Mini-workshop on the use of System Verilog and UVM

Jorgen Christiansen CERN/PH-ESE

Reasons for the workshop

- Several groups now start to use SV and UVM (both inside and outside CERN)
  - Design and verification of complex ASICs
  - (Design and) Verification of complex FPGA's
- Training
  - 24 people has followed 1 week training at CERN (Doulos course)
  - Other training (IMEC, etc.)
- Normal/old RTL Verilog (and VHDL) very good for design (synthesis) but often not sufficient for verification of very complex digital designs/systems
- Many different Verification options
  - Verilog RTL design with SV + UVM
    - Same tool/simulator from gate level to very high level
  - OSVVM: Open Source VHDL Verification Methodology
  - VHDL RTL design with SV + UVM
  - Verilog/VHDL RTL with C++ for complex verification
  - System C (or C++) for Verification (and design)
  - Other ?

It can be discussed for long time what is best

Depends on nature of project

Depends on people in the project (e.g. their background: Verilog, VHDL, C++)

SV + UVM

- Three “levels”:
  - Design/RTL/synthesis/gate level
  - High level verification features in SV: Object oriented, transaction level, Constrained random , ,
  - UVM for standardizing and reuse verification methodology

  The merging of the three levels has become a “usine a gas”.
- The community:
  - Hardware designers
  - Verification software

  In (large) companies making complex designs a person normally does not do both hardware design and high level verification.

    Two different teams/groups

      Different expertise and background (hardware – software)

      One better not verify ones own design (tends to make same “mistakes”)

      Parallel design and verification to shorten time to market

    Companies often have a verification environment from previous similar designs (maximize reuse of both design and verification)

  In HEP we often have to do everything with very few people.

Workshop and our community

- Presentations of who is doing what with SV + UVM
- Exchange of practical experiences:
  - What works well and what does not
  - Ideas of how to do things efficiently
  - Simulators and related tools
  - Sources of useful information
  - ?
- Possible collaboration among people/groups ?.
  - Reuse across projects/groups (often difficult in HEP)
    - RD53: ATLAS/CMS/LCD pixel simulation/verification framework
    - Other ?
- Email group for questions and ideas within our small community: system-verilog@cern.ch
  - Such an Email group is only useful if being actively used by community.
- If appropriate, we can repeat such a mini workshop later.