OpenNAS: Open Source Neuromorphic Auditory Sensor HDL code generator for FPGA implementations

D. Gutierrez-Galan [a,*], J.P. Dominguez-Morales [a], A. Jimenez-Fernandez [a,b], A. Linares-Barranco [a,b], G. Jimenez-Moreno [a,b]

[a] Robotics and Technology of Computers Lab, Universidad de Sevilla, Seville, Spain
[b] SCORE Lab - I3US, Universidad de Sevilla, Seville, Spain

Keywords:

Neuromorphic Auditory Sensor
AER
VHDL
Open hardware

A B S T R A C T

OpenNAS is an open-source tool for automatically generating the source files to create a Neuromorphic Auditory Sensor (NAS) VHDL project for FPGA. OpenNAS guides the user with a friendly interface that allows configuring the NAS' parameters using a five-step wizard for code generation. OpenNAS provides support to several audio input interfaces (AC'97 audio codec, I2S-ADC and PDM microphones), different processing architectures (cascade and parallel), and neuromorphic output interfaces (parallel AER, SpiNNaker). After NAS generation, users have everything ready for building, simulating, and synthesizing the VHDL project for a target FPGA. OpenNAS is fully modular, which allows providing support to new features in an easy way.

1. Introduction

Artificial cochleae are sensors inspired by the way that the biological inner ear works. Several models (hardware and software) can be found in the literature, with a similar architecture. They all have an input stage, where the input sound stimuli are collected; a processing stage consisting of a set of band-pass filters, commonly with a cascaded topology, and finally the output stage, where the filters' outputs are obtained and pre-processed for subsequent steps. Hardware implementations can be divided into two groups: reconfigurable and non-reconfigurable architectures. For the latter, the implementation is fixed (analog [1] or digital silicon [2]) and only few parameters can be tuned.

The main contribution of this paper is the introduction of the first software tool for automatically generating a full-custom Neuromorphic Auditory Sensor (NAS) for FPGA, which was presented in [3]. One of the main advantages of developing a NAS in an FPGA is its flexibility and versatility. However, these eventually become a serious disadvantage, as the complexity of design building and parameter tuning increases the difficulty of designing NAS. With the aim of distributing NAS along the neuromorphic research community, we present this tool, known as OpenNAS. OpenNAS builds a design by instantiating its different blocks and automatically computing all the parameters (including different filter gains, cut-off frequencies, FIFO memories and interfaces), guiding users step by step along the process.

NAS is currently used for several neuromorphic applications developed by different international research groups, demonstrating the utility of OpenNAS for setting up and integration in custom projects. This includes pattern recognition in audio samples [4] and sound source localization [5].

2. NAS architecture and design flow

The NAS architecture is mainly composed of three blocks, presented in Fig. 1. Firstly, the input audio signal is acquired by an audio front-end that converts audio information to pulse-frequency modulation (PFM) spike-coded signals. Next, the spikes generated by the first block excite a spike-based filter bank (SFB), which decomposes the information in different frequency bands using a cascade-fashion or a parallel-fashion processing architecture (user selectable). Finally, the spikes obtained from the output of the SFB are collected by a neuromorphic output interface to propagate the NAS information to any following processing layer. The current OpenNAS version supports a parallel AER monitor as output, as well as the SpiNNaker interface, which is used to connect the NAS to a SpiNNaker board [6].

Users should follow the design flow presented in Fig. 2, adjusting the settings of the three blocks to configure a new

* Corresponding author.
E-mail address: dgutierrez@atc.us.es (D. Gutierrez-Galan).

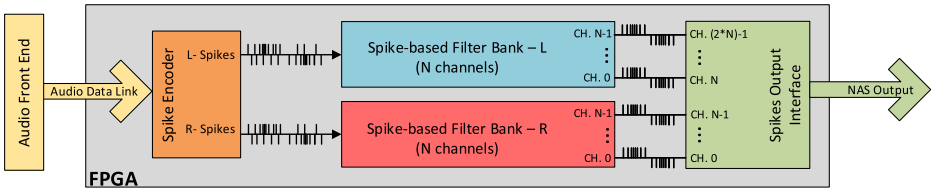

Fig. 1. Block diagram of the complete architecture of a binaural NAS.

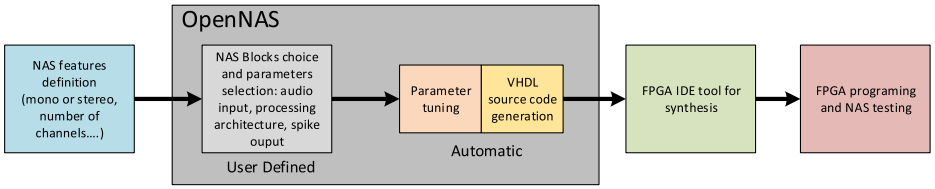

Fig. 2. Design flow diagram for full NAS synthesis.

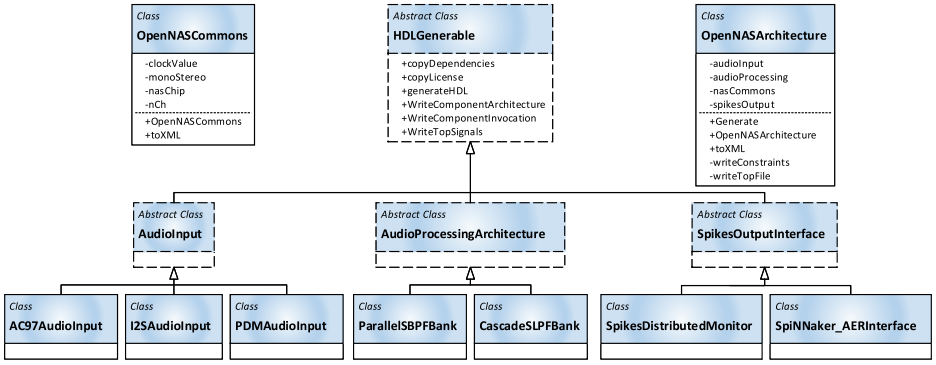

Fig. 3. OpenNAS class diagram.

NAS. After this step, NAS' parameters will be computed and the source code will be generated. Finally, using a development suite for FPGAs, such as Xilinx's Vivado or Altera's Quartus, the NAS can be synthesized and deployed into an FPGA. A detailed explanation of the concepts presented is available in the OpenNAS wiki[1].

3. Software architecture

To represent NAS' components, we used a set of classes, where each class contains all the parameters and HDL information of a specific component. The class hierarchy is presented in Fig. 3. The main NAS class is "OpenNasArchitecture", which contains an instance of common parameters (OpenNASComponents) and one attribute for each of the three NAS components: AudioInput, AudioProcesingArchitecture and SpikesOutputInterface (Fig. 3 mid). These are abstract classes that inherit from the "HDLGenerable" abstract class (Fig. 3 top), which contains the methods for generating the HDL code. Finally, specific component classes inherit from AudioInput, AudioProcesingArchitecture and SpikesOutputInterface, implementing each of the component features. Using this inheritance tree, a NAS is fully modeled and structured, ready for future expansions with new NAS features.

To guide the user, OpenNAS implements a wizard-based graphical user interface (GUI) written in Windows Presentation Foundation (WPF). The last step is HDL generation, which performs the following steps: (1) Each component writes its HDL dependency files and top entity to an output destination folder. (2) OpenNasArchitecture creates the top NAS HDL file. (3) Sequentially, each component writes I/O signals in the NAS top file. (4) Interface signals between components are added to the NAS top file. (5) Sequentially, each component writes its top component architecture. (6) To the NAS top file, each component adds an invocation to its instance, and these are connected to each other using interface signals. (7) A template for constraint files is generated

Table 1
OpenNAS performance comparison between two different processors.

NAS	SFB Error	AMD Ryzen 3900X (3.80 GHz)	Intel Core i7 6700HQ (2.60 GHz)
32ch. Stereo	0.48%	63.48 ms	139.14 ms
64ch. Mono	0.51%	64.36 ms	156.81 ms
128ch. Mono	0.53%	84.62 ms	247.4 ms
256ch.	0.55%	127.28 ms	313.38 ms

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

with all NAS I/O signals. (8) Finally, a NAS summary is written as a XML file.

4. OpenNAS execution results

To measure the tuning error and execution time, different NAS were generated with different CPUs. The results are presented in Table 1. The generated NAS have an average error of around 0.5%, which is lower than the error reported in [7] (around 1.573% for a 64-channel NAS). The time that the software takes to generate a NAS was measured, including internal parameter tuning, with two different processors: AMD Ryzen 3900X and Intel Core i7 6700HQ. For all the different cases, the generation time is below a few hundred milliseconds, increasing with the number of NAS' channels.

5. Conclusions

The main contribution of this work is a novel IP core generator tool that allows researchers to easily design their own NAS for specific applications. Thanks to its friendly interface and the automatic computation of its parameters by only following 5 steps in a GUI, the NAS architecture can be freely distributed to the neuromorphic community, ready for low-cost FPGAs[2]. OpenNAS[3] was designed following a hierarchical class structure to represent NAS' components, with its HDL description and parameters, allowing developers to increase OpenNAS components and functionalities easily.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Spanish grant (with support from the European Regional Development Fund) COFNET (TEC2016-77785-P). Daniel Gutierrez-Galan was supported by a Formación de Personal Investigador Scholarship from the Spanish Ministry of Education, Culture and Sport.

References

[1] M. Yang et al., A 0.5 V 55 μW 64 $\times$2 channel binaural silicon cochlea for event-driven stereo-audio sensing, IEEE J. Solid-State Circuits 51 (11) (2016) 2554–2569.
[2] Y. Xu et al., A FPGA implementation of the CAR-FAC cochlear model, Front. Neurosci. 12 (2018) 198.
[3] A. Jimenez-Fernandez et al., A binaural neuromorphic auditory sensor for fpga: a spike signal processing approach, IEEE Trans. Neural Netw. Learning Syst. 28 (4) (2017) 804–818.
[4] J.P. Dominguez-Morales et al., Deep neural networks for the recognition and classification of heart murmurs using neuromorphic auditory sensors, IEEE Trans. Biomed. Circuits Syst. 12 (1) (2017) 24–34.
[5] T. Schoepe et al., Neuromorphic sensory integration for combining sound source localization and collision avoidance, in: 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS), IEEE, 2019, pp. 1–4.
[6] E. Painkras et al., SpiNNaker: a 1-W 18-core system-on-chip for massively-parallel neural network simulation, IEEE J. Solid-State Circuits 48 (8) (2013) 1943–1953.
[7] A. Jimenez-Fernandez et al., Building blocks for spikes signals processing, in: Int. Joint Conf. on Neural Networks, IEEE, 2010, pp. 1–8.

[2] https://github.com/RTC-research-group/OpenNAS#supported-ides-simulators-and-devices
[3] https://github.com/RTC-research-group/OpenNAS#license

Daniel Gutierrez-Galan received the B.S. degree in computer engineering in 2014 and the M.S. degree in computer engineering and networks in 2016, both from the University of Seville, Sevilla, Spain. Since September 2017, he has been a Ph.D. student in the Department of Computer Architecture and Technology, at University of Seville. His research interests include embedded systems programming, digital design, FPGA, spiking neural networks in embedded systems for audio processing, neuromorphic auditory sensors and neuromorphic robots.

Juan P. Dominguez-Morales was born in Sevilla (Sevilla, Spain) in 1992. He 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, in 2014, 2015 and 2018, respectively. From October 2015 to December 2018, he was a PhD student in the Architecture and Technology of Computers Department of the University of Seville with a research grant from the Spanish Ministry of Education and Science. Since January 2019, he has been working as Assistant Professor in the same department. His research interests include medical image analysis, convolutional neural networks, computer-aided diagnosis systems, neuromorphic engineering, spiking neural networks, neuromorphic sensors and audio processing. In 2016 he became a member of the European Neural Network Society, he has been a member of IEEE for four years.

Angel F. Jimenez-Fernandez received the B.S. Degree in Computer Engineering in 2005, the M.S. Degree in Industrial Computer Science in 2007 and the Ph.D. in Neuromorphic Engineering in 2010 from the University of Seville, Sevilla, Spain. Since October 2007, he has been an Assistant Professor of Computer Architecture and Technology at the University of Seville. In April 2011, he was promoted to Associate Professor. His research interests include neuromorphic engineering applied to robotics, real-time spikes signal processing, neuromorphic sensors, field programmable gate array (FPGA) digital design, embedded systems development, high-speed serial communications, and smart sensors networking.

Alejandro Linares-Barranco (M'06) received the B.S. degree in computer engineering, the M.S. degree in industrial computer science, and the Ph.D. degree in computer science (specializing in computer interfaces for neuromorphic systems) from the University of Sevilla, Sevilla, Spain, in 1998, 2002, and 2003, respectively. From January 1998 to June 1998, he was Second Lieutenant in the Spanish Air Force working as System Administrator and Software Developer. From 1998 to 2000, he was a Member of the Technical Staff at the Sevilla Microelectronics Institute (IMSE-CNM-CSIC). From 2000 to 2001, he was a Development Engineer with the Research and Development Department, at ABENGOA Company, Sevilla, working on VHDL-based field programmable gate array (FPGA) systems for the INSONET European project on power line communications. From 2001 to 2006, he was an Assistant Professor at the Architecture and Technology of Computers Department of the University of Sevilla. In 2009, he was promoted to Associate Professor (civil-servant). Since 2013 he is the secretary of the department. In 2014 he has been visiting professor with the UZH. His Robotics and Computer Technology Laboratory has developed a set of AER-tools for debugging and connecting AER systems starting under the EU project CAVIAR. His research interests include VLSI for FPGA digital design, neuro-inspired chip-to-chip and chip-to-computer interfaces, spike based processing, event-based sensor information filtering and feature extraction, motor control, vision and audio for FPGAs, wireless sensor networks and embedded applications based on microcontrollers, bus emulation, and computer architectures. Prof. Linares-Barranco has been a Review Committee Member of ISCAS since 2009. In 2010, he became a member and officer of the Technical Committee on Neural Systems and Applications (NSATC) of the IEEE Circuits and Systems Society. In 2012 he became member of the Sensory Systems Technical Committee (SSTC).

Gabriel Jimenez-Moreno received the M.S. Degree in Physics (electronics) and the Ph.D. Degree from the University of Seville (Seville, Spain), in 1987 and 1992, respectively. After working with Alcatel, he was granted a Fellowship from the Spanish Science and Technology Commission (CICYT). Currently, he is an Associate Professor of computer architecture at the University of Seville. From 1996 until 1998, he was Vice-Dean of the E.T.S. Ingenieria Informatica, University of Seville. He participated in the creation of the Department of Computer Architecture (also at the University of Seville) and since 2013 has been its Director. He is the author of various papers and research reports on robotics, rehabilitation technology, and computer architecture. He has directed three national research projects on neuromorphic systems. His research interests include neural networks, vision processing systems, embedded systems, computer interfaces, and computer architectures.