ENABLING AFFORDABLE AND SCALABLE AUDIO SPATIALIZATION WITH MULTICHANNEL AUDIO EXPANSION BOARDS FOR FPGA

Maxime POPOFF (maxime.popoff@insa-lyon.fr) [1], **Romain MICHON**[1], and **Tanguy RISSET**[1]

[1]**INSA Lyon**, **Inria**, *CITI*, *UR3720*, 69621 Villeurbanne, France

ABSTRACT

The implementation of audio spatialization and active acoustic control techniques requires a large number of audio channels and significant computational resources, making these techniques costly and hardly accessible. The use of Field-Programmable Gate Arrays (FPGAs) offers a solution by enabling DSP programming within a centralized system with enough computational power and inputs/outputs to handle all audio streams. However, physically interfacing an embedded system with multiple audio channels is a complex process that poses a significant challenge to system accessibility. In this paper, we present two open-source circuit boards facilitating the interfacing between an FPGA and large multichannel systems, both aiming different kinds of approaches. The first board is designed to be cost-effective and easily reproducible. The second involves more complex manufacturing techniques, but it allows us to fully exploit the performances of the FPGA. Each board provides 32 audio channels and has stacking capabilities, allowing for the interfacing of up to 512 channels on a single FPGA. Their integration into the Syfala compilation flow facilitates the implementation of algorithms without hardware concerns. These interfaces aim to enhance accessibility and affordability when implementing audio spatialization and active acoustic control systems.

1. INTRODUCTION

In the fields of active acoustic control and audio spatialization, using a large number of speakers is usually necessary to obtain optimal results.

Some sound spatialization techniques such as Wave Field Synthesis (WFS) [1] can imply the use of thousands of speakers in specialized concert halls [2–4]. The usual approach in handling real-time audio signal processing systems with such requirements typically involves employing a CPU [1] alongside a multichannel audio interface. More recently, accelerators such as GPUs [2] that provide high parallel data processing, have proven to be very efficient with WFS [5] and Active Noise Control (ANC) [6] systems. However, both GPU and CPU implementations rely on external audio interfaces, which implies a significant latency due to buffering. Moreover, the availability of audio interfaces offering more than 32 audio outputs is limited and often comes at a high cost. [3] To allow for more channels, some applications such as Belloch et al. [5] use a combination of multiple audio interfaces, but this approach remains very expensive and depends on the CPU processing capabilities. The same limitations apply when using Ethernet audio interfaces connected in parallel [7]. Other specialized systems employ proprietary multi-channel signal processing systems which are often expensive and not flexible [8].

Field-Programmable Gate Arrays (FPGAs) are very powerful embedded platforms that have been increasingly used in recent years for real-time audio Digital Signal Processing (DSP) applications [9–11]. They provide unique performances in terms of audio latency, computational power, and interfacing.

The use of an FPGA can reduce system expenses by allowing all audio channels to be managed by a single embedded system and by providing enough resources to compute audio samples at a high sampling rate. The high degree of parallelization of FPGAs and their abundance of GPIO [4] pins enables the use of a large number of digital audio channels, making them especially interesting for sound spatialization and acoustic control techniques [12, 13].

However, in order to connect multiple audio inputs/outputs to an embedded system, it is necessary to perform analog-to-digital conversion of the signal with an audio codec, potentially amplify it, and wire each channel. These constraining steps make the practical real-time implementation of massive multichannel algorithms hard to reach. Most multichannel ANC practical implementation on FPGA are either using expensive ADC/DAC [5] [14] or tested on fewer channels [15].

In this paper, we propose to facilitate the use of FPGAs in multichannel systems by developing two different audio expansion boards.

The first one, referred to as the "Frugal Board," facilitates the construction of systems with a high number of outputs at a minimal cost. It focuses on affordability and can be used with a low-cost FPGA-based board due to

[1] *Central Processing Unit*
[2] *Graphics Processing Units*

[3] See L-ISA: https://l-isa.l-acoustics.com – All URLs in this paper were verified on March 8, 2024.
[4] *General Purpose Input/Output*
[5] *Analog to Digital/Digital to Analog Converters*

its PMOD [6] connectors available on entry-level Digilent hardware.

The second board, named "Ultra Low Latency Board" (or ULL), aims to leverage the full potential of the FPGA by employing an ultra-low-latency audio codec capable of achieving sampling frequencies up to 768kHz and latency as low as $11\mu s$. It is designed to operate with Xilinx's Ultrascale+ FPGAs, which are higher-end compared to those used with the Frugal board.

We then detail how their integration into the Syfala toolchain [16] simplifies their usage by providing a compilation flow from high-level audio DSP specifications down to FPGA bit-streams. Finally, we describe two application that we are currently testing for each board.

Each platform is open-source and available on GitHub. [7]

2. FRUGAL BOARD

In [17], we demonstrated the feasibility of designing a frugal multichannel platform by employing low-cost audio codec. We used an affordable Zybo Z7-20 [8] ADM/Xilinx FPGA development board coupled with 32 MAX98357A [9] codecs to make a WFS system for less than 800 USD. The MAX98357A codecs integrate digital-to-analog converters and built-in 3 watts amplifiers. Communication with these codecs occurs through the I2S [10] TDM [11] protocol, enabling communication with multiple codecs through a single wire and significantly reducing the required number of GPIOs.

However, connecting the 32 codecs necessitates substantial wiring, signal buffering and shielding, and non-trivial power management. In our prior work [17], we built the interface board that embeds the codecs using a dot matrix PCB [12] and hand-soldered cables, a time-consuming and non-robust process that resulted in erratic system operation and hard to debug problems.

To improve reliability, we present a custom-designed PCB in Fig. 1 aiming to enable individuals and institutions with limited resources to build a multichannel platform on FPGA. Two different manufacturing techniques are possible: either using existing commercial Adafruit breakout boards [13] allowing for hand-soldering the audio codecs and a complete DIY fabrication; or using specialized equipment to solder the codec directly to the board to enhance integration.

2.1 DIY Fabrication Approach

A reproducible board implies that it should be manufacturable using basic PCB fabrication techniques, without the need for advanced industrial machinery. We followed

[6] *Peripheral Module interface*

[7] `https://github.com/inria-emeraude/syfala`

[8] `https://digilent.com/reference/programmable-logic/zybo-z7/start`

[9] `https://www.analog.com/en/products/max98357a.html`

[10] *Inter-IC Sound (IIS or I2S)* is a serial bus interface standard used to interconnect digital audio devices.

[11] *Time-Division Multiplexing*

[12] *Printed Circuit Board*

[13] `https://www.adafruit.com/product/3006`

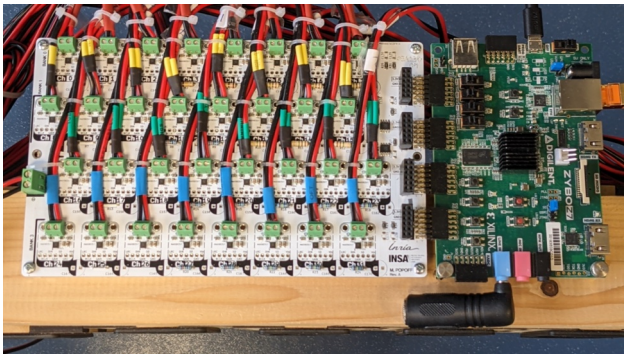

Figure 1. Frugal Board (white) with 32 wired speakers, connected to a Zybo Z7-20.

this requirement by using sufficiently wide PCB track widths to allow for high tolerance during printing, sufficiently large drill holes for manual drilling, routing tracks only on the two outer layers, and soldering all components on one side to facilitate potential hand soldering.

A major challenge of this constraint is that the MAX98357A audio codec has a footprint that is too small for manual soldering and requires the use of a specific equipment. Fortunately, we can use Adafruit breakout boards that integrates the codec as well as a few other essential components soldered onto a small PCB. This PCB can then be hand-soldered onto our board to connect the audio codec to the FPGA (see the top of Fig. 2).

All these constraints make it possible to assemble this board entirely by hand.

2.2 Direct Component Integration Approach

As manufacturing techniques become more accessible, it may be worthwhile to have the board manufactured and the components soldered by a specialized firm. In this case, using Adafruit breakout boards may be more complicated and expensive than purchasing the codec directly. Therefore, in addition to connectors for Adafruit breakout boards, the PCB also includes the footprint of the MAX98357A and peripheral components. This allows for the direct soldering of the audio codec and facilitates integration (see the bottom of Fig. 2).

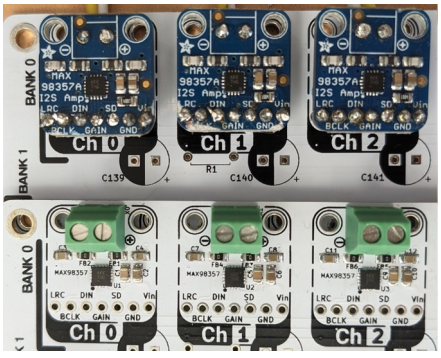

Figure 2. Two Frugal Boards side by side with the two different integration: direct MAX98357A audio codec integration (bottom) or using Adafruit breakout boards (top).

2.3 Stacking Capabilities and Clock Signals Integrity

Multiple boards can be stacked to increase the number of channels on a single FPGA. This is facilitated by the presence of stackable PMOD connectors and the possibility to select other GPIOs than the defaults for communication with the codecs (using jumpers). Using the TDM I2S protocol (as explained in Fig. 3), a 32 codecs board need 6 GPIOs: 4 for the data (one for 8 codecs) and 2 for the clocks (shared between all the amplifiers). Bit Clock and Word Select Clock can be shared between stacked boards, so each added boards will only need 4 GPIOs for data (TDM I2S wires in Fig. 3). Knowing that the board can access 32 GPIOs from the FPGA, we can connect up to 7 Frugal Boards to a Zybo Z7-20, allowing for 224 output channels.

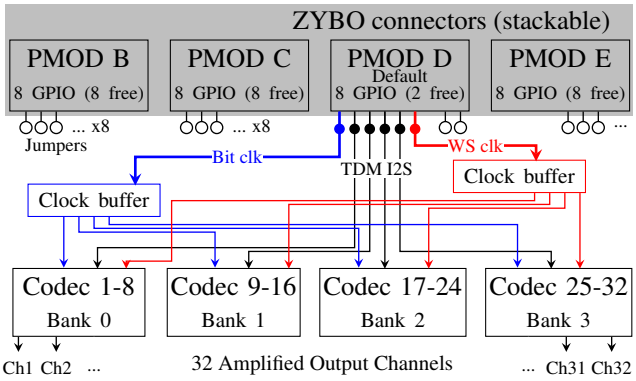

Figure 3. Frugal Board I2S Wiring

To ensure signal integrity and the possibility of connecting multiple boards in parallel to a single FPGA, it is necessary to use a clock buffer on the clock wires. We choose the low cost and low jitter 551S [14] clock buffer that takes the form of a hand-solderable surface-mount component accompanied by damping resistors.

3. ULTRA LOW LATENCY BOARD

The computational power and parallelization capabilities of FPGAs permit the implementation of powerful multichannel algorithms with unmatched low audio latency. While the previous section focused on low cost interfacing, the board described here enables ultra-low latency audio DSP. In our previous work [9], we highlighted the significant role of the audio codec and the associated antialiasing filters, as it is one of the contributors to latency.

To leverage the computational power of the FPGA, it is counterproductive to use inexpensive codecs that introduce excessive latency. Our preliminary work identified the ADAU1787 [15] ultra-low latency codecs that can operate at up to 768kHz and achieve a latency of 11μs from the analog input to the analog output using the Syfala compiler [9]. Moreover, while the Frugal Board in section 2 only featured codecs capable of synthesizing an audio signal (output channel), the ADAU1787 also hosts audio inputs.

For the experimental tests in [9], we used evaluation boards [16] provided by Analog Devices. Building a 32-input/output audio interface can be done using 16 of these evaluation boards (each codec managing 2 channels). However, this approach poses several problems: (i) a prohibitively high cost (∼8000 USD), (ii) a bulky system, due to the form factor of these boards featuring multiple connectors and peripherals unnecessary for our applications, and (iii) substantial wiring, leading to reliability issues.

We propose a custom PCB in Fig. 4 with embedded ADAU codecs that addresses these three issues and facilitates the use of these codecs.

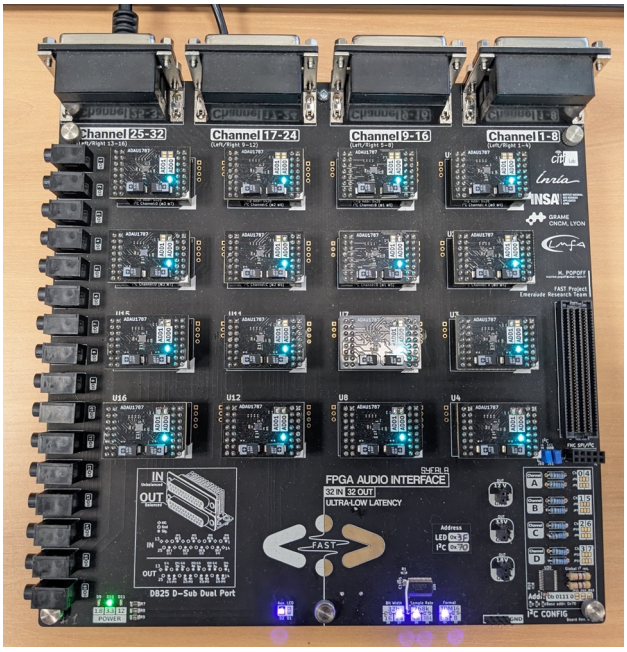

Figure 4. 32 Channels Ultra Low Latency FPGA interface stacked on the Genesys ZU-3EG. The motherboard embeds 16 breakout boards with ADAU1787 codecs.

3.1 Modular Approach

The ADAU1787 codecs are designed for portable devices such as smartphones and digital cameras, thus they are extremely compact. Their small footprint requires advanced equipment for integration, leading to costly PCB printing techniques.

Designing a board with 16 codecs soldered onto it leads to high production costs and several disadvantages, including the inability to individually replace a malfunctioning codec and the impossibility of making later interface changes (e.g., audio port modifications, filter additions, etc.).

For these reasons, we opted for a two-PCB fabrication approach. The first PCB (breakout board) exclusively embeds the codec and its peripheral components (Fig. 5).

[14] https://www.renesas.com/us/en/document/dst/551s-datasheet

[15] https://www.analog.com/en/products/adau1787.html

[16] https://www.analog.com/en/resources/evaluation-hardware-and-software/evaluation-boards-kits/eval-adau1787z.html

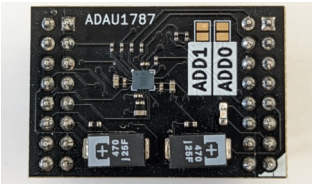

Figure 5. ADAU1787 breakout board.

It employs costly manufacturing techniques but remains minimalist to reduce costs per codec. The second circuit (motherboard) incorporates all the necessary connectors (audio and FPGA) and enables connection to 16 codec PCBs. This setup facilitates the replacement of malfunctioning codecs and eases interface changes in the future, leveraging existing codecs.

3.2 Audio Interface

The ADAU1787 embeds 2 single-ended input channels and 2 output channels, either differential or on an amplified and filtered headphone port. As shown in Fig. 6, on our board, the single-ended input and differential output are connected to DB25 connectors, with 8 channels per connector (4 input connectors and 4 output connectors for a total of 32 channels). These connectors are widely used in the field of audio and facilitate the connection to an amplifier or the use of a multicore cable with jack or XLR connectors.

The headphone output channels are directly routed to a 3.5mm stereo jack port for direct use with headphones.

3.3 FPGA Interface

To communicate with the FPGA, we use an FMC connector. [17] FMC is a standard that defines a compact electromechanical expansion interface for a daughter card to an FPGA baseboard. Most FPGA evaluation boards feature an FMC connector. By using this port, we ensure mechanical and electrical compatibility of our board with a large portion of commercially available FPGA boards.

On the Genesys ZU-3EG, [18] this port allows us to access up to 36 pairs of differential or 72 single-ended signals. Each codec requires 2 dedicated data wires (in and out), and 5 shared wires: 3 clocks (word select, bit clock, and master clock) and 2 wires for the I2C [19] bus. Thus, connecting 16 codecs require 37 wires that can fit on the FMC connector with signals in single-ended mode.

3.4 I2C Configuration

Unlike MAX98357A codecs in section 2, which are very simple and don't have software configuration, the ADAU1787 codecs require configuration via the I2C protocol at each boot. In order to configure each codec independently, they must all have a unique I2C address. The

[17] https://www.samtec.com/standards/vita/fmc/
[18] https://digilent.com/reference/programmable-logic/genesys-zu/start
[19] *Inter-Integrated Circuit (IIC or I2C)* is a serial communication bus used for communication between integrated circuits

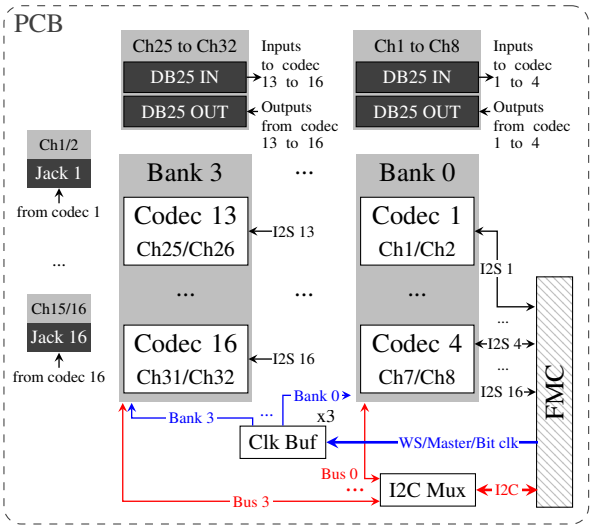

Figure 6. Ultra Low Latency Board I2C and I2S Wiring

codec's address is set by solderable jumpers located on the breakout board. However, the ADAU1787 can only accept four different addresses.

The solution is either to use a new I2C bus for each batch of four codecs, which is costly in terms of GPIO, or to use an I2C multiplexer that can switch on multiple sub-buses from a single I2C bus. If codecs have identical addresses but are on different sub-buses, they can be differentiated. With 16 codecs capable of taking four distinct addresses each, we require at least a 1-to-4 multiplexer. We chose the widely used TCA9548 [20] 1-to-8 multiplexer.

3.5 Stacking Capabilities

A single board embeds 16 codecs, providing 32 input/output channels. However, multiple boards can be parallelized on the same FMC port to increase the number of I/Os. Each codec communicates with the FPGA through its dedicated I2S data in and data out wires, resulting in 16 distinct I2S buses on the FMC port. All buses share the same Word Select, Master Clock, and Bit Clock wires.

By employing the I2S TDM protocol as done in Section 2 instead of the traditional I2S protocol, up to 16 codecs can communicate over the same data in and data out wires.

Therefore, it is straightforward to connect codecs in parallel and utilize TDM8 (up to 8 codecs on one bus) or TDM16 (up to 16 codecs on one bus) to communicate with all of them. Consequently, it becomes possible to parallelize up to 16 motherboards, each containing 16 codecs, resulting in a total of 256 codecs or 512 channels.

3.6 Design Considerations and Features

Given that clock signals (WS, Master, and Bit Clock) are shared from one GPIO to all codecs, it is wise to add a clock buffer to these wires to increase their strength and improve signal integrity. The buffer must have very low latency jitter, very low propagation delay and very low output skew (e.g., LMK1C1104 [21]) to ensure synchronic-

[20] https://www.ti.com/product/TCA9548A
[21] https://www.ti.com/product/LMK1C1104

ity between all codecs. The PCB has also been designed to potentially add damping resistors in series later to help eliminate reflection.

Particular attention has been paid to the routing of tracks on the PCB. Since the tracks are highly parallel and can exceed 40cm, a spacing greater than 3W (where W is the track width) is maintained in order to minimize crosstalk.

The datasheet of the ADAU1787 specifies that it is not necessary to differentiate the analog ground plane (AGND) from the digital ground plane (DGND). We have chosen to use a single ground plane and not to separate them.

To facilitate debugging, each codec is paired with a status LED controlled by the PCA9956 [22] LED multiplexer.

This board should be compatible with any Ultrascale+ FPGA equipped with an FMC port, we also ensured that it is easily stackable with the Digilent Genesys ZU-3EG development board by using the same dimensions.

4. SYFALA TOOLCHAIN COMPATIBILITY

The use of these two interfaces is facilitated by their integration into the Syfala toolchain [9, 16].

The main challenge of using FPGAs for real-time audio signal processing is that they're notoriously complex to program, mostly because of their inherently low-level architecture involving the use of Hardware Description Languages (HDL) such as Verilog or VHDL. Syfala simplifies the programming of real-time audio DSPs on FPGAs using C++ or the Faust [23] programming language.

The number of audio inputs/outputs is directly derived from the Syfala input program and the toolchain automatically generates the adequate FPGA design, including an I2S transceiver with the appropriate number of channels and configurable sampling rate.

A dedicated option added to the compilation chain allows us to specify the use of one of the two interfaces described in Section 2 and 3. A configuration file manages the addresses of all I2C devices and codecs. The toolchain is compatible with Zynq 7000 (Zybo) and Ultrascale+ (Genesys) FPGAs.

This integration allows for the seamless use of these interfaces from the user's perspective, who only needs to focus on the audio DSP algorithm implemented on the FPGA without needing to manage the hardware.

5. APPLICATIONS

5.1 Frugal WFS System

The affordability of the Frugal Board interface opens the door for massive multichannel systems with several hundreds of speakers at a limited cost.

Building upon our previous work [17], we implemented a simple WFS algorithm with Faust on a Zybo Z7-20 with a 32 speakers array (Fig. 7). The WFS algorithm that we used for our system is fairly basic and standard but we obtained very convincing results. Accessibility comes at the expense of (i) a lower sound quality and (ii) a higher latency compared to expensive codecs. The MAX98357A can work at a maximum sampling frequency of 96kHz and introduce a latency of approximately $500\mu s$ at 48kHz. Since each codec is identical and managed by a single FPGA, all speakers are subject to the same latency and thus perfectly synchronized and suitable for audio spatialization applications. The built-in 3W amplifier and associated passive speakers demonstrated sufficient sound quality to achieve effective rendering of the audio spatialization algorithms that we tested.

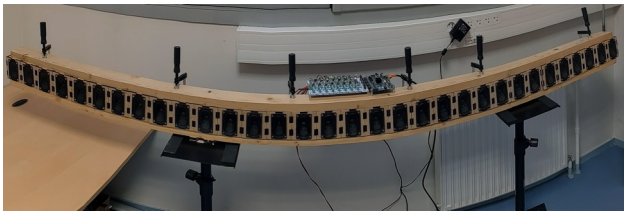

Figure 7. Prototype WFS system using the Frugal Board

5.2 Active Noise Control

The second interface will target more professional applications for spatial audio, especially those involving active control.

We collaborated with École Centrale de Lyon (France) to test sound field Active Noise Control (ANC) algorithms in their listening room (Fig. 8). This room is equipped with 20 speakers and a 6 microphones spherical array, allowing for the study of large-scale algorithms. We implemented two multichannel Filtered-x LMS algorithms to minimize either multi-point pressure or ambisonic components of the sound field [18]. We have tested these DSPs with only 6 speakers and 4 microphones due to the resource limitations of the FPGA at that time. However, our recent optimization efforts should enable us to conduct the test using all speakers and microphones in the room. The DB25 input/output connectors of our board enabled a seamless integration into their system and the power of the Ultrascale+ FPGA coupled with the low-latency ADAU1787 codecs allowed for the implementation of more efficient ANC algorithms.

Figure 8. Experimental setup inside the listening room.

[22] https://www.nxp.com/docs/en/data-sheet/PCA9956A.pdf

[23] https://faust.grame.fr/

6. CONCLUSIONS

In this paper, we presented two expansion boards for FPGA-based platform that facilitate the interfacing of multiple speakers and microphones with an FPGA. The two different approaches of these boards allow for various applications depending on the needs, either by prioritizing cost and ease of fabrication or performance.

We hope that this work will make massive multi-channel low-latency systems more accessible to the audio and music technology research communities. This should allow for the study and testing of many applications that were out of reach with traditional multichannel audio systems until now, especially in the context of active acoustic control and spatial audio.

Acknowledgments

This project has been partially funded by the French ANR (Agence Nationale de la Recherche) through the FAST project [24] (ANR-20-CE38-0001) and the Inria/Stanford Plasma Associate Team. [25]

7. REFERENCES

[1] A. J. Berkhout, D. de Vries, and P. Vogel, "Acoustic control by wave field synthesis," *The Journal of the Acoustical Society of America*, 1993.

[2] T. Ziemer, *Wave Field Synthesis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018.

[3] M. Noisternig, T. Carpentier, and O. Warusfel, "Dispositif de spatialisation sonore 3D à l'Espace de Projection de l'IRCAM," *Acoustique et Techniques : trimestriel d'information des professionnels de l'acoustique*, 2012.

[4] "Experimental media and performing arts center website," https://empac.rpi.edu.

[5] J. A. Belloch, A. Gonzalez, E. S. Quintana-Ortí, M. Ferrer, and V. Välimäki, "GPU-Based Dynamic Wave Field Synthesis Using Fractional Delay Filters and Room Compensation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[6] J. Lorente, M. Ferrer, M. de Diego, and A. González, "GPU Implementation of Multichannel Adaptive Algorithms for Local Active Noise Control," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2014.

[7] T. Reussner, C. Sladeczek, M. Rath, S. Brix, K. Preidl, and H. Scheck, "Audio Network-Based Massive Multichannel Loudspeaker System for Flexible Use in Spatial Audio Research," *Journal of the Audio Engineering Society*, 2013.

[8] N. Epain and E. Friot, "Active control of sound inside a sphere via control of the acoustic pressure at the boundary surface," *Journal of Sound and Vibration*, vol. 299, no. 3, pp. 587–604, 2007.

[9] M. Popoff, R. Michon, T. Risset, Y. Orlarey, and S. Letz, "Towards an FPGA-Based Compilation Flow for Ultra-Low Latency Audio Signal Processing," in *Proc. Int. Conf. in Sound and Music Computing, SMC-22*, Saint-Étienne, France, Jun. 2022.

[10] T. C. Vannoy, "Enabling rapid prototyping of audio signal processing systems using system-on-chip field programmable gate arrays," Master PhD, 2020.

[11] M. Verstraelen, J. Kuper, and G. J. Smit, "Declaratively Programmable Ultra Low-Latency Audio Effects Processing on FPGA." in *DAFx*, 2014.

[12] D. Theodoropoulos, G. Kuzmanov, and G. Gaydadjiev, "Multi-Core Platforms for Beamforming and Wave Field Synthesis," *IEEE Transactions on Multimedia*, 2011.

[13] ——, "A Minimalistic Architecture for Reconfigurable WFS-Based Immersive-Audio," in *2010 International Conference on Reconfigurable Computing and FPGAs*, 2010.

[14] D. Shi, W.-S. Gan, J. He, and B. Lam, "Practical Implementation of Multichannel Filtered-x Least Mean Square Algorithm Based on the Multiple-Parallel-Branch With Folding Architecture for Large-Scale Active Noise Control," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2020.

[15] A. Klemd, B. Klauer, J. Timmermann, and D. Sachau, "A Flexible Multi-Channel Feedback FxLMS Architecture for FPGA Platforms," in *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)*, 2021.

[16] M. Popoff, R. Michon, T. Risset, P. Cochard, S. Letz, Y. Orlarey, and F. de Dinechin, "Audio DSP to FPGA Compilation: The Syfala Toolchain Approach," Univ Lyon, INSA Lyon, Inria, CITI, Grame, Emeraude, Tech. Rep. RR-9507, May 2023.

[17] R. Michon, J. Bizien, M. Popoff, and T. Risset, "Making Frugal Spatial Audio Systems Using Field-Programmable Gate Arrays," in *Proceedings of the 2023 New Interfaces for Musical Expression Conference*, Mexico City, Mexico, May 2023.

[18] L. Alexandre, P. Lecomte, and M.-A. Galland, "Experimental Active Noise Control Using Faust On FPGA: Comparison Between A Multi-Point and Spherical Harmonics Method," in *Forum Acusticum 2023 - 10th Convention of the European Acoustics Association*, Torino, Italy, Sep. 2023.

[24] `https://fast.grame.fr`
[25] `https://team.inria.fr/emeraude/plasma/`