Emulation of a prototype FPGA track finder for the CMS Phase-2 upgrade with the CIDAF emulation framework

Luigi Calligaris\textsuperscript{1,2} on behalf of the CMS collaboration

\textsuperscript{1} STFC Rutherford Appleton Laboratory, Didcot, OX11 0QX, United Kingdom. email: luigi.calligaris@stfc.ac.uk
\textsuperscript{2} Supported by EU FP7-PEOPLE-2012-ITN project nr 317446, INFIERI, “Intelligent Fast Interconnected and Efficient Devices for Frontier Exploitation in Research and Industry”

**Motivation**

The CMS collaboration plans to upgrade its outer silicon tracker detector by 2026. This "Phase-2" upgrade prepares the experiment for the High Luminosity phase of the LHC (HL-LHC). The upgraded tracker electronics will provide a list of reconstructed tracks to the Level-1 trigger, to improve its performance.

A proposed track finding system is based on a Hough transform implemented in firmware, designed for FPGA hardware. Currently, a demonstrator for the system is being commissioned. The latter is based on MP7 cards, featuring a Xilinx Virtex-7 XC7VX690T FPGA.

To test the behavior of the firmware, a software emulator in C++ and an emulation infrastructure to help its development have been written. This software infrastructure has been named CIDAF (Circuit Data Flow), is independent from the specifics of the system being emulated and can be used in many different applications. CIDAF is available for testing from the author of this poster.

**CIDAF**

CIDAF is a library providing a series of facilities that aid in the development of C++ emulations of firmware. It features a clocking system, a data transfer system and tools to help bringing the behavior of the simulation to coincide with the one of the hardware, e.g. support for fixed-width integers.

In a typical use case, a simulation for individual components making up a complex firmware would be written in C++, taking as reference the hardware description language source. The components would then be connected together using CIDAF.

**Results from HT firmware emulation**

Comparison between firmware running on hardware (black dots) and emulator implemented with CIDAF (blue line)

![Emulation Results](image)

Table on the right: average number of tracks per event found over 1000 events, for each of the listed types of simulated Monte Carlo samples.

![Emulation Table](image)

**Emulation example**

class Component1

```
ActiveOutput<Stub> Write(Stub&)
Tick()
```

Class Component2

```
PassiveInput<Stub> Write(Stub&)
Tick()
```

Cycle 0: Evolve internal state

Commit transfers

Cycle 1: Evolve internal state

Commit transfers

... 

A driver program triggers the clock responsible for evolving the internal state of Component1 and Component2, then initiates the transfer of data from Component1 and Component2 by triggering the clock associated to the Component1’s ActiveOutput. This repeats at each emulation cycle.

![Emulation Diagram](image)