Contribution ID: **158** Type: **Oral**

Continuous Integration of FPGA Designs and Automation of the Development Environment

Tuesday, 3 September 2019 14:25 (25 minutes)

The high degree of flexibility in the firmware development makes FPGA designs and the development environment vulnerable to errors. Continuous integration is a fast way to detect a majority of such errors. Additionally, simulations and hardware tests can be automated using test methodologies (e.g. unit test). Continuous integration offers the benefits of reproducible results, fast error detection, error tracing, avoiding human errors in the build process, and minimizing the manual verification of the firmware. This comes at the price of setting it up with comprehensive integration tools such as GitLab. We present such an integration flow within the CMS experiment.

Summary

Especially FPGA designs with multiple contributors require comprehensive verification and testing. This process starts with the build environment and ends with the deployment of the firmware (FW). Therefore, Continuous Integration (CI) offers an automated way to set up, build, verify, control, monitor, and deploy the FPGA implementation. CI is widely used and successful in software development. CI is based on uniform build, simulation, and test environments and ensures a correct build of the FPGA design. Additionally, it covers essential functionalities of specified simulations and tests. Human errors in the build process are eliminated, e.g. missing files, wrong tool version, changed global constants, wrong submodule commit. Another main benefit is that the manual verification of the FW can be minimized, especially if different FPGAs and/or configurations are supported.

CI requires a version control system such as git, which is the baseline for most source code projects. In order to seamlessly set up the CI, a Command Line Interface (CLI) for FW builds is recommended. In this talk, we will introduce such CLI tools and present our choice. In our case, we extended the CLI tool by the feature of Xilinx Vivado simulation.

We use a CI system based on GitLab Continuous Integration / Continuous Deployment (CI/CD). We are presenting other CI tools as well and show how to integrate GitHub in the GitLab CI/CD. We show different aspects of CI in the example of the Track Stand-Alone (TSA) object FPGA board in the level 1 correlator. We elaborate on stages, jobs, pipelines, and runners of the CI. In this project, we set up several Hardware Description Language (HDL) simulations, which are performed in parallel, and evaluate their results. Only if this simulation stage is performed successfully the synthesis stage will be triggered.

We are sharing our experience with CI for FPGA designs and discuss tip and tricks as well as potential issues. Furthermore, we show the integration of High-Level Synthesis (HLS) designs and talk about the possibilists of extending CI.

Primary author: GLEIN, Robert (University of Colorado Boulder (US))

Co-author: PAVLOV, Borislav (University of Sofia (BG))

Presenter: GLEIN, Robert (University of Colorado Boulder (US))

Session Classification: ASIC

Track Classification: ASIC