

Interfacing PDM MEMS Microphones with PFM Spiking Systems: Application for Neuromorphic Auditory Sensors

Daniel Gutierrez-Galan^{1,2} · Antonio Rios-Navarro^{1,2} ·
Juan Pedro Dominguez-Morales^{1,2} · Lourdes Duran-Lopez^{1,2} ·
Gabriel Jimenez-Moreno^{1,2} · Angel Jimenez-Fernandez^{1,2}



Abstract

Neuromorphic computation processes sensors output in the spiking domain, which presents constraints in many cases when converting information to spikes, losing, as example, temporal accuracy. This paper presents a spike-based system to adapt audio information from low-power pulse-density modulation (PDM) microelectromechanical systems microphones into rate coded spike frequencies. These spikes could be directly used by the neuromorphic auditory sensor (NAS) for frequency decomposition in different bands, avoiding the analog or digital conversion to spike streams. This improves the time response of the NAS, allowing its use in more time restrictive applications. This adaptation was conducted in VHDL as an interface for PDM microphones, converting their pulses into temporal distributed spikes following a pulse-frequency modulation scheme with an accurate inter-spike-interval, known as PDM to spikes interface (PSI). We introduce a new architecture of spike-based band-pass filter to reject DC components and distribute spikes in time. This was tested in two scenarios, first as a stand-alone circuit for its characterization, and then integrated with a NAS for

Antonio Rios-Navarro, Juan Pedro Dominguez-Morales, Lourdes Duran-Lopez, Gabriel Jimenez-Moreno and Angel Jimenez-Fernandez contributed equally to this work.

✉ Daniel Gutierrez-Galan
dgutierrez5@us.es

Antonio Rios-Navarro
arios@us.es

Juan Pedro Dominguez-Morales
jpdominguez@us.es

Lourdes Duran-Lopez
lduran2@us.es

Gabriel Jimenez-Moreno
gaji@us.es

Angel Jimenez-Fernandez
angel@us.es

¹ Arquitectura y Tecnología de Computadores, Universidad de Sevilla Spain, Avd. Reina Mercedes s/n, Sevilla, Spain

² I3US, Score Lab, Avd. Reina Mercedes s/n, Sevilla, Spain

verification. The PSI achieves a total harmonic distortion of -46.18 dB and a signal-to-noise ratio of 63.47 dB, demands less than 1% of the resources of a Spartan-6 FPGA and its power consumption is around 7 mW.

Keywords Neuromorphic engineering · FPGA · Address-event · Pulse frequency modulation · Pulse density modulation · Neuromorphic auditory sensor

1 Introduction

Pulse-Density Modulation (PDM) is a sigma-delta modulation technique used to digitize an analog signal with a 1-bit data stream and a high sample rate. In recent years, many low-power microelectromechanical systems (MEMS) [1] microphones designed for mobile applications, such as tablets, laptops and cell phones, among others, have appeared in the market. In PDM data streams, a logic '1' corresponds to a pulse of the maximum positive polarity (+A), and a logic '0' represents the maximum negative polarity ($-A$). A signal value of 0 is codified by an alternation of '1' and '0'. Commonly, this type of modulation is associated with neuromorphic information codification, in the sense of being a rate-coded signal [2].

Neuromorphic computation allows processing Pulse-Frequency Modulation (PFM) information only when it is needed, avoiding periodic or redundant data processing, which is present in Pulse-Coded Modulation (PCM) signals, thus saving power and computational resources [3]. A similar approach can be found in [4], where the authors use the PDM microphones to encode the input sound directly into a spike train, thus having the information ready to be processed using Spiking Neural Networks (SNNs). In addition, a PDM coding strategy is used in [5] for the communication between neurons when sound information is used.

Currently, we can find diverse neuromorphic cochleae, both analog [6–10] and digital [11, 12], inspired by Lyon's cascade model [13] modeling the Inner Hair Cells (IHC). In [14], a Neuromorphic Auditory Sensor (NAS) is presented, based on on spike signal processing (SSP) techniques [15].

Figure 1 shows a global scheme of the NAS architecture. First, the audio information is provided by a digital audio codec, whose discrete audio samples output is converted into spike (1-clock cycle width pulses inside the FPGA) streams, following the Pulse-Frequency Modulation (PFM). The NAS filters these spikes directly, spike after spike, using a set of Spike-based Low-Pass Filter (SLPF) connected in a cascade fashion. Finally, spikes are transmitted to the next layers using the Address-Event Representation (AER) protocol.

NAS has been currently used for many practical applications, such as sound source localization [16], heart murmur diagnosis [17], and speech recognition [18, 19], among others. Great effort has been dedicated to improve NAS features, as it is the input layer of all these systems, improving responses and spreading for new applications of this technology.

The circuit proposed in this paper provides a novel interface with PDM microphones which output a rate-coded signal with a higher sampling rate than audio codecs (3.125 MHz in this case), and a time resolution of 320 ns, while audio codecs have a sampling period from $22.67 \mu\text{s}$ to $10.41 \mu\text{s}$, limiting the temporal capabilities for some applications, e.g., sound localization systems [20]. The output of the proposed circuit supplies a spike stream which can be directly processed by the NAS' filter banks [14], avoiding the use of discrete audio codecs. We also introduce a new Spike-based Band-Pass Filter (SBPF) for spikes distribution.

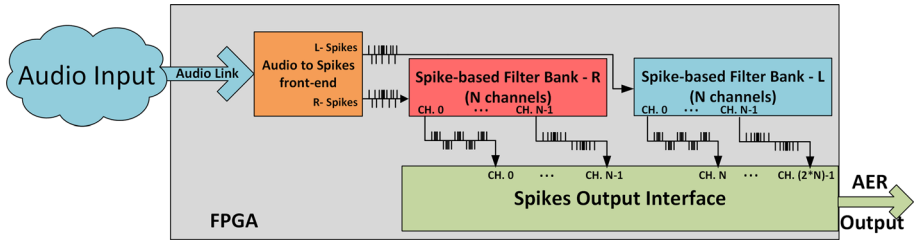


Fig. 1 NAS architecture: audio to spikes, spikes processing banks, and AER output interface

The main contributions of this paper include the following:

- A novel interface for PDM to rate-coded spikes conversion is presented.
- The proposed interface have been simulated in software and implemented in reconfigurable hardware.
- The hardware resources and power consumption needed have been analyzed.
- The proposed interface is capable of working in real time and could be used for a wide range of neuromorphic applications.

The rest of the paper is structured as follows. In Sect. 2 an interface to convert PDM into rate-coded spikes is presented, which is divided in two main blocks: the front-end circuit (Sect. 2.1) and a spike-based second-order band-pass filter (Sect. 2.2). The hardware resources and power consumption of the presented interface are analyzed in Sect. 2.3. In Sect. 3, the experimental results are presented, including hardware simulation (Sect. 3.1) and implementation (Sect. 3.2). Finally, in Sect. 4, the conclusions of this paper are presented.

2 PDM to Spikes Interface (PSI)

PDM information codification technique is substantially different from rate-coded spike-based signals. In rate-coded spike-based signals, the information is given by the spikes frequency, which means that the information is inversely proportional to the temporal Inter-Spike Interval (ISI). Spike-based systems use PFM to distribute the spikes in time properly, in order to accurately represent the signal's information. On the other hand, in PDM signals the information is contained in the density of pulses, and one pulse is generated every clock cycle, where a logic '1' represents a positive value, and a logic '0' a negative one. For example, when there are more '1's than '0's the information is positive, and the more '1's, the greater the amplitude. Thus, to reconstruct the signal's amplitude, it is necessary to collect PDM pulses during a temporal window, performing a downsampling operation, obtaining discrete samples with a fixed period. However, using PFM spike-based signals, with only two spikes, it is possible to reconstruct the amplitude of the original signal if the ISI is accurate enough, allowing it to process information spike-by-spike, and only performing computation when a spike is fired. Consequently, when there are no audio stimuli, spikes are not fired, and no computation is performed, thus saving power.

Digital systems convert PDM signals to digital values using the PCM. PCM is reconstructed from PDM with a digital decimation stage, commonly performing a downsampling by a factor of 64, and providing a multiple-bit word (e.g., 16 bits @ 48.8 kSamples/s) with high frequency noise added. After this stage, an infinite impulse response (IIR) filter is commonly

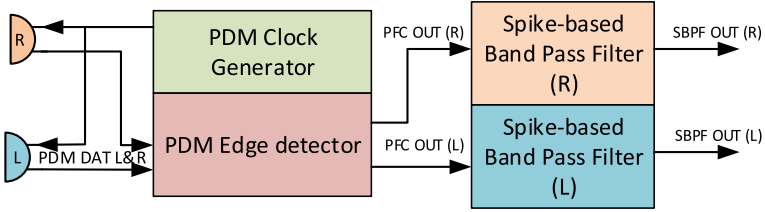


Fig. 2 PDM to spikes interface circuit

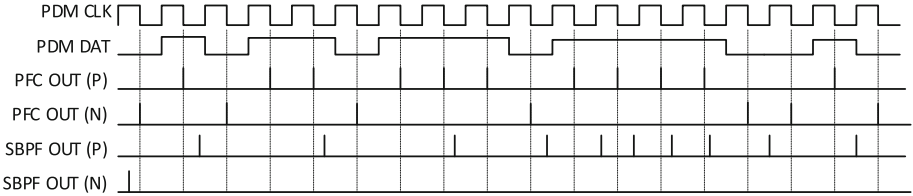


Fig. 3 Filtered spikes evolving from an increasing PDM audio signal

used as a band-pass filter (BPF) to remove DC component and high frequency quantization noise.

The main goal of PSI is to read PDM pulses and redistribute them in time as rate-coded spikes, with an ISI proportional to the sound pressure, performing the similar operations of digital systems but in the context of spike-based signals.

To convert PDM information into rate-coded spikes, a two stages circuit (see Fig. 2) is proposed. The first stage is a Finite State Machine (FSM) circuit for PDM MEMS microphones interfacing, that generates a spike of a single clock cycle for each PDM pulse, detailed in Sect. 2.1. The next stage consists of one (monaural) or two (binaural) banks of SBPF (depending on target application requirements), which process raw spikes from the previous stage, to give a temporal distributed spikes stream, detailed and analyzed in Sect. 2.2.

Since spikes can be both positive and negative, we use different wires to represent signed spikes. The FSM output generates a stream of signed spikes that are still not distributed in time, with the ISI being constant and equal to the PDM clock period. Figure 3 presents an example of a positive increasing audio signal, and how spikes evolve. PDM DATA represents the increasing input signal amplitude which is captured by PDM edge detector using the PDM CLK. This stage generates PFC OUT which are processed by SBPF using Eq. (1).

2.1 PDM Front-End Circuit

The PDM front-end circuit (PFC) has two main functionalities: to generate the PDM clock and to convert long PDM pulses into one clock cycle spikes (*PFC OUT (P)* and *PFC OUT (N)* in Fig. 3). The hardware platform used to implement these blocks is called AER-Node [21] and it has a base clock frequency of 50 MHz. Dividing this clock by a factor of 16, we get a PDM clock of 3.125 MHz, which is the maximum value commonly allowed by MEMS microphones. In every PDM clock cycle (*PDM CLK* signal in Fig. 3) there is a MEMS pulse in the *PDM DAT* signal. If *PDM DAT* has a value of '1' then a positive spike is transmitted to the next stage, and if there is a '0' it will be a negative spike (*PFC OUT (P)* and *PFC OUT (N)*, respectively, in Fig. 3).

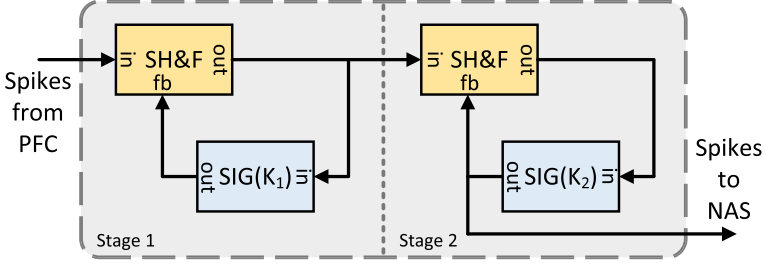


Fig. 4 Spike-based band-pass filter (SBPF) internal blocks with cascade topology

2.2 Second-Order Spikes Band-Pass Filter (SBPF)

The next stage is a Spike-based Band-Pass Filter (SBPF), which has been designed combining neurons for SSP. Two different kinds of neurons were used: Spikes Hold and Fire (SHF) and Spikes Integrate and Generate (SIG). SHF is used to subtract two input spike streams, generating a new stream of spikes with a frequency equivalent to the difference between input frequencies. SIG works like a spikes integrator. Internally, it consists of an up/down counter and a Synthetic Spikes Generator (SSG), with the output of the counter being the SSG's input. Input spikes increase or decrease the counter value according to their polarity, thus performing the integration operation. Finally, the SSG provides an output spiking rate proportional to the counter's value (i.e., integrated value). We can build frequency filters in the context of PFM spikes connecting these neuron models with different typologies. For example, from the point of view of ISI, a Spike-based High-Pass Filter (SHPF) only allows propagating spike rate changes, while a Spike-based Low-Pass Filter (SLPF) rejects high-frequency changes in spike rate.

Figure 4 shows the SBPF architecture, which is composed of two stages. Both stages are closed-loops with a SIG, although with a different SIG placement. These neurons are connected with 2-bit buses to propagate positive and negative spikes between them, in order to represent the bipolar nature of audio.

On the first stage, the SIG is placed in the feedback, while in the second stage it is placed in the main loop. Making an analogy to Laplace Transform, as was done in [14], with k_{SIG1} being the gain of the SIG in the first stage, and k_{SIG2} the one in the second stage, the full transfer function is presented in Eq. (1). It consists of one zero, which rejects the DC component, and two negative poles to filter quantification noise, implemented in the same way as for sampled digital systems, but in the context of PFM spike-coded signals.

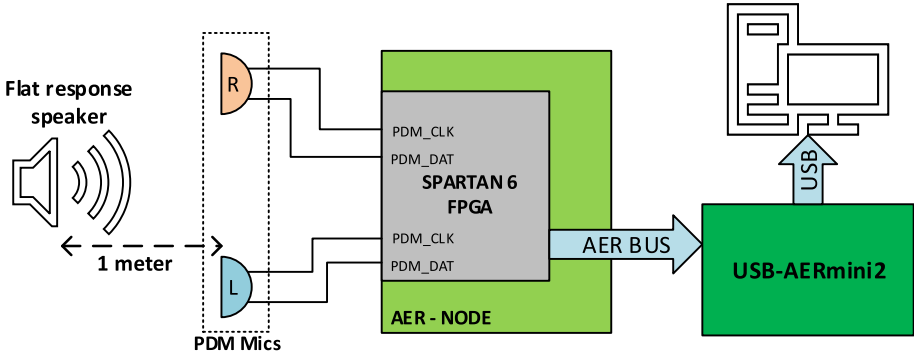
$$\begin{aligned}
 SBPF(s) &= \frac{1}{1 + \frac{k_{SIG1}}{s}} * \frac{\frac{k_{SIG2}}{s}}{\frac{k_{SIG1}}{s} + 1} \\
 &= \frac{k_{SIG2}s}{(s + k_{SIG1})(s + k_{SIG2})}
 \end{aligned} \tag{1}$$

2.3 Hardware Resources and Power Consumption

The PSI design was synthesized and implemented on a Xilinx Spartan 6 FPGA (XC6LX150T) to measure the required resources and its power consumption. Table 1 presents the resources that are needed to implement PSI in FPGA. As can be seen, the amount of resources needed

Table 1 PSI hardware requirements

Post-implementation results (Spartan 6-XC6SLX150T)		
Slices registers (%)	Slices LUT (%)	Max Clock Freq.
173 / 184.304 (0.094%)	409 / 92.152 (0.38%)	148.34 MHz

**Fig. 5** Test scenario. Sound is played by a response speaker, exciting the PDM microphones. Finally, the information is sent to a computer through an AER-to-USB interface

is under 0.1% of the total slice registers and logic (LUT) of the FPGA. The PSI can operate at a clock frequency of up to 148.34 MHz. To measure real power consumption, a 64-channel binaural NAS, was deployed on an AER-Node working at 50 MHz [21]. We made two power consumption measurements to isolate the PSI module consumption. The first is the power consumption of the entire system, and the second is the power consumption of the entire system at reset without the PSI module. The difference between both measurements, 7 mW, corresponds to the real dynamic power consumption of the PSI module. This power consumption is significantly lower than that of similar works [22], which consumes around 33.78 mW. The AER-Node platform with the 64-channel binaural NAS deployed on the FPGA consumes 1.33 W, to which the power consumption of the microphones used should be added.

3 Experimental Results

For testing purposes, a scenario was built to analyze the PSI's standalone behavior. Figure 5 presents the testing setup, where two PDM microphones from ST Microelectronics (MP34DT02) were connected to an AER-Node board, which was in turn connected to an USB-AERmini2 board. MP34DT02 are omnidirectional MEMS microphones with PDM interfaces, with an acoustic overload point of 120 dB_{SPL}, an SNR of 60 dBm, a dynamic range of 86 dB, and a maximum power consumption of 0.98 mW.

The AER-Node board holds the PSI, a 128-channel binaural NAS, and a set of AER interfaces. Its parallel AER output was connected to the USB-AERmini2 board [23], which works like a bridge between AER buses and USB ports, allowing the AER events to be sent from the AER-Node board to a host computer. In the computer, two software tools were running: jAER [24], to visualize and log AER information; and MATLAB, to analyze and

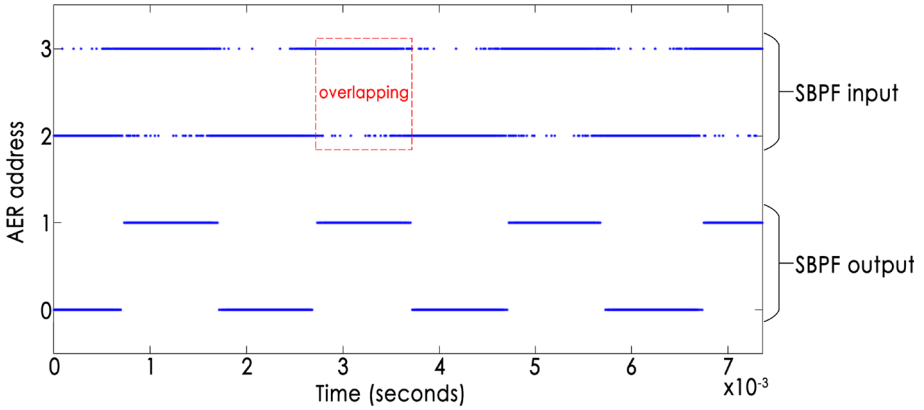


Fig. 6 Spikes from PSI: PDM front-end output (3-2) (top), and PSI’s output (1-0) after filtering (bottom). A section where spikes from the input of the SBPF overlap between the positive and the negative address is highlighted in red

process the events. The sound used to excite the system was played using a flat response audio speaker, in this case a BEHRITONE C5A from Behringer, placed at a 1-meter distance from the PDM microphones and at a fixed gain in order to have an audio level of 65 dB_{SPL} on the microphones’ side. This kind of equipment was used to avoid the influence of audio equalizers and the compensation of domestic Hi-Fi equipment. Thus, no preprocessed sounds were used and, instead, we tried to reproduce sound waves in the most ideal way possible. This will potentially open our system to many stand-alone applications, such as robotics.

3.1 PSI Experimental Results

For the first experiment, the system was stimulated with a clear 500 Hz pure tone audio signal played by the flat response speaker. Figure 6 represents the spikes from each stage of the PSI. Higher addresses (3 and 2) correspond to the spikes fired by the PDM front-end circuit, and lower addresses (1 and 0) to the SBPF output. Spike addresses 3 and 1 are positive, while 2 and 0 are negative.

Figure 6 depicts how the addresses that contain the output of the PDM front-end overlap the information between positive and negative, which does not happen after filtering it with the PSI. In PDM, information makes sense for the average activity of a temporal window. However, in the spikes domain, the information is coded with the time between two consecutive spikes. From the signal sign point of view, we can say that zero-crossing is performed when the polarity of the spikes changes (i.e., after a positive spike, a negative one is produced). In the case of the PDM front-end output, there are several spikes overlapping positive (address 3) and negative activity (address 2). From the point of view of ISI, this represents a considerable amount of high-frequency noise. However, if we check the SBPF output spikes, there is no overlapping between positive (address 1) and negative (address 0) activity, rejecting high frequency noise.

Figure 7 shows the reconstruction of the original signal using the spikes’ ISI. First, the green signal represents the reconstruction from PDM front-end’s output. This is a noisy signal with a measured Total Harmonic Distortion (THD) of -38.11 dB and a Signal-to-Noise Ratio (SNR) of 51.31 dB, with an offset of -487 kSpikes/s introduced by the PDM microphones.

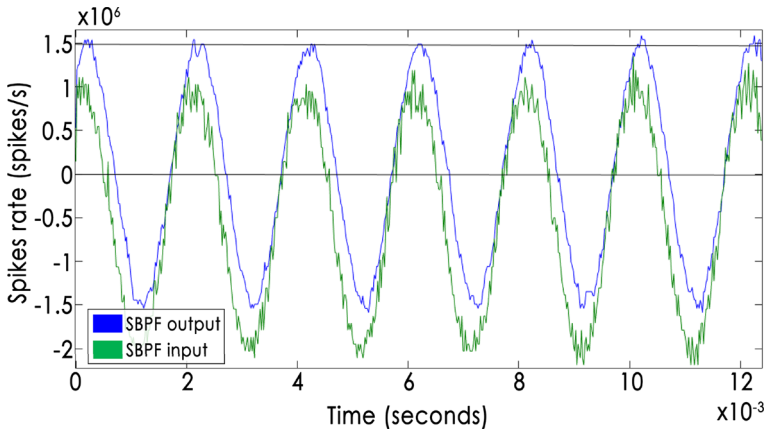


Fig. 7 Temporal reconstruction of a 500 Hz tone. Green: PDM front-end’s output. Blue: SBPF’s output

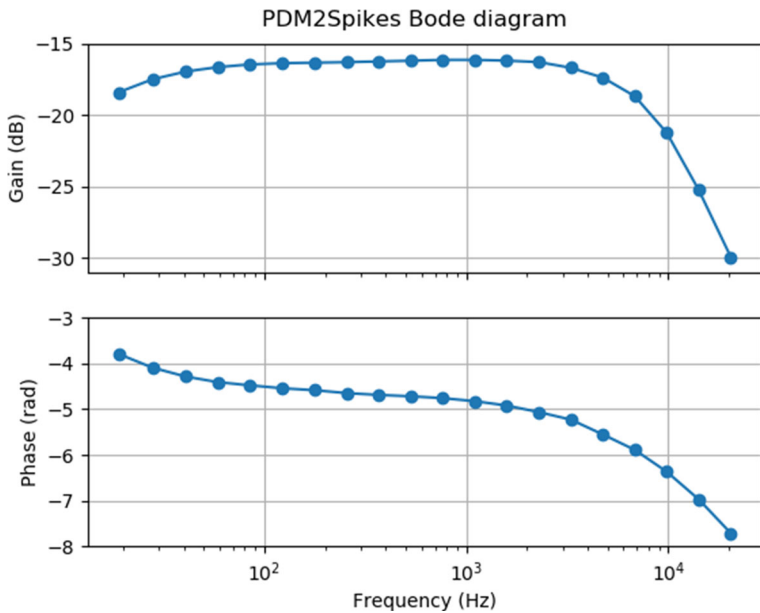


Fig. 8 Bode plots of the PDM2Spikes module

On the other hand, the blue signal is the reconstruction from SBPF’s output. A tone with less noise and offset can be seen, improving the previous audio signal quality. Analyzing this response, we achieve a THD of -46.18 dB and a SNR of 63.47 dB.

To measure the number of zero-crossings, a one second recording was analyzed and the amount of changes from positive spike to negative and vice versa were counted. In the PDM front-end’s output, more than 80k zero-crossings were found. However, in SBPF’s output, 1k zero-crossings were found, which exactly matches a 500 Hz signal.

Our second experiment consisted in a frequency sweep from 20 Hz to 20 kHz to analyze the behavior of the system with different frequencies. For this experiment we have generated

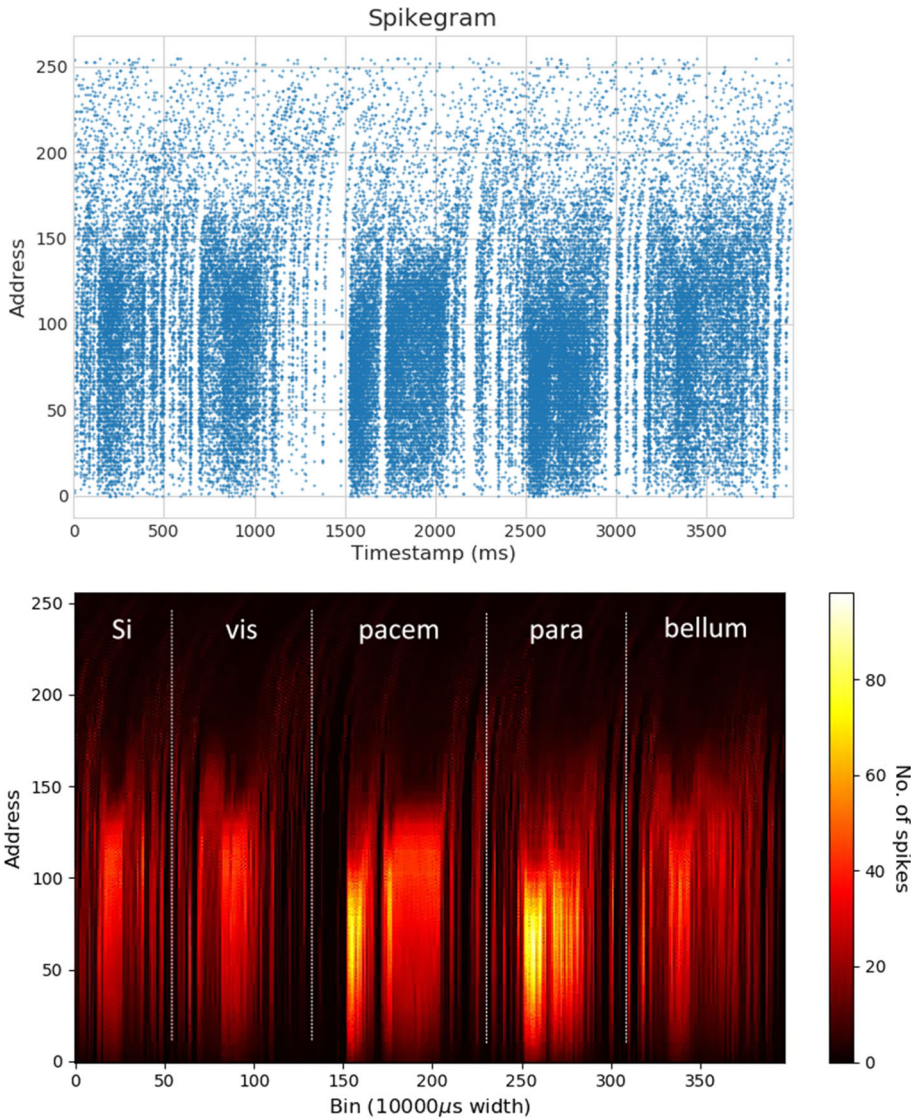


Fig. 9 Spikegram (top) and sonogram (bottom) obtained with NAVIS from a speaker saying “Si vis pacem para bellum”

a set of sinusoidal signals with 1V peak-to-peak amplitude using a Realtek ALC1150 audio codec in a computer. ALC1150 has an SNR of 115 dB and a THD of -88 dB. Figure 8 shows the frequency sweep results as a bode plot. The top curve in Fig. 8 presents the gain for diverse frequencies. PSI gain starts to increase from 70 Hz to 12 kHz, and then decreases, rejecting higher frequencies. This bandwidth is enough for many applications related to speech and speakers recognition. The spike-based filters in the PSI introduce a temporal deviation. It was measured as signal phase (in rads) and the results are included in Fig. 8 bottom. PSI has

a mean phase of -4.5 rads, approximately, increasing when frequency is close to the cut-off frequency, as expected from a low-pass filter.

3.2 NAS Integration

In order to validate the PSI on a real scenario, it was integrated in a 128-channel binaural NAS.¹ This NAS was fed with a male voice saying: “Si vis pacem, para bellum”, and the output activity was recorded using an USB-AERMini2 board as an AER-DATA file. Figure 9 contains the spikegram and the sonogram of this recording, respectively. Each word is clearly distinguishable, and activates middle channels between 200 Hz and 5 kHz. These figures were obtained by using NAVIS software [25].

4 Conclusions

In this paper, a PDM to PFM Spikes circuit is presented. PDM MEMS microphones are useful for low-power, stand-alone, embedded applications. Their output is based on spike density, and it needs to be adapted in order to be used as input to the NAS. A two-stage circuit for FPGA was designed, which is able to convert PDM information to PFM spikes with a consistent ISI. The PSI was synthesized for a Spartan 6 FPGA with low resources and power requirements. It was then tested with real audio stimuli, analyzing its behavior in terms of temporal response and zero-crossings. The PSI was also integrated in a full NAS to demonstrate the viability of the combination of this kind of systems. The use of PDM microphones with NAS considerably simplifies the system, enabling the development of compact and portable spike-based auditory systems with lower power consumption. However, this approach could have some limitations in very specific applications. For instance, the use of PDM microphones limits the preprocessing steps before the digitalization of the sound, like the automatic gain control, amplification, and other analog-based processing, due to the direct conversion from sound to digital pulses. Future work aims at developing adaptive capabilities in real time, as human ear does. Current research is focused on implementing a local automatic gain control [10] for input sound, and SBPF parameters (central pass frequency and Q factor) according to sound features performing an adaptation task in real time.

Acknowledgements The authors would like to thank the Spanish Agencia Estatal de Investigación (AEI) for supporting this work and Prof. Alejandro Linares-Barranco for his active participation and expert suggestions. The work of Daniel Gutierrez-Galan was supported by a Formación de Personal Investigador Scholarship from the Spanish Ministry of Education, Culture and Sport.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work was supported by the Spanish Agencia Estatal de Investigación (AEI) project MINDROB and by an H2020 European Regional Development Fund project CHIST-ERA SMALL (Grant Numbers PID2019- 105556GB-C33/ AEI/10.13039/501100011033 and PCI2019-111841-2)

Code availability The code related to this work is included in the OpenNAS project. See the OpenNAS GitHub project (<https://github.com/RTC-research-group/OpenNAS>) for more information.

¹ This NAS was generated by using OpenNAS tool. Source code available in <https://github.com/RTC-research-group/OpenNAS>.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Maluf N, Williams K (2004) An introduction to microelectromechanical systems engineering. Artech House, Norwood
2. Smith LS (2010) Neuromorphic systems: past, present and future. *Brain Inspired Cogn Syst* 2008:167–182
3. Liu S, Rueckauer B, Ceolini E, Huber A, Delbruck T (2019) Event-driven sensing for efficient perception: vision and audition algorithms. *IEEE Signal Process Mag* 36(6):29–37. <https://doi.org/10.1109/MSP.2019.2928127>
4. Bensimon M, Greenberg S, Haiut M (2021) Using a low-power spiking continuous time neuron (SCTN) for sound signal processing. *Sensors* 21(4):1065
5. Sevuktekin NC, Varshney LR, Hanumolu PK, Singer AC (2019) Signal processing foundations for time-based signal representations: neurobiological parallels to engineered systems designed for energy efficiency or hardware simplicity. *IEEE Signal Process Mag* 36(6):38–50
6. Chan V, Liu S-C, van Schaik A (2007) AER EAR: a matched silicon cochlea pair with address event representation interface. *IEEE Trans Circuits Syst I Regul Pap* 54(1):48–59
7. Hamilton TJ, Jin C, Van Schaik A, Tapson J (2008) An active 2-D silicon cochlea. *IEEE Trans Biomed Circuits Syst* 2(1):30–43
8. Wen B, Boahen K (2009) A silicon cochlea with active coupling. *IEEE Trans Biomed Circuits Syst* 3(6):444–455
9. Liu S-C, Van Schaik A, Mincti BA, Delbruck T (2010) Event-based 64-channel binaural silicon cochlea with q enhancement mechanisms. In: *Proceedings of IEEE international symposium on circuits and systems*. IEEE, pp 2027–2030
10. Kiselev I, Gao C, Liu S-C (2022) Spiking cochlea with system-level local automatic gain control. *IEEE Trans Circuits Syst I* 69:2156–2166
11. Mugliette C, Grech I, Casha O, Gatt E, Micallef J (2011) FPGA active digital cochlea model. In: *18th IEEE international conference on electronics, circuits, and systems*. IEEE, pp 699–702
12. Thakur C.S, Hamilton T.J, Tapson J, van Schaik A, Lyon R.F (2014) FPGA Implementation of the CAR model of the cochlea. In: *IEEE international symposium on circuits and systems (ISCAS)*. IEEE, pp 1853–1856
13. Lyon R (1982) A computational model of filtering, detection, and compression in the cochlea. In: *IEEE international conference on acoustics, speech, and signal processing, ICASSP'82, vol 7*. IEEE, pp 1282–1285
14. Jiménez-Fernández A, Cerezuela-Escudero E, Miró-Amarante L, Domínguez-Morales MJ, de Asís Gómez-Rodríguez F, Linares-Barranco A, Jiménez-Moreno G (2016) A binaural neuromorphic auditory sensor for FPGA: a spike signal processing approach. *IEEE Trans Neural Netw Learn Syst* 28(4):804–818
15. Jimenez-Fernandez A, Linares-Barranco A, Paz-Vicente R, Jiménez G, Civit A (2010) Building blocks for spikes signals processing. In: *The international joint conference on neural networks (IJCNN)*. IEEE, pp 1–8
16. Schoepe T, Gutierrez-Galan D, Dominguez-Morales JP, Jimenez-Fernandez A, Linares-Barranco A, Chicca E (2019) Neuromorphic sensory integration for combining sound source localization and collision avoidance. In: *IEEE biomedical circuits and systems conference (BioCAS)*. IEEE, pp 1–4
17. Dominguez-Morales JP, Jimenez-Fernandez AF, Dominguez-Morales MJ, Jimenez-Moreno G (2017) Deep neural networks for the recognition and classification of heart murmurs using neuromorphic auditory sensors. *IEEE Trans Biomed Circuits Syst* 12(1):24–34

18. Dominguez-Morales JP, Liu Q, James R, Gutierrez-Galan D, Jimenez-Fernandez A, Davidson S, Furber S (2018) Deep spiking neural network model for time-variant signals classification: a real-time speech recognition approach. In: International joint conference on neural networks (IJCNN). IEEE, pp 1–8
19. Rasetto M, Dominguez-Morales JP, Jimenez-Fernandez A, Benosman R (2021) Event based time-vectors for auditory features extraction: a neuromorphic approach for low power audio recognition. arXiv preprint. [arXiv:2112.07011](https://arxiv.org/abs/2112.07011)
20. Indiveri G, Sandamirskaya Y (2019) The importance of space and time for signal processing in neuromorphic agents: the challenge of developing low-power, autonomous agents that interact with the environment. *IEEE Signal Process Mag* 36(6):16–28. <https://doi.org/10.1109/MSP.2019.2928376>
21. Iakymchuk T, Rosado A, Serrano-Gotarredona T, Linares-Barranco B, Jiménez-Fernandez A, Linares-Barranco A, Jiménez-Moreno G (2014) An AER handshake-less modular infrastructure PCB with x8 2.5 Gbps LVDS serial links. In: IEEE international symposium on circuits and systems (ISCAS). IEEE, pp 1556–1559
22. da Silva B, Segers L, Braeken A, Steenhaut K, Touhafi A (2018) A low-power FPGA-based architecture for microphone arrays in wireless sensor networks. In: Voros N, Huebner M, Keramidas G, Goehringer D, Antonopoulos C, Diniz PC (eds) *Applied reconfigurable computing. Architectures, tools, and applications*. Springer, Cham, pp 281–293
23. Berner R, Delbruck T, Civit-Balcells A, Linares-Barranco A (2007) A 5 Meps \$100 USB 2.0 address-event monitor-sequencer interface. In: IEEE international symposium on circuits and systems. IEEE, pp 2451–2454
24. Delbruck T (2008) Frame-free dynamic digital vision. In: *Proceedings of international symposium on secure-life electronics, advanced electronics for quality life and society*, Tokyo, pp 21–26
25. Dominguez-Morales JP, Jimenez-Fernandez A, Dominguez-Morales M, Jimenez-Moreno G (2017) NAVIS: Neuromorphic Auditory VISualizer tool. *Neurocomputing* 237:418–422