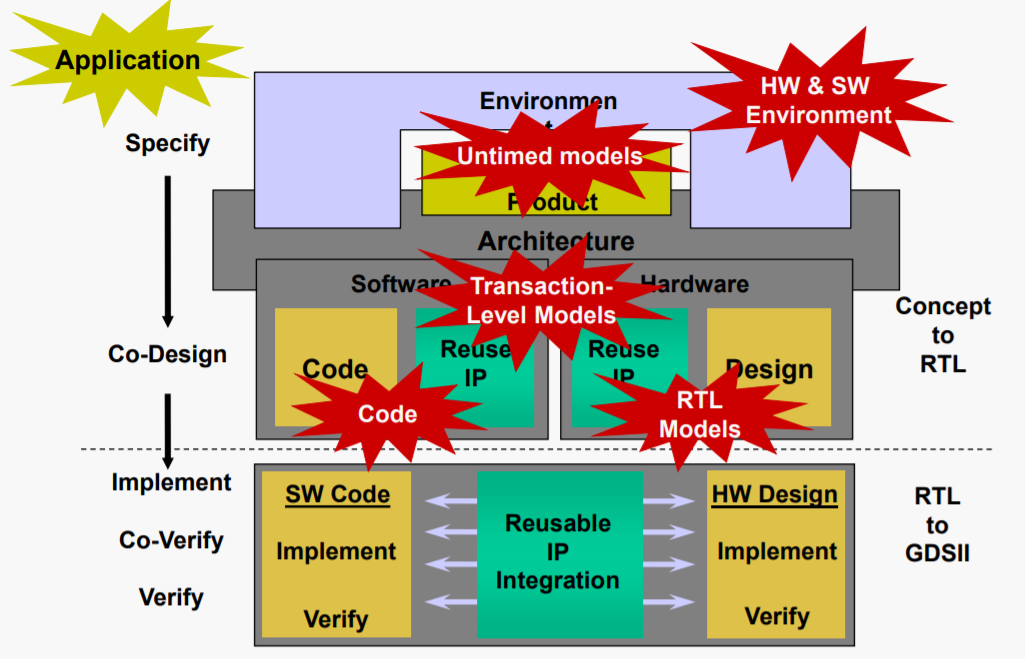
SOC vs. SIP

* SOC: system modules are integrated on the same die
  + Fabrication must be compatible with standard CMOS processes
* SIP: multi-die multi-process “stackable” module integration
  + More cost effective than single-die single-process approach

Hardware & Software Design



Verilog/VHDL not enough?

* Systems become more complex, pushing designers to design and verify at higher levels of abstraction
  + Software is playing an increasing role in system design
    - Need to support early software development, integration with existing C/C++ code, and HW/SW codesign
  + Need a design language that supports system-level IP delivery/integration
* Build on C/C++ so that:
  + Extensive C/C++ infrastructure (compilers, debuggers, language standards, etc) can be reused
  + Users’ existing knowledge of C/C++ can be leveraged
  + Integration with existing C/C++ code is easy

SystemC

* SystemC:
  + Enables early exploration of system-level design trade-offs
  + Enables early and fast system-level verification
* SystemC is a subset of C++ that model’s hardware
  + Language definition is standardized by IEEE
  + It provides a simple form of concurrency
    - Cooperative multitasking
  + SystemC simulation libraries are freely distributed
    - Commercial compilers available for translating SystemC programs into gate netlists
  + SystemC program consists of module definitions plus a top-level main function that starts the simulation

**Briefly State the benefits and drawbacks of SystemC?**

Some of the benefits of SystemC are that since it is a subset of C++ for modelling hardware so it has the benefits of having access to existing C/C++ infrastructure (Compilers, debuggers etc). Users can use their existing knowledge of c & c++ to make using SystemC intuitive and integrate into existing software solutions. This allow for early and fast system level verification using a free language and with an IEEE standard.

One of the drawbacks is that there are quirks with the syntax that can easily be forgotten. One such example is not having a space between consecutive ‘>’ as the compilers will recognize it as a shift operation when it is a variable type. Another error is not ending a SC\_METHOD with a semicolon after the close brace. Besides syntax drawbacks, as the system you are modelling becomes more complex simulation times will greatly increase.