EFCL Winter School – Track 2
Customizing RISC-V Based Microcontrollers

Lecture 4 – PULPissimo emulation on Zybo Z7

Yvan Tortorella - yvan.tortorella@chips.it

Track2 Solutions & Docs

Chips-IT

FIR XIFU exercise

- FIR XIFU RTL: https://github.com/pulp-platform/fir-xifu/tree/main
- FIR XIFU SW: https://github.com/FondazioneChipsIT/regression_tests/tree/EfclSolutions -> fir_xifu_minimal & fir_xifu_complete

HWPE Exercise

- FIR HWPE RTL: https://github.com/FondazioneChipsIT/fir-hwpe/tree/efclws2026_solutions
- FIR HWPE SW: https://github.com/FondazioneChipsIT/regression_tests/tree/EfclSolutions -> fir_new_hwpe

HWPE DOCs

- https://hwpe-doc.readthedocs.io/en/latest/

Overview of the course

Chips-IT

- PULP platform & PULPissimo microcontroller architecture (4h Mon)
- Extending RISC-V cores and the RISC-V LLVM compiler (6h Tue)
- Integrating cooperative HW Processing Engines / HWPEs (4h Wed-Thu)
- **Testing extended PULPissimo on FPGA (4h Thu)**
  - *Short lecture:* Overview of the PULPissimo FPGA emulation flow
  - *Hands-on*: Implementing our PULPissimo from synthesis to bitstream
  - *Hands-on*: Using *our* PULPissimo on the FPGA & OpenOCD-based debugging

TL;DR

Chips-IT

Suggestion: start Task 15 immediately during the lecture ☺

- The synthesis process can be started from your PULPissimo root with

```
make zyboz7
```

- In case of issues, the TAs will intervene after the lecture has ended (it'll be short!)

PULPissimo Zybo Z7 target

Chips-IT

The PULPissimo flow is organized in *targets*

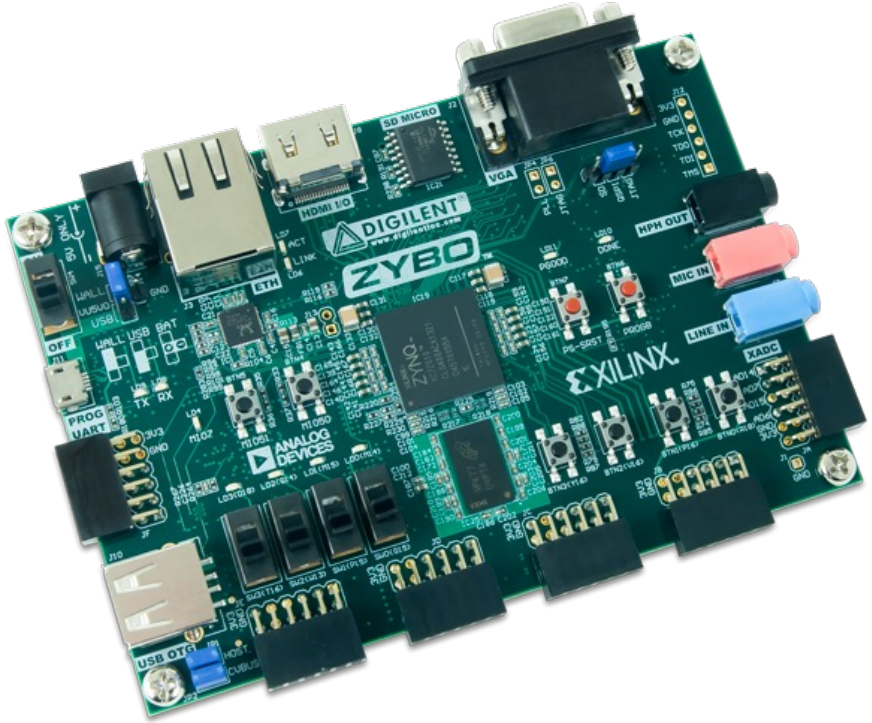
- So far, we only used the *sim/questasim* target
- Now it's time to dig into the *fpga* target

Target: Digilent Zybo Z7 board

- Mounting a Zynq-7000 SoC
- We'll use this primarily as an *emulation* target (e.g., no attempt to push frequency): PULP systems mainly address SoC designs

Folder is `fpga/pulpissimo-zyboz7`

- Starting point is the **`Makefile`**



PULPissimo Zybo Z7 target

Chips-IT

A few relevant rules

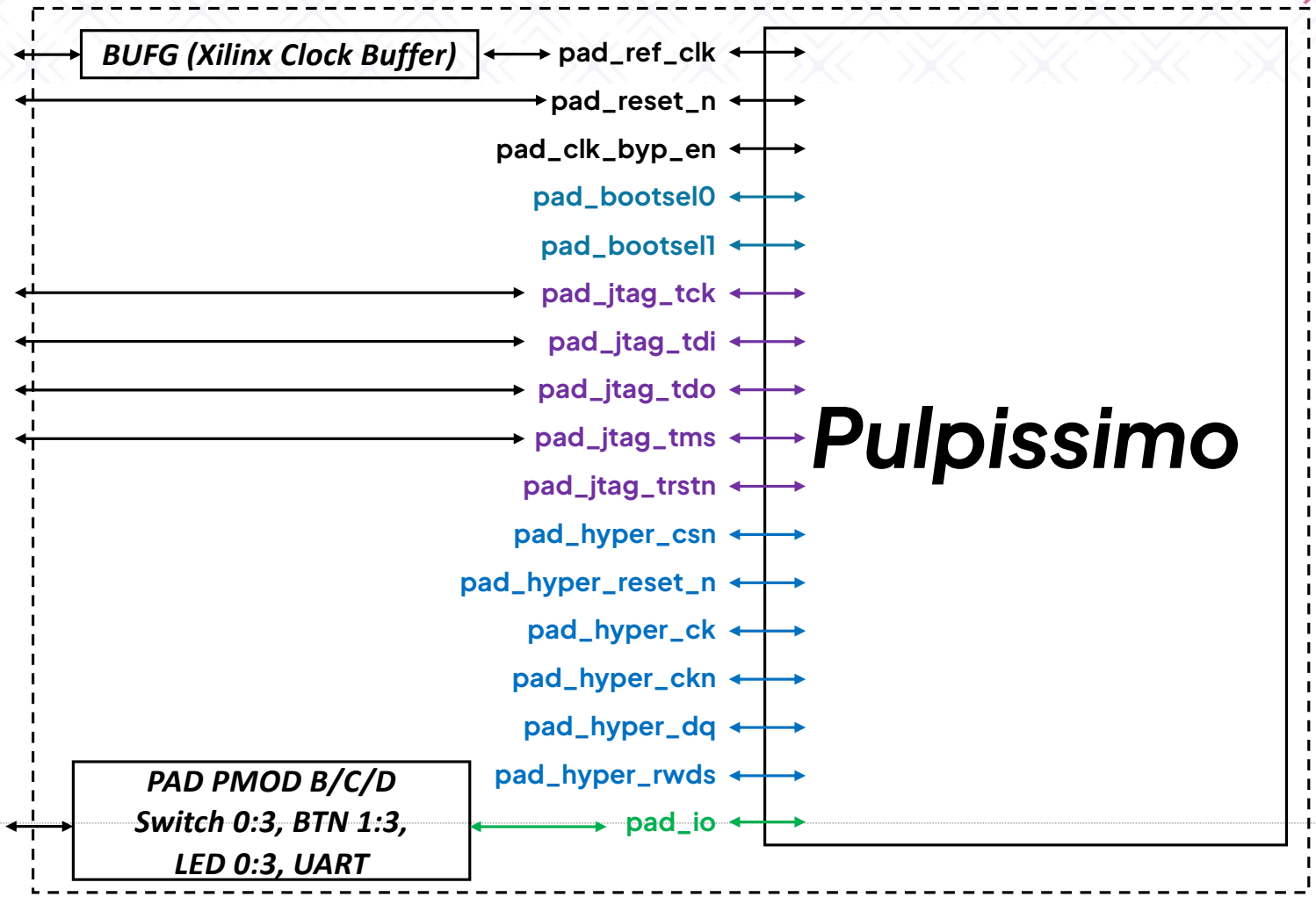
- `ips`: generate Xilinx IPs, mainly *clock managers* (which replace FLLs)
- `all`: synthesize and place&route PULPissimo in batch mode
  - even in batch mode, we utilize Vivado's "project-mode" (useful to recover partial statuses in GUI)
  - this essentially calls a TCL script in *tcl/run.tcl* to drive Vivado

PULPissimo flow makes small changes to design

- A dedicated wrapper to connect to the external board interfaces
- Use "synthesizable" latch-based clock-gates instead of BUFGCE (too scarce!)
  - *Scary, but it works...*

Xilinx PULPissimo wrapper

Chips-IT



Thank you

Chips-IT