

School and Conference on Edge Intelligence for Scientific Instrumentation and Reconfigurable Computing

Introduction to Laboratory Activities

Maynor G. Ballina
September 2026



The Abdus Salam
International Centre
for Theoretical Physics



UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

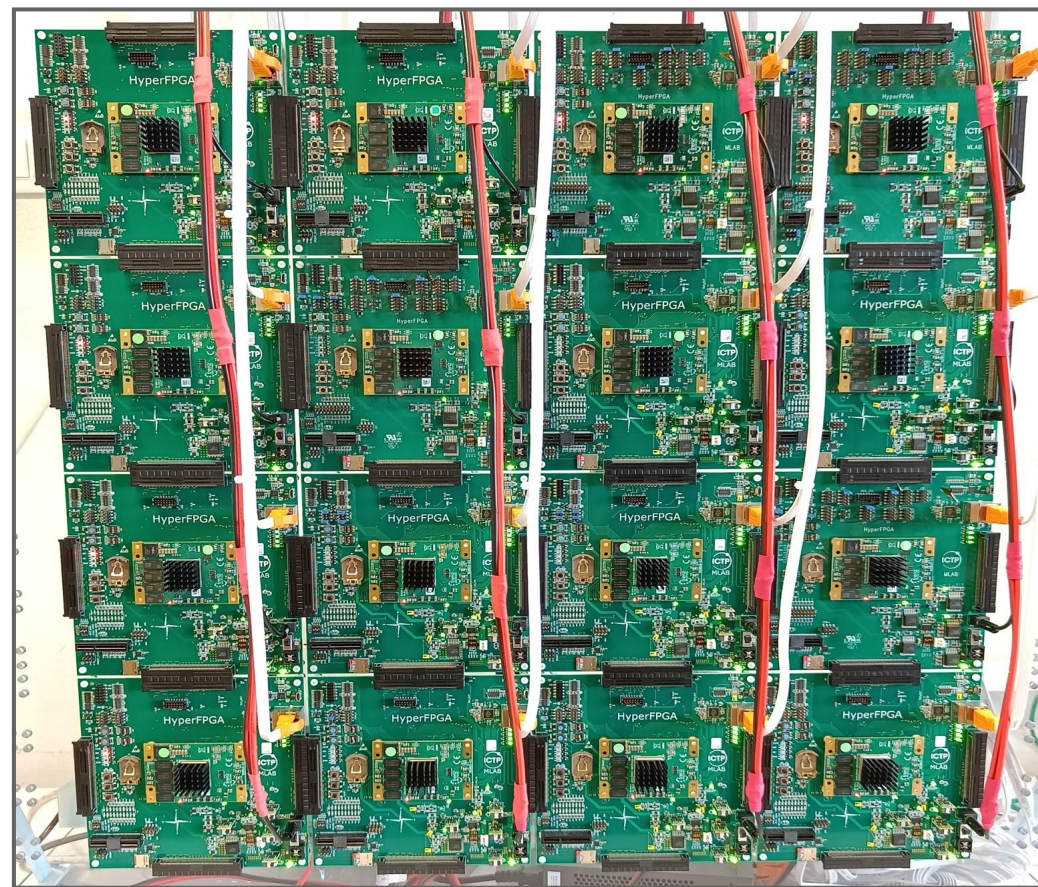
Welcome: Lab tutors

Dr. Romina Molina

Dr. Luis Garcia

Maynor Ballina

Kasun Mannatunga



Labs Overview:

- **Lab 0: User Environment and Access to the HyperFPGA Cluster**
- **Lab 1: Getting Started with MPSoC-FPGA and Comblock**
- **Lab 2 - Reconfigurable Instrumentation on MPSoC-FPGA**
- **Lab 3 - High-Level Synthesis (HLS) for Heterogeneous Computing**
- **Lab 4 - Training and Compression of Machine Learning Models**
- **Lab 5: End-to-End Workflow for Deploying Machine Learning Models on MPSoC-FPGA**
- **Final Project on Edge AI**

https://hyperfpga.sti.ictp.it



Project Overview

HyperFPGA is an open MPSoC-FPGA cluster designed for experimental research on heterogeneous computing and energy-efficient computing paradigms. Developed and maintained by the Multidisciplinary Laboratory (MLab), STI Unit, of the Abdus Salam International Centre for Theoretical Physics (ICTP), the platform offers flexible configurations — from electrical signaling standards to distributed computing models — and integrates independent power-consumption monitors that separate data movement from computation. By decoupling the hardware layer from user space, HyperFPGA allows researchers and students to seamlessly share reconfigurable resources, run hardware-accelerated workloads, and explore fine-grained, scalable heterogeneous computing solutions for a broad range of scientific and engineering problems.

Resources

HyperFPGA BSP

Board support package for the HyperFPGA MPSoC-FPGA cluster nodes.

[View Repository →](#)

HyperFPGA User Guide

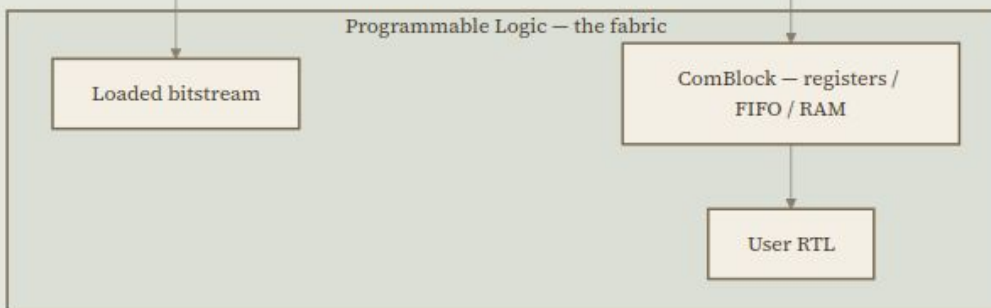
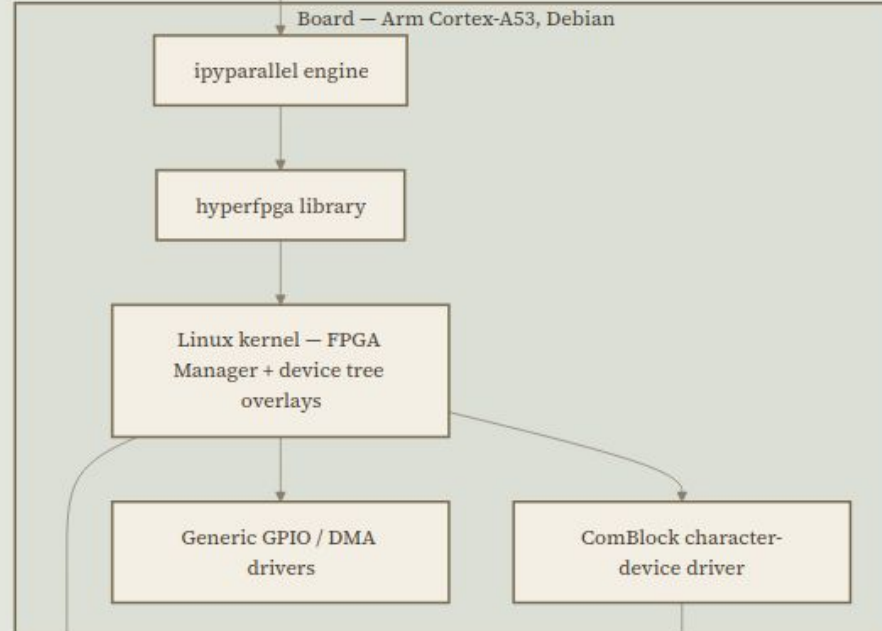
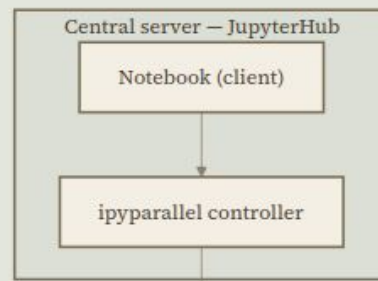
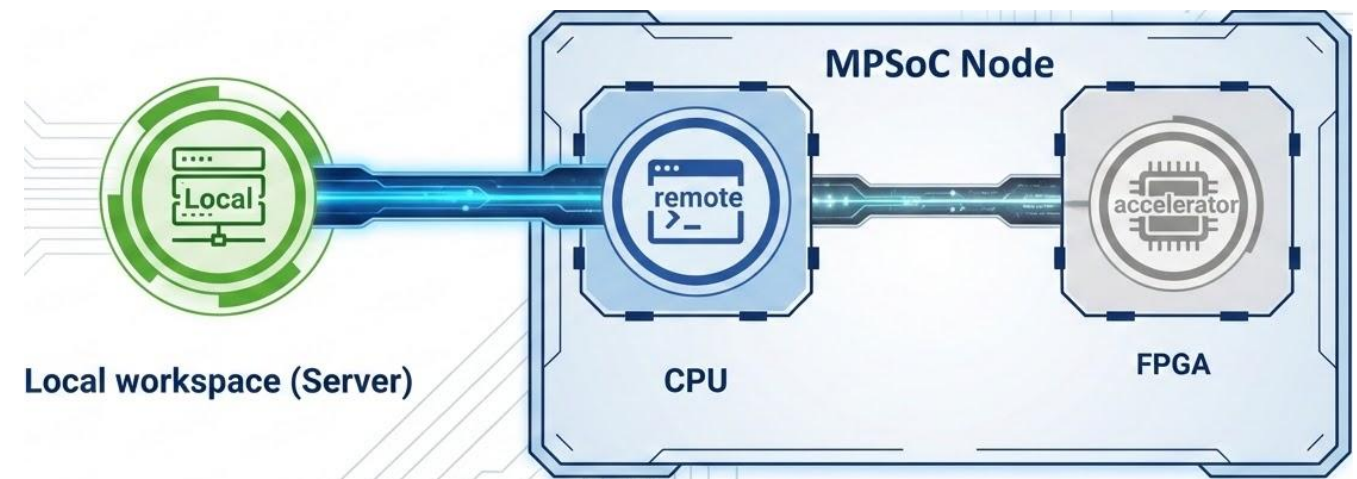
Comprehensive user guide covering the HyperFPGA comutils API, cluster usage, and hardware workflows.

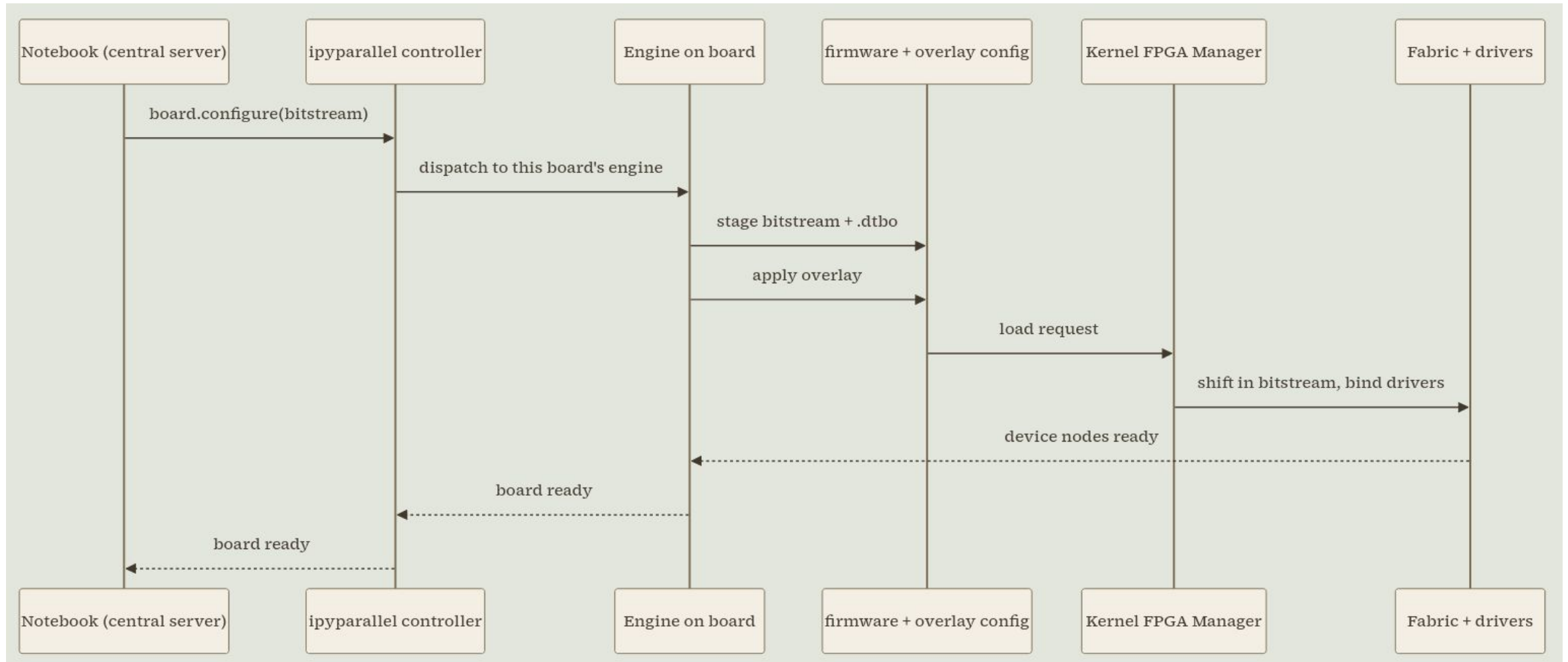
[View User Guide →](#)

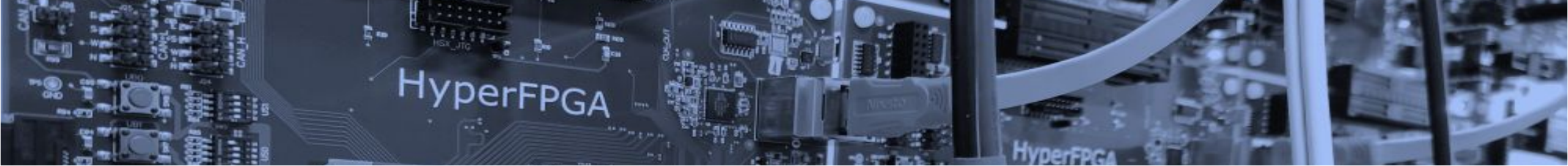
smr4200u**x**

smr4200u**x**

Your Heterogeneous Ecosystem







School and Conference on Edge Intelligence for Scientific Instrumentation and Reconfigurable Computing

Lab 0: User Environment and Access to the HyperFPGA Cluster

Maynor G. Ballina
September 2026



The Abdus Salam
International Centre
for Theoretical Physics



UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

User Dashboard


 The Abdus Salam
**International Centre
for Theoretical Physics**

HyperFPGA Project

[Home](#)
[Projects](#)
[Research](#)
[Resources](#)
[Contact](#)


smr4200 Test
 STUDENT

[Sign out](#)

ALL ROLES

HyperFPGA metrics

Live cluster metrics for every board, no login required — safe to share outside the portal.

PUBLIC, READ-ONLY

View MPSoC metrics

ALL ROLES

JupyterHub

Interactive Python notebooks with dedicated per-user containers. Spawn a session, run experiments, and persist your work across restarts.

USER ACCESS

Open JupyterHub

ALL ROLES

MPSoC Cluster

You can reserve a board before driving its PL from your notebook, and release it when you're done so others can use it. Reservations expire automatically after a few hours.

BOARDS MATCHING YOUR MODEL ENTITLEMENT

No model entitlement yet. Start your JupyterHub server once to generate it, or ask an admin to add you to node_inventory.yml.

ALL ROLES

Remote Desktop

Browser-based SSH and graphical desktop access to the FPGA development VM.

SSH AND RDP SESSIONS

Open Remote Desktop

ALL ROLES

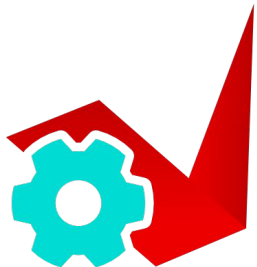
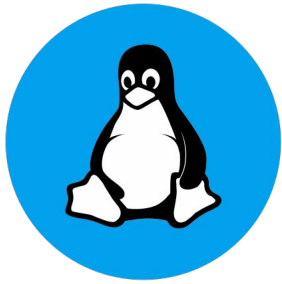
Terms of Use and Conditions

By accessing and using this environment, you acknowledge and accept full responsibility for any misuse of the system, including but not limited to improper handling or manipulation of the hardware and software. You agree that any damage, malfunction, or degradation of the system due to your actions will be your sole responsibility.

You further agree to indemnify and hold

Virtual Machine

User: student
Password: workspace2026



Important Links

<https://gitlab.com/ictp-mlab/smr-4200>

ICTP-MLAB / smr4200

README.md Updated readme links 6 days ago

README.md Edit file



School and Conference on Edge Intelligence for Scientific Instrumentation and Reconfigurable Computing

This school and conference is dedicated to teaching key principles of edge intelligence on FPGA-based heterogeneous devices, focusing on real-time processing near data sources and applications in scientific instrumentation. (SMR4200)

Indico event. Workshop Wiki.




Activities

Source code for lab activities:

- Lab 0: User Environment and Access to the HyperFPGA Cluster
- Lab 1: Getting Started With MPSoC-FPGA and Comblock

<https://gitlab.com/ictp-mlab/smr-4200/-/wikis/home>

ICTP-MLAB / smr4200 / Wiki / Home

« Wiki Pages »

Home

Last edited by Luis Garcia 1 week ago



School and Conference on Edge Intelligence for Scientific Instrumentation and Reconfigurable Computing

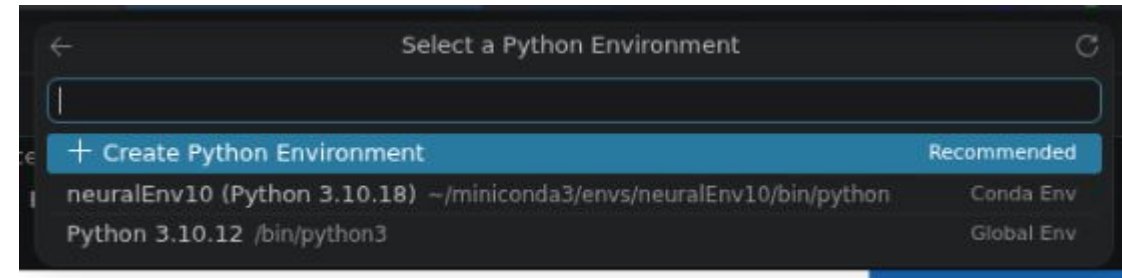
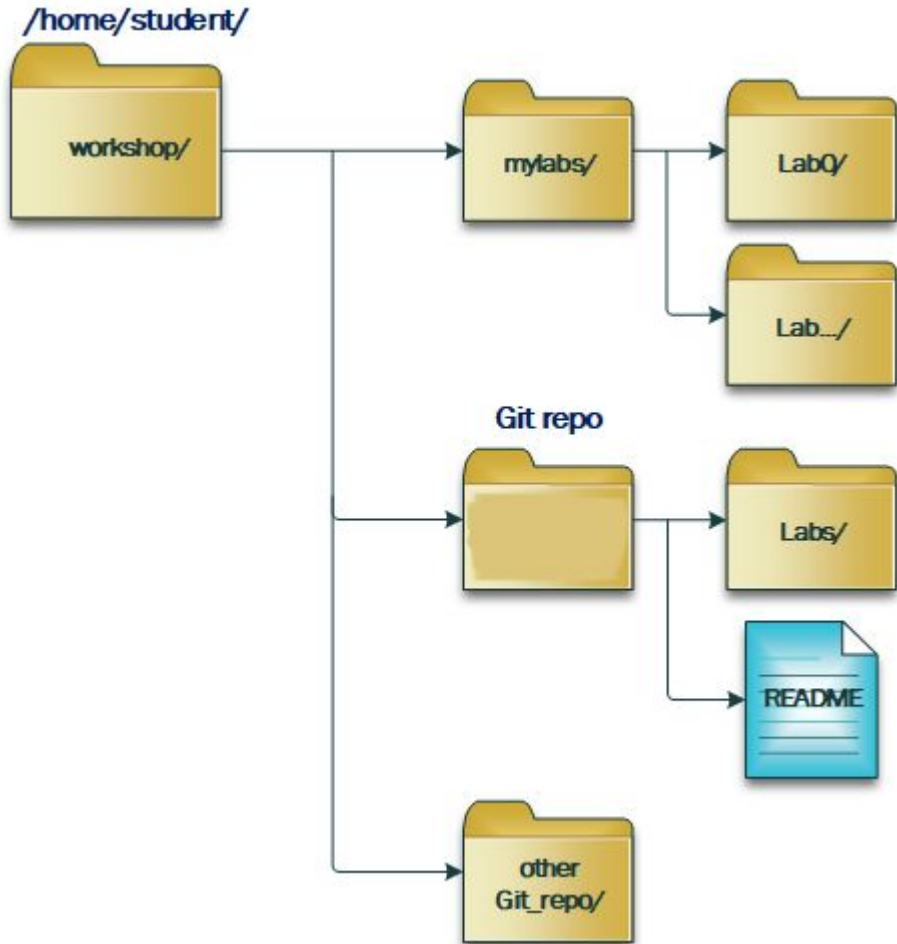
Lab Guides

- Lab 0: User Environment and Access to the HyperFPGA Cluster
- Lab 1: Getting Started With MPSoC-FPGA and Comblock
- Lab 2: Reconfigurable Instrumentation
- Lab 3: High-Level Synthesis for Heterogeneous Computing
- Lab 4: Training and Compression of Machine Learning Models
- Lab 5: End-to-End Workflow for Deploying Machine Learning Models on MPSoC-FPGA
- Final Project:





Objectives



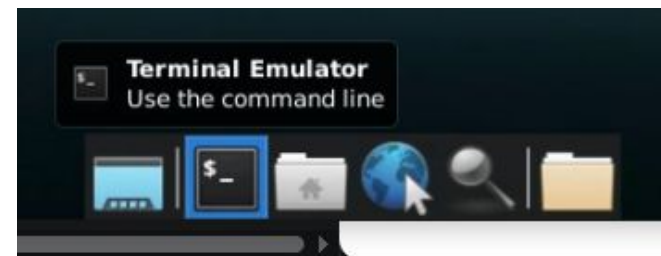
ALL ROLES

MPSoc Cluster

You can reserve a board before driving its PL from your notebook, and release it when you're done so others can use it. Reservations expire automatically after a few hours.

BOARDS MATCHING YOUR MODEL ENTITLEMENT

Auto-allocate a free 4ge21



Development Workflow - Vivado

Default Part

Choose a default AMD part or board for your project.

Parts | **Boards**

To fetch the latest available boards from git repository, click on 'Refresh' button. [Dismiss](#)

[Reset All Filters](#)

Vendor:
Name:
Board Rev:

Display Name	Preview	Status	Vendor	File Version	Part	I/O Pin Count	Board
HyperFPGA 3be11		Installed	ictp.it	1.0	xczu3eg-sfvc784-1-e	784	1.0
HyperFPGA 4ge21		Installed	ictp.it	1.0	xczu4eg-sfvc784-2-e	784	1.0

Export Hardware Platform

Files

Enter the name of your hardware platform file, and the directory where the XSA file will be stored.

-4ge21

XSA file name:

Export Hardware Platform

Files

Enter the name of your hardware platform file, and the directory where the XSA file will be stored.

-3be11

XSA file name:

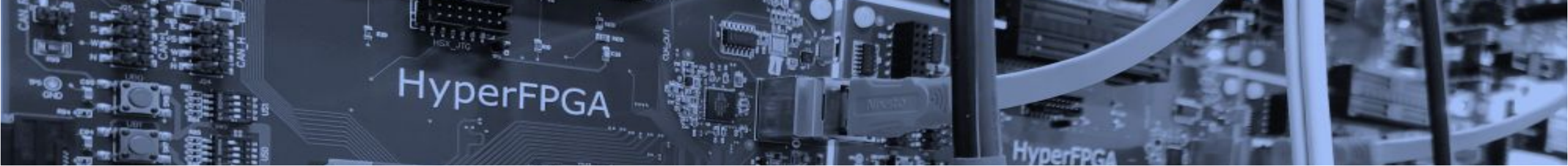
Workflow Overview: Connect, Program, Interact, Deprogram

Every session with a HyperFPGA node follows the same four stages, regardless of what your project does. This tutorial walks through each stage, one cell at a time:

1. **Connect:** `hfc.available_models()` asks hyperfpga-allocator which node(s) are available to you. `request_nodes()` then requests a node and assigns it to you for a set period of time. Finally, `HyperFPGACluster(...)` and `cluster.create_profile()` prepare the local `ipyparallel/SSH` profile needed to reach the node.
2. **Program:** `await cluster.configure()` reserves the node, uploads your bitstream (converting `.xsa` to `.bit.bin/.dtbo` via `xsa2bin-builder` if needed), and programs the FPGA with it. This step also prepares the board's CPU-side `ipyparallel` engine to run your code on the CPU or GPU, an `.xsa` file is not required to configure a node.
3. **Interact:** A Python kernel runs plain Python code on the board's ARM CPU cores.
4. **Deprogram:** When you're done, free the FPGA and release the node reservation using `clean_cluster()` and `release_cluster()`, so another user can use the node.

```
import hyperfpga_cluster as hfc
print(hfc.available_models())
test_node = await hfc.request_nodes('type') #4ge21, 3be11
cluster = hfc.HyperFPGACluster(nodes=test_node)
cluster.create_profile()
await cluster.configure()
```

```
await cluster.clean_cluster()
await cluster.release_cluster()
```



School and Conference on Edge Intelligence for Scientific Instrumentation and Reconfigurable Computing

Lab 1: Getting Started with MPSoC-FPGA and Comblock

Maynor G. Ballina
September 2026



The Abdus Salam
International Centre
for Theoretical Physics

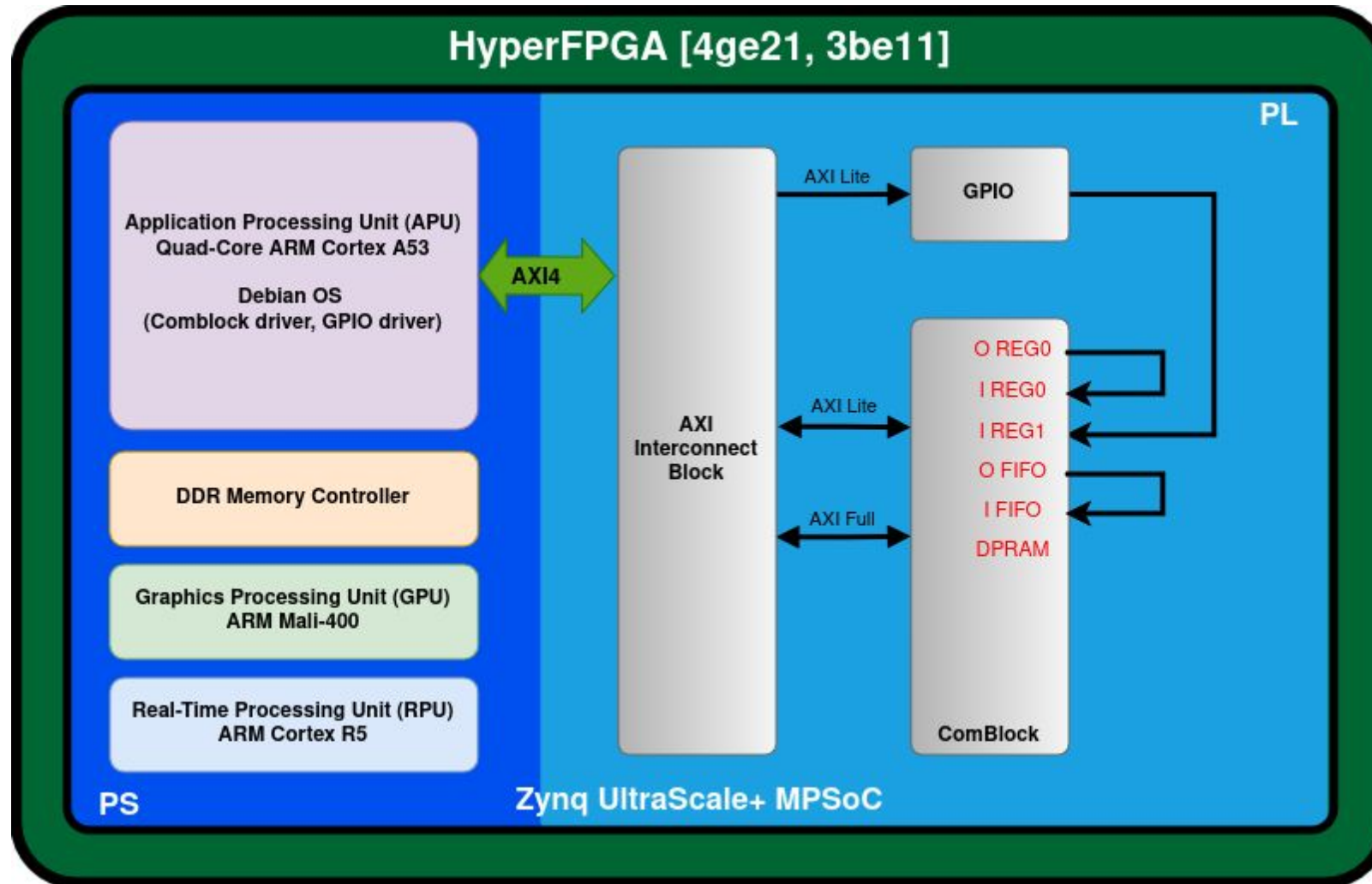


UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Objectives

- Acquire knowledge of the MPSoC-FPGA design flow using the Vitis Unified Software Platform.
- Create the hardware to configure the Processing System (PS) and the Programmable Logic (PL) of the MPSoC, and understand the communication between the different components.
- Understand the usage and configuration of a HyperFPGA node.
- Develop a 'Python' application (in Jupyter Notebook) that will run on the PS.
- Test the complete design on the HyperFPGA platform to verify functionality using Jupyter Hub.

Design description



Development Workflow - Vivado

Default Part

Choose a default AMD part or board for your project.

Parts | **Boards**

To fetch the latest available boards from git repository, click on 'Refresh' button. [Dismiss](#)

[Reset All Filters](#)

Vendor:
Name:
Board Rev:

Search:

Display Name	Preview	Status	Vendor	File Version	Part	I/O Pin Count	Board
HyperFPGA 3be11		Installed	ictp.it	1.0	xczu3eg-sfvc784-1-e	784	1.0
HyperFPGA 4ge21		Installed	ictp.it	1.0	xczu4eg-sfvc784-2-e	784	1.0

Export Hardware Platform

Files

Enter the name of your hardware platform file, and the directory where the XSA file will be stored.

-4ge21

XSA file name:

Export Hardware Platform

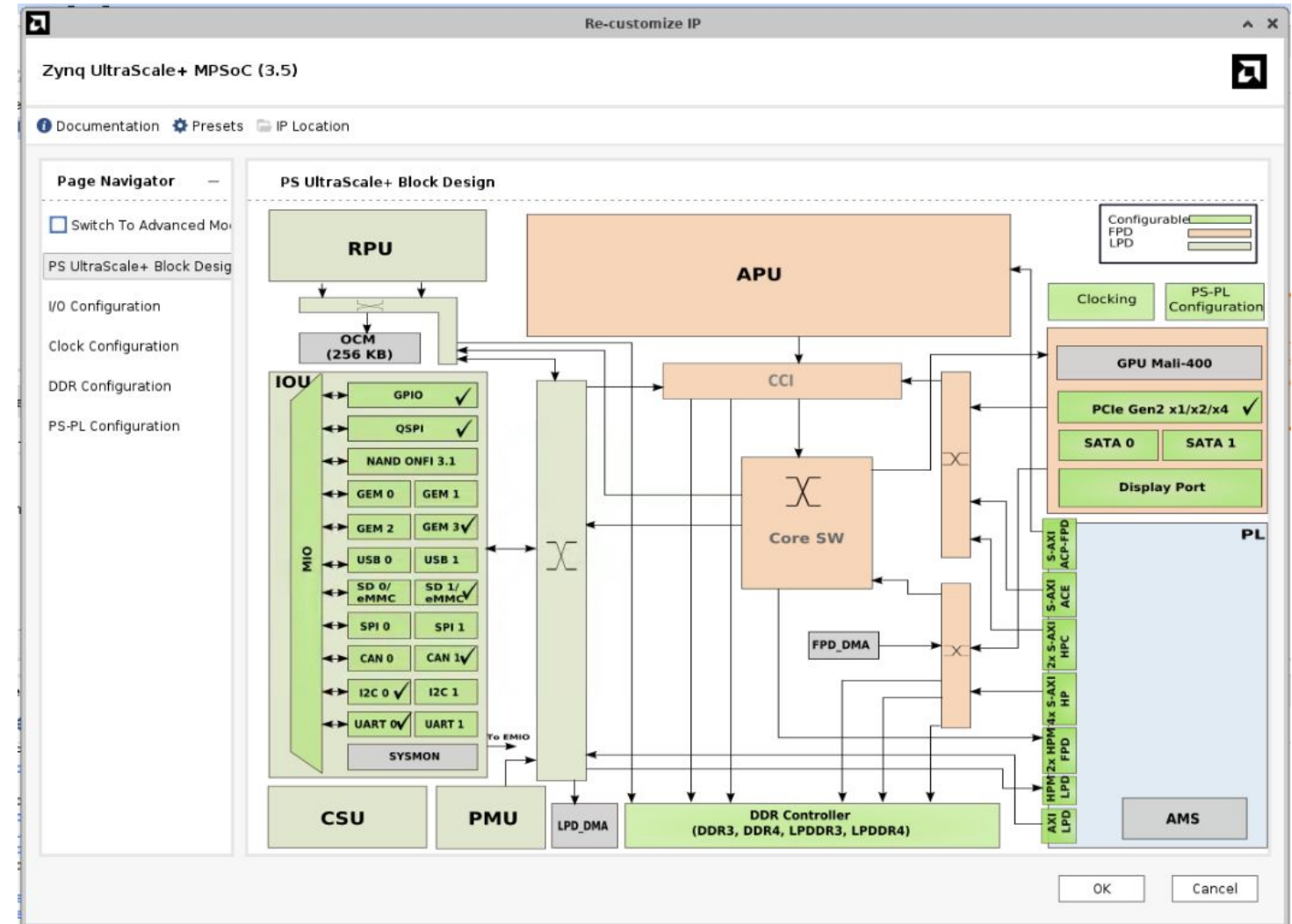
Files

Enter the name of your hardware platform file, and the directory where the XSA file will be stored.

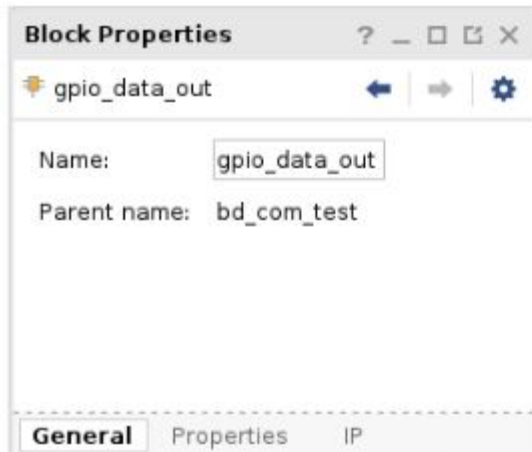
-3be11

XSA file name:

Hardware

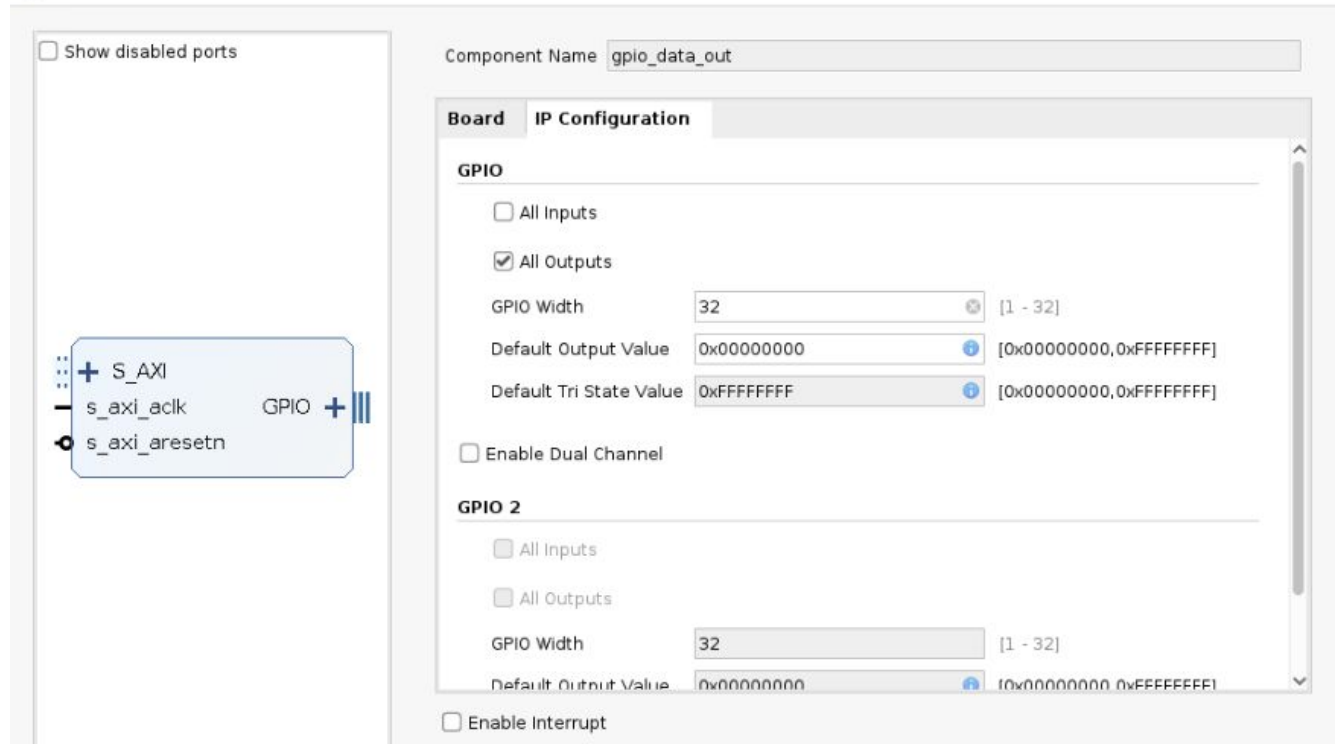


Steps



AXI GPIO (2.0)

Documentation IP Location



Steps

I/O Registers

- 32 Bits

I/O FIFO

- 32 Bits
- 1024 Depth

RAM

- 32 bits
- 16 Address width

Re-customize IP

ComBlock (2.0)

Documentation IP Location

☐ Show disabled ports

+ AXIL
+ AXIF
+ IN_REGS
+ IO_DRAM
+ IN_FIFO
ram_clk_i
fifo_clk_i
fifo_clear_i
axil_ack
axil_aresetn
axif_ack
axif_aresetn

OUT_REGS
OUT_FIFO

Component Name: comblock_0

Registers

Input (FPGA to PROC)	Output (PROC to FPGA)
☒ Enable	☒ Enable
Data Width: 32 [1 - 32]	Data Width: 32 [1 - 32]
Quantity: 4 [1 - 16]	Quantity: 4 [1 - 16]

True Dual Port RAM (FPGA to PROC, PROC to FPGA)

☒ Enable

Data Width: 32 [1 - 32]

Address Width: 16 [1 - 32]

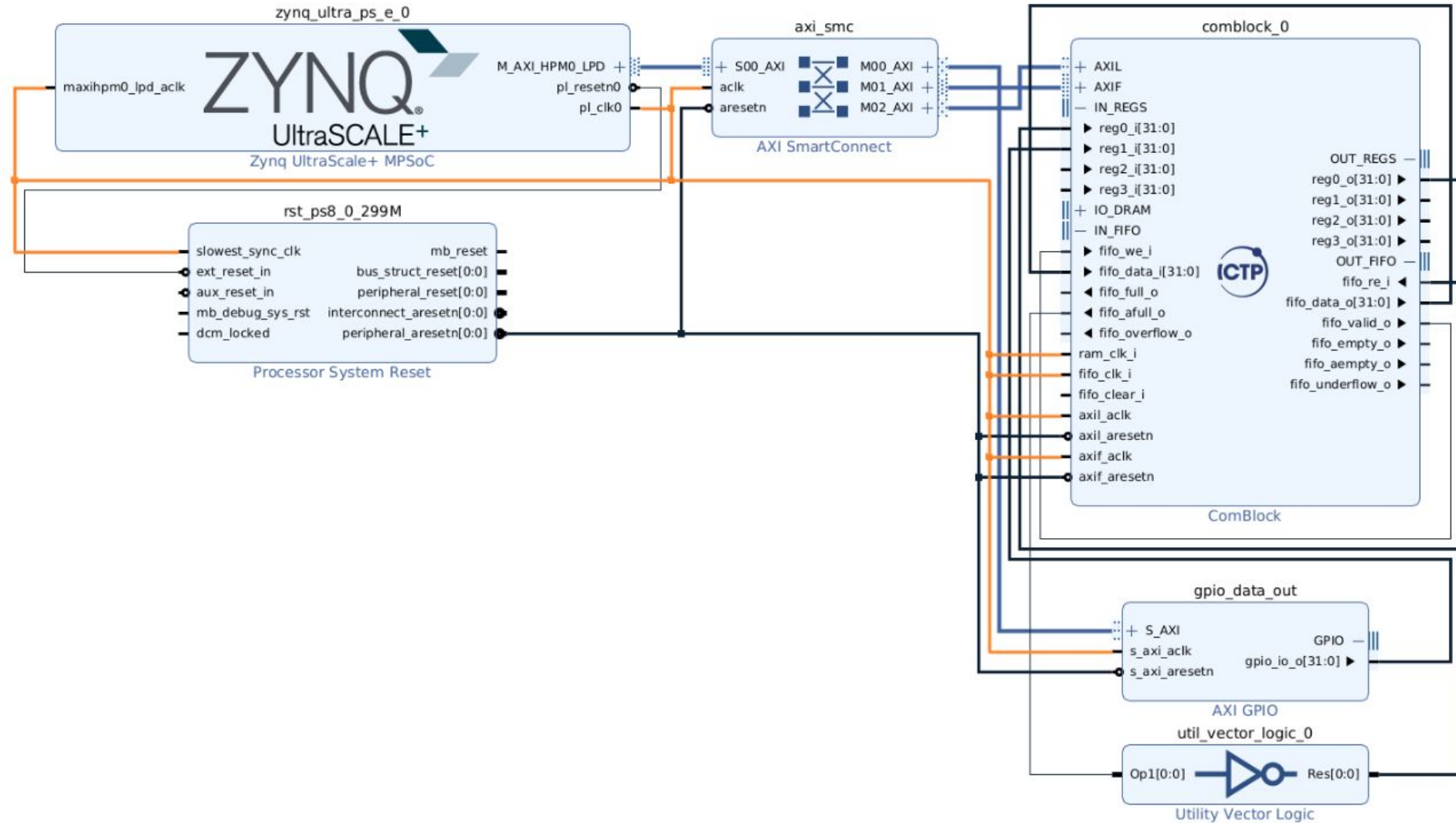
Depth: 0 [0 - 67108863]

FIFOs

Input (FPGA to PROC)	Output (PROC to FPGA)
☒ Enable	☒ Enable
Data Width: 32 [1 - 32]	Data Width: 32 [1 - 32]
Depth: 1024	Depth: 1024
Almost Full Offset: 1	Almost Full Offset: 1
Almost Empty Offset: 1	Almost Empty Offset: 1

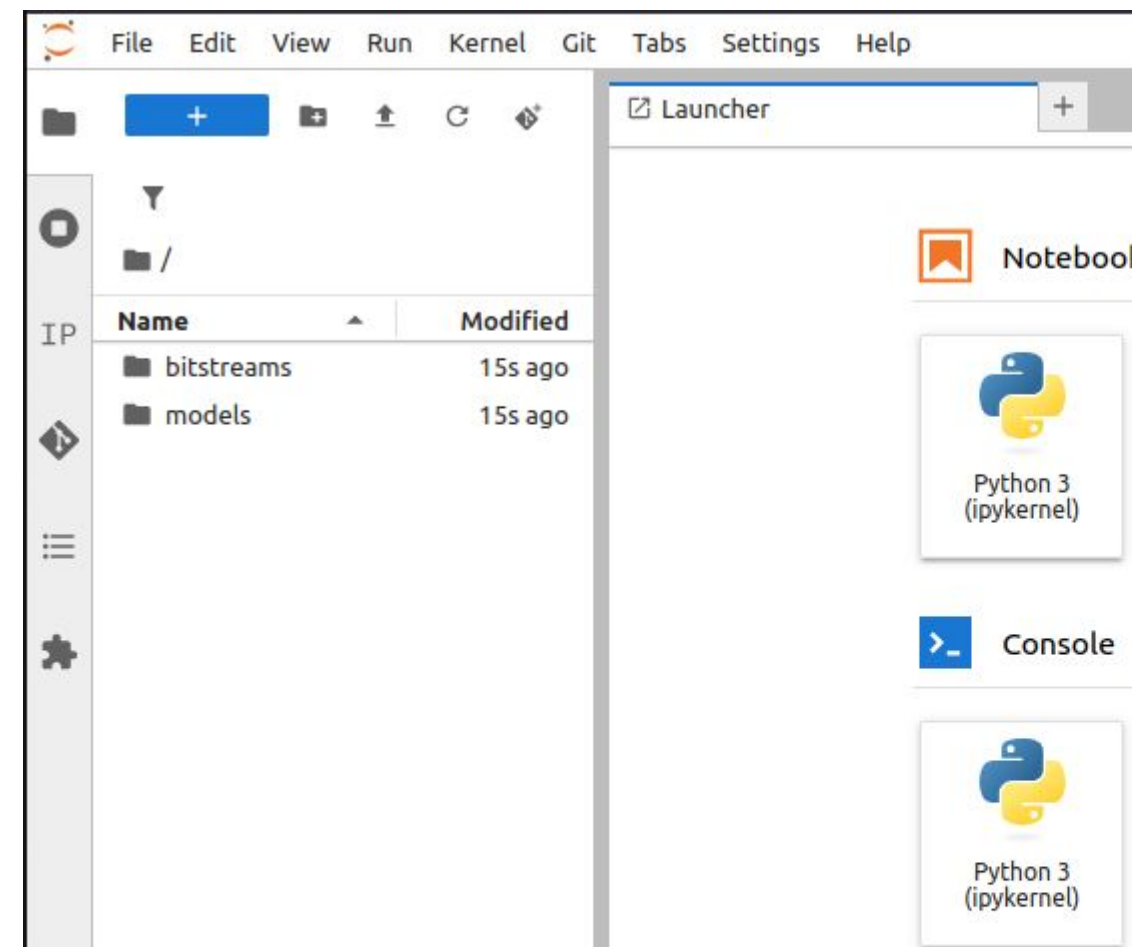
OK Cancel

Final Block Design



Jupyter Hub

- Upload .xsa file to bitstreams folder
- Create smr4200 folder
- Upload the All the files inside the Jupyterhub folder from the git repo
- Create your code based on lab0
- Create your interaction with the hardware
- Validate your design



Thank you

Q&A

mballina@ictp.it