

SystemC

A User's Perspective

Bill Dittenhofer
formerly of: Starkey Labs

Thanks to:
Eileen Hickey
Ed Powell



SystemC Usage

- RTL Verification
- Modeling (UTF,TF, BA/CA)
- System Simulation (no RTL)
- FW development on SW model
- Architectural Verification
- Deployment of System Verilog

Applications

- Enterprise Level Servers
- Embedded Processors
- ASSP's
- Algorithms (FFT, DCT, Wireless, etc.)



Improvements

- Topology / Configuration generation
(similar to generate function)
- Mixed Signal Libraries
- More Academic Proliferation
- Keep entry price low
(benefits start ups, small companies)
- More industry donations
- Tools to support higher abstraction
(formal verification, advanced assertions)
- Increase interactions with System Verilog
(common TLM API's?)
- Improved simulator performance



Enterprise Server Chipset – mega million gates

Hi availability, reliability, throughput

Multiple 64 bit processors

Multiple level caches, hi speed I/O

Development teams 50+, 500K+ LOC

DSP ASSP - 1.4 millions gates

5 embedded processors (2 types, 20+ bits)

4 fixed function DSP cores (FFT, OCL,FE,FBC)

Development team 12+

Embedded Processor 55k gates

Dual Harvard, pipelined, low power, 20 bit fixed point
with custom instruction set, DSP extensions



Why SystemC?

- Non proprietary, IEEE Std., LRM
- Industry Support
- TLM, AVM, SCV, etc.
- Proven capabilities
- Cost Effective

Why SystemC Tomorrow?

- Behavioral (ESL) synthesis
- Plays well with others (System Verilog, VHDL)
- Increasing need for abstraction
- Continual improvements (TLM, SCV, etc.)
- Leverage investment

