ECE 298: System-on-Chip Description and Modeling Lecture 6

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 6: Overview

- Homework Assignment 3
  - Group presentations
  - Discussion
- SoC Exploration and Refinement
  - Architecture Exploration
  - Communication Exploration
  - Implementation Exploration
  - Refinement Decisions
  - Example

Lecture 6: Overview

- Homework Assignment 3
  - Group presentations
  - Discussion
- SoC Exploration and Refinement
  - Architecture Exploration
  - Communication Exploration
  - Implementation Exploration
  - Refinement Decisions
  - Example

Homework Assignment 3

- Tasks
  - Task 1:
    - Complete the MPEG-Audio Specification Model such that the model ...
      - ... contains all necessary code for the decoder
      - ... contains a test bench with stimulator, DUT, and monitor
      - ... compiles successfully with the SpecC compiler
      - ... simulates successfully
  - Task 2:
    - For the decoder (DUT), create the behavioral hierarchy necessary for a well-defined Specification Model
      - Granularity: each function becomes a behavior
      - Hierarchy: try to mimic the given functional hierarchy
      - Concurrency: add explicit concurrency wherever possible
      - Communication: use standard channels or local variables

Homework Assignment 3

- Group presentations, Discussion
  - Task 1:
    - Complete the MPEG-Audio Specification Model such that the model ...

STATUS

Task 1a: – ... contains all necessary code for the decoder
Task 1b: – ... contains a test bench with stimulator, DUT, and monitor
Task 1c: – ... compiles successfully with the SpecC compiler
Task 1d: – ... simulates successfully

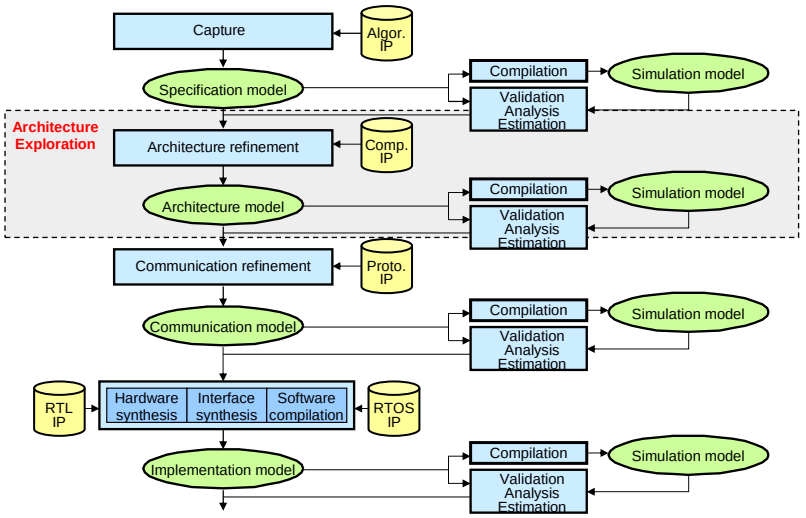
  - Task 2:
    - For the decoder (DUT), create the behavioral hierarchy necessary for a well-defined Specification Model

Task 2a: – Granularity: each function becomes a behavior
Task 2b: – Hierarchy: try to mimic the given functional hierarchy
Task 2c: – Concurrency: add explicit concurrency wherever possible
Task 2d: – Communication: use standard channels or local variables

Lecture 6: Overview

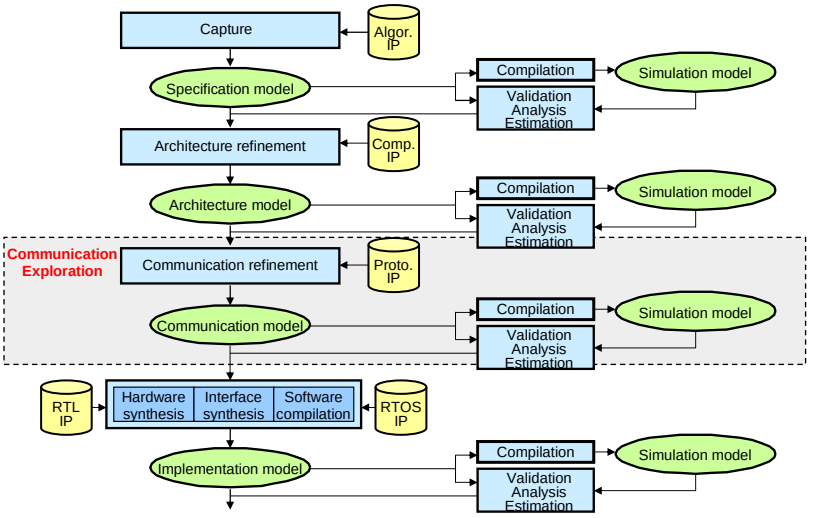
- Homework Assignment 3
  - Group presentations
  - Discussion
- ➢ SoC Exploration and Refinement
  - Architecture Exploration
  - Communication Exploration
  - Implementation Exploration
  - Refinement Decisions
  - Example

SoC Architecture Exploration



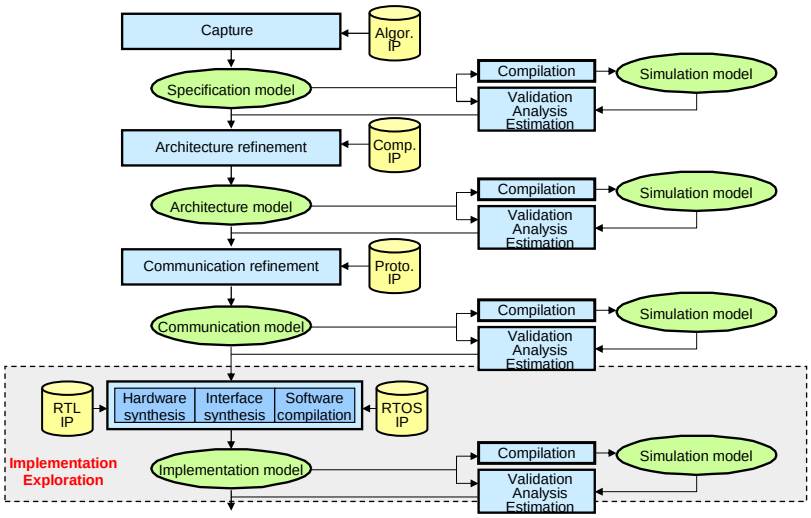
ECE298: SoC Description and Modeling, Lecture 6 (c) 2004 R. Doemer 7

SoC Communication Exploration



ECE298: SoC Description and Modeling, Lecture 6 (c) 2004 R. Doemer 8

SoC Implementation Exploration



Refinement Decisions

- Step 1: Architecture Refinement
  - Allocation of Processing Elements (PE)
    - Type and number of processors
    - Type and number of custom hardware blocks
    - Type and number of system memories
  - Mapping to PEs
    - Map each behavior to a PE
    - Map each channel to a PE
    - Map each variable to a PE
  - Result:
    System architecture of concurrent PEs
    with abstract communication in channels

Refinement Decisions

- Step 2: Scheduling Refinement
  - For each PE, serialize the execution of behaviors to a single thread of control
  - Option (a): Static scheduling
    - For each set of concurrent behaviors, determine fixed order of execution
  - Option (b): Dynamic scheduling by RTOS
    - Choose scheduling policy, i.e. Round-robin or priority-based
    - For each set of concurrent behaviors, determine scheduling priority
  - Result:
    System model with abstract RTOS scheduler inserted in each PE

Refinement Decisions

- Step 3: Communication Refinement
  - Allocation of system busses
    - Type and number of system busses
    - Type of bus protocol for each bus (if applicable)
    - Number of transducers (if applicable)
    - System connectivity
  - Mapping of channels to busses
    - Map each communication channel to a system bus (or multiple busses, if applicable)
  - Result:
    Bus-functional model of the system

Refinement Decisions

- Step 4: Hardware Refinement (for HW PE)
  - Allocation of RTL components
    - Type and number of functional units
    - Type and number of storage units
    - Type and number of interconnecting busses
  - Scheduling
    - Basic blocks assigned to super-states
    - Operations assigned to clock cycles
  - Binding
    - Bind functional operations to functional units
    - Bind variables to storage units
    - Bind transfers to busses
  - Result:
    Clock-cycle accurate model of each HW PE
  - Output: Synthesizable Verilog description

Refinement Decisions

- Step 5: Software Refinement (for SW PE)
  - C code generation
    - For selected target processor
    - For selected target RTOS
  - Compilation to Instruction Set
    - for Instruction Set Simulation (ISS)
  - Assembly
  - Result:
    Clock-cycle accurate model of each SW PE
  - Output: downloadable object code

SoC Exploration and Refinement

- Example:
  - GSM Vocoder design in SCE
- Online demonstration ...