Compilation/System Engineer Position (18 Months) Fixed-Point Extension for the Faust Programming Language

Romain Michon[1], Yann Orlarey[2], Tanguy Risset[1], Florent de Dinechin[1], and Stéphane Letz[2]

[1]Citi/Emeraude, INSA-Lyon/Inria
[2]GRAME-CNCM, Lyon

1 General Information

- **Theme/Domain:** Compilation, FPGA, Digital Signal Processing
- **Duration:** 18 months
- **Starting date:** January 2022 (Flexible)
- **Location:** Lyon (France)
- **Employer:** GRAME, Centre National de Création Musicale (https://www.grame.fr)
- **Funding:** FAST ANR Project[1]
- **Salary:** In function of experience
- **Benefits:** Tickets Restaurant
- **Contact:** michon@grame.fr

2 Context

FAST is a research project funded by the Agence Nationale de la Recherche (ANR – the French National Research Agency). It gathers the strength of GRAME-CNCM,[2] CITI Lab (INSA Lyon),[3] and LMFA (École Centrale Lyon)[4] towards two goals:

- Facilitating the design of ultra-low latency embedded systems for real-time audio signal processing,
- Using such systems in the context of active control of acoustics.

FPGA-based technologies are used at the heart of FAST to create an ultra-low latency (inferior to 50 microseconds) multi-channel sound processing system. The FAST programming toolchain is based on the Faust programming language [2].[5] Faust is specifically designed for real-time audio

[1] https://fast.grame.fr
[2] https://www.grame.fr
[3] http://www.citi-lab.fr/
[4] http://lmfa.ec-lyon.fr/
[5] https://faust.grame.fr

signal processing applications. GRAME-CNCM is the birthplace of Faust. 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 now a recognised standard in the music technology research community as well as in the industry.

One of the problems to solve to improve the efficiency of the FAST compilation chain is to allow Faust to generate fixed point rather than floating point DSP code. Indeed, while floating points are easily handled by modern CPUs hosting a Floating Point Unit (FPU), FPGAs are much more efficient when fixed point operations are used. This is far from being simple, especially because the size/length (precision) of integers can vary (e.g., high precision integers will take more space on the FPGA than lower precision integers).

FloPoCo [1] [6] – which is developed by Florent de Dinechin – will play a central role in this task. FloPoCo aims at the synthesis (in the sense of VHDL synthesis) of high level arithmetic operators and basic functions (log, exp, FIR filters, etc.) on FPGA. FloPoCo will be useful for compiling Faust to FPGA because fixed-point operators have to be precisely tuned to each audio signal processing problem.

Preliminary work in that direction has been carried out as part of two recent masters-level research internships, but much has yet to be done.

3 Tasks

The recruited candidate will be responsible of the following tasks (under the supervision of the FAST core research team):

- Improve fixed-point support in the Faust compiler.
- Contribute to the development of the Faust to VHDL compiler developed as part of FAST.
- Improve the C++ code generated by Faust in the context of HLS.
- Continue the work that was started on using FloPoCo in the context of the FAST toolchain.

4 Skills

The ideal candidate should master the following skills:

- Advanced C/C++ programming,
- Compilation,
- FPGA programming (Xilinx tools, VHDL – we use Vivado/Vitis, Zybo and a Genesys ZU 3EG boards, etc.),
- Unix tools (i.e., using a terminal, git, etc.).

Additionally, while not mandatory, the following skills will be considered as a plus:

- Audio Digital Signal Processing (DSP),
- Experience with fixed-point audio DSP,
- Faust programming.

[6] `http://flopoco.gforge.inria.fr/`

5 Job Conditions

The selected candidate will be recruited by GRAME-CNCM on a determined duration contract (CDD). Office space will be provided at the GRAME research department (11 cours de Verdun-Gensoul, 69002 Lyon, France) but occasional work sessions will take place at CITI Lab (INSA Lyon) and at École Centrale of Lyon. Teleworking will be accepted (2 days per week, max).

References

[1] F. de Dinechin and B. Pasca. Designing custom arithmetic data paths with flopoco. *IEEE Design & Test of Computers*, 28:18–27, 04 2011.

[2] Y. Orlarey, S. Letz, and D. Fober. *New Computational Paradigms for Computer Music*, chapter "Faust: an Efficient Functional Approach to DSP Programming". Delatour, Paris, France, 2009.