First Syfala Workshop: Experimenting With the Faust to FPGA Compilation Flow Using Grid5000 Insa-Lyon 1/12/2023

Pierre Cochard, Maxime Popoff, Romain Michon, Tanguy Risset, Yann Orlarey, Matthieu Imbert and The Emeraude Team

Citi Laboratory (INSA de Lyon, INRIA,GRAME-CNCM)

November 30, 2023

Table of Contents

1. Things to know about today
2. Emeraude Team
3. Syfala Project: Audio to FPGA Compilation
4. High Level Synthesis
5. grid5000 for accessing Xilinx tools
6. Conclusion

Table of Contents

1. Things to know about today
2. Emeraude Team
3. Syfala Project: Audio to FPGA Compilation
4. High Level Synthesis
5. grid5000 for accessing Xilinx tools
6. Conclusion

Syfala Recap: Challenges and Future Works

Program: (`https://paw.grame.fr/#syfala`)

- 09h00: Workshop Introduction (Tanguy Risset)
- 09h30: Introduction to Faust Langage (Yann Orlarey)
- 10h00: Hands-on Faust Language (Yann Orlarey)
- coffee Break
- 10h30: Introduction to FPGA (Maxime Popoff)
- 11h00: Introduction to Syfala-lab (Pierre Cochard)
- 11h30: Connecting to Grid5000 (Syfala Team)
- 12h30: Lunch (Room 'projet A', same floor)
- 14h00: Hands-on: Running Syfala Linux Workfkow
- 15h30: Advanced Syfala Use (Pierre Cochard, Maxime Popoff, Romain Michon)
- 16h30: Hands-on sand box: free experimental Audio to FPGA programming
- 18h00: End

Participants:

- 8 participants with various background
- +3 local attendees

Instructors:

- Pierre Cochard (github master)
- Maxime Popoff (Phd on Syfala)
- Yann Orlarey (Faust inventor)
- Romain Michon (embedded audio)
- Matthieu Imbert (grid5000)
- Tanguy Risset (head)

Table of Contents

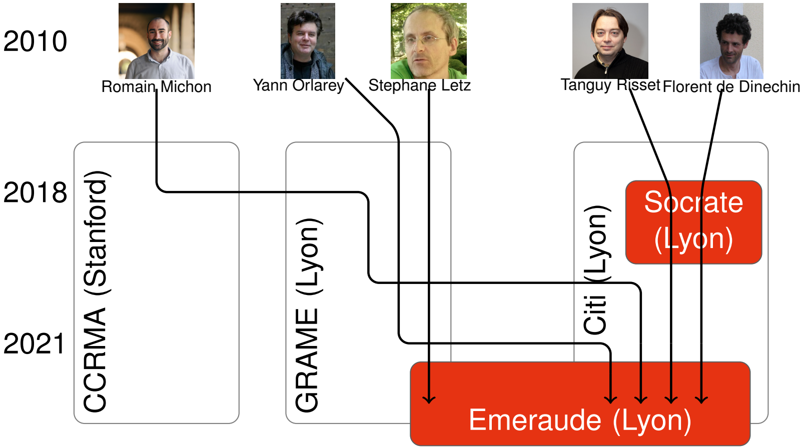
1. Things to know about today
2. Emeraude Team
3. Syfala Project: Audio to FPGA Compilation
4. High Level Synthesis
5. grid5000 for accessing Xilinx tools
6. Conclusion

Why emeraude? and what for?

- Emeraude stands for:
  Embedded Programmable Audio Systems
- New Research Team in Lyon (Mar. 2022) collaboration between The Grame institute and Citi research lab.

- Domains of expertise:
  - Embedded Audio Systems
  - Acoustics
  - Sound synthesis and effects
  - Arithmetic for FPGA.
  - Audio on FPGA
  - Faust language development

A zoom on Emeraude’s origins



2010

2018

2021

Romain Michon

Yann Orlarey

Stephane Letz

Tanguy Risset

Florent de Dinechin

CCRMA (Stanford)

GRAME (Lyon)

Citi (Lyon)

Socrate (Lyon)

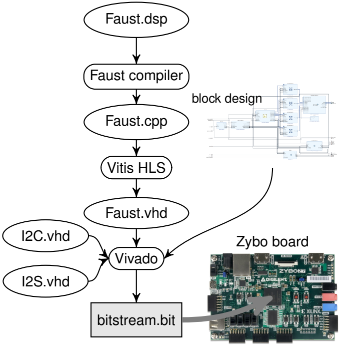
Emeraude (Lyon)

Table of Contents

Syfala development stages

Syfala original objective: reach smallest possible audio latency

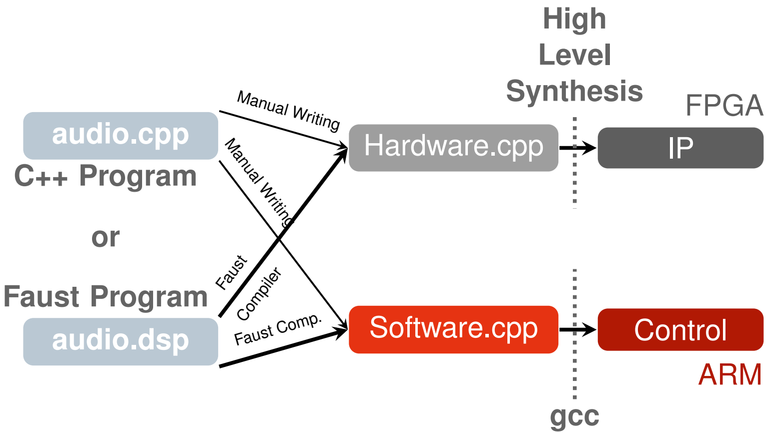
- First version (2020 no ARM control)



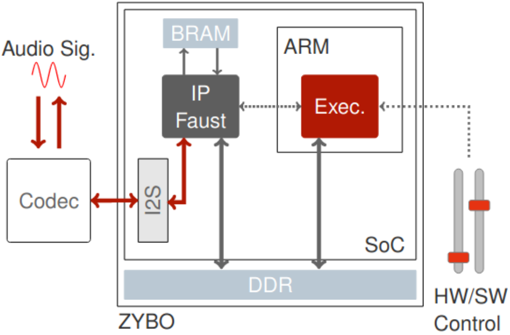
- 2021: Control on ARM (hardware or software-UART-based)
- 2021: Large memory handling (long delay lines)
- 2022: Ultra-low latency (11μs analog to analog)
- 2022: Parametric number of audio channels
- 2022: TDM I2S (up to 256 channels on Zybo Z10)
- 2023: Embedded Linux on ARM (Eth., Wifi, MIDI, etc.)
- 2023: C++ based compilation flow

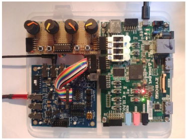
Syfala Compiler (2023) Conceptual View

Two possible flows: from Faust or from C++.

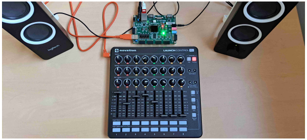


Resulting audio system





Bare metal



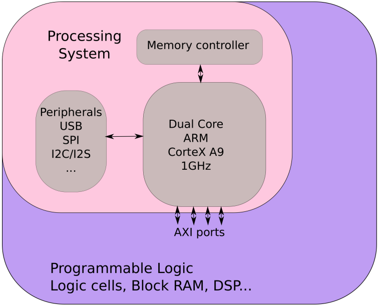
With Linux

Table of Contents

1. Things to know about today
2. Emeraude Team
3. Syfala Project: Audio to FPGA Compilation
4. **High Level Synthesis**
5. grid5000 for accessing Xilinx tools
6. Conclusion

Programming an FPGA Means ...

- many logic cells ("LUT")
- many small (≈ 24 bit) hard multipliers ("DSP blocks")
- many small (≈ 10 kBit) memories
- A SoC which includes a *processing system* (ARM@1GHz)



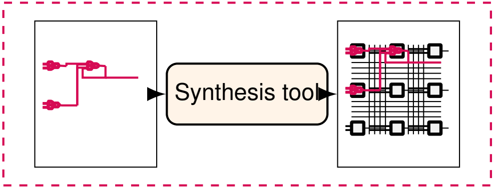
Zynq 7000 Processing System

Example: Xilinx Zynq 7010 (≈ 50€): 2 ARM processor cores

+ **28k** logic cells + **80** DSP blocks + **60** 36kBit memory block ... running at **200 MHz**

The real revolution: High Level Synthesis

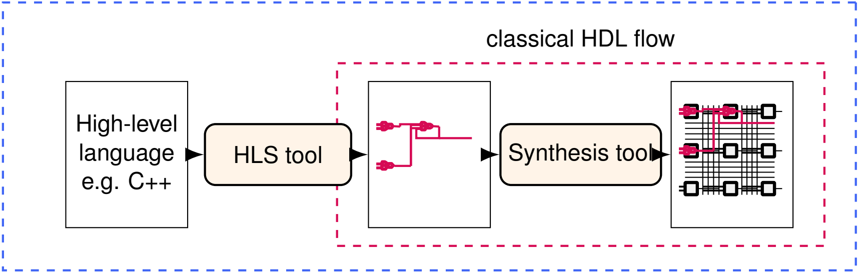
classical HDL flow



- Classic HDL flow: You don’t program, you design a circuit
  - with low-level languages such as VHDL or Verilog
  - with compilers called “synthesis tools” that can take hours

The real revolution: High Level Synthesis

High-level synthesis (HLS) flow



classical HDL flow

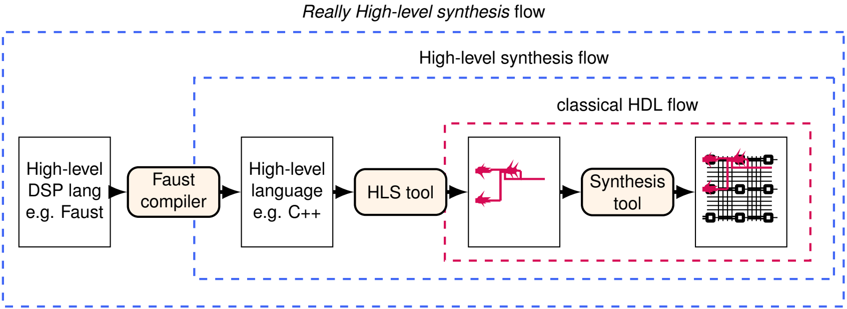
High-level language e.g. C++

HLS tool

Synthesis tool

- Classic HDL flow: You don’t program, you design a circuit
  - with low-level languages such as VHDL or Verilog
  - with compilers called “synthesis tools” that can take hours
- Since the 2010, HLS allows for FPGA programming in C/C++
  - but you won’t escape the synthesis tools

Syfala real challenge: control HLS compilation


Really High-level synthesis flow
High-level synthesis flow
classical HDL flow
High-level DSP lang e.g. Faust
Faust compiler
High-level language e.g. C++
HLS tool
Synthesis tool

- Add a higher level to High Level Synthesis.
  - Use HLS source as a backend of a compiler (Faust compiler)
- Use HLS compiler (`vitis_hls`) as a backend to Faust
  - Take advantage of parallel computation allowed by Faust

Table of Contents

1 Things to know about today

2 Emeraude Team

3 Syfala Project: Audio to FPGA Compilation

4 High Level Synthesis

5 grid5000 for accessing Xilinx tools

6 Conclusion

Syfala Infrastructure

what we need:

- Xilinx tools (80 GB):
  - `vitis_hls` (HLS)
  - `vivado` (hardware synthesis)
  - `vitis` (HW/SW system building)
- Linux building
  - Linux sources
  - `qemu` (cross compilation)
- Faust Compiler

Syfala Infrastructure

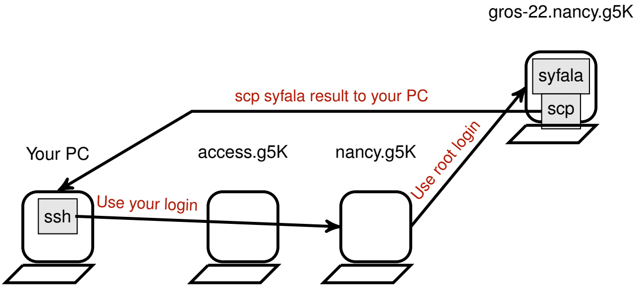
what we need:

- Xilinx tools (80 GB):
  - `vitis_hls` (HLS)
  - `vivado` (hardware synthesis)
  - `vitis` (HW/SW system building)
- Linux building
  - Linux sources
  - `qemu` (cross compilation)
- Faust Compiler

How to run all that:

- A simple solution is a Linux container
- grid5000 allows us to:
  - Use a distant Linux machine (at Nancy)
  - Create a temporary account
  - Import a Syfala container
  - Run Xilinx tools remotely
  - import bistream to flash on local Zybo boards

Syfala using grid5000



Of course before all that you need to access Wifi...

Table of Contents

1. Things to know about today
2. Emeraude Team
3. Syfala Project: Audio to FPGA Compilation
4. High Level Synthesis
5. grid5000 for accessing Xilinx tools
6. Conclusion

Conclusion: next talk to come

- Yann Orlarey
  - 09h30: Introduction to Faust Langage (Yann Orlarey)
  - 10h00: Hands-on Faust Language (Yann Orlarey)
- Maxime Popoff
  - introduction to FPGA
- Pierre Cochard
  - Syfala-lab presentation
- 11h30: hands-on: connect to grid5000 and then lunch

Questions?

`https://github.com/inria-emeraude/syfala`

-