EECS 222A:
System-on-Chip Description and Modeling
Lecture 3

Rainer Dömer

doemer@uci.edu

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

Lecture 3: Overview

- Homework Assignment 1
  - Discussion, Q&A
- System-on-Chip Specification
  - Essential issues
  - Top-down SoC design flow
  - Specification Model
  - Specification Modeling Guidelines
- System-on-Chip Environment (SCE)
  - Demonstration
  - Example: GSM Vocoder
- Homework Assignment 2
  - Tasks

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   2

Homework Assignment 1

- Administration
  - Server
    - **epsilon.eecs.uci.edu**
    - Intel Pentium CPU, 3.0 GHz, 1GB RAM
    - RedHat Linux (Fedora Core 4)
    - Access via secure shell protocol (**ssh**)
  - Accounts
    - User ID same as your UCI net ID
    - Password as discussed in class
  - SpecC Software (© by CECS, UCI)
    - SpecC Compiler and Simulator
      - **/opt/sce-20041007/bin/setup.csh**

Homework Assignment 1

- Task
  - Become familiar with
    - the SpecC compiler, **scc**
    - the SpecC simulator
  - Use **scc** to
    - compile,
    - simulate and
    - modify
    the examples found in
    /opt/sce-20041007/examples/simple/
- Deliverables
  - none (but be prepared for the next assignment)
- Due
  - next week (Week 3)

Essential Issues in Specification

- An Example ...

Proposed by the project team

Product specification

Product design by senior analyst

Product after implementation

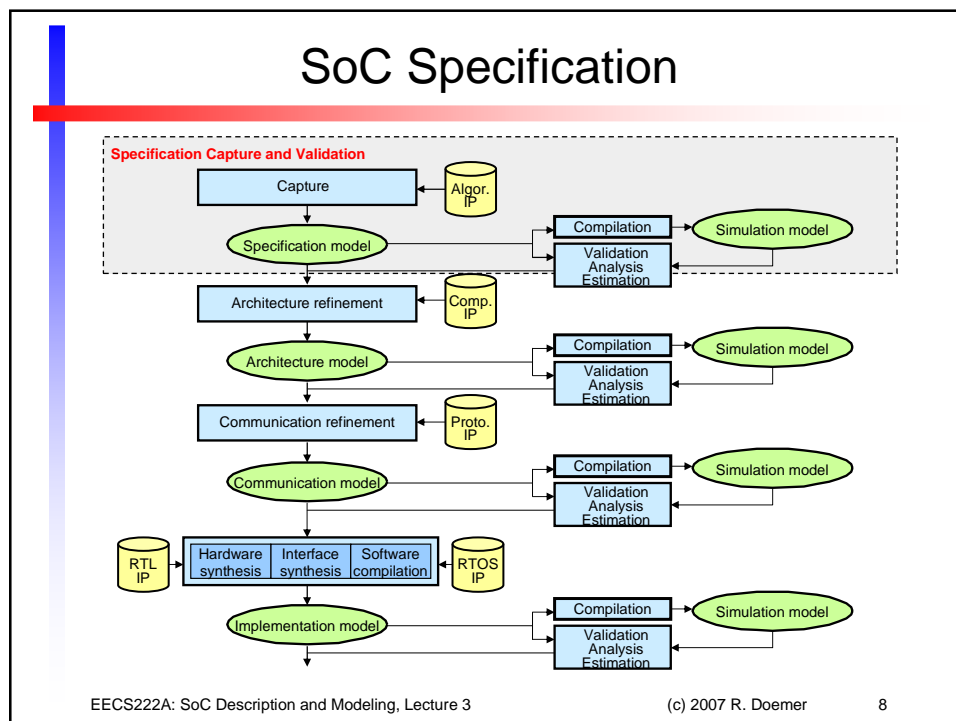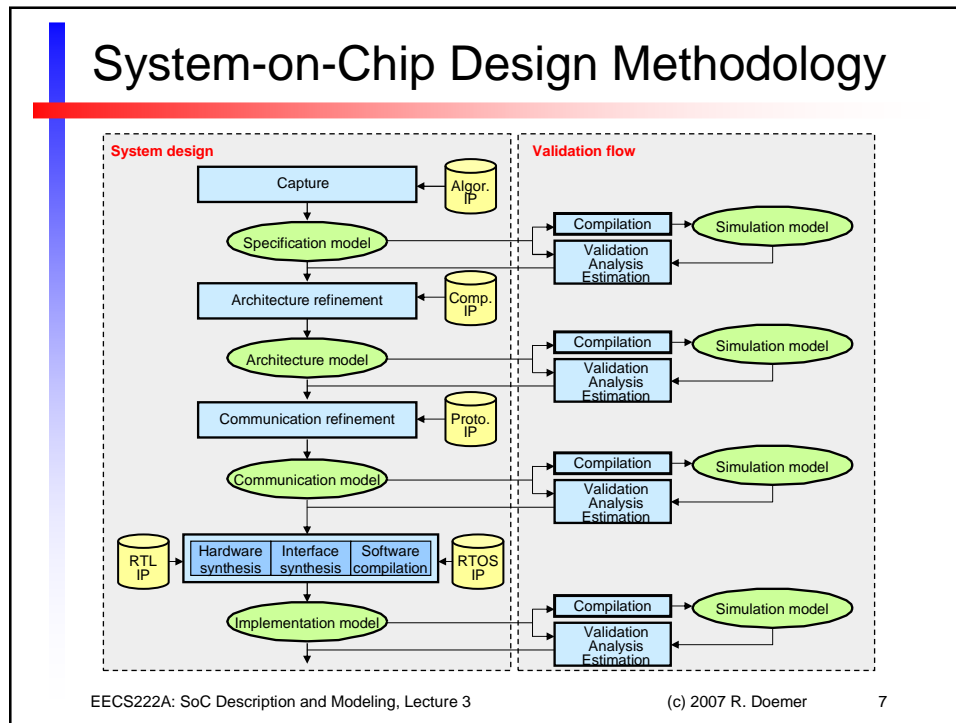Product after acceptance by user

What the user wanted

Source: unknown author

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   5

Top-Down SoC Design Flow

S T R U C T U R E

requirements — pure functional — transaction level — bus functional — RTL / IS

Product specification

Capture — Algor. IP

Specification model

Architecture refinement — Comp. IP

Architecture model

Communication refinement — Proto. IP

Communication model

RTL IP — Hardware synthesis | Interface synthesis | Software synthesis — RTOS IP

Implementation model

Manufacturing

T I M I N G

constraints — untimed — estimated timing — timing accurate — cycle accurate

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   6

System-on-Chip Design Methodology

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   7

SoC Specification

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   8

Specification Model

- High-level, abstract model
  - Pure system functionality
  - Algorithmic behavior
  - No implementation details
- No implicit structure / architecture
  - Pure behavioral hierarchy
- Untimed
  - Execution in zero (logical) time
  - Causal ordering
  - Synchronization

Specification model

Architecture refinement

Architecture model

Communication refinement

Communication model

Cycle-accurate refinement

Implementation model

(Source: A. Gerstlauer)

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   9

Specification Model

- Test bench
  - Main, Stimulator, Monitor
  - no restrictions in syntax and semantics (no synthesis)
- Design under test
  - DUT
  - restricted by syntax and semantic rules (synthesis!)

Main

| v1 | v3 |
| v2 | v4 |

Stimulator   **DUT**   **Monitor**

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   10

Specification Modeling Guidelines

- Functional and executable
  - "golden model" (first functional model in the design flow)
  - all other models will be derived from and compared to this one
- High abstraction level
  - no implementation details
  - unrestricted exploration of design space
- Separation of communication and computation
  - channels and behaviors
- Pure functional
  - no structural information
- No timing
  - exception: timing constraints

Specification Modeling Guidelines

- Computation: Behaviors
  - Hierarchy: explicit concurrency, state transitions, ...
  - Granularity: leaf behaviors = smallest indivisible units
  - Encapsulation: localization, explicit dependencies
  - Concurrency: explicitly specified (par, pipe, fsm, seq, …)
  - Time: un-timed, partial ordering
- Communication: Channels
  - Semantics: abstract communication, synchronization
                        (standard channel library)
  - Dependencies: explicit data dependency,
                        partial ordering, port connectivity

Specification Modeling Guidelines

- Example rules for SoC Environment (SCE)
  - Clean behavioral hierarchy
    - hierarchical behaviors:
      no code other than par, pipe, seq, fsm, try-trap, ... statements
    - leaf behaviors:
      no child behavior calls (basically pure ANSI-C code)
  - Clean communication
    - point-to-point communication via standard channels
    - ports of plain type or interface type, no pointers!
    - port maps to local variables or ports only
- Detailed rules for SoC Environment
  - CECS Technical Report 03-21:
    "*System-on-Chip Specification Style Guide*"
    by A. Gerstlauer, K. Ramineni, R. Doemer, D. Gajski
  - http://www.ics.uci.edu/~doemer/publications/CECS_TR_03_21.pdf

EECS222A: SoC Description and Modeling, Lecture 3 (c) 2007 R. Doemer 13

Specification Modeling Guidelines

- Example: C code conversion to SpecC
  - Functions become behaviors or channels
  - Functional hierarchy becomes behavioral hierarchy
    - clean behavioral hierarchy required
    - if-then-else structure becomes FSM
    - while/for/do loops become FSM
  - Explicitly specify potential parallelism
  - Explicitly specify communication
    - avoid global variables
    - use local variables and ports (signals, wires)
    - use standard channels
  - Data types
    - avoid pointers, use arrays instead
    - use explicit SpecC data types if suitable

EECS222A: SoC Description and Modeling, Lecture 3 (c) 2007 R. Doemer 14

System-on-Chip Environment (SCE)

- SCE Components:
  - Graphical frontend (`sce, scchart`)
  - Editor (`sced`)
  - Compiler and simulator (`scc`)
  - Profiling and analysis (`scprof`)
  - Architecture refinement (`scar`)
  - RTOS refinement (`scos`)
  - Communication refinement (`sccr`)
  - RTL refinement (`scrtl`)
  - Software refinement (`sc2c`)
  - Scripting interface (`scsh`)
  - Tools and utilities ...

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   15

SCE Main Window

EECS222A: SoC Description and Modeling, Lecture 3   Copyright © 2003 CECS   16

SCE Source Editor

EECS222A: SoC Description and Modeling, Lecture 3   Copyright © 2003 CECS   17

SCE Hierarchy Displays

EECS222A: SoC Description and Modeling, Lecture 3   Copyright © 2003 CECS   18

SCE Compiler and Simulator

SCE Profiling and Analysis

Specification Model Example

- GSM Vocoder
    - Enhanced full-rate voice codec
    - GSM standard for mobile telephony (GSM 06.10)
    - Lossy voice encoding/decoding
        - Incoming speech samples @ 104 kbit/s
        - Encoded bit stream @ 12.2 kbit/s
        - Frames of 4 x 40 = 160 samples (4 x 5ms = 20ms of speech)
    - Real-time constraint:
        - max. 20ms per speech frame
          (max. total of 3.26s for sample speech file)
    - SpecC specification model
        - 29 hierarchical behaviors (9 par, 10 seq, 10 fsm)
        - 73 leaf behaviors
        - 9139 formatted lines of SpecC code
          (~13000 lines of original C code, including comments)

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   21

Homework Assignment 2

- Administration
    - Server
        - **epsilon.eecs.uci.edu**
        - Intel Pentium CPU, 3.0 GHz, 1GB RAM
        - RedHat Linux (Fedora Core 4)
        - Access via secure shell protocol (**ssh**)
    - Accounts
        - User ID same as your UCI net ID
        - Password as discussed in class
    - SpecC Software (© by CECS, UCI)
        - System-on-Chip Environment
            - **/opt/sce-20041007/bin/setup.csh**
            - **/opt/sce-20060301/bin/setup.csh**

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   22

Homework Assignment 2

- Task
  - Become familiar with
    - the System-on-Chip Environment (SCE)
  - Follow the initial steps in the SCE Tutorial
    - `/opt/sce-20041007/doc/SCE_Tutorial/`
      `sce-tutorial.pdf`
- Deliverables
  - none (but be prepared for the next assignment)
- Due
  - next week (Week 4)

EECS222A: SoC Description and Modeling, Lecture 3   (c) 2007 R. Doemer   23