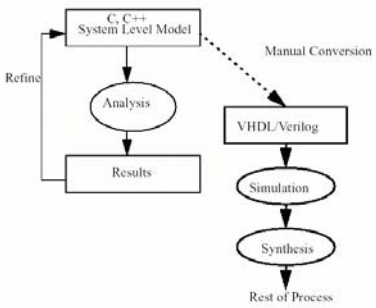
System C

- Event-driven simulation language based on C++
- Supported by Synopsys, Cadence
- Freely available implementation (2.0) available at http://www.systemc.org
  - Requires registration (just email address, username, password)
  - Distribution has source only, several common Unix platforms supported plus NT
- One argument for C-based Simulation languages is mainly speed of execution because of native-code compilation
  - Most VHDL/Verilog simulators compile to an intermediate code (a byte code) which is then interpreted
  - Cadence supports a native-code compiler for Verilog, but costs extra.
  - System C uses normal C++ compilers + library to compile to native code. Claim is anywhere from 10x to 100x faster than byte code approaches.

Other Arguments for SystemC

- For high level behavioral modeling, have already discussed that Verilog is not suitable because cannot define own data types
- VHDL is suitable, but many people just don't like it for one reason or another
  - Too complex, not enough like traditional languages, not enough like Verilog, etc…
- So, many companies have rolled their own high-level simulation environment based on C or C++ or some other traditional programming language
- SystemC attempts to replace those proprietary simulation environments with a open standard that everybody can contribute to
  - Would like to become the standard language for high-level behavorial modeling and behavorial synthesis
  - Also can do RTL modeling in SystemC, and would also like to replace VHDL/Verilog at that level (if you do high leveling modeling in SystemC, why not also do RTL level modeling as well?).

SystemC would like to replace all of this with just SystemC.

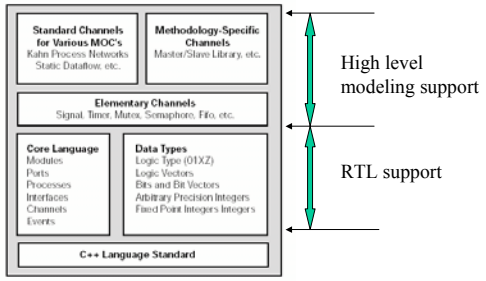


VHDL adherents say 'why bother?' since VHDL can already do this.

SystemC adherents always bring up the speed advantage of native code compilation.

SystemC 1.0 supported RTL modeling, Version 2.0 added high level modeling support.

Architecture of V2.0



Continuous Time Modeling vs Event Driven (Discrete Event)

- Continuous time modeling is what is done in Spice
  - Models analog systems
- Event driven or Discrete Event modeling is what is done in Verilog, VHDL, SystemC
- Holy Grail is a standard simulation environment that marries continuous time + discrete time simulation -- known as mixed signal modeling
- Extensions have been proposed to all languages to support mixed continuous time + discrete event simulation
  - Prototype implementations of continuous time simulation in VHDL, Verilog has been done (anaVHDL, VHDL-AMS, Verilog-AMS)
  - EDA Standard committees exist for VHDL-AMS, Verilog-AMS
  - Other proprietry languages/simulators exist (Spectre from Cadence)
- Cadence has a product called AMS Designer that supports Verilog-AMS