An Event-Based Digital Time Difference Encoder Model Implementation for Neuromorphic Systems

Daniel Gutierrez-Galan, *Member, IEEE*, Thorben Schoepe, *Member, IEEE*, Juan P. Dominguez-Morales, Angel Jimenez-Fernandez, Elisabetta Chicca, *Senior Member, IEEE*, and Alejandro Linares-Barranco, *Senior Member, IEEE*

***Abstract*— Neuromorphic systems are a viable alternative to conventional systems for real-time tasks with constrained resources. Their low power consumption, compact hardware realization, and low-latency response characteristics are the key ingredients of such systems. Furthermore, the event-based signal processing approach can be exploited for reducing the computational load and avoiding data loss due to its inherently sparse representation of sensed data and adaptive sampling time. In event-based systems, the information is commonly coded by the number of spikes within a specific temporal window. However, the temporal information of event-based signals can be difficult to extract when using rate coding. In this work, we present a novel digital implementation of the model, called time difference encoder (TDE), for temporal encoding on event-based signals, which translates the time difference between two consecutive input events into a burst of output events. The number of output events along with the time between them encodes the temporal information. The proposed model has been implemented as a digital circuit with a configurable time constant, allowing it to be used in a wide range of sensing tasks that require the encoding of the time difference between events, such as optical flow-based obstacle avoidance, sound source localization, and gas source localization. This proposed bioinspired model offers an alternative to the Jeffress model for the interaural time difference estimation, which is validated in this work with a sound source lateralization proof-of-concept system. The model was simulated and implemented on a field-programmable gate array (FPGA), requiring 122 slice registers of hardware resources and less than 1 mW of power consumption.**

***Index Terms*— Digital design, event-based processing, neuromorphic systems, spiking neuron, time difference encoder (TDE).**

I. Introduction

MODERN computers are based on the von Neumann architecture. Over the years, developments in traditional computing technologies have been focused on increasing the speed of computation. Unfortunately, the central processing unit (CPU) speed achieved does not match the memory access speed (leading to the so-called von Neumann bottleneck [1]), and the gap has increased exponentially over the years [2]. Furthermore, Moore's law might come to an end in the next years due to physical constraints, such as increasing thermal noise [3]. These developments have triggered increasing research efforts for the design of alternative computational architectures, which may complement and augment traditional von Neumann machines.

An interesting architecture is a human brain, which can compute complex correlations in real time with approximately 100 billion neurons and more than 100 trillion synapses while consuming only 20 W [4]–[6]. Given the ability of the brain and the fundamentally different substrate (asynchronous operation, colocalization of memory and computation, full parallelism, and so on), neural computation is a promising source of inspiration. The capability to make complex decisions in real time by means of limited sensory data poses the basis for the development of a new generation of edge computing devices [7]. To this end, researchers have aimed at modeling specific parts of the brain with artificial neural networks (ANNs). For doing so, neural connectomics and neurophysiological data acquired from biology have been studied intensely to extract the most important characteristics of neural computation. These findings led to the creation of various artificial synapse and neuron models at different levels of abstraction. In this context, the spiking approach observed in the nervous system can offer considerable advantages in terms of latencies, power consumption, and compactness

This work was supported in part by the Spanish Grants Sistema Cognitivo de Fusión Sensorial de Visión y Audio por Eventos (COFNET) under Grant TEC2016-77785-P and Percepción y Cognición Neuromórfica para Actuación Robótica de Alta Velocidad (MINDROB) under Grant PID2019-105556GB-C33/AEI/10.13039/501100011033, in part by the Cluster of Excellence Cognitive Interaction Technology (EXC 277), Bielefeld University through German Research Foundation, in part by the CogniGron Research Center, and in part by the Ubbo Emmius Funds (University of Groningen). The work of Daniel Gutierrez-Galan was supported in part by the "Formación de Personal Investigador" Scholarship from the Spanish Ministry of Education, Culture, and Sport and in part by the "Salvador de Madariaga" Program in the Faculty of Technology, Bielefeld University. The work of Juan P. Dominguez-Morales and Alejandro Linares-Barranco was supported in part by the "Salvador de Madariaga" Program in the Faculty of Technology, Bielefeld University. *(Corresponding author: Daniel Gutierrez-Galan.)*

Daniel Gutierrez-Galan, Juan P. Dominguez-Morales, Angel Jimenez-Fernandez, and Alejandro Linares-Barranco are with the Robotics and Technology of Computers Laboratory and the SCORE Laboratory, I3US, Universidad de Sevilla, 41012 Seville, Spain (e-mail: dgutierrez@atc.us.es).

Thorben Schoepe is with the Faculty of Technology and the Cognitive Interaction Technology Center of Excellence, Bielefeld University, 33619 Bielefeld, Germany, and also with the Bio-Inspired Circuits and Systems Laboratory, Zernike Institute for Advanced Materials and the Groningen Cognitive Systems and Materials Center, University of Groningen, 1022 9701 Groningen, The Netherlands.

Elisabetta Chicca is with the Bio-Inspired Circuits and Systems Laboratory, Zernike Institute for Advanced Materials and the Groningen Cognitive Systems and Materials Center, University of Groningen, 1022 9701 Groningen, The Netherlands.

provided by a digital implementation optimized for the target task [8]. Representative examples of commonly used spiking neuron models are the leaky integrate-and-fire (LIF) and the Izhikevich models, which have already been implemented in field-programmable gate array (FPGA) [9]–[11], other digital circuits [7], [12], and also analog circuits [13].

The communication between neurons is carried out by means of voltage pulses (known as spikes), which is the dominant mode of information transfer in the vertebrate nervous system. The nervous system is excited with external stimuli through the senses, such as vision and hearing. However, this sensory information is in the analog domain and needs to be encoded into the spike domain. For this task, two main coding schemes can be found in the literature. The first one, called rate coding [14], [15], encodes the information in the number of spikes fired over a time period. This value is also known as the firing rate and can be used for simple signal classification [16]. The second one, where the information is encoded in the time between two consecutive spikes, is called temporal coding [17]. This temporal encoding exists in spiking neural networks (SNNs) [18], where the information is encoded using pulse rate or time interval between pulses, and it can also be found in most event-based sensory processing stages and other applications. For instance, it is used in reservoir computing systems [19], sensory preprocessing in SNNs, and encoding of fast visual stimuli through the latency to first spike (LFS) coding [20].

Precise timing of spikes can also be used for general purpose computation. Lagorce *et al.* [21] proposed a general framework for computing every known mathematical function using a neural architecture. This framework makes use of precise timing, transmission delays, and synaptic diversity. The precise time interval between two spikes is used to represent the value of variables. The minimum and maximum interspike interval (ISI) and an elementary time step are fixed in this framework.

The temporal responses of different neuron populations to a dynamic input stimulus provide spatiotemporal information, which is self-contained in their output spike trains. In addition, time difference, or signal temporal correlation between spike trains of consecutive neurons within a network, contains relative information about the nature of the stimulus, which can be useful for complex postprocessing tasks, such as object tracking and autonomous navigation [22].

A recently proposed model that computes temporal dependencies in SNNs is the spiking elementary motion detector (sEMD) [22]. In that work, the sEMD model consisted of two parts: an event-based vision sensor as input and the time difference encoder (TDE) as a sensory preprocessing unit. The TDE unit translates the time difference between two events into a burst of output spikes. Both the number of output spikes and the duration of the burst produced by the model directly reflect the temporal correlation of two input signals, and it is inversely proportional to the time difference. Milde *et al.* developed an analog complementary metal-oxide-semiconductor (CMOS) implementation of the TDE, characterized its performances on silicon, and applied it to the encoding of optical flow (OF). The TDE model has already been used for processing visual [23], auditory, and olfactory information. Its universal applicability has great potential for inspiring innovative preprocessing for SNNs, especially supporting closed-loop neuromorphic systems with low latency requirements.

This wide range of possible applications poses a challenge in terms of time resolution and scalability. Time resolution in analog circuits is constrained by the size of the capacitors. Therefore, for high time constants applications, large capacitors would be needed. Furthermore, mismatch problems and parameter setting difficulties may appear due to the analog nature of the implementation. In this work, a generic, event-based, digital implementation of the TDE model is presented. Its time resolution is configurable by means of a clock divider, covering a time range from nanoseconds to seconds. Moreover, the model can be deployed on FPGA-based platforms. This computational platform suits the integration of SNNs very well due to its highly parallel, low-latency nature. This TDE implementation facilitates the development of complex and reconfigurable neuromorphic networks receiving input from event-based sensors, such as bioinspired retinas [24] and cochleas [25]–[28]. Finally, the TDE's performance was evaluated in simulation by characterizing its response to synthetic input stimuli and also to real-world recordings from a neuromorphic auditory sensor (NAS).

The main contributions of this work include the following.

1) The digital TDE model implementation as an alternative of the analog version for event-based real-time neuromorphic applications with different time constants.
2) The simulation and full characterization of the proposed model, verifying the basic cases and analyzing its response for complex input stimuli.
3) The deployment into an FPGA-based board, thus having flexibility for further designs, with a power consumption of less than 1 mW for each TDE unit and allowing up to 400 units in basic FPGA chips.
4) A proof of concept of a sound source lateralization task using the proposed model, where the events were received in real time from an NAS, providing a new alternative to the state of the art of sound source localization systems.

The article is organized as follows. Section II details the operating principles of the TDE model based on the original model, as well as its design constraints. Section III shows the digital architecture of the model and describes how it works. Section IV presents the full characterization results of the proposed model. Finally, in Section V, the results are compared with the original model, and its usability is discussed.

II. TDE Model Operating Principle

The TDE model [22] translates the temporal difference between two input events into a short burst of output digital pulses. It comprises two inputs: the facilitatory pulse (faci) and the trigger synapse (trig), as well as one spiking output shown in Fig. 1(a). When an event arrives at the facilitatory input, an exponentially decaying facilitatory variable is generated, called gain. If an event enters the trigger synapse shortly

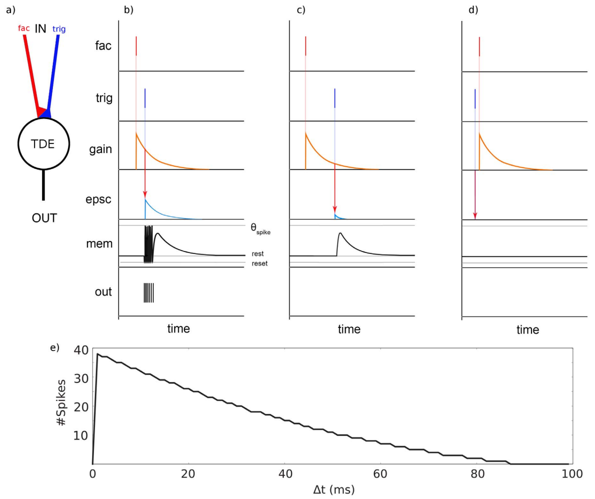


Fig. 1. Theoretical behavior representation of the TDE model based on the model proposed by Milde *et al.* [22]. (a) TDE schematic with facilitatory (fac) and trigger (trig) input and spiking output. (b) Case one: small positive time difference between facilitatory and trigger spikes leads to a high number of output spikes (out). (c) Case two: large positive time difference leads to no output spikes. (d) Case three: negative time difference leads to no output spikes. (e) Number of TDE output spikes in dependence of time difference Δt between two input events (gain: gain factor, epsc: exponential postsynaptic current, and mem: membrane potential).

after (i.e., small time difference Δt), as shown in Fig. 1(b), an excitatory postsynaptic current (EPSC) is produced. In this process, the amplitude of the EPSC depends proportionally on the facilitatory variable value, i.e., on the gain factor. Therefore, the EPSC's amplitude decreases with increasing time difference [see (1)]. The trigger synapse projects onto an LIF neuron that integrates the postsynaptic currents in its membrane potential (V_{mem}) [see (2)]. Every time V_{mem} reaches the spiking threshold τ_{spike}, a digital output pulse is released. The number of spikes generated is antiproportional to the time difference between the two input spikes [see Fig. 1(e)]. When the time difference between the facilitatory and the trigger pulses is long, the gain value at the time of arrival of the trigger signal is not high enough to generate an EPSC. Thus, no spikes are generated. In the case of a negative time difference [an event arrives first at the trigger synapse and then at the facilitatory input, as shown in Fig. 1(d)], with no output spikes being produced. Therefore, the TDE is direction-selective

$$I_{e2} = I_{e2} + (w_{e2} \times I_{e1}) \tag{1}$$

$$\frac{dV}{dt} = \frac{1}{C_m} \times \left(I_e - (I_{lk} - I_i) \times \left(1 - e^{\frac{-V}{U_t}}\right)\right). \tag{2}$$

As introduced in Section I, both the number of spikes within the burst and the burst duration depend on the time difference between the input pulses. The detection time range of the analog CMOS TDE implementation ranges from 10 ns up to hundreds of milliseconds, according to [22]. This range can be tuned by adjusting the LIF neuron's parameters in order to detect the timing differences more precisely in an accurate spectrum, thus obtaining different TDE response profiles. Those profiles are known as tuning curves, which represents the neuron responses against the time difference between the facilitatory and the trigger input pulses. Furthermore, a nonlinear behavior of the tuning curves was expected to be obtained from the analog CMOS TDE implementation due to the transistors; however, a linear profile was observed. Milde *et al.* highlighted in [22] that the nonlinear response was manifested at the population response level and in the temporal evolution of the ISI distribution within a burst. This feature was taken into account for the digital model design proposed in this work, since it determines the way in which the temporal modules are implemented.

III. Architecture and Implementation

The proposed architecture is shown in Fig. 2. There are two event-based inputs: the facilitatory input (*"facilitatory"*) and the trigger input (trigger). In addition, four configuration signals are available to set the model's parameters: the facilitatory weight, called *"detection_time,"* which defines the maximum time difference that the model is able to encode, i.e., the time during which the gain value is nonzero; the gain factor that influences the trigger synaptic weight ("tau"); the gain factor that influences the spike generation process ("weight"); and the decay time factor (decay) of the EPSC signal value. As output, there is a single event-based signal ("spike"), which is the spike fired by the encoder. Beyond those signals, the system is governed by the system clock signal ("clock").

Both control- and event-based signals have 1-bit width. Internal data lines, as well as the *"detection_time"* signal, have n-bit width, with n being a generic parameter of the model denoted by *"NBITS."* The rest of the data lines has m-bit width, with m being also a generic parameter of the model denoted by *"LOG2NBITS,"* which represents the result of the $\log_2 NBITS$. By default, the *"NBITS"* value is set to 16; thus, *"LOG2NBITS"* is set to 4. Data width plays a key role in the model behavior since it defines the timing resolution and affects the output response due to the implementation details of the spike generator module.

By following the schematic presented by Milde *et al.* [22], the proposed architecture was divided into three computational blocks: the gain generation, the EPSC generation, and the spike generation, as shown in Fig. 2 in red, blue, and green, respectively. A phenomenological design strategy was followed to implement the digital TDE model in order to avoid the computation of differential equations. Therefore, no floating-point operations were employed. Instead, integer values were used. This approach was successfully adopted in Frenkel *et al.* [10], where linear operations were performed, thus reducing both the hardware cost and the model complexity.

A. Gain-Generator Block

When an event is received at the facilitatory synapse, an exponentially decaying signal is generated (called gain). The decay time constant and the input synaptic weight determine the maximum time in which the facilitatory synapse current is not zero, i.e., the maximum time difference that the TDE is able to detect. In addition, if more than one event arrives at the facilitatory input consecutively, while the gain is higher than zero, the resulting gain value is the sum of

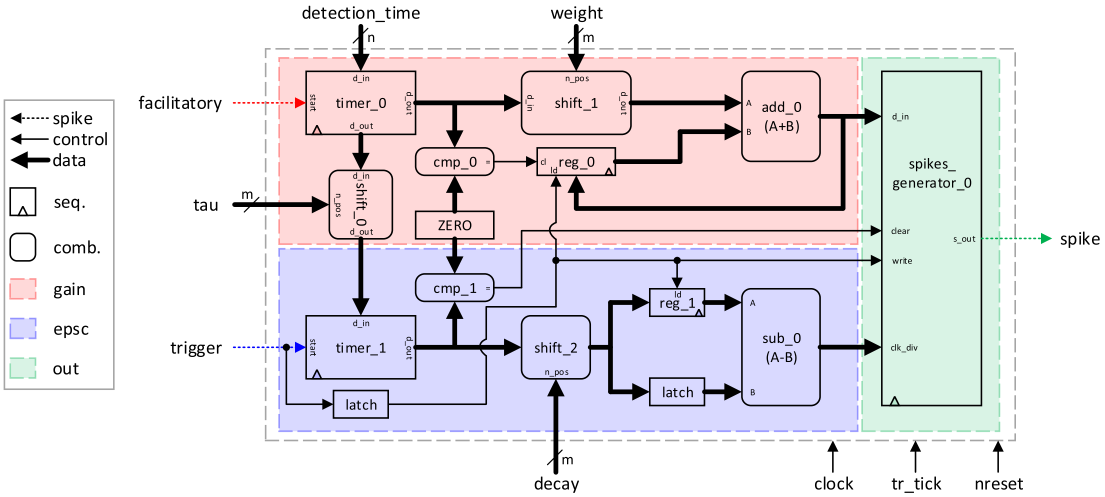


Fig. 2. Detailed block diagram of the TDE digital architecture. It is composed of three main blocks: gain-generator block (red), EPSC-generator block (blue), and spike-generator block (green). Synchronous modules are indicated by squared corner blocks with a small triangle, while asynchronous modules are indicated with rounded corner blocks. The spike arrow is a 1-bit width signal, where events are either received as input or sent as output. Control arrows are also 1-bit width, and they act as flags. Finally, data arrows can be either n-bit or m-bit width, being used for the internal communication between blocks and also for loading configuration values.

the remaining gain value and the new gain value generated due to the input event. Therefore, a feedback mechanism is needed. In order to prevent the overflow effect, the gain block saturation level is controlled by the GAIN_GEN_SAT parameter.

The decaying signal was implemented as a decreasing linear function by means of a countdown timer with preload value (represented by the timer_0 module in Fig. 2). The preload value establishes the initial configuration of the timer, i.e., the amount of time that the timer is activated. Thus, this temporal window restricts the maximum time difference that the model is able to encode. The input signal *"detection_time"* sets that value, and it can be updated in real time.

The feedback feature is achieved by internally appending an adder to the timer where its inputs are the timer's output and the aforementioned *"detection_time"* signal value, and the output is the timer's load value. Therefore, for each rising edge of the time reference signal *"tr_tick,"* the timer_0 module is updated according to the following:

$$\mathrm{timer\_0}[k] = \begin{cases} \mathrm{timer\_0}[k-1] + d\_t & \text{if faci} == 1 \\ \mathrm{GAIN\_GEN\_SAT} & \text{if satu} == 1 \\ \mathrm{timer\_0}[k-1] - 1 & \text{if } \mathrm{timer\_0}[k-1] > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where timer_0$[k]$ is the timer's output value at the time reference tick k, $k-1$ is the previous time reference tick, d_t is the unsigned integer constant value defined by the *"detection_time"* signal, faci corresponds to the *"facilitatory"* input signal, and satu is a flag that is activated when the condition $(\mathrm{timer\_0}[k-1] + d\_t) >= \mathrm{GAIN\_GEN\_SAT}$ is true.

Two clock domains were used to implement the digital TDE model. The main clock signal, as defined in Fig. 2 as "clock," governs the control processes of the sequential blocks, as well as the input events' detection and the output events' generation. Furthermore, a second clock signal, called *"tr_tick,"* is provided as a time reference tick in order to allow the model to operate with different time scales, thus achieving an operational time range between nanoseconds and seconds. With this, the model acquires enough flexibility to be used along with a wide set of neuromorphic sensors, which can operate at different time resolutions. No internal clock generator was implemented. Instead, an external configurable clock frequency divider is needed when a TDE module is instantiated. When multiple instances of a TDE unit are present, a shared clock frequency divider can be used rather than a single one per unit, thereby reducing the overall hardware resources' consumption and increasing the number of units that can fit into a design.

While, in the standard LIF neuron model [29], each synapse outputs a postsynaptic current that integrates onto the membrane potential, the TDE facilitatory block generates a gain factor that regulates the trigger synapse weight. Therefore, to cover this feature in the proposed architecture, two mechanisms were implemented. First, the timer_0 output is weighted by the input signal "tau" in such a way that the timer value is either right or left shifted by "tau" positions. The shift operation was implemented according to the Barrel shifter [30], represented as follows:

$$d\_\mathrm{out} = \begin{cases} d\_in * 2^{n\_\mathrm{pos}} & \text{if } l\_r == 0 \\ d\_in / 2^{n\_\mathrm{pos}} & \text{if } l\_r == 1 \end{cases} \quad (4)$$

where *d_out* is the output value, *d_in* is the input data, *n_pos* is the number of positions to shift the input data, and *l_r* is for selecting whether the signal has to be shifted either to the left or to the right. Since this is a combinational circuit, the output result is available at the same clock cycle, and thus, sequential blocks are not required for the synchronization. The computed value is then fed as input of timer_1, which generates the EPSC signal, acting as the trigger synapse weight.

Second, timer_0's output value is weighted by the input signal "weight" also through a Barrel shifter module. In this

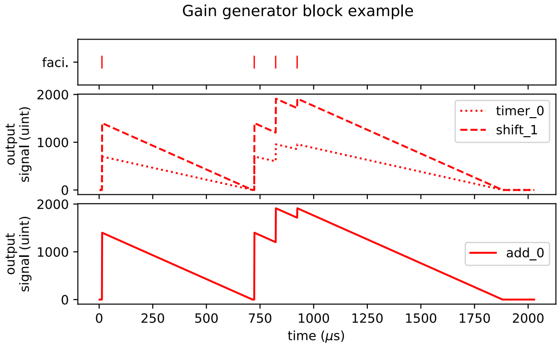


Fig. 3. Gain-generator block output example. First, the block receives a single facilitatory event. Immediately after, timer_0 is loaded with the *"detection_time"* signal's value. The output value of timer_0 decreases by one unit for each time reference tick, which was set to microseconds. Since the "weight's" value was set to 1 (meaning that the timer's value is left shifted by one position), the *shift_1*'s output value is twice the value of timer_0. Then, the block receives multiple facilitatory events in order to show the accumulative behavior. The *add_0* plot shows the value that is used as input for the spike-generator block. In this case, its value matches the *shift_1*'s value since no trigger event was received.

case, the result influences the spike generation process in such a way that it controls the number of spikes to be generated and, therefore, the precision of the encoding.

The shifted value is read by the spike generator block when a trigger pulse is detected. Based on the operating principles of the TDE model, as detailed in Section II, it can be deduced that the ratio of the number of spikes to the duration of the whole burst is proportional to the time difference between the facilitatory input and the trigger input. For short time differences, the model will produce many output spikes over a longer time bin, and for long time differences, it will produce fewer spikes but in a shorter time bin.

A register is included to store the last value used as input for the spike generator. Thus, it can be used as feedback value to be added to the gain value, increasing the final gain value and, therefore, increasing the output spike rate. Equation (5) describes the gain feedback register, as identified in Fig. 2 as reg_0

$$\text{reg\_0} = \begin{cases} d\_in & \text{if trigger} == 1 \\ 0 & \text{if timer\_0}[k] == 0 \\ \text{reg\_0} & \text{otherwise} \end{cases} \tag{5}$$

where reg_0 is the value stored by the register, *d_in* is the last value loaded on the spike generator block, trigger corresponds to the input signal trigger, and timer_0[k] is the timer's output value at the time reference tick k. This register is reset to zero when the timer_0 reaches zero, and it can also be disabled if needed.

Conceptually, it can be affirmed that the gain value influences both the temporal aspect (through the tau factor) and the amplitude aspect (through the weight factor) of the TDE response. Fig. 3 shows a response example of the gain-generator block when both single and multiple facilitatory inputs are provided.

B. EPSC-Generator Block

Similar to the gain factor generation, when an incoming event is detected at the trigger's input by the analog implementation from [22], an exponentially decaying signal is generated, known as EPSC. In the field of neuroscience, the EPSC is defined as the current coming from an artificial synapse that integrates onto a neuron's membrane potential. The amplitude of the EPSC is proportional to the gain signal due to the influence of the facilitatory block over the trigger synaptic weight [22]. Thus, the smaller the arriving time difference (Δt) between the facilitatory and trigger events, the higher the gain factor, and therefore, the higher the amplitude of the trigger synaptic current.

Consequently, if a trigger pulse is detected without any previous facilitatory pulse, no EPSC current is generated since the gain factor is zero, as shown in Fig. 1(c). Nevertheless, if a trigger pulse is detected shortly after a facilitatory pulse (i.e., low Δt), an EPSC current proportional to the gain signal value at that time is generated. The generated EPSC current is high enough to generate spikes when it is integrated onto the membrane potential, as shown in Fig. 1(b). Equivalently, if a trigger pulse is detected long after the facilitatory pulse (i.e., large Δt), the resulting EPSC current may not be enough to produce output spikes, as shown in Fig. 1(c).

Multiple events can arrive at the trigger synapse, while the gain factor is higher than zero, thus producing an accumulated EPSC current signal. The resulting signal is the sum of the left over EPSC current value and the left overweighted gain current value. Therefore, a feedback circuit is needed to limit the output current. The feedback value tends to decrease due to the decaying gain factor. However, a high input spike rate may saturate the EPSC current generation. This saturation level is set by the TDE generic parameter EPSC_GEN_SAT.

Following the same implementation principle of the gain generator block, the EPSC decaying signal was implemented as a decreasing linear function also by means of a countdown timer with preload value, identified as timer_1 in Fig. 2. In this case, we can affirm that the preload value is the remaining time to zero of the gain-generator block timer (timer_0), i.e., the gain current signal is zero. In order to maintain the synchronization with the gain generator block, the timer_1 module is updated at every rising edge of the time reference signal *"tr_tick"* according to the following:

$$\begin{aligned} &\text{timer\_1}[k] \\ &\quad = \begin{cases} \text{timer\_1}[k-1] + & \\ \left(\text{timer\_0}[k-1] \middle/ 2^{\text{tau}}\right) & \text{if trigger} == 1 \\ \text{EPSC\_GEN\_SAT} & \text{if satu} == 1 \\ \text{timer\_1}[k-1] - 1 & \text{if timer\_1}[k-1] > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{6}$$

where timer_1[k] is the timer's output value at the time reference tick k, $k-1$ is the previous time reference tick, tau is a factor to weight timer_0's output value, timer_0[$k-1$] is timer_0's output value at the previous time reference tick, trigger corresponds to the input signal trigger, and satu is a flag that is activated when the condition (timer_1[$k-1$] + (timer_0[$k-1$] × 2^{tau})) >= EPSC_GEN_SAT is true.

As previously mentioned, the trigger timer determines the output spike burst duration in the same way the EPSC synaptic current decay in [22] is set by a voltage parameter. This synaptic current is injected into the neuron that integrates the current until it generates a spike as soon as the membrane potential rises above its threshold. Therefore, the neuron is able to produce spikes, while the EPSC signal is higher than zero. In addition, the number of generated spikes is directly proportional to the EPSC signal duration and, thus, inversely proportional to the time difference between the facilitatory and trigger input spikes.

In the proposed design, the value that is loaded in timer_1 (i.e., in the EPSC generator) is called the remaining time to zero. This value influences the spike-generator block in two similar aspects. First, the spike generator block is active, i.e., producing spikes, while the EPSC value is higher than zero. Thus, it acts as an enable signal. Moreover, the remaining time to zero is used to handle the temporal evolution of the spike-generation process in order to mimic the ISI increment of the original analog TDE model [22] by exploiting a feature of the exhaustive synthetic spikes generator models proposed in [14] and implemented in [31] and [32].

The temporal evolution can be adjusted by the factor "decay," which weights the timer_1's output also by means of a Barrel shifter module identified in Fig. 2 as *shift_2*. The timer's value decreases by one unit for each *"tr_tick"* rising edge. With this factor, we can scale the decreasing speed, allowing us to obtain a different range of values, although preserving the time bin, i.e., the activation time of the spike generator module. Further details about the effect of this parameter are discussed in Section III-C.

Due to the implementation details of the spike-generator block, the generated EPSC signal needs to be inverted, thus obtaining an incremental signal instead of a decreasing signal. This transformation can be achieved by storing the reference value and periodically subtracting the original value every time it is updated. In this case, the reference value corresponds to the preload value of timer_1 when a pulse is detected at the trigger input, and the original value is the timer_1's output value. A generic register, denoted by reg_1 in Fig. 2, was added to the proposed architecture, and its behavior is described as follows:

$$\text{reg\_1} = \begin{cases} \text{timer\_1}[k] * 2^{\text{decay}} & \text{if trigger} == 1 \\ \text{reg\_1} & \text{otherwise} \end{cases} \tag{7}$$

where reg_1 is the value stored by the register, decay is the factor that weights the timer_1 value, and trigger corresponds to the trigger input signal.

Note that the trigger signal is latched to let the timer load the preload value and output the correct value, which takes one clock cycle of the main clock signal ("clock"). This latched trigger signal is shared by reg_1, reg_0, and the *spike_generator_0* modules to keep the synchronization and operate with the precise values.

The output of the subtractor module, whose output ranges from zero to reg_1's output value, is then used as input of the spike-generator module, which generates spikes according to both the *add_0*'s output value and the *sub_0*'s output value, i.e., the gain factor value and the EPSC factor value, respectively. Fig. 4 shows a response example of the EPSC-generator block when both single and multiple trigger inputs are provided after the arrival of a single facilitatory event.

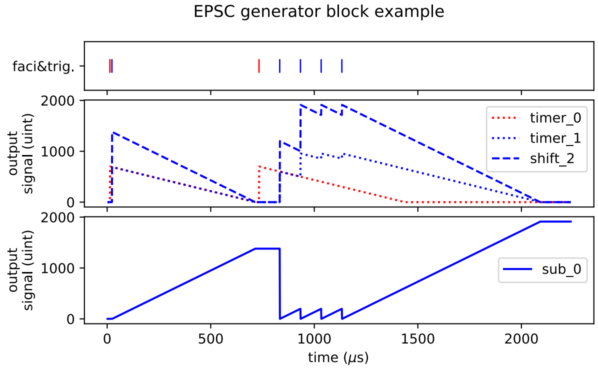


Fig. 4. EPSC-generator block output example. First, the model is stimulated with a single facilitatory event before the first trigger event. Then, the current value of *shift_1* is loaded in timer_1. The accumulative effect is also shown when multiple triggers are received. The *sub_0* module generates an increasing signal, which is used as the clock divider value for the spike generator block.

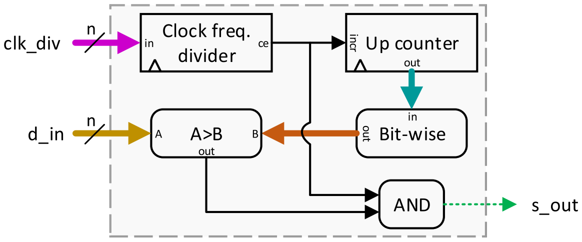


Fig. 5. EU-SSG block diagram. A complete description of both the implementation and the behavior of this module is presented in [31].

C. Spike-Generator Block

In the presence of an input facilitatory spike and an input trigger spike, a burst of output spikes is produced by the spike generator block. As detailed in Sections III-A and III-B, both the amplitude and the duration of the burst depend on the generated gain factor and the generated EPSC factor, respectively. In contrast to the LIF neuron, which integrates the presynaptic current into the membrane and produces a spike if the membrane potential reaches a threshold, an event-based unsigned integer-to-spike converter was implemented based on the model implemented by Jimenez-Fernandez *et al.* [31].

This converter, called exhaustive unsigned synthetic spikes generator (EU-SSG), takes an unsigned integer value as input and produces a burst of spikes where both the number of output spikes and their distribution over time are proportional to the input value. Similar to the synaptic current integration process, which polarizes the membrane, producing a membrane potential, the EU-SSG implements an up counter that increases its value every time a pulse is detected, as shown in Fig. 5 as *up_counter*. Then, the output of the counter is processed by the *exhaustive synthetic method (ESM)* block, which determines the integration method. Finally, analogous to the comparison between the firing threshold and the membrane potential, the counter's output value and the input data are compared, and a spike is fired only when both values are equal.

To generate the pulse according to the desired output firing rate (expressed in spikes per second), a clock frequency divider module is used, identified as *clk_freq_div* in Fig. 5. It has two input signals: the system clock ”clock” signal and the *“clk_div”* signal, which is the clock frequency division factor; and one output signal, i.e., the clock enable *“ce”* signal, which generates a pulse when corresponding.

By combining both the input value and the clock frequency divider value of the spike generator block properly, the desired behavioral output response of the original TDE model can be achieved. Analytically, the EU-SSG's output firing rate for a given input value was obtained in the following:

$$f(d\_in) = \frac{F_{\text{clock}} * d\_in}{2^n(clk\_div + 1)} \tag{8}$$

where f is the spike generator's output firing rate (expressed in spikes per second), *d_in* is the input value to be converted to spikes, the constant F_{clock} is the system clock frequency (in Hz), n is data width, and *clk_div* is the internal clock frequency divider value.

Although (8) is almost identical to the one defined in [31], there is a difference in the component 2^n, due to the fact that, in the original model, the sign is taken into account (2^{n-1} is used instead), while, in (8), the unsigned version is used. However, the number of bits selected to represent the input value does not affect the maximum firing rate achievable by the EU-SSG block. *d_in* is the maximum value that can be represented with n bits, which is 2^n. In that case, if *d_in* is replaced in (8), the maximum output firing rate is expressed as follows:

$$f_{\text{max}} = \frac{F_{\text{clock}}}{(clk\_div + 1)} \tag{9}$$

where f_{max} is the spike generator's output maximum firing rate, the constant F_{clock} is the system clock frequency (in Hz), and *clk_div* is the internal clock frequency divider value. Therefore, the output firing rate only depends on both the system clock frequency and the clock frequency divider value. Since the maximum generating frequency is independent of the number of bits used to represent the values, this parameter can be set up according to the desired time resolution, thus benefiting the resource saving and allowing the implementation of a larger number of TDE units.

Two particular differences can be highlighted from this spike-generator block compared to the LIF neuron model. First, the EU-SSG block does not implement any refractory period. Therefore, the spikes can be produced through consecutive clock cycles. Instead, the clock frequency divider value needs to be adjusted in order to achieve the desired output spikes distribution. Second, the spike generator does not stop producing spikes. If the input value is higher than zero, the spike generator continues generating spikes. To stop the spike generation process, a clear input signal *“clear”* was added to the EU-SSG block, which resets its internal registers to zero. This control line is activated by the comparator *cmp_1* when the EPSC's timer output value reaches zero, meaning that the EPSC synaptic current is zero, and therefore, there is no current to integrate.

According to the architecture shown in Fig. 2, the EU-SSG input data (*“d_in”*) correspond to the TDE gain value. This signal depends on the current value of timer_0, which determines the detection time. The multiplication factor also controls the number of spikes to generate, as well as the last input value loaded into the generator. Thus, the gain block's output signal, and, therefore, the *“d_in”* signal, can be defined as follows:

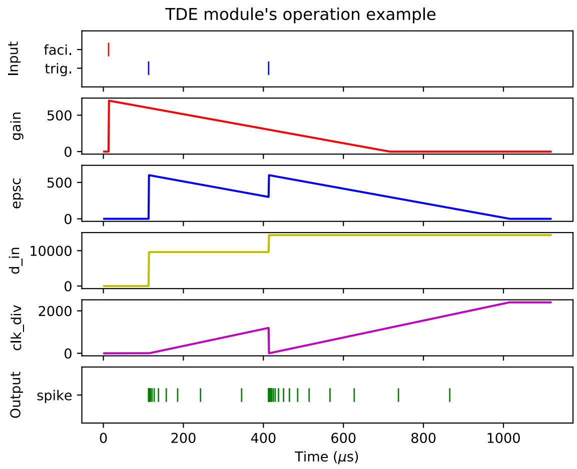


Fig. 6. Operation example of the proposed model. The detection time was set to 700 μs; the tau value was set to 0; the weight value was set to 4; and the decay value was set to 2. The main clock was set to 50 MHz, the time reference tick was set to 1 MHz, and the data width was set to 16. The Δt value between the facilitatory event and the first trigger event is 200 μs, while the Δt value between the facilitatory event and the second trigger event is 400 μs.

$$d\_in = \text{reg\_0} + \left(2^{\text{weight}} * \text{timer\_0}\right) \tag{10}$$

where *d_in* is the spike generator input value, reg_0 is the value stored in the register defined by (5), weight represents the timer_0 factor, and timer_0 is the timer value defined by (3).

In the same way, the EU-SSG clock divider value (*“clk_div”*) corresponds to the TDE EPSC value. It depends on the current value of timer_1, which, in turn, depends on timer_0 and *shift_0*, according to (6). In addition, the timer_1's value is weighted to control the output firing rate and the ISI variation. Therefore, the EPSC block's output value can be defined as follows:

$$clk\_div = \text{reg\_1} - \left(2^{\text{decay}} * \text{timer\_1}\right) \tag{11}$$

where *clk_div* is the spike generator input value, reg_1 is the value stored in the register defined by (7), decay represents the timer_1 factor, and timer_1 is the timer value defined by (6).

A behavioral example of the proposed model is shown in Fig. 6. The first row shows the input events differentiated by colors and following the color code used in Fig. 1: red color for the facilitatory event and blue for the trigger event. The second and third rows represent the evolution of the gain and EPSC timers over time, respectively. The fourth row illustrates the *“d_in”* signal value loaded into the spike generator block, and the fifth row represents the clock divider value of that block. Finally, the sixth row shows the output spikes produced in the presence of the shown input stimuli.

When an event is detected at the facilitatory input, the gain timer is initialized according to its detection time value. For each time reference tick, the gain timer decreases its value.

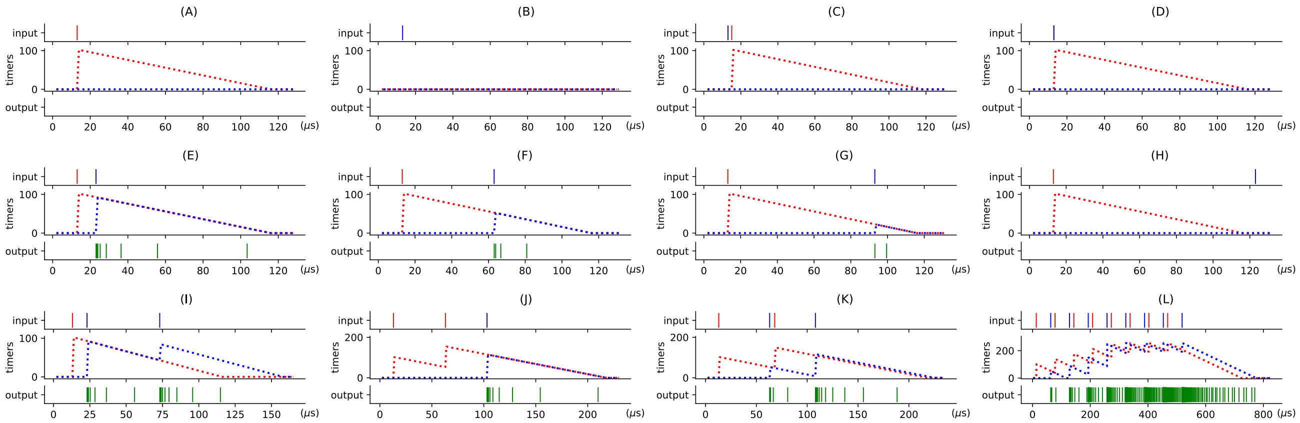


Fig. 7. RTL simulation for twelve basic cases of the TDE model with a time resolution of microseconds. Following the color code, red is related to the facilitatory, blue is related to the trigger, and green is related to the spike generator. (A) Single facilitatory spike. (B) Single trigger spike. (C) Single trigger spike before the single facilitatory spike. (D) Single facilitatory single trigger $\Delta t = 0$. (E) Single facilitatory single trigger short Δt. (F) Single facilitatory single trigger medium Δt. (G) Single facilitatory single trigger long Δt. (H) Single facilitatory single trigger very long Δt. (I) Single facilitatory multiple triggers. (J) Multiple facilitatory single triggers. (K) Multiple facilitatory multiple triggers. (L) Saturation.

As soon as an event is detected, the current value of the gain timer is loaded into the EPSC timer. Concurrently, that value is weighted and loaded into the spike generator as input data. Therefore, the spikes begin to be generated.

At this moment, the clock divider value of the spike generator is set to zero, thus producing the spikes at the maximal firing rate. For each time reference tick, the EPSC timer decreases its value, and the clock divider value of the spike-generator block is updated to a higher value. This causes a decrement in the output firing rate and, consequently, an increment of the ISI between two consecutive output events.

When another spike is detected at the trigger input, the current value of the facilitatory timer is added to the current value of the trigger timer. The input data of the spike generator block are updated, and the clock divider value is set to zero. This leads to a reset of the internal counter of the spike-generator block, which leads to an update of the ISI value according to the new input value. The TDE produces spikes until the EPSC timer reaches zero, when the internal stop signal is enabled. A more exhaustive behavioral analysis of the TDE response to both simple and complex stimuli is presented in Section IV.

IV. Analysis and Results

Three test scenarios were considered in order to validate and analyze the proposed model. First, a behavioral simulation was performed for the most common input stimulus combinations. The results were cross-validated with the results presented in the reference work [22]. Once validated, the model was analyzed and characterized by carrying out different experiments, where the ISI distribution and the number of output spikes were measured. Second, a single TDE unit was synthesized for an FPGA platform. The output spikes obtained from the FPGA were measured using an oscilloscope and recorded using a computer. Quantitative comparison was carried out between the simulation and the deployed version of the TDE unit. Third, a proof of concept of a sound source lateralization system was designed and tested using a population of TDE units.

A. Simulation

Since the TDE model has two inputs, many different input event combinations can occur. It is important to study each of these scenarios since they will directly affect the behavior of the model and its response. With the aim of verifying whether the behavior of the proposed model matches the expected output, 12 cases were simulated. For this experiment, a single TDE unit was instantiated, with 100 μs as detection time, a tau value of 0, a gain value of 5, a decay value of 1, and a value of 256 for both facilitatory and trigger saturation. In addition, the time resolution was set to microseconds. Fig. 7 presents the response of the TDE unit when being excited by twelve different sequences of input events.

Fig. 7(A) and (B) depicts simple examples where either a single facilitatory or trigger event is received by the TDE unit. No events were produced at the output. However, while the gain signal started being generated in Fig. 7(A), as a response to the facilitatory event, the trigger signal remained at zero in Fig. 7(B) since no facilitatory event was received before. Indeed, this effect can also be seen in Fig. 7(C), where the trigger event is received just a few microseconds before the facilitatory event. The same response of the model is obtained when both events arrive at the TDE unit at the same time, as shown in Fig. 7(D).

When a facilitatory event is presented at the TDE's input before the trigger event, the TDE's response is inversely proportional to the time difference (also called Δt) between them. Fig. 7(E)–(G) shows the output events generated by the TDE unit for short, medium, and long Δt values, respectively. As can be seen, the number of output events decreases with higher Δt values, while the ISI increases, matching the expected behavior. When Δt is higher than the detection time [Fig. 7(H)], the resulting response is the same as having Fig. 7(A) first and then Fig. 7(B), meaning that no events are generated at the output.

The proposed model was also simulated and evaluated in the presence of more realistic input patterns. In a real-world application, the input events are not received one by one. Instead, a continuous rate of events can be injected into the input. Fig. 7(I) and (J) shows the TDE model response when multiple facilitatory or trigger events are received at the input for a single opposite event.

On the one hand, Fig. 7(I) shows how the EPSC signal is incremented by a value proportional to the remaining time to zero of the gain signal with the arrival of the second trigger event, following (6). The first burst produced contains more events with lower ISI, whereas the opposite happens in the second burst. On the other hand, Fig. 7(J) shows how the gain signal is incremented by the detection time value when the second facilitatory event arrives according to (3). This alters the response of the simple case shown in Fig. 7(F), generating a burst with a higher number of events.

When many consecutive facilitatory-trigger pairs with Δt lower than the detection time parameter are received, the accumulated EPSC/gain signal values increase. Thus, after some time, the signals saturate according to the saturation parameter. At the saturation level, the TDE output firing rate is considered maximal, and its behavior can be estimated by using (9). We can affirm that the saturation parameter limits the TDE response, and its value would have to be set depending on the application and the related statistics of the input stimuli.

In this regard, the behavioral validation of the proposed TDE model shown in Fig. 7 has been proven to act in accordance with the reference model [22] in terms of both performance and requirements. Moreover, cases that were not evaluated in the original model were also reported in order to fully characterize the proposed model.

After the behavioral validation, more precise timing analysis of the TDE response was carried out. In [22], this study was performed by investigating the ISI distribution within a burst for six different Δt values. The ISI was calculated as $\text{ISI}_n = t_n - t_{n-1}$, where t_n is the timestamp of the n_{th} event. The authors reported that the obtained results matched the expected nonlinear response in the temporal evolution of the ISI within a burst.

The same test was done in order to verify that the nonlinear ISI variation feature was also achieved by the proposed model; 14 facilitatory-trigger pairs of events with different Δt values were used as input stimuli. Two TDE units were configured to work at different time scales by setting the time reference tick to microseconds and milliseconds, respectively. Fig. 8 presents the results obtained from simulations with a time reference tick in the scale of microseconds, instantiating a TDE unit with 700 μs as detection time, 0 as tau, 4 as weight, and 2 as decay. Similarly, Fig. 9 presents the results obtained from the simulations, although setting the time reference tick to milliseconds, instantiating a TDE unit with 70 ms as detection time, 0 as tau, 0 as weight, and 3 as decay.

Similar to the analog CMOS implementation, nonlinear profiles can be clearly observed in all the cases shown in both Figs. 8 and 9. These profiles were obtained by using exclusively linear operations and circuits, thus avoiding explicit circuitry for generating exponential behaviors. This feature allows reducing the needed resources and, therefore, increasing the total number of TDE units that can be instantiated into an FPGA or application-specific integrated circuit (ASIC). When the Δt value is almost equal to the configured detection time (e.g., above 550 μs for microseconds and 55 ms for milliseconds), the produced output events are not enough to represent the characteristic curve that cases with lower Δt presented. On the other hand, the first output event pairs seem to have the same ISI value for most Δt values (specifically for lower Δt values). These ISI values cannot be correctly appreciated in the plot since, according to the global clock, the precision of the minimum time difference is in the order of nanoseconds, and the Y-axis of the plot is represented in milliseconds.

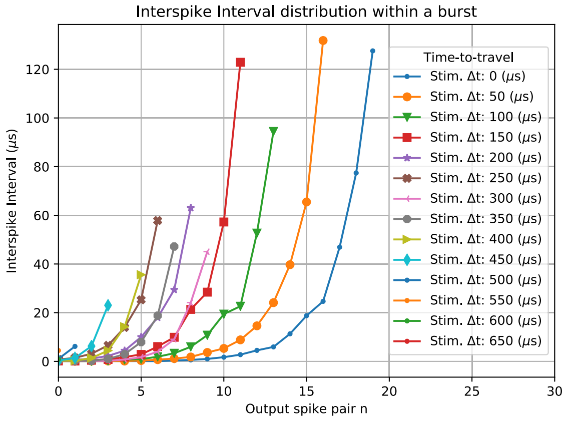


Fig. 8. TDE ISI response for a facilitatory-trigger pair with different Δt values for a microseconds resolution configuration. Note that the smallest Δt value used was not zero (no output events would be produced) but 20 ns (one clock cycle).

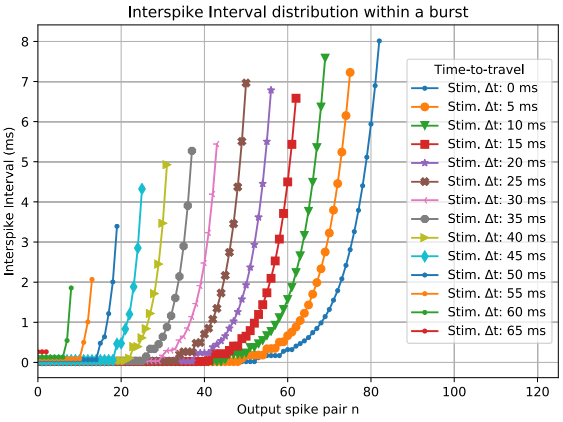


Fig. 9. TDE ISI response for a facilitatory-trigger pair with different Δt for a milliseconds' resolution configuration with a detection time of 70 ms.

The time scale set by the time reference tick affects not only the ISI curves, which has a better and more regular distribution for the millisecond time reference, but also the number of output events produced. This effect is caused by the combined use of both clock domains in the spike generator module, where the time reference clock is used to manage the inputs and the global clock is used to produce the events.

Another simulation was carried out in order to prove the variation in the number of output events generated by the TDE unit using both different time references and detection times. The results are depicted in Figs. 10 and 11 for microsecond and millisecond time references, respectively.

Note that the peak located at $\Delta t = 300$ μs (as shown in Fig. 10) is caused by the implementation of the spike

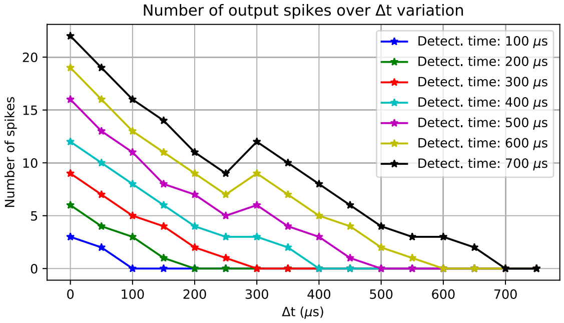


Fig. 10. Number of TDE output spikes over Δt variation for microsecond time reference tick.

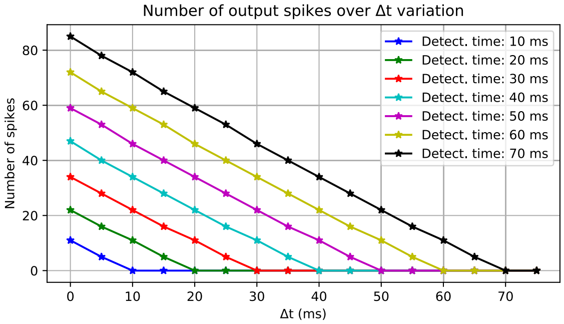


Fig. 11. Number of TDE output spikes over Δt variation for millisecond time reference tick.

TABLE I
Parameters of the TDE Population

Neuron ID	weight (μs/ms)	decay (μs/ms)	detection time (μs/ms)
TDE 0	9 / 4	2 / 1	100 / 10
TDE 1	6 / 2	1 / 2	300 / 30
TDE 2	5 / 1	2 / 3	500 / 50
TDE 3	3 / 0	2 / 12	700 / 60

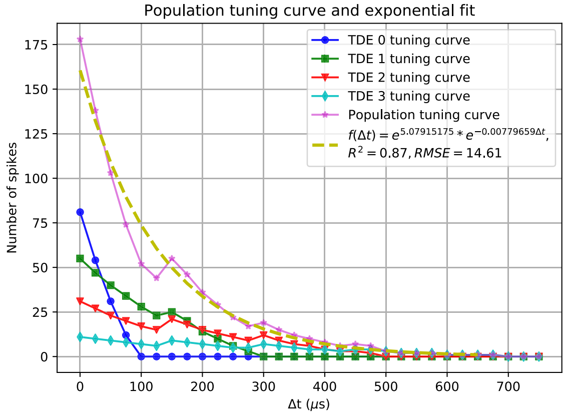


Fig. 12. Individual TDE tuning curves and population tuning curve for microseconds' time reference tick.

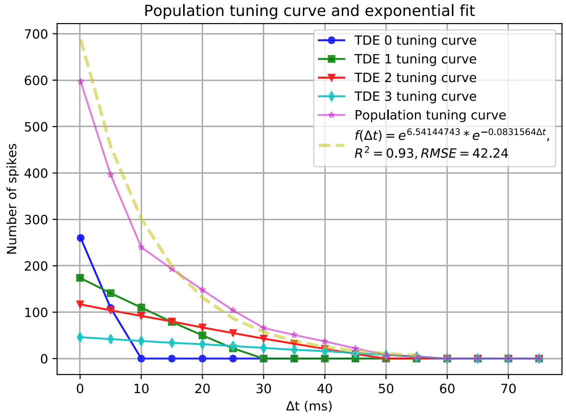


Fig. 13. Individual TDE tuning curves and population tuning curve for milliseconds' time reference tick.

generator module since the conversion from an integer value to a spike stream has an intrinsic error. This error is maximal at that time in this particular example, and it is deeply analyzed in [32]. Due to the timing resolution used in Fig. 11, even if the error exists, the peak cannot be appreciated.

The characteristic curve that relates the number of output events produced by the TDE with respect to the Δt value between its facilitatory and trigger inputs is known as the tuning curve. By varying the parameters' values of the TDE unit, its tuning curve can be adjusted. Therefore, it is possible to have a set of TDE units with different tuning curves.

This feature allows configuring a TDE population with different tuning curves responding to different input patterns or using the population response as a global response. Unlike the tuning curve test carried out in [22], in which all the neurons shared the same parameters, we conducted a similar test with different TDEs configurations.

Table I summarizes the values used for each TDE unit within the population created for this test. The population size is four units. All the units share the saturation value, set to 3000 for both the gain and EPSC signals, as well as the tau value, which was set to zero.

The time difference of the two input events was varied from 20 ns to 750 μs for the microsecond time reference, with a 20-μs step size. Similarly, the relative timing was varied from 20 ns to 75 ms for the millisecond time reference, with a 5-ms step size. The results obtained from the simulations are depicted in Figs. 12 and 13.

Both figures show the effect of the different tuning parameters in the output response of the TDE units. The TDE0's tuning curve presents a noticeable slope, meaning that it has a high output firing rate in a short time bin. On the contrary, the TDE1's tuning curve has an almost flat slope, which means that it produces fewer spikes but in a longer time period.

The tuning curves in Figs. 10 and 11 look practically linear. These tuning curves show a comparable behavior to the analog model, which is considered nonlinear for large temporal differences and linear for small time differences by Milde *et al.* [22]. The global behavior of the population, calculated as the sum of the output events for each TDE unit for each Δt value, fits an exponential curve. For the microseconds' time reference, the obtained exponential fitting curve had $R^2 = 0.87$ with root-mean-square error

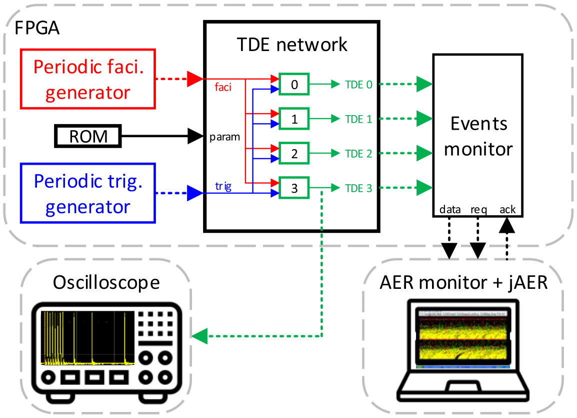


Fig. 14. Block diagram of the setup for real-time measurements' acquisition.

(RMSE) = 14.61. For the milliseconds' case, the R^2 value was 0.93 with RMSE = 42.24.

The spike generator intrinsic error directly affects the exponential approximation although the fitting curve can be considered acceptable taking into account that a nonlinear profile was obtained by using exclusively linear modules.

B. Field-Programmable Gate Array

After simulating the proposed model, analyzing and validating its behavior, a TDE population was deployed into an FPGA-based device in order to verify the results obtained in the simulation using a hardware platform. Fig. 14 depicts the setup used for this test. The upper part describes in detail the implemented design deployed into the FPGA. Two timers with periodic interruptions were used to generate both the facilitatory and trigger events. The time reference tick was set to microseconds, and the Δt value was fixed to 100 μs, having a wait time of 1 s between two consecutive stimulus generations. The population size was set to four in order to maintain the same architecture as in the simulation. Therefore, a read only memory (ROM) module was added for storing the population parameters, which were the same as those presented in Table I for the microseconds case. Finally, an events' monitor was connected to the population output to collect the events and send them to the computer by using an address-event representation (AER) protocol.

The lower part of Fig. 14 describes the two approaches used to measure the population response directly from the hardware. On the one hand, an oscilloscope was used to both measure and visualize the output events from the TDE3. Fig. 15 shows a screenshot with the captured events, where the increment of the ISI over time can be appreciated. On the other hand, a computer running jAER [33] was used to visualize and save the population output events in real time.

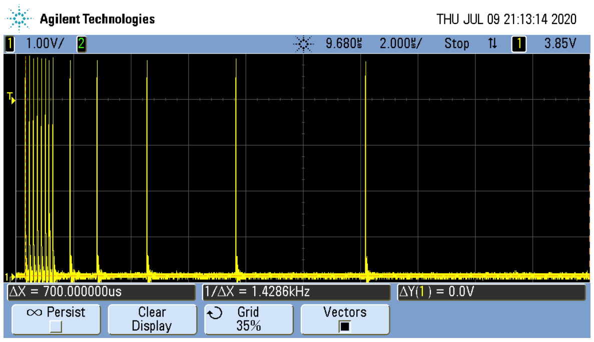

Agilent Technologies
THU JUL 09 21:13:14 2020
1 1.00V/ 2
9.680µs 2.000µs/ Stop 1 3.85V
ΔX = 700.000000us
1/ΔX = 1.4286kHz
ΔY(1) = 0.0V

Fig. 15. Output spikes captured by using an oscilloscope.

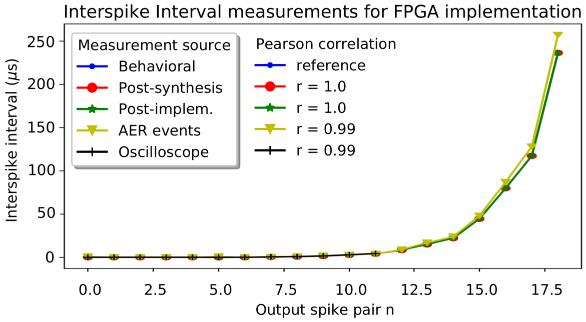


Fig. 16. Comparing the TDE response from different measurement sources.

TABLE II
HARDWARE RESOURCES UTILIZATION FOR DIFFERENT FPGA DEVICES

FPGA chip	Slice Registers Used / Available	LUT Slice Used / Available	Max. num. TDE units
XC6SLX150T (Spartan-6)	122/184304 (0.07%)	207/92152 (0.23%)	445
XC7A75T (Artix-7)	140/94400 (0.15%)	179/47200 (0.38%)	263
XC7K480T (Kintex-7)	140/597200 (0.02%)	179/298600 (0.06%)	1668

In addition to the behavioral simulation, a postsynthesis simulation and a postimplementation simulation were performed. The behavioral simulation was used as a reference and compared to the simulation results, as well as the measurements from the oscilloscope and the events collected by the events' monitor. The output of TDE3 was used to compute the Pearson correlation value [34] for a quantitative comparison.

The results are plotted in Fig. 16, showing a high correlation level (0.99 as the lowest value) and having the greatest differences in the later spike pairs. This could be caused by the inherent sampling error of the devices used. Nevertheless, the high correlation degree demonstrates that the TDE behavior does not change when it is deployed into an FPGA-based hardware in real time.

The resources needed by a single TDE unit were estimated for three different FPGA chips. In addition, the maximum number of TDE units that can be instantiated on each of them was reported. Table II summarizes all estimations. In addition, a high-level resource consumption comparison can be carried out between the analog implementation and the digital implementation. The former uses four capacitors for implementing the temporal decay of the signals, whereas the latter uses three timers instead. The difference lies in the absence of the refractory period in the proposed digital version.

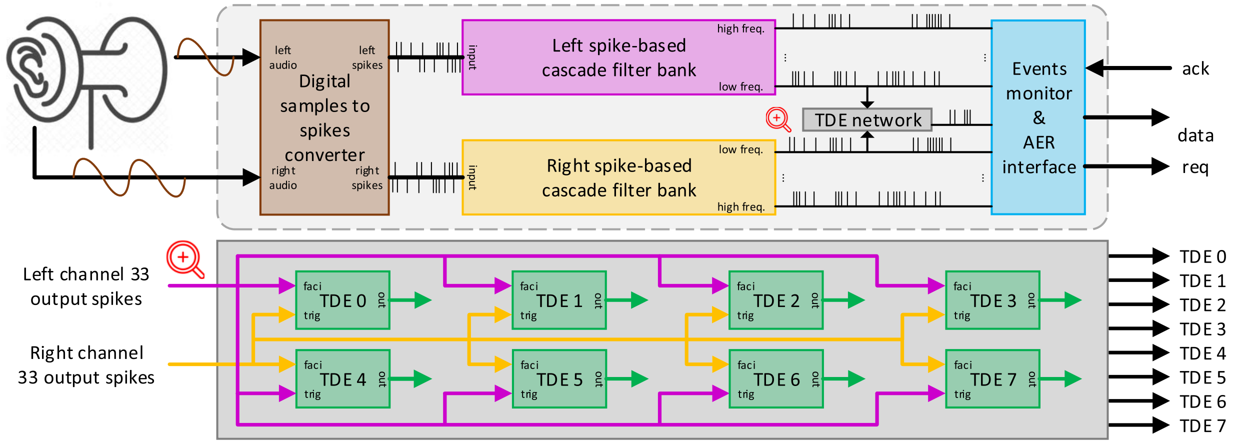


Fig. 17. Detailed block diagram of the FPGA top module for the proof of concept, containing both the NAS and the TDE populations.

Finally, a power consumption study was carried out for the XC6SLX150T chip, which was also used for all the measurements and real-time experiments in this work. First, a set of switching activity interchange format (SAIF) files was used for a realistic and precise estimation, where a single TDE unit was stimulated with a simple pair of facilitatory and trigger events. The reported power consumption was less than 1 mW with a static power consumption of 98% (intrinsic to the FPGA). Then, the system power consumption was measured directly from the real hardware setup by measuring the power consumption in two different cases: 1) when the board was programmed and the reset signal was active (553.6 mW) and 2) when the board was programmed, the reset signal was not active, and an input events pair was sent (555.1 mW). Therefore, the measured power consumption for a TDE unit was approximately 1.5 mW, having a deviation of 0.5 mW with respect to the simulation estimation.

For comparison purposes, the power consumption of a single analog CMOS TDE implementation in the XFAB XP018 technology was estimated by means of a circuit simulation. The static power consumption amounts to 1.4 nW, while the dynamic power consumption increases with the TDE's output spiking frequency, reaching approximately 500 μW at 500 Hz.

C. Real-Time Neuromorphic Application

The applicability of the proposed TDE model was evaluated by means of a proof-of-concept application. In the work presented in Milde *et al.* [22], the proof of concept was focused on a neuromorphic application using visual information generated by event-based cameras. In this work, a real-time sound source lateralization application for FPGA was implemented using the NAS.

Briefly defined, the sound source lateralization is considered as the capability to identify where the sound source is by using only binaural cues [35]. Different neuromorphic approaches to solve this task have been proposed in the last two decades [36]–[39]. Simplifying the concept, we will consider the sound source lateralization as the ability to determine whether the sound is on the left, on the middle, or on the right.

For this task, the same binaural cues that are commonly used for sound source localization [interaural time difference (ITD) and interaural level difference (ILD)] can be used. A binaural sensor is needed in order to be able to capture those cues. The NAS, as proposed in [26], is a neuromorphic sensor capable of decomposing the input sound from a pair of microphones into its frequency components, emulating the human cochlea. The general NAS architecture is depicted in Fig. 17 (top). Since it is an event-based sensor, its output is encoded as events; thus, the information is coded not only in the number of output spikes but also in the relative time between them. Only the ITD cue was used in the proof-of-concept application due to the timing nature given by the proposed TDE model.

The position of a sound source in space can be encoded by the temporal difference between the arrival of the sound waves at the ipsilateral side and the contralateral side. This time difference is known as the ITD. According to the specifications of the TDE model, an output response is exclusively produced if the incoming facilitatory event arrives before the incoming trigger event. Thus, two TDE populations were needed to perform the sound source lateralization task: one for detecting when the sound source is located at the left of the reference (the microphones pair) and one for detecting when the sound source is located at the right. Although the auditory information used was the same, thus containing the same temporal information, it projects onto the two TDE populations in an opposite way. The network architecture is shown in Fig. 17 (bottom), and the parameters values used for the TDE units' configuration were the same as those presented in Table I for the microseconds' time reference. Therefore, the individual tuning curves, as well as the population tuning curve, correspond to the plot shown in Fig. 12.

The test scenario was designed as follows: first, a virtual room of 10 m × 10 m × 2 m was created, using the room impulse response (RIR) generator [40] software tool. A pair of directional microphones were placed in the center of the room, imitating the human's ears disposition in a regular head, and at a height of 1 m over the floor. Then, three sound sources were placed at -90, 0, and $+90°$ with respect to the microphones pair, corresponding to the left, front, and right

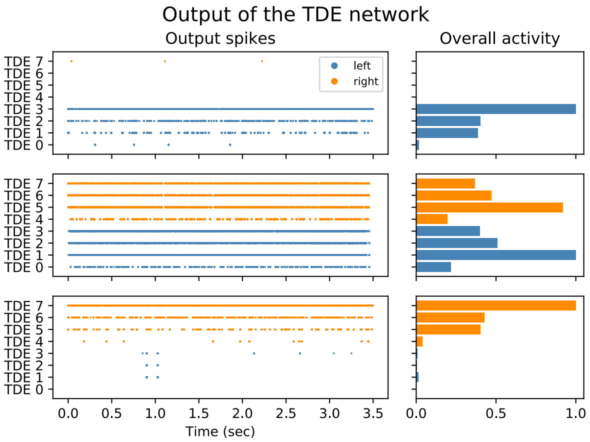


Fig. 18. Raster plot of the output events from the TDE population and normalized overall activity. The plots were generated using pyNAVIS tool [42]. TDEs 0—3 (left population) correspond to indices 0–3 in Table I, and TDEs 4–7 (right population) also correspond to indices 0–3 in the same table. Therefore, TDE 0 and TDE 4 use the same configuration and so on.

TABLE III

Comparison of a Single Neuron Implementations Using an FPGA. Zero Slice Registers Used Means That a Memory was Used to Store the Neuron State

Paper reference	Neuron model	Modeling strategy	Temporal decay type	FPGA resources (LUTs/reg)
(Perez, 2019) [43]	LIF	Biophysical emulation	Exponential	110 / 54
(Frenkel, 2017) [10]	LIF	Phenomenological	Linear	90 / 0
(Frenkel, 2017) [10]	Izhikevich	Phenomenological	Linear	247 / 0
(Soleimani, 2012) [44]	Izhikevich	Biophysical emulation	Exponential	493 / 617
(Levi, 2018) [45]	Hodgkin - Huxley	Biophysical emulation	Exponential	619 / 521
This work	TDE	Phenomenological	Linear	179 / 140

positions, respectively, with a separation of 2 m. The sound sources generated a pure tone beep of 500 Hz, with a duration of 0.5 s, every second.

Regarding the auditory sensor, a 64-frequency-channel, binaural NAS with a frequency range from 22 Hz to 22 kHz was generated using the OpenNAS tool [41]. The output events from frequency channel number 33 were used as input of the TDE population since the center frequency of its associated event-based bandpass filter was set to 502.38 Hz.

The sound was sent from the computer to the FPGA in real time, and the TDE population's output events were collected in a computer by using jAER. The results obtained from the real-time experiments are shown in Fig. 18. Top and bottom plots show the response of the TDE populations when the sound source was placed on the very left and very right positions (high ITD values), respectively. As expected, most of the activity was produced by the corresponding TDE population (left and right, respectively). The maximum overall activity was found in those TDE units with higher detection time (TDE 3 for the left-hand side and TDE 7 for the right-hand side). On the contrary, TDE 0 and 4 barely presented any activity due to their low value for the detection time.

The center plot shows the response of both populations when the sound source was placed in front of the microphones pair. In this case, all the instantiated TDE units produced output events as a response to the low ITD value inherent to the input stimuli. The small asymmetry in the overall response can be explained by the fact that the integrated interchip sound (I2S) protocol samples the input sound sequentially. Therefore, left and right samples provided to the left and right cochleas as input are slightly different, thus producing different spike activity at the filter's output. This activity could be postprocessed in order to extract more precise spatial information of the sound source.

V. Discussion and Conclusion

In this work, we presented a digital implementation of the TDE model. The full project code is open-source and publicly available on GitHub (https://github.com/dgutierrezATC/TDE_vhdl). This event-based design encodes temporal differences into a burst of events. A phenomenological design strategy was followed in order to implement the TDE behavior with reduced design complexity. The proposed implementation needs 179 lookup tables (LUTs) and 140 registers. The comparison in Table III shows that the phenomenological design approaches use fewer FPGA resources since the design complexity is reduced.

The simulation data presented here faithfully reproduce the behavior reported in [22], where also a phenomenological analog implementation approach was followed. Once simulated, the model was deployed into an FPGA-based device in order to characterize its response in a real-time platform. Although the proposed model is also suitable to be implemented in a full custom ASIC, an FPGA-based platform was considered due to its reconfigurability, fast, and affordable prototyping workflow. This allows creating custom TDE populations with different parameters and scaling up the population's size if needed, which is not possible in neither analog nor digital ASIC, unless specified at design time with high silicon overhead.

Another advantage of the FPGA TDE implementation is the wide temporal resolution range that it offers. It can be adjusted to different temporal domains by setting a few integer values depending on the specific applications, such as sound source localization (microseconds), vision (milliseconds), or odor localization (seconds). Since some neuromorphic systems, such as the SpiNNaker [12] board, only provide a milliseconds resolution by default, the digital TDE supports the application to tasks that require a higher temporal resolution, as the sound source localization example provided in this work. Furthermore, the FPGA implementation does not suffer from the mismatch problem common in analog CMOS circuits.

However, it is also important to mention that, unlike the analog implementation, the model is not fully asynchronous. This feature forces the model to have clock signals, thus increasing the static power consumption due to the switching currents. Therefore, the reported power consumption of the proposed model is significantly higher (less than 1 mW) compared to the few microWatts of power consumption for low output spike frequencies (measured in CADENCE using XFAB XP018 technology) given by the subthreshold operation level in which the analog version works.

In this work, a proof of concept of a sound source lateralization application was presented to evaluate the usability and performance of the proposed TDE model with realistic input stimuli. The use of the TDE model for sound source lateralization represents an efficient alternative to the biologically inspired Jeffress model [46]. The latter consists of an array of coincidence detector neurons with variable delay lines. The coincidence detector with a delay most similar to the signal's ITD indicates the sound source location with its dominant response. Since the TDE model encodes temporal differences in a frequency-coding manner, only two TDE's (one left and one right) are needed to encode the full range of sound source angles. Therefore, the hardware resources needed to implement this approach will be less compared to other state-of-the-art alternatives. Nevertheless, the combination of TDEs with different facilitatory time constants leads to an exponential population response, increasing the model's accuracy.

As demonstrated in Section IV-C, we can distinguish between at least three different sound source angles in the horizontal plane using the TDE's spiking frequency with four units for each side. Moreover, the output of the TDE population could be sent to other neuromorphic processors (e.g., Loihi [7] or SpiNNaker [12]) to further improve the localization accuracy by using an SNNs. This work proposes a novel FPGA implementation of the TDE, which, due to its modular structure, can be adopted by the neuromorphic research community and seamlessly integrated with event-driven sensors to support the investigation of novel algorithms for bioinspired sensing.

Acknowledgment

The authors would like to thank Moritz Milde, Philipp Klein, and Michele Mastella for their technical support. The authors would like to acknowledge the 2019 Capocaccia Neuromorphic Workshop.

References

[1] L. Alex, "The Von Neumann architecture topic paper #3," *Comput. Sci.*, vol. 319, pp. 360–8771, Jan. 2009.

[2] S. Wild, "Why is dual-pivot quicksort fast?" *CoRR*, vol. abs/1511.01138, Nov. 2015. [Online]. Available: http://arxiv.org/abs/1511.01138

[3] L. B. Kish, "End of Moore's law: Thermal (noise) death of integration in micro and nano electronics," *Phys. Lett. A*, vol. 305, nos. 3–4, pp. 144–149, Dec. 2002.

[4] J. Rigden, *Macmillan Encyclopedia of Physics* (MacMillan Science Library). 1996.

[5] D. Drubach, *The Brain Explained*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

[6] H. Moravec, "When will computer hardware match the human brain?" *J. Evol. Technol.*, vol. 1, no. 1, p. 10, 1998.

[7] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

[8] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers Neurosci.*, vol. 5, p. 73, May 2011.

[9] R. Tapiador-Morales, A. Linares-Barranco, A. Jimenez-Fernandez, and G. Jimenez-Moreno, "Neuromorphic LIF row-by-row multiconvolution processor for FPGA," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 1, pp. 159–169, Feb. 2019.

[10] C. Frenkel, J.-D. Legat, and D. Bol, "A compact phenomenological digital neuron implementing the 20 Izhikevich behaviors," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2017, pp. 1–4.

[11] S. Gomar and A. Ahmadi, "Digital multiplierless implementation of biological adaptive-exponential neuron model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1206–1219, Apr. 2014.

[12] S. B. Furber *et al.*, "Overview of the SpiNNaker system architecture," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2454–2467, Dec. 2013.

[13] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers Neurosci.*, vol. 5, no. 73, pp. 1–23, 2011.

[14] A. Linares-Barranco, G. Jimenez-Moreno, B. Linares-Barranco, and A. Civit-Balcells, "On algorithmic rate-coded AER generation," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 771–788, May 2006.

[15] E. T. Rolls and A. Treves, "The neuronal encoding of information in the brain," *Prog. Neurobiol.*, vol. 95, no. 3, pp. 448–490, Nov. 2011.

[16] J. P. Dominguez-Morales *et al.*, "Multilayer spiking neural network for audio samples classification using SpiNNaker," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2016, pp. 45–53.

[17] F. Theunissen and J. P. Miller, "Temporal encoding in nervous systems: A rigorous definition," *J. Comput. Neurosci.*, vol. 2, no. 2, pp. 149–162, Jun. 1995.

[18] N. K. Kasabov, *Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence*. Cham, Switzerland: Springer, 2019.

[19] Y. Yi *et al.*, "FPGA based spike-time dependent encoder and reservoir design in neuromorphic computing processors," *Microprocess. Microsyst.*, vol. 46, pp. 175–183, Oct. 2016.

[20] D. S. Reich, F. Mechler, K. P. Purpura, and J. D. Victor, "Interspike intervals, receptive fields, and information encoding in primary visual cortex," *J. Neurosci.*, vol. 20, no. 5, pp. 1964–1974, Mar. 2000.

[21] X. Lagorce and R. Benosman, "STICK: Spike time interval computational kernel, a framework for general purpose computation using neurons, precise timing, delays, and synchrony," *Neural Comput.*, vol. 27, no. 11, pp. 2261–2317, Nov. 2015.

[22] M. B. Milde, O. J. N. Bertrand, H. Ramachandran, M. Egelhaaf, and E. Chicca, "Spiking elementary motion detector in neuromorphic systems," *Neural Comput.*, vol. 30, no. 9, pp. 2384–2417, Sep. 2018.

[23] T. Schoepe, D. Gutierrez-Galan, J. P. Dominguez-Morales, A. Jimenez-Fernandez, A. Linares-Barranco, and E. Chicca, "Neuromorphic sensory integration for combining sound source localization and collision avoidance," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2019, pp. 1–4.

[24] P. Lichtsteiner *et al.*, "A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Jan. 2008.

[25] M. Yang, C.-H. Chien, T. Delbruck, and S.-C. Liu, "A 0.5 V 55 μW 64×2-channel binaural silicon cochlea for event-driven stereo-audio sensing," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 388–389.

[26] A. Jiménez-Fernández *et al.*, "A binaural neuromorphic auditory sensor for FPGA: A spike signal processing approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 804–818, Apr. 2017.

[27] S.-C. Liu, A. van Schaik, B. A. Mincti, and T. Delbruck, "Event-based 64-channel binaural silicon cochlea with Q enhancement mechanisms," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 2027–2030.

[28] V. Chan, S.-C. Liu, and A. van Schaik, "AER EAR: A matched silicon cochlea pair with address event representation interface," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 1, pp. 48–59, Jan. 2007.

[29] W. Gerstner, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[30] A. Ito, "Barrel shifter," U.S. Patent 4 829 460, May 9, 1989.

[31] A. Jimenez-Fernandez, A. Linares-Barranco, R. Paz-Vicente, G. Jimenez, and A. Civit, "Building blocks for spikes signals processing," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–8.

[32] F. Gomez-Rodriguez, R. Paz, L. Miro, A. Linares-Barranco, G. Jimenez, and A. Civit, "Two hardware implementations of the exhaustive synthetic AER generation method," in *Computational Intelligence and Bioinspired Systems*. Berlin, Germany: Springer, 2005.

[33] T. Delbruck. (2007). jAER open source project. Internet. [Online]. Available: http://jaer.wiki.sourceforge.net

[34] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*. Berlin, Germany: Springer, 2009, pp. 1–4.

[35] G. Plenge, "On the differences between localization and lateralization," *J. Acoust. Soc. Amer.*, vol. 56, no. 3, pp. 944–951, Sep. 1974.

[36] H. Finger and S.-C. Liu, “Estimating the location of a sound source with a spike-timing localization algorithm,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 2461–2464.

[37] P. K. J. Park *et al.*, “Fast neuromorphic sound localization for binaural hearing aids,” in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 5275–5278.

[38] A. van Schaik and S. Shamma, “A neuromorphic sound localizer for a smart MEMS system,” *Anal. Integr. Circuits Signal Process.*, vol. 39, no. 3, pp. 267–273, Jun. 2004.

[39] V. Y.-S. Chan, C. T. Jin, and A. V. Schaik, “Adaptive sound localization with a silicon cochlea pair,” *Frontiers Neurosci.*, vol. 4, p. 196, Nov. 2010.

[40] E. A. Habets, “Room impulse response generator,” Technische Univ. Eindhoven, Eindhoven, The Netherlands, Tech. Rep. 2.4, 2006, p. 1, vol. 2.

[41] D. Gutierrez-Galan, J. P. Dominguez-Morales, A. Jimenez-Fernandez, A. Linares-Barranco, and G. Jimenez-Moreno, “OpenNAS: Open source neuromorphic auditory sensor HDL code generator for FPGA implementations,” *Neurocomputing*, vol. 436, pp. 35–38, May 2021.

[42] J. P. Dominguez-Morales, D. Gutierrez-Galan, A. Rios-Navarro, L. Duran-Lopez, M. Dominguez-Morales, and A. Jimenez-Fernandez, “PyNAVIS: An open-source cross-platform software for spike-based neuromorphic audio information processing,” *Neurocomputing*, vol. 449, pp. 172–175, Aug. 2021.

[43] F. Perez-Peña, M. A. Cifredo-Chacon, and A. Quiros-Olozabal, “Digital neuromorphic real-time platform,” *Neurocomputing*, vol. 371, pp. 91–99, Jan. 2020.

[44] H. Soleimani, A. Ahmadi, and M. Bavandpour, “Biologically inspired spiking neurons: Piecewise linear models and digital implementation,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 12, pp. 2991–3004, Dec. 2012.

[45] T. Levi, F. Khoyratee, S. Saïghi, and Y. Ikeuchi, “Digital implementation of Hodgkin–Huxley neuron model for neurological diseases studies,” *Artif. Life Robot.*, vol. 23, no. 1, pp. 10–14, Mar. 2018.

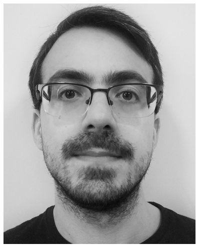
[46] L. A. Jeffress, “A place theory of sound localization,” *J. Comparative Physiol. Psychol.*, vol. 41, no. 1, p. 35, 1948.

Daniel Gutierrez-Galan (Member, IEEE) received the B.S. degree in computer engineering and the M.S. degree in computer engineering and networks from the University of Seville, Seville, Spain, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Architecture and Technology, with a research grant from the Spanish Ministry of Education and Science.

His research interests include embedded systems programming, digital design, field-programmable gate array (FPGA), spiking neural networks in embedded systems for audio processing, neuromorphic auditory sensors, and neuromorphic robots.

Thorben Schoepe (Member, IEEE) received the B.Sc. degree in biomimetics from Westphalian University of Applied Sciences, Gelsenkirchen, Germany, in 2015, and the M.Sc. degree in biomechatronics from Bielefeld University, Bielefeld, Germany, in 2018. He is currently pursuing the joint Ph.D. degree with the Neuromorphic Behaving Systems Group, Bielefeld University, and the Bio-Inspired Circuits and Systems Laboratory, University of Groningen, Groningen, The Netherlands.

His research interests include insect-inspired neurorobotics, navigation, analog design, neuromorphic sensors, and sensory processing.



Juan P. Dominguez-Morales received the B.S. degree in computer engineering, the M.S. degree in computer engineering and networks, and the Ph.D. degree in computer engineering (specializing in neuromorphic audio processing and spiking neural networks) from the University of Seville, Seville, Spain, in 2014, 2015, and 2018, respectively.

His Ph.D. was granted with a research grant from the Spanish Ministry of Education and Science. Since January 2019, he has been working as an Assistant Professor with the University of Seville. His research interests include neuromorphic engineering, spiking neural networks, audio processing, and deep learning.

Angel Jimenez-Fernandez received the B.S. degree in computer engineering, the M.S. degree in industrial computer science, and the Ph.D. degree in neuromorphic engineering from the University of Seville, Seville, Spain, in 2005, 2007, and 2010, respectively.

In October 2007, he became an Assistant Professor with the Department of Computer Architecture and Technology, University of Seville. His research interests include neuromorphic engineering applied to robotics, real-time spikes signal processing, neuromorphic sensors, field-programmable gate array (FPGA) digital design, embedded intelligent systems development, and biomedical robotics.

Elisabetta Chicca (Senior Member, IEEE) received the degree in physics from the University of Rome, Rome, Italy, in 1999, and the joint Ph.D. degree in natural sciences from ETH Zürich, Zürich, Switzerland, and in neuroscience from the Neuroscience Center Zürich (ZNZ), Zürich, in 2006.

She held a post-doctoral position at the Institute of Neuroinformatics, University of Zürich, Zürich, and ETH Zürich, where she continued working as a Research Group Leader from May 2010 to July 2011. She led the Neuromorphic Behaving Systems Laboratory, Bielefeld University, Bielefeld, Germany, from August 2011 to July 2020. She is the Chair of the Bio-Inspired Circuits and Systems (BICS) Laboratory, Faculty of Science and Engineering, University of Groningen, Groningen, The Netherlands. She has a long-standing experience with developing event-based neuromorphic systems and their application to biologically inspired computational models. Her research activities cover models of cortical circuits for brain-inspired computation, learning in spiking neural networks, bioinspired sensing, and motor control.

Alejandro Linares-Barranco (Senior Member, IEEE) received the B.S. degree in computer engineering, the M.S. degree in industrial computer science, and the Ph.D. degree in computer science from the University of Seville, Seville, Spain, in 1998, 2002, and 2003, respectively.

He has been the Head of the Department of Architecture and Technology of Computers and the Chair of the Neuromorphic Systems Group, Excellence Unit SCORE, University of Seville, since 2017 and 2020, respectively. He has been a member of the Robotics and Computer Technology Laboratory since 2001. His research interests include very large-scale integration (VLSI) digital design, embedded systems, neuromorphic computing for sensors, robotics, and deep learning.