A proposal: FPGA project

F.R. Palomo[1], J.M.Hinojo[1], F.Muñoz[1], Jan Hammerich[2]
fpalomo@us.es
[1]Departamento Ingeniería Electrónica, Escuela Superior de Ingeniería
Universidad de Sevilla, Spain
[2] Particle Physics Group
University of Liverpool, UK

The infrastructure for the FPGA group could be comprised of:

1.Gitlab main project: the current repository could be used (RD50 HV-CMOS) or a new one can be generated to store the FPGA and GUI resources. Each element will be tracked by its own subgroup.

2.A Virtual Server to execute the runners related to the Continuous Integration. They will be in charge of executing the corresponding test for the Continuous Integration after each commit and determine if the code passes or not the test.

3. **A Slack Server**: In order to increase the communication between members, a **slack server** can be created or deployed in a private server. This tool is widely used in industry and open source projects. https://slack.com/intl/es-es/: We prefer *Mattermost* because is used today in our HVCMOS RD50 group.

4. **Maintainer Role**: Only one person of each institution will keep the maintainer role, the rest of members should be developers in order to ensure a proper work flow

Detailing about the first point, the main project organization could come as:

1.CERN Gitlab main project: the current repository could be used (RD50 HV-CMOS) or a new one can be generated to store the FPGA and GUI resources. Each element will be tracked by its own subgroup.
We should try to organize the FPGA repository to store the following items:

1. **Source Code:** sources for the DAQ system
2. **Testbenches:** unitary tests for the corresponding entities.
3. **Verification plan**: a functional testbench should be created in order to validate the developed code.
4. **Constraints** to implement the design
5. **Scripts** to synthesize and simulate the design.
6. **Doc:** doxygen or markdown can be used to generate the documentation of each system
7. **pcbs:** it will contain the schematics and layout views for each board. A specific folder can be used to track each board. It is important to remove the auxiliary files from the repo (i.e., logs)

   For the GUI, the organization should be similar, eliminating the PCB folder

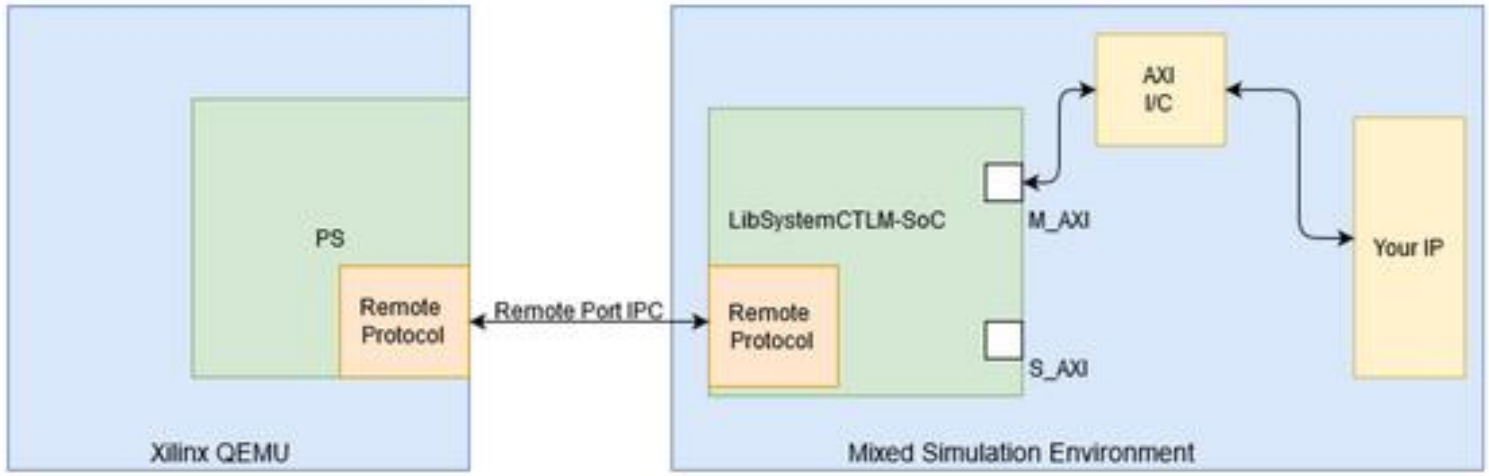
   In both cases, we should make use of the continuous integration in order to assess the code. Additionally, Vivado o Quartus projects should not be stored in the repository in order to avoid issues.

1. Digital blocks:

In order to verify the different digital blocks, a testbench framework should be developed. This framework should ensure minimal efforts when a Digital Block is changed. For this purpose, we should identify reusable parts, stablish metrics to evaluate the code and functional coverages, and make the code for testbench as generic as possible. In order to ensure the compatibility with the previous code and the tools used, the following package can be used:

- **UVM:** flexible class library that supports creating and monitoring reusable hierarchical testbenches.
- **Pyuvm:** package that implements UVM specification in Python. Reduce the complexity of creating testbenches
- **Cocotb:** package that allows to communicate python scripts with the simulator. It is possible to include modules to extract data from mixed signal designs.

2. Firmware: In order to speed up the GUI software for the DAQ, QEMU flow can be introduced. It is supported by Petalinux and it is compatible with the ZC706 board. Software could be tested at the same time that the hardware is developed.



Thanks for your attention

fpalomo@us.es