

PhD Topic

Ultra-Low Latency Audio Processing on FPGA Through High Level Synthesis

Tanguy Risset¹, Florent de Dinechin¹, Yann Orlarey², and Romain Michon²

¹Citi/Socrate, Insa-Lyon, Inria <https://team.inria.fr/socrate/>

²GRAME-CNCM, Lyon <http://www.grame.fr/>

Topic

The objective of this PhD is to combine the Faust programming language¹ Orlarey *et al.* (2009), High Level Synthesis tools, and FloPoCo² de Dinechin et Pasca (2011) to implement a “Faust to FPGA compiler” to facilitate the design of FPGA-based systems for ultra-low latency (less than 50 microseconds) real-time audio signal processing. It will be funded by the ANR FAST project³ (Fast Audio Signal-processing Technologies on FPGA) that will start in March 2021. The goal of FAST is to use the prototype FPGA compiler developed during this PhD to solve difficult problems such as active acoustic control.

Faust is a programming language specifically designed for real-time audio signal processing applications. The Faust compiler makes it possible to quickly develop efficient and reliable software for the main audio platforms (i.e., VST and AU plugins, Max/MSP, PureData, macOS, iOS, Android, Web, Linux, etc.). Faust is developed by GRAME and was created by Yann Orlarey who will act as one of the supervisor for this PhD. Faust has been increasingly used in recent years in the music technology research community as well as in the industry (e.g., Moforte, Analog Devices, etc.).

All real-time audio systems induce some latency: software sound processing needs each audio sample to go through all the software layers and the hardware audio interface. For example, it might take a few milliseconds between a key-press and the resulting sound. While a latency of a few milliseconds is acceptable for most applications, the active correction of the resonance of an acoustic instrument, for example, will benefit from ultra-low latency (in the order of one or two samples), which cannot be achieved by “standard” audio systems. On the other hand, FPGA platforms can take latency down below 50 microseconds. Some works have already proposed this type of FPGA acceleration in this context Verstraelen *et al.* (2014). However, FPGAs must be manually programmed, which is much more complex than programming a conventional computer.

FloPoCo is developed by Florent de Dinechin who will serve as one of the advisor of this PhD. It aims at the synthesis (in the sense of VHDL synthesis) of high level arithmetic operators and basic functions (e.g., log, exp, FIR filters, etc.) on FPGAs. FloPoCo will be useful for compiling Faust to FPGA because fixed-point operators have to be precisely tuned to each audio signal processing problem. A first internship (before the PhD) will propose an extension of the Faust compiler to generate fixed-point arithmetic operators from Faust programs using FloPoCo. This component (i.e., Faust to Fixed Point component) will be continuously improved throughout the PhD.

¹<https://faust.grame.fr>

²<http://flopoco.gforge.inria.fr/>

³<https://fast.grame.fr>

The Faust compiler can currently produce C++ code. In order to program FPGA platforms, we will rely on recent industrial High Level Synthesis (HLS) tools. These HLS tools are able to compile a high level software function (in C for instance) down-to an FPGA bitstream with an interface with the embedded processor present on the FPGA chip. Hence, one of the technical difficulty is the mastering of classical FPGA design flow (through VHDL synthesis) and new HLS design flow (including automatic parallelization, and hardware software co-design). An alternative direct flow from FPGA to VHDL will also be explored.

A first prototype compiler was built as part of the Syfala project⁴ Risset *et al.* (2020) using the `vivado_hls` tool from Xilinx. We were able to compile simple Faust programs on the Zybo-Z7 Xilinx board, but many problems remain unsolved: hardware software interface with the Zynq processor, control of the audio effect with external controllers, efficient memory accesses for the audio samples, etc.

Hence the objective of this PhD is to study how Faust, FloPoCo, and HLS can provide a breakthrough in rapid and efficient compilation tools for real-time audio processing on FPGA to reach ultra-low audio latency.

Conditions

- **Duration:** 3 years
- **Fellowship/Funding:** yes
- **Teaching Load:** no (100% research)
- **Starting Date:** Anytime between March and September 2021
- **Prerequisite:** MS

The candidate will be granted a 3 years fellowship. He will be based in Lyon (France), in the Socrate/Emeraude team (CITI, INSA, the team is moving from Socrate to Emeraude). The Emeraude team is working on FloPoCo, embedded systems and FPGA, and very low consumption systems. This PhD will also be carried out in collaboration with the GRAME research team responsible for the development of Faust. Frequent discussions will occur with the Center for Acoustics at LMFA⁵ (École Centrale Lyon) which is the third partner of the FAST ANR project.

The PhD will be organized as follows:

- The candidate will get familiar with the different tools: Faust, FloPoCo, Xilinx tools (synthesis and HLS) as well as the existing prototype compilation flow from the SyFaLa project.
- Building upon developments already carried out as part of SyFaLa, the PhD candidate will:
 - Study optimization of compiled programs by various means: parallelization, fixed-point arithmetic optimization, hardware/software mapping between FPGA and Zynq processing.
 - Allow for various control means of audio processing: via software (Zynq processor) of hardware (knobs)
 - Validate the flow on non-trivial audio processing used in the applications targeted by the FAST ANR project.
 - Consider potential open source solutions.
- Then the candidate will focus its effort on porting the tool-chain on a more expensive FPGA platform (such as Ultrascale FPGA Genesys ZU) in order to implement more computationally expensive audio processing algorithms such as that used in active acoustic control.

⁴<https://faust.grame.fr/syfala/>

⁵<https://acoustique.ec-lyon.fr/>

The main required skills are basic training in compilation, system programming and FPGA. Experience in embedded programming will be necessary too. The mastering of the C and C++ languages as well as solid foundations in computer architecture are important too. Finally, basic signal processing (and if possible audio signal processing) and/or HLS will be very useful.

Contacts

For any further information, please contact Tanguy Risset: tanguy.risset@insa-lyon.fr.

References

Florent de DINECHIN et Bogdan PASCA : Designing custom arithmetic data paths with flopoco. *IEEE Design & Test of Computers*, 28:18–27, 04 2011.

Yann ORLAREY, Stéphane LETZ et Dominique FOBER : *New Computational Paradigms for Computer Music*, chapitre “Faust: an Efficient Functional Approach to DSP Programming”. Delatour, Paris, France, 2009.

Tanguy RISSET, Romain MICHON, Yann ORLAREY, Stéphane LETZ, Gero MÜLLER, Adeyemi GBADAMOSI, Luc FORGET et Florent de DINECHIN : Faust2fpga for ultra-low audio latency: Preliminary work in the syfala project. *In International Faust Conference (IFC)*, 2020.

Martinus Johannes Wilhelmina VERSTRAELEN, Jan KUPER et Gerardus Johannes Maria SMIT : Declaratively programmable ultra-low latency audio effects processing on fpga. *In Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)*, pages 263–270. Fraunhofer Institut, 9 2014.