Design and Verification with SystemC

Bhanu Kapoor, Ph.D.

Consultant/Owner, Mimasic

Designing Big Digital Systems

- Every system company was doing this differently
- Every system company used its own simulation library

- “Throw the model over the wall” approach makes it easy to introduce errors

- Problems:
  - System designers don’t know Verilog or VHDL
  - Verilog or VHDL coders don’t understand system design

Standard Methodology for ICs

- System-level designers write a C or C++ model
  - Written in a stylized, hardware-like form
  - Sometimes refined to be more hardware-like
- C/C++ model simulated to verify functionality
- Model given to Verilog/VHDL coders
- Verilog/VHDL model synthesized
- Offered by the Open SystemC Initiative (OSCI)
- Includes constructs for testing/simulation as well as synthesis.

Idea of SystemC

- C and C++ are being used as ad-hoc modeling languages
- Why not formalize their use?
- Why not interpret them as hardware specification languages just as Verilog and VHDL were?
- SystemC developed at Synopsys to do just this

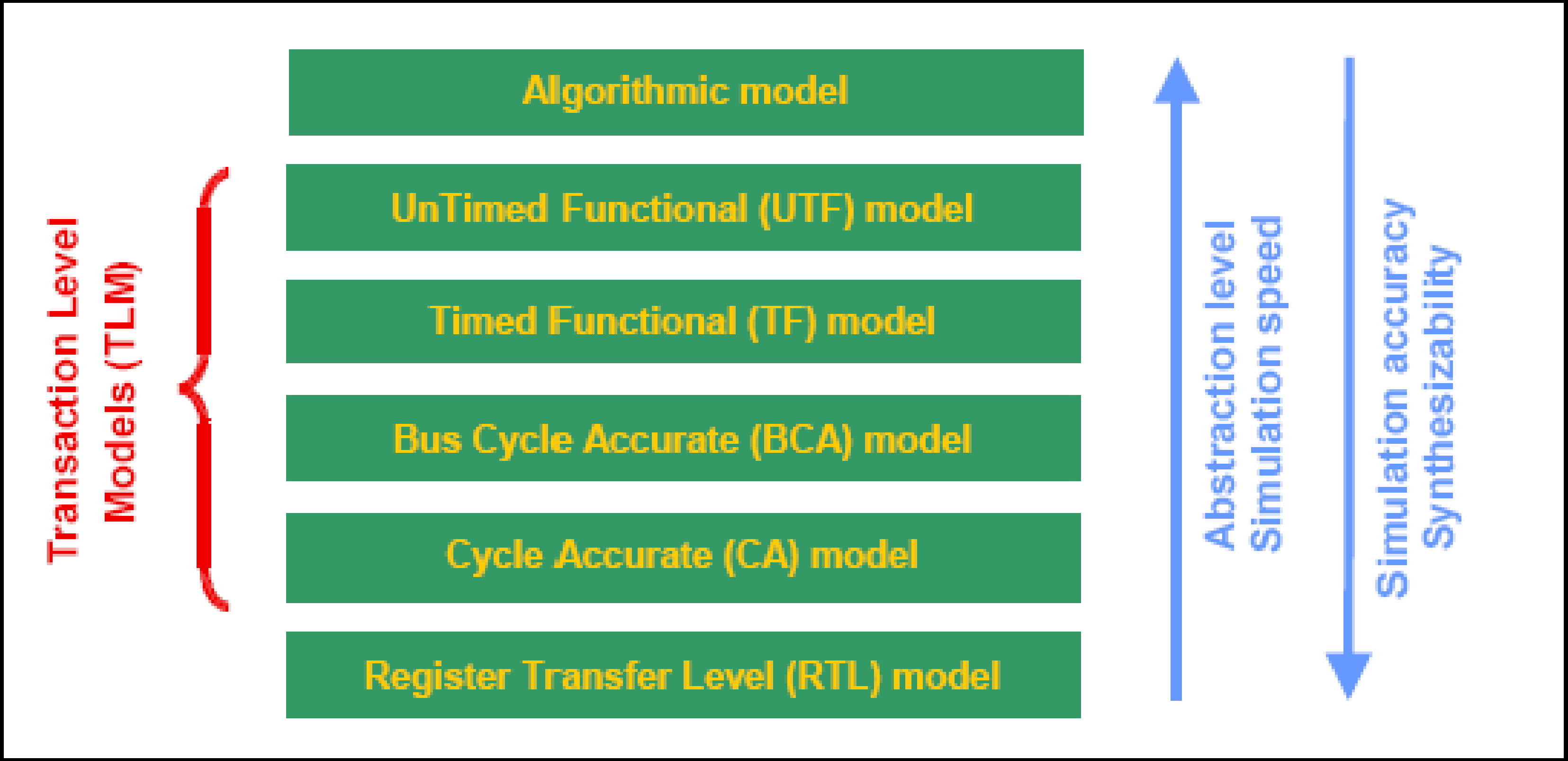
What Is SystemC?

- A subset of C++ that models/specifies synchronous digital hardware
- A collection of simulation libraries that can be used to run a SystemC program
- A compiler that translates the “synthesis subset” of SystemC into a netlist

What Is SystemC?

- Language definition is publicly available
- Libraries are freely distributed
- Compiler is an expensive commercial product
- See www.systemc.org for more information

System C model


Algorithmic model
UnTimed Functional (UTF) model
Timed Functional (TF) model
Bus Cycle Accurate (BCA) model
Cycle Accurate (CA) model
Register Transfer Level (RTL) model
Transaction Level Models (TLM)
Abstraction level
Simulation speed
Simulation accuracy
Synthesizability

source: Synthesizable C

mimasic

Quick Overview

- A SystemC program consists of module definitions plus a top-level function that starts the simulation
- Modules contain processes (C++ methods) and instances of other modules
- Ports on modules define their interface
  - Rich set of port data types (hardware modeling, etc.)
- Signals in modules convey information between instances
- Clocks are special signals that run periodically and can trigger clocked processes
- Rich set of numeric types (fixed and arbitrary precision numbers)

SystemC Tutorial

- Gets you up to speed with System C language, RTL and Transaction-Level
- Hands-on experience with Open Source SystemC Simulator on some concepts
- Includes design and verification situations to stimulate in-depth knowledge of the language