observed at the soma is attenuated less. [14] Membrane potential (ϕ) within the dendrite decay exponentially with a length constant (λ), which is function of resistance per unit length of the membrane acting as an insulator (r_m) and the intradendritic (conducting core) resistance, r_i i.e. $\phi(x) = \phi_o e^{-\frac{x}{\lambda}}$, where x is the axial distance along the dendrite from the synapse input and $\lambda = \sqrt{\frac{r_m}{r_i}}$. Emulating this spatial weighting effect in hardware can be achieved by simply scaling the corresponding synaptic weight by the appropriate attenuation factor. By the assumption that synaptic inputs are permanently localized at a dendritic site, the attenuation factor is constant.

On the other hand, one can imagine how STDP as a temporal weighting strategy is curated by the dendrite. Through a cascade of chemical pathways facilitated by ligand- and voltage-sensitive channels, the dendrite is able to memorized recent activity and resets the event-time memory upon postsynaptic spiking. We use the term "*dendrocentric*"[1] here to emphasize the role of the postsynaptic neuron at reseting the local synaptic "stopwatch" together with latched event times.

2.3 Dendrocentric NN STDP Synaptic Timekeeping

The dendrocentric NN STDP presented here focuses on two boundary postsynaptic spikes and the intermediary earliest and latest presynaptic spike(s). Synaptic clocking managed by the dendrite is needed to determine the event times. Since the digital timer is merely a counter-increment every clock period T_{clk}, it is important to make adjustments to eq. 2 to account for discreteness of time. Here, all time parameters are expressed as integer multiples of qT_{clk}, where q is the time-acceleration factor. The constraint for q is $q > 0$ ($q = 1$: real-time, $q > 1$: accelerated-time, $q < 1$: delayed-time). As such,

$$\mathbf{s}_+ = qT_{clk}\begin{bmatrix} -\eta_+ \\ \Delta n_c \end{bmatrix} \text{ and } \mathbf{s}_- = qT_{clk}\begin{bmatrix} \eta_- \\ \Delta n_a \end{bmatrix}$$

$$\text{where } \eta_+ = \frac{\tau_+}{qT_{clk}},\ \eta_- = \frac{\tau_-}{qT_{clk}}$$

whereas Δn_a is the acausal duration obtained as a difference between the initial postsynaptic event time count (n_{post_1}) and the the earliest presynaptic time count (n_{pre_1}) and Δn_c is the causal duration obtained as the difference between the terminating postsynaptic event and latest time count (n_{post_2}) and the latest presynaptic time count (n_{pre_2}). Here on, we switch from using explicit time parameters to their time count equivalents.

Two time rollover conditions are adopted here. The first and more prioritized condition is determined by the terminating postsynaptic event time. Here, the timer is reset to 1 instead of 0 as the later is reserved for the *Primus Spike Lock* (PSL) condition, which is explained later. The second rollover condition is determined by the maximum sizes of the depression and the potentiation windows and an additional headroom for instances when the earliest and/or latest presynaptic event(s) occur(s) outside of the sum of the potentiation

[1] Perhaps the first and recent use of this term in the context of neuromorphics is by Kwabena Boahen [12]

and depression time windows (i.e., $\alpha_k(\eta_- + 2\eta_+)$). This provides an upper bound to the timer instead of a naive arbitrarily large timer bit-width and consequently offers computation-/memory-saving advantages per synapse. In fact, the minimum bit width for the timer can be determined from

$$N_{k,min} = ceil\left(\log_2(\alpha_k(\eta_- + 2\eta_+))\right) \tag{7}$$

Intuitively, increasing the clock frequency f_{clk} for a real-time implementation improves the resolution at the expense of storing larger-bit width register values and arithmetic operations, and vice versa for lower clock frequencies. To ameliorate such a burden, we encourage a clock frequency scaling in powers of 2, so register values are merely scaled through static shifts instead of explicit multiplications. This leads to adjusted count relation of $n = (2^{N_o+M+Q}) \cdot t$, where N_o is the minimum bit-width for the parameters $T_{clk} = 1\ ms$ and $q = 1$ (i.e. $N_o = 8$), while M and Q are scaling exponents for f_{clk} and q respectively.

By the NN principle adopted here, only the peripheral presynaptic events within postsynaptic spike interval are of importance, while the medial ones are considered redundant. As such, this ameliorates the large memory requirement for storing all possible event times. It is also important to note that a terminating postsynaptic spike in the recent past interval becomes the initial in the next interval as such resets event-time registers. In all, three time register are used to cache a maximum of four event times. By resetting the initial postsynaptic event time is inferred from a rollover, i.e. $n_{post_1} = 0$. The terminating postsynaptic event time is tracked by the current timer value, i.e., $n_{post_2} = n_i$, while the earliest and the latest postsynaptic event times are cached (as and when) in the n_{pre_1} and n_{pre_2} registers respectively. These are adopted in time vectors of the linear approximation $\mathbf{s}_- = qT_{clk}\mathbf{u}_-$ and $\mathbf{s}_+ = qT_{clk}\mathbf{u}_+$. where $\mathbf{u}_- = \begin{bmatrix} \eta_- \\ -n_{pre_1} \end{bmatrix}$ and $\mathbf{u}_+ = \begin{bmatrix} -\eta_+ \\ n_i - n_{pre_2} \end{bmatrix}$

Importantly, n_{pre_1} and n_{pre_2} is reset to either the lower or the upper bound where a coincidence between postsynaptic and presynaptic events can be checked without explicit need for time. This involves monitoring the single-bit event registers for a concurrent bit assertion. We arbitrarily choose the lower bound of 0 as reset for the presynaptic event time registers and 1 for n_i. Thus, any weight update computation is preceded by a check for such reset value in the earliest event time register (i.e. $n_{pre_1} = 0$) and if true, Δw is set to 0. This is important as it prevents a rollover reset generating an already-handled spike coincidence. We reserve an on-start initialization of $n_i = 0$ for the PSL condition, which involves the suppression of weight update computation until the second ever postsynaptic event is observed. The first-ever postsynaptic event sets $n_i = 1$, and readies the system for weight update on the second.

The NN STDP idea presented here is based on an arbitrated selection of acausal and causal postsynaptic-presynaptic spike pairs depending on the regimes in which the earliest and latest spike(s) occur. Since the postsynaptic spike interval is randomly dynamic, there is the need to verify the relevance of lateral presynaptic events to the overall weight update determination. Figure 3 shows how the potentiation and depression windows of interest can overlap partially and at the extreme either fully-overlap or become adjacent, i.e., $n_i - \alpha_k\eta_+ \in [0, \alpha_k\eta_-]$, as well as non-overlapping with a gap, in which case $n_i - \alpha_k\eta_+ \in [0, +\infty]$.

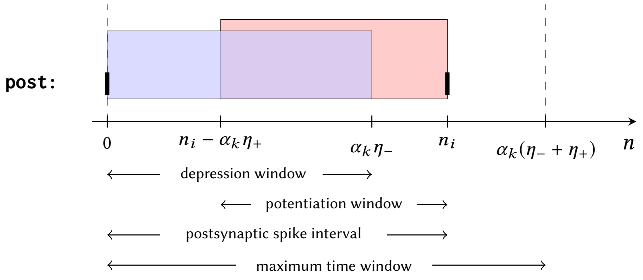

Figure 3: Dynamic overlap between the potentiation and depression windows.

Accommodating this infinite time range seems impractical. We, however, present a workaround for this by pausing the count when $\alpha_k(\eta_- + \eta_+)$ is reached and a terminating postsynaptic event is absent. We term this *Spike Limbo* (SL). In this case, the acausal pre-post paired-spike, (hence n_{pre_1}) is not of much use, as the depression window is guaranteed to have been exceeded. Rather, the causal pre-post paired-spike (involving n_{pre_2}) is tracked by resuming the count in the event of a presynaptic spike occurence at this point with the anticipation that a postsynaptic spike arrives afterwards within the potentiation window (i.e. $< \alpha_k(\eta_- + 2\eta_+)$). The count is reset to $\alpha_k(\eta_- + \eta_+)$ if the terminating postsynaptic event does not show up. It is important to note that, a successive presynaptic event after the first spike in this regime is given attention by resetting the count to $\alpha_k(\eta_- + \eta_+) + 1$, which is just above the pause condition hence, the count resumes. We introduce a two single-bit spike pairing flags, v_0 (associated with the initial acausal pre-post pair) and v_1 (associated with the terminating causal pre-post pair) here to switch between no-, paired-, triplet- or quadruplet-spike selection as shown in Figure 5. Conditions for asserting or clearing these flags are as follows,

$$v_0 = \begin{cases} 1 & n_{pre_1} \leq \alpha_k\eta_- \\ 0 & \text{otherwise} \end{cases} \quad v_1 = \begin{cases} 1 & n_{pre_2} \geq n_i - \alpha_k\eta_+ \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

In summary, the general equation for the net linear weight update, denoted by $\Delta\widehat{w}_k$ (shown in eq. 9), involves a summation of the causal and acausal weight update contributions while allowing the paired-, triplet-, and quadruplet-spike selection via the spike pairing flags v_1 and v_2.[2]

$$\Delta\widehat{w}_k = -\left(v_0\frac{|A_-|}{\eta_-}\mathbf{a}_k^T\mathbf{u}_- + v_1\frac{A_+}{\eta_+}\mathbf{a}_k^T\mathbf{u}_+\right) \tag{9}$$

In the case of a symmetric STDP, i.e. $A = |A_-| = A_+$ and $\eta = \eta_- = \eta_+$, eq. 9 simplifies to:

$$\Delta\widehat{w}_k = -\frac{A}{\eta}\mathbf{a}_k^T\mathbf{U}\mathbf{v} \tag{10}$$

[2] Triplet and quadruplet STDP rules used in other related works may vary. While it is common to adopt a product of causal and acausal weight contributions[7], we rather adopt an additive (instead of multiplicative) rule, guided by the aim of implementing lightweight compute

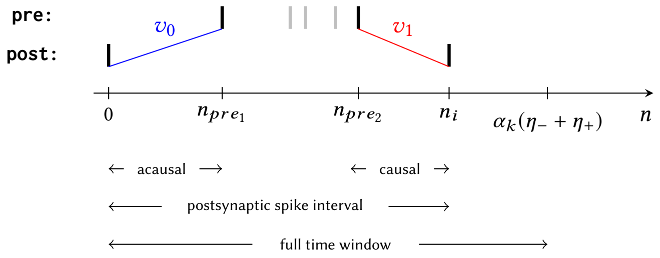

Figure 4: Spike groupings. Up to two spike pairs can be selected: an acausal pair involving the initial postsynaptic event and the earliest presynaptic event and a causal pair involving the terminating postsynaptic event and the latest presynaptic event. A single presynaptic event in the postsynaptic interval doubles as the earliest and latest presynaptic event leading to triplet-spike selection when v_0 and v_1 are asserted. If multiple presynaptic events occur, then earliest and the latest event are distinct leading to a quadruplet-spike selection when v_0 and v_1 are asserted, or either an acausal or causal paired-spike if only one of v_0 and v_1 is asserted.(see eq. 8 for spike grouping flag conditions)

where $\mathbf{U} = \begin{bmatrix} \mathbf{u}_- & \mathbf{u}_+ \end{bmatrix}$, $\mathbf{v} = \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}$.

For the triplet-spike selection and the quadruplet spike selection, the critical/ anticipated temporal parameters leading to $\Delta w = 0$, $n_i^{(0)}$, $n_{pre_1}^{(0)}$ and $n_{pre_2}^{(0)}$ have a planar relation shown in eq. 11:

$$n_i^{(0)} = n_{pre_2}^{(0)} + \left(\frac{|A_-|}{A_+} \frac{\eta_+}{\eta_-} \right) n_{pre_1}^{(0)} + (k+1) \left(1 - \frac{|A_-|}{A_+} \right) \eta_+ \qquad (11)$$

with the constraint $0 < n_{pre_1} \leq n_{pre_2} < n_i$ for any causal-acausal analysis. In the case of the triplet-spike, a given $n_{pre}^{(0)} = n_{pre_1}^{(0)} = n_{pre_2}^{(0)}$, $n_i < n_i^{(0)}$ yields $\Delta w > 0$ as it effectively shifts $n_{pre}^{(0)}$ further into the causal regime than acausal, and vice versa for $n_i > n_i^{(0)}$. The thinking is reversed if an analysis of a variable n_{pre} compared against $n_{pre}^{(0)}$ determined from a given $n_i^{(0)}$ is preferred. More so, the quadruplet spike selection follows a similar thought. Eq. 11 is useful in the sense that it also shows the time relations for symmetric STDP; the right hand side of the equation reduces to $n_{pre_1}^{(0)} + n_{pre_2}^{(0)}$. On another hand, if one manages to implement the exponential function in digital domain, albeit at most likely an expensive computational cost, the sum of the exponentials in eq.1, yields critical event time planar relation of:

$$n_i^{(0)} = n_{pre_2}^{(0)} + \left(\frac{\eta_+}{\eta_-} \right) n_{pre_1}^{(0)} + \log \left(\frac{|A_-|}{A_+} \right) \eta_+ \qquad (12)$$

Similarly, STDP symmetry yields to the $n_{pre_1}^{(0)} + n_{pre_2}^{(0)}$ on the RHS of eq.12 and the resulting Δw sign follows as before.

3 IMPLEMENTATION

Two implementations of the NN STDP per clock instance are presented – algorithmic and RTL. The algorithmic implementation, which is suitable for a sequential implementation, has been shown in algorithm 1 where A_+^*, A_-^*, η_+^*, η_-^*, k^* are tunable hyperparameters.

Algorithm 1 Proposed Nearest Neighbor Algorithm

```
function NN_STDP($E_{pre}$, $E_{post}$, $\Delta w$, $n_i$, $n_{pre_1}$, $n_{pre_2}$, $v_0$, $v_1$, $A_+^*$, $A_-^*$, $\eta_+^*$, $\eta_-^*$, $k^*$)
    $\alpha_k \leftarrow k + 1$
    if $E_{post} = 1$ then
        if $E_{pre} = 1$ then
            $\Delta w' \leftarrow 0$
        else
            if $n_{pre1} = 0$ then
                $\Delta w_- \leftarrow 0$
            else
                $\Delta w_- \leftarrow v_0 \frac{|A_-|}{\eta_-} [n_{pre_1} - \alpha_k \eta_-]$
            if $n_{pre2} = 0$ then
                $\Delta w_+ \leftarrow 0$
            else
                $\Delta w_+ \leftarrow v_1 \frac{A_+}{\eta_+} [n_{pre_2} - n_i + \alpha_k \eta_+]$
            $\Delta w' \leftarrow e^{-k} (\Delta w_- + \Delta w_+)$
        $n_i' \leftarrow 1$, $n_{pre_1}' \leftarrow 0$, $n_{pre_2}' \leftarrow 0$, $v_0 \leftarrow 1$, $v_1 \leftarrow 1$
    else
        if $n_i = 0$ then                                          ▷ Primus Spike Lock
            $n_i' \leftarrow 0$, $n_{pre_1}' \leftarrow 0$, $n_{pre_2}' \leftarrow 0$, $v_0 \leftarrow 0$, $v_1 \leftarrow 0$
        else
            $\Delta w \leftarrow 0$                                          ▷ Optional
            if $n_i = \alpha_k(\eta_- + \eta_+)$ then                         ▷ Spike Limbo
                if $E_{pre} = 1$ then
                    $n_i' \leftarrow n_i + 1$
                else
                    $n_i' \leftarrow \alpha_k(\eta_- + \eta_+)$
            else if $n_i > \alpha_k(\eta_- + 2\eta_+)$ then
                $n_i' \leftarrow \alpha_k(\eta_- + \eta_+)$
            else if $n_i > \alpha_k(\eta_- + \eta_+)$ then
                $n_i' \leftarrow \alpha_k(\eta_- + \eta_+) + 1$
            else
                $n_i' \leftarrow n_i + 1$
            if $E_{pre} = 1$ then                                     ▷ Flags Logic
                $n_{pre_2}' \leftarrow n_i$
                if $n_i \geq n_i - \alpha_k \eta_+$ then
                    $v_1' \leftarrow 1$
                else
                    $v_1' \leftarrow 0$
                if $n_{pre_1} = 0$ then
                    $n_{pre_1}' \leftarrow n_i$
                    if $n_i \leq \alpha_k \eta_-$ then
                        $v_0' \leftarrow 1$
                    else
                        $v_0' \leftarrow 0$
            else
                if $n_{pre_1} < \alpha_k \eta_-$ then
                    $v_0' \leftarrow 1$
                else
                    $v_0' \leftarrow 0$
                if $n_{pre_2} > n_i - \alpha_k \eta_+$ then
                    $v_1' \leftarrow 1$
                else
                    $n_{pre_2}' \leftarrow 0$
                    $v_1' \leftarrow 0$
    return $\Delta w'$, $n_i'$, $n_{pre_1}'$, $n_{pre_2}'$, $v_0'$, $v_1'$
  * indicates tunable hyperparameters
```

The RTL implementation suitable for a concurrent implementation has been presented in Figure 7 and design considerations are as follows. Precise multiplications by $\frac{A}{\eta}$ are posed as multiplierless static shifts in powers of 2 to reduce compute, whereas the v_0 and v_1 scalings are multiplexed. For generality, we introduce $\psi = round(\log_2(|\frac{A}{\eta}|))$ to provide the discrete shift magnitude and direction, shifting $|\psi|$ times left if $\psi > 0$, right if $\psi < 0$ and no shift

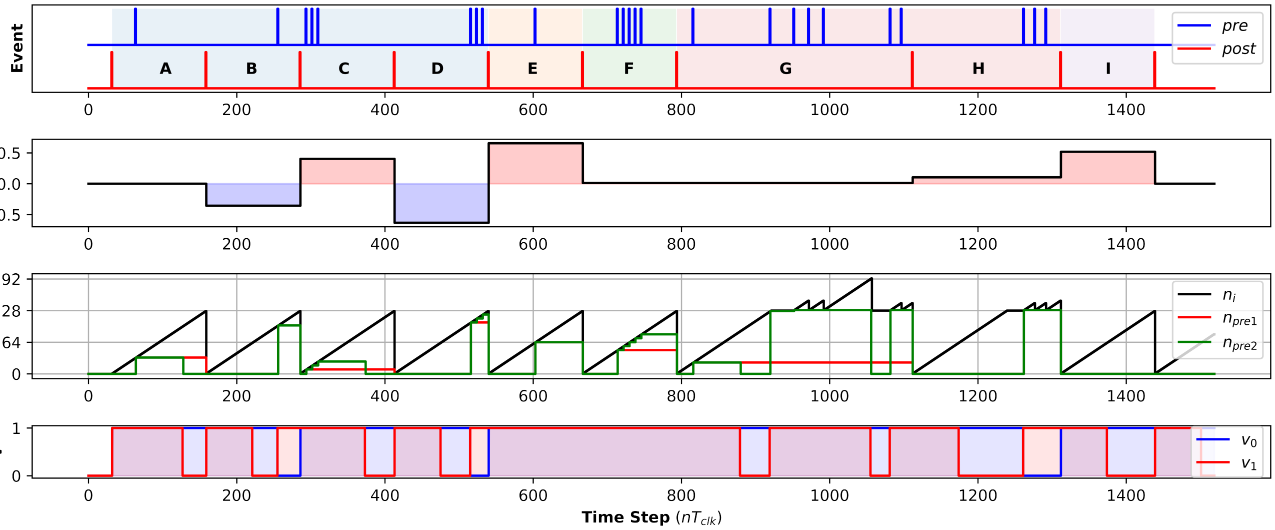

Figure 5: Behavioral simulation of LODeNNS: Weight updates, along with the underlying timekeeping and causal-acausal flag selection for various prominent pre-post cases are presented. $A_+ = |A_-| = 1$, $\eta_+ = \eta_- = 32$, $k = 1$**. Primus Spike Lock (i.e.** $n_i = 0$**) suppresses** Δw **computation until first postsynaptic spike A (C): acausal pre-post spike pair selection on a single (multiple) presynaptic spike, B (D): causal spike pair selection on a single (multiple), E: spike triplet selection, F: quadruplet spike selection, G, H: Spike Limbo cases for paired and quadruplet spikes. I: No spike.**

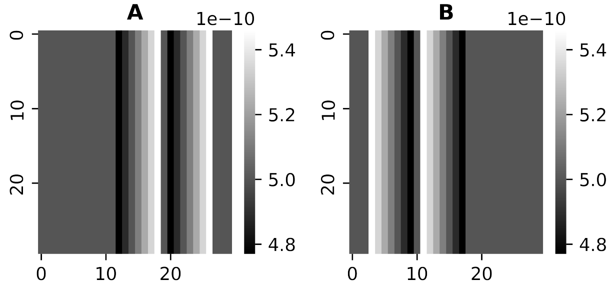

Figure 6: Memory-encoded spatial feature maps (synaptic weights) for A: Pattern A, an eastward moving columnar event and B: Pattern B, a westward moving columnar event generated using LODeNNS incorporated in a two-layer fully connected network. Spatiotemporal patterns A and B have a common spatial energy per instance but reversed sequence.

$\psi = 0$ (i.e. 2^{ψ}). Additionally, k values that favor α_k to be powers of two such that $\rho = \log_2(\alpha_k)$, $\rho \in \mathbb{Z}$ for the same reason as $\frac{A}{\eta}$ can be chosen. This leads to

$$\begin{aligned} \Delta\widehat{w}_{k-} &= v_0 \cdot \textbf{shift}\left(n_{pre_1} + \textbf{shift}(-\eta_-, \rho),\ \psi_-\right) \\ \Delta\widehat{w}_{k+} &= v_1 \cdot \textbf{shift}(n_{pre2} - n_i + \textbf{shift}(\eta_+, \rho),\ \psi_+) \\ \Delta\widehat{w}_k &= e^{-k} \times (\Delta\widehat{w}_{k-} + \Delta\widehat{w}_{k+}) \end{aligned} \tag{13}$$

where × refers combinational logic multiplier.

As opposed to the algorithmic implementation, tuning hyperparameters in the RTL implementation post synthesis comes at an elevated computational complexity as shifts become dynamic. Notwithstanding, if a specific application with computational efficiency is desired, one can algorithmically determine optimal hyperparameters prior to RTL synthesis. Biologically plausible parameter values chosen for demonstrative purposes are: $A_+ = 1$, $\tau_+ = 16\ ms$, $A_- = -0.5$, $\tau_- = 32\ ms$ as convenient approximations to experimentally-determined[6] values of $A_+ = 1.03$, $\tau_+ = 14\ ms$, $A_- = -0.51$ and $\tau_- = 34\ ms$ adopted by [9]. Running at $T_{clk} = 1\ ms$ and in real-time ($q = 1$), $\eta_+ = 16$ and $\eta_- = 32$ which are powers of 2 and thus, allow multiplierless scaling by $\frac{A}{\eta}$ factors through static shifts. Choosing $k = 1$ for an optimal tangential approximation previously discussed yields $\rho = 1$, $\psi_+ = -4$, $\psi_- = -6$ and consequently leads to eq 14. In all, 5 adders and a single multiplier are used - four for Δw and one for the timer update, which is fewer than a naive implementation of eq. 9 that may require at least four multipliers (two each for the depressive and the potentiating portions).

$$\begin{aligned} \Delta\widehat{w}_{1-} &= v_0 \cdot \textbf{shift}\left(n_{pre_1} + \textbf{shift}(-\eta_-, 1),\ -6\right) \\ \Delta\widehat{w}_{1+} &= v_1 \cdot \textbf{shift}(n_{pre2} - n_i + \textbf{shift}(\eta_+, 1),\ -4) \\ \Delta\widehat{w}_1 &= e^{-1} \times (\Delta\widehat{w}_{1-} + \Delta\widehat{w}_{1+}) \end{aligned} \tag{14}$$

By the linear approximation, $\Delta\widehat{w}_k$ inherits a bit width of the timer, N_k. With $N_{1,min} = 8$ for the above mentioned clock parameters. As such, a 9-bit signed fixed point (FP) representation for $\Delta w \in [-0.5, 1]$ can be used. The most significant bit (MSB) is reserved for the sign while the remaining bits are dedicated to the fractional part. No integer part is reserved since almost all the magnitude range of Δw is sub-unity. The resulting adjusted range is $\Delta\widehat{w}_1 \in (-e^{-1}, 2e^{-1}) \approx (-0.3672, 0.7343)$. Δw is set to 0 on a spike coincidence as neither causality nor acausality can be deciphered,

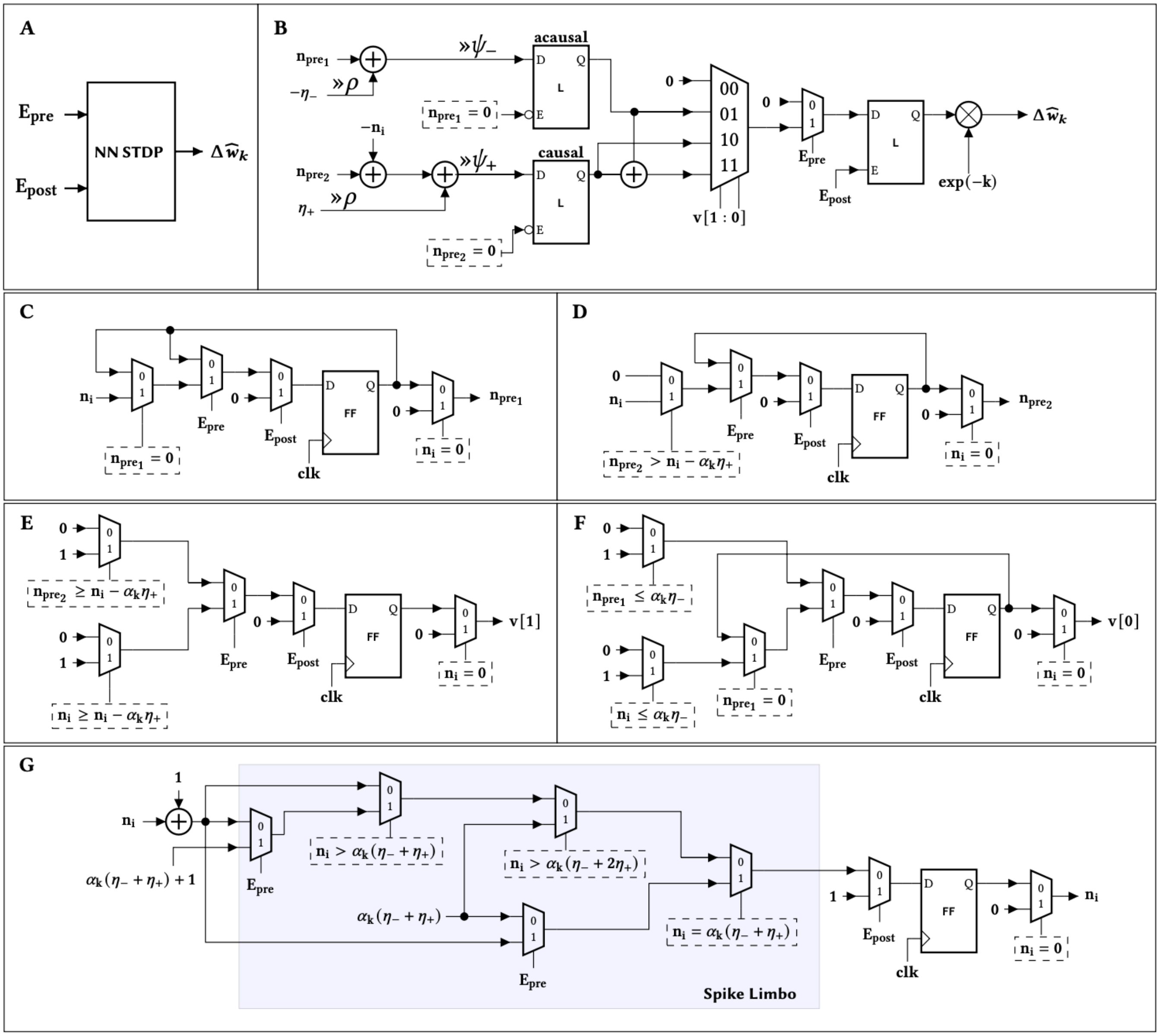

Figure 7: RTL System Architecture for LoDeNNS. A: General system abstraction showing the inputs (presynaptic, $\mathrm{E}_{\mathrm{pre}}$ and postsynaptic $\mathrm{E}_{\mathrm{post}}$ events) and the $\Delta\widehat{w}_k$ output B: Top Level RTL architecture, showing how earliest and latest presynaptic event times, n_{pre_1} and n_{pre_2}, are deployed in generating solely or combined latched acausal and causal $\Delta\widehat{w}_k$ contributions. Static shifts instead of explicit multiplications are adopted where possible to reduce the computational burden. C-G captures the subcircuits that generate the control signals used at the top level. In G, the accommodation for a delayed arrival of $\mathrm{E}_{\mathrm{post}}$, *Spike Limbo*, is highlighted. $\bigoplus$ and $\bigotimes$ represent combinational adder and multiplier respectively

hence the exclusion of the end-points (positive and negative y-intercepts) from the Δw interval. The 9-bit FP representation is also sufficient for storing the scaling factor e^{-1} (in eq. 14) as this is also sub-unity. Effectively, $e^{-1} \approx 0.3672$ and a corresponding binary value of b"001011110".

4 FEATURE EXTRACTION DEMONSTRATION

Indeed, timing is everything. In this section, we deploy LODeNNS in converting a spatiotemporal pattern to a spatial synaptic weight feature map encoding temporal information post exposure as a preliminary demonstration. The highlight here is that the memory-encoding dimensionality-reduction property inherent in STDP can be leveraged in a prior dynamic feature extraction for a subsequent static classifier such as a convolutional neural network. We show this with two spatiotemporal patterns with the same spatial energy per temporal instance occurring at a reversed sequence. Simply put, pattern A, is a columnar event travelling eastward, while pattern B travels westward. Both Patterns A and B are of size 30 × 30 and change in unit step over 30 time steps. The feature extraction mechanism is composed of a two-layer fully connected network with the input layer being a vectorized instantaneous spatial input activity and the output/postsynaptic neuron being a Leaky Integrate-and-Fire (LIF) neuron. The neuron parameters used are as follows: membrane time constant (τ_m) of 30 ms, refractory period of 4 ms, spike threshold of 10 mV and reset potential of 0 mV. The input synaptic current was modeled as a Heaviside alpha function with a time constant (τ_s) of 5 ms, convolved with all connected presynaptic activity. A time step, $dt = 1$ ms was used. Synaptic weights were initialized with $0.5e - 9$ and were independently amenable by LODeNNS. A symmetry STDP with parameters same as used in Figure 5 was adopted. Figure 6 shows the simulation results for the memory-encoding feature maps (synaptic weights) for both patterns. A time-depth and consequently a direction of travel can be perceived from both spatial feature maps.

5 CONCLUSION

We present LoDeNNS, an algorithmic and optimized RTL implementation for the nearest nearest STDP suitable for event-driven weight updates in spiking neural networks and by extension holds implications for learning-on-the-fly as against epoch-based methods. We adopted a time constant-based linear approximation to the exponential STDP function that allows trade-off adjustments between weight update intensity and pre-post spike interval window. By substituting, combinational multipliers with static shifts where possible along with other accommodations such as *Primus Spike Lock* and *Spike Limbo*, we arrive at a computationally optimized NN STDP mechanism. More so, through a dendrocentric timekeeping, timer saturation or overflow become avoidable thus allowing for computing indefinitely. Future work include an analog/ mixed-signal implementation to allow ease of hyperparameter programmability post-hardware realization, as well as adopting LoDeNNS at a network level on a myraid of supervised and unsupervised spatiotemporal recognition tasks. Implementation code have been made available at https://github.com/Adakwaboah/LODeNNS

ACKNOWLEDGMENTS

We would like to thank NSF for the support through Grants #SMA-1540916, #1607522 & #2020624.

REFERENCES

[1] John V Arthur and Kwabena Boahen. 2005. Learning in silicon: Timing is everything. *Advances in neural information processing systems* 18 (2005), 75–82.

[2] Chiara Bartolozzi and Giacomo Indiveri. 2007. Synaptic dynamics in analog VLSI. *Neural computation* 19, 10 (2007), 2581–2603.

[3] Bilel Belhadj, Jean Tomas, Yannick Bornat, Adel Daouzli, Olivia Malot, and Sylvie Renaud. 2009. Digital mapping of a realistic spike timing plasticity model for real-time neural simulations. In *Proceedings of the XXIV conference on design of circuits and integrated systems*. IEEE, Zaragoza, Spain, 1–6.

[4] Andrew Cassidy, Andreas G Andreou, and Julius Georgiou. 2011. A combinational digital logic approach to STDP. In *2011 IEEE international Symposium of Circuits and Systems (ISCAS)*. IEEE, IEEE, Rio de Janeiro, Brazil, 673–676.

[5] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro* 38, 1 (2018), 82–99.

[6] Robert C Froemke and Yang Dan. 2002. Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature* 416, 6879 (2002), 433–438.

[7] Julijana Gjorgjieva, Claudia Clopath, Juliette Audet, and Jean-Pascal Pfister. 2011. A triplet spike-timing–dependent plasticity model generalizes the Bienenstock–Cooper–Munro rule to higher-order spatiotemporal correlations. *Proceedings of the National Academy of Sciences* 108, 48 (2011), 19383–19388.

[8] Neil Hardingham, James Dachtler, and Kevin Fox. 2013. The role of nitric oxide in pre-synaptic plasticity and homeostasis. *Frontiers in cellular neuroscience* 7 (2013), 190.

[9] Eugene M Izhikevich and Niraj S Desai. 2003. Relating stdp to bcm. *Neural computation* 15, 7 (2003), 1511–1523.

[10] Corey Lammie, Tara Julia Hamilton, André van Schaik, and Mostafa Rahimi Azghadi. 2018. Efficient FPGA implementations of pair and triplet-based STDP for neuromorphic architectures. *IEEE Transactions on Circuits and Systems I: Regular Papers* 66, 4 (2018), 1558–1570.

[11] Jean-Pascal Pfister and Wulfram Gerstner. 2006. Triplets of spikes in a model of spike timing-dependent plasticity. *Journal of Neuroscience* 26, 38 (2006), 9673–9682.

[12] MIT Press. 2021. *The future of AI Hardware: A 3D silicon brain.* MIT. Retrieved March 08, 2022 from https://web.mit.edu/deblina-sarkar/talks_kwabena.html

[13] Harel Z Shouval, Samuel S-H Wang, and Gayle M Wittenberg. 2010. Spike timing dependent plasticity: a consequence of more fundamental learning rules. *Frontiers in computational neuroscience* 4 (2010), 19.

[14] Peter Sterling and Simon Laughlin. 2015. *Principles of neural design.* MIT press, Cambridge, Massachusetts. 170–174 pages.

[15] Jun Wang, Gert Cauwenberghs, and Frédéric D Broccard. 2019. Neuromorphic dynamical synapses with reconfigurable voltage-gated kinetics. *IEEE Transactions on Biomedical Engineering* 67, 7 (2019), 1831–1840.