Programmable Spike-Timing Dependent Plasticity learning circuits in neuromorphic VLSI architectures

Mostafa Rahimi Azghadi*, University of Zurich and ETH Zurich and the University of Adelaide
Saber Moradi*, University of Zurich and ETH Zurich
Daniel B. Fasnacht, University of Zurich and ETH Zurich
Mehmet Sirin Ozdas, University of Zurich and ETH Zurich
Giacomo Indiveri, University of Zurich and ETH Zurich

Hardware implementations of spiking neural networks offer promising solutions for computational tasks that require compact and low power computing technologies. As these solutions depend on both the specific network architecture and the type of learning algorithm used, it is important to develop spiking neural network devices that offer the possibility to reconfigure their network topology and to implement different types of learning mechanisms. Here we present a neuromorphic multi-neuron VLSI device with on-chip programmable event-based hybrid analog/digital circuits; the event-based nature of the input/output signals allow the use of Address-Event Representation infrastructures for configuring arbitrary network architectures, while the programmable synaptic efficacy circuits allow the implementation of different types of spike-based learning mechanisms. The main contributions of this paper are to demonstrate how the programmable neuromorphic system proposed can be configured to implement specific spike-based synaptic plasticity rules and to depict how it can be utilised in a cognitive task. Specifically, we explore the implementation of different Spike-Timing Plasticity learning rules on-line, in a hybrid system comprising a workstation and when the neuromorphic VLSI device interfaced to it, and we demonstrate how after training the VLSI device can perform, as a stand-alone component (i.e., without requiring a computer), binary classification of correlated patterns.

Categories and Subject Descriptors: CCS [**Emerging technologies**]: Neural systems

General Terms: VLSI, emerging technologies

Additional Key Words and Phrases: Neuromorphic, STDP, asynchronous, AER, subthreshold, learning, plasticity, real-time

ACM Reference Format:
Mostafa Rahimi Azghadi, Saber Moradi, Daniel B. Fasnacht, Mehmet Sirin Ozdas and Giacomo Indiveri, 2014. Programmable Spike-Timing Dependent Plasticity learning circuits in neuromorphic VLSI architectures. *ACM J. Emerg. Technol. Comput. Syst.* 0, 0, Article 0 (2013), 17 pages.
DOI : http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Artificial spike-based neural networks offer a promising paradigm for a new generation of brain-inspired computational models. A wide range of theoretical and computational models have already been proposed for both basic neuroscience research [Kempter et al. 1999; Gerstner and Kistler 2002] and practical applications [Belatreche et al. 2006; Rowcliffe and Feng 2008]. Neuromorphic Very Large Scale Integration (VLSI) circuits represent an ideal technology for implementing these types of networks using hybrid analog/digital design techniques, and for building devices that have

This work was supported by the European Community's Seventh Framework Programme ERC grant # 257219 – "neuroP".
*Authors equally contributed to this work.
Author's addresses: M. Rahimi Azghadi, Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland and School of Electrical and Electronic Engineering, The University of Adelaide, Australia; Saber Moradi, Daniel B. Fasnacht, Mehmet S. Ozdas and Giacomo Indiveri, Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland.

DOI : http://dx.doi.org/10.1145/0000000.0000000

a very high potential in a wide range of applications such as pattern and data classification [Mitra et al. 2009; Giulioni et al. 2009; Schmuker et al. 2014], object recognition [Nere et al. 2012; Khosla et al. 2014], feature extraction [Vogelstein et al. 2007], and orientation selectivity [Choi et al. 2004; Chicca et al. 2007]. In particular, the main advantage of implementing these spiking neural networks in neuromorphic VLSI technology is their compactness and low power consumption which are critical features when implementing large scale neural architectures [Mead 1990; Chicca et al. 2014; Azghadi et al. 2014b].

In these types of networks, synapses represent an essential component for signal processing, as they are at the same time the site of memory (they store the network's synaptic weight values), and play a fundamental role in computation (they implement crucial temporal and non-linear dynamics). Synaptic weight values can be updated following the prescription of different types of learning algorithms that typically depend on the pre- and post-synaptic neuron activity [Abbott and Gerstner 2004; Brader et al. 2007; Graupner and Brunel 2012]. The different learning strategies have a profound effect on the post-synaptic neuron functionality and on the spiking-neural network behavior [Laughlin and Sejnowski 2003]. Implementing such types of synapses and learning mechanisms in compact electronic systems is essential for developing efficient large-scale spiking neural networks and brain-inspired computing technologies [Azghadi et al. 2013a; Azghadi et al. 2014a]. However, as the implementation of the learning algorithm often depends on the specific application domain and on the nature of the data to process, it can be useful to develop compact electronic implementation of spiking neural networks in which the weights can be adjusted by off-chip learning algorithms (e.g. implemented on a workstation, micro-controller, or Field Programmable Gate Arrays (FPGAs)).

In this paper we demonstrate how it is possible to train a hardware spiking neural network, using software-defined spike-based learning algorithms. We present a hybrid Software (SW) - Hardware (HW) neural processing system that comprises a mixed signal analog/digital spiking neural network VLSI device interfaced to a workstation. The custom VLSI device comprises analog silicon neuron circuits [Indiveri et al. 2011], analog synaptic dynamics elements, and asynchronous digital event-based interfacing circuits for transmitting and receiving spikes. The device also integrates asynchronous digital programmable synaptic weight circuits for setting and changing the strengths of the silicon synapses with off-chip learning algorithms that can implement different types of spike-based synaptic plasticity rules [Pfister and Gerstner 2006; Brader et al. 2007; Clopath et al. 2010; Graupner and Brunel 2012]. The mixed signal analog/digital neural network device used in this work has already been described in detail in [Moradi and Indiveri 2014]. Here we apply the device to spike based learning problems, describing the details of the experimental setup required for controlling and programming the proposed neural device. We interface the device to a workstation and show how the hybrid SW - HW system can implement different types of spike-based learning algorithms on-line. Finally, after training a perceptron architecture, we demonstrate how the device can act as an efficient stand alone binary classifier of correlated patterns, without requiring a workstation in the loop.

2. NEUROMORPHIC SPIKING NEURAL NETWORKS

The simulation of Spiking Neural Network (SNN) on standard computers can be very onerous, requiring large amounts of memory and/or CPU time [Azghadi et al. 2014b]. Attempts are being made to speed-up these simulations using Graphical Processing Units (GPUs) based approaches [Nageswaran et al. 2009; Fidjeland et al. 2009; Fidjeland et al. 2013], or dedicated computing architectures [Furber et al. 2013]. Alternatively, full custom VLSI implementations of spiking neural network can be implemented using dedicated analog, digital, or mixed signal analog/digital circuits. Examples of both fully digital [Arthur et al. 2012] and hybrid analog/digital [Schemmel et al. 2010; Moradi and Indiveri 2014] spiking neural network chips have been recently proposed. In these systems spiking neurons are typically implemented as Integrate-and-Fire (I&F) models [Koch 1999] which collect all of the signals (typically currents) produced by the synapses, and produce a spike event when the integrated sum exceeds a threshold. The equation governing the neuron's

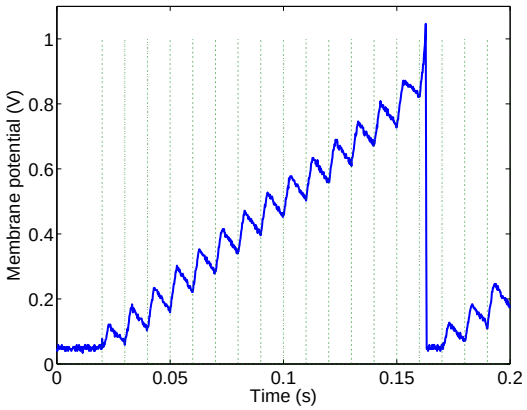


Fig. 1: Measured response of a silicon neuron, implementing a model of a leaky I&F neuron, being stimulated by a 100 Hz input spike train, via an excitatory synapse.

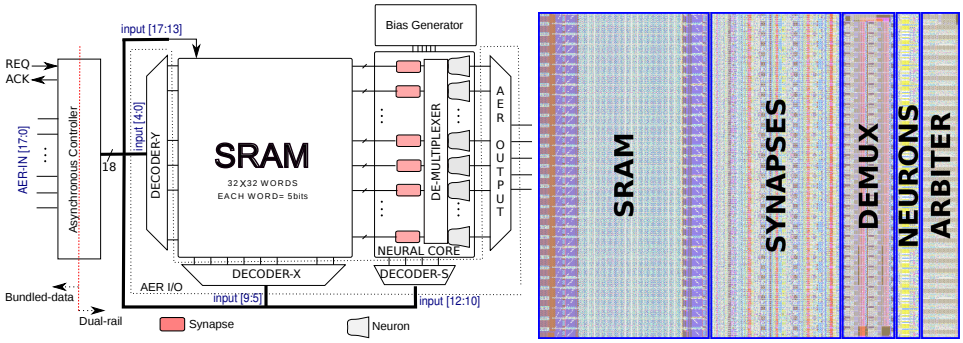
subthreshold dynamics, e.g. in the case of a leaky I&F model neuron [Gerstner and Kistler 2002], is given by:

$$\tau \frac{d}{dt} u(t) = -u(t) + RI(t) \tag{1}$$

where τ is the leaky integrator time constant, $1/R$ represents the neuron leak conductance, and $I(t)$ represents the total synaptic input current. Synaptic currents are produced by the neuron's synapses when they get stimulated by the input spikes. Figure 1 shows an example of the response measured from a silicon neuron being stimulated via an input excitatory synapse with a regular input spike train of 100 Hz.

Synapses also typically have first order dynamics; but they can have both linear and non-linear response properties. In all cases however their response is proportional to their synaptic weight. Learning occurs by adapting the synaptic weights of all the synapses afferent to a neuron, following the prescription of the specific learning rule considered. In these types of neural networks, the "source" neurons produce spikes and transmit them to synapses of other "destination" neurons. Depending on the network structure (e.g., multi-layer feed-forward architectures, recurrent architectures, etc.) different types of computational models can be implemented [Dayan and Abbott 2001]. Information is processed through the propagation of spikes and the generation of the weighted synaptic responses in the network [Gerstner and Kistler 2002]. It is widely believed that learning, computation and memory processes take place in synapses [Sjöström et al. 2008]. Since the learning and computation in a SNN is a dynamic process, the synapses should also be dynamic and modifiable. However, the open question is how these modifications take place in the synapses within the brain, and how they can lead to network properties that allow the system to carry out robust and real-time computation so efficiently and accurately. Although there is no general agreement as to the answer to this question, there are several hypotheses stating that these modifications take place in relation to the activity of pre- and post-synaptic neurons connected to the synapse [Sjöström et al. 2008]. These hypotheses that govern the synaptic weight changes, are so-called synaptic plasticity models [Mayr and Partzsch 2010].

Due to the variety of neuron, synapse and synaptic plasticity models, and because of different spiking neural network structures, it is essential to develop programmable neural network architectures, such as the one presented in this work device, to explore the role of different spike based learning rules and different neural network structures.



(a) (b)

Fig. 2: IFMEM chip block diagram. (a) The device comprises a neural-core module with an array of synapses and integrate-and-fire neurons, an asynchronous SRAM module to store the synaptic weight values, a bias generator to set the parameters in the analog circuits, and asynchronous control and interfacing circuits to manages the AER communication. (b) Layout picture comprising the SRAM, neural core and AER output blocks. In particular, the layout of the SRAM block measures $524 \times 930\ \mu$m; the synapse array measures $309\ \mu$m in length, the synapse de-multiplexer measures $132\ \mu$m, the neuron array $60\ \mu$m, and the output AER arbiter $105\ \mu$m.

3. THE "*IFMEM*" CHIP

The multi-neuron chip used in this work is characterized by the fact that it comprises circuits that implement models of I&Fs neurons, and a programmable memory for storing the synaptic weights. Therefore, we will refer to this device as the "*IFMEM*" chip. The *IFMEM* chip was fabricated using a standard $0.35\ \mu$m Complementary Metal Oxide Semiconductor (CMOS) VLSI technology and was fully characterized in [Moradi and Indiveri 2014]. It implements a neural network of 32 adaptive exponential I&F neuron circuits [Indiveri et al. 2011] with dynamic synapse circuits. The *IFMEM* chip makes use of the Address Event Representation (AER) protocol to receive and transmit events that represent input and output spikes, respectively.

A block diagram of the chip architecture is shown in Fig. 2a. All circuits on the chip that implement the neural and synapse dynamics are in the "Neural Core" block. The neuron circuits are implemented using an "adaptive exponential integrate and fire" model [Brette and Gerstner 2005; Indiveri et al. 2010], while the part of the synapse circuits responsible for integrating input spikes and producing temporal response properties that have biologically plausible time constants are implemented using a Differential Pair Integrator (DPI) circuit [Bartolozzi and Indiveri 2007]. Depending on the input address-event, different types of synapse dynamics can be triggered: excitatory with slow time constants (e.g., to emulate NMDA-type synapses), excitatory synapses with faster time constants (e.g., to emulate AMPA-synapses), or inhibitory synapses (e.g., to emulate GABA-type synapses). Since the DPI can be used as a linear low-pass filter, it is possible to make use of a single integrator circuit for any of the synapse dynamics considered (e.g., NMDA, AMPA, or GABA), and multiplex it in time to integrate the contributions from multiple spiking inputs (e.g., via multiple SRAM cells), thus saving precious silicon real-estate.

The analog components of these circuits have programmable bias parameters that can be set with an on-chip 32-bit temperature-compensated programmable bias generator [Delbruck et al. 2010]. The synaptic weights of the synapses are stored in a 32×32 5-bit digital SRAM block, designed with asynchronous circuits for interfacing to the AER components. The digital weight values are converted into currents with an on-chip Digital to Analog Converter (DAC), so that the addressed synapse circuits produce Excitatory Post Synaptic Current (EPSC) with amplitudes proportional to their weights. Thanks to the synapse time-multiplexing scheme, the total number of synapses that

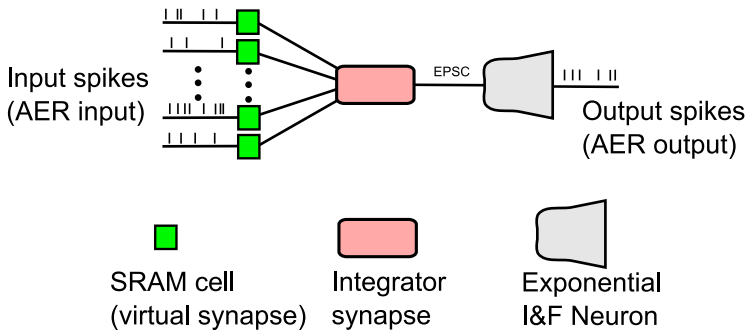


Fig. 3: A simple spiking neural network architecture implemented on the proposed neuromorphic device shown in Fig. 2a. The input spikes are integrated by a physical synapse that acts as an integrator. The synapse generates EPSC proportional to input firing rate and the synaptic weights stored on the SRAM cells, through on-chip programmable DACs. The generated EPSC will result in the silicon neuron to fire spikes if its membrane potential exceeds the spiking threshold. A similar structure has been used in all of the experiments presented in this paper.

a neuron sees is equivalent to the total number of SRAM cells present in each row. The SRAM cells can work in "feed-through" mode or in storage mode. In feed-through mode, input events contain both the address of the destination SRAM cell and the synaptic weight bits, and the synapses generate EPSC on-line, as the data is received. In storage mode, the input events contain only the address of the destination SRAM cell, and the weight bits used by the synapses are the ones stored in the addressed SRAM cell [Moradi and Indiveri 2011]. Therefore it is possible to interface the device to a workstation and use it in "feed-through" mode to train the spiking neural network on-line, with all of the HW components in the loop, eventually storing the final synaptic weight matrix in the SRAM block at the end of the training phase. Once the training has completed, it is possible to use the device in stand-alone mode, without requiring a PC in the loop, and use the stored weights to carry out the learned task.

Figure 2b shows a section of the layout of the *IFMEM* chip comprising the main blocks described above. As shown, each block is extremely compact, so it is possible in principle to scale up the network to very large sizes (e.g., a chip fabricated using an inexpensive 0.35 μm technology, using a relatively small area of 55 mm^2 would implement a network of 512 neurons and 256k synapses each having 5 bits precision.

In Fig. 3 we show an example of a simple spiking neural network architecture that can be formed using the IFMEM chip, using its silicon neurons, integrator synapses and SRAM cells. The SRAM cells that act as virtual synapses keep the synaptic weights and can be updated according to any desired spike-based learning algorithm. A neuron in the proposed architecture, shown in Fig. 2a, can be stimulated by up to $32 \times 32 \times 4 = 4096$ virtual synapses thanks to the demultiplexing scheme available on the proposed neuromorphic architecture [Moradi and Indiveri 2014]. The post-synaptic neuron generates spikes when its integrated input currents, i.e. the EPSCs generated by its related synapses, exceeds a predetermined threshold. At this time, the neuron generates an action potential and its membrane potential is then reset to a controllable reset potential [Indiveri et al. 2011].

4. THE HARDWARE-SOFTWARE NEUROMORPHIC SYSTEM

4.1. Experimental Setup

The experimental setup, shown in Fig. 4, consists of three main components: a Linux PC, a generic AER interface, and the neuromorphic hardware. The PC is used to control and interact with the neuromorphic system. It generates input spike trains (AER input) and transfers them to the *IFMEM*

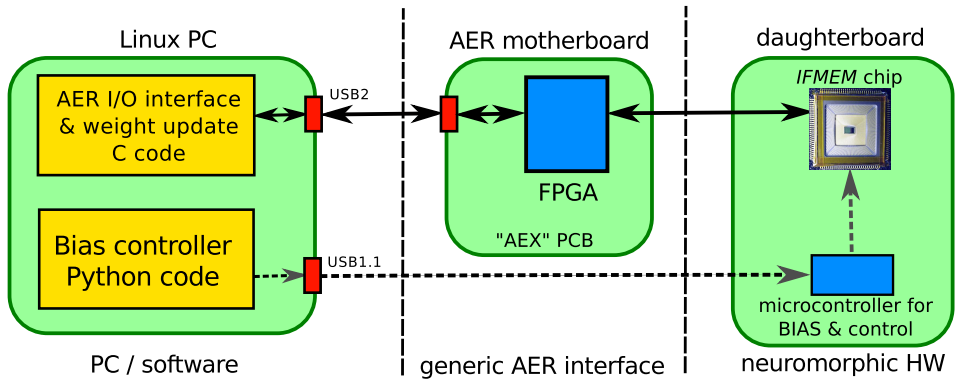


Fig. 4: Experimental setup of the hardware-software neuromorphic system. Dashed lines represent the control path for setting analog parameters and configuring the *IFMEM* chip, solid lines represent the path for the address-events data flow (from and to the *IFMEM* chip).

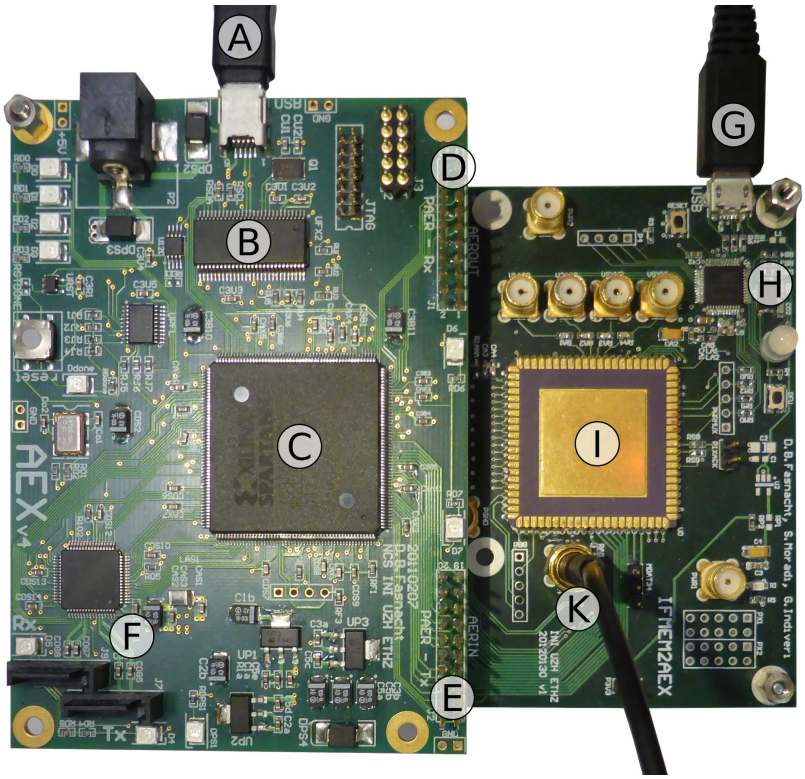


Fig. 5: The *AEX* printed circuit board with the attached daughterboard carrying the *IFMEM* chip: A: high-speed USB interface for AER communication, B: USB interface chip, C: FPGA for AER monitoring and sequencing, D: Parallel AER interface (chip to FPGA), E: Parallel AER interface (FPGA to chip), F: Serial AER section (unused), G: full-speed USB interface for *IFMEM* bias control, H: microcontroller for bias control, I: the *IFMEM* chip, K: analog voltage output connection.

chip via an AER interface. The PC also monitors, records and analyzes the AER output of the chip. Via a separate channel, the PC also sends bias values to the *IFMEM* chip, which control its various circuit parameters.

Figure 5 shows two printed circuit boards that host the three main hardware components of the system (shown in Fig. 4) including an FPGA, a microcontroller and the *IFMEM* neuromorphic chip. The PCB shown on the left is the AER motherboard, so called *AEX* board, which contains the FPGA. Directly attached to the *AEX* board, is a daughterboard containing the *IFMEM* chip and the microcontroller. The *AEX* board is a generic AER communication platform derived from the board first presented in [Fasnacht et al. 2008]. It consists of a high-speed (480 MHz) USB2.0 interface and an FPGA device. The USB interface enables the FPGA to communicate bi-directionally with the PC attached. The FPGA receives spike trains from the PC via USB and then generates them accordingly on its Parallel AER output interface to stimulate the *IFMEM* chip. Vice versa, the FPGA monitors the AER output of the *IFMEM* chip: each address-event received by the FPGA is sent to the PC, together with a 128 ns resolution timestamp of when exactly the spike was received at the Parallel AER input of the FPGA. The *AEX* board also contains a high-speed Serial AER interface to communicate with other *AEX* boards. Since only one such board is required in the single-chip experimental setup described, the Serial AER interface was not used.

Directly attached to the AEX communication board is a daughter-board. Figure 5 shows the two boards together. The daughterboard contains both the *IFMEM* chip and the circuitry needed to support the chip, such as voltage regulators and connectors to measure analog output voltages generated by the chip. It also contains a simple microcontroller that includes a full-speed (12 MHz) USB interface. Via this second USB interface the PC sends the bias values to the microcontroller. The microcontroller then programs the on-chip bias generator circuits to set the circuit bias voltages to the values specified by the user.

4.2. Programming the neuromorphic system

As shown in Fig. 4, the synaptic plasticity weight updates, the AER I/O interfacing, as well as the bias controlling are all programmed in software, on the Linux PC in the neuromorphic setup. In order to form a spiking neural network with a specific synaptic plasticity rule on this setup, several steps need to be taken. The first step is to calibrate the silicon neurons, synapse integrators, and the programmable DACs in the IFMEM chip using an automated routine (developed in Python). This calibration routine on the host PC can access the IFMEM chip via a microcontroller hosted on the IFMEM chip daughterboard, and interface to the PC via a USB port. The parameters changed on the IFMEM chip via the calibration routine, control the behaviors of the neural components including the response properties of silicon neurons and the dynamics of integrator synapses, on the neuromorphic device. The calibration process is useful to determine the parameters for setting desired properties of the neural network, such as membrane time constants, learning rates, etc.

In a second step, after calibration, the PC uses the AEX board to transfer the input spike trains to the chip and at the same time records AER addresses that correspond to the post-synaptic spikes being generated by the neurons on the chip. Each AER input (pre-synaptic event) contains four slots of information including 18 bits as shown in Fig. 2a. These bits describe different specifications including:

— the address of the post-synaptic neuron (5 bits),
— the address of the SRAM block containing the required synaptic weight (5 bits),
— the type (either inhibitory or excitatory) and the address of the desired physical synapse (3 bits), and
— the desired digital value for the synaptic weight that will be written to the addressed SRAM block (5 bits).

Each post-synaptic event (AER output) however only shows the address of the post-synaptic neuron that generated an event (spike).

In the third step, the recorded input (pre-synaptic) and output (post-synaptic) spikes are time-stamped and represented as Address Event (AE) with a time resolution of 1 μs. This data is then fed to a software module, which implements a specific spike-based learning rule such as STDP. This module, implemented in C language in our setup, then calculates the required synaptic weight changes, depending on the spike timings of various pre- and post-synaptic neurons, and uses this information to update the 5-bit synaptic weights stored in the related SRAM cells of the IFMEM chip. As the chip produces post-synaptic spikes, in response to the inputs arriving to the chip, and as a function of the learned synaptic weights, the system continues to learn, in an on-line "feed-through" mode.

When learning is not required anymore, the system can be set to "storage mode". In this mode the synaptic weights cannot be updated anymore and the chip can be used as an stand-alone device (i.e. disconnected from the PC), to receive input spikes from external sensory devices (e.g., from an event-based sensory VLSI device [Liu and Delbruck 2010]), process them according to its synaptic weights, and generate output spikes in result.

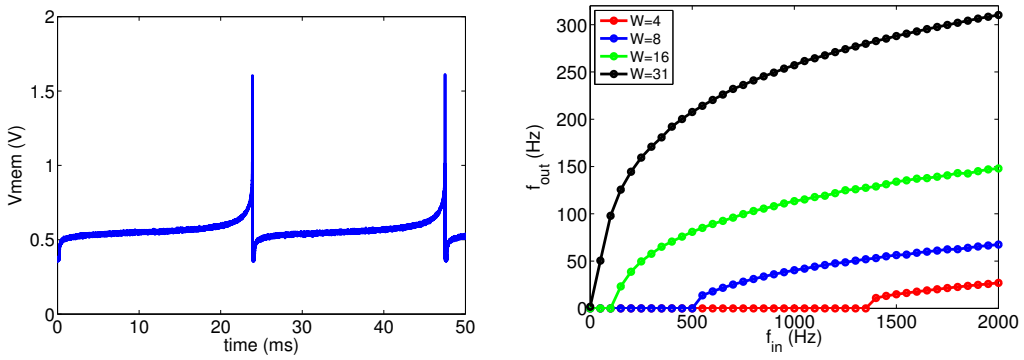
5. EXPERIMENTAL RESULTS

In this section we demonstrate how the device can be used in a hybrid SW-HW system to implement different forms of spike-based plasticity. In particular, we demonstrate that the *IFMEM* chip can work with both standard Spike-Timing Dependent Plasticity (STDP) learning prescriptions, as well as with more elaborate ones, that are being proposed in the computational neuroscience literature [Gjorgjieva et al. 2011; Clopath et al. 2010; Graupner and Brunel 2012]. In addition, we demonstrate how the specific variants of STDP that we use can reproduce the properties of the Bienenstock Cooper Munro (BCM) rate-based learning rule [Bienenstock et al. 1982]. Before implementing any synaptic plasticity rules on the IFMEM chip, first we characterize the response properties of available silicon neurons and programmable synapses.

5.1. Silicon neuron and programmable synapse response properties

The silicon neurons and programmable synapses available on the *IFMEM* chip, should be first calibrated to respond correctly to the input stimuli. In order to calibrate the chip parameters, we first injected constant current in the neuron circuit and adjusted the silicon neuron parameters to obtain biophysically realistic response properties, with biologically realistic time-constants (see Fig. 6a). Furthermore, to calibrate the synapse parameters, we first adjusted the parameters of the on-chip DACs that convert the SRAM bits into analog subthreshold currents, then we stimulated the synapse with regular spike trains at different frequencies, and measured the neuron response. Since the synapse is configured to behave as a linear filter, we stimulated a single synapse with input frequencies as high as 2000 Hz to represent inputs from many neurons at lower frequencies (by means of the superposition principle). Figure 6b shows the response of a silicon neuron to these input spike trains for different synaptic weight values. As shown, we calibrated the on-chip DACs to set synaptic weights that have a low gain, even for the highest weight value (w=31). As stated earlier, the DAC parameters for the synapse, as well as the silicon neuron parameters, were set using a software module implemented on the PC, to control the on chip bias generator circuit [Delbruck et al. 2010], via a microcontroller, integrated on the host Printed Circuit Board (PCB) (see Fig. 5).

Note that the high input spiking rates are not biologically plausible, for single synapses (which typically receive spikes with rates in the order of tens of Hertz or less). These inputs however are meant to represent the spikes arriving from multiple sources (e.g., a $2K$ Hz input spike train could represent spikes arriving from 2000 different neurons, each spiking at 1 Hz). This is possible thanks to the linear properties of the synapse integrator circuit, which can therefore exploit the superposition principle to collect low-rate spikes arriving from multiple "virtual" synapses into a single high-rate integration node. In addition to demonstrating the ability of the circuits to process high input firing rates, we have shown in a previous study that these silicon neurons and synapses can operate correctly also with biologically realistic times and firing rates [Azghadi et al. 2013].



(a) (b)

Fig. 6: Neuron and synapse characterization. (a) Silicon neuron membrane potential in response to constant current injection. (b) Output neuron frequency versus frequency of incoming spikes, representing either a very high firing rate of a single source, or multiple sources at lower firing rates.

5.2. Competitive Hebbian learning through STDP

The STDP learning rule has been widely used in many computational studies [Song et al. 2000] as well as multiple VLSI implementations [Bofill-I-Petit and Murray 2004; Indiveri 2002; Bamford et al. 2012; Azghadi et al. 2012; 2013; Azghadi et al. 2013b]. The basic STDP rule is expressed as:

$$\Delta w = \begin{cases} \Delta w^+ = A^+ e^{(\frac{-\Delta t}{\tau_+})} & \text{if } \Delta t > 0 \\ \Delta w^- = -A^- e^{(\frac{\Delta t}{\tau_-})} & \text{if } \Delta t \leq 0, \end{cases} \tag{2}$$

where $\Delta t = t_{\text{post}} - t_{\text{pre}}$ is the timing difference between a single pair of pre- and post-synaptic spikes. According to this model, the synaptic weight will be potentiated if a pre-synaptic spike arrives in a specified time window (τ_+) before the occurrence of a post-synaptic spike. Analogously, depression will occur if a pre-synaptic spike occurs within a time window (τ_-) after the post-synaptic spike. These time windows are not usually longer than about 50 ms. According to Eq. 2 the magnitude of potentiation/depression will be determined as a function of the timing difference between pre- and post-synaptic spikes, their temporal order, potentiation and depression time constants, and their relevant amplitude parameters (A^+ and A^-).

It has been shown that changes in synapses trained by the basic STDP learning mechanism lead eventually to a bi-modal distribution, in which synapses are either strongly potentiated, or strongly depressed. This behavior has been described as "Competitive Hebbian Learning" [Song et al. 2000]. In particular, the bi-modal distribution arises when (i) the weight of individual synapses is bounded and (ii) on average synapses tend to be depressed more than they tend to be potentiated (e.g. $A^+\tau_+ < A^-\tau_-$ in Eq. 2, when the area under the STDP curve for LTP and LTD are compared).

Here, we reproduce the competitive Hebbian learning behavior in our system by configuring a network composed of a single I&F neuron connected to its 32 5-bit digital synapses. We demonstrate how, when governed by STDP, the synaptic weights diverge into two distinguished groups over time.

The experiment is summarized as follows: We initialize the system by injecting a small constant current to one silicon neuron, to increase its excitability, and by setting all the weights of its 32 input synapses to their mid value of w=16. Next, we apply 32 independent Poisson spike trains with firing rates of 50 Hz to all these 32 synapses. The weighted EPSCs produced by the synapses are integrated by the silicon neuron, which eventually produces spikes, and transmits them as address-events to the workstation, using the AER protocol. The timing of the post-synaptic spikes (measured from the HW) and of the pre-synaptic spikes (synthesized in SW), applied to each of the 32 synapses is

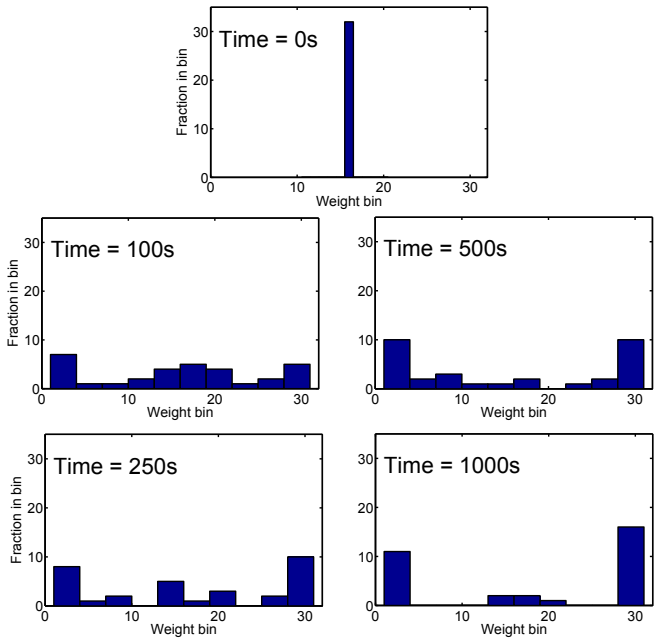


Fig. 7: Synaptic weights evolve to reach an equilibrium state, when modified by STDP learning rule. Here, 32 synaptic weights (weight bins) each one with 32 digital states, are altered by STDP over time. The top figure shows that all 32 synaptic weights are set to 16 in the beginning at time=0 s, i.e. the fraction of weights in weight bin 16 is 32. The other figures show the evolution of weights over time to reach a steady state at time=1000 s. The synaptic weights stay almost fixed thereafter, and the post-synaptic neuron firing rate held in an almost direct relation to the mean firing rate of pre-synaptic spike trains, i.e. 50 Hz.

then used to update the values of the 5-bit weights stored in the synapse corresponding SRAM cells, following a SW algorithm that implements the rule of Eq. 2. The SW algorithm first measures the Δts among the pre-synaptic from various synapses and the post-synaptic spikes, and then according to the STDP learning rule shown in Eq. 2 and its parameters, computes and returns the magnitude of the weight changes for all 32 synapses connected to the neuron. The STDP parameters used for this experiments are: A^+=0.5, A^-=0.527, and τ_+=τ_+=20 ms. Please also note that, during all synaptic plasticity experiments presented in this paper, the nearest neighbour spike interaction model (as opposed to the all-to-all spike interaction) is utilised.

Figure 7 shows how the synaptic weights evolve over time to reach a stable state in which they show the expected bi-modal distribution. Note how we were able to reproduce the competitive Hebbian learning behavior even with weights bounded to 5-bit resolution, and with hybrid SW-HW spike-timing interactions.

5.3. Implementing BCM through STDP

Although BCM is an inherently rate-based rule and depends on the activities of pre- and post-synaptic neurons, recent studies have shown that timing-based triplet STDP learning rule can reproduce BCM-like functionality [Gjorgjieva et al. 2011]. Here we demonstrate how this rate-based functionality can be realized by our SW-HW system, by using the triplet STDP learning rule [Gjorgjieva et al. 2011] to update the 5-bit synaptic weight values of the *IFMEM* chip.

The triplet-based STDP, can be formulated as

$$\Delta w = \begin{cases} \Delta w^+ = e^{(\frac{-\Delta t_1}{\tau_+})}\Big(A_2^+ + A_3^+ e^{(\frac{-\Delta t_2}{\tau_y})}\Big) \\ \Delta w^- = -e^{(\frac{\Delta t_1}{\tau_-})}\Big(A_2^- + A_3^- e^{(\frac{-\Delta t_3}{\tau_x})}\Big), \end{cases} \tag{3}$$

where $\Delta w = \Delta w^+$ for $t = t_{\rm post}$ and if $t = t_{\rm pre}$ then the weight change is $\Delta w = \Delta w^-$. A_2^+, A_2^-, A_3^+ and A_3^- are potentiation and depression amplitude parameters, $\Delta t_1 = t_{\rm post(n)} - t_{\rm pre(n)}$, $\Delta t_2 = t_{\rm post(n)} - t_{\rm post(n-1)} - \epsilon$ and $\Delta t_3 = t_{\rm pre(n)} - t_{\rm pre(n-1)} - \epsilon$, are the time differences between combinations of pre- and post-synaptic spikes. Here, ϵ is a small positive constant which ensures that the weight update uses the correct values occurring just before the pre or post-synaptic spike of interest, and finally τ_-, τ_+, τ_x and τ_y represent time constants [Pfister and Gerstner 2006].

It has been shown [Pfister and Gerstner 2006] that for Poisson distributed spike trains Eq. 3 can be approximated as:

$$\langle dw/dt \rangle = -A_2^- \tau_- \rho_{pre} \rho_{post} - A_3^- \tau_- \tau_x \rho_{pre}^2 \rho_{post} + A_2^+ \tau_+ \rho_{pre} \rho_{post} + A_3^+ \tau_+ \tau_y \rho_{post}^2 \rho_{post} \tag{4}$$

where ρ_{pre} and ρ_{post} represent the mean firing rates of the pre- and post-synaptic spike trains, respectively.

Generally, the BCM theory suggests that the synaptic weight changes are in a linear relationship with the pre-synaptic, and a non-linear relationship with the post-synaptic mean firing rates [Bienenstock et al. 1982]. Therefore, a general description of the BCM rule can be written as:

$$dw/dt = \rho_{pre} \cdot \phi(\rho_{post}, \theta) \tag{5}$$

where ϕ is a function that satisfies the conditions $\phi(\rho_{post} > \theta, \theta) > 0$, $\phi(\rho_{post} < \theta, \theta) < 0$ and $\phi(0, \theta) = 0$. Essentially, if the post-synaptic firing rate ρ_{post} is below the threshold θ, then dw/dt is negative and the synaptic weight is depressed. Conversely, the synaptic weight is potentiated if the post-synaptic firing rate is larger than the threshold θ, and it is left unchanged if $\phi = 0$, i.e., if $\rho_{post} = \theta$ [Pfister and Gerstner 2006].

The equations 4 and 5 can be mapped together, if two conditions are satisfied. The first condition requires having a linear relationship between the pre-synaptic firing activity (ρ_{pre}) and the synaptic weight change ($\langle dw/dt \rangle$), as shown in Eq. 5. This condition is satisfied if $A_3^- = 0$, in the triplet STDP equation (Eq. 4). This will lead to a minimal version of the Triplet Spike-Timing Dependent Plasticity (TSTDP) rule presented in [Pfister and Gerstner 2006], which has been shown to account for various synaptic plasticity neuroscience experiments, including those dealing with higher order spike trains [Wang et al. 2005]. The second condition requires that the sliding threshold θ, that determines the frequency, in which depression turns to potentiation, is proportional to the expectation of the p^{th} power of the post-synaptic firing rate (ρ_{post}) [Pfister and Gerstner 2006; Bienenstock et al. 1982]. This second condition can be satisfied if the threshold of the BCM rule is defined as

$$\theta = \langle \rho_{post}^p \rangle (A_2^- \tau_- + A_2^+ \tau_+) / \rho_0^p A_3^+ \tau_+ \tau_y \tag{6}$$

Given this equation, the sliding threshold effect of the BCM rule is proportional to the post-synaptic firing rate, with the proportionality factor set by the STDP rule parameters.

To implement BCM via the triplet STDP rule in the *IFMEM* chip setup, we used a single synapse, connected to a post-synaptic silicon neuron and changed its efficacy using the STDP rule of Eq. (3). At the beginning of the experiment, the initial weight of the synapse is set to its maximum value of 31. This high synaptic weight makes the post-synaptic neuron fire at a high rate, proportional to the pre-synaptic firing rate [Azghadi et al. 2013]. The SW pre-synaptic spike train, and the spike train produced by the silicon neuron, are then used to calculate the amount of weight changes in the corresponding synaptic efficacy, according to a minimal model of triplet STDP [Gjorgjieva et al. 2011].

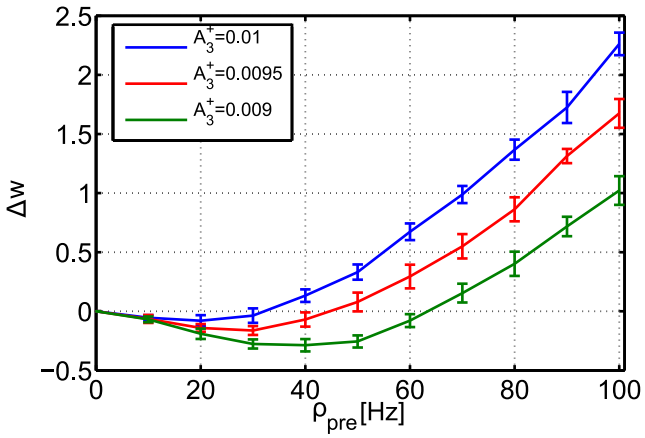


Fig. 8: The sliding threshold feature of the rate-based BCM rule is replicated through Triplet STDP rule, implemented on the *IFMEM* chip.

Figure 8 shows the total amount of weight changes in response to Poisson spike trains of 20 s length, for a range of pre-synaptic spike rates from 0 Hz up to 100 Hz. In this figure, the sliding threshold feature of the BCM learning rule is regenerated through changing the amount of one of the parameters of the TSTDP learning rule, i.e. A_3^+. According to Eq. 6, with increase in A_3^+ parameter, the threshold decreases and slides toward lower post- synaptic firing rates. Please note that, in the presented experiment, the silicon neuron parameters, as well as the synaptic weight parameters in its corresponding physical synapse, i.e. the differential pair integrator, are calibrated in a way that pre- and post-synaptic neuron are in a relatively linear relationship [Moradi and Indiveri 2014; Azghadi et al. 2013]. In this figure, each data point corresponds to the mean of the weight changes over 10 trials, and the error bar represents the standard deviation of the weight change over these trials. This amount of weight changes can then be discretized and written back into the SRAM. The STDP parameters that have been used in this experiment are as follows: A_3^-=A_2^+=0, A_2^-=0.0068, τ_+=16.8 ms, τ_-=33.7 ms, and τ_y=114 ms.

5.4. Classification of complex correlated patterns

Here we use the TSTDP learning rule, with its parameters tuned for exhibiting BCM behavior (see Fig. 8), to demonstrate how the proposed VLSI device can perform classification of binary patterns with high levels of correlations.

The neural classifier implemented is composed of one neuron and 30 synapses, which are arranged in a single layer perceptron-like architecture. The goal is to train the perceptron synaptic weights, via the TSTDP algorithm, to learn to distinguish two input patterns, UP and DOWN, in an unsupervised fashion. After training, the HW perceptron should be able to respond with a high firing rate to pattern UP, and a low one to pattern DOWN. This is a similar experimental scenario, to the semi-supervised learning scenario utilized in a similar classification task using spiking neural networks [Giulioni et al. 2009].

The two UP and DOWN patterns can have various degrees of correlations. The correlation determines the amount of overlap in the input synapses used, and the similarity in the output response of the neuron: when there is no correlation, one pattern is applied to 15 random synapses and the other pattern is applied to the remaining 15 synapses (no overlap).

The pattern UP stimulates 15 synapses with Poisson spike trains that have a high mean firing rate of 300 Hz, while pattern DOWN comprises 15 Poisson spike trains with a low mean firing rate of 20 Hz. Therefore, in the case of zero correlation, the two patterns are likely to produce different

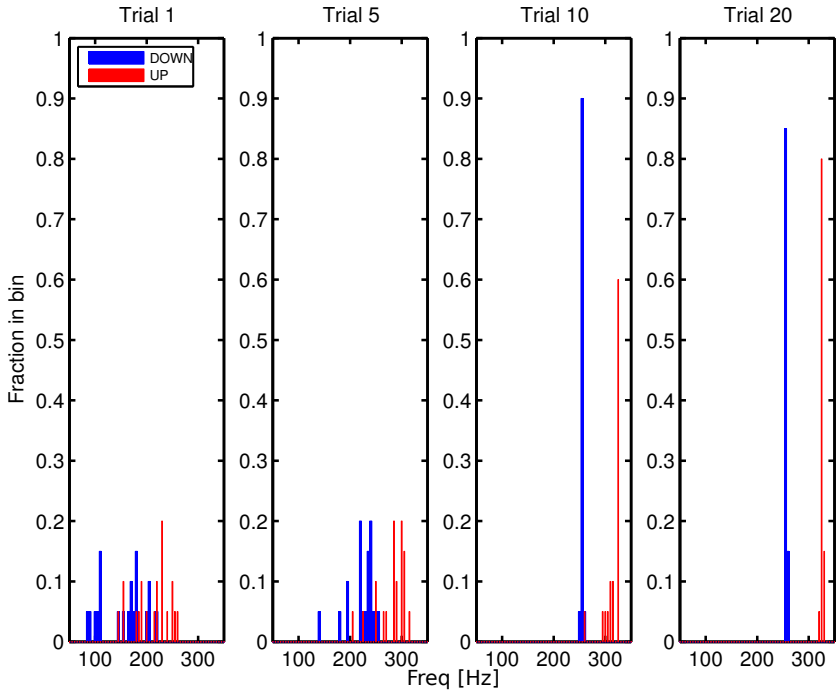


Fig. 9: Distribution of the neuron output frequencies during different stages of learning. In the beginning of the learning phase, when initial weights are random, the neuron cannot distinguish between the two patterns. During the learning trials, the synapses are being modified and the neuron begins to effectively discriminate between the two patterns from trial 20. In this experiment the correlation is equal to 20 %, i.e., there are 6 inputs that are common to the two patterns that always receive high firing rates.

outputs (depending on the values of the synaptic weights) even before learning. However, for the case of non-zero correlations, a random subset of N input synapses are always stimulated by high mean firing rate spike trains of 300 Hz, while the rest of the synapses are assigned to the two UP and DOWN patterns. For instance, if the number of correlated synapses is 10, 10 randomly synapses are stimulated by Poisson spike trains of 300 Hz, and the remaining 20 synapses will be reserved for the UP and DOWN patterns. In this case, because of the N common high input synapses, the two patterns will have closer mean firing rates, and therefore their classification becomes more challenging. Therefore, in the beginning of learning phase, the output frequency range of the perceptron cannot be distinguished between the two patterns and as a result learning is required to classify the two patterns.

The training phase is composed of several trials. In each trial, one of the two patterns, UP or DOWN is randomly applied to the 30 input synapses, with a set degree of correlation, and with a new distribution of Poisson spikes. The two patterns have equal probability to be selected. For each trial the synaptic weights are modified according to the TSTDP. In our experiment the synaptic weights reach a steady state and do not change significantly after about 20 trials, in which the input spike trains lasted 10 s each.

Figure 9 shows how the distribution of the neuron output firing rates changes with learning, after 1, 5, 10, and 20 trials. The output neuron rates were collected over 20 classification runs, with each run comprising 20 learning trials and 20% correlation. In each run the synaptic weights are

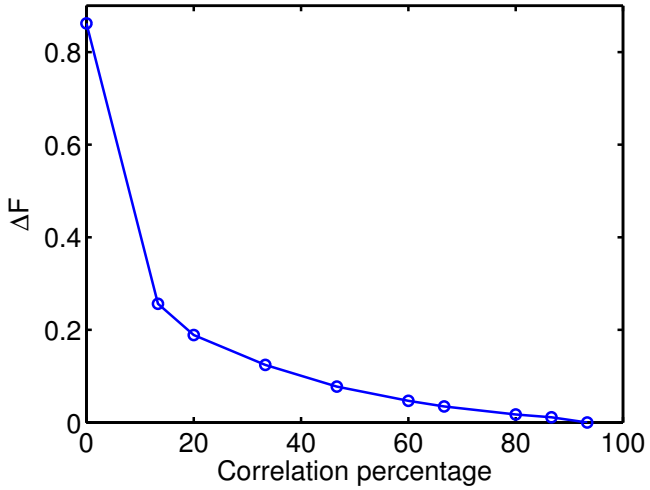


Fig. 10: The performance of the classifier implemented on the *IFMEM* chip. Here $\Delta F = (F_{UP}^{min} - F_{DOWN}^{max})/F_{UP}^{min}$, where F_{UP}^{min} and F_{DOWN}^{max} are the minimum and the maximum frequencies for pattern UP and DOWN, respectively, for all 20 runs at the end of learning in trial 20.

initialized to random 5-bit values, the definition of UP and DOWN patterns is changed, and a new random order of UP and DOWN patterns applied across trials is defined.

As expected, the TSTDP learning rule used tends to decrease the weights of the synapses targeted by the DOWN pattern, while it tends to increase the weights of both the UP and correlated (overlapping) synapses. After learning, the neuron will therefore fire with high firing rates when stimulated with UP patterns, and low firing rates when stimulated by DOWN patterns. While after a few trials (e.g. see second and third panels of Fig. 9) the neuron already performs above chance levels, many trials (20 in our experiments) are required to unambiguously classify the two patterns.

This classification performance is robust and holds for also large amounts of correlations (up to 90 %) in the input patterns. In terms of classification accuracy, we consider a DOWN pattern correctly classified if the neuron output frequency is less than a set threshold in response to that pattern; and similarly, an UP pattern is correctly classified, if the neuron response to such pattern has a firing rate higher than the threshold. In our experiments the classifier has 100 % correct performance, even with correlation levels of 87 % (i.e., 26 overlapping synapses), if the classification threshold is adaptive (e.g., if it is set just below to the minimum frequency in response to the UP patterns). What changes however is the difference in the responses to the two patterns. Figure 10 shows how this difference decreases as the correlation among the input patterns increases.

Although the developed classification device demonstrates promising results in the targeted classification scenario, its ability in more challenging pattern classification tasks yet to be evaluated in future research.

6. CONCLUSIONS

We presented a hybrid SW-HW neuromorphic system that utilizes a previously developed programmable neuromorphic VLSI device (*IFMEM* chip) that comprises silicon neurons and event-driven synapses, with programmable synaptic weight circuits. We demonstrated how this device can be used in the developed system, to implement different types of spike-timing dependent plasticity learning rules, and demonstrated how these rules can reproduce interesting competitive Hebbian learning and rate-based behaviors, even with the limitations of the hardware implementation (5-bit resolution for the weights, mismatch of the analog subthreshold circuits, etc.). Finally we described

how the hybrid SW-HW learning setup proposed can be used to train a perceptron to perform binary classification in an unsupervised way, and to be robust to extremely high correlations in the input patterns.

The device and setup proposed therefore represents a useful real-time low-power computing platform for exploring the effectiveness of different types of spike-based learning algorithms, validating their performance at run-time on real-time custom analog/digital hardware, and implementing robust perceptron-like neural networks to carry out real-time classifications tasks. If the task can be solved after training the weights of the neural network, without requiring continuous or on-line training, then the platform proposed represents a stand-alone compact and low-power alternative to standard full-digital computing solutions (no PC is required in the loop). The use of the AER representation for receiving inputs, computing with spikes, and transmitting signals in output, make this device an ideal computational platform for building embedded neuromorphic event-based computational systems that process events generated by neuromorphic sensory systems [Liu and Delbruck 2010].

REFERENCES

L.F. Abbott and W. Gerstner. 2004. Homeostasis and learning through spike-timing dependent plasticity. In *Methods and Models in Neurophysics. Editors: D. Hansel, C. Chow, B. Gutkin, and C. Meunier*.

J.V. Arthur, P. Merolla, F. Akopyan, R. Alvarez, A. Cassidy, A. Chandra, S. Esser, N. Imam, W. Risk, D.B.D. Rubin, R. Manohar, and D.S. Modha. 2012. Building Block of a Programmable Neuromorphic Substrate: A Digital Neurosynaptic Core. In *International Joint Conference on Neural Networks, IJCNN 2012*. DOI: http://dx.doi.org/10.1109/IJCNN.2012.6252637

M. Rahimi Azghadi, S. Al-Sarawi, D. Abbott, and N. Iannella. 2013a. A neuromorphic VLSI design for spike timing and rate based synaptic plasticity. *Neural Networks* 45 (2013), 70–82.

M. Rahimi Azghadi, S. Al-Sarawi, D. Abbott, and N. Iannella. 2013b. Pairing Frequency Experiments in Visual Cortex Reproduced in a Neuromorphic STDP Circuit. In *2013 IEEE International Conference on Electronics, Circuits, and Systems*. 229–232.

M. Rahimi Azghadi, S. Al-Sarawi, N. Iannella, and D. Abbott. 2012. Efficient Design of Triplet Based Spike-Timing Dependent Plasticity. In *International Joint Conference on Neural Networks, IJCNN 2012*. DOI: http://dx.doi.org/10.1109/IJCNN.2012.6252820

M. Rahimi Azghadi, S. Al-Sarawi, N. Iannella, and D. Abbott. 2013. A new compact analog VLSI model for Spike Timing Dependent Plasticity. In *2013 IFIP/IEEE 21st International Conference on Very Large Scale Integration (VLSI-SoC)*. 7–12.

M. Rahimi Azghadi, S. Al-Sarawi, N. Iannella, and D. Abbott. 2014a. Tunable Low Energy, Compact and High Performance Neuromorphic Circuit for Spike-Based Synaptic Plasticity. *PLoS ONE* 9, 2 (2014), art. no. e88326. DOI: http://dx.doi.org/10.1371/journal.pone.0088326

M. Rahimi Azghadi, N. Iannella, S. F. Al-Sarawi, G. Indiveri, and D. Abbott. 2014b. Spike-Based Synaptic Plasticity in Silicon: Design, Implementation, Application, and Challenges. *Proc. IEEE* 102, 5 (2014), 717–737.

M. Rahimi Azghadi, S. Moradi, and G. Indiveri. 2013. Programmable neuromorphic circuits for spike-based neural dynamics. In *2013 IEEE 11th International New Circuits and Systems Conference (NEWCAS)*. DOI: http://dx.doi.org/10.1109/NEWCAS.2013.6573600

S. Bamford, A.F. Murray, and D.J. Willshaw. 2012. Spike-timing-dependent plasticity with weight dependence evoked from physical constraints. *IEEE Transactions on Biomedical Circuits and Systems* 6, 4 (2012), 385–398.

C. Bartolozzi and G. Indiveri. 2007. Synaptic dynamics in analog VLSI. *Neural Computation* 19, 10 (2007), 2581–2603.

A. Belatreche, L.P. Maguire, and M. McGinnity. 2006. Advances in Design and Application of Spiking Neural Networks. *Soft Computing* 11, 3 (2006), 239–248.

E.L. Bienenstock, L.N. Cooper, and P.W. Munro. 1982. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience* 2, 1 (1982), 32–48.

A. Bofill-I-Petit and A.F. Murray. 2004. Synchrony detection and amplification by silicon neurons with STDP synapses. *IEEE Transaction on Neural Networks* 15, 5 (2004), 1296–1304.

J. Brader, W. Senn, and S. Fusi. 2007. Learning real world stimuli in a neural network with spike-driven synaptic dynamics. *Neural Computation* 19 (2007), 2881–2912.

R. Brette and W. Gerstner. 2005. Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity. *Journal of Neurophysiology* 94 (2005), 3637–3642.

E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri. 2014. Neuromorphic electronic circuits for building autonomous cognitive systems. *Proc. IEEE* (2014). DOI: http://dx.doi.org/DOI:10.1109/JPROC.2014.2313954

E. Chicca, A.M. Whatley, P. Lichtsteiner, V. Dante, T. Delbrück, P. Del Giudice, R.J. Douglas, and G. Indiveri. 2007. A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity. *IEEE Transactions on Circuits and Systems I* 5, 54 (2007), 981–993.

T.Y.W. Choi, B.E. Shi, and K. Boahen. 2004. An ON-OFF Orientation Selective Address Event Representation Image Transceiver Chip. *IEEE Transactions on Circuits and Systems I* 51, 2 (2004), 342–353.

C. Clopath, L. Büsing, E. Vasilaki, and W. Gerstner. 2010. Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nature Neuroscience* 13, 3 (2010), 344–352.

P. Dayan and L.F. Abbott. 2001. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Taylor & Francis.

T. Delbruck, R. Berner, P. Lichtsteiner, and C. Dualibe. 2010. 32-bit Configurable bias current generator with sub-off-current capability. In *International Symposium on Circuits and Systems, ISCAS 2010*. 1647–1650.

D.B. Fasnacht, A.M. Whatley, and G. Indiveri. 2008. A Serial Communication Infrastructure for Multi-Chip Address Event System. In *International Symposium on Circuits and Systems, ISCAS 2008*. 648–651.

A. Fidjeland, D. Gamez, M. P. Shanahan, and E. Lazdins. 2013. Three Tools for the Real-Time Simulation of Embodied Spiking Neural Networks Using GPUs. *Neuroinformatics* 11, 3 (2013), 267–290.

A. Fidjeland, E.B. Roesch, M.P. Shanahan, and W. Luk. 2009. NeMo: A platform for Neural Modelling of spiking neurons using GPUs. In *IEEE Application-specific Systems, Architectures and Processors Conference ASAP*. 137–144.

Steve B. Furber, David R. Lester, Luis A. Plana, Jim D. Garside, Eustace Painkras, Steve Temple, and Andrew D. Brown. 2013. Overview of the SpiNNaker System Architecture. *IEEE Trans. Comput.* 62, 12 (2013), 2454–2467.

W. Gerstner and W.M. Kistler. 2002. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press.

M. Giulioni, M. Pannunzi, D. Badoni, V. Dante, and P. Del Giudice. 2009. Classification of correlated patterns with a configurable analog VLSI neural network of spiking neurons and self-regulating plastic synapses. *Neural Computation* 21, 11 (2009), 3106–3129. DOI:http://dx.doi.org/10.1162/neco.2009.08-07-599

J. Gjorgjieva, C. Clopath, J. Audet, and J.-P. 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.

M. Graupner and N. Brunel. 2012. Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location. *Proceedings of the National Academy of Sciences* 109 (2012), 3991–3996.

G. Indiveri. 2002. Neuromorphic Bistable VLSI Synapses with Spike-Timing-Dependent Plasticity. In *Advances in Neural Information Processing Systems*, Vol. 15. 1091–1098.

G. Indiveri, B. Linares-Barranco, T.J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen. 2011. Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience* 5 (2011), 1–23.

G. Indiveri, F. Stefanini, and E. Chicca. 2010. Spike-based learning with a generalized integrate and fire silicon neuron. In *International Symposium on Circuits and Systems, ISCAS 2010*. 1951–1954.

R. Kempter, W. Gerstner, and J.L. van Hemmen. 1999. Hebbian learning and spiking neurons. *Physical Review E* 59, 4 (1999), 4498–4514.

D. Khosla, D.J. Huber, and C. Kanan. 2014. A neuromorphic system for visual object recognition. *Biologically Inspired Cognitive Architectures* 8 (2014), 33–45.

C. Koch. 1999. *Biophysics of Computation: Information Processing in Single Neurons*. Oxford University Press.

S.B. Laughlin and T.J. Sejnowski. 2003. Communication in Neuronal Networks. *Science* 301, 5641 (2003), 1870–1874.

S.-C. Liu and T. Delbruck. 2010. Neuromorphic sensory systems. *Current Opinion in Neurobiology* 20, 3 (2010), 288–295.

C.G. Mayr and J. Partzsch. 2010. Rate and pulse based plasticity governed by local synaptic state variables. *Frontiers in Synaptic Neuroscience* 2, 33 (2010).

C. Mead. 1990. Neuromorphic Electronic Systems. *Proc. IEEE* 78, 10 (1990), 1629–36.

S. Mitra, S. Fusi, and G. Indiveri. 2009. Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI. *IEEE Transactions on Biomedical Circuits and Systems* 3, 1 (2009), 32–42.

S. Moradi and G. Indiveri. 2011. A VLSI network of spiking neurons with an asynchronous static random access memory. In *Biomedical Circuits and Systems Conference, BioCAS 2011*. 277–280.

S. Moradi and G. Indiveri. 2014. An Event-Based Neural Network Architecture With an Asynchronous Programmable Synaptic Memory. *IEEE Transactions on Biomedical Circuits and Systems* 8, 1 (2014), 98–107.

J.M. Nageswaran, N. Dutt, J.L. Krichmar, A. Nicolau, and A.V. Veidenbaum. 2009. A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors. *Neural Networks* 22, 5-6 (2009), 791–800.

A. Nere, U. Olcese, D. Balduzzi, and G. Tononi. 2012. A neuromorphic architecture for object recognition and motion anticipation using burst-STDP. *PloS ONE* 7, 5 (2012), art. no. e36958.

J.P. Pfister and W. Gerstner. 2006. Triplets of spikes in a model of spike timing-dependent plasticity. *The Journal of Neuroscience* 26, 38 (2006), 9673–9682.

P. Rowcliffe and J. Feng. 2008. Training Spiking Neuronal Networks With Applications in Engineering Tasks. *IEEE Transactions on Neural Networks* 19, 9 (2008), 1626–1640.

J. Schemmel, D. Bruderle, A. Grubl, M. Hock, K. Meier, and S. Millner. 2010. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *IEEE International Symposium on Circuits and Systems, ISCAS 2010*. 1947–1950.

M. Schmuker, T. Pfeil, and M.P. Nawrot. 2014. A neuromorphic network for generic multivariate data classification. *Proceedings of the National Academy of Sciences* 111, 6 (2014), 2081–2086.

P.J. Sjöström, E.A. Rancz, A. Roth, and M. Häusser. 2008. Dendritic excitability and synaptic plasticity. *Physiological Reviews* 88, 2 (2008), 769–840.

S. Song, K.D. Miller, and L.F. Abbot. 2000. Competitive Hebbian learning through spike-timing-dependent plasticity. *Nature Neuroscience* 3, 9 (2000), 919–926.

R.J. Vogelstein, U. Mallik, J.T. Vogelstein, and G. Cauwenberghs. 2007. Dynamically reconfigurable silicon array of spiking neurons with conductance-based synapses. *IEEE Transactions on Neural Networks* 18, 1 (2007), 253–265.

H.X. Wang, R.C. Gerkin, D.W. Nauen, and G.Q. Bi. 2005. Coactivation and timing-dependent integration of synaptic potentiation and depression. *Nature Neuroscience* 8, 2 (2005), 187–193.

Received December 2013; revised X; accepted X

Online Appendix to: Programmable Spike-Timing Dependent Plasticity learning circuits in neuromorphic VLSI architectures

Mostafa Rahimi Azghadi*, University of Zurich and ETH Zurich and the University of Adelaide
Saber Moradi*, University of Zurich and ETH Zurich
Daniel B. Fasnacht, University of Zurich and ETH Zurich
Mehmet Sirin Ozdas, University of Zurich and ETH Zurich
Giacomo Indiveri, University of Zurich and ETH Zurich

DOI : http://dx.doi.org/10.1145/0000000.0000000