EE382N.23: Embedded System Design and Modeling

Lecture 7 – System Refinement & Modeling

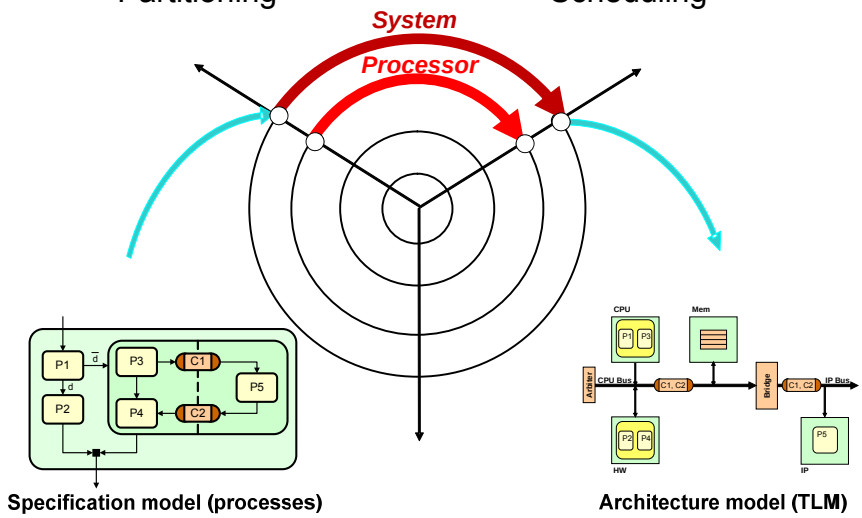
Andreas Gerstlauer
Electrical and Computer Engineering
University of Texas at Austin
gerstl@ece.utexas.edu

Lecture 7: Outline

- **System-level synthesis**
  - Design flow: From specification to implementation
  - X-Chart: Decision making + refinement
- **System-level refinement**
  - Refinement flow & process
  - Refinement example
- **System-level modeling**
  - Virtual protoyping & virtual platform models

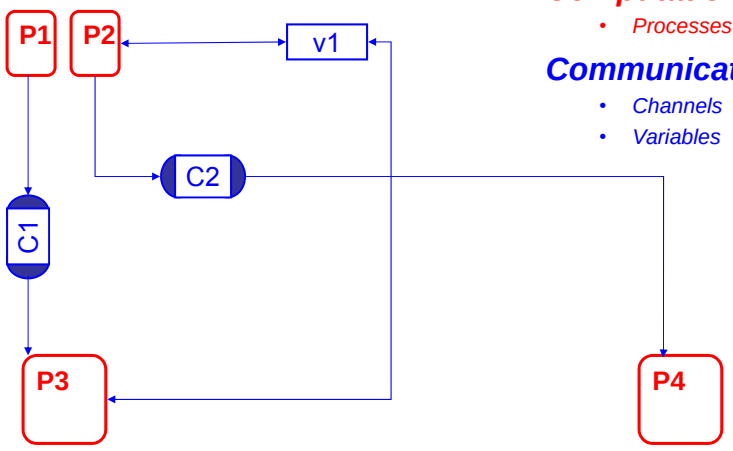
System-Level Design Flow

- **Structure**
  - Partitioning
- **Timing**
  - Scheduling



Specification model (processes) Architecture model (TLM)

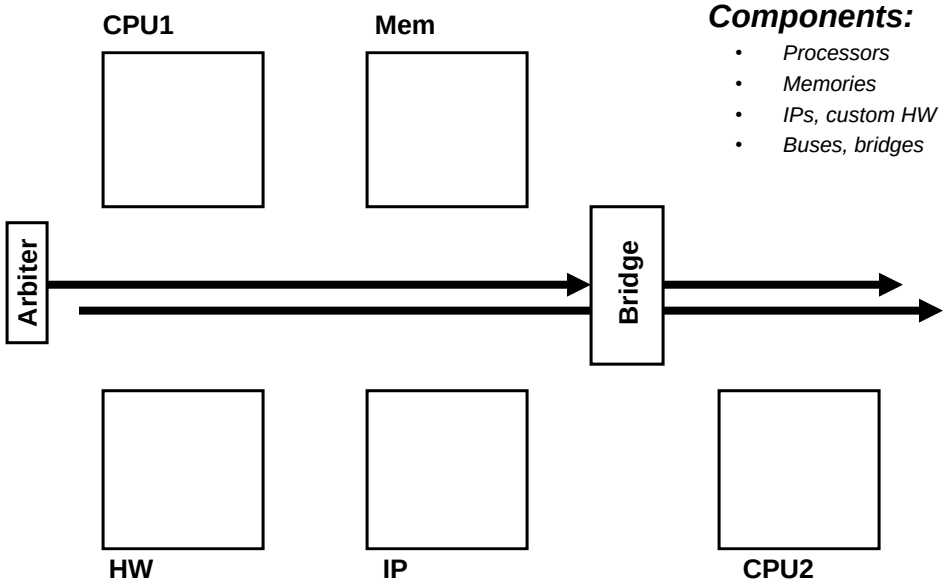
Application Specification (MoC)



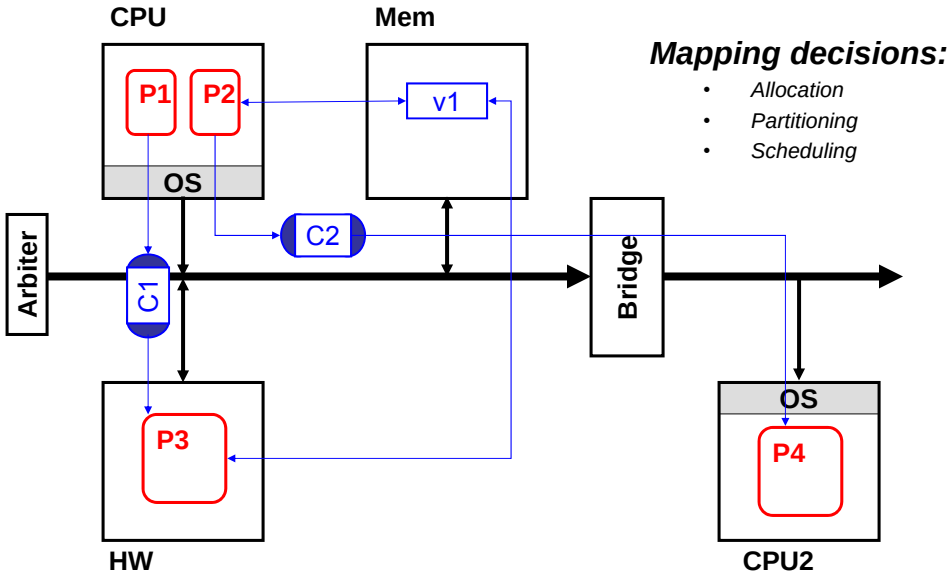
Computation
- *Processes*

Communication
- *Channels*
- *Variables*

Platform Architecture Template

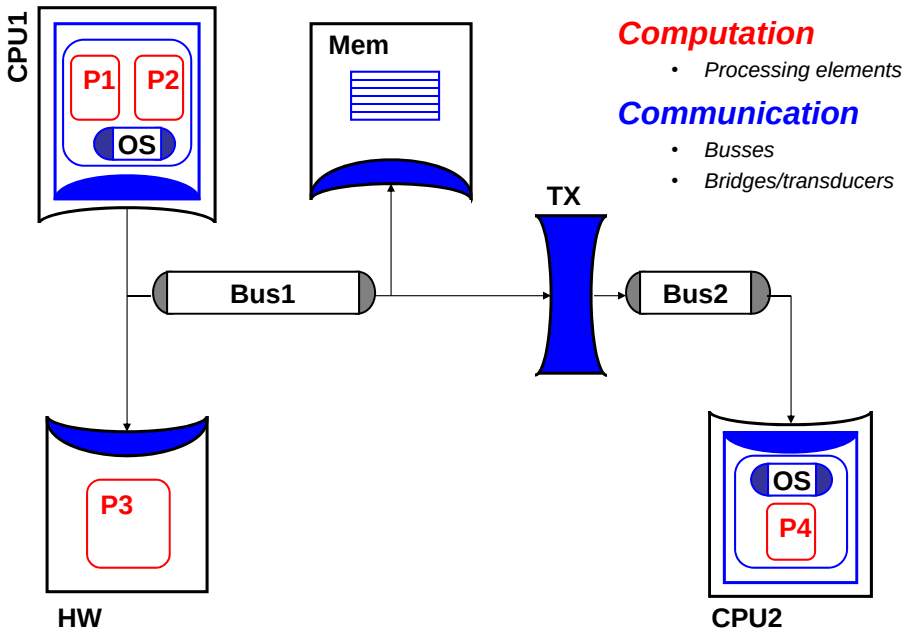


System Definition

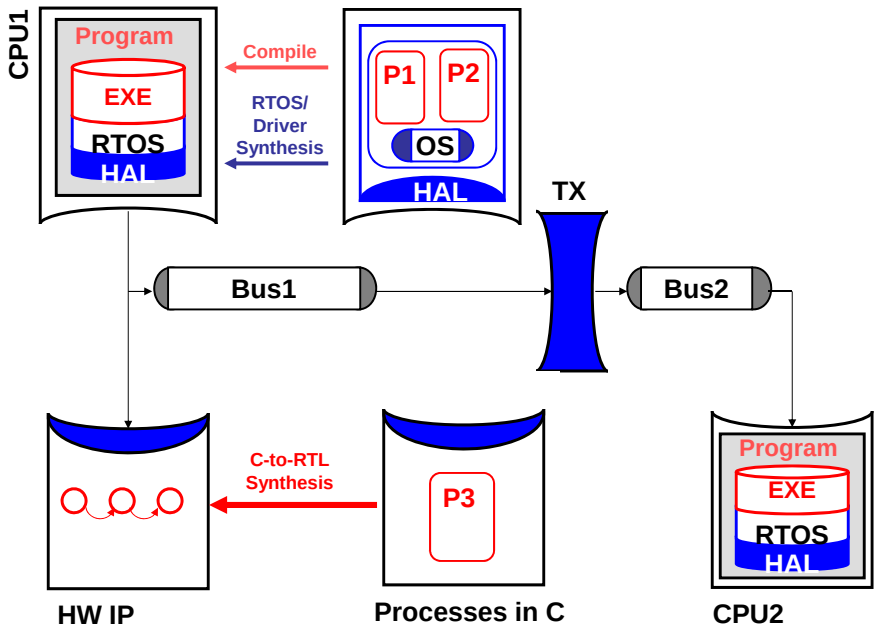


System Definition = Application + Platform + Mapping

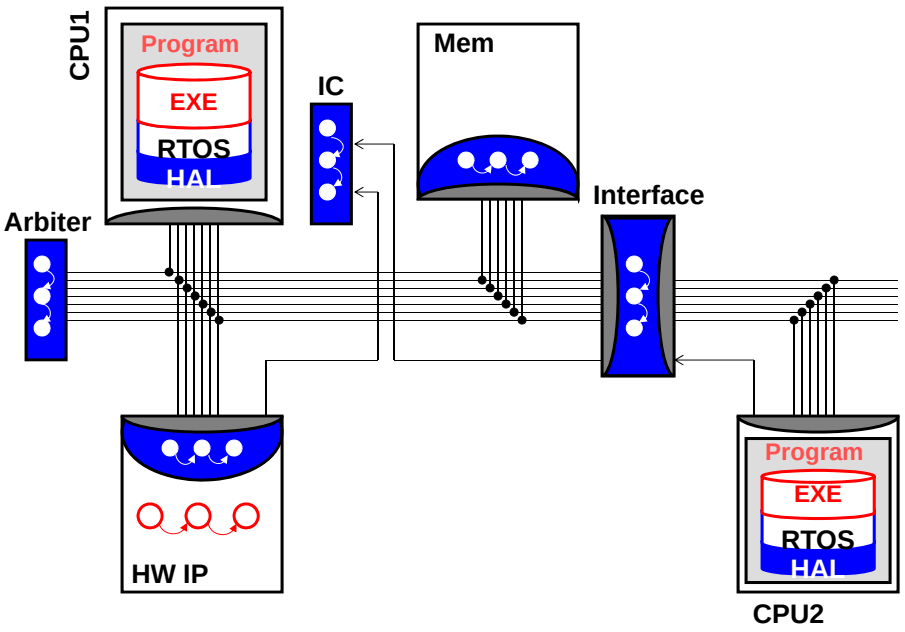
Architecture Model



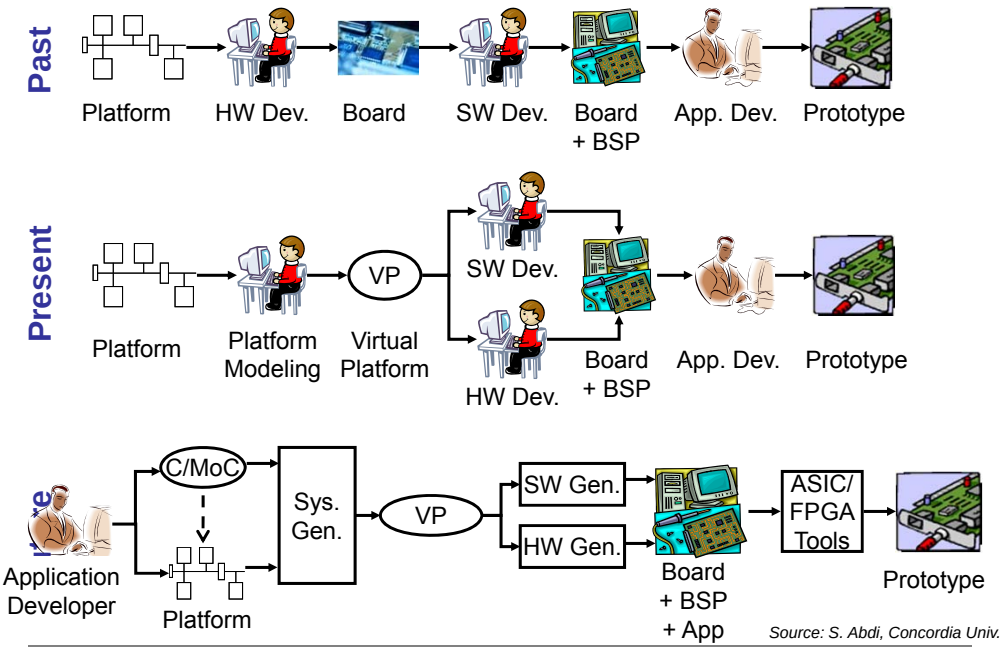
Backend Hardware/Software Synthesis



Implementation Model

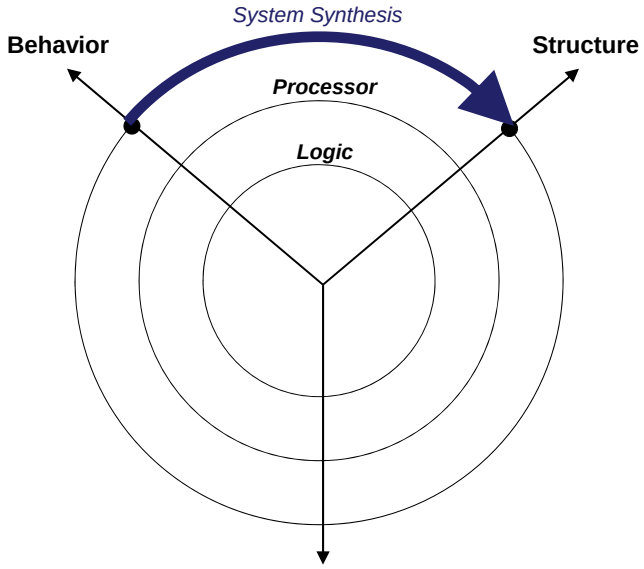


System Design Evolution



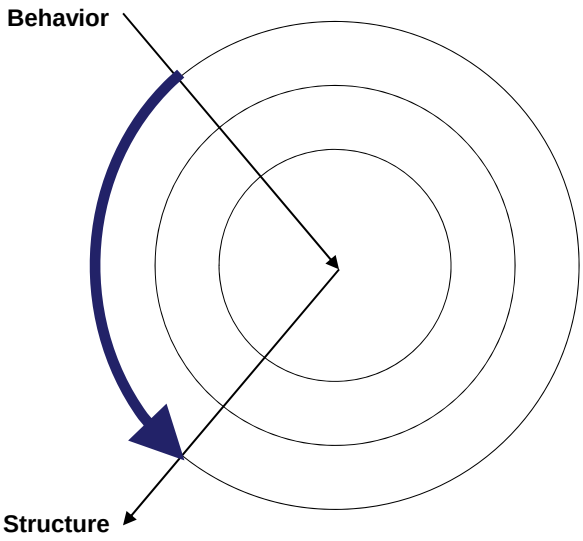
Synthesis

- **Gajski's Y-Chart**



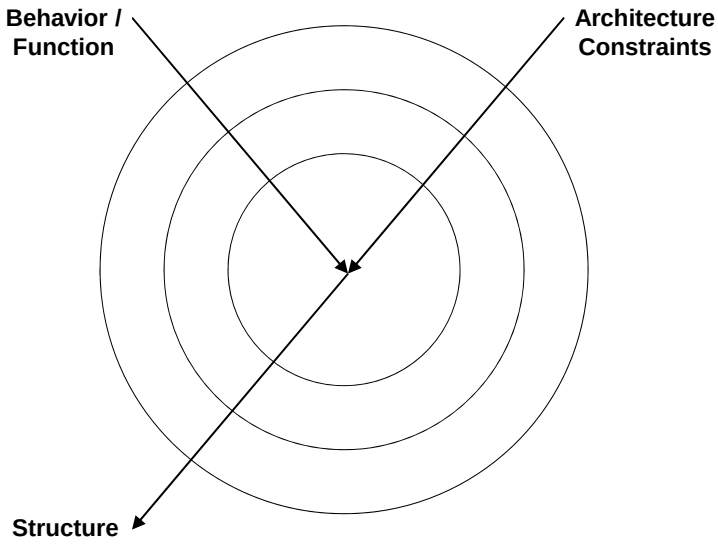
Synthesis

- **Gajski's Y-Chart**



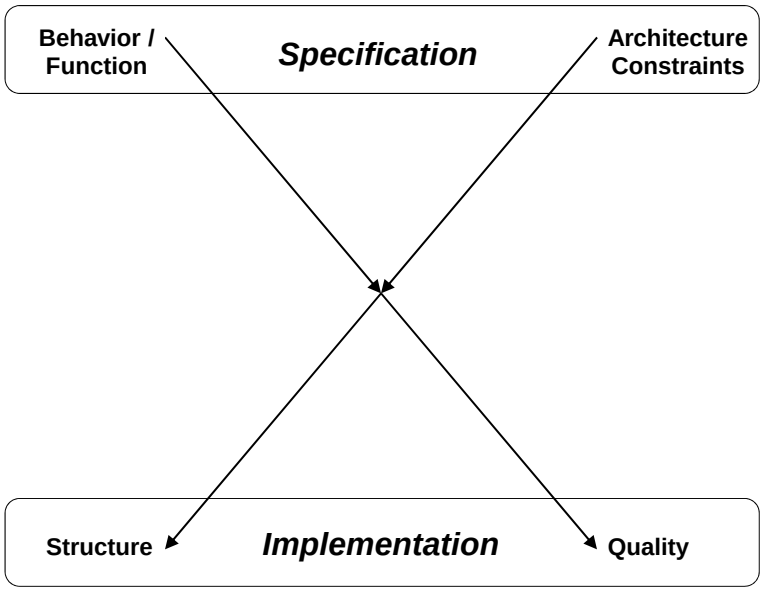
Synthesis

- **Platform-based design (the other Y Chart)**



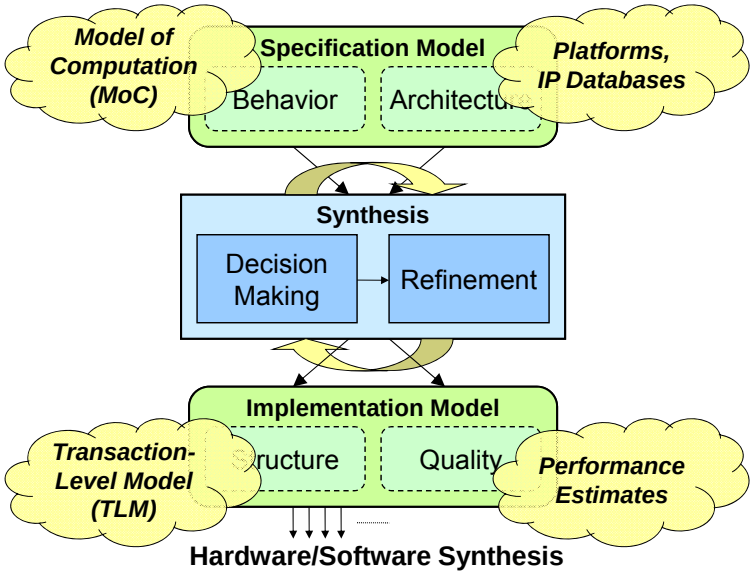
Synthesis

- **X-Chart**



Synthesis

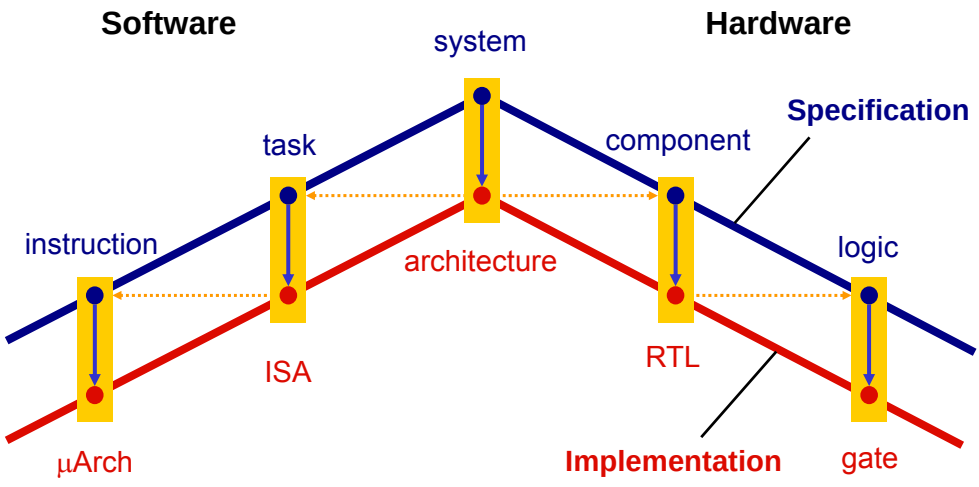
- System-Level Synthesis



Source: A. Gerstlauer, C. Haubelt, A. Pimentel, et al., "Electronic System-Level Synthesis Methodologies," TCAD, 2009.

Hardware vs. Software

- Double-roof model



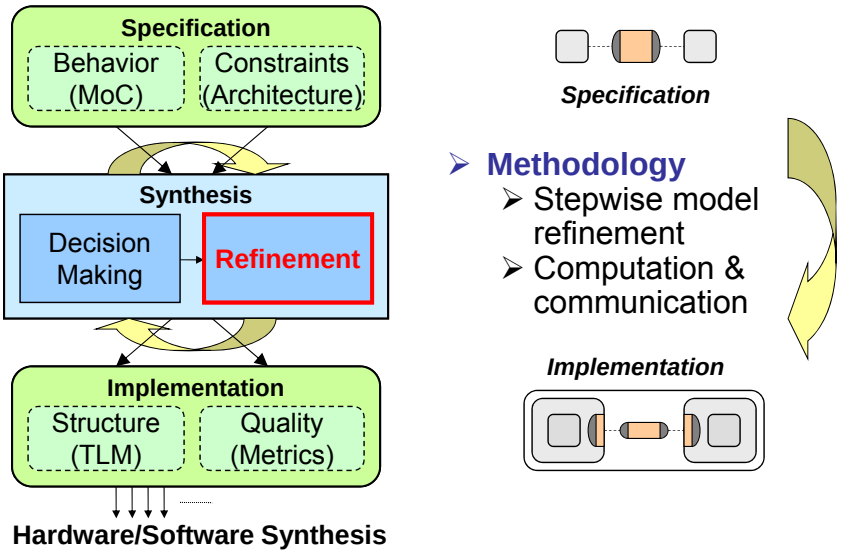
Source: A. Gerstlauer, C. Haubelt, A. Pimentel, et al., "Electronic System-Level Synthesis Methodologies," TCAD, 2009.

Lecture 7: Outline

- ✓ System-level synthesis
  - ✓ Design flow: From specification to implementation
  - ✓ X-Chart: Decision making + refinement
- **System-level refinement**
  - Refinement flow & process
  - Refinement example
- System-level modeling
  - Virtual protoyping & virtual platform models

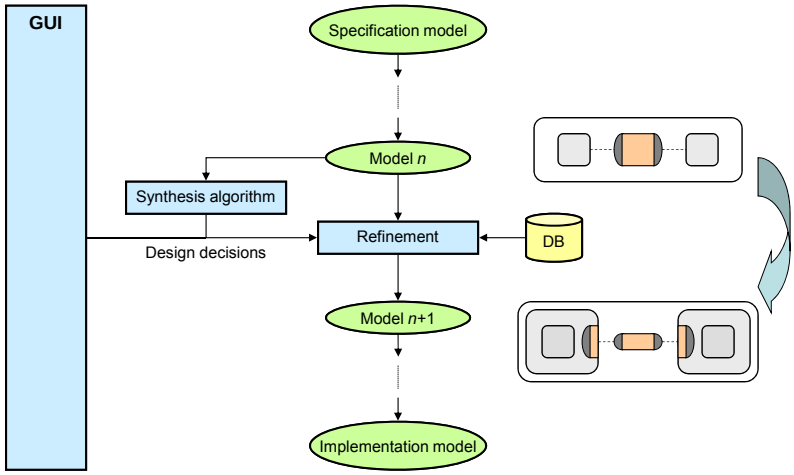
System-Level Synthesis

- **X-Chart**



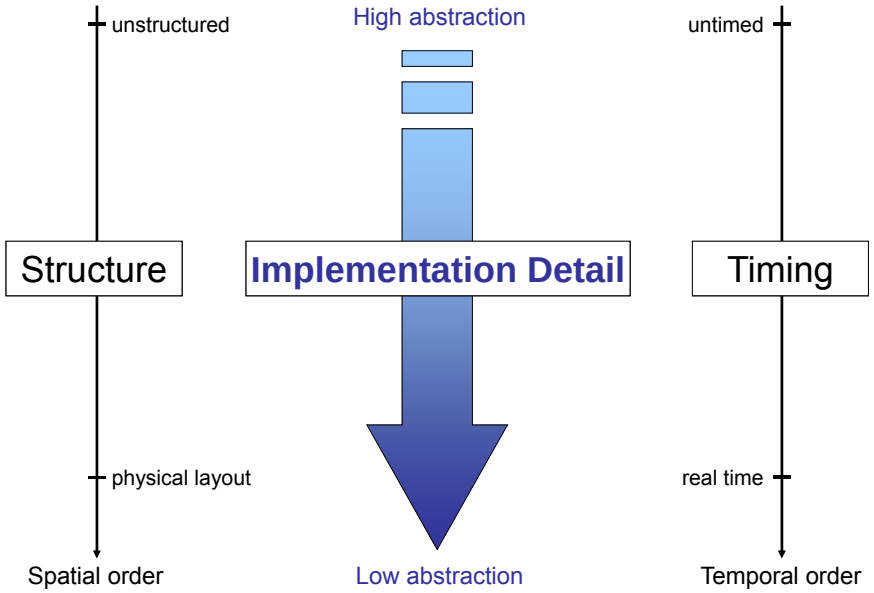
Synthesis & Refinement Flow

- **Synthesis = Decision making + model refinement**



- ➢ **Successive, stepwise model refinement**
- ➢ **Layers of implementation detail**

Refinement

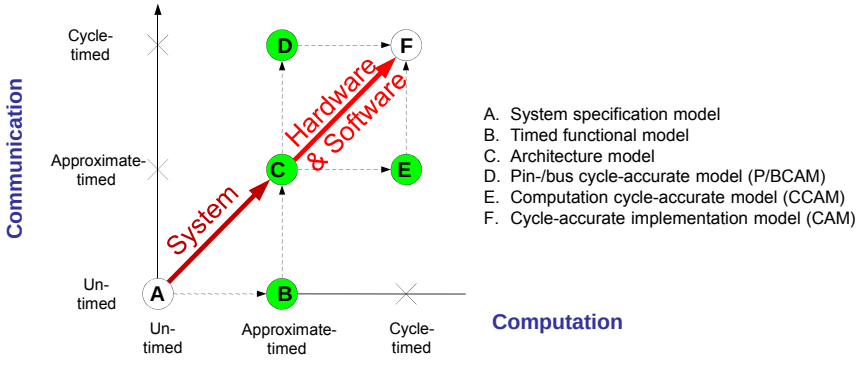


Modeling

- **Basis of any design flow and design automation**
  - Inputs and outputs of design steps
    - Capability to capture complex systems
    - Precise, complete and unambiguous
  - Models at varying levels of abstraction
    - Level and granularity of implementation detail
    - Speed vs. accuracy
- **Design models as an abstraction of a design instance**
  - Representation of some aspect of reality
    - Virtual prototyping for validation through simulation or formal analysis
  - Specification for further implementation
    - Describe desired functionality
  - Documentation & Specification (Simulation & Synthesis)
    - Abstraction to hide details that are not relevant or not yet known
    - Different parts of the model or different use cases for the same model

Computation vs. Communication

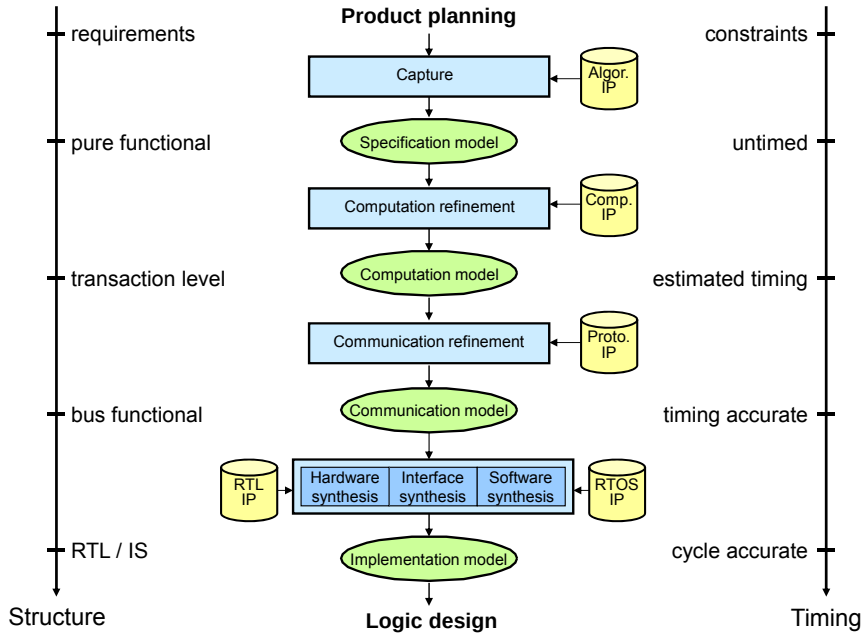
- **System design flow**
  - Path from model A to model F



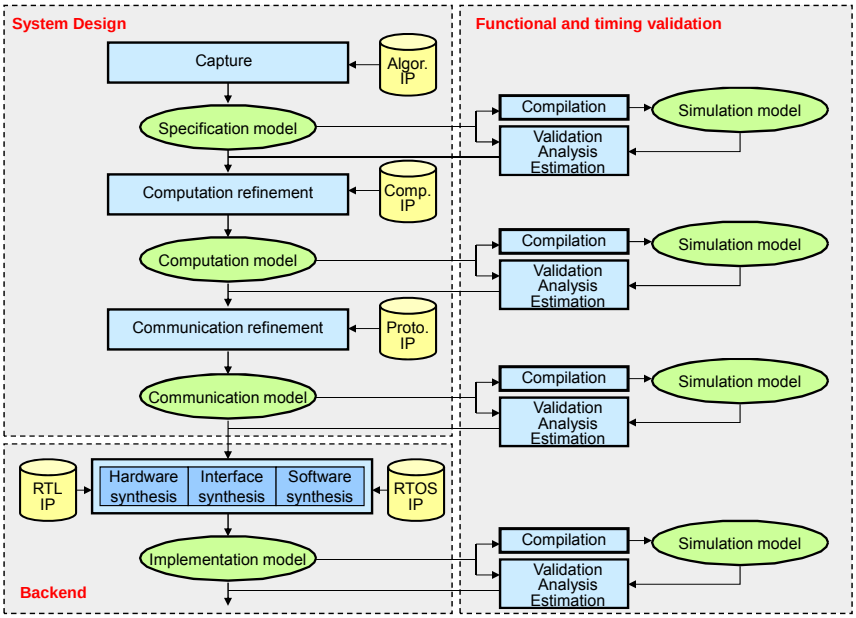
Source: L. Cai, D. Gajski. "Transaction level modeling: An overview", ISSS 2003

- **Design methodology and modeling flow**
  - Set of models and transformations between models

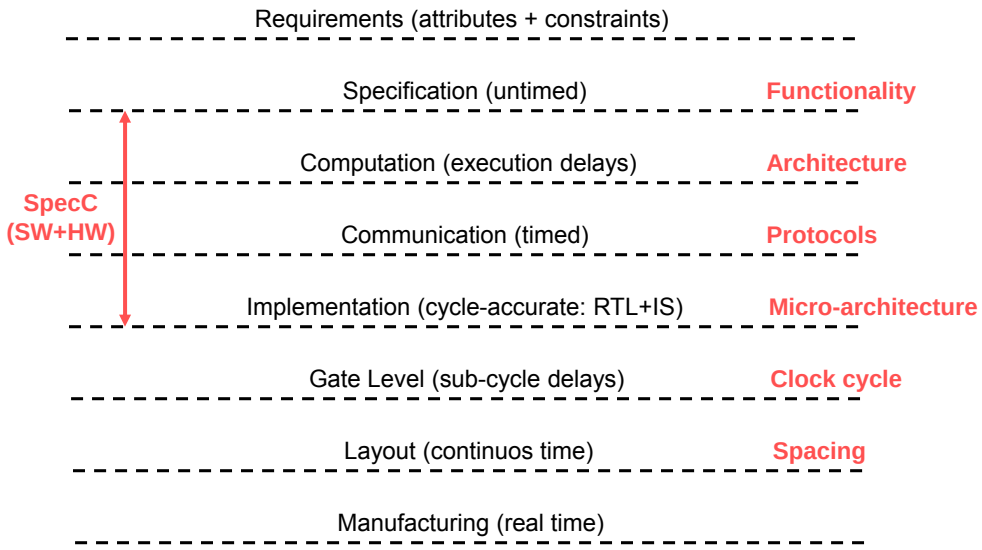
SpecC Design Flow



Validation Flow

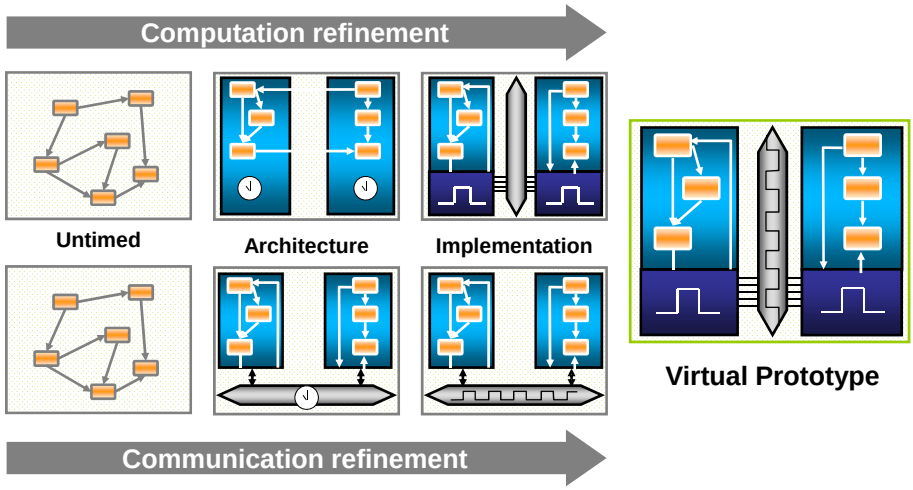


Validation Levels



➢ **Structure & Timing**

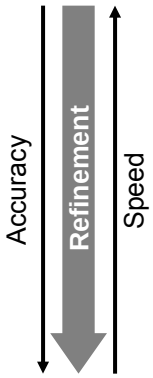
Virtual Platform Prototyping



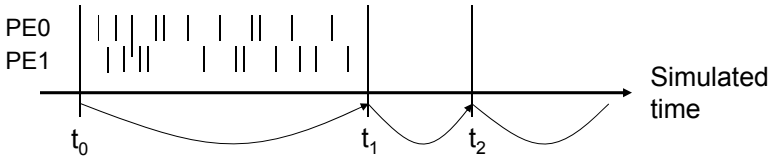
Source: C. Haubelt, Univ. of Rostock

Virtual Prototyping Speed vs. Accuracy

- **Discrete-event simulation speed**
  - Proportional to number of simulated events
  - Proportional to granularity of simulated time/detail
    - "Real-time": simulated vs. simulation time > 1
- **Discrete-event simulation accuracy**
  - Proportional to simulated implementation order
  - Inversely proportional to simulated granularity
    - Where order matters (structural concurrency)



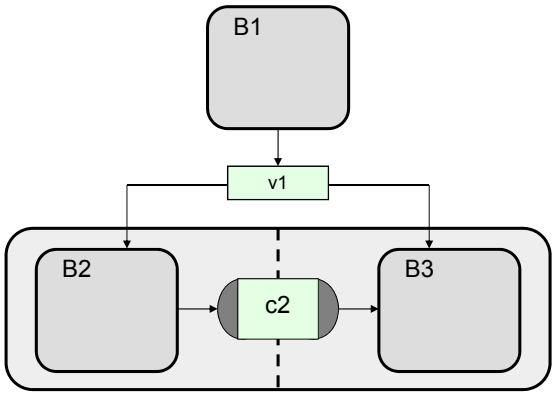
➢ **Fundamental modeling tradeoff**



Lecture 7: Outline

- ✓ System-level synthesis
  - ✓ Design flow: From specification to implementation
  - ✓ X-Chart: Decision making + refinement
- **System-level refinement**
  - ✓ Refinement flow & process
  - Refinement example
- System-level modeling
  - Virtual protoyping & virtual platform models

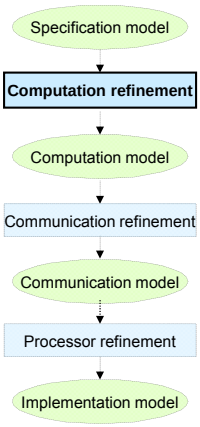
Specification Model Example



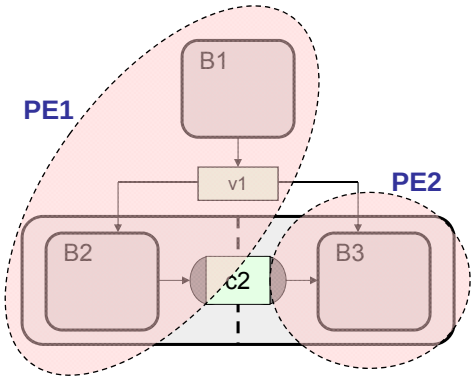
- **Synthesizable specification model**
  - Hierarchical parallel-serial composition
  - Communication through variables and standard channels

Computation Refinement

- **PE allocation / selection**
- **Behavior partitioning**
- **Variable partitioning**
- **Scheduling**



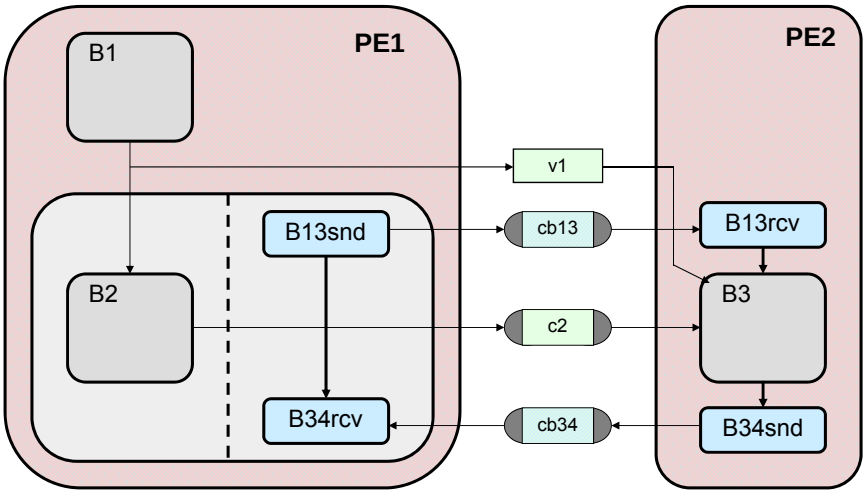
PE Allocation, Behavior Partitioning



- Allocate PEs
- Partition behaviors
- Globalize communication

➢ **Additional level of hierarchy to model PE structure**

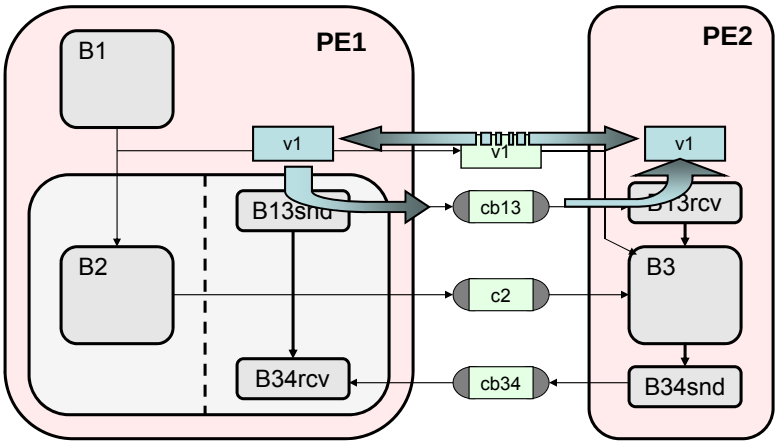
Model after Behavior Partitioning



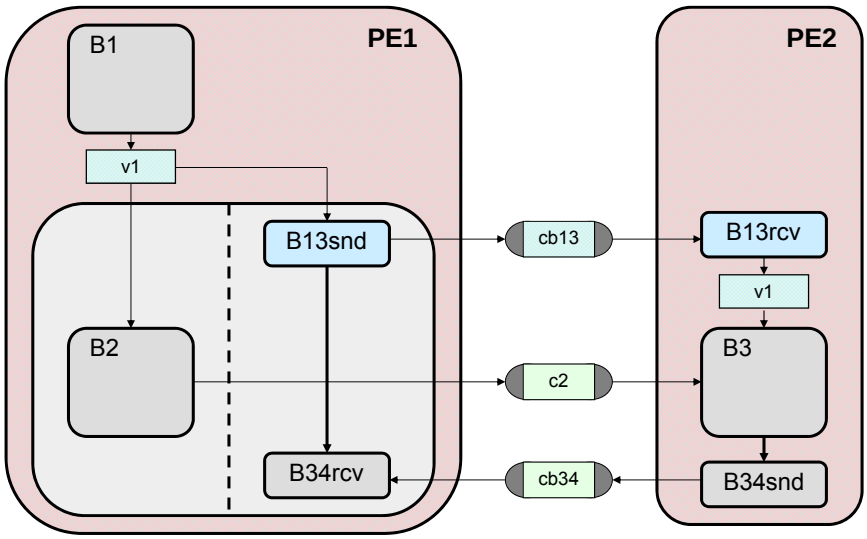
➢ **Synchronization to preserve execution order/semantics**

Variable Partitioning

- **Shared memory vs. message passing implementation**
  - Map global variables to local memories
  - Communicate data over message-passing channels



Model after Variable Partitioning



- **Keep local variable copies in sync**
  - Communicate updated values at synchronization points
  - Transfer control & data over message-passing channel

Timed Computation

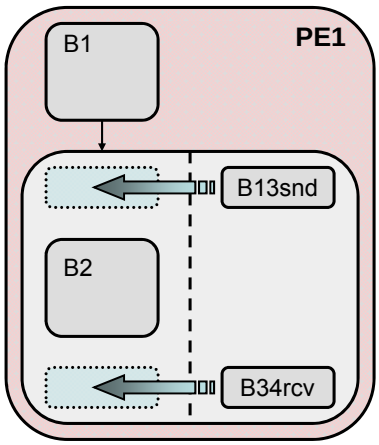
- **Execution time of behaviors**
  - Estimated target delay / timing budget
- **Granularity**
  - Behavior / function / basic-block level

➢ **Annotate behaviors**
- Simulation feedback
- Synthesis constraints

```
1  behavior B2( in int v1, ISend c2 )
   {
     void main(void) {
       ...
5      waitfor( B2_DELAY1 );
       c2.send( ... );
       ...
10     waitfor( B2_DELAY2 );
     }
   };
```

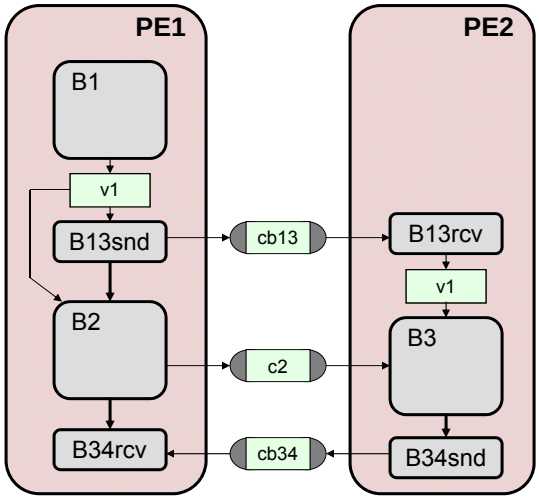

Scheduling

➢ **Serialize behavior execution on components**



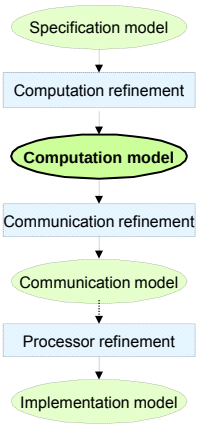
- Static scheduling
  - Fixed behavior execution order
  - Flattened behavior hierarchy
- Dynamic scheduling
  - Pool of tasks
  - Scheduler, abstracted OS

Computation Model Example



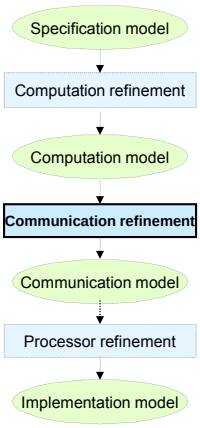
Computation Model

- **Component structure/architecture**
  - Top level of behavior hierarchy
- **Behavioral/functional component view**
  - Behaviors grouped under top-level component behaviors
  - Sequential behavior execution
- **Timed**
  - Estimated execution delays

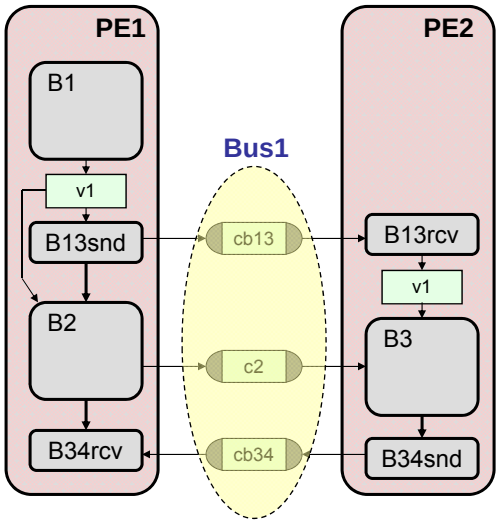


Communication Refinement

- Network allocation / protocol selection
- Channel partitioning
- Protocol stack insertion
- Inlining



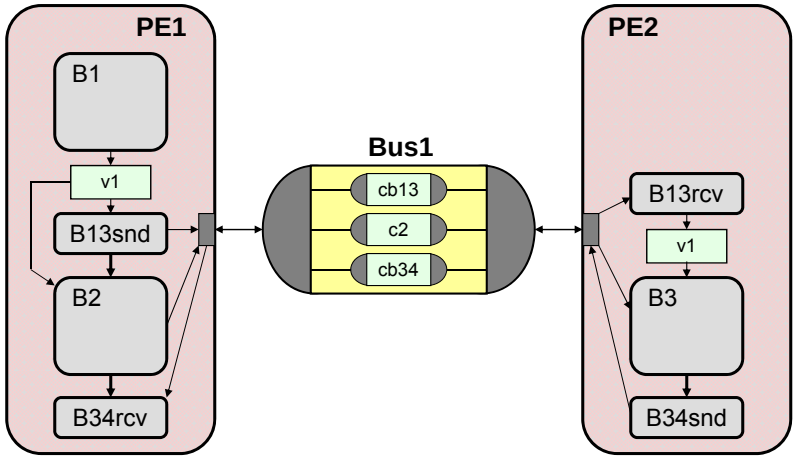
Network Allocation / Channel Partitioning



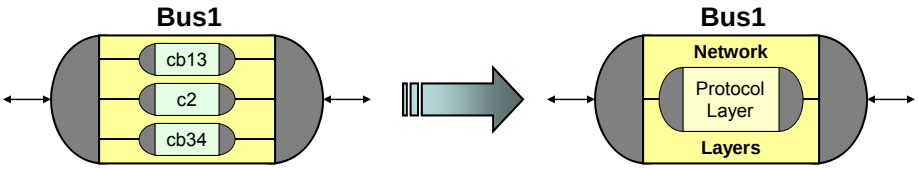
- Allocate busses
- Partition channels
- Update communication

➢ Additional level of hierarchy to model bus structure

Model after Channel Partitioning

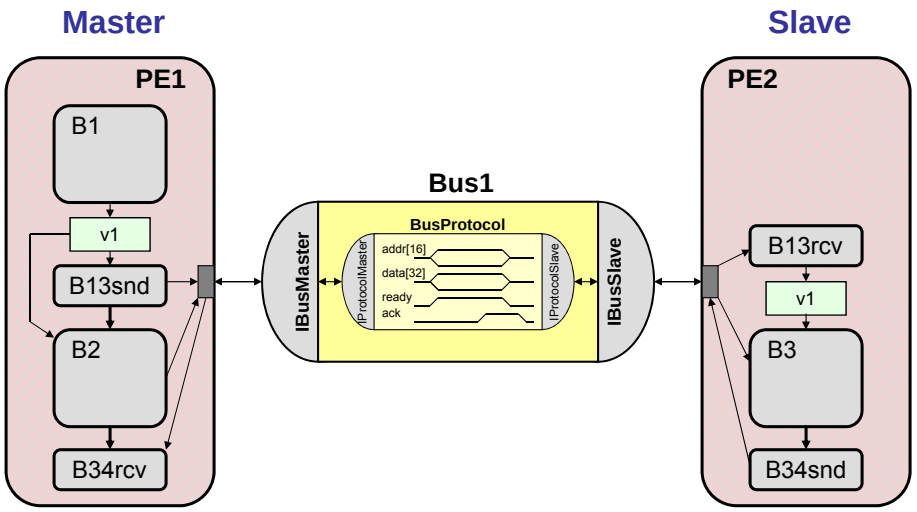


Protocol Insertion

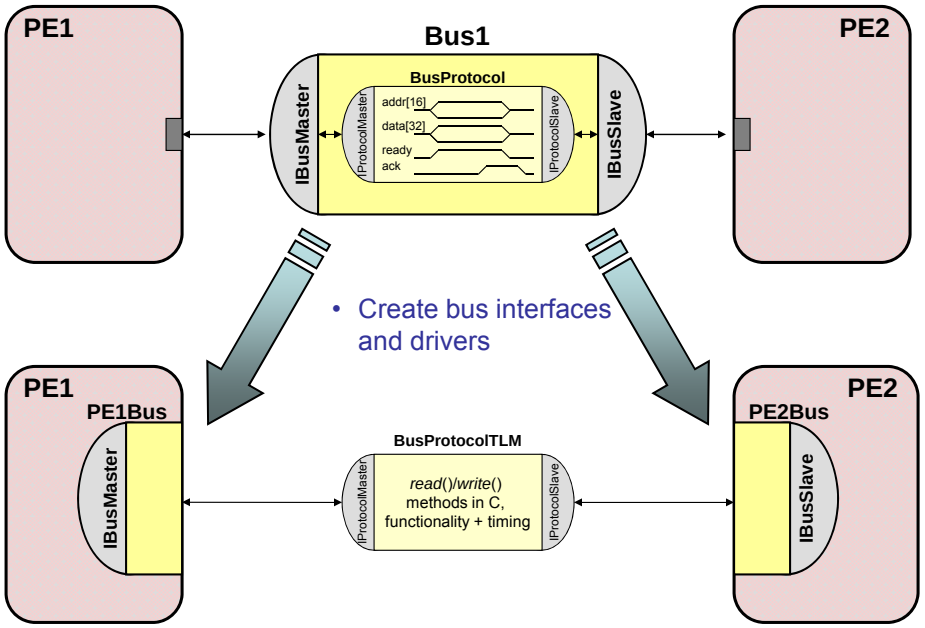


- **Insert protocol layer**
  - Bus protocol channel from database
- **Create network layers**
  - Implement message-passing over bus protocol
- **Replace bus channel**
  - Hierarchical combination of complete protocol stack

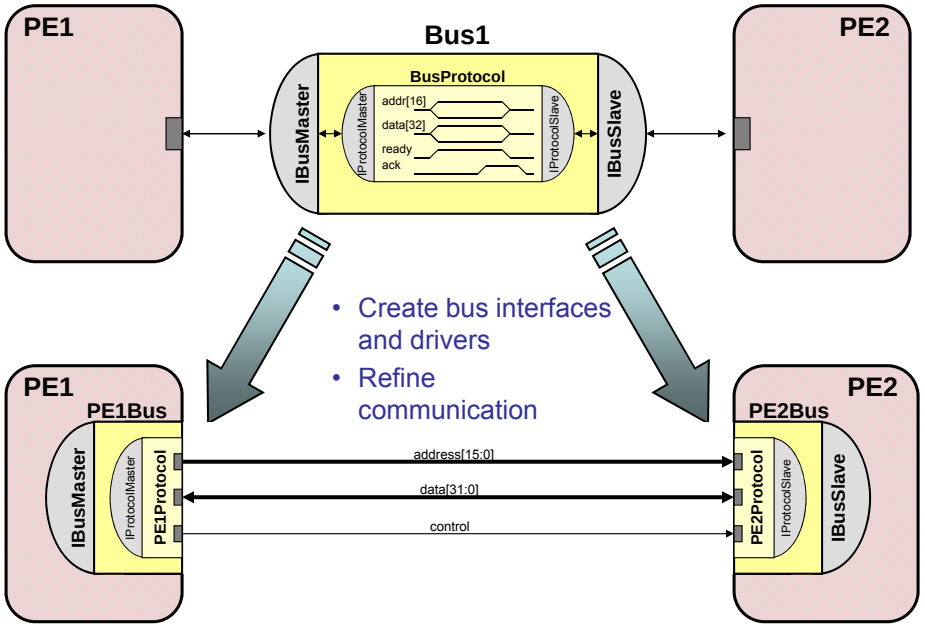
Model after Protocol Insertion



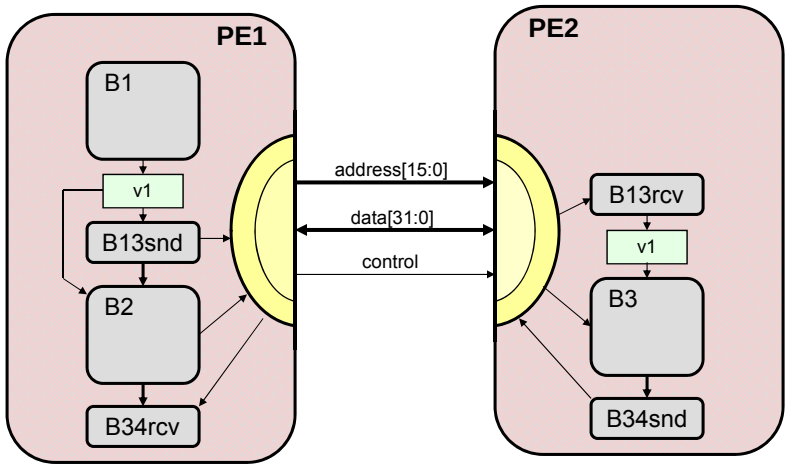
Inlining: Transaction-Level Model (TLM)



Inlining: Pin-Accurate Model (PAM)



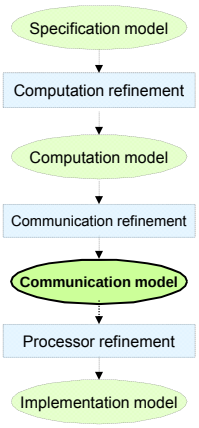
Communication Model Example



Communication Model

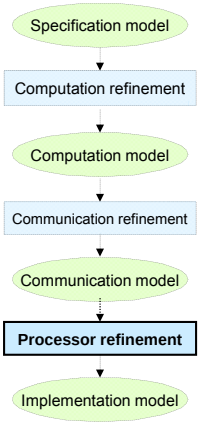
- **Component & bus structure/architecture**
  - Top level of hierarchy
- **Bus-functional component models**
  - Timing-accurate bus protocols
  - Behavioral component description
- **Timed**
  - Estimated component delays
  - Timing-accurate communication

➢ **Transaction-level model (TLM)**
➢ **Pin-accurate model (PAM)**
  ➢ Bus cycle-accurate model (BCAM)

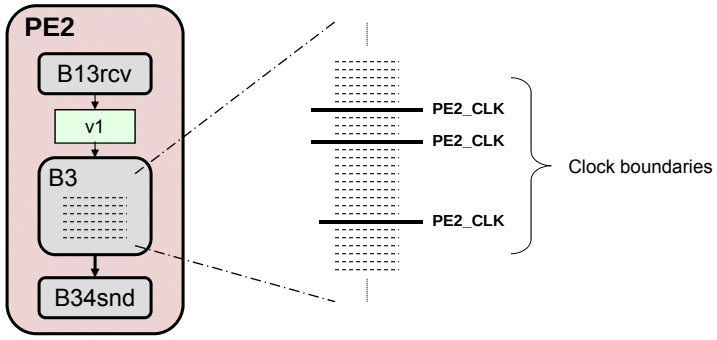


Processor Refinement

- **Cycle-accurate implementation of PEs**
  - Hardware synthesis down to RTL
  - Software synthesis down to IS
  - Interface synthesis down to RTL/IS

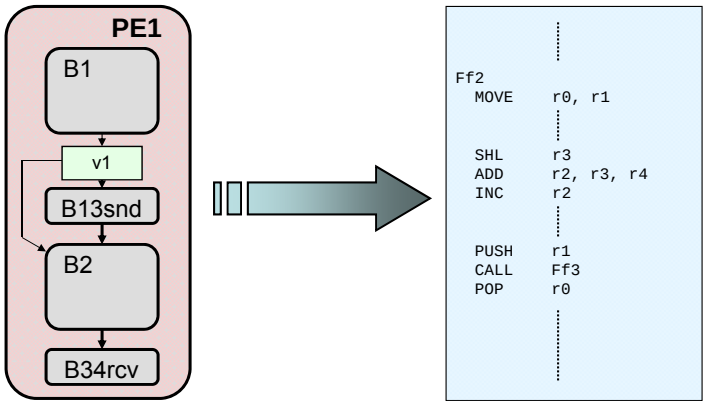


Hardware Synthesis



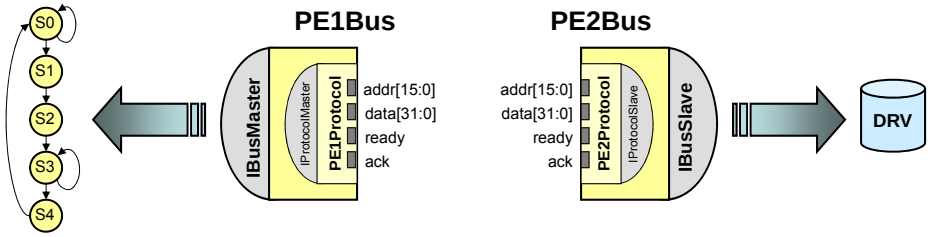
- **Schedule operations into clock cycles**
  - Define clock boundaries in leaf behavior C code
  - Create FSMD model from scheduled C code
    - Controller + datapath

Software Synthesis



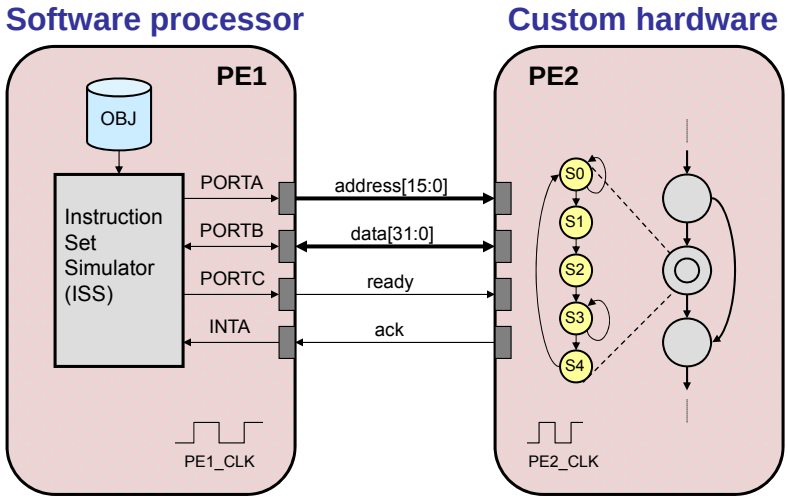
- **Implement behavior on processor instruction-set**
  - Code generation
  - Compilation

Interface Synthesis



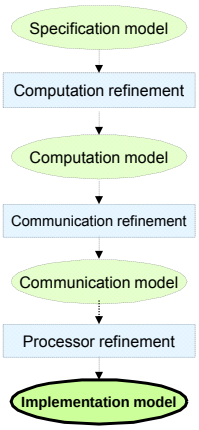
- **Implement communication on components**
  - Hardware bus interface logic
  - Software bus drivers

Implementation Model



Implementation Model

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



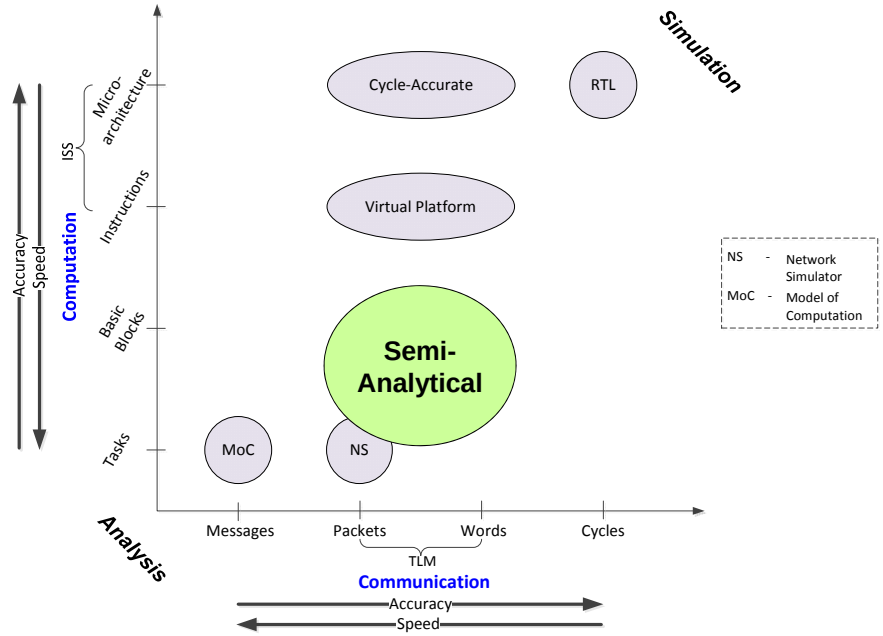
Lecture 7: Outline

✓ **System-level synthesis**
  ✓ Design flow: From specification to implementation
  ✓ X-Chart: Decision making + refinement

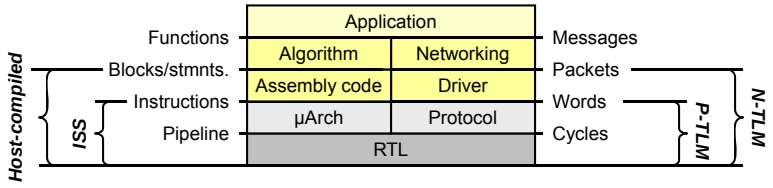
✓ **System-level refinement**
  ✓ Refinement flow & process
  ✓ Refinement example

- **System-level modeling**
  - Virtual protoyping & virtual platform models

System-Level Modeling Space

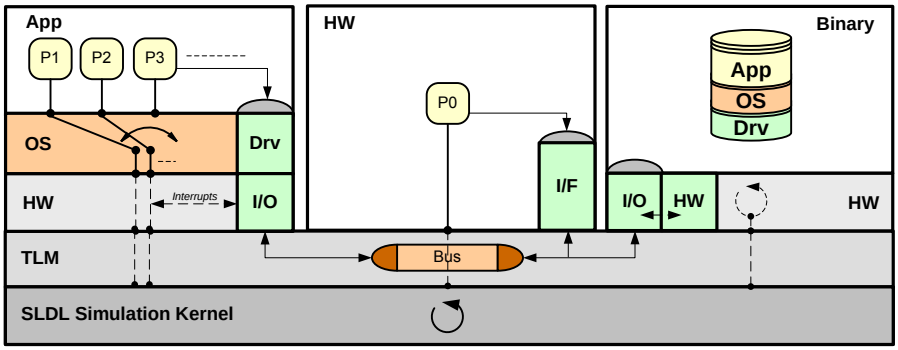


Modeling Levels



- **Host-compiled modeling of computation**
  - Abstract execution above instructions
  - ➢ Native execution of functionality
  - ➢ Back-annotation of timing, energy, …
  - ➢ Models of execution environment (OS & processor)
- **Transaction-level modeling (TLM) of communication**
  - Abstract transactions above pins and wires
  - ➢ Function calls for data transfer functionality
  - ➢ Back-annotation of timing, energy, …
  - ➢ Models of topology and glue logic

Virtual Platform Models



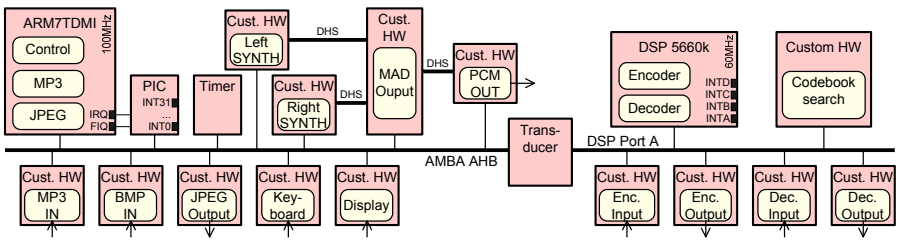
- **CPU model**
  - Source-level timing, energy, .. back-annotation
  - OS & processor models
- **Hardware/IP model**
  - Functional model
  - Timing, energy, … back-annotation
- **ISS model**
  - Cycle-accurate [GEM5]
  - Functional [QEMU] + timing, energy, … back-annotation

- **System-level design language (SLDL) & TLM backplane [SpecC, SystemC]**

Cellphone Example

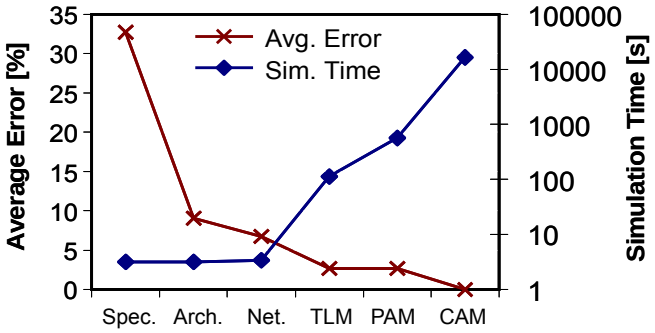
- **2 Subsystems**
  - ARM7TDMI
    - MP3 Decoding
    - Jpeg Encoding
  - Motorola DSP 56600k
    - GSM Transcoding

- **4 Accelerator HW blocks**
- **10 I/O HW blocks**
- **5 Busses**
  - AMBA AHB
  - DSP Port A bus
  - 3 Custom busses



Cellphone Results

- **Experimental setup**
  - 1.5 second MP3
  - 640x480 picture
  - 1.5 speech GSM
  - 3s / 300M ARM cycles / 180M DSP cycles
- **Simulation speed**
  - 300 Mcycles/s
- **Accuracy**
  - <3% error



- **Transaction-level modeling (TLM) of communication**
- **Host-compiled software, OS and processor modeling**

Lecture 7: Summary

- **System-level synthesis**
  - Decision making + refinement
- **System refinement**
  - Functionality and performance
- **System modeling**
  - Computation and communication