Interfacing PDM sensors with PFM spiking systems: application for Neuromorphic Auditory Sensors

A. Jimenez-Fernandez, J. P. Dominguez-Morales, D. Gutierrez-Galan, A. Rios-Navarro,
R. Tapiador-Morales, A. Linares-Barranco

Escuela Técnica Superior de Ingeniería Informática. University of Seville. Seville, SPAIN
ajimenez@atc.us.es http://www.rtc.us.es/

Abstract—In this paper we present a sub-system to convert audio information from low-power MEMS microphones with pulse density modulation (PDM) output into rate coded spike streams. These spikes represent the input signal of a Neuromorphic Auditory Sensor (NAS), which is implemented with Spike Signal Processing (SSP) building blocks. For this conversion, we have designed a HDL component for FPGA able to interface with PDM microphones and converts their pulses to temporal distributed spikes following a pulse frequency modulation (PFM) scheme with an accurate configurable Inter-Spike-Interval. The new FPGA component has been tested in two scenarios, first as a stand-alone circuit for its characterization, and then it has been integrated with a full NAS design to verify its behavior. This PDM interface demands less than 1% of a Spartan 6 FPGA resources and has a power consumption below 5mW.

I. Introduction

Pulse-density modulation (PDM) is a sigma-delta modulation technique used to digitize an analog signal with 1-bit data stream and high sample rate. In recent years, many low-power microelectromechanical (MEMS) microphones designed for mobile applications such as tablets, laptops and cell phones among others have appeared in the market, being in the state-of-art of digital microphones. 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 1s and 0s. Commonly, this kind of modulation is associated to neuromorphic information codification, in the sense of being o rate-coded signals. In [1] a Neuromorphic Auditory Sensor (NAS) is presented inspired in the Lyons cascade model [2] of the biological cochlea, based on spike signals processing (SSP) techniques [3], [4].

Other bio-inspired audio sensors, digital [5], [6] or analog [7], [8], [9], implement a cascade of filters to model the basilar membrane and convert filtered components to spikes, modeling the behavior of the inner-hair cells

In Fig. 1 the authors show a global scheme of the NAS architecture. First, audio information is provided by a digital audio codec; next, discrete audio samples are converted into spike streams, following the pulse frequency modulation (PFM). These spikes are filtered directly, spike-by-spike, using a set of Spikes Low-Pass Filters (SLPF) connected in cascade. Finally, spikes are communicated to the next layers using the Address-Event Representation (AER) [10].

NAS has been currently used for many practical applications performed in real-time, as pitch frequency detection [11],

musical tones identification [12], echolocation [13], heart murmurs diagnosis [14], and speech processing [15]. Many efforts are dedicated to improve NAS features, as it is the input layer of all these systems, looking for better responses and new applications of this technology.

One NAS disadvantage is the need for a discrete audio codec to capture analog audio, providing a set of discrete periodical samples that have to be converted in spikes. These samples have a relatively high time between them (from 22.67uSec. to 10.41uSec.) limiting the temporal capabilities, e.g. sound localization applications. However, with the use of PDM microphones, the NAS could be provided with a stream of rate-coded signals with higher sample rate (3.125MHz in this case) with a time resolution of 320nSec., and directly perform a pulse-by-pulse processing, avoiding the need to generate spike streams synthetically as is done in previous NAS implementations with AC97 codecs [1].

PDM information codification is substantially different from rate coded spikes. In the case of spikes, the information is given by the spikes frequency, which means that the information is inversely proportional to temporal Inter-Spike-Interval (ISI); so, with only two spikes, we are able to reconstruct the original signal amplitude. Spike-based systems try to have a good spike distribution in time to accurately represent the signals information. In PDM the information is contained in the density of pulses and there is a pulse every clock cycle. For example, when there are more 1s than 0s the information is positive, and the more 1s there are, the higher the signal amplitude will be. Hence, for reconstructing the signal amplitude, it is needed to recollect PDM pulses along a temporal window, performing a down sampling operation. The main objective of this work is to design a HDL circuit able to read PDM pulses and redistribute them in time as rate coded spikes, with an ISI proportional to the audio intensity. Fig. 2 shows briefly how signals evolve from PDM pulses to PFM spikes.

II. PDM to spikes interface (PSI)

Making an analogy of how digital systems converts PDM signals in digital samples using the pulse coded modulation (PCM), this paper presents an input stage module for the NAS to convert PDM pulses into rate coded spikes using SSP building blocks [3]. Digital systems reconstruct PCM information from PDM using a digital decimation stage, commonly
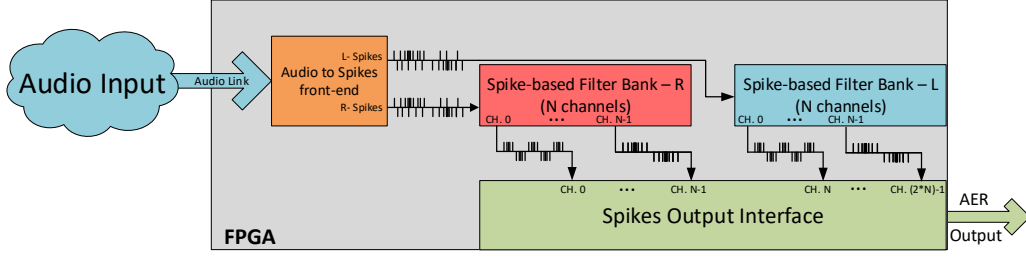
Fig. 1: Gobal NAS Architecture: Audio input to spikes, spikes processing banks, and AER output interface to higher level processing layers.

performing a down sampling by a factor of 64, and providing a multiple-bits word (e. g. 16 bits @ 48.8kSamples/sec) with high frequency noise added. After this stage, an infinite impulse response (IIR) filter is used as a band-pass filter (BPF) to remove DC components and high frequency quantization noise.

To convert PDM information into rate coded spikes a two stage circuit (Fig. 3) has been designed. The first stage consists of a finite state machine (FSM) that works as an edge detector, generating a single clock cycle spike for each PDM pulse. Clock frequency depends on specific NAS frequency (current desings varies from 27MHz to 50MHz).

As there could be both positive and negative spikes, we use two wires to represent them. Pikes. Pdm edge detector, generates a stream of signed spikes, where still these are coded in time, being the ISI connected positively to the audio signal rate. Fig. 2 presents an increasing positive PDM audio signal, and how spikes evolve.
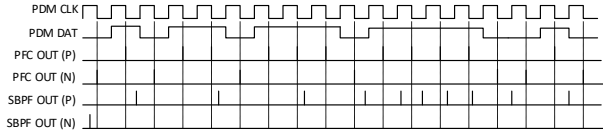

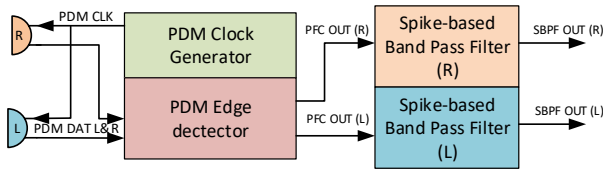Fig. 2: Filtered spikes evolving from an increasing PDM audio signal.

Fig. 3: PDM to spikes interface circuit.

A. 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. The hardware platform used for implementing this blocks is called AER-NODE [16] and it is clocked at 50MHz. Dividing this clock by a factor of 16 we get a PDM clock of 3.125MHz, which is the maximum value allowed by this kind of MEMS microphones. In every PDM clock cycle there is a PDM pulse in the PDM DAT line. 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 one.

B. Second order Spikes Band-Pass Filter (SBPF)

Next stage is a Spike Band-Pass Filter (SBPF), which can be found detailed in [17], including equations and parameters. This filter is composed by two first-order SLPF and one Spike Hold & Fire (SH&F), see Fig. 4. SH&F is a SSP building block that subtracts the spike rate between two spiking signals (detailed in [4]). The SLPF connected to the SH&F positive input has a higher cut-off frequency than the SLPF connected to the negative input. Subtracting both spike-based filters, only the information in the middle band remains, rejecting the DC and high frequency components. Signals between elements are buses with 2 bits width, as each bus has a dedicated line for positive spikes, and other one for negatives. These blocks use positive and negative activity to represent the bipolar nature of audio.
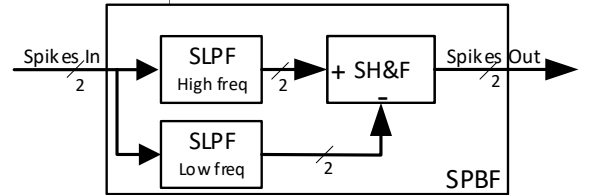

Fig. 4: Spike Band-Pass Filter (SBPF) internal blocks.

C. Hardware resources and power consumption

The has been synthesized and implemented on a Xilinx Spartan 6 FPGA (XC6LX150T) to quantify the number of demanded resources and the power consumption. Table I presents the resources that are needed for implementing PSI in a FPGA. As can be seen, the amount of resources needed is under 0.45% of total slices registers and logic (LUT). The PSI can operate up to a 147.18 MHz clock frequency but, in our case, we use a 50MHz clock. After synthesis, we have simulated the power consumption using Xilinx XPower tool, obtaining a consumption estimation of 2.67mW of the PSI. This power consumption should be added to the MEMS microphones power, which depends on the microphone selected. In our case, each microphone demands 0.98mW (according with manufacturer documentation), so the full system will demand 4.63mW.

TABLE I: PSI hardware requirements

Post-implementation results (Spartan 6 - XC6SLX150T)		
Slices Registers (%)	*Slices LUT (%)*	*Max Clock Freq.*
204 / 184.304 (0.11%)	409 / 92.152 (0.44%)	147.18 MHz

III. EXPERIMENTAL SETUP

For testing purposes, we have built a scenario for analyzing PSI standalone behavior. Later in this paper we will use the same scenario to test a full NAS implementation. Fig. 5 presents the testing setup, where we have connected two PDM microphones from ST Microelectronics (MP34DT02) to an AER-Node board, and this one to an USB-AERmini2 board. MP34DT02 are omnidirectional MEMS microphones with PDM interfaces, with an acoustic overload point of 120dB$_{SPL}$, a SNR of 60dBm, a dynamic range of 86dB, and a maximum power consumption of 1.1mW (as denoted before).
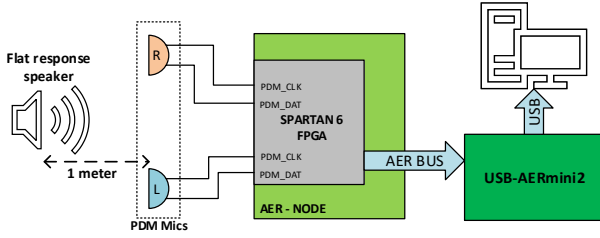

Fig. 5: Test scenario. Sound is played by a flat response speaker, exiting PDM microphones for NAS exciting. Finally information is sent to a computer through a specific AER-to-USB interface.

The AER-Node board has a Xilinx Spartan 6 FPGA (XC6S150T) and a set of AER interfaces. Next, we have connected the parallel output AER to an USB-AERmini2 board [18], which works as a bridge between AER buses and USB ports, being able to send the AER events from AER-Node board to a host computer. In the computer we run two software tools: jAER [19], to visualize and record AER information; and MATLAB, for events processing and analysis. The sound used to excite the system was played using a flat response audio monitor (BEHRITONE C5A), placed at a 1-meter distance from the PDM microphones and fixed volumes to have an audio level of 65dBSPL on the microphones side. We use this kind of equipment to avoid the influence of audio equalizers and compensation that domestic Hi-Fi equipment presents. In this way, we have no preprocessed sounds; instead, we try to reproduce the most ideal sound waves as possible.

A. Experimental results

As a first experiment we have stimulated the system with a clear 500Hz pure tone audio signal, played by the flat response speaker. Fig. 6 represents the spikes from each of the stages 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 spikes in the SPBF output.

Fig. 6 denotes how the addresses that contain the output of the PDM front-end block overlap the information, while this
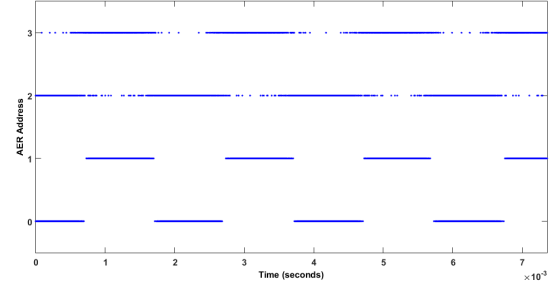

Fig. 6: Spikes from PSI: PDM front-end output (3-2) (top), and PSI output (1-0) after filtering (bottom).

does not happen after filtering it in the PSI, as can be seen in lower addresses of the figure. In PDM, information makes sense in the average activity of a temporal window. However, in the spikes domain the information should be in 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, for example, if 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 4) and negative activity (address 3). From the point of view of ISI this represents a considerable amount of high-frequency noise. If we check to SBPF output spikes, there is no overlapping between positive (address 1) and negative (address 0) activity, rejecting high frequency noise.
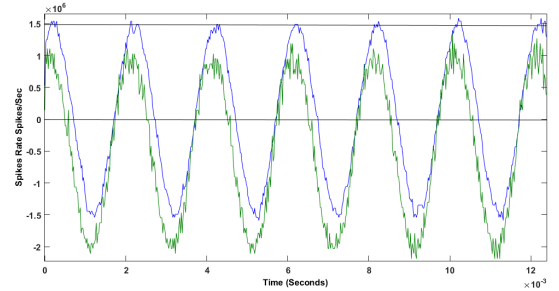

Fig. 7: Temporal reconstruction of a 500Hz tone. Green PDM front-end output, blue SBPF output.

Fig. 7 shows the reconstruction of the original signal from the spikes ISI. The green line represents the reconstruction from PDM front-end output, being a noisy signal and having an offset introduced by the PDM microphones. The blue line is the reconstruction from SBPFs output. A clear tone without noise and offset can be seen, improving the previous audio signal quality.

To measure the number of zero-crossings, we have taken a one-second recording and analyzed the number of changes from positive spike to negative one and vice versa can be found. In the PDM front-end output we found more than 80000 zero-crossings, however, in SBPF we found exactly 1000 zero-crossings, corresponding to a 500Hz signal.

B. NAS integration

In order to test it on a real scenario, the PSI has been integrated in a 128 channel NAS. The NAS has been excited

with a male voice saying: "Si vis pacem, para bellum" and the output activity has been recorded using an USB-AERMini2 board as an AER-DATA file. Fig. 8 contains the cochleogram and Fig. 9 the sonogram of this sentence, obtained thanks to NAVIS software [20]. Each word is clearly distinguishable, and activates middle channels between 200Hz and 5kHz.
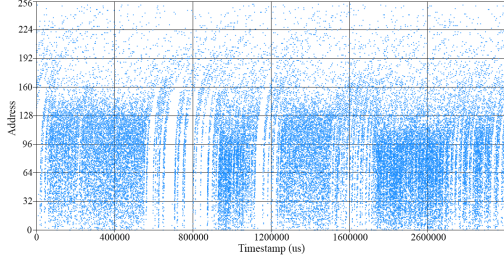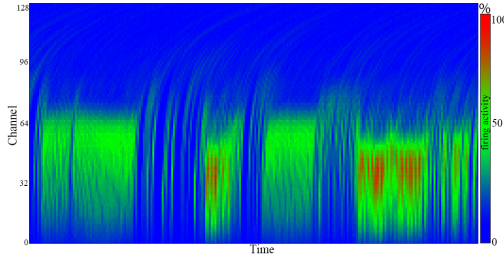

Fig. 8: NAVIS cochleogram: "Si vis pacem para bellum".

Fig. 9: NAVIS sonogram: "Si vis pacem para bellum".

IV. Conclusions

In this paper a PDM to PFM Spikes circuit has been presented. PDM MEMS microphones are perfect to be combined with SSP systems, as for example NAS. We have designed a two stage circuit for FPGA, which is able to convert PDM information to PFM spikes with a consistent ISI. PSI has been synthesized for a Spartan 6 FPGA with low resources and power requirements. PSI has been tested with real audio stimulus, analyzing its behavior in terms of temporal response and zero-crossings. PSI has been integrated in a NAS to demonstrate the viability of the combination of these kind of systems[1]. The results obtained with NAS are comparable to previous implementations of NAS with AC97 audio codecs. The use of PDM microphones with NAS simplifies it at system level, achieving a compact and portable auditory system with lower power consumption.

V. Acknowledgements

References

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

[2] R. Lyon, "A computational model of filtering, detection, and compression in the cochlea," in *ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 1982, vol. 7, pp. 1282–1285.

[3] A. Jimenez-Fernandez, et al., "Building blocks for spikes signals processing," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, 2010, pp. 1–8.

[4] A. Jimenez-Fernandez, et al., "A neuro-inspired spike-based pid motor controller for multi-motor robots with low cost fpgas," *Sensors*, vol. 12, no. 4, pp. 3831–3856, 2012.

[5] C. Mugliette, et al., "Fpga active digital cochlea model," in *2011 18th IEEE International Conference on Electronics, Circuits, and Systems*, Dec 2011, pp. 699–702.

[6] C. S. Thakur, et al., "Fpga implementation of the car model of the cochlea," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, June 2014, pp. 1853–1856.

[7] S. Liu, et al., "Event-based 64-channel binaural silicon cochlea with q enhancement mechanisms," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 2027–2030.

[8] B. Wen and K. Boahen, "A silicon cochlea with active coupling," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 3, no. 6, pp. 444–455, Dec 2009.

[9] T. J. Hamilton et al., "An active 2-d silicon cochlea," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, no. 1, pp. 30–43, March 2008.

[10] E. Cerezuela-Escudero, et al., "Spikes monitors for fpgas, an experimental comparative study," in *International Work-Conference on Artificial Neural Networks*. Springer, Berlin, Heidelberg, 2013, pp. 179–188.

[11] J. P. Dominguez-Morales, et al., "Multilayer spiking neural network for audio samples classification using spinnaker," in *International Conference on Artificial Neural Networks*. Springer, Cham, 2016, pp. 45–53.

[12] E. Cerezuela-Escudero, et al., "Sound recognition system using spiking and mlp neural networks," in *International Conference on Artificial Neural Networks*. Springer, Cham, 2016, pp. 363–371.

[13] E. Cerezuela Escudero, et al., "Real-time neuro-inspired sound source localization and tracking architecture applied to a robotic platform," *Neurocomputing*, vol. 283, pp. 129–139, 2018.

[14] J. P. Dominguez-Morales, et al., "Deep neural networks for the recognition and classification of heart murmurs using neuromorphic auditory sensors," *IEEE transactions on biomedical circuits and systems*, vol. 12, no. 1, pp. 24–34, 2018.

[15] J. P. Dominguez-Morales, et al., "Deep spiking neural network model for time-variant signals classification: a real-time speech recognition approach," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[16] T. Iakymchuk et al., "An aer handshake-less modular infrastructure pcb with x8 2.5gbps lvds serial links," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, June 2014, pp. 1556–1559.

[17] M. Domínguez-Morales, et al., "On the designing of spikes band-pass filters for fpga," in *International Conference on Artificial Neural Networks*. Springer, Berlin, Heidelberg, 2011, pp. 389–396.

[18] R. Berner, et al., "A 5 meps $100 usb2.0 address-event monitor-sequencer interface," in *2007 IEEE International Symposium on Circuits and Systems*, May 2007, pp. 2451–2454.

[19] T. Delbruck, "Frame-free dynamic digital vision," in *Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, 2008, pp. 21–26.

[20] J. P. Dominguez-Morales, et al., "NAVIS: Neuromorphic Auditory VISualizer Tool," *Neurocomputing*, vol. 237, pp. 418–422, 2017.

[1]https://github.com/RTC-research-group/OpenNAS