ECE 298: System-on-Chip Description and Modeling Lecture 5

Rainer Dömer

doemer@uci.edu

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

Lecture 5: Overview

- Homework Assignment 2
  - Discussion and Solution
- Homework Assignment 3
  - Tasks
- SoC Exploration and Refinement
  - Architecture Exploration
  - Communication Exploration
  - Refinement vs. Synthesis
  - Refinement-based Design Flow
  - Refinement Decisions
  - Example

Lecture 5: Overview

- Homework Assignment 2
  - Discussion and Solution
- Homework Assignment 3
  - Tasks
- SoC Exploration and Refinement
  - Architecture Exploration
  - Communication Exploration
  - Refinement vs. Synthesis
  - Refinement-based Design Flow
  - Refinement Decisions
  - Example

Homework Assignment 2

- Discussion and Solution
  - Administration
    - Accounts on **`alpha.eecs.uci.edu`** Login OK
    - RedHat Source Navigator Found OK
      - **`/opt/sourcenav-5.2b2/bin/snavigator`**
    - CECS System-on-Chip Environment Found OK
      - **`/opt/sce-20030530/bin/setup.csh`**
  - Task 1:
    - Make yourself familiar with the SpecC compiler
      - Use **`scc`** to compile and run the examples found in **`/opt/sce-20030530/examples/simple/`**
    - Online demonstration
      - (if necessary)

Homework Assignment 2

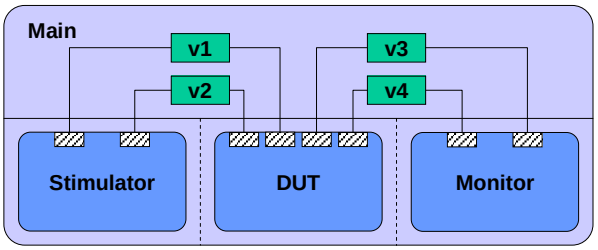
- Discussion and Solution
  - Task 2:
    - Make yourself familiar with the SoC Environment
      - Go through the initial steps of the SCE tutorial found in **`/opt/sce-20030530/doc/SCE_Tutorial/sce-tutorial.pdf`**
  - Online demonstration
    - (if necessary)

Homework Assignment 2

- Discussion and Solution
  - Task 3:
    - Create an initial Specification Model with test bench for the MPEG-Audio example
      - Create an MPEG-Audio model with Stimulator, Decoder, and Monitor behaviors
  - Open discussion
    - Issues with setup (e.g. **`Makefile`**)
    - Issues with SpecC compiler **`scc`** (e.g. initialization bug)
    - Issues with languages (SpecC, ANSI-C, GNU-C)
    - Issues with header files (e.g. **`signal.h, math.h`**)
    - Issues with channel library
    - Issues with file organization (e.g. **`#include`** vs. **`import`**)

Homework Assignment 2

- Discussion and Solution
  - Task 3: Status
    - How far are we from a working specification model?
      - Step 1: Code is collected and separated
      - Step 2: Code compiles with **scc**
      - Step 3: Test files with audio data are prepared
      - Step 4: Design simulates successfully



Lecture 5: Overview

- Homework Assignment 2
  - Discussion and Solution
- ➢ Homework Assignment 3
  - Tasks
- SoC Exploration and Refinement
  - Architecture Exploration
  - Communication Exploration
  - Refinement vs. Synthesis
  - Refinement-based Design Flow
  - Refinement Decisions
  - Example

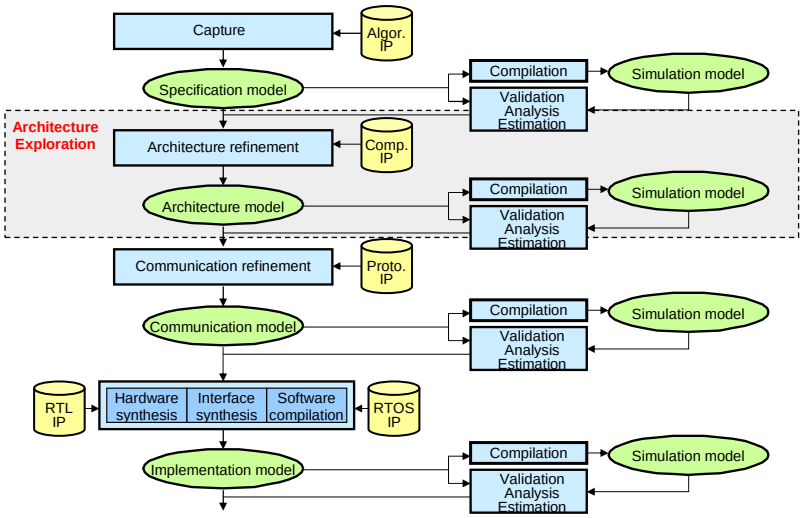
Homework Assignment 3

- 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

Lecture 5: Overview

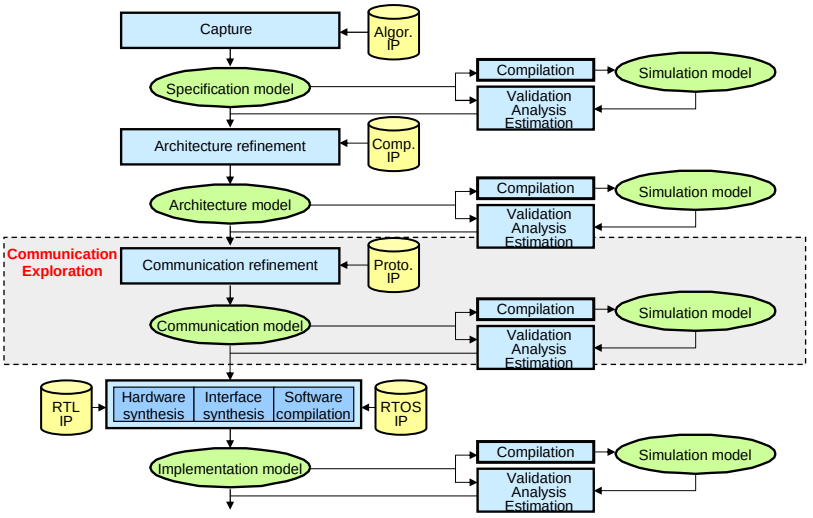
- Homework Assignment 2
  - Discussion and Solution
- Homework Assignment 3
  - Tasks
- ➢ SoC Exploration and Refinement
  - Architecture Exploration
  - Communication Exploration
  - Refinement vs. Synthesis
  - Refinement-based Design Flow
  - Refinement Decisions
  - Example

SoC Architecture Exploration



ECE298: SoC Description and Modeling, Lecture 5 (c) 2004 R. Doemer 11

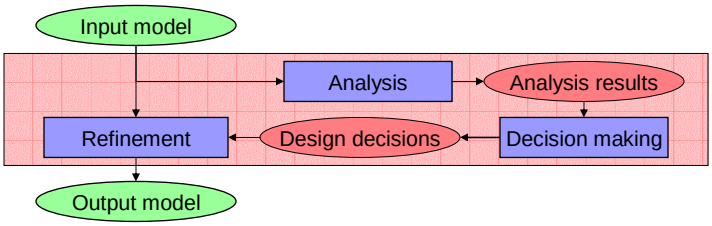
SoC Communication Exploration



ECE298: SoC Description and Modeling, Lecture 5 (c) 2004 R. Doemer 12

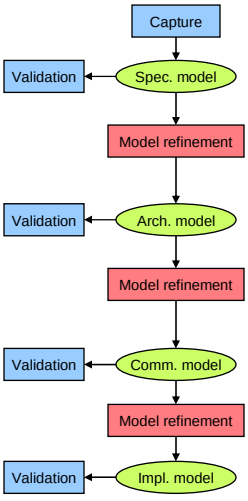
Refinement vs. Synthesis

- Every step in the design process includes
  - Model refinement
  - Model analysis
  - Decision making

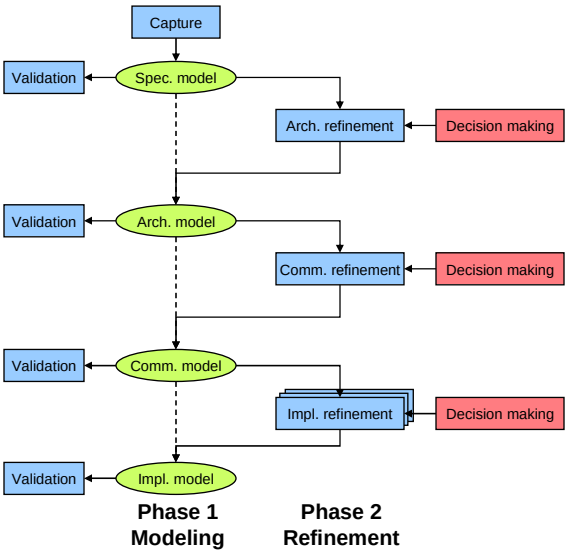


  - Synthesis = Analysis + Decision Making + Refinement

Refinement-based Design Flow

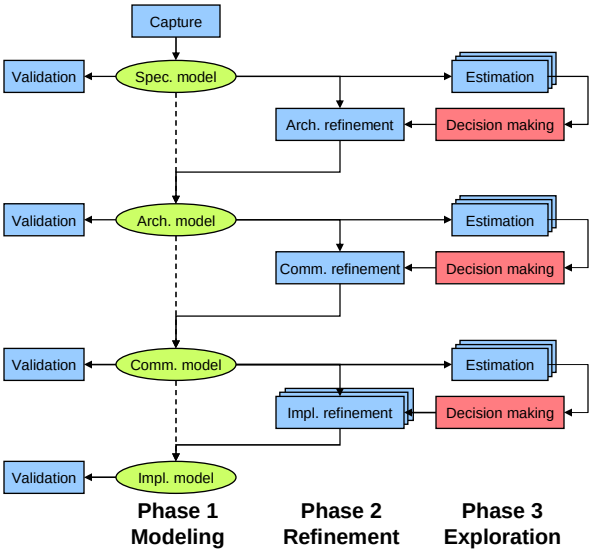


Refinement-based Design Flow



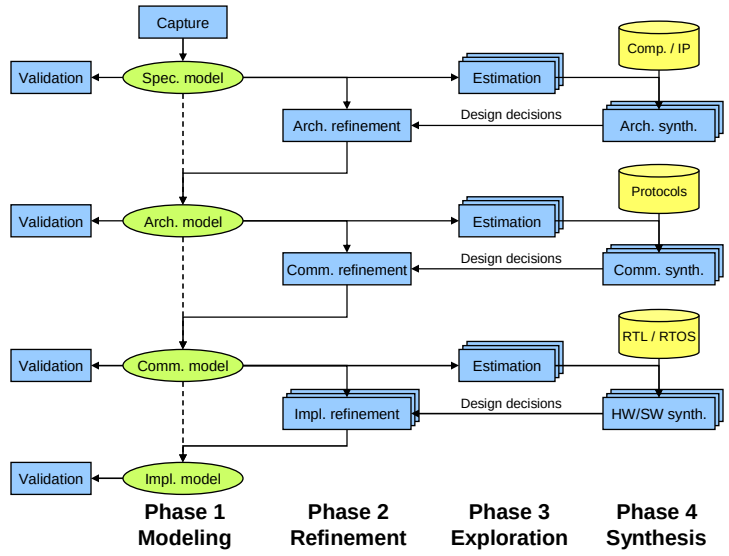
ECE298: SoC Description and Modeling, Lecture 5 (c) 2004 R. Doemer 15

Refinement-based Design Flow



ECE298: SoC Description and Modeling, Lecture 5 (c) 2004 R. Doemer 16

Refinement-based Design Flow



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

SoC Exploration and Refinement

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