An Event-Based Neural Network Architecture With an Asynchronous Programmable Synaptic Memory

Originally published in:
IEEE Transactions on Biomedical Circuits and Systems 8(1), https://doi.org/10.1109/TBCAS.2013.2255873

An Event-Based Neural Network Architecture With an Asynchronous Programmable Synaptic Memory

Saber Moradi, *Student Member, IEEE*, and Giacomo Indiveri, *Senior Member, IEEE*

***Abstract*—We present a hybrid analog/digital very large scale integration (VLSI) implementation of a spiking neural network with programmable synaptic weights. The synaptic weight values are stored in an asynchronous Static Random Access Memory (SRAM) module, which is interfaced to a fast current-mode event-driven DAC for producing synaptic currents with the appropriate amplitude values. These currents are further integrated by current-mode integrator synapses to produce biophysically realistic temporal dynamics. The synapse output currents are then integrated by compact and efficient integrate and fire silicon neuron circuits with spike-frequency adaptation and adjustable refractory period and spike-reset voltage settings. The fabricated chip comprises a total of 32 $\times$ 32 SRAM cells, 4 $\times$ 32 synapse circuits and 32 $\times$ 1 silicon neurons. It acts as a transceiver, receiving asynchronous events in input, performing neural computation with hybrid analog/digital circuits on the input spikes, and eventually producing digital asynchronous events in output. Input, output, and synaptic weight values are transmitted to/from the chip using a common communication protocol based on the Address Event Representation (AER). Using this representation it is possible to interface the device to a workstation or a micro-controller and explore the effect of different types of Spike-Timing Dependent Plasticity (STDP) learning algorithms for updating the synaptic weights values in the SRAM module. We present experimental results demonstrating the correct operation of all the circuits present on the chip.**

***Index Terms*—Address event representation (AER), analog/digital, asynchronous, circuit, event-based, learning, neural network, neuromorphic, programmable weights, real-time, sensory-motor, silicon neuron, silicon synapse, spike-timing dependent plasticity (STDP), spiking, static random access memory (SRAM), synaptic dynamics, very large scale integration (VLSI).**

I. Introduction

SPIKING neural networks represent a promising computational paradigm for solving complex pattern recognition and sensory processing tasks that are difficult to tackle using standard machine vision and machine learning techniques [1], [2]. Much research has been dedicated to software simulations of spiking neural networks [3], and a wide range of solutions have been proposed for solving real-world and engineering problems [4], [5]. But these solutions are often implemented as software algorithms running on bulky and power-hungry workstations. In order to apply this computational paradigm to compact efficient neural processing and sensory-motor systems, that compute on real-world sensory signals and interact with the environment in real-time, it is necessary to develop dedicated hardware implementations of spiking neural networks which are low-power and can operate with biologically plausible time-constants. In this specific scenario, this suggests the design of full custom analog/digital Very Large Scale Integration (VLSI) neuromorphic systems [6]. However, to meet the requirement of real-time interaction with the environment, some of the recently proposed VLSI design solutions that operate only on "accelerated time" scales (i.e., in which unit of real time is "simulated" in hardware two or three orders of magnitude faster), are not suitable [7], [8]. Similarly, neural VLSI solutions that focus on large-scale systems simulations are not ideal, as they compromise the low-power or compactness requirements [9]–[12]. In this paper we propose a compact full-custom VLSI device that comprises low-power sub-threshold analog circuits and asynchronous digital circuits to implement networks of spiking neurons with programmable synaptic weights [13], [14]. In our implementation neural computation is performed in the analog domain while the communication of spikes between neurons is carried out asynchronously in the digital domain. Specifically, the analog circuits implement neural and synaptic dynamics in a very compact and power efficient way, while digital asynchronous circuits implement a real-time event (spike) based communication protocol. We designed a new set of asynchronous circuits for interfacing the asynchronous events to conventional five-bit Static Random Access Memory (SRAM) cells, to manage the storage of the network's synaptic weight values. In this way, the programmable SRAM cells can update the network's synaptic weights using the same asynchronous communication protocol used to transmit spiking events across the network. The use of SRAM cells as digital memory storage for synaptic weights in neuromorphic chips has already been proposed in the past (e.g., see [14] for a recent study). Also the idea of programming different parameters in spiking neural networks, such as synaptic weights [13], [15], [16], or even dendritic tree and synaptic routing structures [17]–[20], is not new. However, as these solutions typically require long settling times, they are not ideal for integration in circuits that employ fast asynchronous digital event-based communication circuits. Here we propose a solution that uses both SRAM cells and fast Digital to Analog Converters (DACs) interfaced to asynchronous digital circuits, to either set the synaptic weights

Manuscript received November 14, 2012; revised February 12, 2013; accepted March 22, 2013. This work was supported by the European Community's Seventh Framework Programme: Grant 231467—"eMorph" and ERC Grant 257219—"neuroP." This paper was recommended by Associate Editor J. Van der Spiegel.

The authors are with the Institute of Neuroinformatics, University of Zurich and ETH Zurich, CH-8057 Zurich, Switzerland (e-mail: saber@ini.phys.ethz.ch; giacomo@ini.phys.ethz.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBCAS.2013.2255873

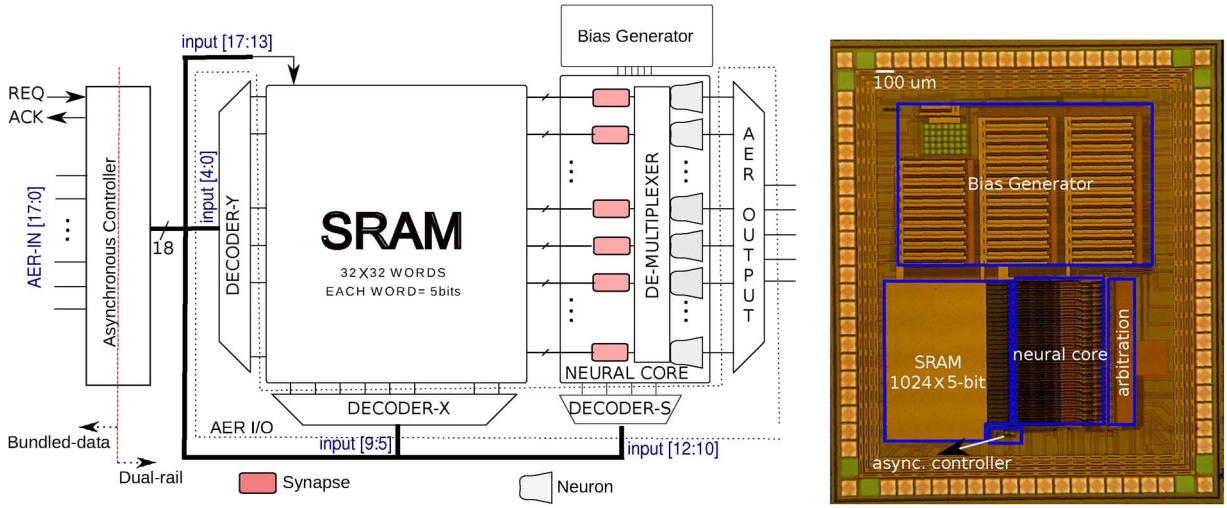

Fig. 1. (a) Chip block diagram. The device comprises a neural-core module with an array of synapses and integrate-and-fire neurons, an asynchronous SRAM module to store the synaptic weight values, a bias generator to set the parameters in the analog circuits, and asynchronous control and interfacing circuits to manages the AER communication. (b) Chip micro-graph. The chip was fabricated using a standard 0.35 μm CMOS technology and occupies an area of 2.1×2.5 mm^2 including pad-frame.

on-line with each incoming event, or to program the weights off-line and use the values stored in the SRAM in standard multi-chip event-based architectures. The asynchronous communication scheme implemented in this VLSI device is based on the Address Event Representation (AER) communication protocol, commonly used to build large-scale multi-chip neuromorphic systems [6], [21]–[23], while the SRAM and DAC circuits described, as well as the basic architecture design are based on a previous VLSI prototype device that we presented in [24].

The ability to set/change the synaptic weights on-line [25], [26] and to program/store them for off-line or batch-mode use allows users to explore different spike-based learning algorithms and methods, for example by implementing them in software on a PC or on an external micro-controller interfaced to the chip. By including the VLSI device in the training loop, the circuit non-idealities and variability can be potentially adapted away through the PC-based learning algorithms [27]. Once the network has been trained and the synaptic weight values stored in the SRAM, the VLSI device can be used in stand-alone mode to carry out neural computation in real-time, exploiting its low-power, and compact size properties. Typical use-case scenario for these types of devices in an actual system is that of implementing event-based spiking neural network architectures, that computing weighted sums of their inputs and produce the desired outputs in the form of asynchronous spikes. As such these devices are ideal for processing event-based sensory data, for example generated by silicon cochleas or silicon retinas [28], [29] and driving robotic actuators in real-time. In the next section we describe the overall chip architecture. Details about the circuits of each block are discussed in Section III. In Section IV we present experimental results from the fabricated chip, and show how the measurements match with the expected functionality. Finally we present the discussion and conclusion in Section V.

II. The Chip Architecture

The architecture of the chip is illustrated in Fig. 1. The chip was fabricated using AMS 0.35 μm Complementary Metal-Oxide Semiconductor (CMOS) technology. It comprises five main blocks: the *asynchronous controller*, the *SRAM* block, the *neural-core*, the *bias generator* and the *AER Input/Output (IO) interfaces*.

The AER I/O interfaces are standard encoder/decoder circuits commonly used in AER systems [30]. The bias generator block provides 38 temperature-compensated analog currents as global biases used in the neural-core [31]. In the current prototype chip, the bias generator occupies a significant fraction of the layout area. But, as opposed to the other blocks on this chip, the size of this block will not change when scaling up these devices to large network sizes. The asynchronous controller manages the communication between the external digital asynchronous signals and the on-chip ones. The asynchronous SRAM block is used to store synaptic weight values using standard memory circuits but with the inclusion of a filter circuit that generates a dual-rail representation of the data [32]. The neural core block comprises a column of 32×1 adaptive integrate-and-fire neurons [33] and an array of 4×32 synapses with DAC circuits to convert the digitally encoded weight into an analog current. In principle, given that the synapse integrator circuits implement linear filters, it would be sufficient to use one synapse per neuron, and multiplex it in time to represent n virtual synapses (where n is the number of memory cells in the corresponding row of the SRAM block). But we designed four synapse circuits per neuron, to implement both excitatory and inhibitory synapses, as well as different dynamics (synapses with different time-constants could not be time-division multiplexed). In total there are three excitatory synapse circuits with independent time constants, and one inhibitory synapse circuit. An additional element of flexibility present in the neural core is provided by a "synaptic ad-

dress-demultiplexer", which allows the user to re-configure the connectivity between the synapse circuits and the neurons (see Section III-C). This circuit is typically set once, to configure the network at start-up time, and is not changed during the experiment.

As described in Fig. 1, external input signals are encoded using a Bundled Data (BD) representation with an 18 bit wide bus for the data and two additional lines for the control signals (REQ and ACK). In the input bus 10 bits encode the X- and Y-addresses of the memory cells, five bits encode the synaptic weight values and three bits encode the synapse type to be used. Communication transactions are event-based: if the receiver's ACK signal is inactive (i.e. after previous transactions are completed), the sender can trigger a new event transaction to send data by activating the REQ signal. Given the BD representation, we make a timing assumption and assume that the data is valid, when the REQ signal arrives at the receiver end. Once the asynchronous data is received and latched on-chip, we convert the data representation from BD to Dual Rail (DR). These data encoding schemes require more routing resources (e.g., they use multiple lines per bit), but they have the advantage of being delay-insensitive and do not require any timing assumptions (the request signal, encoded by the data itself, is active only if the data is valid) [34]. In our system, we use off-chip BD representations to reduce the number of input pad requirements, and an on-chip DR representation with a "1-of-2" coding scheme (which requires two wires per bit), to implement a quasi delay-insensitive coding scheme. Once converted to DR, the input data is sent to both the SRAM and the neural-core blocks. The input to the SRAM block is used by the row and column decoders to select the proper memory cell and to read or write a five-bit synaptic weight value, depending on the status of a *Write_enable* control signal. The input to the neural-core block is used to select one of the four synapse circuits belonging to the row of the SRAM cell addressed. The content of the SRAM cell drives the DAC of the selected synapse, which in turn produces the synaptic current with the desired amplitude.

The asynchronous communication cycle is completed when the memory's content is delivered to the synapses: the synapse circuits in the neural-core block acknowledge the asynchronous controller; the controller waits for the chip REQ signal to become low and de-asserts the ACK signal; the input data to the SRAM and neural-core block becomes neutral; and the neural-core circuits de-assert the acknowledge to the controller.

III. Circuit Descriptions

A. Asynchronous Controller

The circuits that implement the asynchronous controller and the translation from the BD representation to the DR one are shown in Fig. 2. Upon receiving a *REQ* signal, a "C-element" [35] sets the acknowledge *ACK* signal (which is also the internal *PixReq* signal) to high; the BD input data is latched by *PixReq* and converted to DR representation using a "1-of-2" coding scheme (see bit lines $d0.0-d1.1$ in Fig. 2). To ensure that the buffers latch valid data at the input when the request signal arrives, we typically delay the generation of the *REQ* output on the sender side.

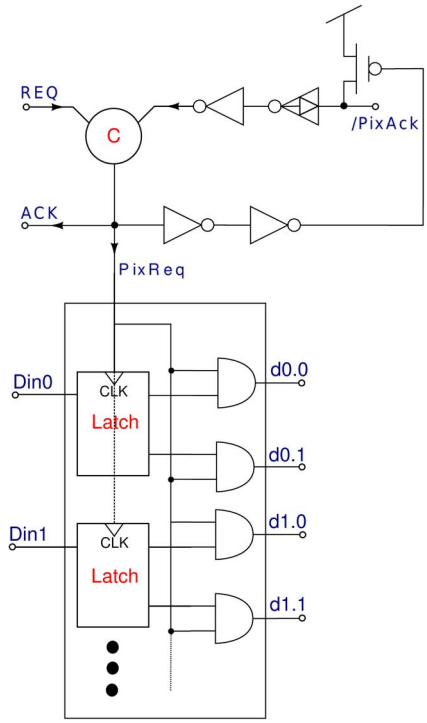

Fig. 2. Asynchronous controller circuits implement the handshaking protocol and convert the data representation from bundled-data to dual-rail.

B. Memory

The SRAM architecture is illustrated in Fig. 3. Two row and column decoders receive five bits each, encoded in dual-rail, and generate a one-hot code at the output. A standard six-transistor circuit (6T SRAM design) is used to implement the memory cells [Fig. 3(b)]. The memory array has 32×32 words, each word comprising five bits. An output filter [Fig. 3(c) and (d)] produces a dual-rail representation of the data [32]. During idle mode, when there is no input, the *Bitline* and */Bitline* signals are pulled up to VDD and the output of the filter circuits [*b0.0* and *b0.1* of Fig. 3(c)] are both set to Gnd. During a "Read" operation, the X-decoder of the memory block selects a column (via the *WL* word-line of Fig. 3(b)); the *Bitline* and */Bitline* signals of the five memory cells in the selected column are then set to values that correspond to the content of the five-bit memory word; and the Y-decoder enables the transmission gates of the corresponding row, thus allowing all the driven *Bitline* and */Bitline* signals of the selected word to reach the input of the filter circuit. Finally, the filter circuit generates dual-rail data from the *Bitline* and */Bitline* signals, setting either of the *b0.0* or *b0.1* lines to VDD, according to the content of the memory cell.

The content of the memory block is programmed by setting the *Write_enable* signal to VDD and transmitting the five bits that represent the content of the memory cells together with the standard address-event data. In this "write" mode the memory bits can drive the set of *Bitline* and */Bitline* signals belonging to the row selected by the Y-decoder input data. As the X-decoder input data enables only one of the SRAM column *WL* word-

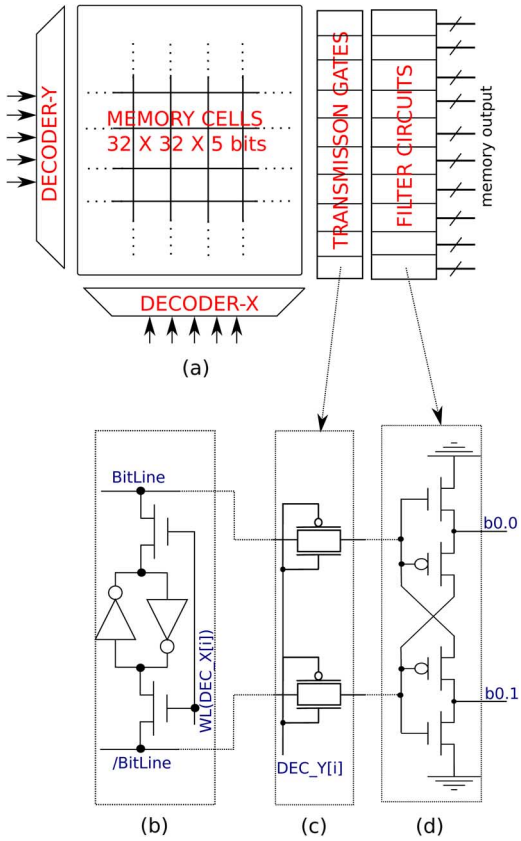

Fig. 3. (a) Memory architecture. (b) A six-transistor standard memory cell. (c) Transmission gate. (d) Memory output filter circuit to produce dual-rail data representation.

lines, only the memory cell with the corresponding X- and Y-address will change its content. In addition to being stored in the 6T SRAM cell, the content of the memory word is also passed through to the neural-core, for producing synaptic currents with the desired amplitude. This "pass-through" mode allows users to both store and set synaptic weights on-line, during normal operation of the spiking neural network, by simply writing the synaptic weight on the AER bus together with the address-event data. Conversely, in "read" mode the memory cell bits set by the external signals cannot alter the content of the memory cells. Rather, standard address-event data (e.g., produced by a silicon retina or other types of neuromorphic sensors [29]) access the data stored in the addressed memory cell to set the synapse weight and stimulate the corresponding neuron, without having to provide a synaptic weight with each event.

Another novel aspect of this architecture lies in the possibility to select a memory address column and stimulate all of the cells in all rows, effectively "broadcasting" an address-event to all neurons in the array. This is achieved by using a dedicated address in the S-decoder of the neuron block (bottom right decoder in Fig. 1): if the "broadcast" address is set in the input address-event data, the corresponding neuron block decoder output is set to one and OR'ed with the SRAM block Y-decoder outputs to enable all SRAM output transmission gates. All synapses will therefore be stimulated with the selected weights, and all neurons will receive their corresponding weighted post-synaptic current. This broadcast feature is useful in experiments in which

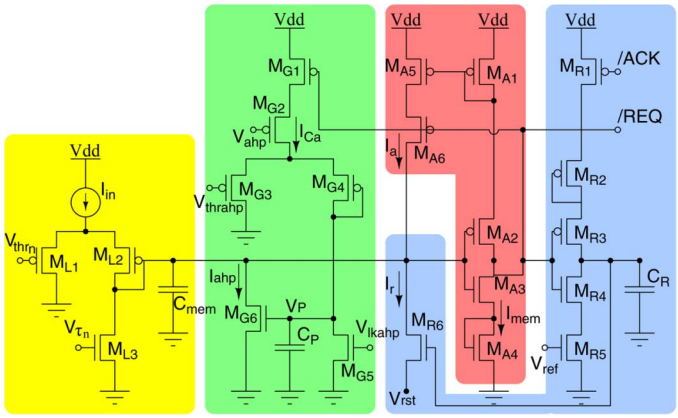

Fig. 4. Neuron circuit schematic. An input DPI low-pass filter ($\mathrm{M_{L1-3}}$) implements the neuron leak conductance. A non-inverting amplifier with current-mode positive feedback ($\mathrm{M_{A1-6}}$) produces address-events at extremely low-power operation. A reset block ($\mathrm{M_{R1-6}}$) resets the neuron to the reset voltage V_{rst} and keeps it reset for a refractory period, set by the $\mathrm{V_{ref}}$ bias voltage. An additional DPI low-pass filter ($\mathrm{M_{G1-6}}$) integrates the output events in a negative feedback loop, to implement a *spike-frequency adaptation* mechanism.

input patterns need to stimulate the synapses of all neurons in parallel.

C. Neural-Core

The neural-core block comprises 32 Integrate-and-Fire (I&F) neurons, four synapse circuits (three excitatory and one inhibitory) per neuron, and a synapse address-demultiplexer circuit.

1) Neuron Circuit: The neuron circuit is the "Adaptive exponential I&F neuron" described in [33], but with an extra free parameter corresponding to the neuron's reset potential. The circuit diagram of this new design is shown in Fig. 4. The neuron's input DPI integrates the input current until it reaches the neuron's threshold voltage. At this point there is an exponential rise due to the positive feedback in the silicon neuron's circuit that causes the neuron to generate an action potential. The membrane potential is then reset to the neuron's tunable reset potential V_{rst}.

Analogous circuits implemented in a previous chip have been shown to be extremely low power, consuming about 7 pJ per spike [36]. In addition, the circuit is extremely compact compared to alternative designs [33], while still being able to reproduce interesting dynamics, such as spike-frequency adaptation (as demonstrated in Section IV).

2) Synapse Circuit: The synapse circuit includes three main functional blocks (see Fig. 5): a DPI [37] to implement the synaptic dynamics; a DAC circuit to generate the appropriate weighted current fed in input to the DPI; and a *validity-check* circuit to activate the DAC when there is valid data at its input, and to produce an acknowledge signal fed back to the asynchronous controller.

As the output of the memory block generates valid DR representation data, the synapse validity-check block raises its *PiXAck* signal and feeds the memory content data to the DAC. The *PiXAck* signals of all synapses are *wire-OR'ed* together. The result is used by the asynchronous controller to complete

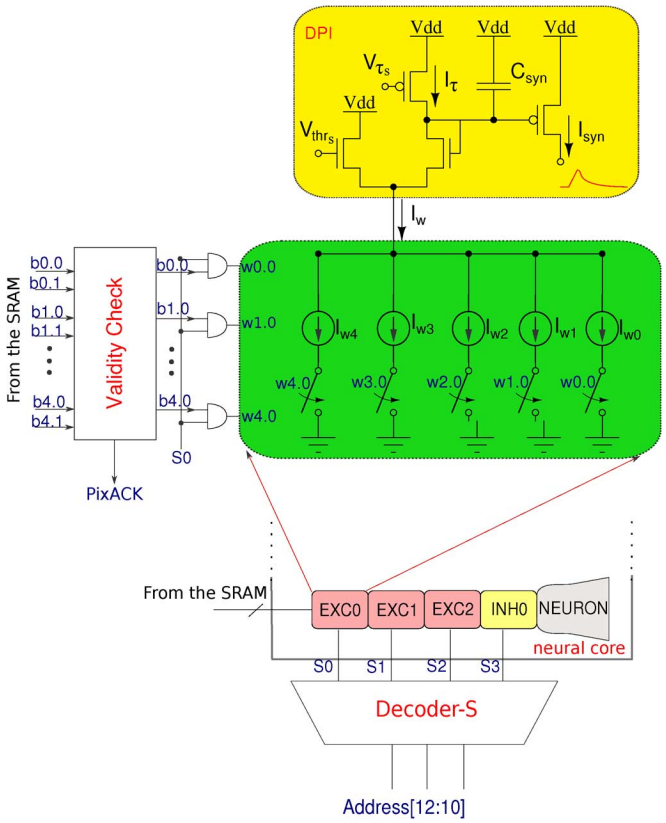

Fig. 5. Programmable synapse circuit: a synapse decoder (Decoder-S) selects the type of circuit to use (three different excitatory variants, or inhibitory). A validity-check block processes the memory block output. Upon receiving valid data (i.e., with incoming address-events) this circuit generates the acknowledge signal *PiXAck* and passes the weight-value bits on to the synapse DAC. The synaptic weight word is then converted to an analog current fed in input to a DPI synapse circuit. The DPI circuit then produces a post-synaptic current with biologically realistic temporal dynamics [37].

the handshaking cycle (see Fig. 2). The type of synapse circuit selected (different variants of excitatory/inhibitory DPI circuits) depends on the address-event data sent to the neural-core S-decoder. The asynchronous data and control paths of SRAM and neural-core blocks are independent. For correct operation, the S-decoder output should be ready before the weight bits are sent to the synapse DAC. We assume that this is true, because we make the timing assumption that the Decoder-S data path is faster than the memory access-time. The memory access time includes both the decoding time and the time required for the *Bitline* signals to be driven by the memory control circuits.

The synapse DAC circuit is activated by both the Decoder-S output and the validity-check block. The five bits that encode the weight value control switches on a corresponding number of branches, each connected to a current source, programmable via the bias-generator block [31]. In principle, for perfect binary encoding the current in each branch should be twice as large as the current in the previous branch. But we chose to have five independent current sources in order to fine tune them and compensate for mismatch effects across the synapse population. The sum of the currents from the five branches of each synapse DAC produces the final I_W current, used by the corresponding DPI synapse circuit. We bias the DPI circuit in its linear range [38] to

TABLE I
NEURON AND SYNAPSE CIRCUIT PARAMETERS. THE REFERENCE CURRENTS I_{w_i} OF THE SYNAPSE WERE TUNED TO BE $2^{i-4} I_w$

parameter	value	comment
I_w	12.5 nA	Current reference in synapse circuitry
V_{thr_s}	3.1V	Threshold bias in synapse DPI circuit
V_{τ_s}	3.1V	Synapse time-constant bias
V_{thr_n}	100mV	Threshold bias in neuron DPI circuit
V_{τ_n}	100mV	Neuron time-constant bias
V_{rst}	100mV	Neuron reset voltage
V_{ref}	200mV	Neuron's refractory period bias

implement a linear first-order low-pass filter. In this way we can use a single DPI for a row of 32 "virtual" synapses, time-division multiplexing it in time to integrate their independent contributions. The DPI output current I_{syn} will therefore be the integral of the weighted current pulses produced by the address-events sent to the memory-cells of the corresponding row.

D. Synapse Address-Demultiplexer

This circuit is a digital programmable switch-matrix that re-configures the connectivity between the synapse output nodes with neuron input ones. In its default configuration the synapse output nodes of each row are connected to the neuron input node of the same row, therefore giving rise to a network of 32 neurons, each receiving 4 $\times$ 32 virtual synaptic inputs (4 synapse types and 32 synaptic weights, stored in the SRAM cells). By changing the address-demultiplexer state, the synapse output currents can be re-routed to different subsets of neurons. From the default case in which each synapse row is connected only to the neuron of its corresponding row, one can configure the address-demultiplexer to merge pairs of synapse rows, thus routing all synaptic currents to only half the total neurons, or merge four rows and route their outputs to one fourth of the neurons, and so forth, all the way to merging all synapse row addresses to route their output currents to one single neuron.

IV. EXPERIMENTAL RESULTS

To validate the correct operation of the proposed circuits we conducted experiments in which we measured the neuron firing rates as a function of synaptic weight values and injection current. Table I shows the synapse and the neuron parameters used in these experiments.

A. Asynchronous Event-Based Communication

The first experiment shows how the circuits can operate correctly using signals with temporal characteristics that differ by more than seven orders of magnitude (i.e., from tens of nanoseconds to hundreds of milliseconds). Fig. 6 shows both the fast digital signals involved in the asynchronous communication (REQ, ACK and PixAck signals), and a slow one, involved in the generation of synaptic currents with biologically plausible time constants. The figure inset shows a single event transaction, with REQ, ACK, and PixAck signals switching over a period of less than 100 ns, following a four-phase handshaking protocol: as REQ is activated, the asynchronous controller asserts the chip ACK signal. This triggers the conversion of the input address data from BD to DR representation and within

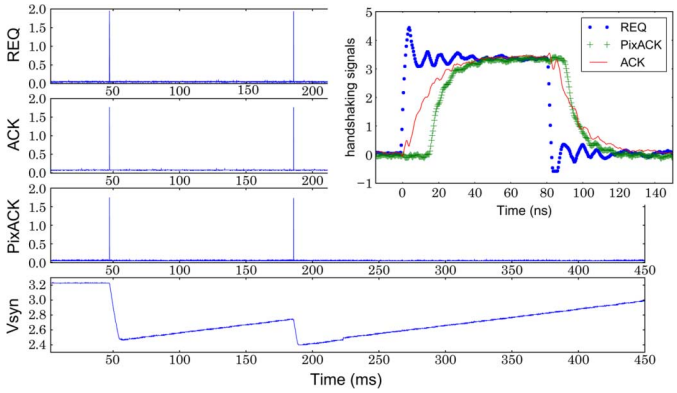

Fig. 6. Signals involved in (fast) event-based communication and (slow) synaptic dynamics: the REQ, ACK, and PixAck signals are asynchronous digital pulses that last less than 100 ns, while the V_{syn} signal is related the the post-synaptic current and lasts hundreds of milliseconds. The figure inset shows a single event transaction, with REQ, ACK, and PixAck signals switching.

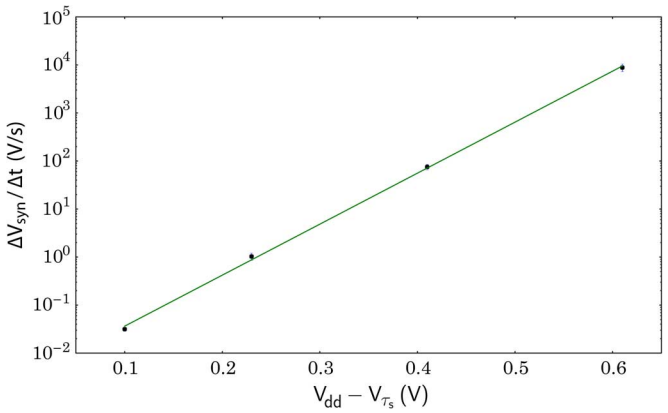

Fig. 7. Wide range of synapse dynamics. The plot shows the slope of the V_{syn} signal, measured for different values of the V_{τ_s} bias voltage. The five orders of magnitude spanned by this signal are related to the synapse time constants via equation (4).

20 to 40 ns the synapse validity-check block signal receives the memory content addressed by the input event, asserts the pixel PixAck signal, and enables the synapse DAC which stimulates the DPI circuit. The DPI produces a slow post-synaptic current (generated by a subthreshold p-type Metal-Oxide Semiconductor Field Effect Transistor (MOSFET) driven by the V_{syn} signal of Fig. 6). When the sender removes its REQ signal, the asynchronous controller resets the ACK signal, which renders the DR data invalid and therefore resets the PixAck signal in the synapse validity-check block. The speed with which this hand-shaking cycle is completed determines the chip's maximum event input rate. From the inset of Fig. 6, we estimate a maximum input rate of about 7.1 M,events/sec. However, in our setup this rate is limited by the sender's speed (the REQ signal is held active for several nano-seconds, even after the ACK signal has been raised by the receiver). Fig. 6 shows that the receiver sets the PixAck signal high about 40 ns after the arrival of the REQ signal. Therefore we estimate that in principle, the receiver chip should be able to consume events at a maximum rate of 12.5 M events/sec.

While the digital signals are required to switch as fast as possible, the analog V_{syn} signal of the synapse DPI circuit can change with different time constants, which depend on the circuit's V_{τ_s} bias voltage [37] (see also Fig. 5). It can be shown that V_{syn} is related to the circuit time constant τ_s via the following equations:

$$I_{syn}(t) = I_{syn_0} e^{-\frac{t}{\tau_s}} \quad (1)$$

$$I_{syn}(t) = I_0 e^{-\kappa \frac{V_{syn} - V_{dd}}{U_T}} \quad (2)$$

$$I_{\tau_s} = I_0 e^{-\kappa \frac{V_{\tau_s} - V_{dd}}{U_T}} \quad (3)$$

where where the term I_{syn_0} depends on the state of the DPI after a spike, I_0 represents the transistor dark current, U_T represents the thermal voltage κ the subthreshold slope factor [39], and $\tau_s = (U_T C_{syn})/(\kappa I_{\tau_s})$. From these equations we can derive

$$\tau_s = \frac{U_T}{\kappa} \frac{\Delta t}{\Delta V_{syn}}. \quad (4)$$

Fig. 7 shows the slope of the V_{syn} trace (e.g., see also bottom plot of Fig. 6) measured across all 32 excitatory synapses of the array, for different values of $(V_{dd} - V_{\tau_s})$. The data, spanning over five orders of magnitude, can be fitted by a line and related to the circuit time constant via (4).

B. Neuron Output Frequency vs Input Current

To study the relationship between the spike frequency of the neurons in response to their input currents, we ran an experiment sweeping the injection current to all neurons in the array and measured their average spike frequency and their standard deviations. Fig. 8 shows two examples of Frequency-Current (F-I) curves measured by injecting currents ranging from 10 nA to 100 nA, and measuring the neuron's firing rates for different sets of bias parameters. In a first experiment we biased the circuit to produce relatively low, biologically plausible, firing rates (see Fig. 8(a)) by setting $V_{thr_n} = 80$ mV, $V_{\tau_n} = 120$ mV, and all other bias values as in Table I. This produced a linear response, with mean frequencies measured across the 32 neurons ranging from about 5 Hz to 50 Hz, but also with an average standard deviation of 18.3%. By biasing the neurons in a regime that produced higher firing rates, we could reduce the effect of mismatch significantly [the average standard deviation in Fig. 8(b) is about 6%]. Also in this case the response is linear, but the curve saturates for high values of input currents (as observed experimentally also in real neurons [40]), thanks to the effect of refractory period circuit. We were able to reduce the effect of mismatch in these circuits, also due to careful layout design. The largest transistor (W/L) size in the neuron circuit is about 7 μm/3.5 μm while the C_{mem} capacitance is 1.6 pF the C_P capacitance is approximately 120 fF and the C_R is 28 fF. The inset of Fig. 8(b) shows a trace of the membrane potential for a constant current injection of 10 nA.

Fig. 9 shows a "raster plot" plotting the neurons output spikes (address-events) over time in response to synaptic inputs, rather than to direct current injection. The synapses were stimulated with a regular spike train of 400 Hz, generated on the workstation. All synapses were stimulated in parallel by using the "broadcasting" feature of this architecture. For the data shown

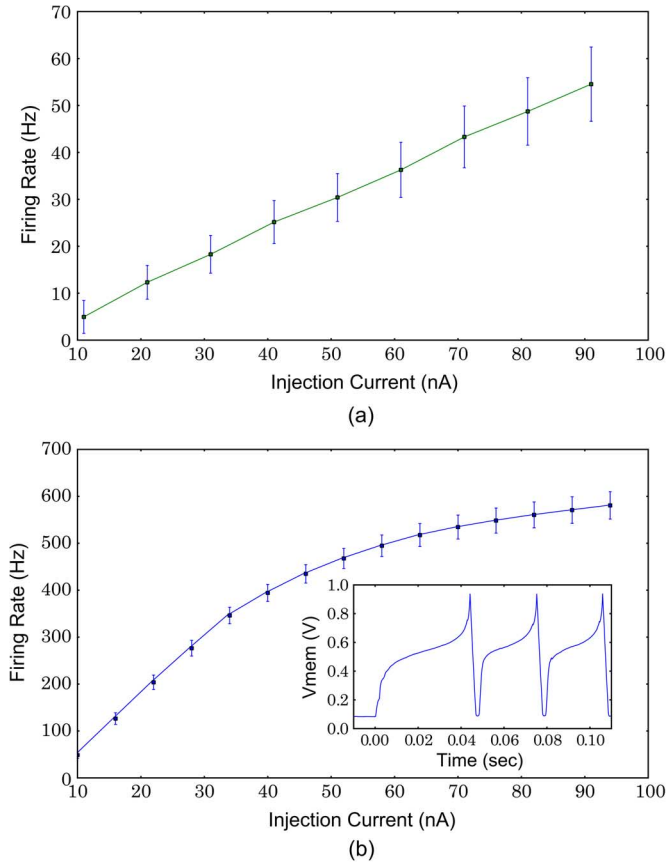

Fig. 8. F-I Curves: Firing rate averaged over all neurons in response to an injection current, measured with different neuron bias settings. (a) F-I curve measured with neuron bias settings that produce biologically realistic firing rates. (b) F-I curve for bias settings that produce higher-firing rates. The inset illustrates a trace of the membrane potential in response to a constant input current of 10 nA while refractory bias voltage is set to 120 mV.

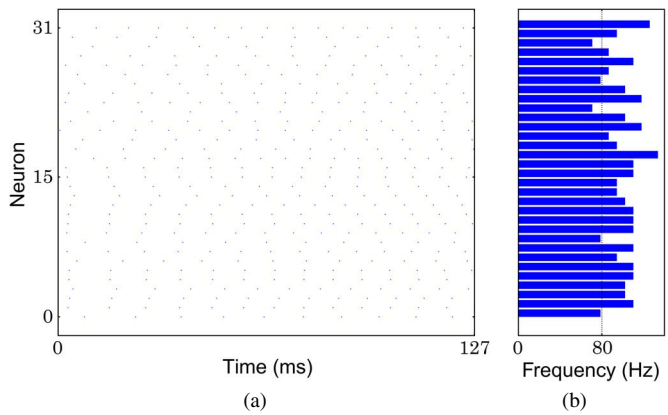

Fig. 9. (a) Raster plot of the silicon neurons, in response to regular input spike trains of 400 Hz broadcast to synapses of all rows with the same weights values of $(01001)_2$. (b) Histogram of the neuron firing rates.

in Fig. 9, all memory words were set to $(01001)_2$, the S-decoder address was set to activate the "broadcast" feature and the row decoder was set to stimulate row 10 (an irrelevant arbitrary number, given the input events are broadcast to all rows).

As shown in the figure, all neurons fire regularly at a rate of about 90 Hz. The standard deviation measured across the whole

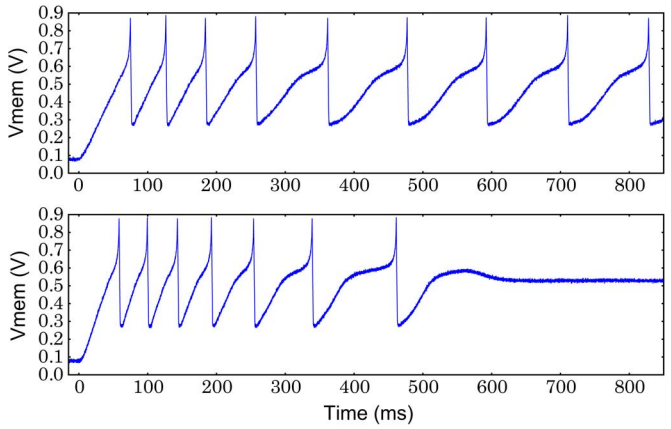

Fig. 10. Spike frequency adaptation measurements, for two different input currents: the input current was produced using a p-type MOSFET with bias voltage $V_{in} = 2.5\,\mathrm{V}$ (top plot) and $V_{in} = 2.6\,\mathrm{V}$ (bottom plot). In both cases the neuron was biased with the following settings: $V_{lkahp} = 0.057V$, $V_{ahp} = 3.05V$, $V_{thrahp} = 0.22V$, $V_{thr} = 0.09V$ and $V_{lk} = 0.08V$.

population is of approximately 16.6%. This is due to the variability induced by the AER circuits, and the mismatch of the analog circuits involved in the response of the circuit (including the input DACs and synapse DPIs).

C. Spike-Frequency Adaptation and Adjustable Reset Voltage

The spike-frequency adaptation sub-circuit of the silicon neuron ($\mathrm{M_{G1-6}}$ in Fig. 4) produces a slow current (I_{ahp} of Fig. 4) which represents the after-hyper-polarization current activated by Ca-influx during action potentials in real neurons [41]. This negative feedback mechanism introduces a second "slow" variable in the neural dynamics equation that can endow the neuron with a wide variety of dynamical behaviors, as demonstrated in [42]–[44]. Fig. 10 shows the response of the neuron to constant current injection for two different input amplitudes and for bias settings that activated the adaptation mechanism.

A parameter that plays an important role in producing different types of spiking dynamics is the neuron's reset potential: this parameter induces behaviors in bi-dimensional models that are typically only observed in higher dimensional continuous systems [45]. In many previous implementations of silicon neurons this was equal to the resting potential (and both were equal to Gnd). The silicon neuron implemented in this device has an explicit reset potential bias (V_{rst} of Fig. 4) that can be set to arbitrary values. In Fig. 11 we show how this bias voltage can be used to adjust the neuron's reset potential: at the beginning of the experiment V_{rst} is set to Gnd, so that membrane potential resting state and neuron reset potentials are both the same. After 600 m sec the reset voltage is set to 0.45 V (for the top plot of Fig. 11), or to 0.35 V (middle plot), or to 0.25 V (bottom plot).

By exploring the parameter space given by the spike-frequency adaptation voltage biases and by the reset-potential bias it is possible to produce neuron firing patterns that include *tonic*, *adapting*, *bursting*, and *irregular spiking* (e.g., see Fig. 6 of [44]). The exploration of this parameter space goes beyond the scope of this paper and will be subject to further investigations.

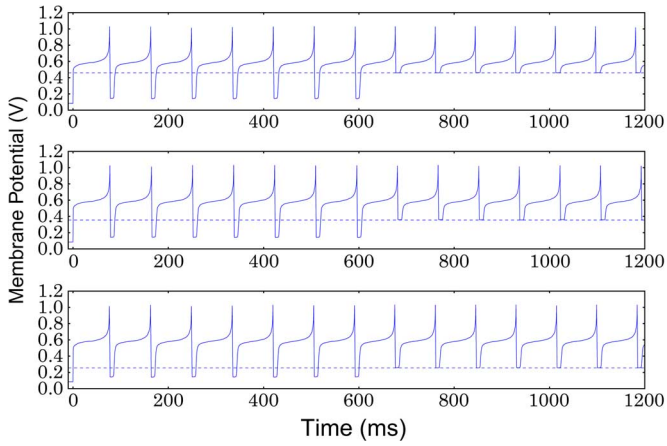

Fig. 11. Demonstration of adjustable reset voltage in the neuron circuit.

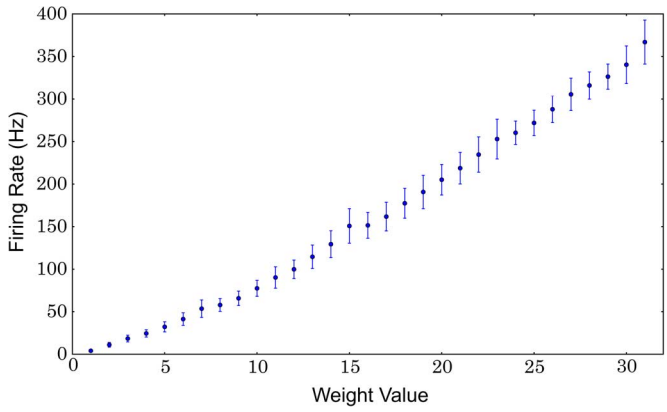

Fig. 12. Firing frequency vs. synaptic weight values in response to regular spike trains of 400 Hz.

D. Programmable Synaptic Weights

The final set of experiments we performed was meant to characterize the properties of the SRAM block and fast DAC circuits for producing weighted synaptic inputs. To examine the precision of the synapse weights encoded with the SRAM 5-bit words, we stored the 32 word values ranging from $(00000)_2$ to $(11111)_2$ in the 32 corresponding columns of the SRAM block, for each row. In this way, each neuron is effectively connected to 32 *virtual synapses* with 32 different synaptic weights. We set the nominal synaptic current I_W to 12.5 nA (see Table I), stimulated one virtual synapse of all neurons with a 400 Hz input spike train in each run, and repeated the experiment for all memory values (i.e., for all columns in the SRAM block). Fig. 12 shows the combined synapse-neuron response, averaged across the population of 32 neurons. The variability in these responses takes into account the mismatch in the neuron and synapse circuits, the mismatch and imprecision of the DAC circuits, and the temporal jitter present due to the asynchronous nature of the communication circuits used. Despite all these effects, the standard deviation in Fig. 12 is always less than $\sigma = 12.5\%$, which results in an effective resolution of $b = log_2(1/\sigma) = 3$ bits or more. This is encouraging, because it demonstrates that the approach followed can be used to implement spiking neural networks with programmable weights in a compact and efficient manner.

V. Conclusion

We proposed a novel neuromorphic VLSI device comprising both a spiking neural-core with biophysically realistic analog synapse and neuron circuits, as well as a fully asynchronous digital memory block. We showed how it is possible to integrate fast digital circuits next to very slow analog ones, using time constants that span over seven orders of magnitude, and to obtain remarkable performance figures with low mismatch. Although implementing the interface to the SRAM block and the SRAM itself could be done off-chip (e.g., using Field Programmable Gate Array (FPGA) devices), we verified in this prototype chip the correct functionality of the new asynchronous SRAM interfacing circuits. More generally, we showed experimental results that demonstrate the proper operation of all the major circuit blocks in the chip. The data of Fig. 6 demonstrate that input events are successfully transmitted through the input AER stages onto the SRAM block, that the SRAM provides in output the expected bits (previously programmed into the memory), that the synapse converts the stored digital word into a properly weighted synaptic current, and that the synaptic dynamics block has the expected slow dynamics and linear filtering properties. The data of Fig. 8(b) and Fig. 9 shows that the synaptic currents get properly integrated by the spiking neurons and that the spikes get properly converted into AER events and transmitted by the output AER stages. The experimental results shown in Fig. 10 and Fig. 11 show how the neuron implements the adaptation and reset mechanism required to produce a wide range of neural dynamics, and the data of Fig. 12 shows how it is possible to implement spiking neural networks with programmable weighted synaptic currents. The proposed chip could be used for implementing different Spike-Timing Dependent Plasticity (STDP) learning strategies, and employed to solve tasks in the context of real-time neuromorphic sensory-motor systems.

Acknowledgment

The authors would like to thank A. Whatley for constructive feedback on the manuscript, and the NCS group of the Institute of Neuroinformatics (http://ncs.ethz.ch/) for support and contributions to the development of the AER experimental setup.

References

[1] W. Maass and E. Sontag, "Neural systems as nonlinear filters," *Neural Comput.*, vol. 12, no. 8, pp. 1743–1772, 2000.

[2] A. Belatreche, L. P. Maguire, and M. McGinnity, "Advances in design and application of spiking neural networks," *Soft Comput.*, vol. 11, no. 3, pp. 239–248, Jan. 2006.

[3] R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. Bower, M. Diesmann, A. Morrison, P. H. J. F. Goodman, M. Zirpe, T. Natschläger, D. Pecevski, B. Ermentrout, M. Djurfeldt, A. Lansner, O. Rochel, T. Vieville, E. Muller, A. Davison, S. El Boustani, and A. Destexhe, "Simulation of networks of spiking neurons: A review of tools and strategies," *J. Comput. Neurosci.*, vol. 23, no. 3, pp. 349–398, Dec. 2007.

[4] J. Brader, W. Senn, and S. Fusi, "Learning real world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Comput.*, vol. 19, pp. 2881–2912, 2007.

[5] P. Rowcliffe and J. Feng, "Training spiking neuronal networks with applications in engineering tasks," *IEEE Trans. Neural Netw.*, vol. 19, no. 9, pp. 1626–1640, Sep. 2008.

[6] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S.-C. Liu, R. Douglas, P. Häfliger, G. Jimenez-Moreno, A. Civit-Ballcels, T. Serrano-Gotarredona, A. Acosta-Jiménez, and B. Linares-Barranco, "CAVIAR: A 45 k neuron, 5 M synapse, 12 G connects/s are hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, Sep. 2009.

[7] J. Wijekoon and P. Dudek, "Compact silicon neuron circuit with spiking and bursting behaviour," *Neural Netw.*, vol. 21, no. 2–3, pp. 524–534, Mar.–Apr. 2008.

[8] S. Millner, A. Grübl, K. Meier, J. Schemmel, and M.-O. Schwartz, "A VLSI implementation of the adaptive exponential integrate-and-fire neuron model," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds. La Jolla, CA, USA: NIPS, 2010, vol. 23, pp. 1642–1650.

[9] R. Silver, K. Boahen, S. Grillner, N. Kopell, and K. Olsen, "Neurotech for neuroscience: Unifying concepts, organizing principles, and emerging tools," *J. Neurosci.*, vol. 27, no. 44, p. 11807, 2007.

[10] X. Jin, M. Lujan, L. Plana, S. Davies, S. Temple, and S. Furber, "Modeling spiking neural networks on SpiNNaker," *Comput. Sci. Eng.*, vol. 12, no. 5, pp. 91–97, Sep.–Oct. 2010.

[11] J. V. Arthur, P. A. Merolla, F. Akopyan, R. Alvarez, A. Cassidy, A. Chandra, S. K. Esser, N. Imam, W. Risk, D. B. D. Rubin, R. Manohar, and D. S. Modha, "Building block of a programmable neuromorphic substrate: A digital neurosynaptic core," in *Proc. IEEE Int. Joint Conf. Neural Networks*, Jun. 2012, pp. 1946–1953.

[12] N. Imam, F. Akopyan, J. Arthur, P. Merolla, R. Manohar, and D. Modha, "A digital neurosynaptic core using event-driven qdi circuits," in *Proc. 18th IEEE Int. Symp. Asynchronous Circuits and Systems*, May 2012, pp. 25–32.

[13] Y.-X. Wang and S.-C. Liu, "Programmable synaptic weights for an aVLSI network of spiking neurons," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 2006, pp. 4531–4534.

[14] T. Pfeil, T. C. Potjans, S. Schrader, W. Potjans, J. Schemmel, M. Diesmann, and K. Meier, "Is a 4-bit synaptic weight resolution enough?—Constraints on enabling spike-timing dependent plasticity in neuromorphic hardware," *Frontiers Neurosci.* vol. 6, 2012.

[15] J. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A calibration technique for very low current and compact tunable neuromorphic cells. Application to 5-bit 20 nA DACs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 6, pp. 522–526, Jun. 2008.

[16] T. Yu and G. Cauwenberghs, "Analog VLSI biophysical neurons and synapses with programmable membrane channel kinetics," *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 3, pp. 139–148, Jun. 2010.

[17] T. Yu, J. Park, S. Joshi, C. Maier, and G. Cauwenberghs, "65 k-neuron integrate-and-fire array transceiver with address-event reconfigurable synaptic routing," in *Proc. IEEE Biomedical Circuits and Systems Conf.*, Nov. 2012, pp. 21–24.

[18] S. Ramakrishnan, R. Wunderlich, and P. Hasler, "Neuron array with plastic synapses and programmable dendrites," in *Proc. IEEE Biomedical Circuits and Systems Conf.*, Nov. 2012, pp. 400–403.

[19] Y. Wang and S.-C. Liu, "Multilayer processing of spatiotemporal spike patterns in a neuron with active dendrites," *Neural Comput.*, vol. 8, pp. 2086–2112, 2010.

[20] A. Basu, S. Ramakrishnan, C. Petre, S. Koziol, S. Brink, and P. Hasler, "Neural dynamics in reconfigurable silicon," *IEEE Trans. Biomed. Circuits Syst.* vol. 4, no. 5, pp. 311–319, Oct. 2010.

[21] P. Merolla, J. Arthur, B. Shi, and K. Boahen, "Expandable networks for neuromorphic chips," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 2, pp. 301–311, Feb. 2007.

[22] E. Chicca, A. Whatley, P. Lichtsteiner, V. Dante, T. Delbruck, P. Del Giudice, R. Douglas, and G. Indiveri, "A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity," *IEEE Trans. Circuits Syst. I, Reg. Papers* vol. 5, no. 54, pp. 981–993, 2007.

[23] S. Scholze, S. Schiefer, J. Hartmann, C. Mayr, S. Höppner, H. Eisenreich, S. Henker, B. Vogginger, and R. Schüffny, "VLSI implementation of a 2.8 gevent/s packet based AER interface with routing and event sorting functionality," *Frontiers Neurosci.*, vol. 5, no. 117, 2011.

[24] S. Moradi and G. Indiveri, "A VLSI network of spiking neurons with an asynchronous static random access memory," in *Proc. IEEE Biomedical Circuits and Systems Conf.*, 2011, pp. 277–280.

[25] S. Mitra, S. Fusi, and G. Indiveri, "Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI," *IEEE Trans. Biomed. Circuits Syst.* vol. 3, no. 1, pp. 32–42, Feb. 2009.

[26] M. Giulioni, M. Pannunzi, D. Badoni, V. Dante, and P. Del Giudice, "Classification of overlapping patterns with a configurable analog VLSI neural network of spiking neurons and self-regulating plastic synapses," *Neural Comput.*, vol. 21, no. 11, pp. 3106–3129, 2009.

[27] M. Valle, "Analog VLSI implementation of artificial neural networks with supervised on-chip learning," *Anal. Integr. Circuits Signal Process.*, vol. 33, no. 3, pp. 263–287, 2002.

[28] B. Wen and K. Boahen, "A silicon cochlea with active coupling," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, no. 6, pp. 444–455, 2009.

[29] S.-C. Liu and T. Delbruck, "Neuromorphic sensory systems," *Current Opin. Neurobiol.*, vol. 20, no. 3, pp. 288–295, 2010.

[30] K. Boahen, "Point-to-point connectivity between neuromorphic chips using address-events," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 416–34, 2000.

[31] T. Delbruck and P. Lichtsteiner, "Fully programmable bias current generator with 24 bit resolution per bias," in *Proc. IEEE Int. Symp. Circuits and Systems*, 2006.

[32] V. Ekanayake and R. Manohar, "Asynchronous DRAM design and synthesis," in *Proc. 9th IEEE Int. Symp. Asynchronous Circuits and Systems*, Vancouver, BC, Canada, May 2003, pp. 174–183.

[33] G. Indiveri, B. Linares-Barranco, T. 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, "Neuromorphic silicon neuron circuits," *Frontiers Neurosci.* vol. 5, pp. 1–23, 2011.

[34] M. Singh and S. Nowick, "High-throughput asynchronous pipelines for fine-grain dynamic datapaths," in *Proc. IEEE 6th Int. Symp. Advanced Research in Asynchronous Circuits and Systems*, 2000, pp. 198–209.

[35] A. Martin and M. Nystrom, "Asynchronous techniques for system-on-chip design," *Proc. IEEE*, vol. 94, pp. 1089–1120, 2006.

[36] P. Livi and G. Indiveri, "A current-mode conductance-based silicon neuron for address-event neuromorphic systems," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 2009, pp. 2898–2901.

[37] C. Bartolozzi and G. Indiveri, "Synaptic dynamics in analog VLSI," *Neural Comput.* vol. 19, no. 10, pp. 2581–2603, Oct. 2007.

[38] C. Bartolozzi, S. Mitra, and G. Indiveri, "An ultra low power current-mode filter for neuromorphic systems and biomedical signal processing," in *Proc. IEEE Biomedical Circuits and Systems Conf.*, 2006, pp. 130–133.

[39] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbruck, and R. Douglas, *Analog VLSI: Circuits and Principles*. Cambridge, MA, USA: MIT Press, 2002.

[40] A. Rauch, G. L. Camera, H.-R. Luescher, W. Senn, and S. Fusi, "Neocortical pyramidal cells respond as integrate-and-fire neurons to *in vivo*-like input currents," *J. Neurophysiol.*, vol. 79, 2003.

[41] D. Madison and R. Nicoll, "Control of the repetitive discharge of rat CA 1 pyramidal neurones in vitro," *J. Physiol.*, vol. 354, no. 1, pp. 319–331, 1984.

[42] E. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, 2003.

[43] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *J. Neurophysiol.*, vol. 94, pp. 3637–3642, 2005.

[44] R. Naud, N. Marcille, C. Clopath, and W. Gerstner, "Firing patterns in the adaptive exponential integrate-and-fire model," *Biolog. Cybern.*, vol. 99, no. 4–5, pp. 335–347, Nov. 2008.

[45] L. Badel, S. Lefort, R. Brette, C. Petersen, W. Gerstner, and M. Richardson, "Dynamic i-v curves are reliable predictors of naturalistic pyramidal-neuron voltage traces," *J. Neurophysiol.*, vol. 99, pp. 656–666, 2008.

Saber Moradi (S'11) received the M.S. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2008.

Currently, he is working toward the Ph.D. degree at the Institute of Neuroinformatics, Department of Electrical Engineering, Swiss Federal Institute of Technology, Zurich, Switzerland. His research activities involve asynchronous VLSI circuits, mixed signal design for neuromorphic systems, and bio-inspired circuit design.

Giacomo Indiveri (SM'06) received the M.Sc. degree in electrical engineering and the Ph.D. degree in computer science and electrical engineering from the University of Genoa, Genoa, Italy.

He was a Postdoctoral Research Fellow in the Division of Biology at the California Institute of Technology, Pasadena, CA, USA, and at the Institute of Neuroinformatics of the University of Zurich and ETH Zurich, Zurich, Switzerland, where he attained the Habilitation in Neuromorphic Engineering in 2006. His research interests lie in the study of real and artificial neural processing systems, and in the hardware implementation of neuromorphic cognitive systems, using full custom analog and digital VLSI technology.

Prof. Indiveri is a Fellow of the European Research Council.