SP-545:
HPS2FPGA second DeTrl interface

Demo of the DeTrl and RegDef systems used for TalonDX Firmware and Software development.

Dr William Kamp, High Performance Computing Research Lab, AUT, NZA

Alphabet Soup

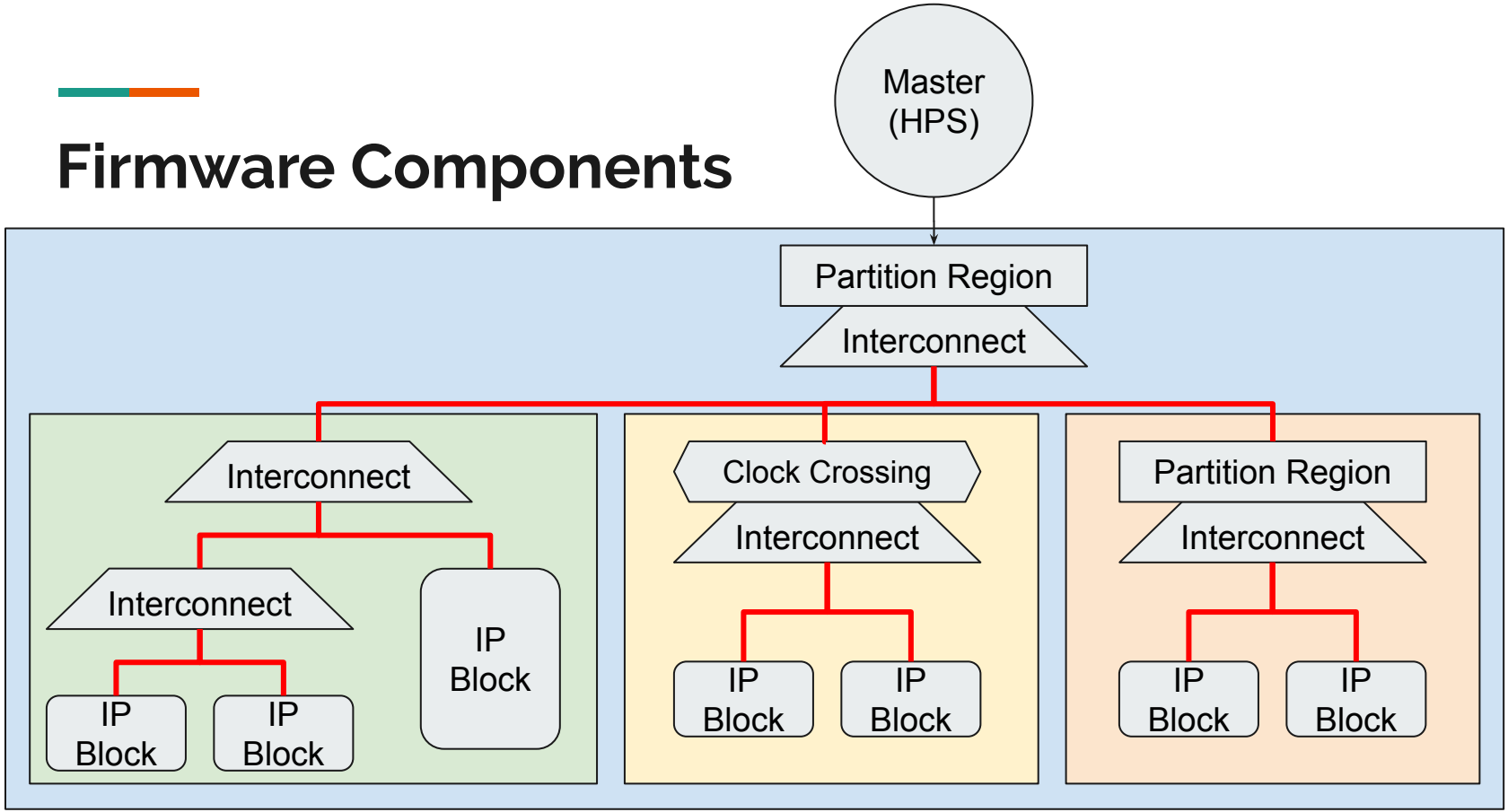
- DeTrI - Device Tree Interconnect - pronounced “Dee-Tree”.
- RegDef - JSON based register definition and code generation.
- HPS - Hard Processor System - the ARM processor on the Stratix10 FPGA
- HPS2FPGA - name of the high performance memory mapped interface from HPS to FPGA fabric.
- TalonDX - Stratix10 FPGA hardware platform developed for Mid.CBF and Dish.

Objectives

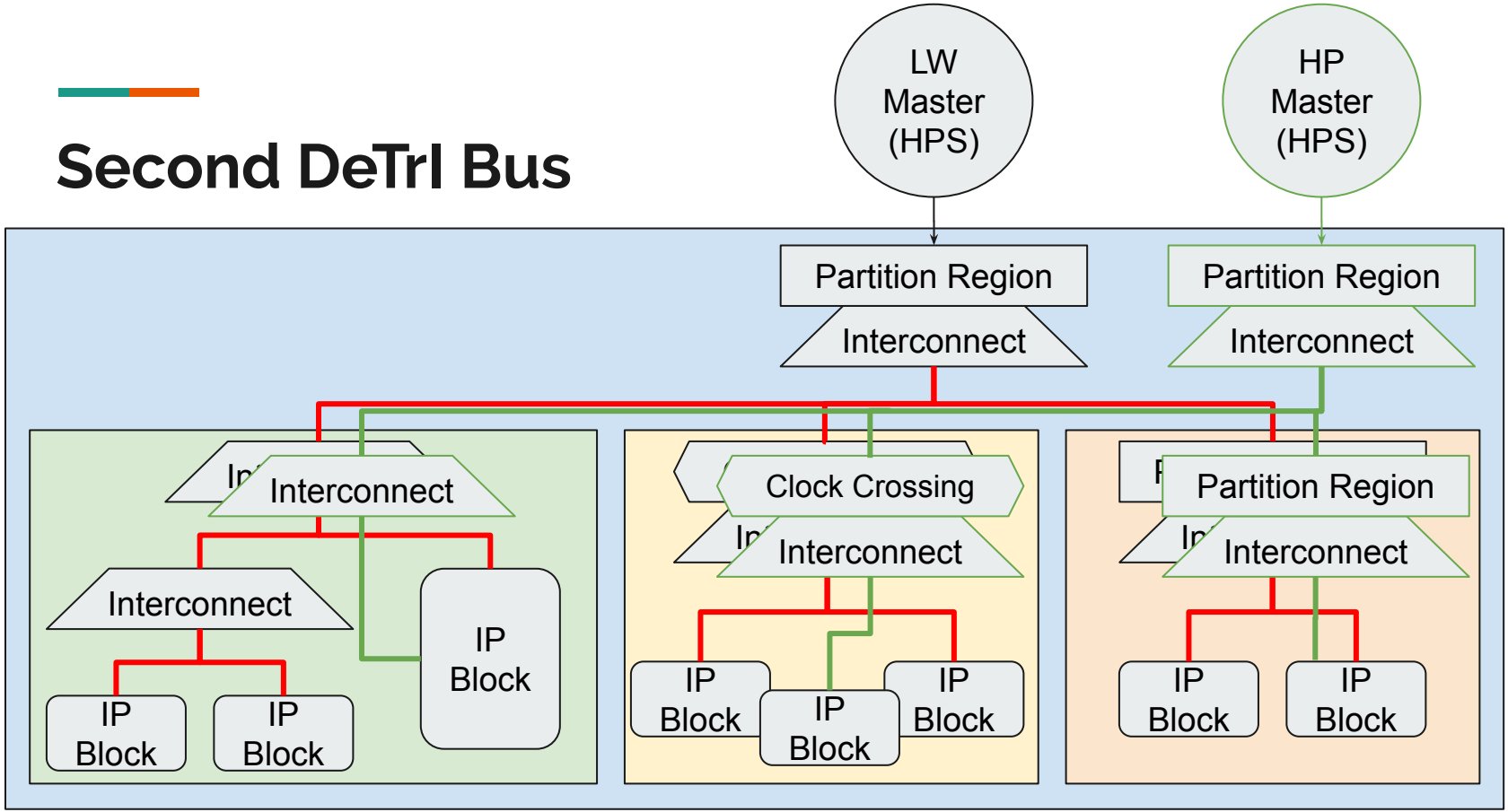
- **SPO-287**
  - Progress the design and implementation of TDC on the path to AA0.5
- **SPO-237**
  - Provide functional software and firmware infrastructure for the TALON-DX Board

- RegisterDef
  - Single source the definition of Register Sets and their fields between FW and SW.
- DeTrI
  - Automate addresses assignment and communication of those addresses to Software.

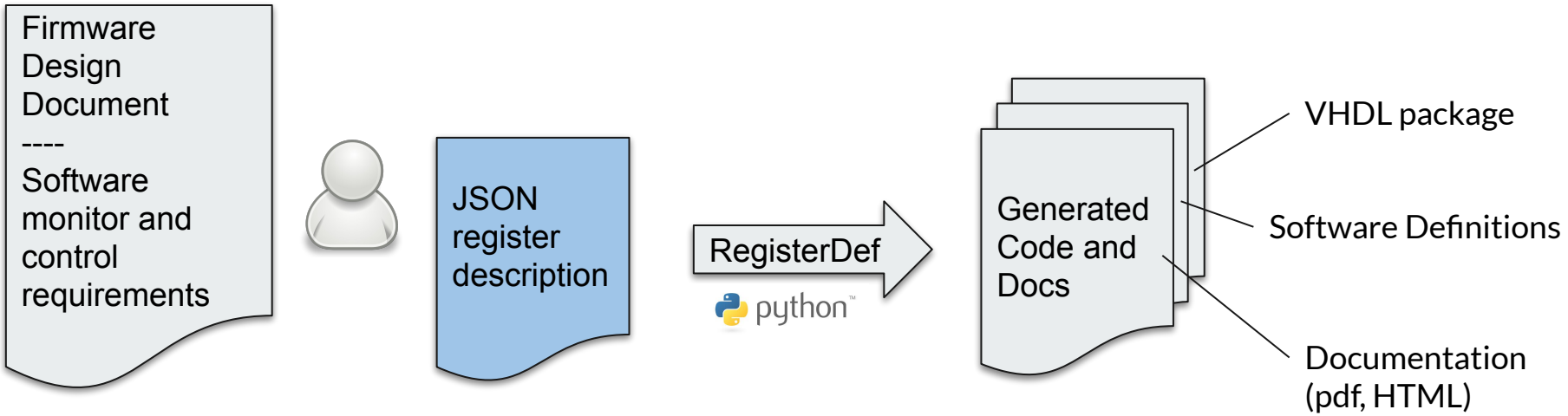
Firmware Components



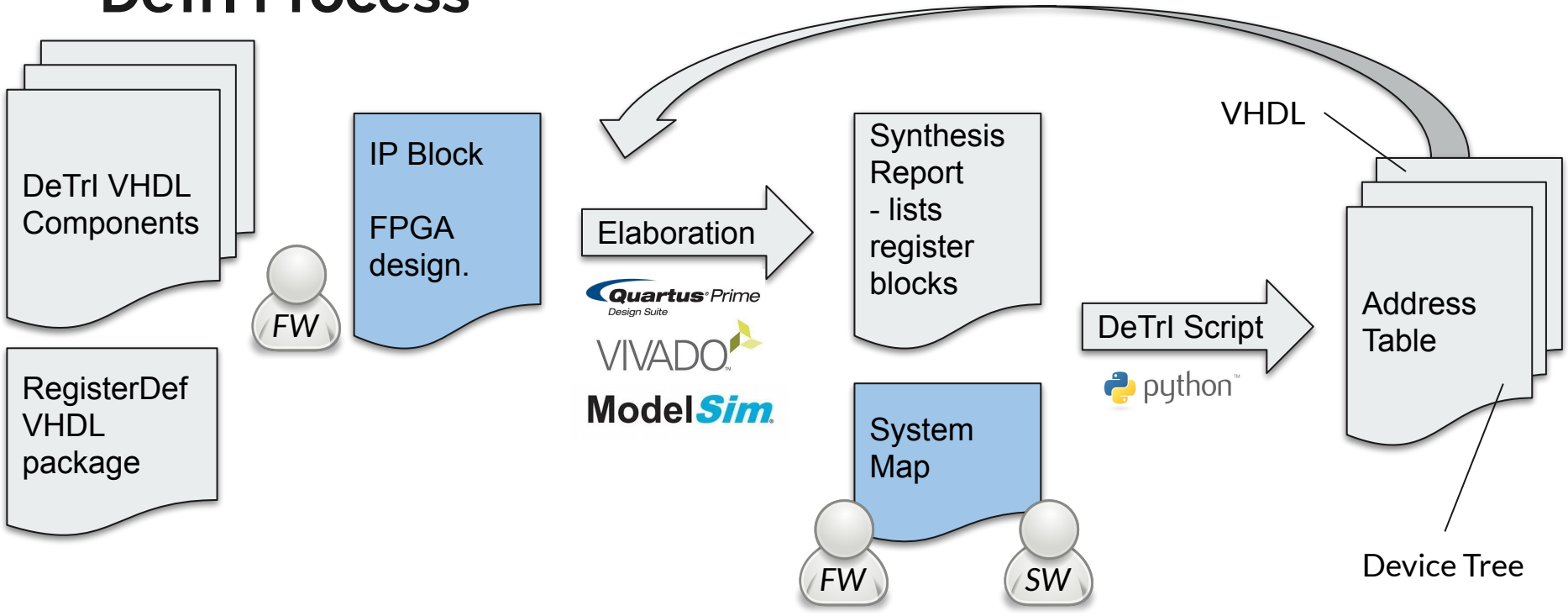
Second DeTrI Bus



RegisterDef Process



DeTrI Process



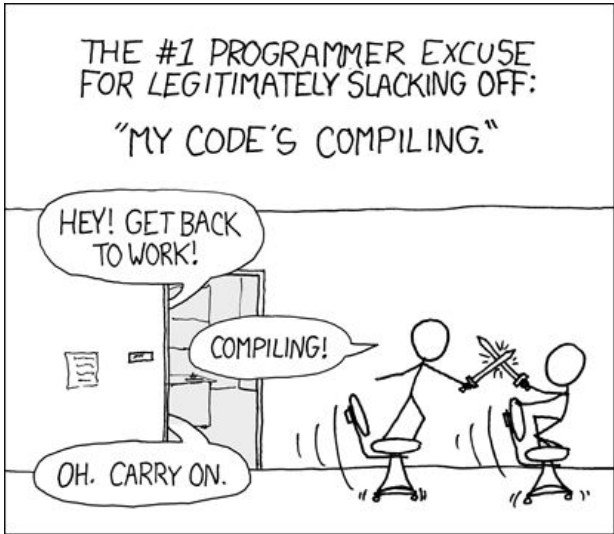
DeTrI - What and Why

Three parts:

1. FPGA Firmware components:
    a. Abstract VHDL entities that build up the interconnect.
        i. Register bank, RAM bridge, Interconnect, Clock domain crossing, ...
    b. Concrete implementation - currently AvalonMM based, but could be AXI or Wishbone or ...
2. Address assignment Script: written in Python.
    a. Extracts register instances from Synthesis Report. Packs them into the address map. Outputs them in various formats: VHDL address table, Linux Device Tree format.
3. Automation of the automation, written in TCL.
    a. Scripts to hook into the Quartus Tool.

Example: Add a field to an existing register set

1. Add field description to JSON file. (2 mins)
2. Run RegisterDef.py script on JSON file to generate VHDL code. (1 min)
3. Implement field logic in VHDL. (1 - 15 mins?)
4. Click Build and go get coffee (automation takes over.)
    a. VHDL elaborated - synthesis report generated.
        i. auto sized the register bank for the new field.
    b. Addresses assigned to all modules



Problems solved

- Adding to the register bus is error prone. Mistakes often mean complete lockup of the register bus.
  - Use standardised components.
- Manually assigning addresses - tedious work, and error prone, doesn't support rapid development.
  - Use automated address assignment.
- Communication of register addresses - error prone.
  - Single source the addresses.