BLUWIRELESS TECHNOLOGY

Verification Challenges

Andy Lunness

Challenge #1

- No model to compare our implementation against
  - What would traditionally be the model (C model) is our implementation
  - Our upper level model is Matlab, too abstract.
  - Verification requires a partial model embedded in the test bench.
  - Verification effort is higher

Challenge #2

- Integrating our auto-generating framework to third party delivered IP
  - Third party CPUs, specialist blocks (LDPC), busses/interconnects delivered as RTL
  - Hard to simulate the whole device in System-C
    - Verilator is not great, hard debug – auto generated
    - Simulate with generated RTL, hard to debug System-C, slow.
    - Constrained by licences

Challenge #3

- SystemC rather than System Verilog
  - High level implementation language with a flow to silicon
  - Cutting edge, fewer trained engineers
  - UVM and coverage frameworks coded by TVS is a viable alternative to System Verilog
  - CRAVE public domain third party library
    - Plugs deficiencies in SCV
    - University of Bremen
  - Autogenerated framework for blocks
    - Effort spent on scoreboard, checker, randomized basic test. Refine for missing coverage.
    - Integration at higher level: Regression, Jenkins, …