HyperFPGA: Heterogeneous Computing on MPSoC-FPGA

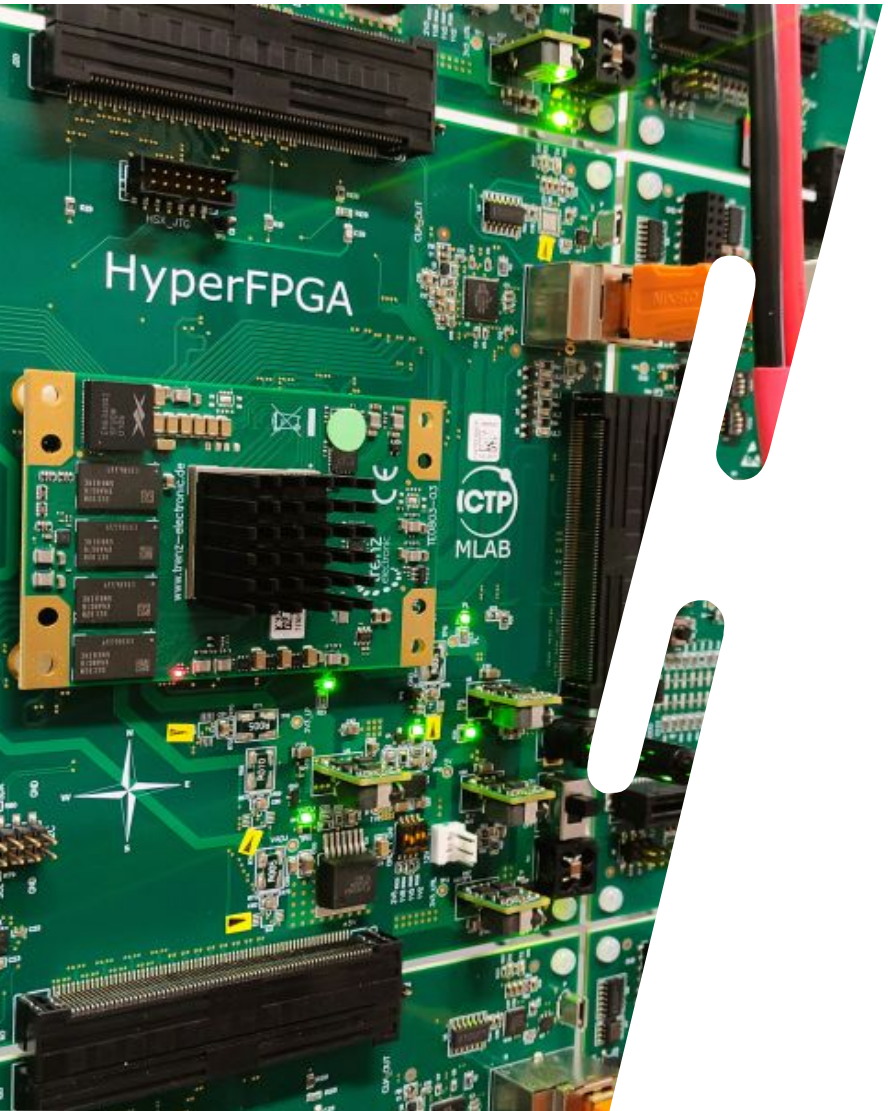
School on Applied AI for Sustainable Development
smr 4210, Trieste 25 March 2026

Maynor Ballina

The Abdus Salam International Centre for Theoretical Physics

UNIVERSITÀ DEGLI STUDI DI TRIESTE

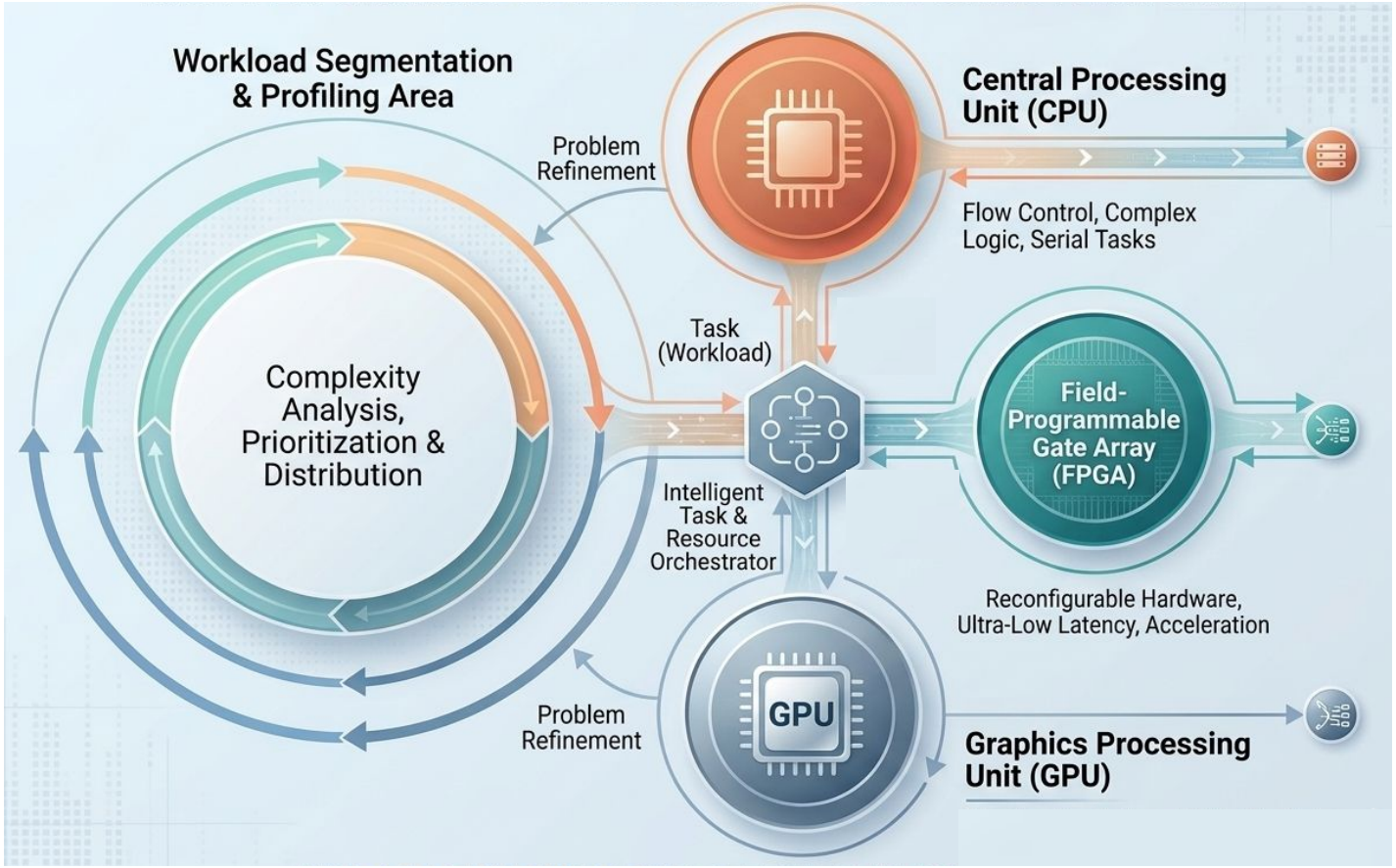
Outline



- Heterogeneous computing
- Evolution of heterogeneous computing
- Distributed Computing
- Flynn Taxonomy
- Application of Heterogeneous Computing in Simulation
- The HyperFPGA cluster
- Infrastructure and management
- User Interface
- Jupyter Notebooks
- Broad spectrum of computational tasks
- Education and collaborative projects

Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

Heterogeneous Computing

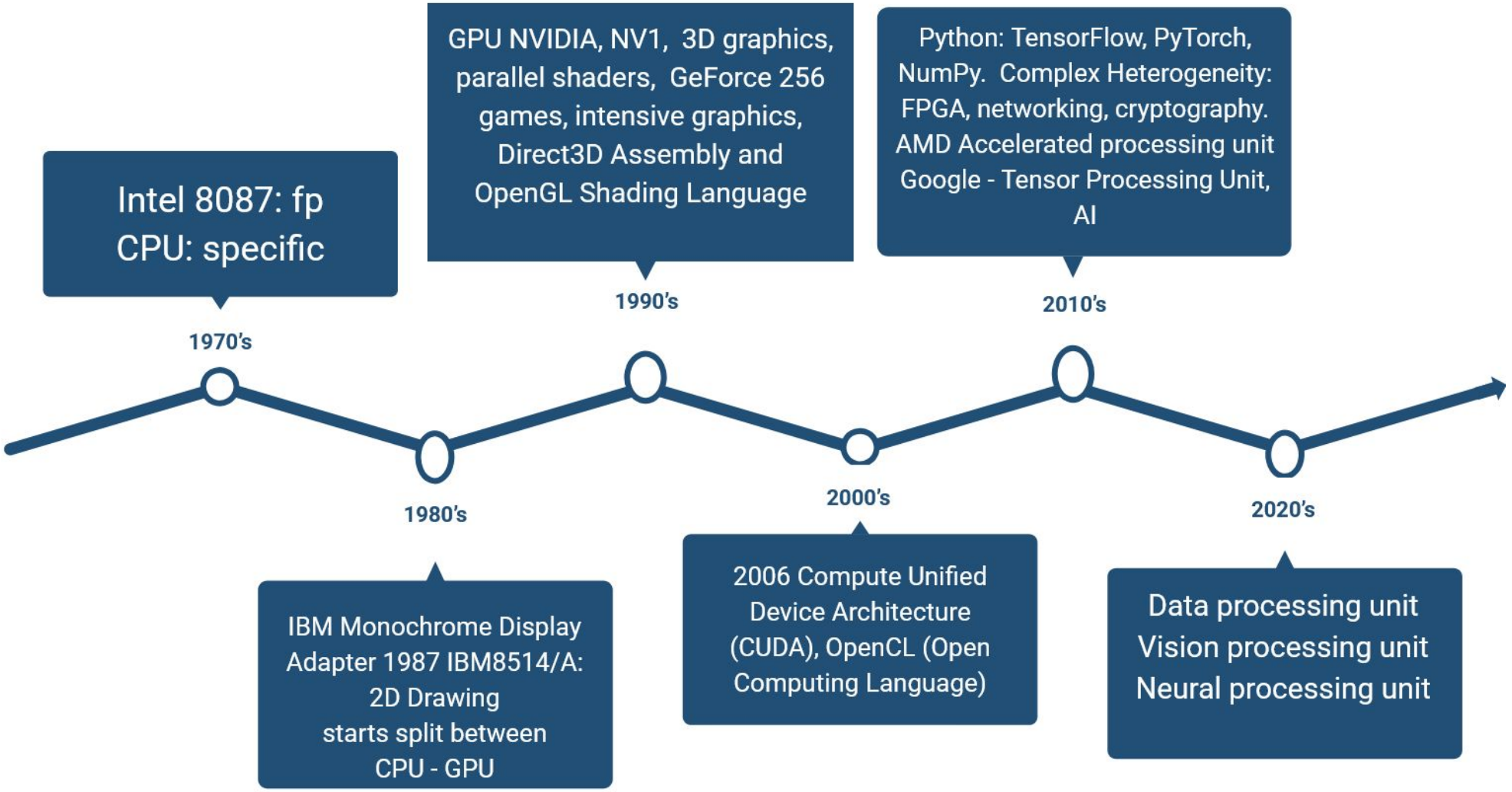


It enhances computational performance by accelerating workloads such as graphics processing, artificial intelligence, and scientific simulations through specialized hardware.

The coordinated use of different types of processing cores within a single system to maximize performance and energy efficiency.

Maynor Ballina

Evolution of Heterogeneous Computing



Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

Distributed Computing

A model in which multiple nodes work together in a coordinated manner to solve a problem, sharing resources to improve performance, scalability, and fault tolerance.

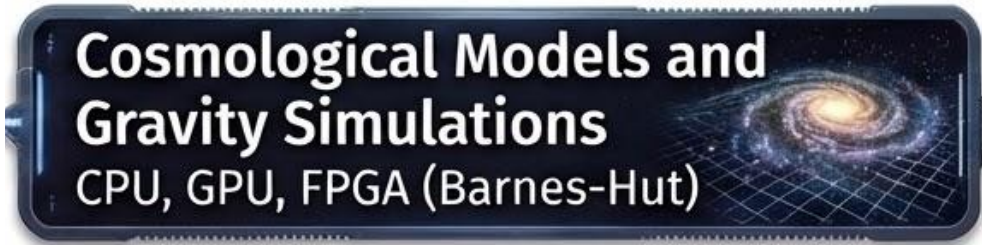
Distributed computing in clusters is a data processing approach that uses a set of interconnected nodes to work together as if they were a single machine.

Maynor Ballina

UNIVERSITÀ DEGLI STUDI DI TRIESTE

The Abdus Salam International Centre for Theoretical Physics

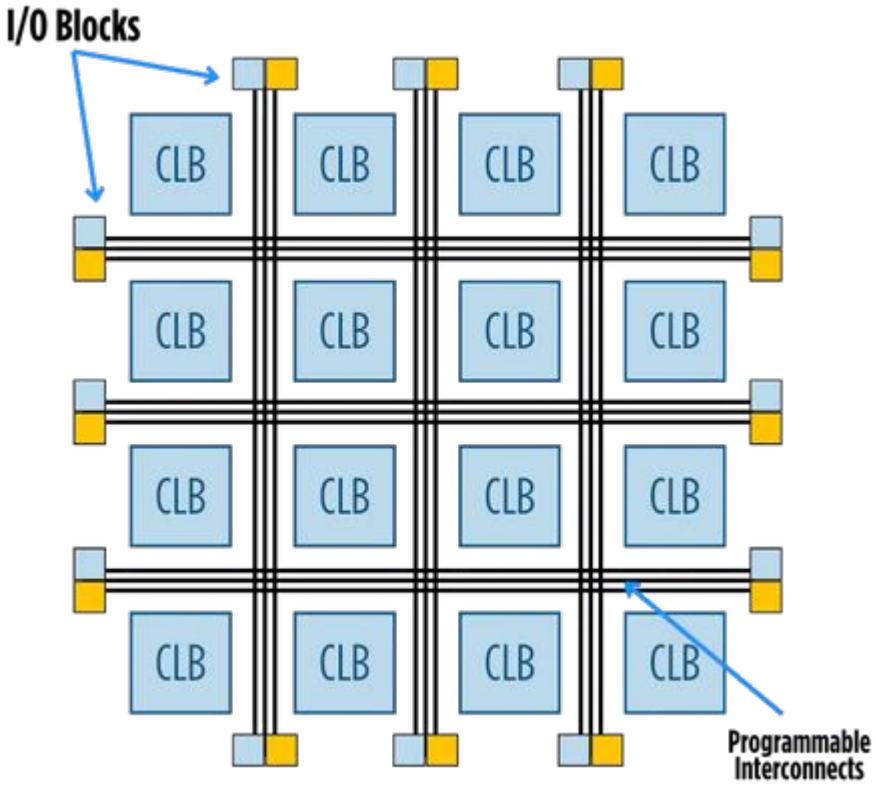
Application of Heterogeneous Computing in Simulation



Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

What is a FPGA

An FPGA (Field-Programmable Gate Array) is an integrated circuit that can be programmed after manufacturing



Allows designers to configure hardware behavior instead of just running software

Made up of logic blocks, configurable interconnects, and I/O blocks

Common Uses:

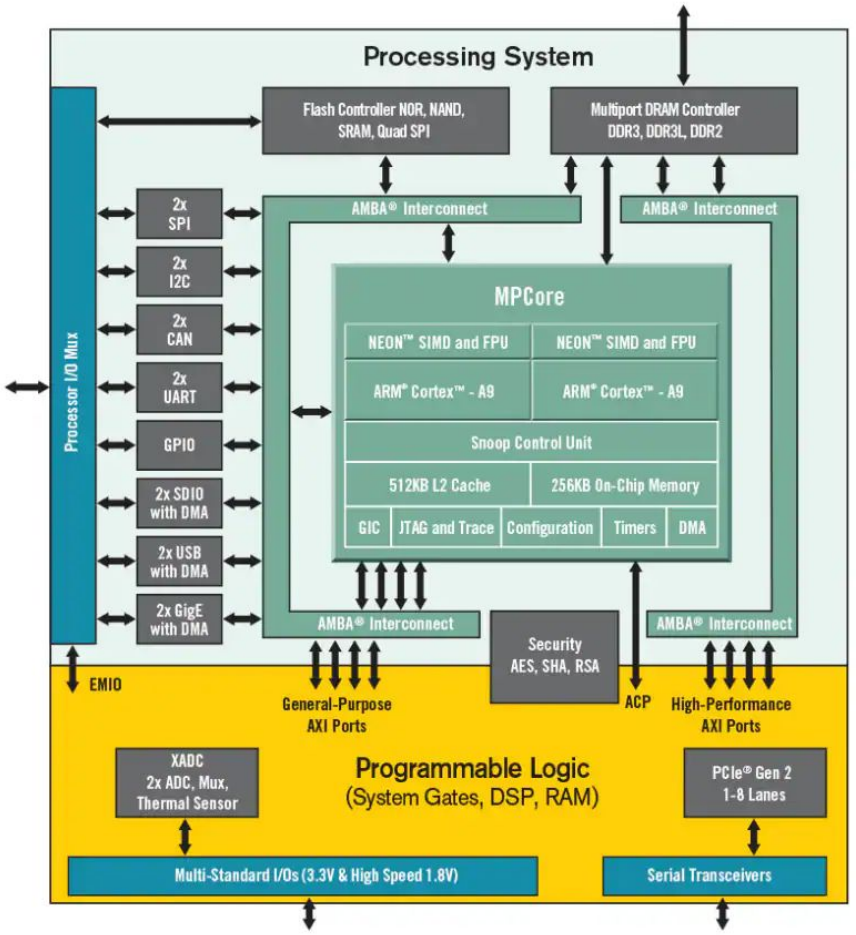
- Digital signal processing (DSP)
- Embedded systems
- Hardware prototyping
- AI and high-performance computing

Key Advantage:

High flexibility and parallel processing

What is a SoC-FPGA

An System on Chip FPGA (SoC FPGA) combines, a traditional FPGA fabric and a processor (CPU) on the same chip.



Tight integration between hardware (FPGA) and software (CPU).

Faster communication compared to separate components.

Lower power consumption and smaller size.

Typical Components:

- ARM processor
- Memory controllers
- Peripherals (USB, Ethernet, etc.)

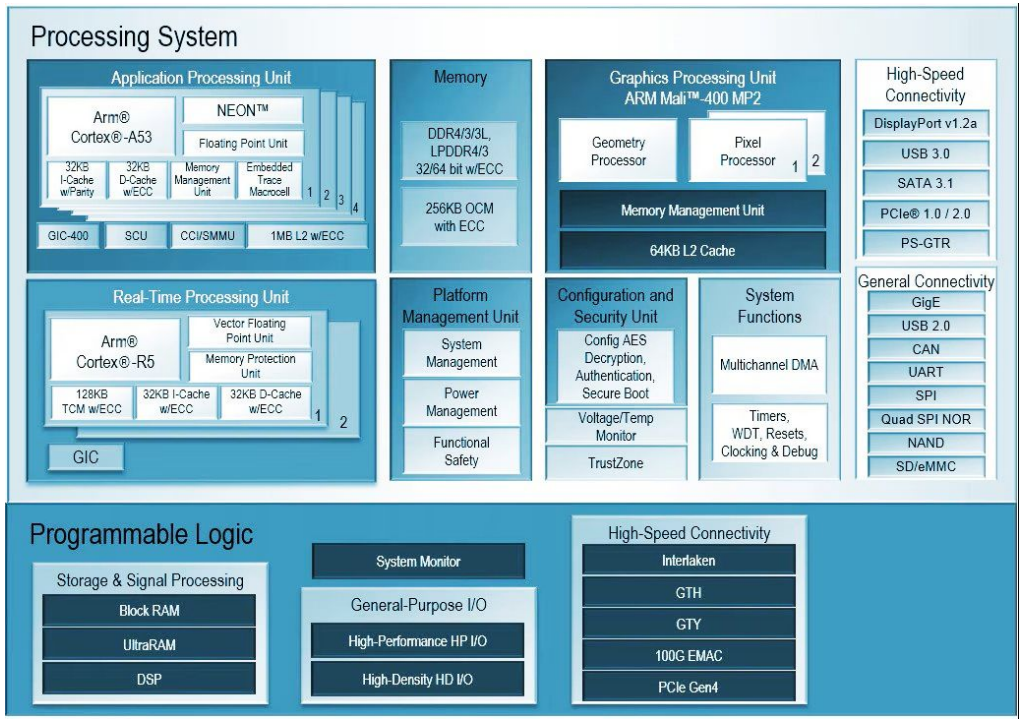
Use Cases:

- Embedded systems
- Real-time processing
- Industrial and automotive applications

Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

What is a MPSoC-FPGA

An Multi-Processor System-on-Chip (MPSoC FPGA) is an advanced version of SoC FPGA with a Multiple processors (multi-core CPUs) and a FPGA programmable logic.



Combines multi-core processing + programmable hardware

Supports high-performance computing tasks

Can include CPUs, GPUs, and specialized accelerators

Advantages:

- Handles complex workloads
- Enables hardware/software co-design at scale
- Optimized for performance, power, and flexibility

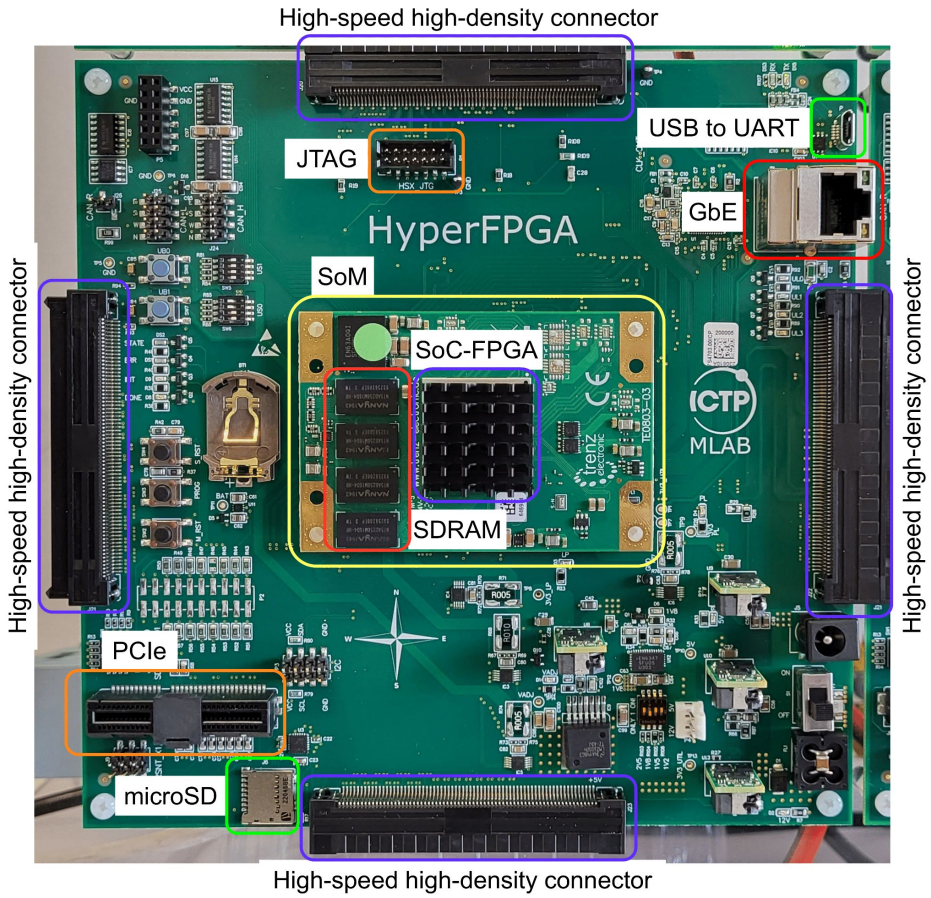
Use Cases:

- AI and machine learning
- Autonomous systems
- 5G and advanced communications

Maynor Ballina

Hyper-FPGA node

A System on Module (SoM) is a small, ready-to-use embedded computer



Typical Components:

- Processor (CPU / FPGA / MPSoC)
- Memory (RAM, Flash)
- Power management
- Communication interfaces

System-on-Module (Zynq UltraScale+ MPSoC-FPGA)

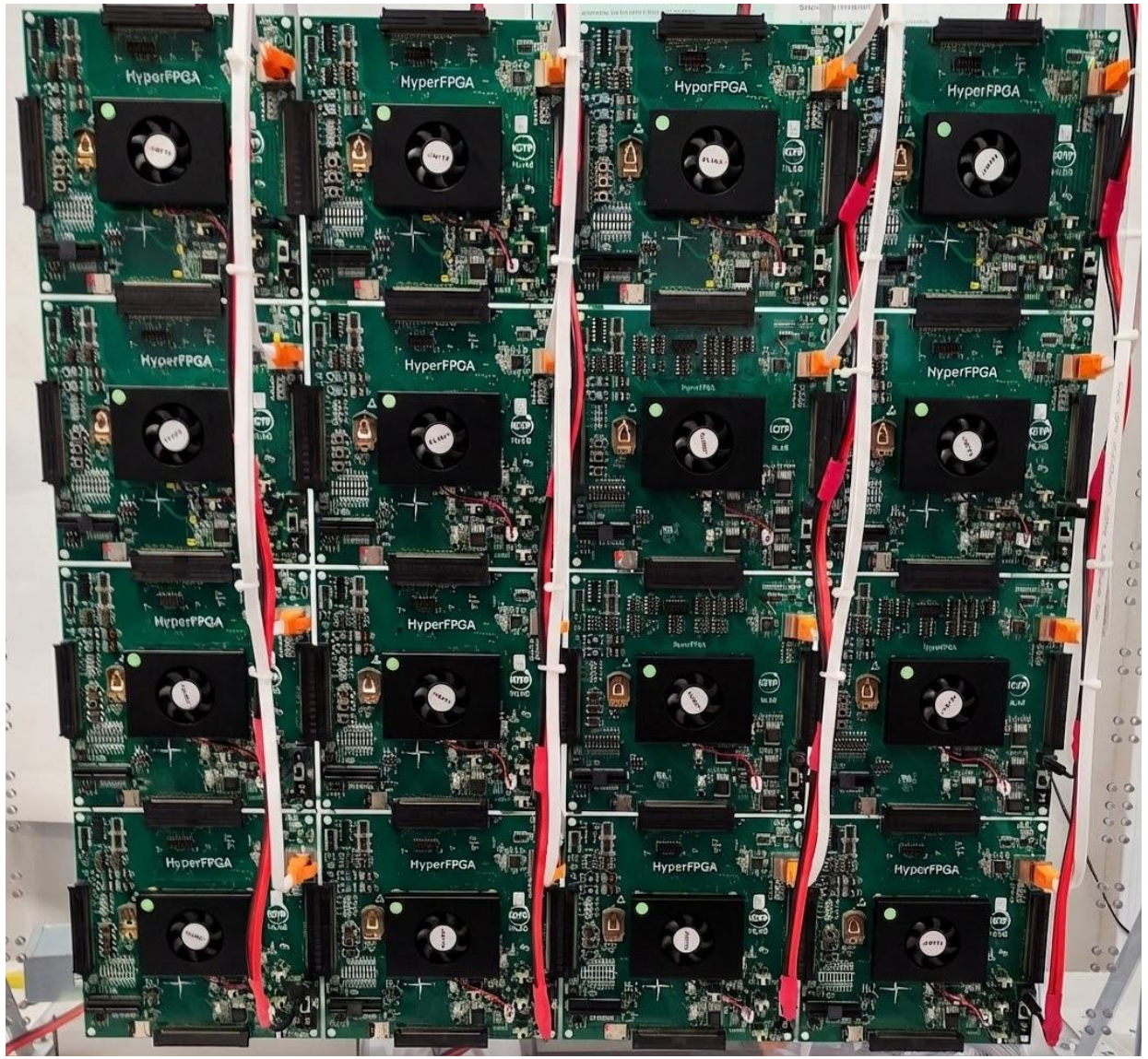
- CPU: Quad Arm Cortex, Dual Arm Cortex
- GPU: ARM Mali 400 MP2
- FPGA: ZU4EG
- 2 or 4 GB RAM

Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

The HyperFPGA Cluster

MPSoC-FPGA-based experimental cluster for scientific computing.

- 16 Nodes
- Zynq UltraScale+ MPSoC with APU, RPU, GPU.
- Flexible Network, HP, HD, GTH, MIO, Ethernet.
- Debian OS
- Power Monitoring



Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

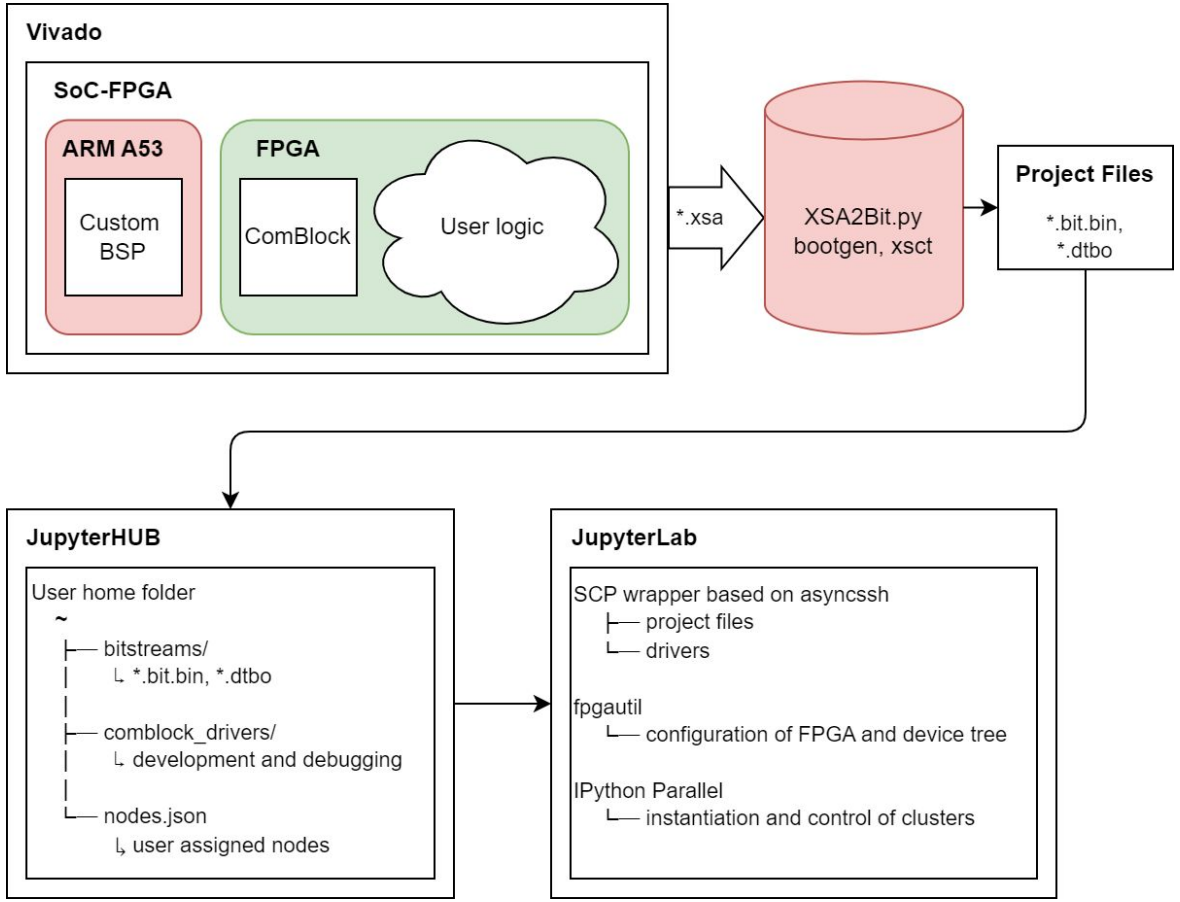
Development Workflow

Hardware definition: Vivado
Board support package
ComBlock as abstraction layer between CPU and user logic

Middleware: XSA2Bit
Generate device tree overlay from an XSA and compile sources

Management: JupyterHub
Authentication
Organized storage

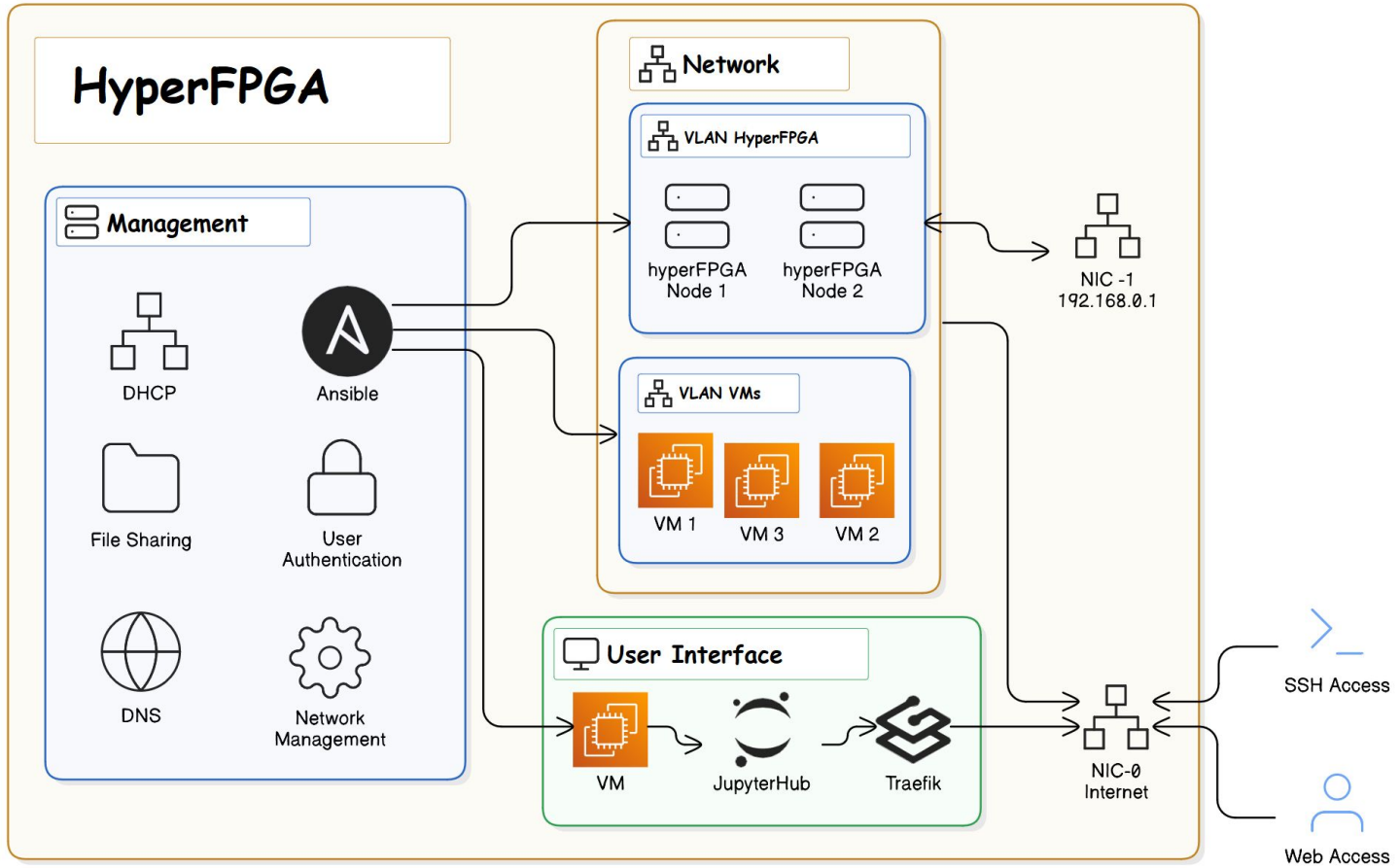
Programming: JupyterLab
User programmable context



Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

The HyperFPGA Cluster Management

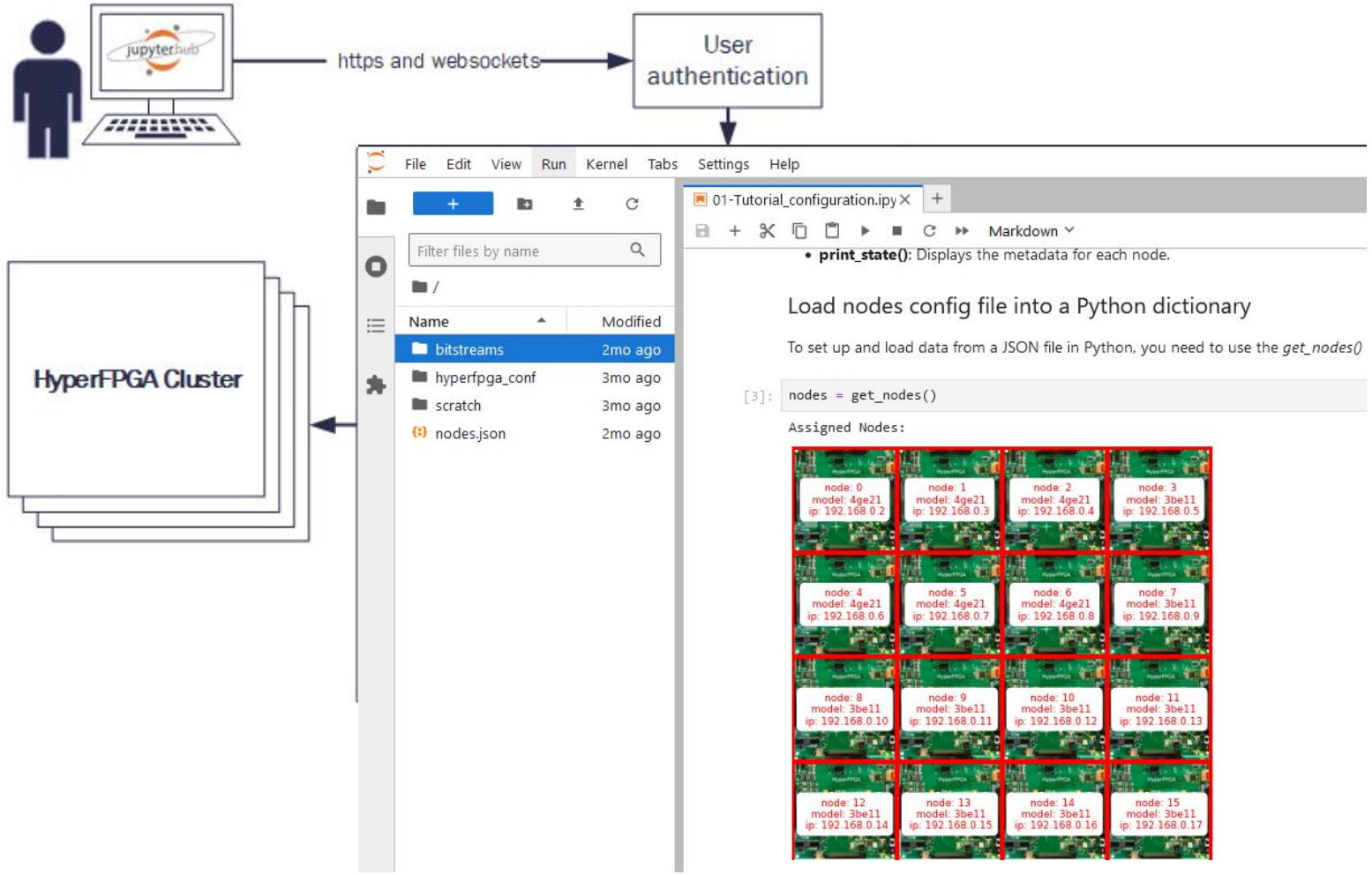
- Infrastructure as Code (IaC).
- JupyterHub & Microservices for multi-user access.



Maynor Ballina

Remote Access to the Cluster

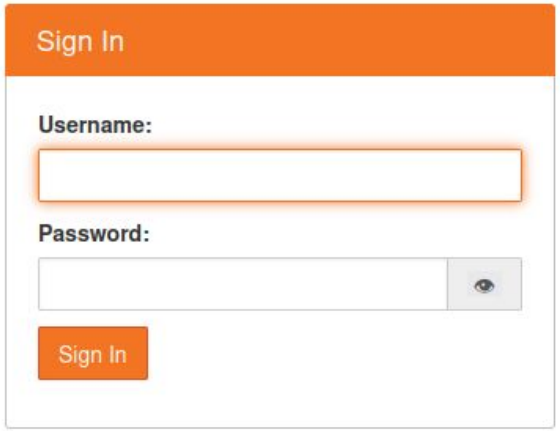
Practical experience with heterogeneous computing, facilitated by Jupyter Hub and Python programming.

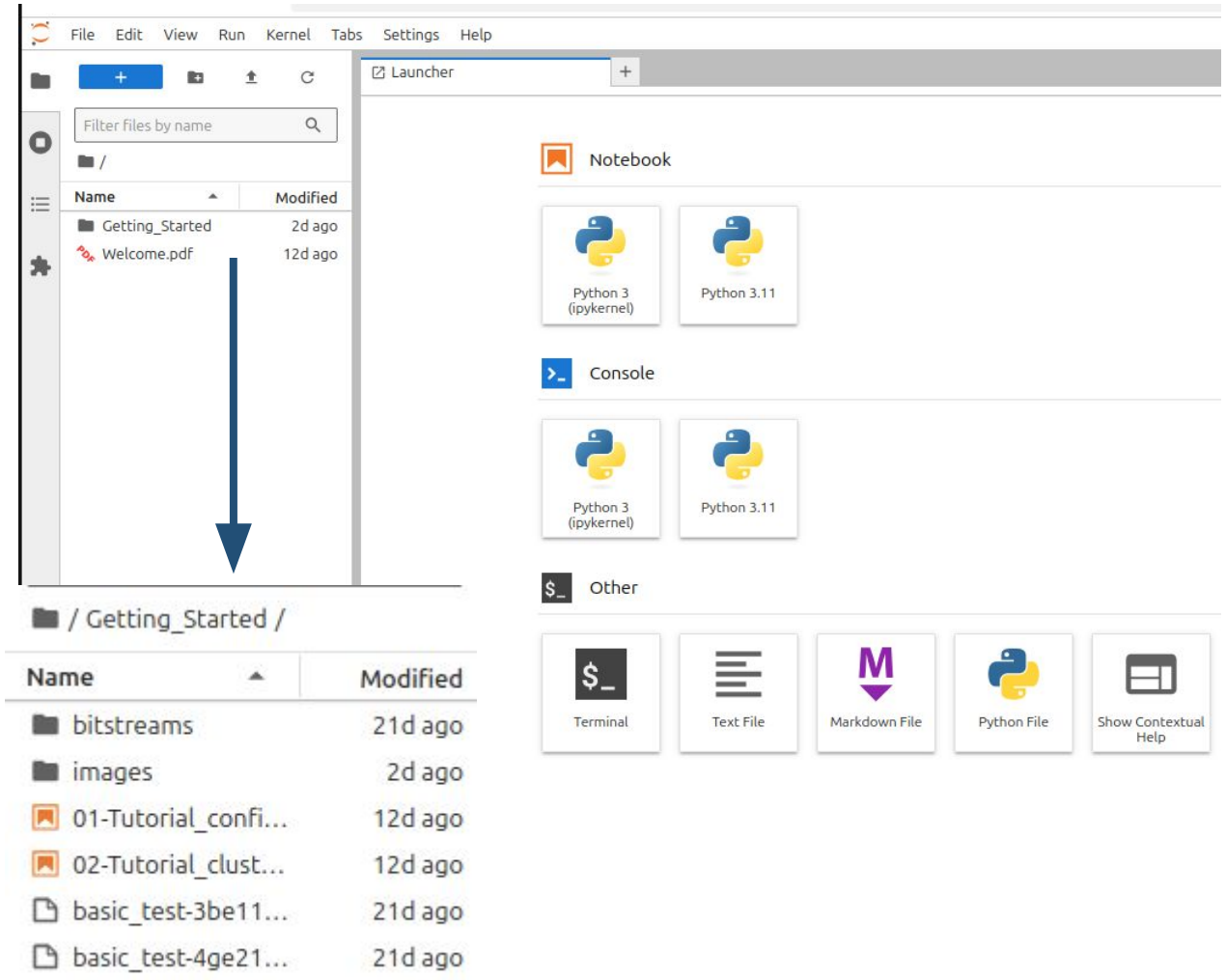


Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

Remote Access to the Cluster

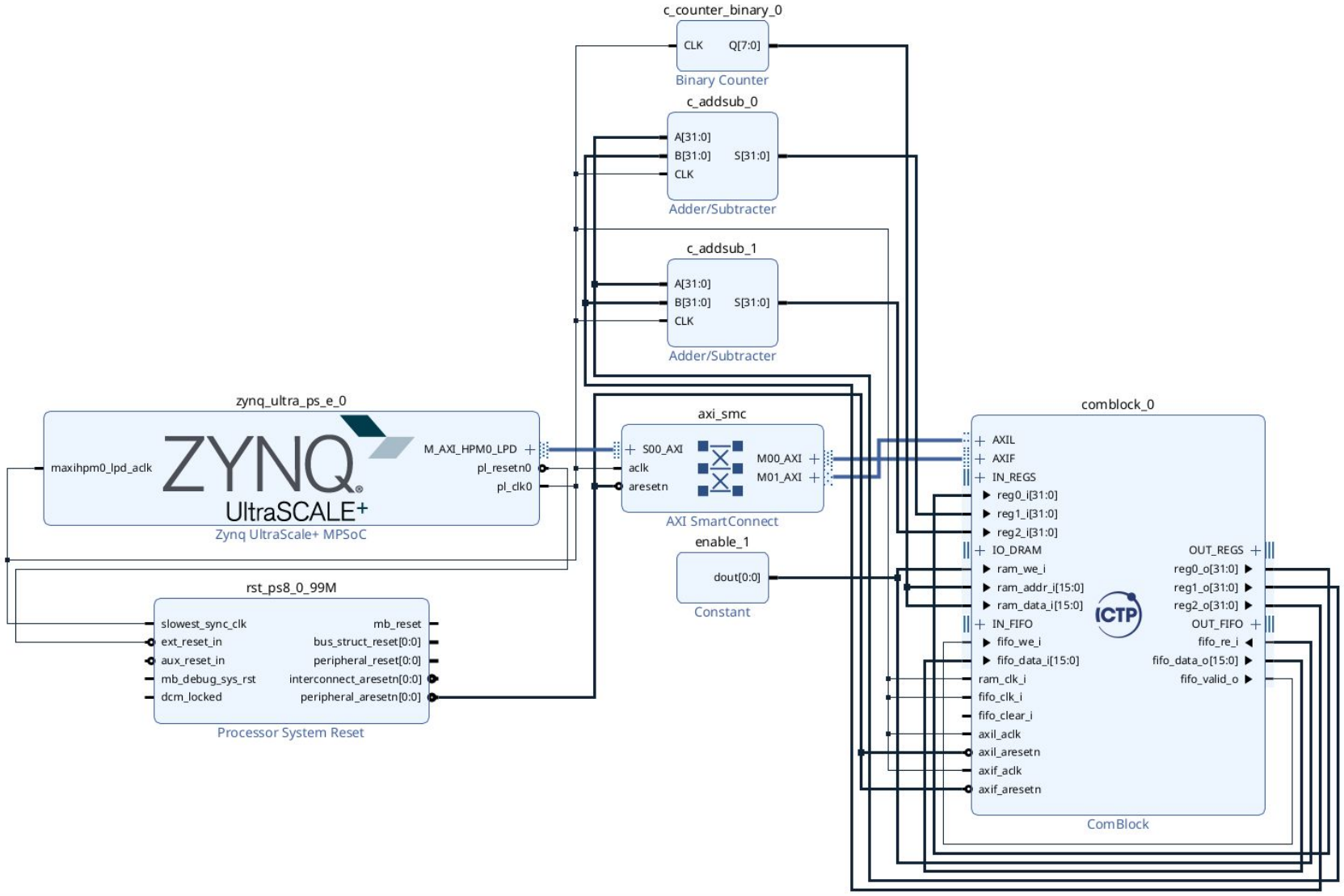
https://hyperfpga.sti.ictp.it





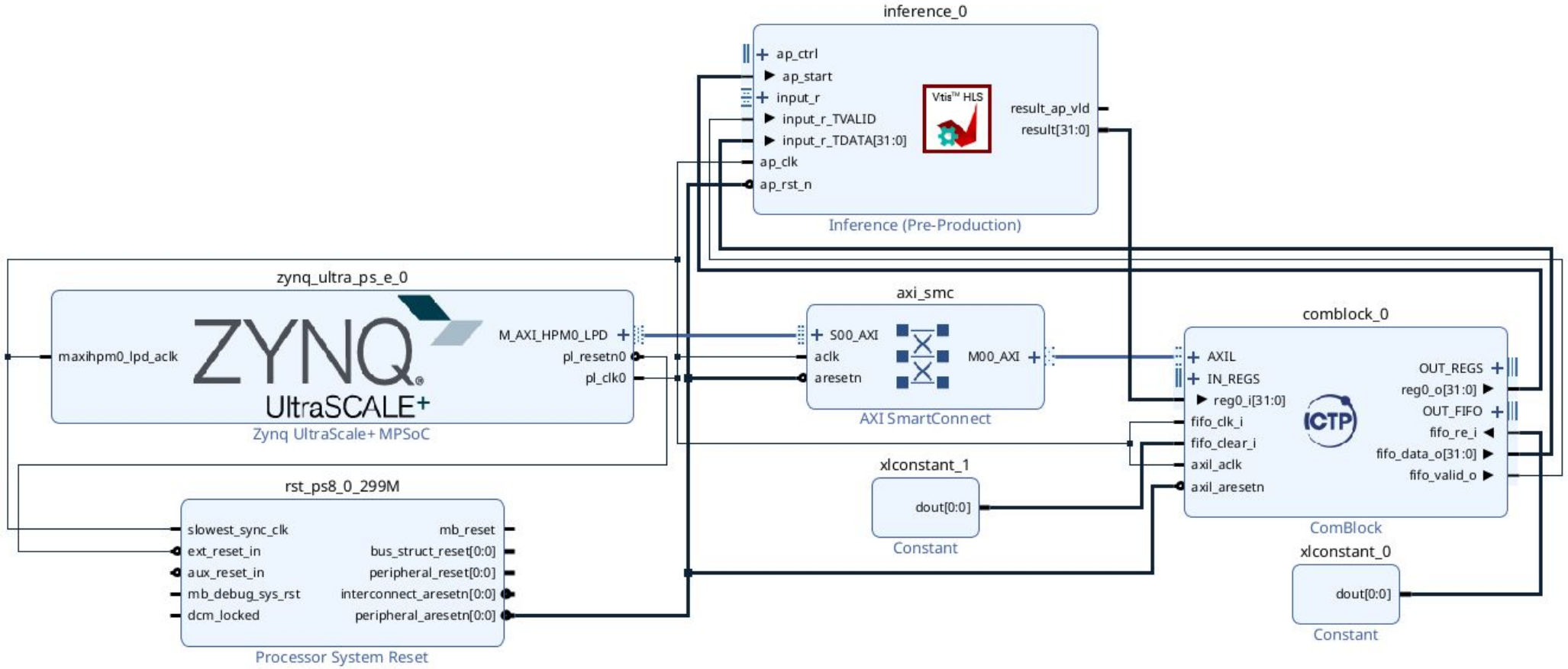
Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

Development Workflow - Vivado



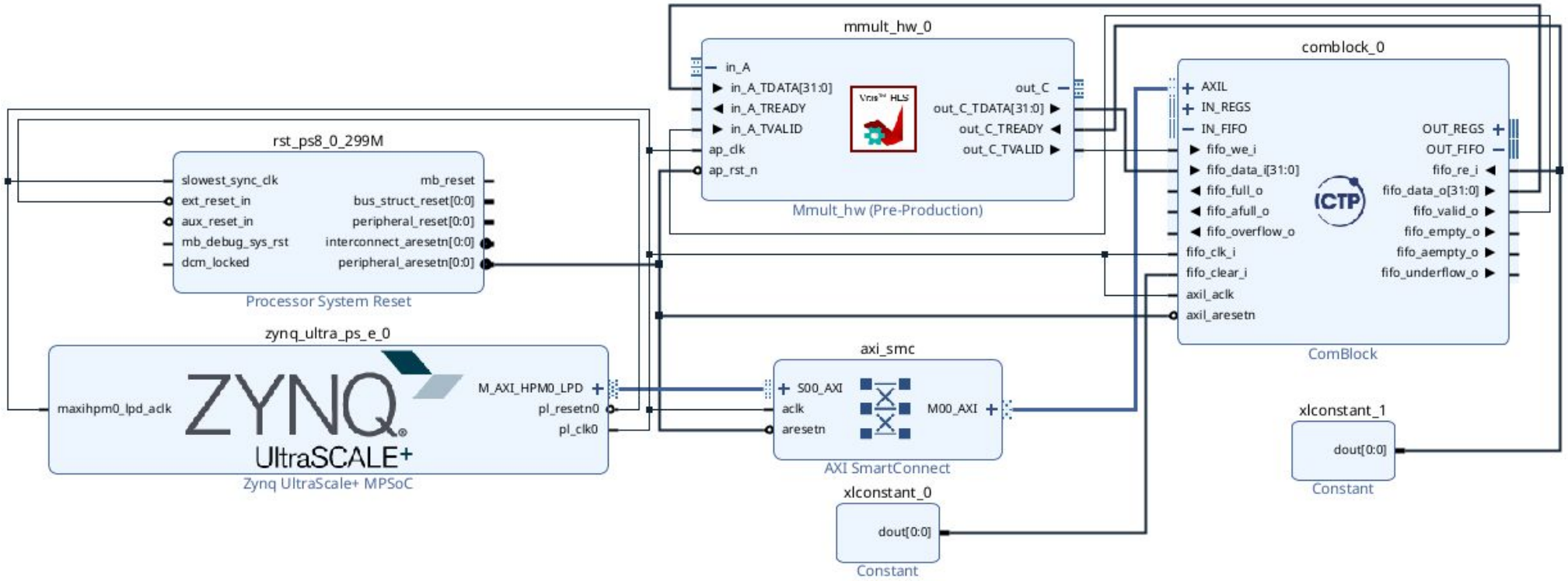
Maynor Ballina

Development Workflow - Vivado



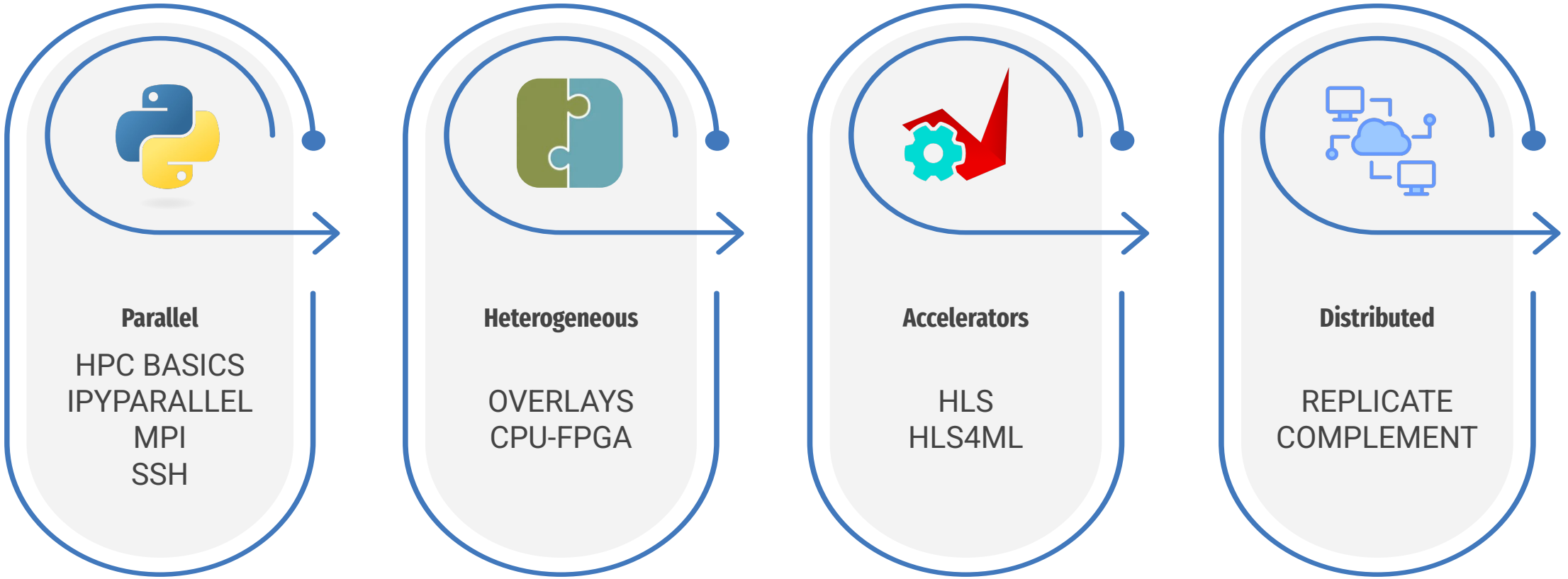
Maynor Ballina

Development Workflow - Vivado



Maynor Ballina

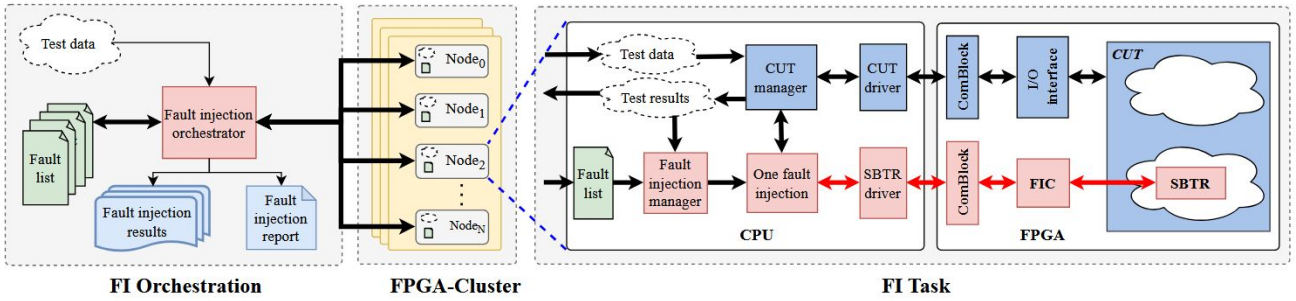
Broad spectrum of computational tasks

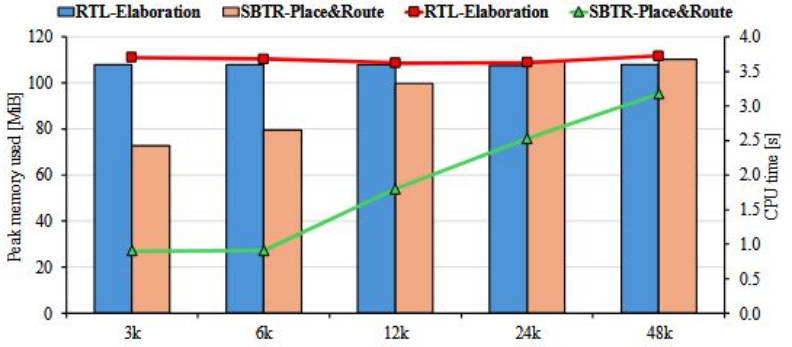


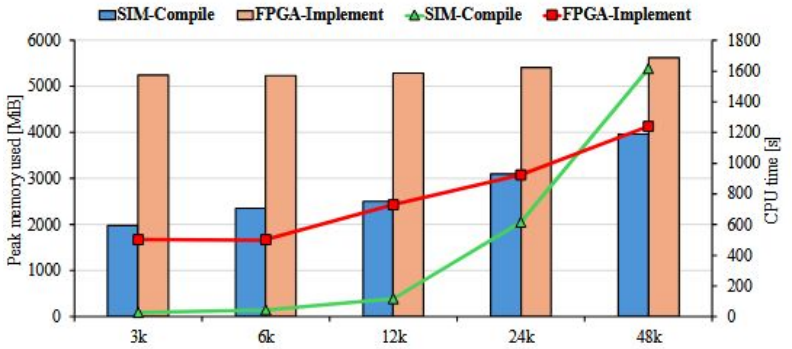
Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

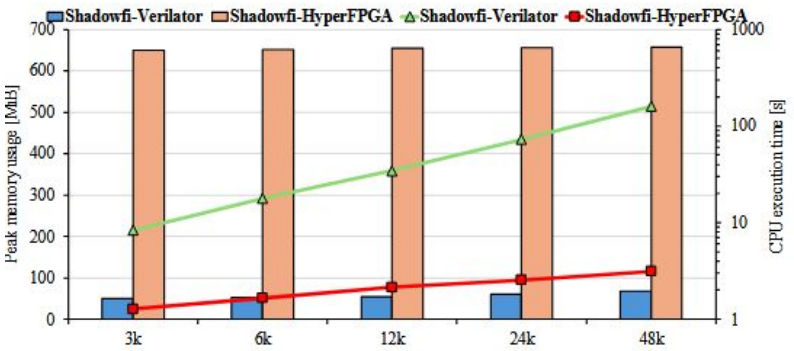
Dynamic platforms for interaction with FPGA clusters

- Collaboration with the Politecnico di Torino.
- Fault Evaluation of Complex IC Designs using Hyperscale Computing
- The workflow was evaluated on a set of IC design benchmarks, demonstrating practical usability and significant speedup in fault injection





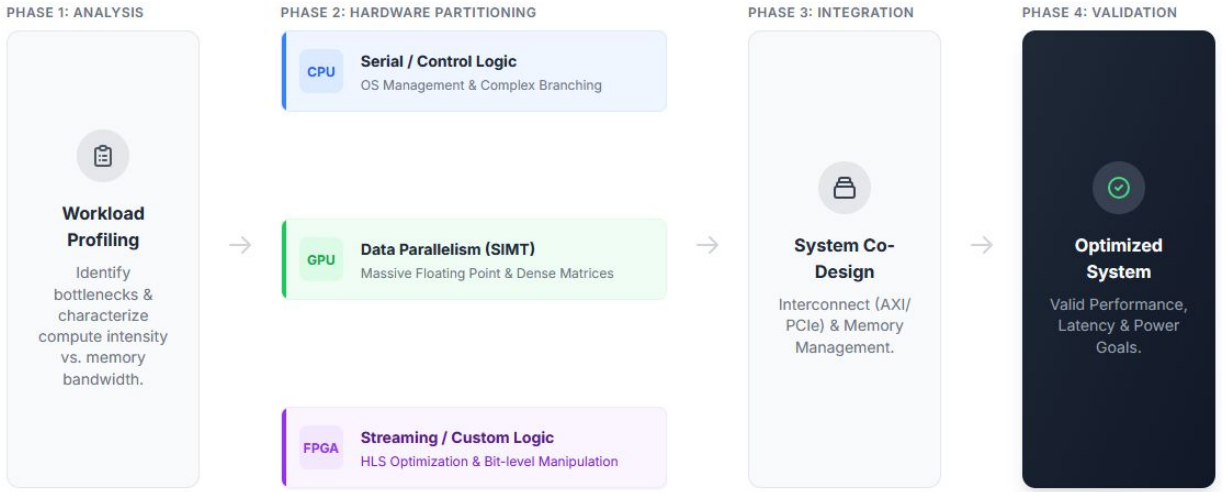


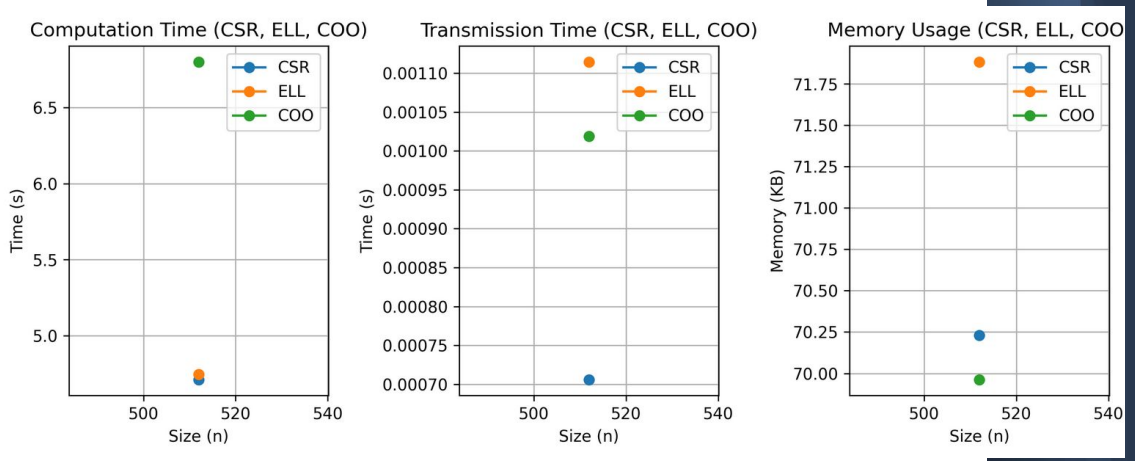


Maynor Ballina

Methodology for profiling computing problems on heterogeneous edge devices.

- Definition of a methodology in the context of computing problems.
- Testing of parallel and distributed implementations of the same algorithms.



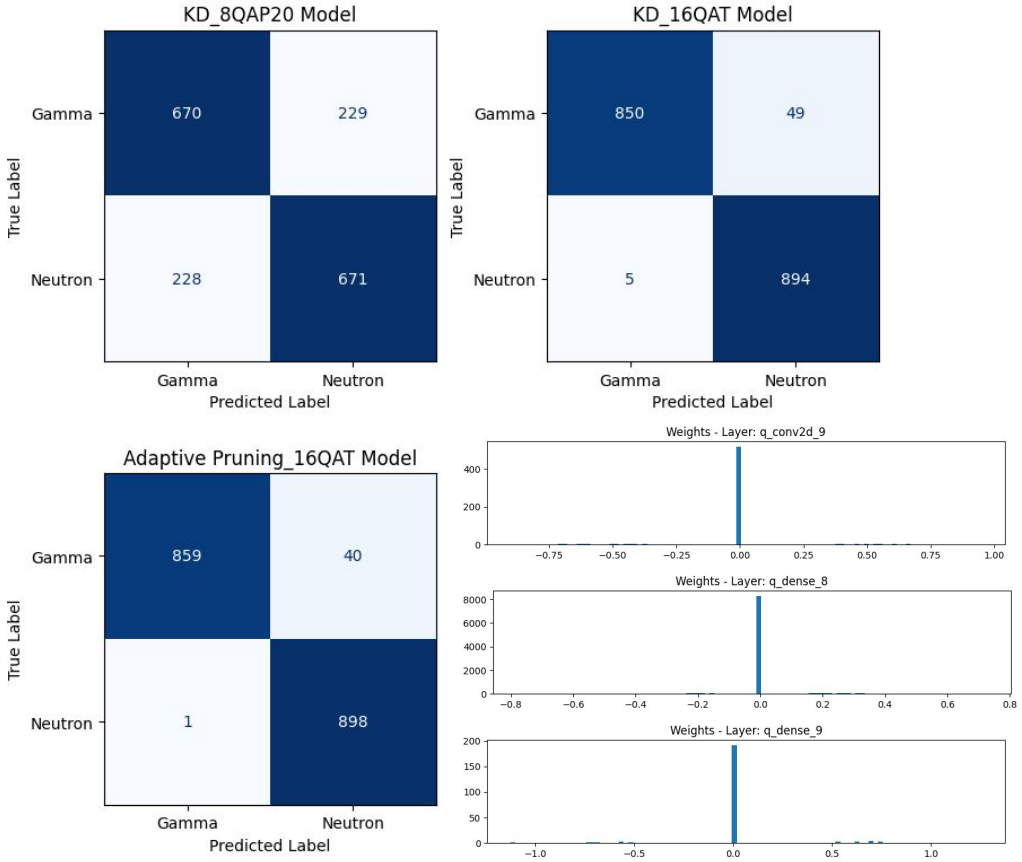


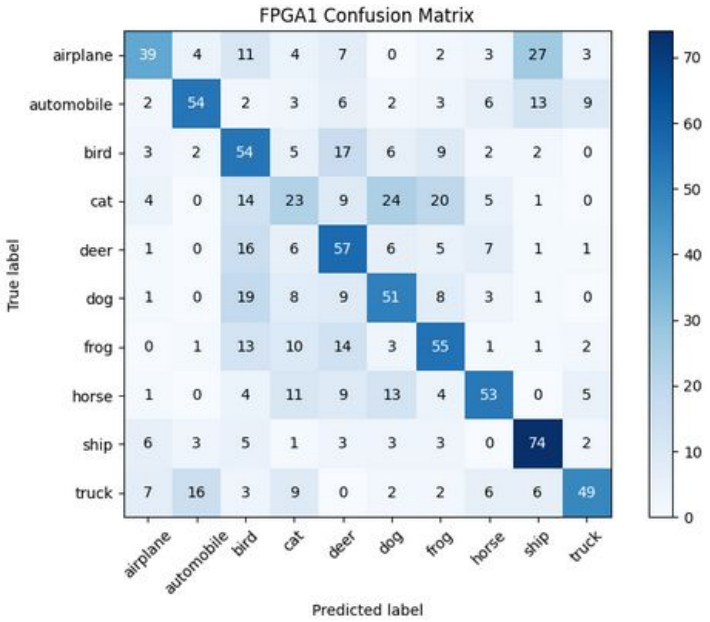
	Algorithm	Input	Computation Time (s)	Memory (KB)	Transmission Time (s)
0	CSR	512	4.710890	70.229492	0.000706
1	ELL	512	4.744235	71.881836	0.001115
2	COO	512	6.800936	69.962891	0.001019

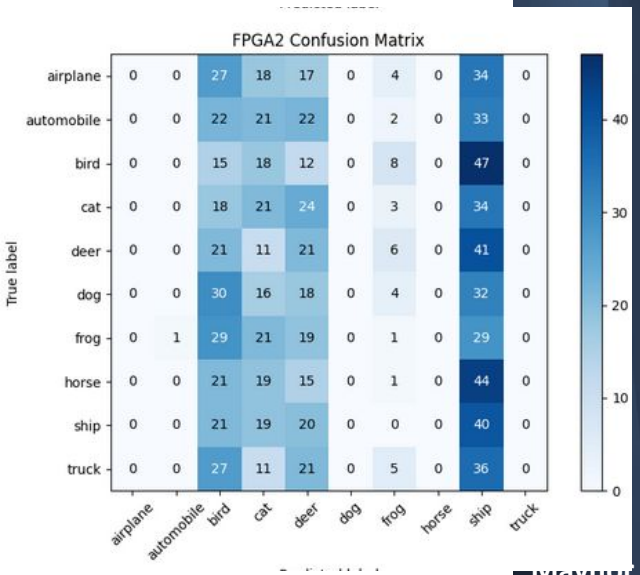
Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

ML compression techniques and validation thought parallel testing

- Profile and Test different compression techniques on a machine learning model.
- Deploy different models on a distributed cluster.
- Test and validations on parallel, reducing time.



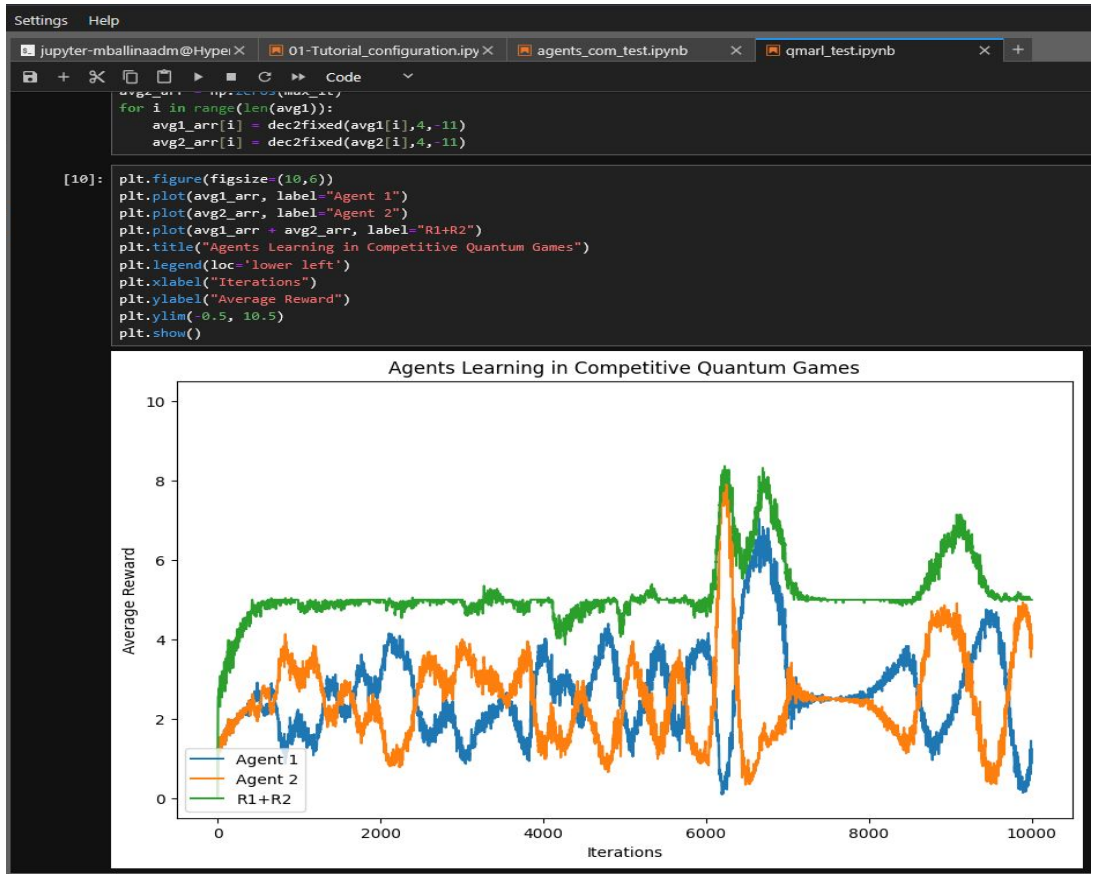


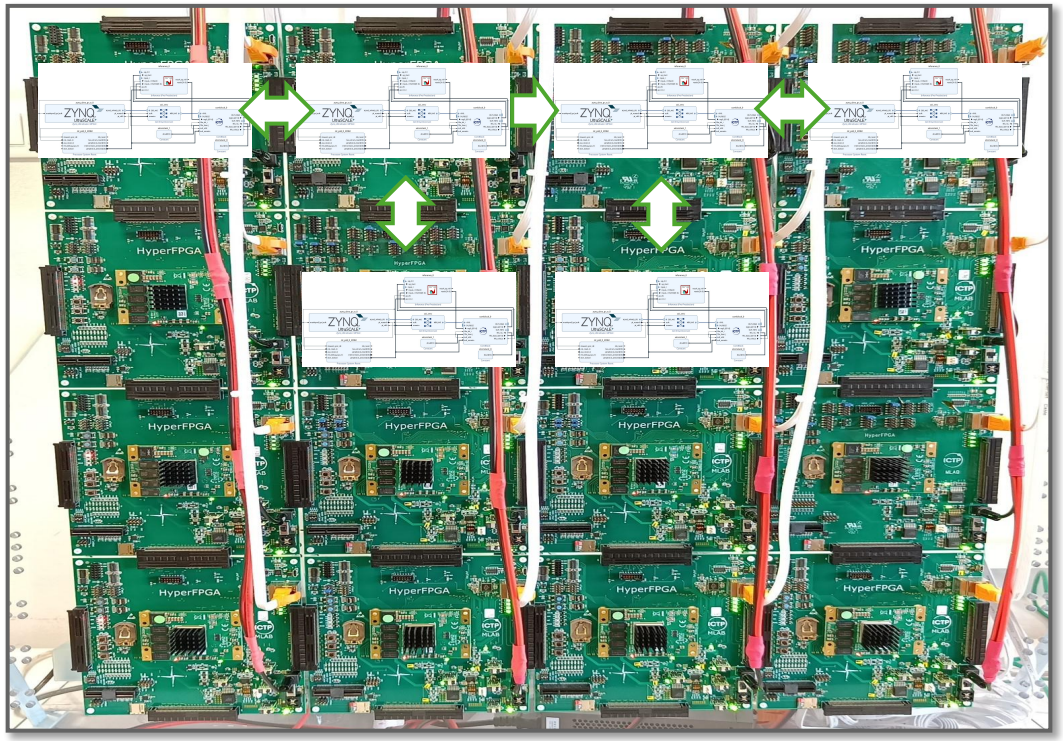


Mayhor Ballina

Validation of communication protocols for data intensive computing problems

- Test different communication protocol thought different ports.
- Implementation of distributed architectures.
- Split of large computational models.

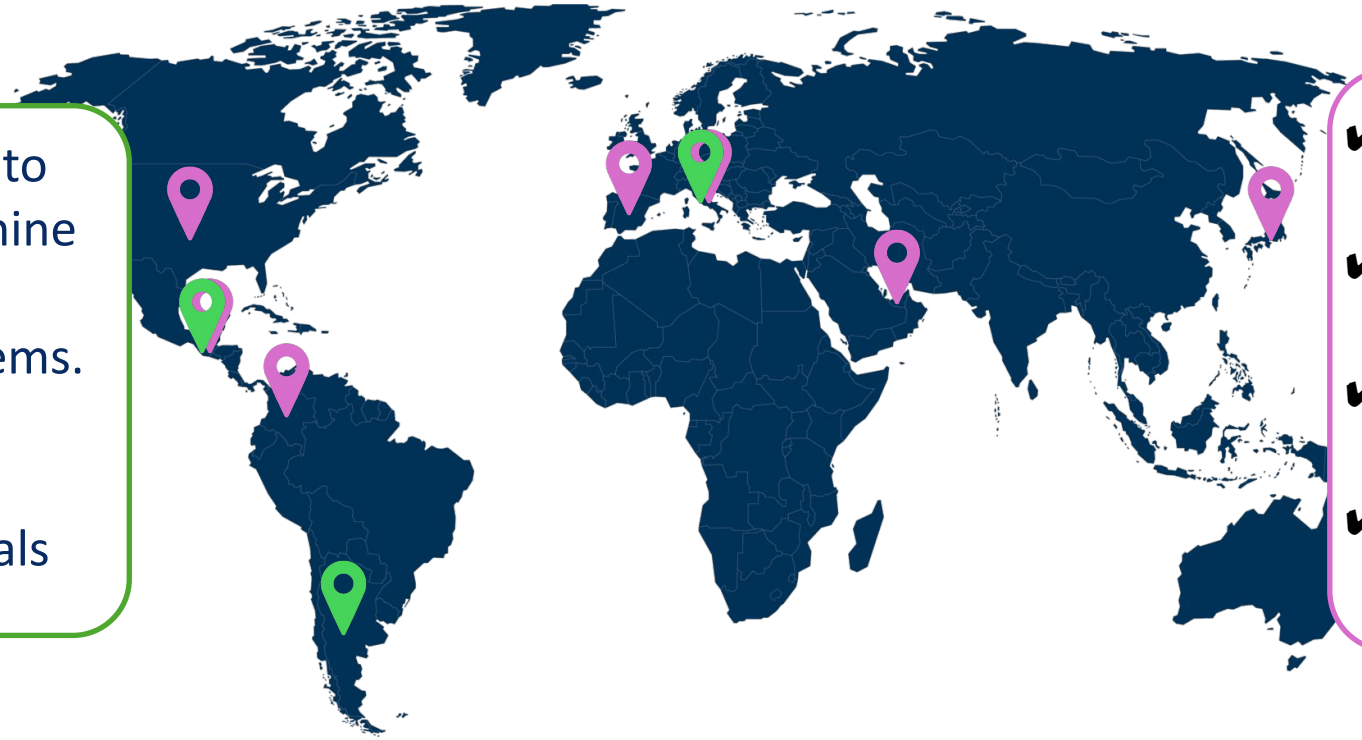




Maynor Ballina
UNIVERSITÀ DEGLI STUDI DI TRIESTE
The Abdus Salam International Centre for Theoretical Physics

Education and collaborative projects

- ✔ From Algorithm to Hardware: Machine Learning in Embedded Systems.
- ✔ Heterogeneous computing.
- ✔ HPC fundamentals



- ✔ Distributed computing
- ✔ Heterogeneous computing
- ✔ Fault Injection simulation
- ✔ Data driven ML architectures

Used in 3 courses with over 150 students across different countries and 7 different collaborative projects.

Conclusion:
Why is heterogeneous computing necessary?

Performance Optimization: GPUs excel at massive parallel processing, while CPUs are better suited for complex sequential and control tasks. FPGAs allow for specific customization of critical tasks and real-time optimizations.

Energy Efficiency: Some accelerators, such as FPGAs, are more energy-efficient than CPUs or GPUs, which is crucial for simulations requiring extensive computational resources over long periods.

Scalability: Accelerators enable simulations to run efficiently at larger scales. Without heterogeneous computing, some simulations requiring real-time data processing or massive scale would be unfeasible.

Latency Reduction: In certain problems, such as those needing real-time simulations or immediate feedback, FPGAs and other accelerators can significantly reduce system latency.

Thank you

Q&A

mballina@ictp.it