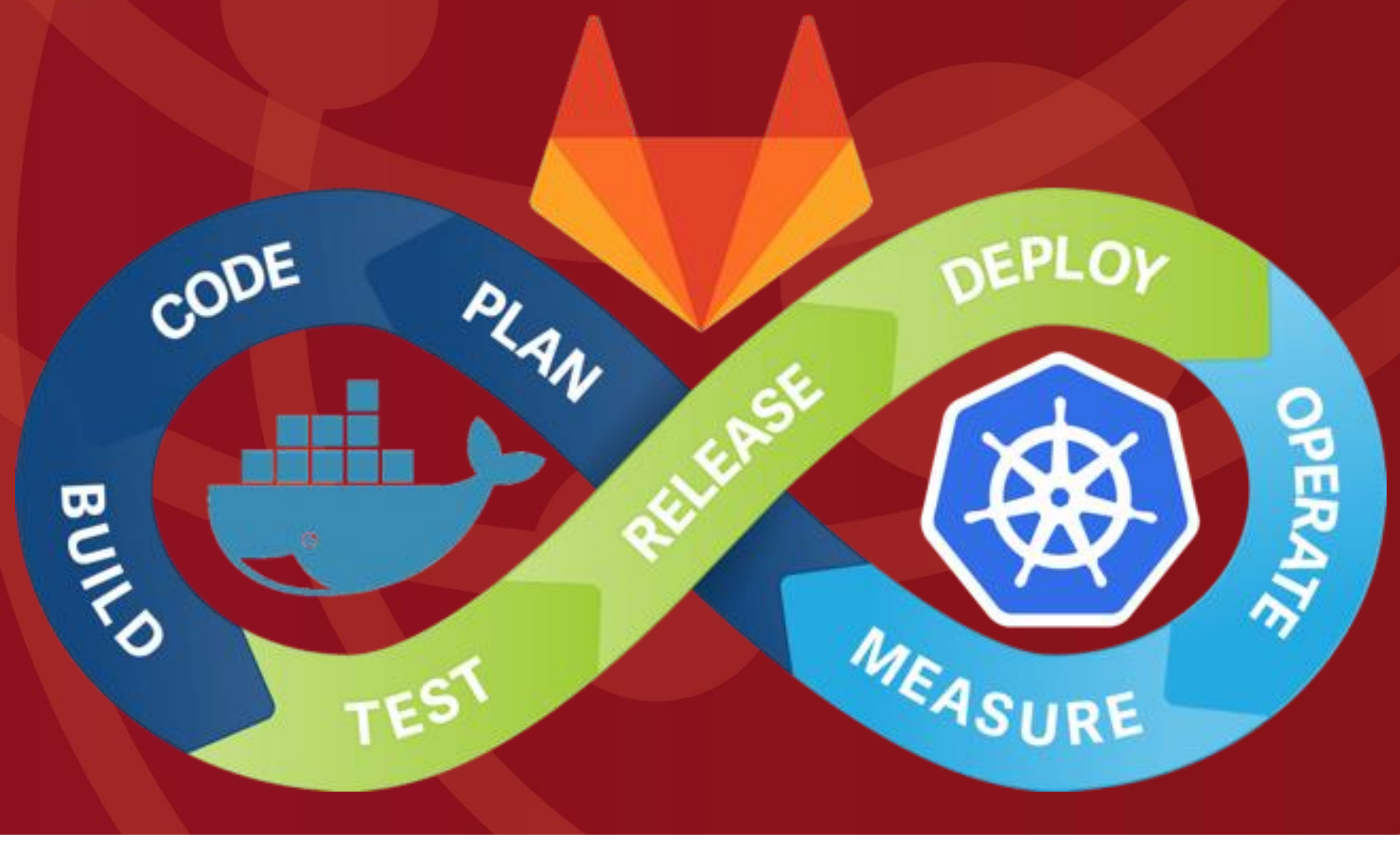
FW/SW framework for SRF cavity active resonance control

M. Donna(1, 2) – mad73@slac.Stanford.edu , J. Diaz Cruz (1, 2), A. Ratti (1, 2, 3), A. Benwell (1, 2), F. Wang (1, 2), L. Doolittle (3), S. D. Murthy (3), C. Bisegni (1, 2)

1. SLAC National Accelerator Laboratory, USA. 2. Stanford University. 3. LBNL Lawrence Berkeley National Laboratory

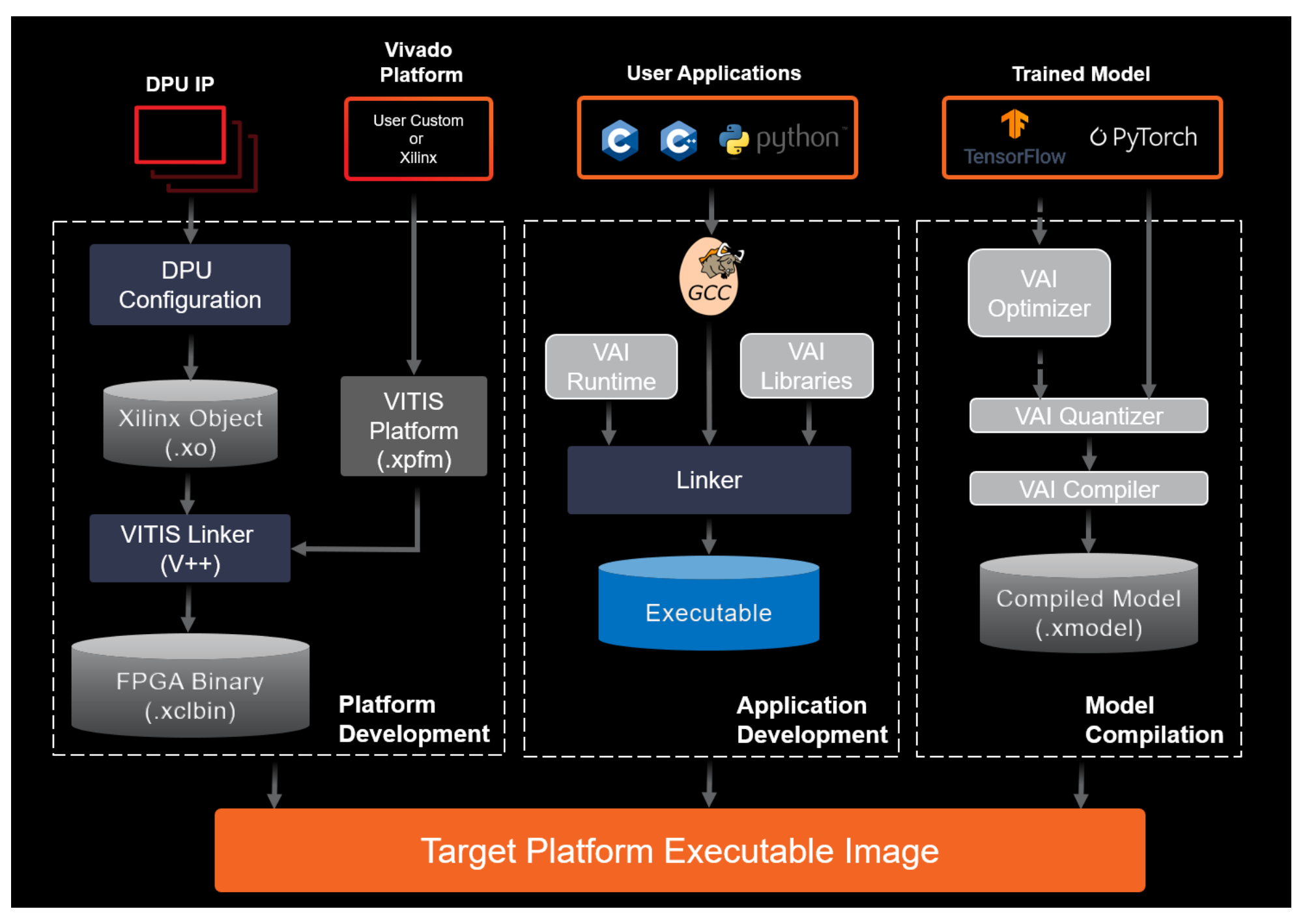


DESCRIPTION

Related to the high precision active motion controller based on ML, we are going to describe the CI/CD pipeline for testing and deploying the FPGA FW and embedded SW for the AMD Xilinx MicroBlaze processor that is in use at SLAC. We also introduce the different option to accelerate the SW using HLS flow to target FPGA FW blocks that replace the non-performing code. Latest part is to describe the porting from of the XILINX uBlaze processor to the RISC-V architecture and design the CI/CD pipeline to obtain the same results with the open-source architecture.

MICROBLAZE VS RISC-V

AMD (XILINX) MicroBlaze™ CPU is a family of drop-in, modifiable preset 32-bit/64-bit RISC microprocessor configurations. System designers can leverage the Vitis™ core development kit (2019.2), or the Eclipse-based Software Development Kit (SDK in 2019.1), to start developing for the MicroBlaze processor using select evaluation kits.

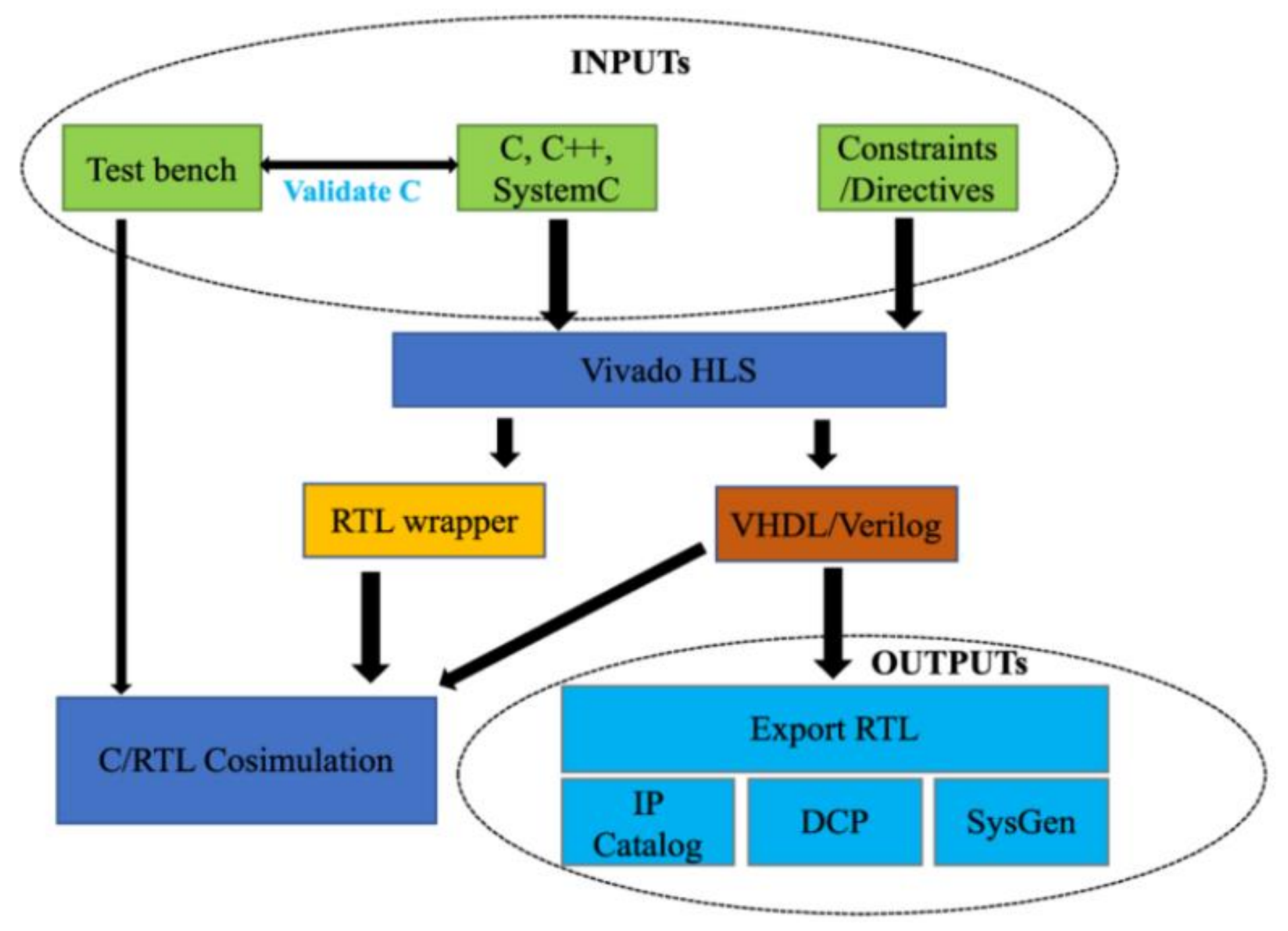


Simplified Vitis Workflow

PicoRV32 is a CPU core that implements the RISC-V RV32IMC Instruction Set. It can be configured and optionally contains a built-in interrupt controller. PicoRV32 is free and open hardware licensed under the ISC license (a license that is similar in terms to the MIT license or the 2-clause BSD license).
LiteX is a SoC builder / IP library and utilities that can be used to create SoCs and full FPGA designs. Besides being open-source and BSD licensed, its originality lies in the fact that its IP components are entirely described using Migen Python internal DSL, which simplifies its design in depth.
LiteX already supports various soft cores CPUs and essential peripherals, with no dependencies on proprietary IP blocks or generators.

HIGH LEVEL SYNTHESIS

High-level synthesis is the process of converting a high-abstraction-level description of a design to a register-transfer-level (RTL) description for input to traditional ASIC and FPGA implementation workflows. This high-level design description can be expressed in high-level languages such as C, C++, SystemC™, or MATLAB®, or graphical environments such as Simulink®. High-level synthesis tools use these as forms of design entry, and then synthesize—or generate—synthesizable Verilog® or VHDL® from them for use in ASIC or FPGA designs.



Vivado HLS design flow

CI/CD AT SLAC

GitLab's ecosystem is enriched with many features:

- the built-in continuous integration system (CI/CD) allows you to create a pipeline and control the lifecycle of the application's deployment, from downloading the code to the repository, until it is uploaded to the production environment
- with Auto DevOps, you can establish a CI/CD pipeline that automatically detects, builds, tests, and deploys your projects. Integrated with a Kubernetes (K8S) cluster, it allows you to deploy applications with no provisioning of extra CI/CD resources or configurations required

We have runners set up at LBNL, and we are going to add many K8 nodes that will support testing, simulation, compilation, build and deployment, using dockers

BOARD SUPPORT PACKAGE

In embedded systems, a board support package (BSP) is the layer of software containing hardware-specific boot firmware and device drivers and other routines that allow a given embedded operating system, for example a real-time operating system (RTOS), to function in each hardware environment (a motherboard), integrated with the embedded operating system. We like extends the concept to include also the first layer of HDL code that is directly connected to the FPGA PIN, so the full stack SW/HW/FW for the board is specified.

CONCLUSION

This poster is merely a list of action we are (and are going to) perform on the LLRF system at SLAC in the coming years.

A big effort has been profuse to prepare the test for the high precision active motion controller based on ML; the next step will be to replace the MicroBlaze with the Risc-V open-source platform, and create the needed pipeline to compile, test, and co-simulate it using Dockers containers: LiteX is a starting point, we need to improve the toolchain to been able to achieve the same results that we get with MB.

The CI/CD work at SLAC was started 1 year ago, and the final goal is to have all the LLRF supported board automatically deployed in our bench test (both at LBNL and SLAC), fully tested and verified, and ready for production operations; part of the effort is on the pipeline and is to set up the runners on our S3DF K8 cloud, for both EPICs and the different FPGA FW.

Another big effort is to improve the quality of the code, and for that we need to have better Code Coverage on our testing, and improve the open-source tool that perform different type of checking like Linting, Clock Domain Crossing , etc…

ACKNOWLEDGEMENTS

Use of the Linac Coherent Light Source (LCLS), SLAC National Accelerator Laboratory, is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-76SF00515
Many thanks to Claudio Bisegni for his work as DevOps, to Amedeo Perazzo and Yee-Ting Li part of TID SCS for the help with S3DF settings, to the all LLRF team at SLAC and LBNL for their support and help, and to Alessandro Ratti and Joshep Delong of EED AD .

REFERENCES

Icalepcs2023 -TUMBCMO14 Initial Test of a Machine Learning Based SRF Cavity Active Resonance Control
AMD uBlaze (https://www.xilinx.com/products/design-tools/microblaze.html)
PicoRV32 on XILINX FPGA (https://github.com/YosysHQ/picorv32)
LiteX library (https://arxiv.org/abs/2005.02506)
CDC detection with yosys (https://github.com/YosysHQ/yosys/discussions/3956)
RISC-V SoC Open-Source Tools (https://www.ijraset.com/best-journal/implementation-of-riscv-soc-from-rtl-to-gds-flow-using-opensource-tools)
Kubernetes (https://docs.google.com/presentation/d/1e6RxrR9VMwFN6gESqHmFjNTTHp5lwiGxh9BmqnCm8Ig/edit#slide=id.p1)
SLAC Shared Scientific Data Facility (S3DF) (https://s3df.slac.stanford.edu/public/doc/#/)