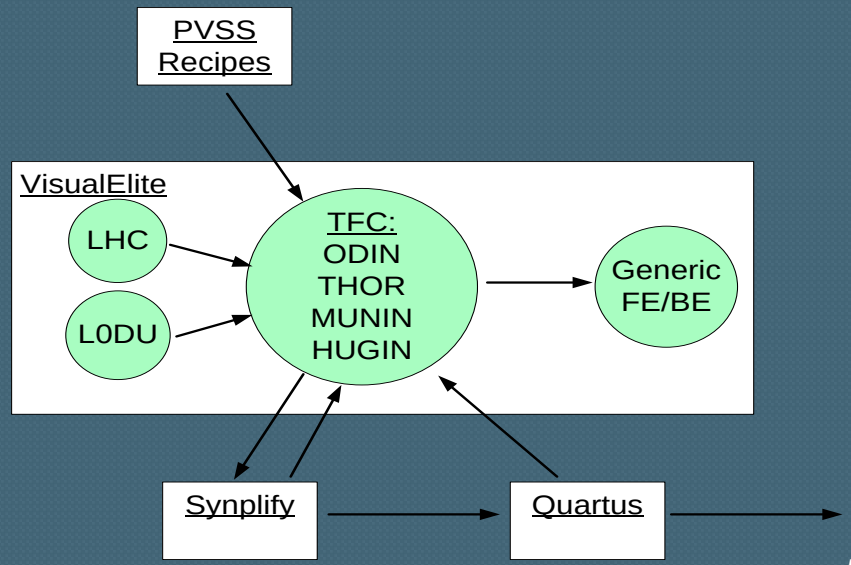
TFC Development Methodology

Development methodology which was defined for the *current* TFC system:

- Timing and readout control requires cycle-timed complete system-level simulation
- Aim find tool for
  - Design and integration environment for synthesizable code
  - Library organization
  - Transaction-level modeling of surrounding system components
  - Graphical environment for system-level design
  - Allow multi-FPGA simulation together with discrete logic
  - Plug-in approach to system simulation with any object implemented as behavioral, procedural/data flow, synthesized RTL netlist, placed/routed netlist with final timing.

➔ Tool VisualElite from MentorGraphics chosen



Development Methodology

- System simulation maintained up to date *always (11 years now)*
  - Typical application: reproduce exact situation at the pit by downloading exact copy of configuration and control registers via CCPC emulation
- Coding
  - Procedural/data flow style in VHDL
  - Strict code conventions and nomenclature
  - Control interface strongly coupled to readout control logic
  - Contain configurable test functions
  - Very large amount of monitoring (real-time and sampled status flags, counters)
- Timeline
  - TFC system must be defined and available at a very early stage
    - Specs and simulation test bench before launching prototype production of FE
    - Plug'n'play hardware+control before finishing design of FE/BE

    ➔ Flexible implementation with large amount of reserve resources to adapt to changes in operational specs of experiment
- Important: TFC firmware, in particular ODIN, is the place where non-conformities and additional operational requirements come up
  - E.g. current ODIN has been updated on the average every ~3weeks since 2009 !

  ➔ Fast validation procedures

  ➔ Consequences also for development strategy of TFC control system