SpecC Design Environment

Abstraction Levels

- Requirements (attributes + constraints)
- Specification (untimed)
- Architecture (execution delays)
- Communication (timed)
- Implementation (cycle-accurate: RTL+IS)
- Gate Level (sub-cycle delays)
- Layout (continuous time)
- Manufacturing (real time)
Validation Levels

- Requirements (attributes + constraints)
  - Specification (untimed)
  - Architecture (execution delays)
  - Communication (timed)
  - Implementation (cycle-accurate: RTL+IS)
  - Gate Level (sub-cycle delays)
  - Layout (continuous time)
  - Manufacturing (real time)

- Functionality
  - Structure
  - Protocols
  - Performance
  - Clock cycle
  - Spacing

SpecC Scope

- Requirements (attributes + constraints)
  - Specification (untimed)
  - Architecture (execution delays)
  - Communication (timed)
  - Implementation (cycle-accurate: RTL+IS)
  - Gate Level (sub-cycle delays)
  - Layout (continuous time)
  - Manufacturing (real time)

- Specification (SW+HW)
  - VHDL
  - Verilog (HW)
  - EDIF
  - GDS II
  - Masks
SpecC Methodology

- 4 models
- 4 databases
- 3 refinements
- 2 flows
- 1 language
- “0” effort

SpecC Refinement

Capture

Specification model

Arch. refinement

Architecture model

Comm. refinement

Communication model

Impl. refinement

Implementation model
Refinement User Interface (RUI)

- Algorithm selection
- Browsing
- Spec. optimization
- Allocation
- Beh. partitioning
- Scheduling / RTOS
- Protocol selection
- Channel partitioning
- Arbitration
- Cycle scheduling
- Protocol scheduling
- SW assembly

Capture

Specification model

- Arch. refinement
- Architecture model

Communication model

- Comm. refinement
- Communication model

Specification model

- Impl. refinement
- Implementation model

User Feedback

- Algorithm selection
- Browsing
- Spec. optimization
- Allocation
- Beh. partitioning
- Scheduling / RTOS
- Protocol selection
- Channel partitioning
- Arbitration
- Cycle scheduling
- Protocol scheduling
- SW assembly

Capture

Profiling

Profiling data

Arch. refinement

Architecture model

Estimation

Architecture model

Estimation results

Comm. refinement

Communication model

Estimation

Communication model

Estimation results

Impl. refinement

Implementation model

Implementation model
Conclusions

- **SpecC methodology fits a new paradigm**
  - SW = HW = SOC = Embedded System

- **SpecC methodology enabling technology**
  - e-Design
  - IP trading
  - SW/HW co-design
  - Mass customization

- **4 SpecC engines**
  - Modeling
  - Refinement
  - Exploration
  - Synthesis