RESEARCH ARTICLE

Digital multiplier-less implementation of high precision SDSP and synaptic strength-based STDP

Hajar Asgari[1] | **Babak Mazloom-Nezhad Maybodi**[1] | **Yulia Sandamirskaya**[2]

[1]Department of Electrical Engineering, Shahid Beheshti University, Tehran, Tehran, Iran

[2]Institute of Neuroinformatics, University of Zurich (UZH) and ETH Zurich, Zurich, 8057, Switzerland

Correspondence
Babak Mazloom-Nezhad Maybodi, Department of Electrical Engineering, Shahid Beheshti University, Tehran, Tehran, Iran
Email: b-mazloom@sbu.ac.ir

Funding information
SNSF Ambizione, grant number: PZOOP2-168183

Summary

Spiking neural networks (SNNs) can achieve lower latency and higher efficiency compared to traditional neural networks if they are implemented in dedicated neuromorphic hardware. In both biological and artificial spiking neuronal systems, synaptic modifications are the main mechanism for learning. Plastic synapses are thus the core component of neuromorphic hardware with on-chip learning capability. Recently, several research groups have designed hardware architectures for modeling plasticity in SNNs for various applications. Following these research efforts, this paper proposes multiplier-less digital neuromorphic circuits for two plasticity learning rules: the spike-driven-synaptic-plasticity (SDSP) and synaptic-strength-based spike-timing-dependent plasticity (SSSTDP). The proposed architectures have increased the precision of the plastic synaptic weights and are suitable for spiking neural network architectures with more precise calculations. The proposed models are validated in MATLAB simulations and physical implementations on a Field-Programmable Gate Array (FPGA).

KEYWORDS

Neuromorphic engineering, Plastic synapse, Spiking neural network (SNN), FPGA

1 | INTRODUCTION

One of the motivations in the field of neuromorphic engineering is developing configurable hardware to facilitate fundamental neuroscience research on complex brain-based neural networks [1–3]. Commonplace von Neumann computers have limitations in simulating spiking neuronal dynamics [4] because of their sequential processors and the memory "bottleneck". Neuromorphic engineering aims to build scalable and energy efficient computers for running neuronal models and to enable a large number of parallel computations with efficient real-time performance [4–6]. Depending on the used technology, the state of the art neuro-computing platforms can be divided into analog, digital, mixed signal, and memristor-based systems [1,7,8]. Several research groups around the world are developing custom hardware systems for simulating large-scale neural models [1,9,10].

So far, these systems have limitations in applications that require accurate calculations. For instance, simulating networks with reinforcement learning capabilities [11] or convolutional spiking neural networks for pattern recognition rely on precise values of weights [12,13]. Today, research on designing neuromorphic systems for those applications that require precise calculations and high precision of neuronal parameters has a high priority in the field [3,14–22]. Networks with reduced precision of weights show severely reduced performance in many applications. For instance, only 68% accuracy on the ImageNet dataset after optimization for an 8-bit precision network was reported in [23]. Moreover, networks with course discretization of weights are more prone to adversarial attacks [24]. Complimentary to work on higher-precision efficient hardware implementation, as presented here, efforts on improving performance of low-precision networks have shown considerable progress recently [25–27]. Currently, these methods require off-chip processing during training and do not target online on-device learning in neuromorphic hardware.

Synapses and neurons are the basic structural elements of neuro-computing systems. The functionality of a silicon neuron in hardware models is equivalent to the soma of biological neurons. Silicon neurons receive spikes from other neurons via one or more synapses, integrate the input signals, and generate output spike events [28]. Synapses –either plastic or static– are the second crucial block in brain-inspired neural networks. In biology, plastic synapses rely on complex chemical reactions to adapt their strength based on pre-synaptic and post-synaptic activities. These short-term or long-term modifications bridge the short timescales of neural signals and timescales of the learning tasks [29–31]. Several attempts have been made to develop synaptic plasticity in Very Large Scale Integration (VLSI) technologies [31,32], e.g., analog plastic synapses [33,34], mixed-signal learning rules [35], memristor-based adaptation algorithms [36,37], and digital models [18,38–40]. While analog (mixed-signal) VLSI and memristor-based devices often require less area and show lower power consumption, they are not easily reconfigurable for studying different algorithms. Reconfigurability and flexibility of FPGAs, to the contrary, make them a better suitable substrate for studying different neurally based models [14,18,41,42]. In digital neural models, efficient methods for implementing bio-inspired circuits have been developed. Several researchers suggested piece-wise linear approximations, memory look-up tables, and shift-add multiplication techniques for reducing implementation complexities of neuromorphic designs [14–20,43].

Following these research lines, in this study we present digital multiplier-less neuromorphic realizations of two types of the long-term plastic synapses. In one of them, the synaptic efficacy alters based on the post-synaptic depolarization and the calcium concentration in the post-synaptic neuron on the arrival of a pre-synaptic spike (SSSTDP model). In the other model, the timing difference between the pre-synaptic and post-synaptic spikes determines the direction of synaptic adaptation, while the amount of adaptation depends on the initial synaptic strength (SDSP model) [11,30].

The proposed synapse models facilitate hardware implementation of spiking neural network applications with more precise calculations. The dynamics of the adaptation rules include exponential terms. Since exact exponential functions are non-synthesizable in FPGA, direct implementation of the original plastic synapse causes high implementation overheads. Furthermore, the multiplier is an expensive digital block in terms of area, latency, and power consumption.

Therefore replacing exponential and multiplier terms with more hardware friendly blocks can significantly reduce digital design complexity. This study proposes several novel digital architectures for the SDSP and SSSTDP rules and presents an estimation of required hardware resources and their speed. Taking advantage of the pre-evaluated optimization methods, the proposed multiplier-less models can be used in studying spiking neural network models which need higher resolutions of activation and weight values.

The paper starts with a description of the two plasticity dynamic equations in Section 2. After that, Section 3 introduces the proposed implementation methods for SDSP and SSSTDP. Implementation results and validation in a small neuronal network are presented in Section 4. Finally, Section 5 concludes the paper.

2 | DYNAMICS OF THE LEARNING RULES

In the neuromorphic field, Spike-Timing-Dependent-Plasticity (STDP) is a widely used learning rule. This plasticity rule provides an online learning mechanism according to a Hebbian-like biologically inspired learning scheme [5]. In the computational neuroscience, several models of STDP have been proposed [1,44]. Some of these models do not rely just on the spike-timing difference. Sometimes other factors such as the timing of the pre-synaptic spike, the state of the post-synaptic neuron's membrane potential, synaptic strength, and post-synaptic cell type are taken into account [30,45]. Next, two learning rules that use such factors are presented. In one of them, adaptation depends on the pre-synaptic spike arrival and the amount of post-synaptic neuron activity. In the other rule, the synaptic weight value together with the spike-timing difference affects the synaptic modification.

2.1 | The Spike-Driven Synaptic Plasticity Rule

This section considers a plastic synapse model in which synaptic weight starts adapting at the arrival of pre-synaptic spikes. The change in synaptic efficacy is a function of post-synaptic depolarization and the recent post-synaptic spiking activity [45]. The Spike-Driven Synaptic Plasticity (SDSP) learning rule is a biologically realistic rule and thus compatible with implementation constraints on neuromorphic devices. This rule is also competitive among the state-of-the-art spike-based machine learning methods [1,10,43]. According to this rule, every time the post-synaptic neuron emits a spike, an internal variable Ca^{2+}, which represents calcium concentration and is proportional to the neuron's recent spiking activity, is incremented by a value J_c and then decays with a time constant τ_{Ca}, according to the dynamics of Equation (1):

$$\frac{dCa^{2+}(t)}{dt} = -\frac{1}{\tau_{Ca}} Ca^{2+}(t) + J_c \sum_k \delta(t - t_k), \tag{1}$$

where t_k is the emission time of the k_{th} spike. The simplified governing equations of the synaptic efficacy modification for a given synapse at the arrival of each pre-synaptic spike are as follows:

$$X = \begin{cases} X + \Delta X_P : \vartheta_m < V_{m^{Post}}(t_{pre}), \vartheta_P^l < Ca^{2+}(t_{pre}) < \vartheta_P^h; \\ X - \Delta X_D : \vartheta_m > V_{m^{Post}}(t_{pre}), \vartheta_D^l < Ca^{2+}(t_{pre}) < \vartheta_D^h, \end{cases} \tag{2}$$

where X represents an internal variable; the terms ΔX_P and ΔX_D determine the amplitude of potentiation and depression; $V_{m^{Post}}(t_{pre})$ represents the post-synaptic neuron's membrane potential at the arrival time of the pre-synaptic

spike; and ϑ_m is a threshold value that determines whether the weight should be increased or decreased. The terms ϑ_P^l, ϑ_P^h, ϑ_D^l, and ϑ_D^h are threshold values that determine in which conditions the weights are allowed to increase, decrease, or should not be updated. Along with the instantaneous adaptations, the internal variable of the synapse, X, is continuously driven toward one of two stable states, depending on whether the current value is above or below a given threshold ϑ_X:

$$\frac{dX}{dt} = \begin{cases} \alpha, & \vartheta_X \leqslant X < X_{Max}, \\ -\beta, & X_{Min} < X < \vartheta_X, \end{cases} \tag{3}$$

where α and β are the rates of the constant increase and decrease respectively, at which the synapse is driven to its high and low bounds (X_{Max} and X_{Min}). Consequently, the synaptic weights are bounded, and on long time-scales, they converge to either a high or a low state. All related parameters for the SDSP synapse are extracted from [45] and presented in Table 1. Figure 1 shows how this learning rule works for LTD in more detail. As shown in the figure, at the pre-synaptic spike's arrival, whenever the calcium concentration and membrane voltage of the post-synaptic neuron meet the weight update requirements, LTD happens. The internal variable X is not the synapse output. The actual output of the synapse is a shifted sigmoidal function output of the internal variable X. This function is used to produce the Excitatory Post-Synaptic Potential (EPSP) on the arrival of the pre-synaptic spike:

$$V_{post}(X) = V_{Max} f(X, \vartheta_X), \tag{4}$$

where f(X,ϑ_X) can be a sigmoidal or hard-threshold function and V_{Max} is the maximum synaptic efficacy [1].

2.2 | Synaptic Strength-Based STDP Rule

In the STDP algorithm, if a pre-synaptic spike arrives several milliseconds before the post-synaptic spike, the positive timing difference leads to the synaptic long-term potentiation (LTP). On the other hand, if the post-synaptic neuron fires before the pre-synaptic neuron, the timing difference is negative, and the synaptic long-term depression (LTD) occurs. This section considers synaptic strength-based STDP learning algorithm (SSSTDP), in which the timing difference Δt determines the direction of the modification (and whether a modification happens at all), while the synaptic state determines the magnitude of the change: strong LTP is induced in synapses with low initial strength while strong LTD happens in synapses with high initial strength [30]. The following equations describe the synaptic modification model [11,46]:

$$\tau_w \Delta W = \begin{cases} (W_{Max} - W) A^+ e^{(\Delta t/\tau_+)}, & \Delta t > 0, \\ (W_{Min} - W) A^- e^{(\Delta t/\tau_-)}, & \Delta t < 0, \end{cases} \tag{5}$$

where W, W_{Max}, and W_{Min} determine the synapse's amplitude and the dynamic range of synaptic strength variation. Δt is the difference between the arrival time of the pre-synaptic and post-synaptic spikes. The time constants τ_+, τ_-, and τ_w determine rates of exponential decay of LTP, LTD, and weight adaptation, respectively. In biology, the time-scale of these effects is within a time window of 20 ms. The amplitude for depression (A^-) is defined as a ratio of the amplitude for potentiation ($A^- = 1/3A^+$). All parameters are extracted from [11] and presented in Table 1. Dependency between synaptic weight and synaptic modification is shown in Figure 2. In the smaller synaptic weights, LTP is stronger than LTD. When the synaptic weights increase enough, LTD becomes stronger than LTP. The output of the synapse is computed according to Equation (4).

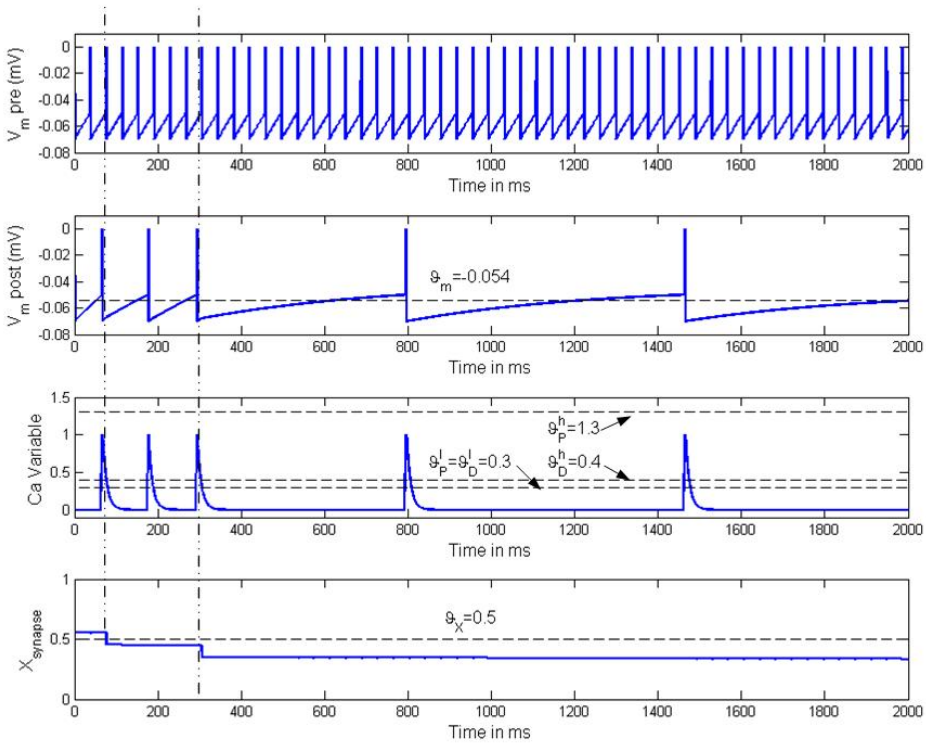


FIGURE 1 SDSP synapse behaviour during LTD based on the selected parameters. If calcium concentration and post-synaptic membrane voltage meet the requirements at the moment of the pre-synaptic spike arrival, LTD or LTP (LTD in this example) happens.

3 | PROPOSED ARCHITECTURES

3.1 | Proposed Model For SDSP

This section presents the digital architecture for implementing a single SDSP synapse. State-of-the-art synapse models for large scale neural networks [39] present synaptic weights using up to four bits. In comparison, the proposed architecture is suitable for applications that require more accurate calculations and need a higher-precision learning memory [11,12]. As the proposed SDSP uses 32-bit registers in fixed-point format –one bit can be chosen for representing fixed part and 31 bits for representing fractional part– the order of accuracy in synaptic weight adaptation can be given in terms of 2^{-31}. To describe the proposed SDSP model functionality, a minimum-size neural network is considered. Figure 3(A) shows the high-level schematic of the minimal-size neural network. This network includes a proposed plastic synapse and two LIF neurons as pre- and post-synaptic neurons. In the shown architecture, an SDSP synapse (SDSP block) connects a pre-synaptic neuron (N_{pre}) to a post-synaptic neuron (N_{post}). Besides reacting to the arrival of pre-synaptic spikes, SDSP block needs the post-synaptic neuron's membrane potential and an estimation of the post-synaptic neuron's calcium concentration (Ca[n]). Therefore, SDSP block receives the pre-synaptic input spikes,

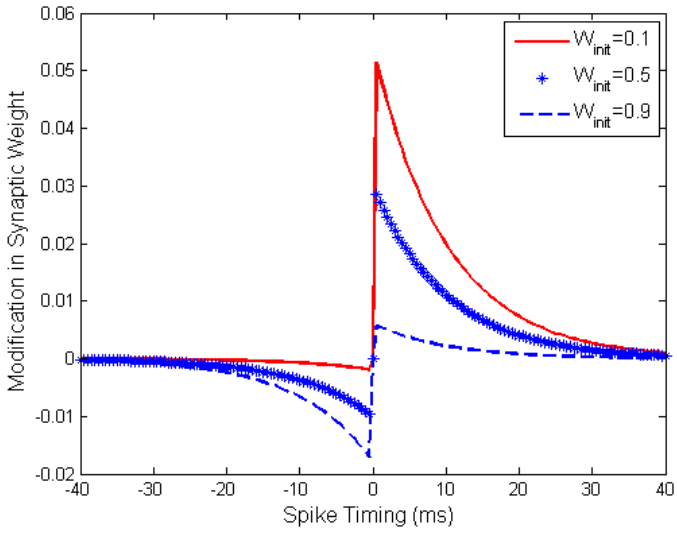


FIGURE 2 Modification in synaptic weights in SSSTDP learning rule. The weight modification depends on the current (initial) synaptic weight (W_{init}). Strong LTP happens for lower synaptic weights while strong LTD happens for higher synaptic weights.

TABLE 1 The parameters list of the synapse models [11,45]

SDSP		SSSTDP	
Parameter	Value	Parameter	Value
Vth and $Vreset$	50 and 70 (mV)	A^+	1.2
X_{Max} and X_{Min}	1 , 0	A^-	-0.4
ΔX_P and ΔX_D	$0.1X_{Max}$	τ_+	10 (ms)
τ_{Ca}	60 (ms)	τ_-	10 (ms)
α and β	$3.5 \times 10^{-3} X_{Max}$	τ_w	10 (ms)
ϑ_m	$Vreset + 0.8(V_{th} - Vreset)$ (mV)	W_{Max}	1.0
ϑ_X	$0.5X_{Max}$	W_{Min}	0.0
ϑ_P^l	0.3		
ϑ_P^h	1.3		
ϑ_D^l	0.3		
ϑ_D^h	0.4		
Δt	0.1 (ms)		

post-synaptic membrane voltage, and post-synaptic calcium concentration as inputs. Its output is the post-synaptic potential. As SDSP model operation depends on the post-synaptic calcium concentration, the post-synaptic neuron should include a core for estimating its activity. The proposed simplified equation for modelling the recursive behavior of the calcium variable Ca^{2+}, Equation (6), is based on the calcium dynamics of Equation (1):

$$Ca^{2+}[n+1] = Ca^{2+}[n] \times J_c e^{-\frac{\Delta t}{\tau_{Ca}}}. \tag{6}$$

Figure 3(B) illustrates the embedded Calcium Variable Estimator (CAVE) block in the post-synaptic neuron. According to Equation (6), whenever the post-synaptic neuron fires, Ca^{2+} is increased by J_c, and then decays with a time constant τ_{Ca}. Since for a constant time step Δt, the value of $e^{-\frac{\Delta t}{\tau_{Ca}}}$ is a constant value, Equation (6) can be implemented simply using shift-and-add operations.

As shown in Figure 3(A), the post-synaptic neuron includes a CAVE block, while the SDSP synapse block consists of two sub-blocks: the Internal Variable Adaptor (IVA) and the exitatory post-synaptic potential generator (EPSP). In the proposed synapse block, the internal variable X is modified in the IVA block based on the Equations (2) and (3). The extended IVA block is depicted in Figure 3(C). Whenever the pre-synaptic neuron fires, the post-synaptic neuron's membrane potential ($V_{mPost}(t_{pre})$) is compared with ϑ_m. If $V_{mPost}(t_{pre})$ is larger than ϑ_m, then for LTP to happen, the value of Ca[n] should meet both margins ϑ_P^l and ϑ_P^h. If $V_{mPost}(t_{pre})$ is smaller than ϑ_m, then for LTD to happen, the value of Ca[n] should meet margins ϑ_D^l and ϑ_D^h. In both cases, if Ca[n] does not meet the margins, there is no potentiation or depression. Regardless of any potentiation and depression, there is an instant drift for the internal variable X[n]. If X[n] is bigger than ϑ_X, there is a positive drift (α). For X[n] smaller than ϑ_X, there is a negative drift ($-\beta$).

The local variable X[n] produces the post-synaptic neuron's potential using the EPSP block. Figure 3(D) shows the digital implementation of a simple linear approximation of the thresholded sigmoidal function as the EPSP block. The EPSP block receives X[n] as input and generates the post-synaptic neuron's input voltage. Note that there are memory registers at the output of the blocks (B) and (C), storing the outputs and feeding them back for the next iteration.

3.2 | Proposed models for SSSTDP

As described in Section II, the original SSSTDP learning algorithm contains multiplications and exponential functions. Since multipliers and dividers are expensive digital building blocks, implementation of the original model leads to a high area overhead. Proposed hardware models in this section aim to remove multipliers and exponential parts of the synapse's dynamical equations and approximate them with simple shift and add operations.

3.2.1 | The Look Up Table Model

Table 1 determines the SSSTDP parameters, using which we pre-compute the ΔW based on different values of Δt and W and load it into a look-up-table (LUT). Utilizing a look up table approximation is more efficient compared to common hardships of real-time computations of exact ΔW based on original equations.

3.2.2 | The Piecewise Linear Models

Learning parameters in several applications of spiking neural networks with high-accuracy calculations take small values. For instance, in a spiking deep convolutional neural network [12], synaptic weight values are positive and maximally take 1 as synapse strength. In the mentioned network, values of the synaptic weights' modifications are on the order

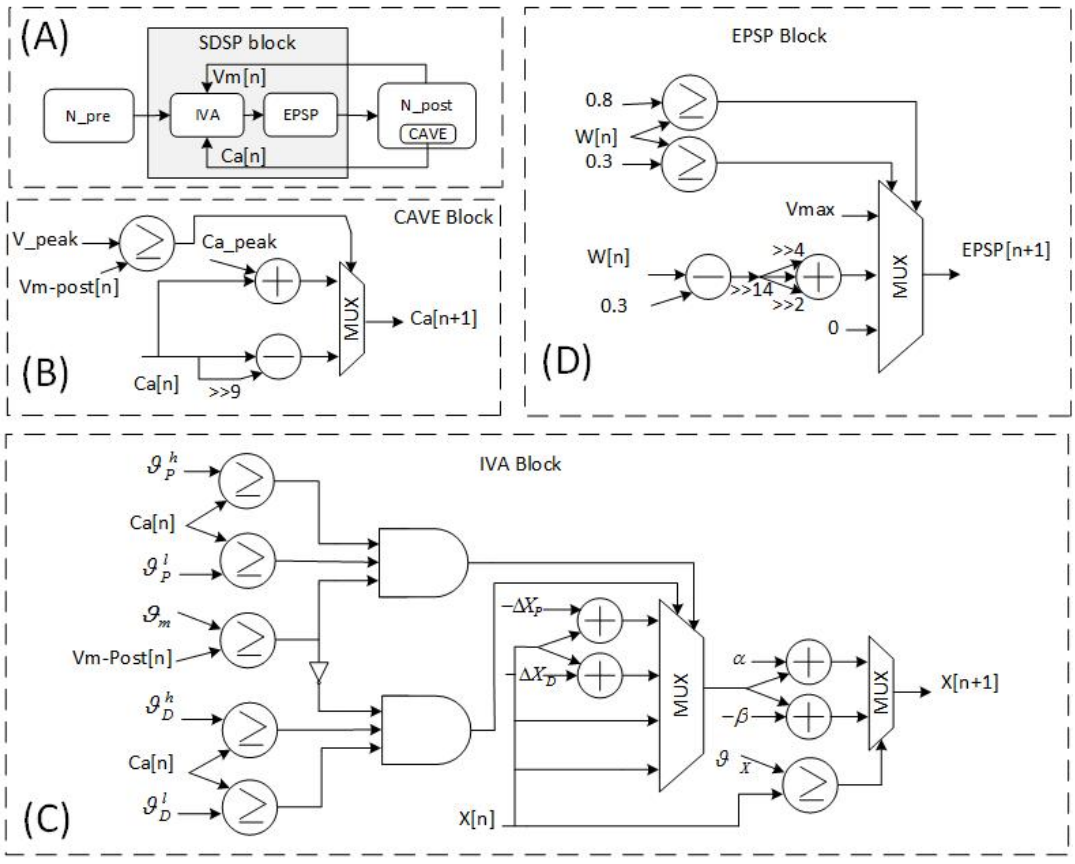


FIGURE 3 (A) High-level schematic of the minimal-size neural network. N_{pre} is the pre-synaptic neuron and N_{post} is the post-synaptic neuron. Pre- and post-synaptic neurons are connected through one SDSP synapse. (B) The proposed architecture for calcium variable estimator (CAVE block) embedded in the post-synaptic neuron. (C) The proposed block for calculating the X variable (IVA block). (D) The proposed simple linear approximation of a thresholded sigmoidal function (EPSP block).

of 0.001. These small adaptation values lead to an increase in the learning memory required by SNNs. This leads to the conclusion that to support small synaptic variations and high learning memory on hardware, a high resolution of bits is needed. To modify critical parts in the digital design and reduce implementation overhead of the SSSTDP algorithm along with acceptable precision, three piecewise linear (PWL) models are designed. The proposed models aim to substitute the exponential parts of the synapse equations by implementable PWL functions. The proposed models do not need implementing multipliers or exponential terms and provide the required resolution for applications that need high-precision plastic weights. In this work, the 2PWL, 4PWl, and 6PWL models divide the domain of function into 2, 4,

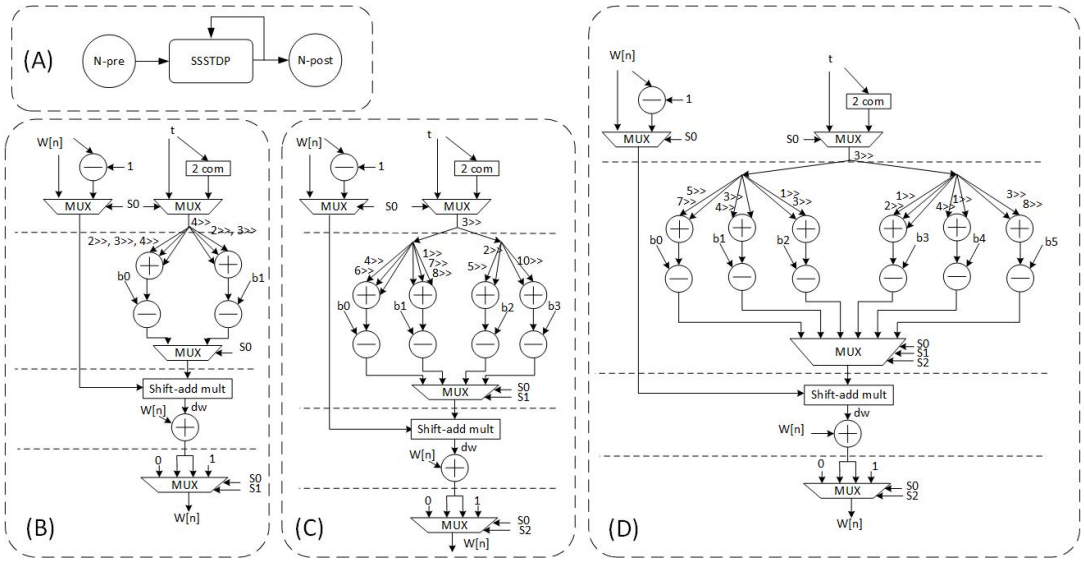


FIGURE 4 Flow diagram of the proposed PWL approximated models: (A) High level schematic of the proposed synapse models in a minimum size neural network. (B) $SSSTDP_{2PWL}$. (C) $SSSTDP_{4PWL}$. (D) $SSSTDP_{6PWL}$.

and 6 segments, respectively. The proposed $SSSTDP_{PWL}$ equations are given as follows:

$$\frac{dW}{dt} = \begin{cases} (W_{Max} - W)F(\Delta t), & \Delta t > 0\,, \\ 0, & \Delta t = 0\,, \\ (W_{Min} - W)F(\Delta t), & \Delta t < 0\,, \end{cases} \tag{7}$$

where $F(\Delta t)$ is a linear function. Here, $F(\Delta t)$ has three different shapes, that can be formulated as $F_{nPWL}(\Delta t) = m_i \times |\Delta t| + b_i$, for $i = \{0, 1, \ldots n-1\}$, where m_i and b_i are the slope and intercept of F, respectively. All the pre-calculated slopes and F-intercepts for $SSSTDP_{nPWL}$ are given in Table 2. Obviously, the accuracy of the approximations increases as a result of applying additional breakpoints, for which greater computational overhead would be expected. To describe

TABLE 2 Coefficients of the proposed PWL models.

$SSSTDP_{2PWL}$		$SSSTDP_{6PWL}$	
m_0, b_0	-4.72×10^{-4}, 0.0192	m_0, b_0	-8.22×10^{-5}, 0.0037
m_1, b_1	-0.0014 , 0.0578	m_1, b_1	-3.86×10^{-4}, 0.0113
$SSSTDP_{4PWL}$		m_2, b_2	-0.0012 , 0.0196
m_0, b_0	-1.65×10^{-4}, 0.007	m_3, b_3	-0.0037 , 0.0589
m_1, b_1	-1.0×10^{-3}, 0.0195	m_4, b_4	-0.0011 , 0.0336
m_2, b_2	-0.003 , 0.0586	m_5, b_5	-2.533×10^{-4}, 0.0112
m_3, b_3	-4.9×10^{-4}, 0.021		

the functionality of the proposed digital systems, the data flow diagrams of the proposed $SSSTDP_{PWL}$ models are illustrated in Figure 4. Furthermore this figure shows the proposed synapse model inside a simple neural network. The approximated PWL Equations (7) contain two multipliers. For multiplication, the shift-add based multiply operations are employed. This structure facilitates applying a pipeline technique to the neural network implementation, which uses more registers, but increases the speed of clock frequency.

4 | IMPLEMENTATION RESULTS AND DISCUSSION

This paper presents several digital hardware models for two plastic synapse rules: SDSP and SSSTDP. The proposed models avoid using any multiplier due to their cost overheads. Application targets of the proposed synapses are those with high precise calculations that need a higher resolution of weights to achieve good performance. For example, in [11] and [12] all synaptic weights have fractional values in the range of [0, 1] and need very small synaptic variations. For presenting small synaptic variations on the order of $10^{(-3)}$ or $10^{(-4)}$, high bit-resolution values are needed. As a proof of concept, the proposed SDSP, the minimal-size neural network architecture for examining the proposed SDSP, and the different models of SSSTDP are implemented on Atlys circuit board including Xilinx Spartan-6 LX45 FPGA. All the architectures were synthesized and simulated in ISE Xilinx tool and data stored in different files and analyzed in Matlab environment.

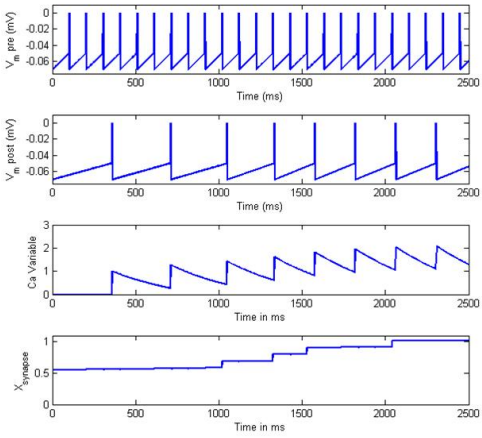
Figure 5 illustrates the behavior of SDSP synapse in a minimal-size network containing two neurons and a synapse during LTP and LTD, respectively. In Figure 5 (left), the synaptic weight is increased based on firing activities of the pre- and post-synaptic neurons and the post-synaptic neuron's calcium concentration value. When the synaptic weight value saturates, no more increases happen in the synapse. Figure 5 (right) shows that based on the pre- and post-synaptic neuronal activity, according to Equation (2), the synaptic weight is depressed. The dependence of the synaptic update values on the spike-timing difference for all multiplier-less SSSTDP models is presented in Figure 6. All proposed models are hardware friendly and have enough resolution for high precision applications.

Table 3 presents the device utilization of the proposed SDSP and SSSTDPs models compared to evaluations of previous studies on implementing other synapse models. Some of the previous synapse models were designed to be used in large scale neural networks. Some of them used more precise models than the ones introduced here. The used FPGA devices and synthesized tools are different in different works. Therefore, the result of Table 3 should not be used to compare different synapse model but as an overview of similar work.

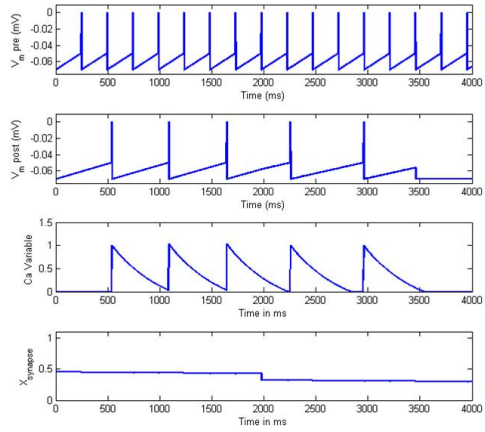
For examining the proposed SDSP model, a minimum-size neural network is used. Table 4 summarizes the utilized resources for implementing this network in comparison with other studies on calcium-based synapse models on similar substrate. Hardware implementation results show that the proposed SDSP implementation and the neural network models have low computing cost compared to other models, while showing a decrease in the maximum operating frequency.

The proposed SSSTDP hardware models approximate the original synapse model of Equation (7). Usually the root mean square error (RMSE) and normalize root mean square error (NRMSE) factors are used to specify the differences between the original model and the approximated one. In this paper we used the following error factors:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(W_{original} - W_{estimated})^2}, \tag{8}$$



Left (LTP). From top to bottom: pre-synaptic membrane potential, post-synaptic membrane potential, calcium variable, and internal synaptic variable.



Right (LTD). From top to bottom: pre-synaptic membrane potential, post-synaptic membrane potential, calcium variable, and internal synaptic variable.

FIGURE 5 Output of the proposed SDSP synapse in the minimal-size neural network model implemented on the XILINX Spartan-6.

$$NRMSE = \frac{RMSE}{(W_{max} - W_{min})}. \tag{9}$$

Based on the results from Table 5, the RMSE and NRMSE of the approximations decrease as a result of applying additional breakpoints, which, on the other hand, require us to use more resources and lower the maximum operating frequency.

Although Tables 3, 4, and 5 all present results of the proposed models in 32-bit resolution, it is straight-forward to change the bit resolution in our synapse models based on the requirements of the target application. To show the effects of changing the bit resolution on the results, one of the proposed models ($SSSTDP_{LUT}$) is also implemented with 8- and 16-bit resolutions. Figure 7 shows the synthesized result and the estimated error for the proposed LUT-based model over three different bit precision values: 8, 16 and 32. By increasing the resolution, the accuracy is increasing, while the design becomes slower. These results show that for a more precise model, more resources are utilized. As shown in Figure 7, the accuracy can be achieved using 32 bits resolution is slightly higher than accuracy with 16 bits. Having a possibility to choose different bit resolutions is important to explore a wide range of applications – those that require accurate calculations and those that may profit from a different trade-off between precision and resources used.

For instance, in our recent work on implementing a spiking neural network with reinforcement learning (RL) for learning a context-dependent task[47,48], the 3-layer SNN contained two Winner Take All (WTA) networks. Implementing a soft WTA requires a subtle balance between lateral excitation and inhibition in a network, in order to achieve dynamical properties such as sustained activation or multi-bump solutions[49]. These networks require shaping sum-of-Gaussians patterns of lateral connections. Although WTA behavior can be achieved with a low resolution of weights, tuning the dynamics of the network in real-world applications is facilitated if precise weights values can be used. More importantly, a high resolution of plastic weights was required for the SNN to model the learning process in the biological

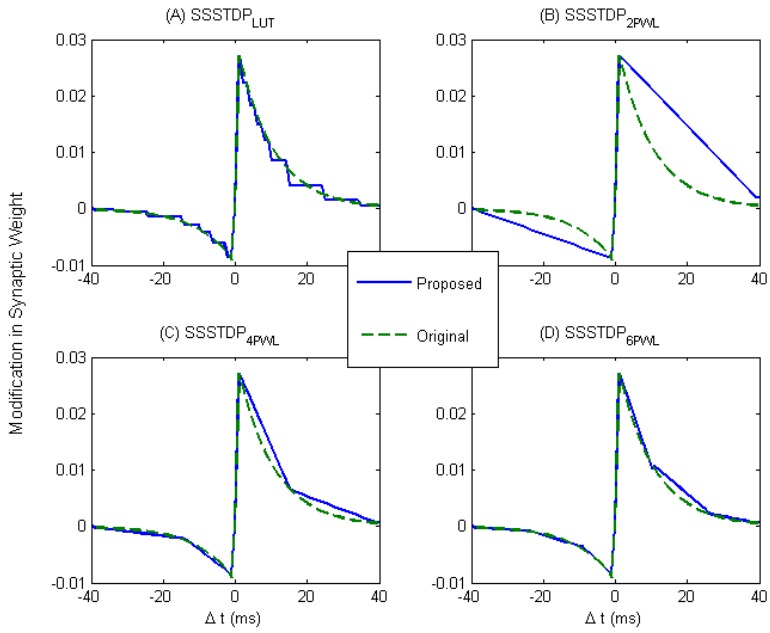


FIGURE 6 Comparison of original synapse model with Output of the proposed models implemented on the XILINX Spartan-6 (XC6SLX45) FPGA development board. (A) $SSSTDP_{LUT}$. (B) $SSSTDP_{2PWL}$. (C) $SSSTDP_{4PWL}$. (D) $SSSTDP_{6PWL}$.

RL experiment [50]. Learning the task in these experiments requires hundreds of trials and the increments on individual weights are small. High resolution of weight increments allowed us to reproduce biological learning curves using the hardware SNN model [47,48].

In order to demonstrate the proposed models in a practically more relevant setting, we use $SSSTDP_{LUT}$ model in a larger neural network. As shown in Figure 8(A), we implemented a two layers feed-forward network containing 22 integrate-and-fire neurons. The two layers are connected with 112 exitatory synapses. Each neuron in the output layer is connected to the other neurons in the same layer using fixed strong inhibitory synapses. Therefore, 56 inhibitory synapses with strong fixed values construct a winner-take-all in the output layer (inhibitory connections for one neuron are depicted in Figure 8(A)). Using the plastic hardware synapse model, this network can learn different network structures on-chip. The proposed synapses can be used for supervised, unsupervised, and reinforcement learning (RL) algorithms in spiking neural networks. For instance, we have trained an SNN in a supervised way with a set of training input and output pairs. Figure 8(B) and (C) show active neurons in input and output layers during each training trial. For example, neuron N10 in the input layer and neuron N14 in the output layer are active during Trial 1. The network learns a sinusoidal function between input and output layers during nine trails.

Figure 8(D) shows network behavior and the neurons' activity raster plots after the training phase. Neurons N0 to N13 belong to the input layer while neurons N14 to N21 belong to the output layer. Different active neurons in the input layer make just one of the output layer's neurons to fire (we use WTA connectivity in the output layer). Figure 9(A) shows the values of all synaptic weights in this network before and after learning. In the beginning, all synapses are randomly initiated. After training, some of the synapses between input and output layers are potentiated and others

remain unchanged. In the network structure, each neuron of the input layer is connected to the output layer using eight plastic synapses (all-to-all connectivity between layers). As an example, we show change over time of all synapses that connect neuron N1 to the output layer during training phase in Figure 9(B). Based on the training patterns, one of the synaptic weights increases.

Synapse modification consists of several LTP instances that are based on the pre- and post-synaptic spikes' orderings. In the proposed models, synaptic weights remain constant after learning which is a useful feature in SNNs substrates. To show the obtained resolution, the synapse parameters have been set several times in Figure 9(C). One of the parameters that effects the learning time is the size of the buffer for input spikes. If pre- and post-synaptic events are stored longer, then the synaptic weights get potentiated more often. According to different learning times, the synaptic weight can settle on different values.

Keeping the synaptic variation constant over time, besides enabling a high-bit resolution, makes the proposed synapses suitable for using them in deep spiking neural networks for recognition tasks with on-chip learning capability. Moreover, in our previous study on implementing a spiking neural network with reinforcement learning capability, accurate calculations in synaptic modifications and membrane voltage adaptations were required [48]. Therefore, the proposed synapses have the potential for using in such RL-based network implementations.

TABLE 3 Device utilization and performance comparison for the implemented synapses modules. Abbreviations used in this table correspond to the circuit used for each plasticity rule. These abbreviations are: Slice Flip Flops (S.FFs), Slice LUTs (S.LUTs), Pair-based STDP (PSTDP), Minimal TSTDP Hippocampal (MTH), Minimal TSTDP Visual Cotrex (MTVC), Calcium-Based (CAB), Target Applications (TA), Large Scale Networks (LSN), High Precision Networks (HPN).

ref	Learning Rule	S.FFs	S.LUTs	F_{Max}(MHz)	TA	Device
[39]	PSTDP	39 (0.3%)	18 (0.13%)	-	LSN	Spartan-3 XC3S1500
[18]	CAB	292 (0.5%)	309 (1.13%)	332	HPN	Spartan-6 XC6SLX45T
[51]	PSTDP	46 (0.4%)	36 (0.6%)	138	-	Spartan-6 XC6SLX9
[51]	MTH	54 (0.47%)	41 (0.7%)	192	-	Spartan-6 XC6SLX9
[51]	MTVC	47 (0.4%)	26 (0.4%)	192	-	Spartan-6 XC6SLX9
[38]	PSTDP	398 (0.1%)	1430 (0.95%)	200	LSN	Virtex-6 XC6VLX240T
[41]	PSTDP	16 (0.1%)	8 (0.13%)	816	LSN	Spartan-6 XC6SLX9
[41]	PSTDP	671 (6%)	859 (7.36%)	362	HPN	Spartan-6 XC6SLX9
Proposed	SDSP	129 (0.23%)	265 (0.93%)	154	HPN	Spartan6
Proposed	$SSSTDP_{2PWL}$	209 (0.38%)	394 (1.44%)	141	HPN	Spartan6 XC6SLX45
Proposed	$SSSTDP_{4PWL}$	483 (0.88%)	910 (3.3%)	129	HPN	Spartan6 XC6SLX45
Proposed	$SSSTDP_{6PWL}$	552 (1%)	964 (3.5%)	121	HPN	Spartan6 XC6SLX45
Proposed	$SSSTDP_{LUT}$	164 (0.3%)	474 (1.68%)	271	HPN	Spartan6 XC6SLX45

TABLE 4 Device utilization summary of the SPARTAN-6 for the proposed SDSP synapse in minimal-size neural network

	[18]	This work
Slice Registers	808 (1.4%)	249 (0.45%)
Slice LUTs	723 (2.64%)	450 (1.64%)
F_{Max} **(MHz)**	322	131
Number of bits	32	32

TABLE 5 RMSE and NRMSE calculations for the proposed SSSTDP models

	2PWL	4PWL	6PWL	LUT
RMSE(%)	1.18	0.26	0.159	0.155
NRMSE (%)	16.31	3.587	2.193	2.146

5 | CONCLUSION

This paper presents several multiplier-less digital architectures for spike driven synaptic plasticity and synaptic strength-based STDP models in order to implement high precision spiking neural networks. The substitution of multipliers and exponential functions of original learning rules with simple arithmetic operations (addition/subtraction and shift) leads to a greater cost and area efficiency. The proposed models have been validated by implementing on the Xilinx Spartan-6 FPGA. This paper focuses on considering high precision synapse models implementation. In the next research, the proposed synapses will be used in spiking neural network models with reinforcement learning capability that requires high precision calculations.

REFERENCES

[1] Qiao N, et al. A reconfigurable on-Line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. frontiers in Neuroscience 2015;9(141):1–17.

[2] Moradi S, Qiao N, Stefanini F, Indiveri G. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs). IEEE Trans Biomedical Syst 2018;12(1):106–122.

[3] Brown AD, et al. SpiNNaker - Programming modele. IEEE Trans Computers 2015;64(6):1769–1782.

[4] Merolla PA, Arthur JV, Alvarez-Icaza R, Cassidy AS, Sawada J, Akopyan F, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science 2014;345(6197):668–673. `https://science.sciencemag.org/content/345/6197/668`.

[5] Schuman CD, et al. A survey of neuromorphic computing and neural networks in hardware. arXiv:170506963 2017;.

[6] Akopyan F, Sawada J, Cassidy A, Alvarez-Icaza R, Arthur J, Merolla P, et al. TrueNorth: design and lool flow of a 65 mW 1 million neuron programmable neurosynaptic chip. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 2015 Oct;34(10):1537–1557.

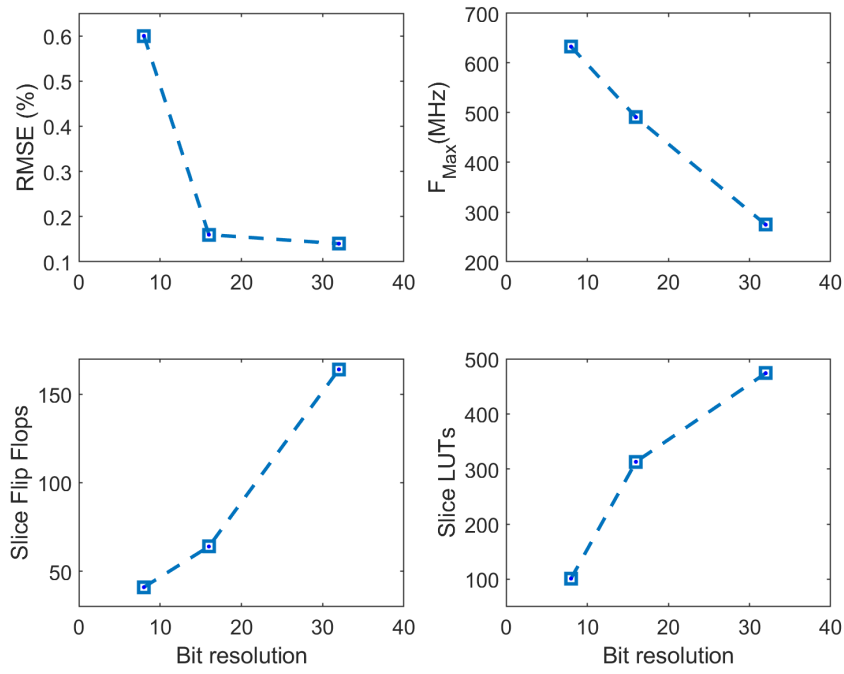


FIGURE 7 Root mean square error (RMSE%) and the synthesis result of the proposed $SSSTDP_{LUT}$ model over different bit resolutions. This demonstration shows how the performance of the proposed models degrades when increasing the resolution of the synapses from 8 to 32 bits. Furthermore, the synthesis results show that for a higher resolution, more resources are used.

[7] Benjamin BV, et al. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. Proceedings of the IEEE 2014;102(5):699–716.

[8] Nawrocki RA, et al. A mini review of neuromorphic architectures and implementations. IEEE Trans Electron Devices 2016;63(10):3816–3829.

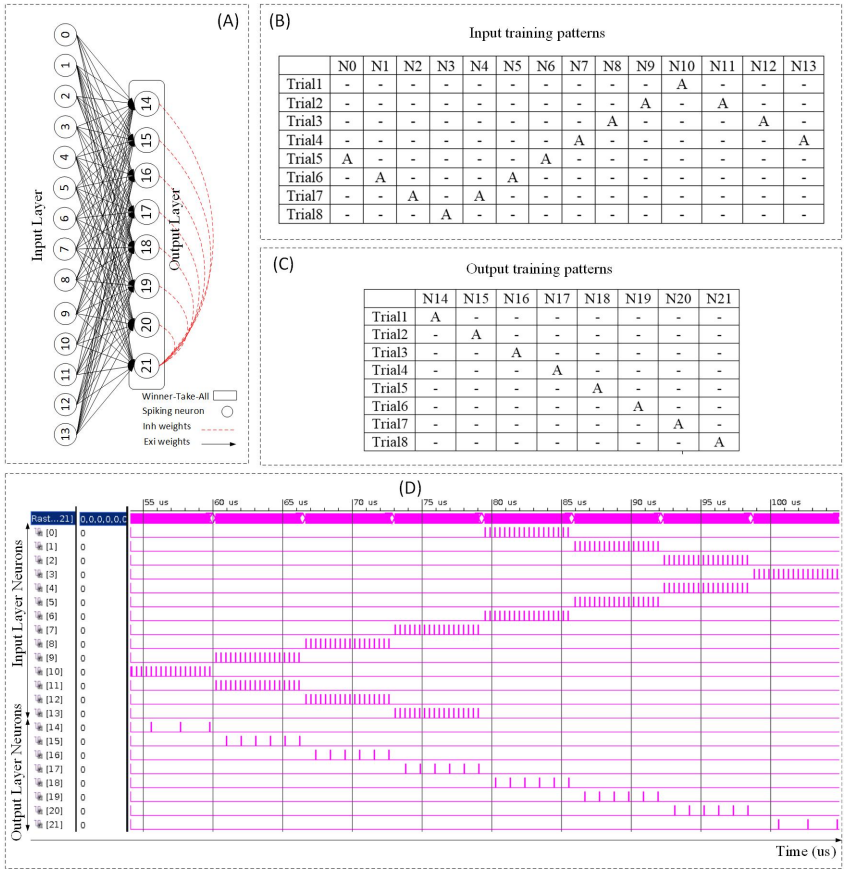
[9] Davies M, Srinivasa N, Lin T, Chinya G, Cao Y, Choday SH, et al. Loihi: A neuromorphic manycore processor with on-chip learning. IEEE Micro 2018 January;38(1):82–99.

[10] Frenkel C, et al. A 0.086-mm212.7-pJ/SOP 64k-synapse256-neuron online-learning digital spiking neuromorphic processor in 28nm CMOS system. IEEE Trans Biomedical Syst 2019;13(1):145 – 158.

[11] Raudies F, Hasselmo ME. A model of hippocampal spiking responses to items during learning of a context-dependent task. frontiers in System Neuroscience 2014;23(8):1–12.

[12] Kheradpisheh SR, Ganjtabesh M, Thorpe SJ, Masquelier T. STDP-based spiking deep convolutional neural networks for object recognition. Neural Networks 2018;99:56–67.

[13] Tavanaei A, Ghodrati M, Kheradpisheh SR, Masquelier T, Maida A. STDP-based spiking deep convolutional neural networks for object recognition. Neural Networks 2019;111:47–63.



(B) Input training patterns

	N0	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13
Trial1	-	-	-	-	-	-	-	-	-	-	A	-	-	-
Trial2	-	-	-	-	-	-	-	-	-	A	-	A	-	-
Trial3	-	-	-	-	-	-	-	-	A	-	-	-	A	-
Trial4	-	-	-	-	-	-	-	A	-	-	-	-	-	A
Trial5	A	-	-	-	-	-	A	-	-	-	-	-	-	-
Trial6	-	A	-	-	-	A	-	-	-	-	-	-	-	-
Trial7	-	-	A	-	A	-	-	-	-	-	-	-	-	-
Trial8	-	-	-	A	-	-	-	-	-	-	-	-	-	-

(C) Output training patterns

	N14	N15	N16	N17	N18	N19	N20	N21
Trial1	A	-	-	-	-	-	-	-
Trial2	-	A	-	-	-	-	-	-
Trial3	-	-	A	-	-	-	-	-
Trial4	-	-	-	A	-	-	-	-
Trial5	-	-	-	-	A	-	-	-
Trial6	-	-	-	-	-	A	-	-
Trial7	-	-	-	-	-	-	A	-
Trial8	-	-	-	-	-	-	-	A

FIGURE 8 (A) A 2-layer spiking neural network. Input layer contains 14 neurons and output layer contains 8 neurons. Input layer is connected to output layer with all-to-all excitatory plastic synapses. Strong inhibitory connections among all neurons in the output layer create a Winner-Take-All network (inhibitory connections for one neuron are shown). (B) Input training patterns for learning a sinusoidal function. (C) Output training patterns for learning a sinusoidal function. (D) Neurons raster plot after training.

[14] Soleimani H, Ahmadi A, Bavandpour M. Biologically inspired spiking neurons: piecewise linear models and digital implementation. IEEE Transactions on Circuits and Systems I: Regular Papers 2012 Dec;59(12):2991–3004.

[15] Hayati M, et al. Digital multiplier-less realization of two coupled biological morris-lecar neuron model. IEEE Trans Circuits Syst II 2015;62(7):1805–1814.

[16] Hayati M, et al. Digital multiplierless realization of two coupled biological hindmarsh-rose neuron model. IEEE Trans Circuits Syst II 2016;63(5):463–467.

[17] Gomar S, Ahmadi A. Digital multiplierless implementation of biological adaptive-exponential neuron model. IEEE Trans Circuits Syst I 2014;61(4):1206–1219.

[18] Jokar E, Soleimani H. Digital multiplierless realisation of a calcium–based plasticity model. IEEE Trans Circuits Syst II 2017;64(7):832–836.

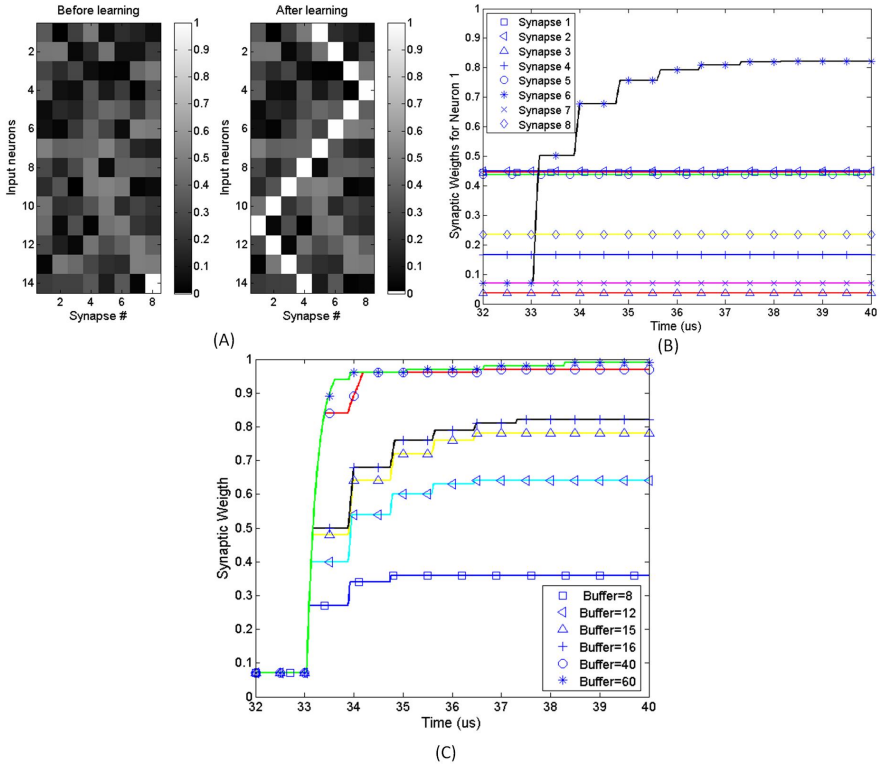


FIGURE 9 (A) All synaptic weights of the network (Fig. 9A) before learning (randomly initialized) and after supervised learning. (B) Weight change of all synaptic weights that connect neuron N1 in the input layer to all neurons of the output layer. (C) Behaviour of the high precision synapse model depending on the learning duration. The learning time is varied by changing the number of time steps for which input spikes are kept in the buffer.

[19] Hayati M, Nouri M, Haghiri S, Abbott D. A digital realization of astrocyte and neural glial interactions. IEEE Trans Biomedical Syst 2016;10(2):518–529.

[20] Soleimani H, Drakakis EM. An efficient and reconfigurable synchronous neuron model. IEEE Trans Circuits Syst II 2018;65(1):91–95.

[21] Karimi G, Gholami M, Farsa EZ. Digital implementation of biologically inspired Wilson model, population behavior, and learning. International Journal of Circuit Theory and Applications 2018;46(4):965–977.

[22] Amiri M, Nazari S, Faez K. Digital realization of the proposed linear model of the H odgkin-H uxley neuron. International Journal of Circuit Theory and Applications 2019;47(3):483–497.

[23] Rusci M, Capotondi A, Benini L. Memory-driven mixed low precision quantization for enabling deep network inference on microcontrollers 2019;http://arxiv.org/abs/1905.13082.

[24] Lin J, Gan C, Han S. Defensive quantization: when efficiency meets robustness 2019;p. 1–14. http://arxiv.org/abs/1904.08444.

[25] Gu J, Li C, Zhang B, Han J, Cao X, Liu J, et al. Projection convolutional neural networks for 1-bit CNNs via discrete back propagation 2018;http://arxiv.org/abs/1811.12755.

[26] Wiedemann S, Muller KR, Samek W. Compact and computationally efficient representation of deep neural networks. IEEE Transactions on Neural Networks and Learning Systems 2019;p. 1–14.

[27] Kim H, Sim J, Choi Y, Kim L. NAND-Net: minimizing computational complexity of in-Memory processing for binary neural networks. In: 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA); 2019. p. 661–673.

[28] Indiveri G, et al. Neuromorphic silicon neuron circuits. frontiers in Neuroscience 2011;5(73):1–23.

[29] Markram H, et al. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. Science 1997;275(5297):213–215.

[30] Bi GQ, Poo MM. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. J Neurosci 1998;18(24):10464–10472.

[31] Bartolozzi C, Indiveri G. Synaptic dynamics in analog VLSI. Neural Computations 2007;19(10):2581–2603.

[32] Indiveri G, et al. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. IEEE Transaction on Neural Networks 2006;17(1):211–221.

[33] Albrecht JMC, Yung MW, Srinivasa N. Energy-efficient neuron, synapse and STDP integrated circuits. IEEE Trans Biomedical Syst 2012;6(3):246–256.

[34] Friedmann S, et al. Demonstrating hybrid learning in a flexible neuromorphic hardware system. IEEE Trans Biomedical Syst 2017;11(1):128–149.

[35] Moradi S, Indiveri G. An event-based neural network architecture with an asynchronous programmable synaptic memory. IEEE Trans Biomedical Syst 2014;8(1):98–107.

[36] Azghadi MR, Barranco BL, Abbott D. A hybrid CMOS memristor neuromorphic synapse. IEEE Trans Biomedical Syst 2017;11(2):434–445.

[37] Ziegler M, et al. Memristive hebbian plasticity model: device requirements for the emulation of hebbian plasticity based on memristive devices. IEEE Trans Biomedical Syst 2015;9(2):197–206.

[38] Wang RM, et al. A neuromorphic implementation of multiple spike-timing synaptic plasticity rules for large-scale neural networks. frontiers in Neuroscience 2016;9(180):1–17.

[39] Cassidy A, Andreou AG, Georgiou J. A combinational digital logic approach to STDP. In: 2011 IEEE International Symposium of Circuits and Systems (ISCAS); 2011. p. 673–676.

[40] Frenkel C, Indiveri G, Legat J, Bol D. A fully-synthesized 20-gate digital spike-based synapse with embedded online learning. In: 2017 IEEE International Symposium on Circuits and Systems (ISCAS); 2017. p. 1–4.

[41] Lammie C, Hamilton TJ, van Schaik A, Azghadi MR. Efficient FPGA implementations of pair and triplet-based STDP for neuromorphic architectures. IEEE Trans on Circuits and Systems I 2019;66(4):1558–1570.

[42] Rahimian E, Zabihi S, Amiri M, Linares-Barranco B. Digital implementation of the two-compartmental pinsky–rinzel pyramidal neuron model. IEEE Transactions on Biomedical Circuits and Systems 2018;12(1):47–57.

[43] Galluppi F, et al. A framework for plasticity implementation on the SpiNNaker neural architecture. frontiers in Neuroscience 2015;8(429):1–19.

[44] Markram H, Gerstner W, Sjöström PJ. Spike-timing-dependent plasticity: a comprehensive overview. frontiers in Synaptic Neuroscience 2012;4(2).

[45] Brader JM, et al. Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. Neural Comput 2007;19(11):2881–2912.

[46] Kistler WM, v Hemmen JL. Modeling synaptic plasticity in conjunction with the timing of pre- and postsynaptic action potentials. Neural Computation 2000 Feb;12(2):385–405.

[47] Asgari H, Maybodi BMN, Kreiser R, Sandamirskaya Y. A digital multiplier-less neuromorphic model forLearning a context-dependent Task. In: 2NndIEEE International Conference ON Artificial Intelligence Circuits and Systems (AICAS); 2020. .

[48] Asgari H, et al. Digital multiplier-less event-driven spiking neural network architecture for learning a context-dependent task. arXiv:190609835 2019;.

[49] Sandamirskaya Y, Zibner SKU, Schneegans S, Schöner G. Using Dynamic Field Theory to extend the embodiment stance toward higher cognition. New Ideas in Psychology 2013;31(3):322–339. http://www.sciencedirect.com/science/article/pii/S0732118X13000111.

[50] Komorowski RW, Manns JR, Eichenbaum H. Robust Conjunctive Item-place Coding by Hippocampal Neurons Parallels Learning What Happens Where. Neural Comput 2009;29(31):9918–9929.

[51] Nouri M, Jalilian M, Hayati M, Abbott D. A digital neuromorphic realization of pair-based and triplet-based spike-timing-dependent synaptic plasticity. IEEE Transactions on Circuits and Systems II: Express Briefs 2018 June;65(6):804–808.