SystemC Tools

Grant Martin
Fellow, Cadence Berkeley Labs
22 October 2002
Agenda

• The Context for SystemC
• SystemC Tool Survey
• Cadence and SystemC
• Future SystemC Tool Possibilities
The Context for SystemC

(Hugo De Man’s “7th Heaven of Software”)

System and SW Modeling: UML, SDL, etc.

System Level Integration Infrastructure: SystemC

Mere Implementation!! VHDL, Verilog, SystemVerilog

(Hugo De Man’s “Deep Submicron Hell of Physics”)
SystemC needs a ceiling as well as a floor

System and SW Modeling:
UML, SDL, etc.
-Talks on UML and SystemC later

Mere Implementation!!
VHDL, Verilog, SystemVerilog
How the Industry Looks at the Many Language Choices

A Single Language Alone Cannot Effectively Cover All of the Design Flow
System Level Modeling in SystemC 2.0

• Real potential for SystemC is to be the industry standard language for system level design, verification and IP delivery for both HW and SW.

• Towards this goal, SystemC 2.0 supports generalized modeling for communication and synchronization with *channels*, *interfaces*, and *events*. Hardware signals are now modeled as a specialization of channels.

• System level extensions in SystemC 2.0 support transaction-level modeling, communication refinement, executable specification modeling, HW/SW co-design.

• Ability to refine HW portions of design to RTL level within a single language is a unique strength of SystemC, as is the fixed point modeling capability, and easy integration of existing C/C++ models.
What are Companies Doing Today with SystemC?

• A few companies are using SystemC for RTL modeling, but this is not where the real interest is.

• Many companies are in the process of replacing in-house C/C++ system level modeling environments with SystemC.

• Many companies view SystemC as both a modeling language and a modeling “backplane” (e.g. for ISS integration).

• A number of companies have completed TLM & TBV modeling efforts using SystemC 2.0 and are very excited & interested. Some of the results are starting to be made publicly available. Some companies are about to announce that they will provide system-level IP using SystemC.

  – For example, see on-line ARM presentation with audio:

SystemC EDA Products –OSCI Web Pages

- Actis Design, LLC: AccurateC™ SystemC Language Rule Checker & Rule Generator
- Adelante Technologies: AIRT Builder Automatic HDL generation from SystemC input
- Adelante Technologies: AIRT Designer Automatic HDL generation from SystemC input
- Adelante Technologies: AIRT Library Fixed point algorithm using SystemC input
- Ascend Design Automation: Verilog2SC High Performance Verilog to SystemC Translaser
- Axys Design: MaxSim™ Developer Suite - Featuring SystemC v2.0 Support Tool for modeling and verification of multi-core SoCs
- Blue Pacific Computing: BlueWave Simulation GUI Simulation waveform display
- Cadence Design Systems: Signal Processing Worksystem® (SPW) 4.8 Signal Processing Worksystem® (SPW) 4.8 with SystemC 2.0 co-simulation
- Co-Design Automation, Inc: SYSTEMSIM™ Multilingual simulator, supporting Superlog, C, C++ and SystemC, or co-simulation
- CoWare: N2C™ Napkin to Chip in Half the Time. Full SystemC Co-Design Environment
- Dynalith: iSave In System Algorithm Verification System - In system emulation of algorithms expressed in SystemC modeling platform.
- Emulation & Verification Engineering (EVE): Zebu Hardware-assisted co-modeling product
- Forte Design Systems: Cynthesizer High-level synthesis from SystemC
- Forte Design Systems: ESC Extended SystemC Library Implementation
- Forte Design Systems: GigaScale Verification Full SystemC Verification Environment
- Future Design Automation: Design Prototyper™ High Level Behavioral Synthesis
- Innoveda/Summit Design: Visual Elite™ System Level Design Design environment for defining and verifying systems
- LisaTEK: Embedded Processor Tool Suite Tool suite for the automated design of embedded processors used in SoCs
- Mentor Graphics: Seamless® C-Bridge™ Hardware/Software co-verification environment.
- Synopsis: CoCentric® Fixed Point Designer Automated floating point to fixed point algorithm conversion
- Synopsis: CoCentric® SystemC Compiler Synthesis of hardware from SystemC models
- Synopsis: CoCentric® SystemC HDL CoSim High performance co-simulation tool for VCS / Scirocco / MTI
- Synopsis: CoCentric® System Studio Commercial SystemC simulator and design environment
- Tension Technology EDA Ltd.: VTOC (Verilog to C++/SystemC Compiler) Converts Verilog design descriptions to SystemC
- TNI-Valiosys: Cosimate Simulation backplane that allows a number of heterogeneous simulators and models (including SystemC) to be connected together.
- TNI-Valiosys: VHDL2SystemC A translator that translates VHDL synthesizable RTL into SystemC 2.0. A Verilog version in the works.
- TOPS-SLD: TS-SLD Generates cycle accurate C++ SystemC compatible simulation models, which can be incorporated into the LSI/System design flows.
- Verilati: Specman Elite™ Interface to system-level designs developed with SystemC
- Veritools: SuperC™ A very fast SystemC simulator that writes a highly compressed data format.
- Xilinx / EVE: ZeBu Co-Simulation Accelerated Co-simulation of designs driven by C/C++/SystemC testbenchs
- Xilinx: FPGA SystemC Flow Automated path into a platform FPGA implementation from SystemC

### Taxonomy of SystemC products from OSCI web pages

- Total number of products = 32
  - Commercial SystemC Simulators: 3
  - Co-Simulators: 4
  - Links to Emulation: 4
  - Synthesis: 6
  - HDL to SystemC Model Converters: 4
  - SystemC Extended Libraries: 2
  - Analysis, Display, Verification and Checkers: 3
  - System Level Modelling and Design Tools: 6

\[\frac{1}{3}\]
Example of SystemC Tools: Cadence

Snapshot, 22 October 2002
Cadence Vision

- Provide a set of tools that allows the user to transition/refine from high level System Design through implementation with reusable TestBench without loss of productivity
What is Cadence doing with SystemC?

- Cadence is very active within the SystemC standards organization (OSCI).
- Cadence is incorporating SystemC into its tools for system design and verification.
- We have announced several products related to SystemC:
  - NC-SystemC Integration
    - SystemC becomes another “native language” within NCSim. Provides a single unified Verilog/VHDL/SystemC design hierarchy, single unified graphical debugger for all languages, very high performance.
  - Testbuilder-SC (more on this in a later presentation)
    - Testbuilder Verification capabilities built natively on top of SystemC, available as open source & standardised through OSCI. TB-SC provides transaction recording, viewing & analysis and tight integration with NC-SystemC.
  - SPW
    - SPW-SystemC co-simulation capability.
Cadence Complete Vision – SPW - Dataflow

- **SPW capabilities**
  - High performance simulation for dataflow to base platform
  - Adding algorithmic libraries and analysis capabilities
- **Mix control and datapath in simulation using SystemC language**
- **System Level Design capabilities**
  - Architectural/communication performance modeling and libraries
- **SW verification tools**
SPW 4.8 & SystemC: Integrated Design & Verification

HDL – Verilog, VHDL

Control Entry
Signal Analysis
Data Path Entry
Block Wizard
Verilog AMS

Software on ISS
Integrated Debug
Cross Debug C/C++/RTL
HDL Waveform
Signal Analysis
SPW 4.8 & SystemC - Models and Systems

- Add your own Intellectual Property
- Easily create C++ Polymorphic blocks
- Import C, C++, MATLAB, Verilog, VHDL, Verilog AMS, SystemC 2.0
- Auto symbol and parameter creation done by block wizard

- Through participation in standardisation Organisations
- 3GPP
  - V3.6.0 release 4 shipped -
- GSM / EDGE
  - GSM Enhanced Circuit Switch Data System
  - GSM EGPRS Packet Switch Data System
- WLAN
  - IEEE802.11g available end of April’02
  - Practical receivers for 11a available in July’02
- Smart Antenna
  - Available end of April’02
NC-SIM Simulation Environment

• NC-SIM Simulator
  – Support for mixed language (SystemC 2.0, Verilog, VHDL)
  – Abstract modeling and analysis possible with SystemC 2.0
  – Strong debugging environment for mixed-language designs

• Verification Environment - TestBuilder and Verification Cockpit
  – Gives user additional testbench capabilities with TestBuilder additions on top of SystemC all in free C/C++ constructs; merge existing C/C++ open source standards into 1 strong standard (more on SystemC Verification extensions – later presentation)
NC-Sim Simulation Environment

NC-SystemC

Same NC-Sim simulation and debug environment
- Fast simulation
- Mixed language support: SystemC, Verilog, VHDL, AMS
- Multiple levels of abstraction and analysis
- Strong debugging environment for mixed language designs and C/C++
Navigator – Source Code Browsing
SystemC Source Code – SystemC Top

![Image of Cadence NC Verilog software interface with SystemC source code]

Navigator Panel:
- Scope: Displays project hierarchy.
- Objects: Lists available objects.
- File: Navigation buttons for files.
- Options: Various editing options.

Editor Panel:
- Code editor with SystemC source code.
- Debugger console.
- Include files and compilation commands.

Command Line:
- Linux commands for building and debugging SystemC projects.
- Examples include:
  - `source /sw/avx/designs/tools/1SV_3.9/tools/linx/files/avxweeks`
NC-Sim Simulation Environment Plus SystemC Verification extensions

NC-SystemC + TB-SC

Add to NC-Sim simulation and debug environment
- Transaction recording
- Complete database transaction analysis
- Constraint solver, randomization, etc
- TxE/SimVision
- Re-use same testbench from System Design through implementation
Cadence SystemC/C/C++ Vision

Transaction Level Analysis capabilities put on top of SystemC
Cadence SystemC/C/C++ Vision

Transaction Level Analysis capabilities put on top of SystemC
Transaction Explorer (TxE) for complete/complex database query
Results of TxE Search
All Ethernet Frames Destined for WAN - Illuminated
# How Do The SystemC/RTL Domains Communicate?

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
<th>Illustration</th>
<th>Domain</th>
</tr>
</thead>
</table>
| Signal      | Instantaneous Value Change  
Restricted Set of Possible Values                                        |              | RTL (Verilog/VHDL)    |
| Transaction | Value Change Takes Time  
Value Unrestricted – User-Defined                                           |              | Abstract (SystemC)    |

Transaction → Signal

Transaction → Transaction
SystemC Use Models
Functional Verification of Hardware

RTL Function Verification Questions

Is the final version of the design error-free?

Has all of the functionality of the design been proven to work correctly?

How can the verification engineer be sure that an error found in the design is a logical error instead of a performance error?
Design Space Exploration in System-Level Design

System-Level Design Questions

Do the components within the design work properly together?
How can the design be globally optimized?

How can the system-level design engineer be confident that the results obtained from design exploration will hold true when the system is implemented?
Validation of Transaction-Level Models
Embedded Software Verification Method: Hardware Model Abstraction

Abstract Hardware Model in SystemC 2.0
Faster Simulation

Software Debugging Environment
Implement Abstract Module in RTL / Legacy RTL Method: Top-Down / Bottom-up Design
Conclusion – Future Tool Possibilities

• A *Personal* View:

  – Links to Implementation are important
    – But the world has not figured out behavioural synthesis yet
    – And using SystemC as an RTL entry vehicle is not the best approach
  – System level modelling, analysis and refinement is still not a well-understood and well-adopted approach
    – This is where users of SystemC need to spend most of their time, experimenting and working out methodologies
  – Calls out for:
    – Methodology-driven design flows
    – Analysis capabilities
    – Design space exploration concepts
    – Flows from higher level modelling e.g. UML, and links to embedded SW
  – From the system level designer viewpoint, this is the most useful area for tool development