



SiCortex

# Hardware-Software Co-development

May, 2007

# Problem Statement

---

- To co-develop hardware and software, software must be able to run and debug code on the hardware simulation model(s)
- What do software developers need from simulators to support co-development?
  - More speed
- But in addition to that, what do they need?
  - **Still more speed**
- Seriously, they surely have to have...
  - ***Even more speed!***



# (Affordable) Strategies for Speed

---

- RTL (Verilog) – slowest (100Hz)
  - Directly simulates the design (pre- or post-synthesis)
  - Difficult environment for software debug
- RTL converted to System C – slow (1Khz)
  - Free!
- Behavioral – fast (5-10Mhz)
  - Little or no relationship to hardware implementation
  - Excellent environment for software debug
  - Also free!
- FPGA's – fastest (20-40Mhz)
  - “Directly” implements the design
  - Requires alternate CAD tools and debug boards



# SiCortex' Approach

---

- Mixed mode simulation starting with pure behavioral and ending with pure gate-level
- Debug the Linux port on a software-friendly behavioral simulator
- Next: speed-optimized mixed-mode
  - Behavioral cores
  - System C high-level models (HLM's) wherever possible
  - System C verilated RTL everywhere else
- Then: substitute verilated RTL (and then Verilog RTL) for HLM blocks
- Then: replace behavioral cores with RTL cores
- Last: replace System C global wrapper with Verilog global wrapper



# Linux Boot Times

---

- Pure behavioral simulator
  - 50 seconds
- Mixed mode - behavioral cores, verilated RTL, System C
  - 3 hours, 27 minutes
- Mixed mode - verilated cores, verilated RTL, System C
  - 14 hours, 17 minutes
- Full chip RTL under Verilog simulator
  - 28 hours
- Full chip netlist (gate level) RTL under Verilog simulator
  - 101 hours
- All runs are streamlined SMP kernel to user mode init
  - Approximately  $16 \times 10^6$  instructions
- Full Linux kernel, gate level RTL under Verilog simulator
  - 360 hours



# Needed (and Not Needed)

---

- Needed
  - Real models of external devices (what's at the other end of all the modern serial interconnects)
    - BFM's don't hack it for software bring-up
  - Vendor acceptance and support of open source EDA tools and techniques
    - Behavioral libraries, for example
- Not needed
  - More proprietary languages

