

ECE 298: System-on-Chip Description and Modeling Lecture 2

Rainer Dömer

doemer@uci.edu

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

Lecture 2: Overview

- Introduction to System-on-Chip
 - SoC design flow
 - Computation vs. communication
 - Intellectual Property (IP)
- System-on-Chip Design Methodology
 - Specification model
 - Architecture model
 - Communication model
 - Implementation model

Introduction to System-on-Chip

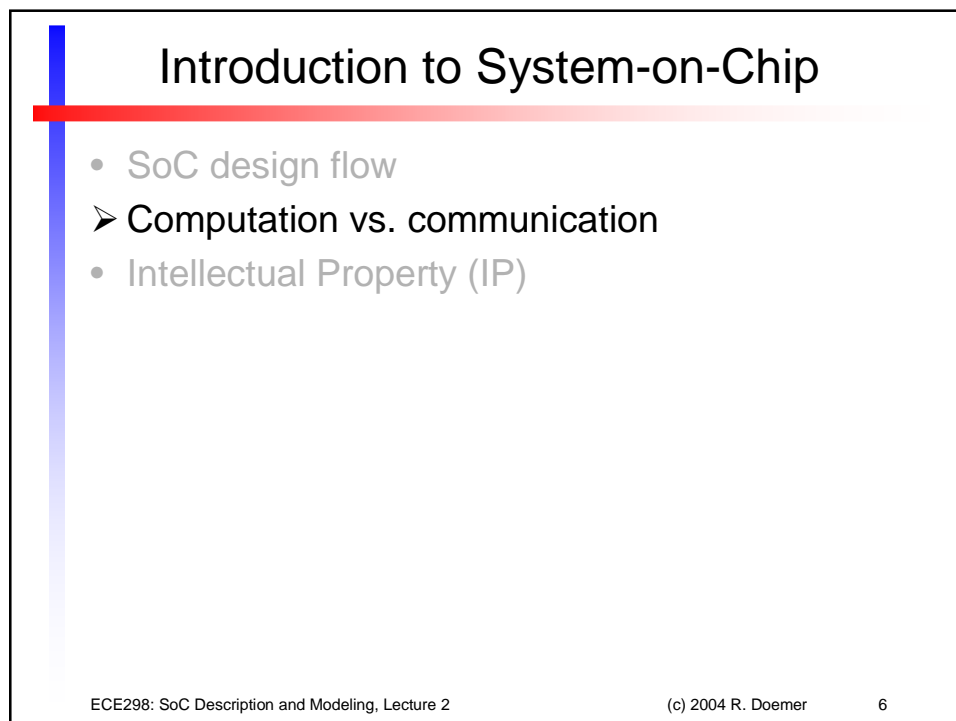
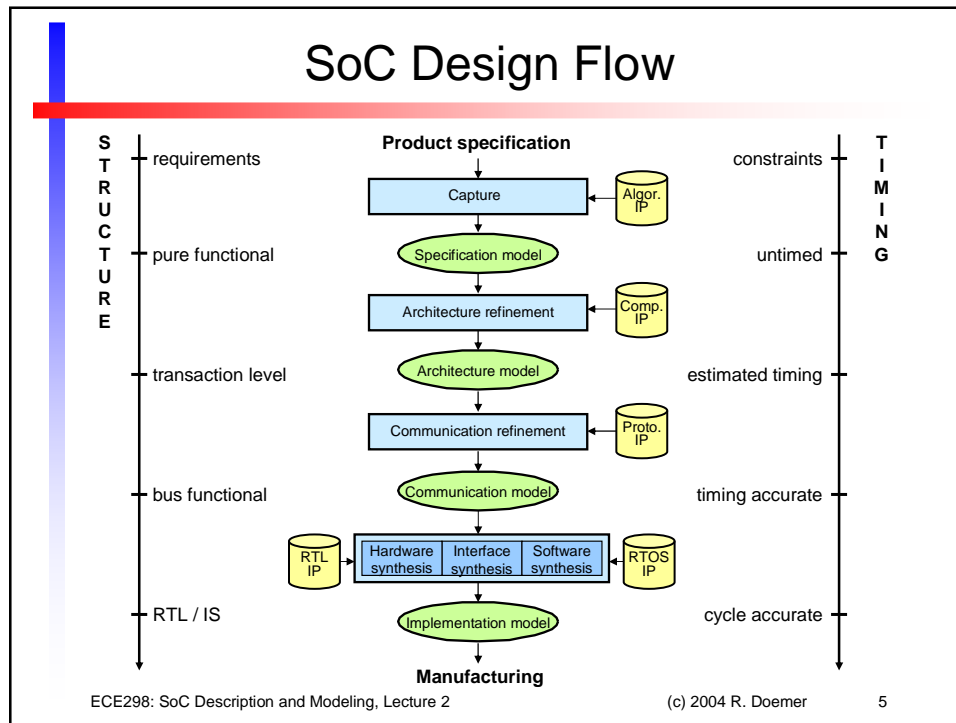
- SoC design flow
- Computation vs. communication
- Intellectual Property (IP)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 3

Introduction to System-on-Chip

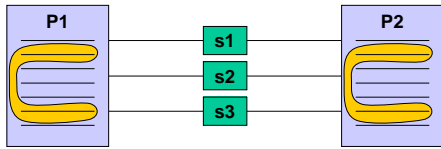
- SoC design flow
- Computation vs. communication
- Intellectual Property (IP)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 4



Computation vs. Communication

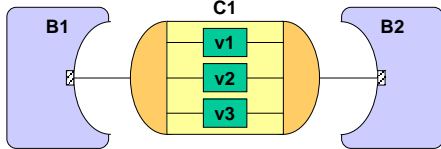
- Traditional model
 - Processes and signals
 - Mixture of computation and communication
 - Automatic replacement impossible



ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 7

Computation vs. Communication

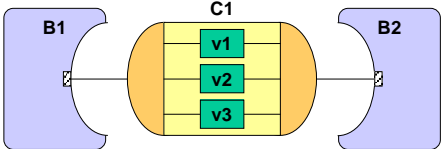
- Traditional model
 - Processes and signals
 - Mixture of computation and communication
 - Automatic replacement impossible
- SpecC model
 - Behaviors and channels
 - Separation of computation and communication
 - Plug-and-play



ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 8

Computation vs. Communication

- Protocol Inlining
 - Specification model
 - Exploration model

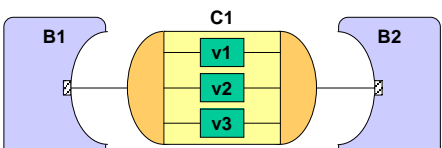


- Computation in behaviors
- Communication in channels

ECE298: SoC Description and Modeling, Lecture 2
(c) 2004 R. Doemer
9

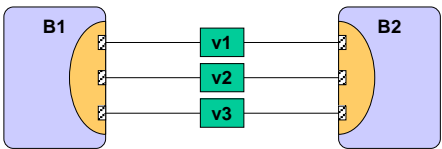
Computation vs. Communication

- Protocol Inlining
 - Specification model
 - Exploration model



- Computation in behaviors
- Communication in channels

- Implementation model



- Channel disappears
- Communication inlined into behaviors
- Wires exposed

ECE298: SoC Description and Modeling, Lecture 2
(c) 2004 R. Doemer
10

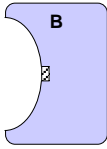
Introduction to System-on-Chip

- SoC design flow
- Computation vs. communication
- Intellectual Property (IP)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 11

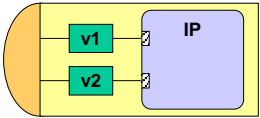
Intellectual Property (IP)

- Computation IP: Wrapper model



B

Synthesizable
behavior



IP in wrapper

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 12

Intellectual Property (IP)

- Computation IP: Wrapper model

Synthesizable behavior
Transducer
IP in wrapper

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 13

Intellectual Property (IP)

- Computation IP: Wrapper model
- Protocol inlining with wrapper

Synthesizable behavior
Transducer
IP in wrapper

before
after

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 14

Intellectual Property (IP)

- Computation IP: Adapter model

Synthesizable behavior Transducer Adapter IP

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 15

Intellectual Property (IP)

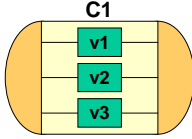
- Computation IP: Adapter model
- Protocol inlining with adapter

before after

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 16

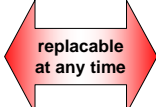
Intellectual Property (IP)

- Communication IP: Channel with wrapper

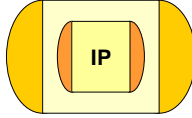


C1

Virtual channel



replacable
at any time



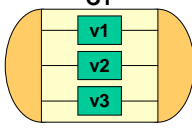
C2

IP protocol channel in wrapper

ECE298: SoC Description and Modeling, Lecture 2
(c) 2004 R. Doemer
17

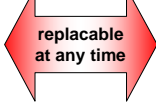
Intellectual Property (IP)

- Communication IP: Channel with wrapper

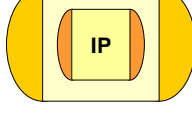


C1

Virtual channel



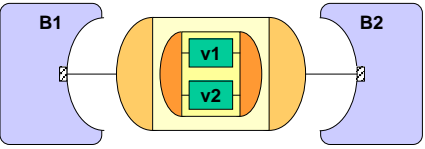
replacable
at any time



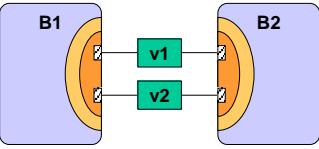
C2

IP protocol channel in wrapper

- Protocol inlining with hierarchical channel



before



after

ECE298: SoC Description and Modeling, Lecture 2
(c) 2004 R. Doemer
18

Intellectual Property (IP)

- Incompatible busses: Transducer insertion

Synthesizable behavior
System bus
Transducer
Adapter
IP bus
IP

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 19

Intellectual Property (IP)

- Incompatible busses: Transducer insertion
- Protocol inlining with transducer

Synthesizable behavior
System bus
Transducer
Adapter
IP bus
IP

B1
v1
v2
v3
T
v4
v5
IP

after

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 20

System-on-Chip Design Methodology

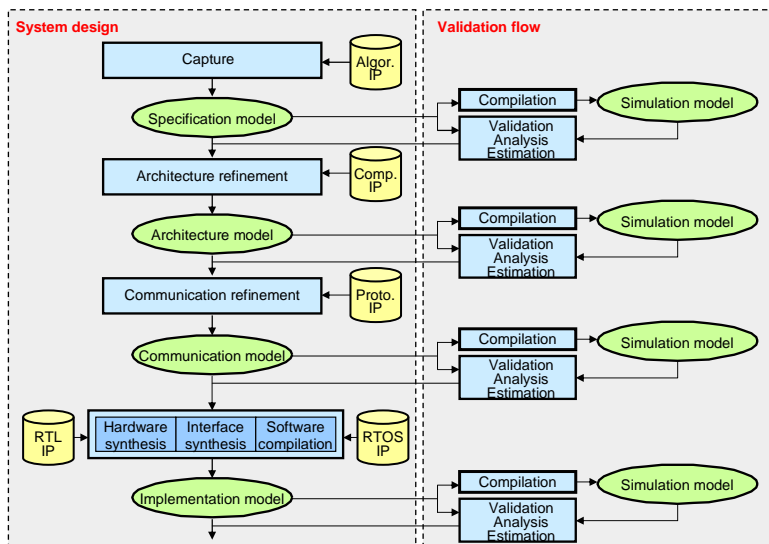
- Specification model
- Architecture model
- Communication model
- Implementation model

ECE298: SoC Description and Modeling, Lecture 2

(c) 2004 R. Doemer

21

System-on-Chip Design Methodology



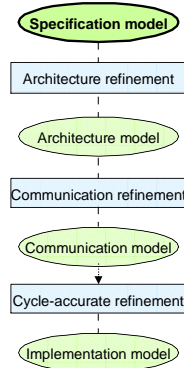
ECE298: SoC Description and Modeling, Lecture 2

(c) 2004 R. Doemer

22

Specification Model

- High-level, abstract model
 - Pure system functionality
 - Algorithmic behavior
 - No implementation details
- No implicit structure / architecture
 - Behavioral hierarchy
- Untimed
 - Executes in zero (logical) time
 - Causal ordering
 - Events only for synchronization



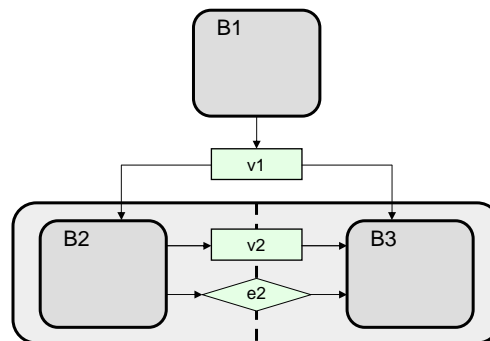
(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2

(c) 2004 R. Doemer

23

Specification Model Example



- Simple, typical specification model
 - Hierarchical parallel-serial composition
 - Communication through ports and variables, events

(Source: A. Gerstlauer)

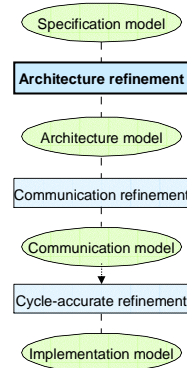
ECE298: SoC Description and Modeling, Lecture 2

(c) 2004 R. Doemer

24

Architecture Refinement

- Component allocation / selection
- Behavior partitioning
- Variable partitioning
- Scheduling



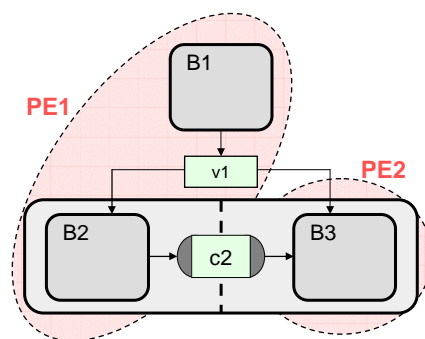
(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2

(c) 2004 R. Doemer

25

Allocation, Behavior Partitioning



- Allocate PEs
- Partition behaviors
- Globalize communication

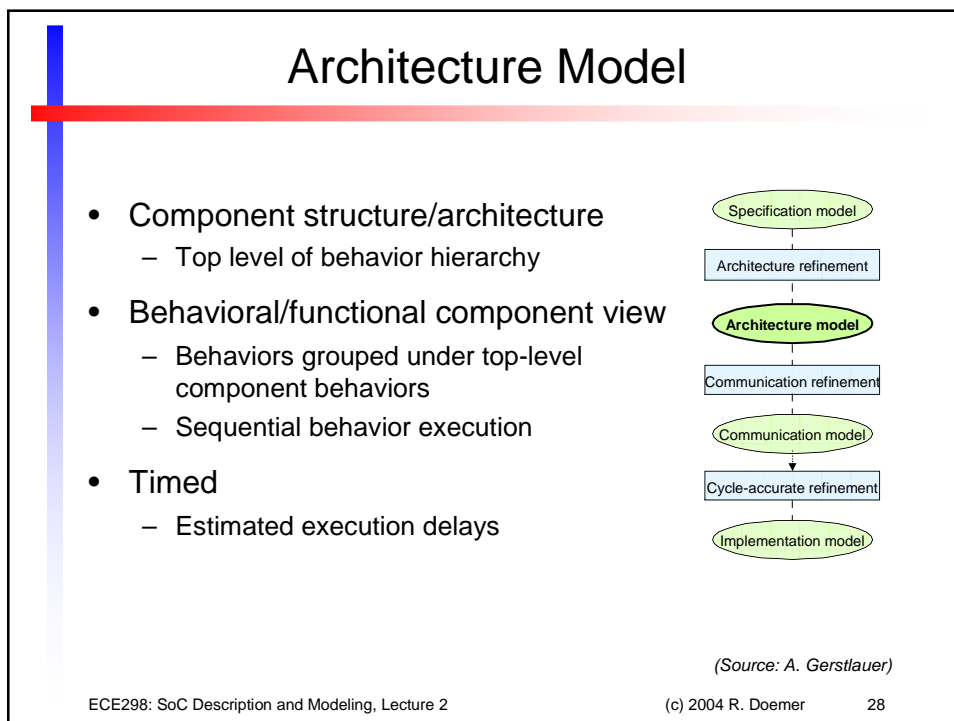
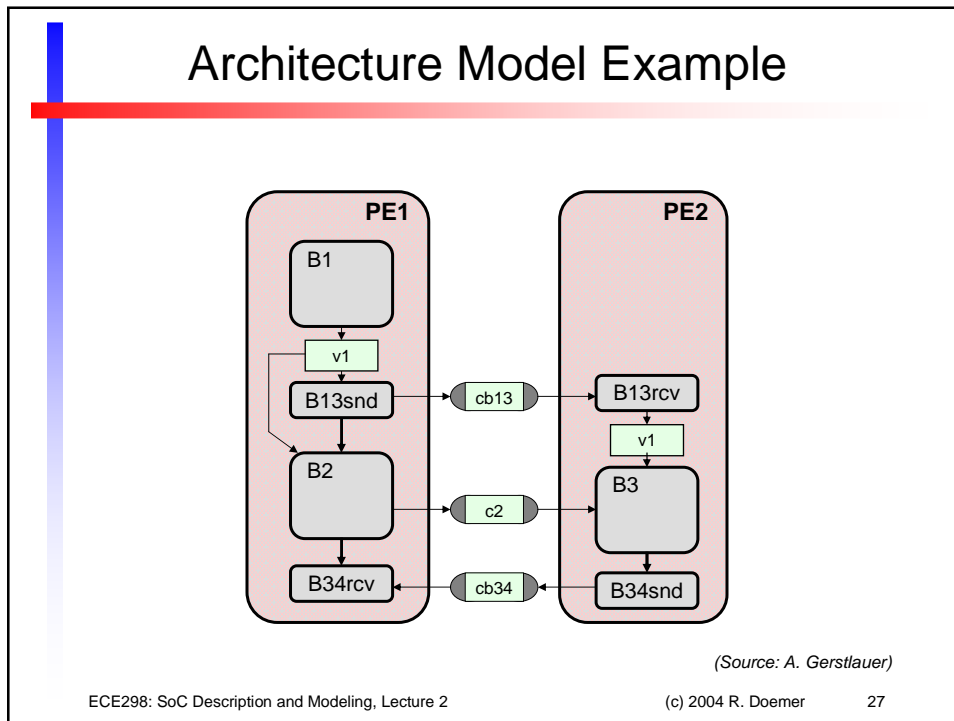
➤ Additional level of hierarchy to model PE structure

(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2

(c) 2004 R. Doemer

26



Communication Refinement

- Bus allocation / protocol selection
- Channel partitioning
- Protocol, transducer insertion
- Inlining

```

graph TD
    A([Specification model]) --> B[Architecture refinement]
    B --> C([Architecture model])
    C --> D[Communication refinement]
    D --> E([Communication model])
    E --> F[Cycle-accurate refinement]
    F --> G([Implementation model])
    
```

(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 29

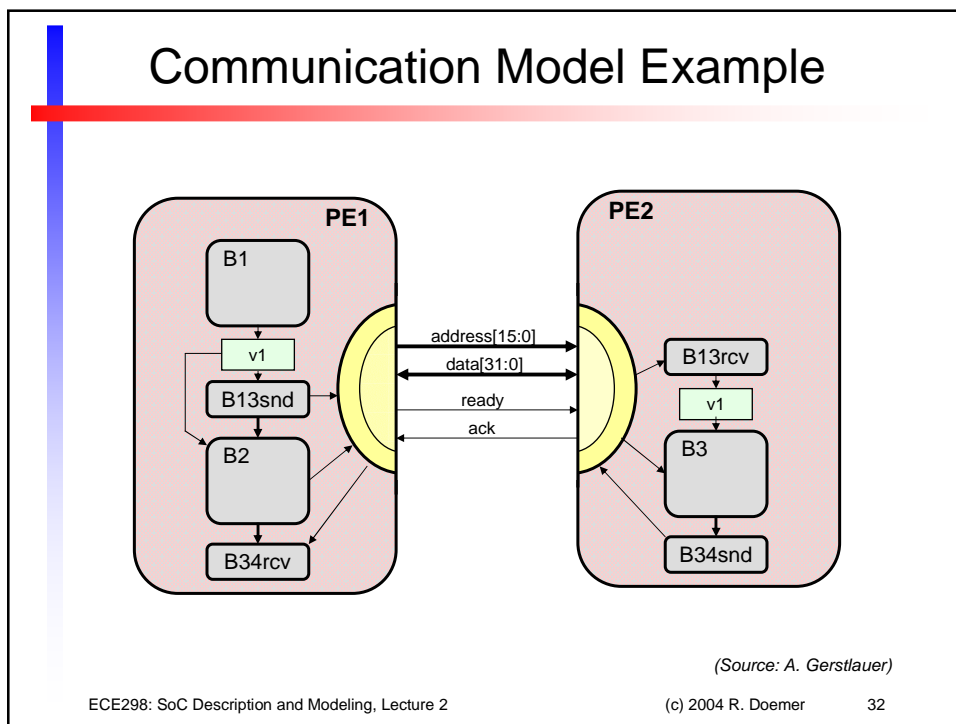
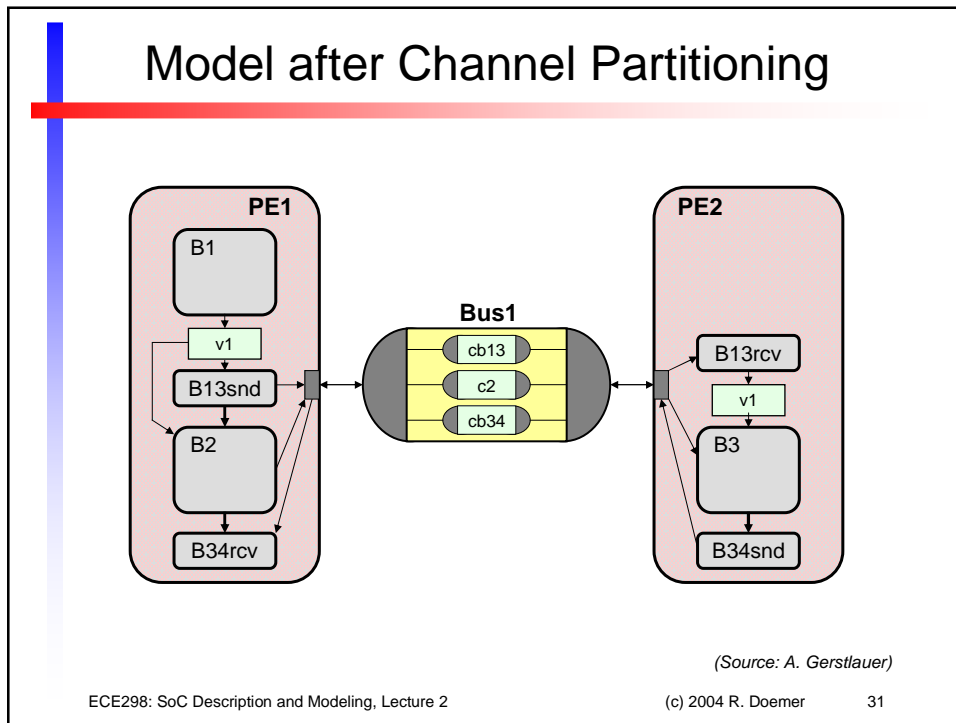
Bus Allocation / Channel Partitioning

- Allocate busses
- Partition channels
- Update communication

➤ Additional level of hierarchy to model bus structure

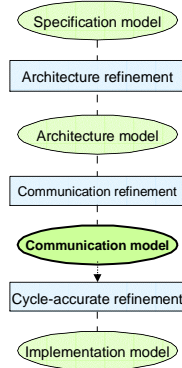
(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 30



Communication Model

- Component & bus structure/architecture
 - Top level of hierarchy
- Bus-functional component models
 - Timing-accurate bus protocols
 - Behavioral component description
- Timed
 - Estimated component delays



(Source: A. Gerstlauer)

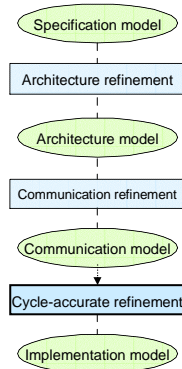
ECE298: SoC Description and Modeling, Lecture 2

(c) 2004 R. Doemer

33

Cycle-accurate Refinement

- Clock-accurate implementation of PEs
 - Hardware synthesis
 - Software synthesis
 - Interface synthesis



(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2

(c) 2004 R. Doemer

34

Hardware Synthesis

The diagram illustrates the hardware synthesis process for a Processing Element (PE2). On the left, a vertical flowchart shows the components of PE2: B13rcv, v1, B3, and B34snd. On the right, a timing diagram shows a sequence of operations represented by a grid of dots. Three horizontal lines labeled PE2_CLK indicate the clock boundaries. A bracket on the right side of the timing diagram is labeled 'Clock boundaries'.

- Schedule operations into clock cycles
 - Define clock boundaries in leaf behavior C code
 - Create FSMD model from scheduled C code

(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 35

Software Synthesis

The diagram illustrates the software synthesis process for a Processing Element (PE1). On the left, a vertical flowchart shows the components of PE1: B1, v1, B13snd, B2, and B34rcv. A green arrow points from this flowchart to a list of assembly instructions for PE2. The instructions are: MOVE r0, r1; SHL r3; ADD r2, r3, r4; INC r2; PUSH r1; CALL PE3; POP r0.

- Implement behavior on processor instruction-set
 - Code generation
 - Compilation

(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 36

Interface Synthesis

- Implement communication on components
 - Hardware bus interface logic
 - Software bus drivers

(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 37

Implementation Model Example

(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 38

Implementation Model

- Cycle-accurate system description
 - RTL description of hardware
 - Behavioral/structural FSM/D view
 - Object code for processors
 - Instruction-set co-simulation
 - Clocked bus communication
 - Bus interface timing based on PE clock

```

graph TD
    A([Specification model]) --> B[Architecture refinement]
    B --> C([Architecture model])
    C --> D[Communication refinement]
    D --> E([Communication model])
    E --> F[Cycle-accurate refinement]
    F --> G([Implementation model])
  
```

(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 39

System-on-Chip Design Methodology

- Four levels of abstraction
 - Specification model: untimed, functional
 - Architecture model: estimated, structural
 - Communication model: timed, bus-functional
 - Implementation model: cycle-accurate, RTL/IS
- Three refinement steps
 - Architecture refinement
 - Communication refinement
 - Cycle-accurate refinement
 - HW / SW / interface synthesis
- Well-defined, formal models & transformations
 - Automatic, gradual refinement
 - Executable models, test bench re-use
 - Simple verification

(Source: A. Gerstlauer)

ECE298: SoC Description and Modeling, Lecture 2 (c) 2004 R. Doemer 40