SYSTEM-LEVEL SYNTHESIS:

From Specification to Transaction-Level Models

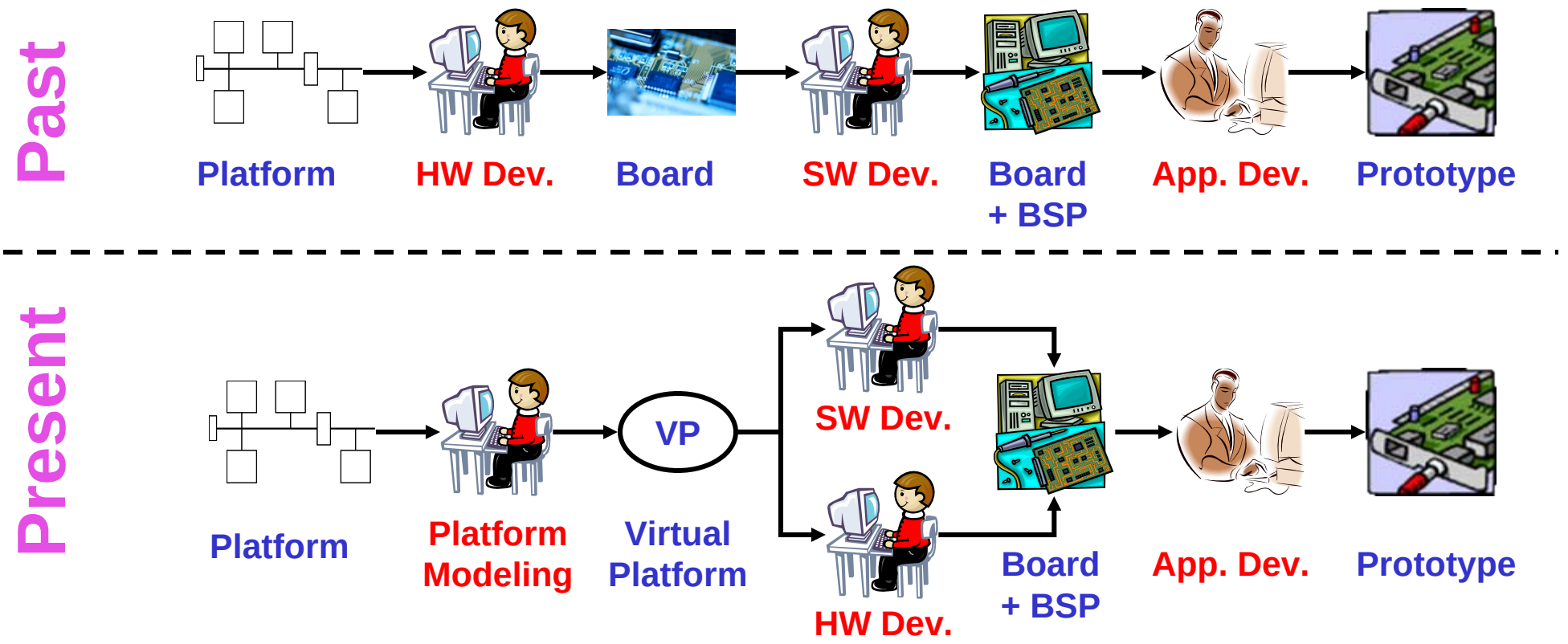
Daniel D. Gajski

Center for Embedded Computer Systems

University of California, Irvine

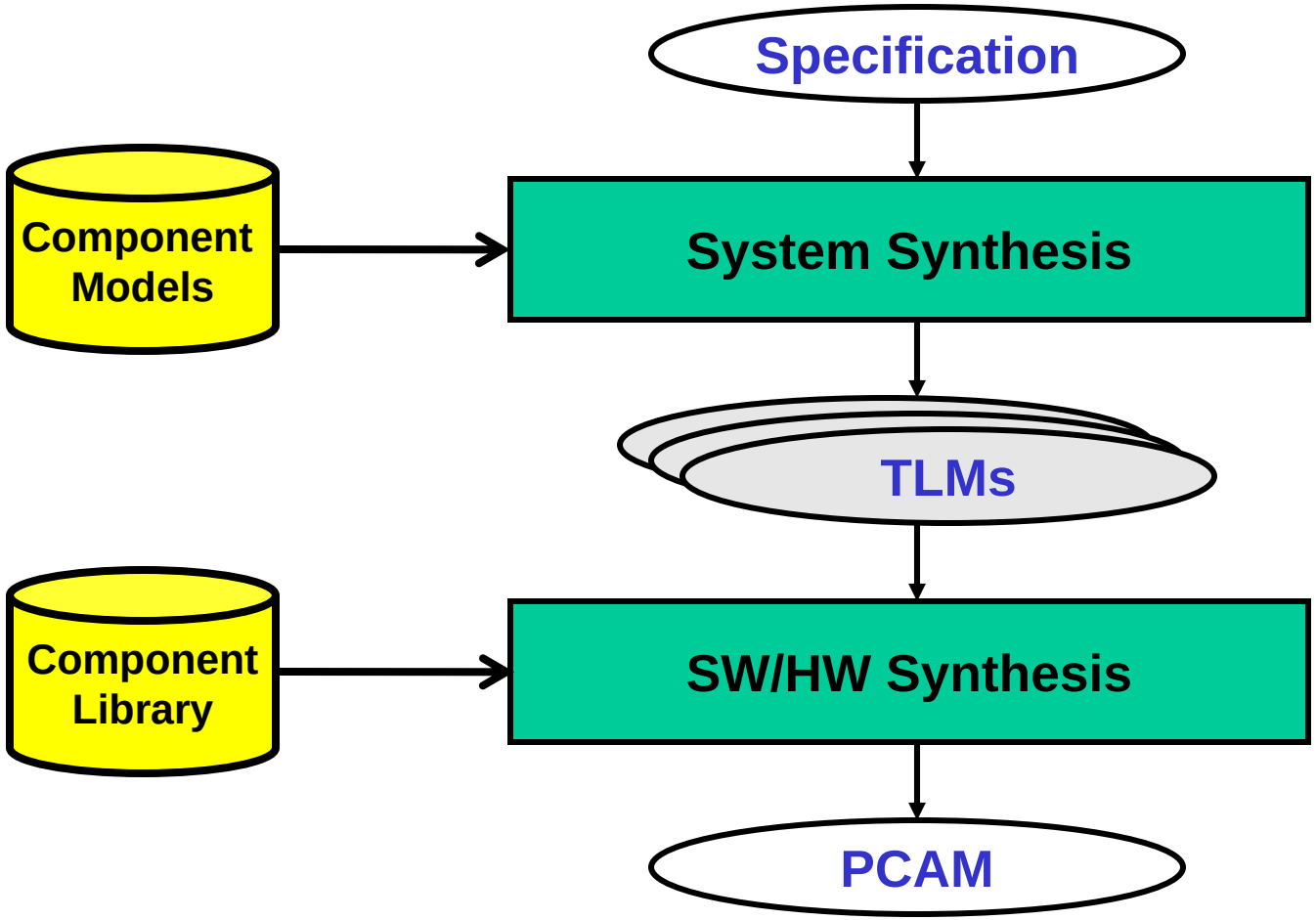
http://www.cecs.uci.edu/~ese/

System Design Trends

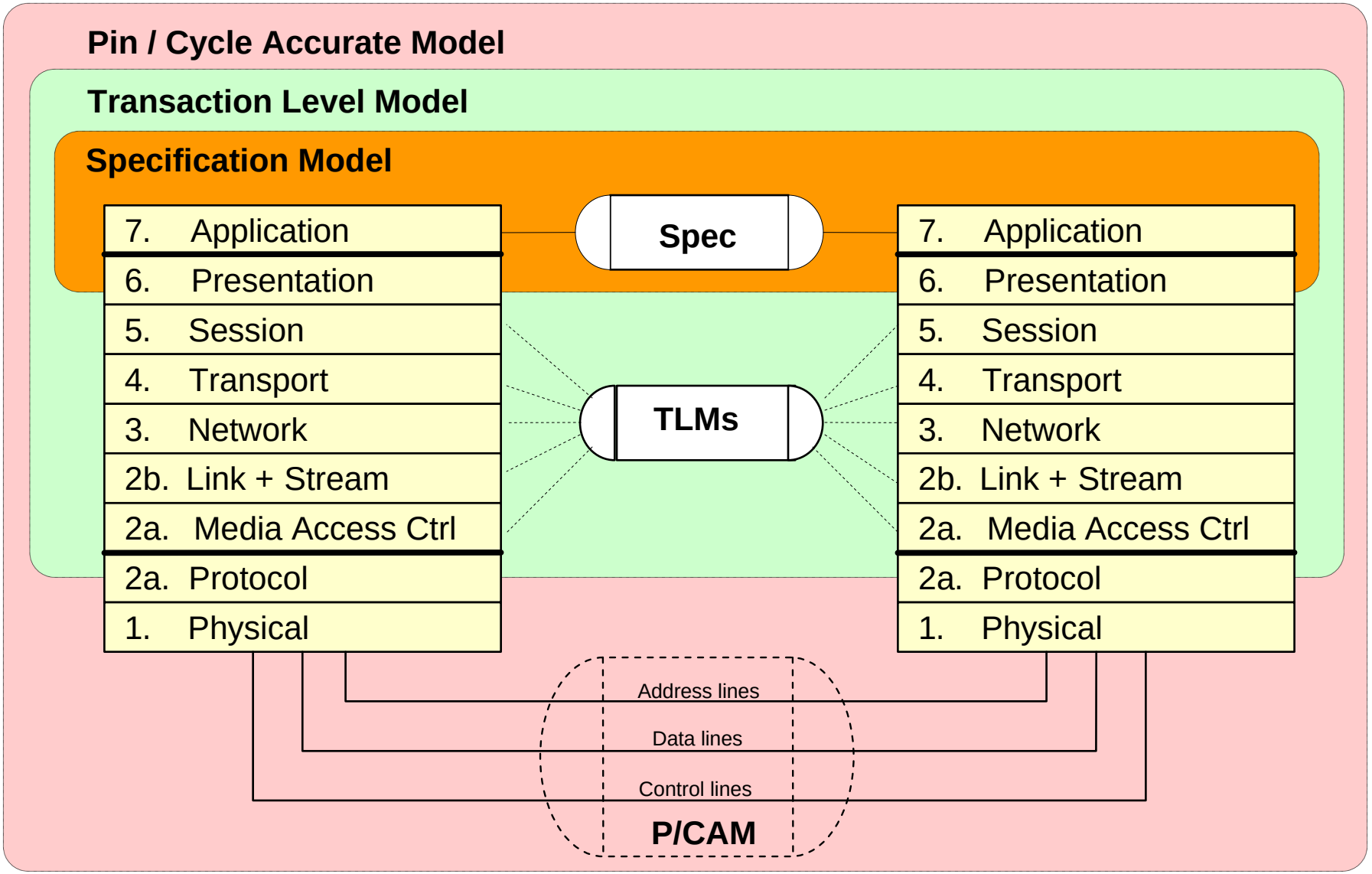


Source: S. Abdi

Model Based Synthesis



Three Models Needed

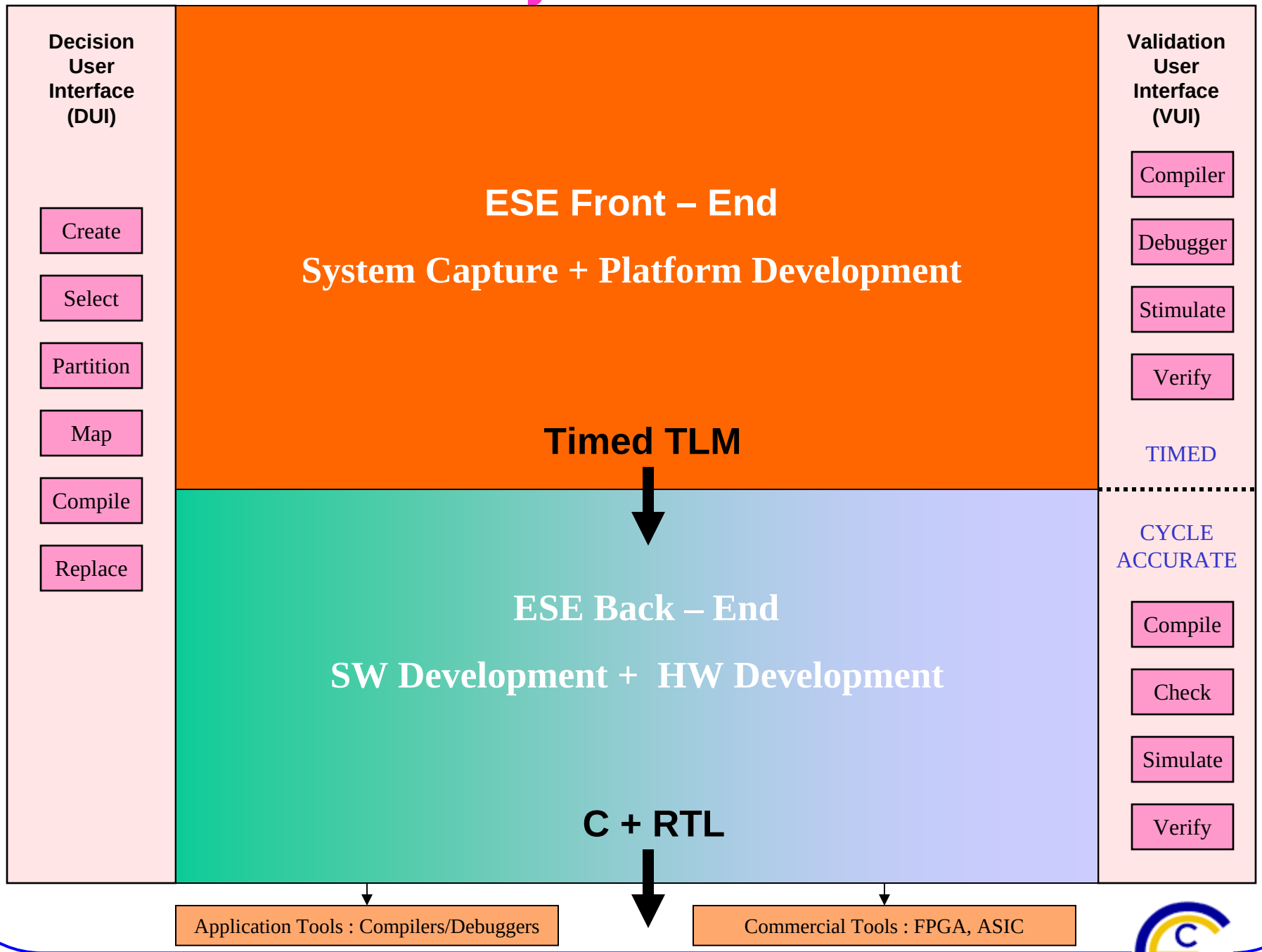


Source: G Schirner

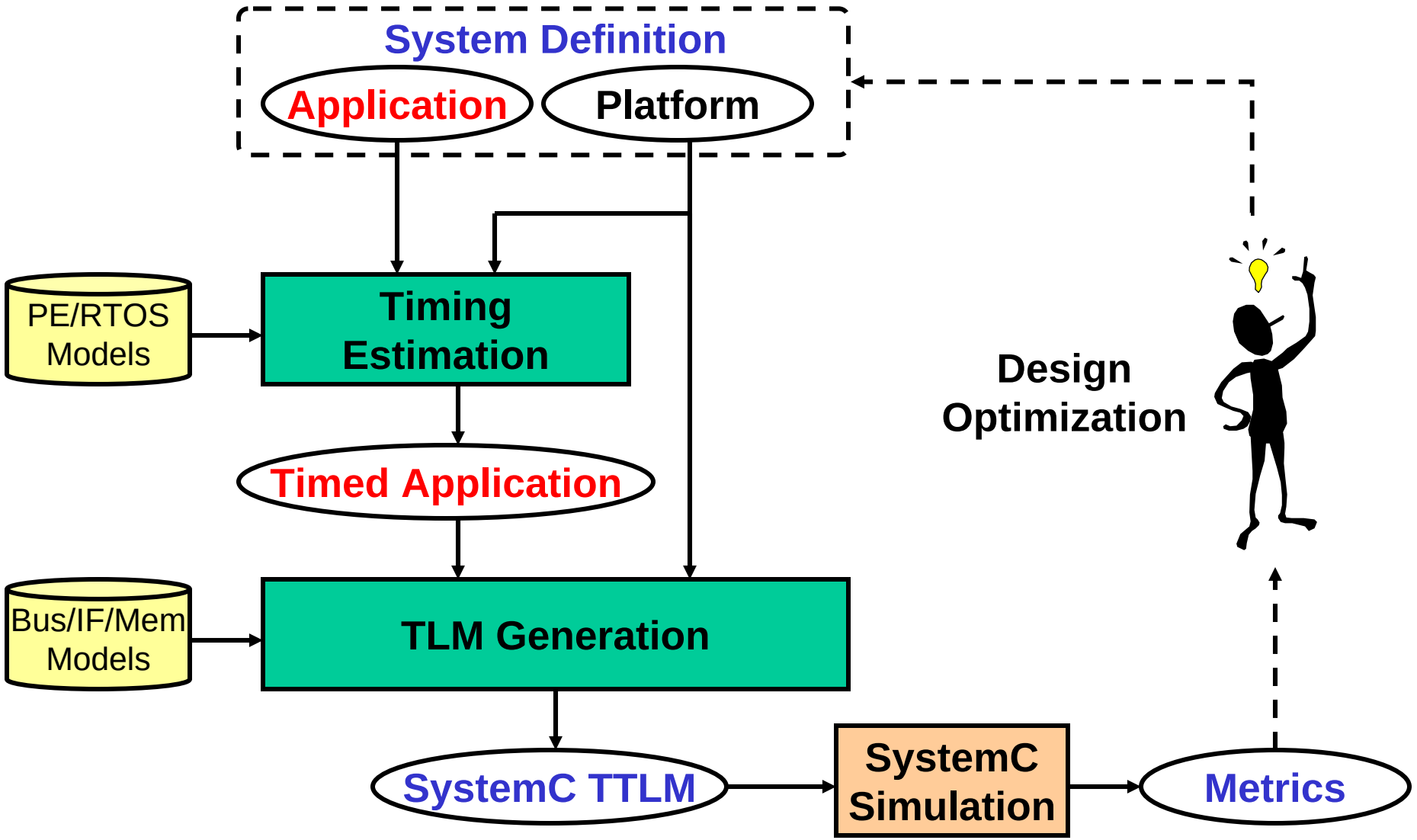
Model-Based-Synthesis Advantages

- No basic change in design methodology required
  - Design flow follows present design process
- Productivity gain of more than 1000X demonstrated
  - Designers do not write models
- Simple design update: 1-day change
  - No rework for new design decisions
- High error-reduction: Automation + verification
  - Error-prone tasks are automated
- Simplified globally-distributed design
  - Fast exchange of design decisions and easy impact estimates
- Benefit through derivatives designs
  - No need for complete redesign
- Better market penetration through customization
- Shorter Time-to-Market through automation

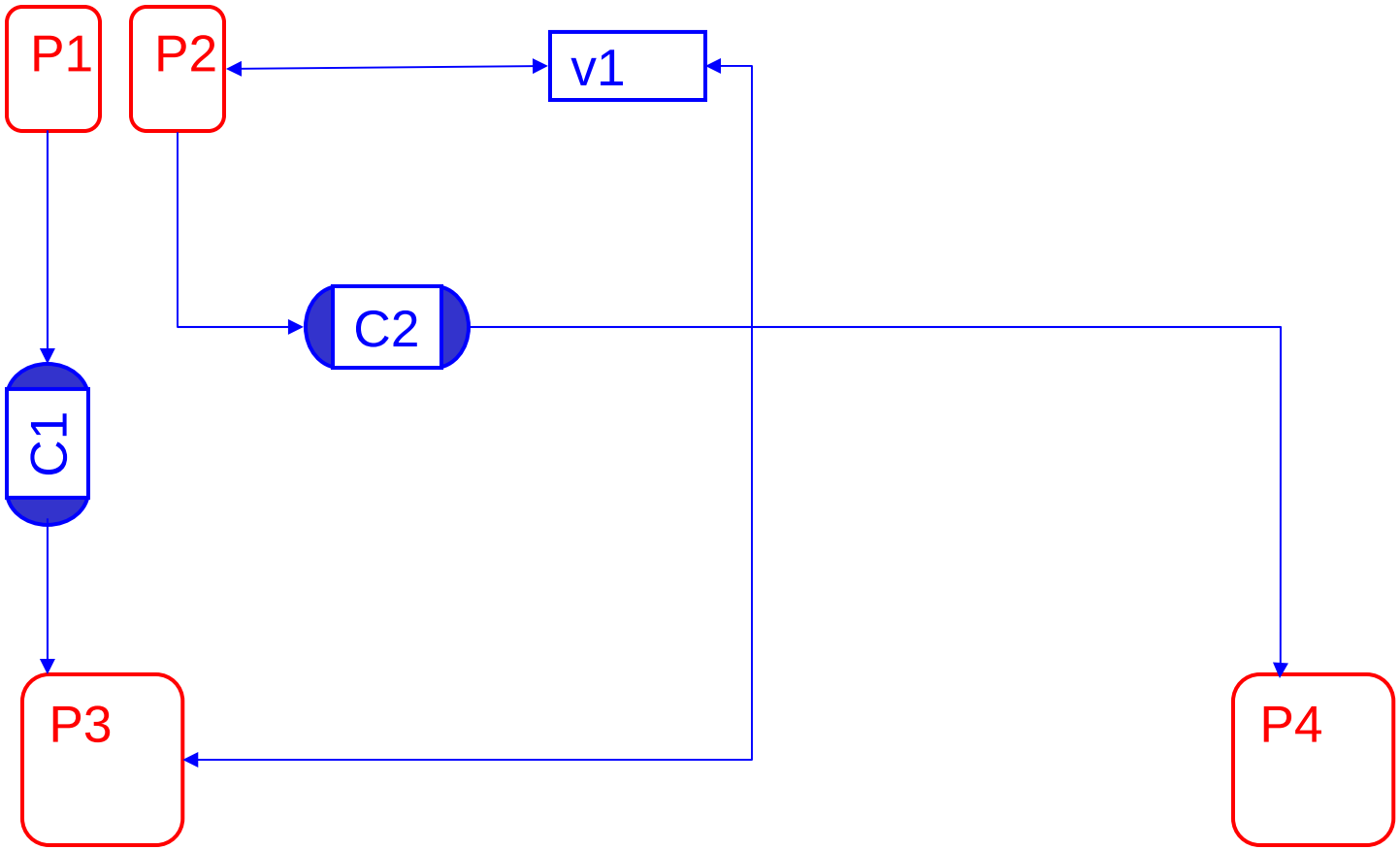
Embedded Systems Environment



ESE Front End Design Flow

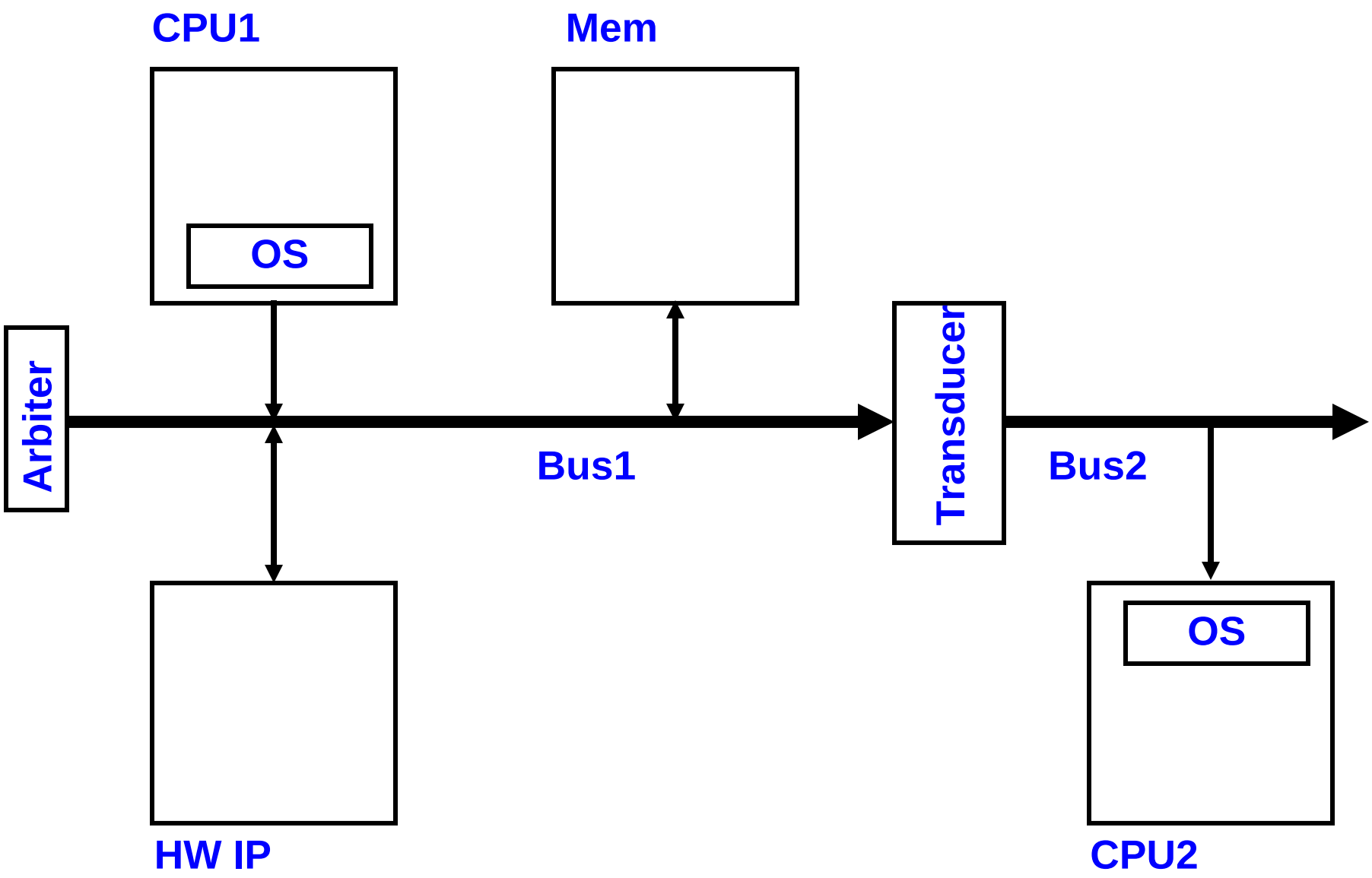


Application Model



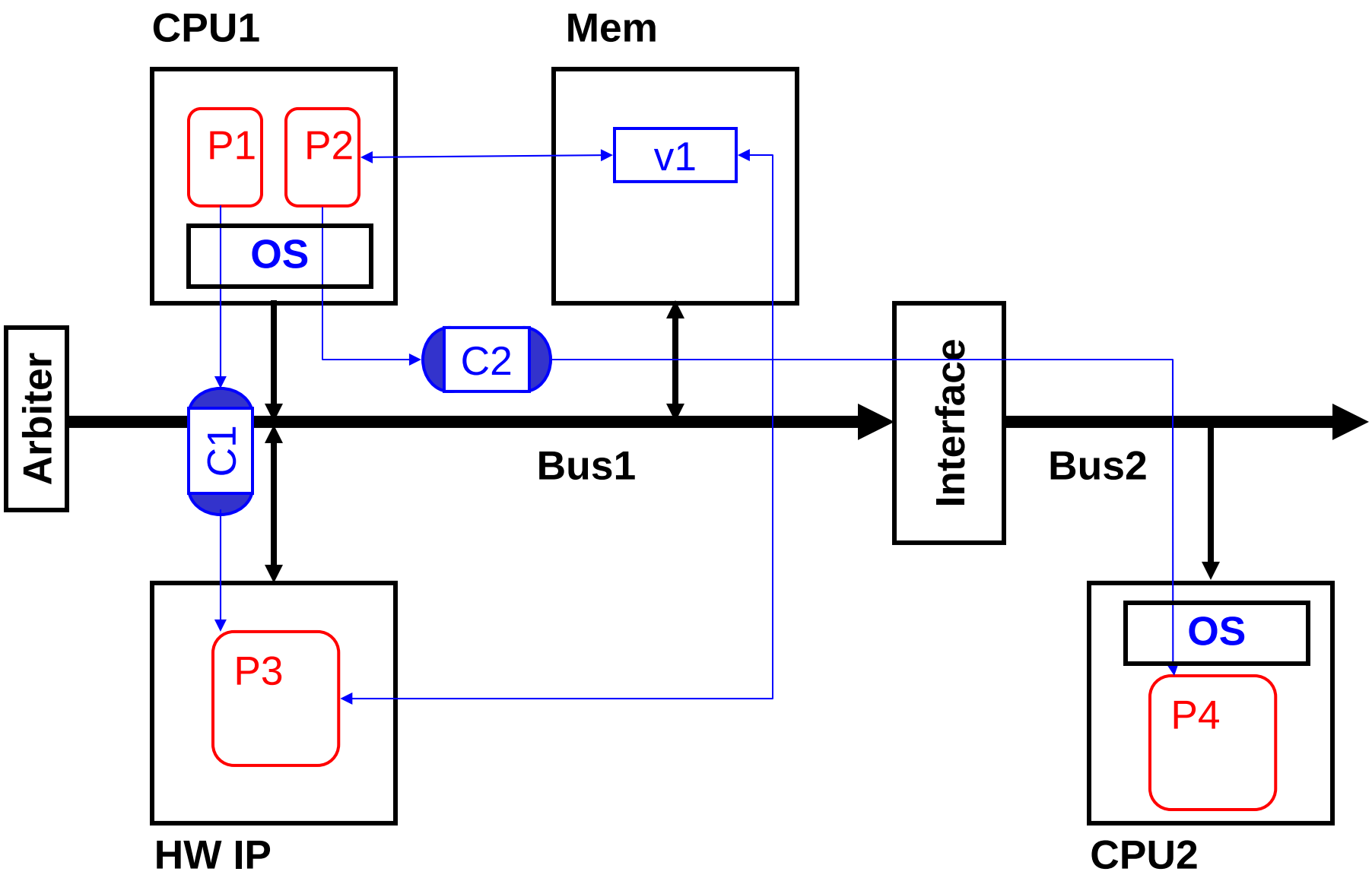
Application Model: set of communicating processes

Platform Architecture



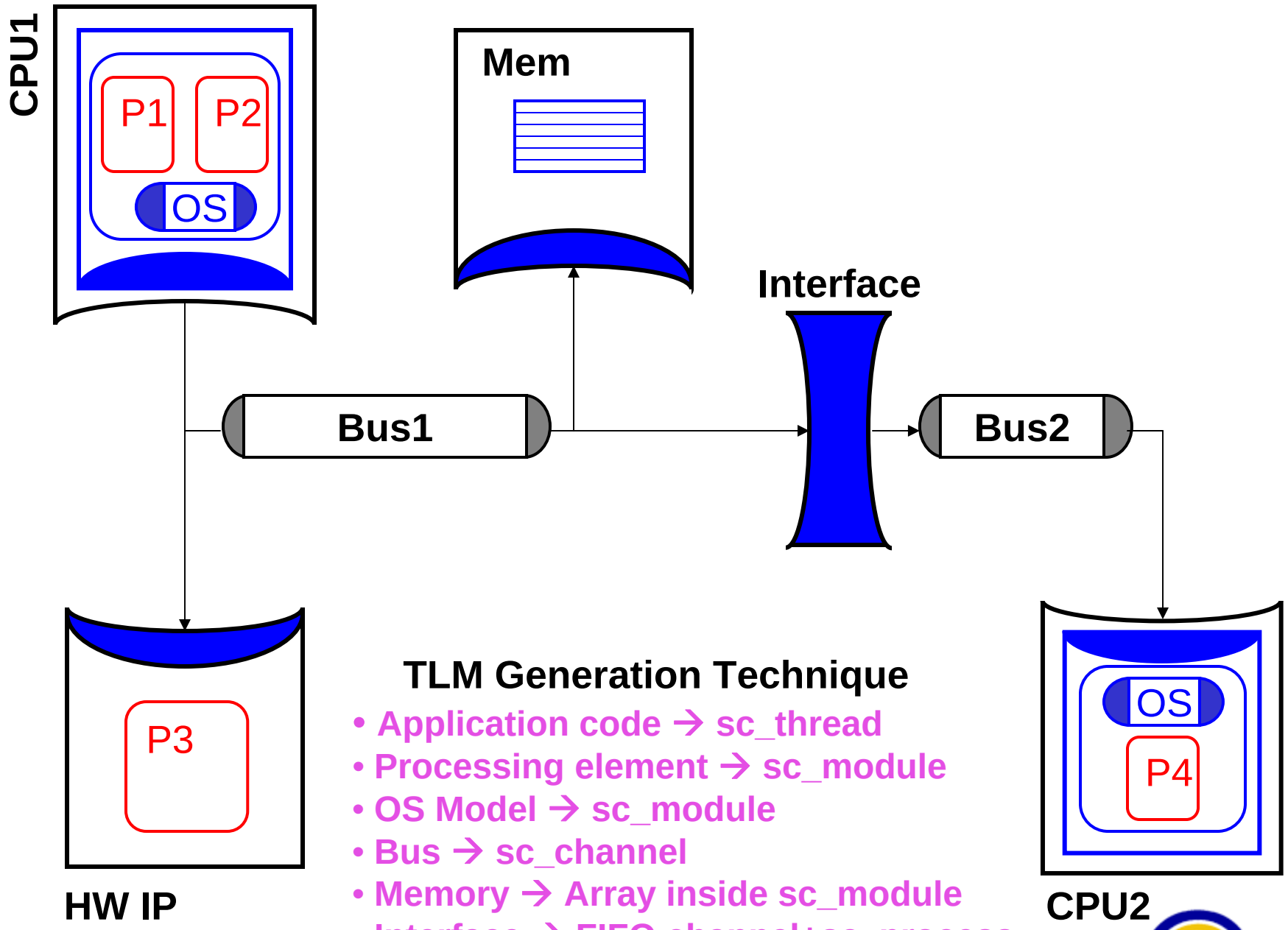
Netlist of SW processors, HW, buses and interfaces

Input: System Definition



System Definition* = *Platform* + *Application

Output: SystemC Timed TLM

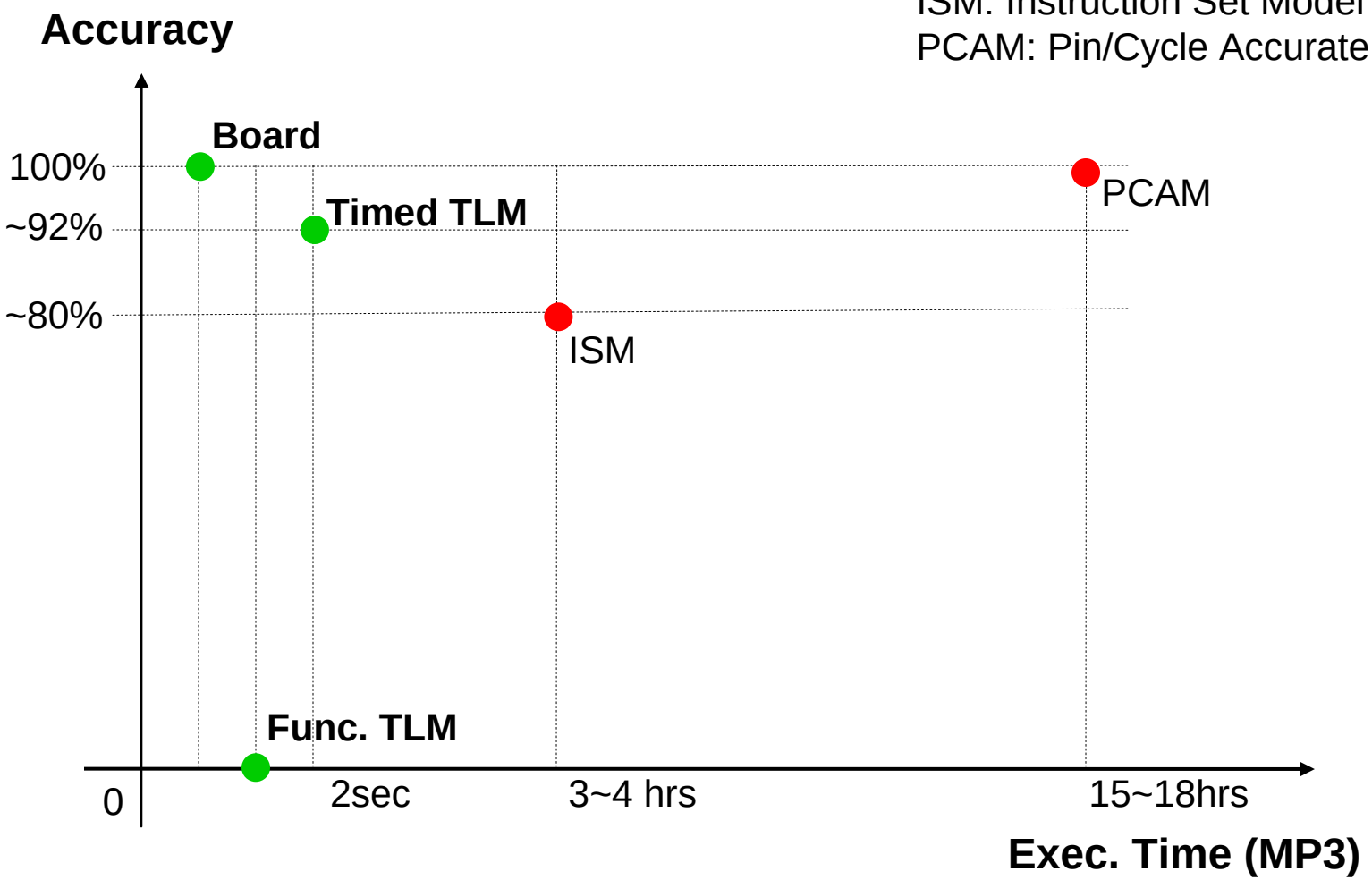


TLM Generation Technique

- Application code → sc_thread
- Processing element → sc_module
- OS Model → sc_module
- Bus → sc_channel
- Memory → Array inside sc_module
- Interface → FIFO channel+sc_process

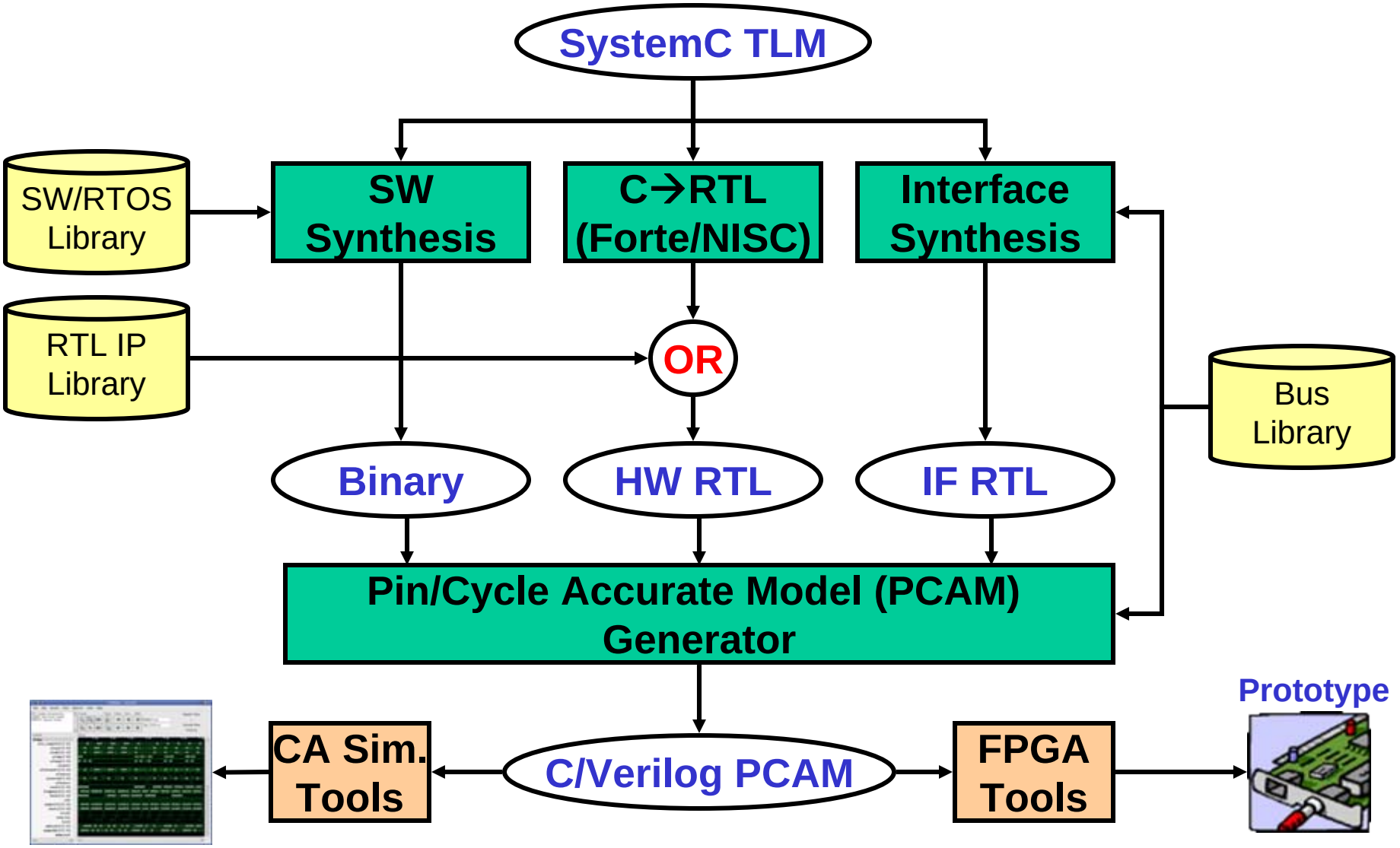
Model Accuracy vs. Execution Time

TLM: Transaction Level Model
ISM: Instruction Set Model
PCAM: Pin/Cycle Accurate Model

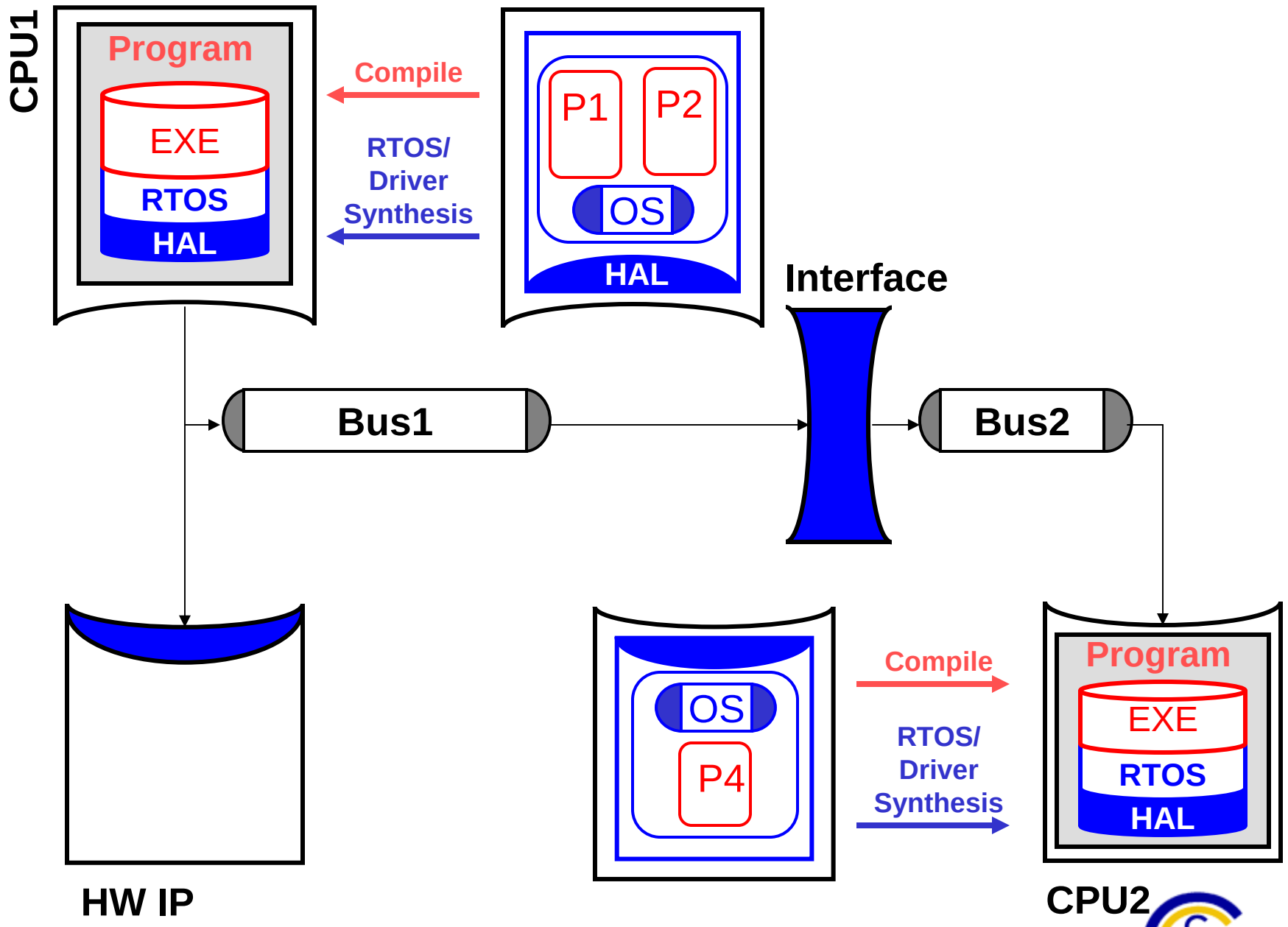


Time and accuracy trade off among different models

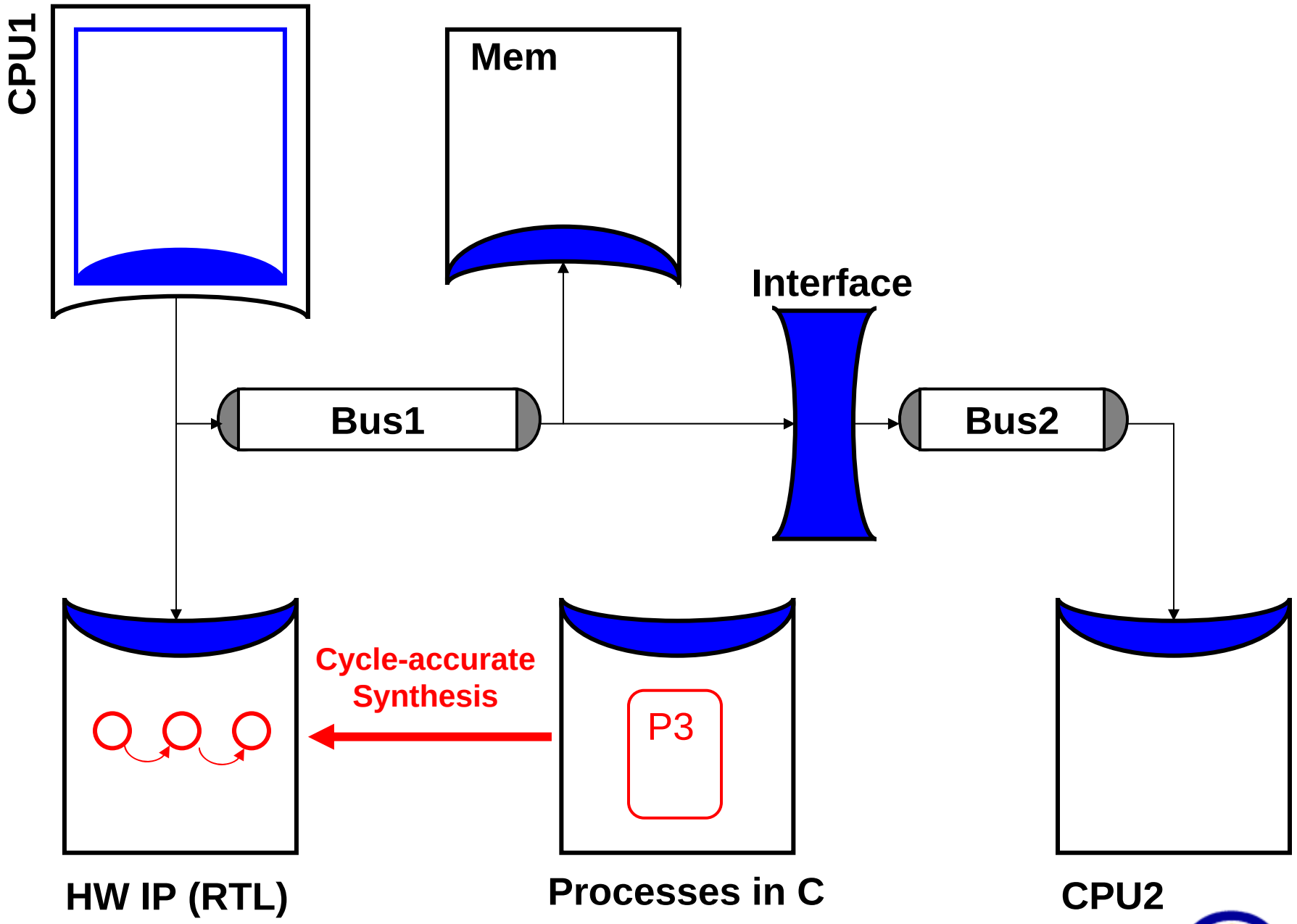
ESE Back End Prototyping Flow



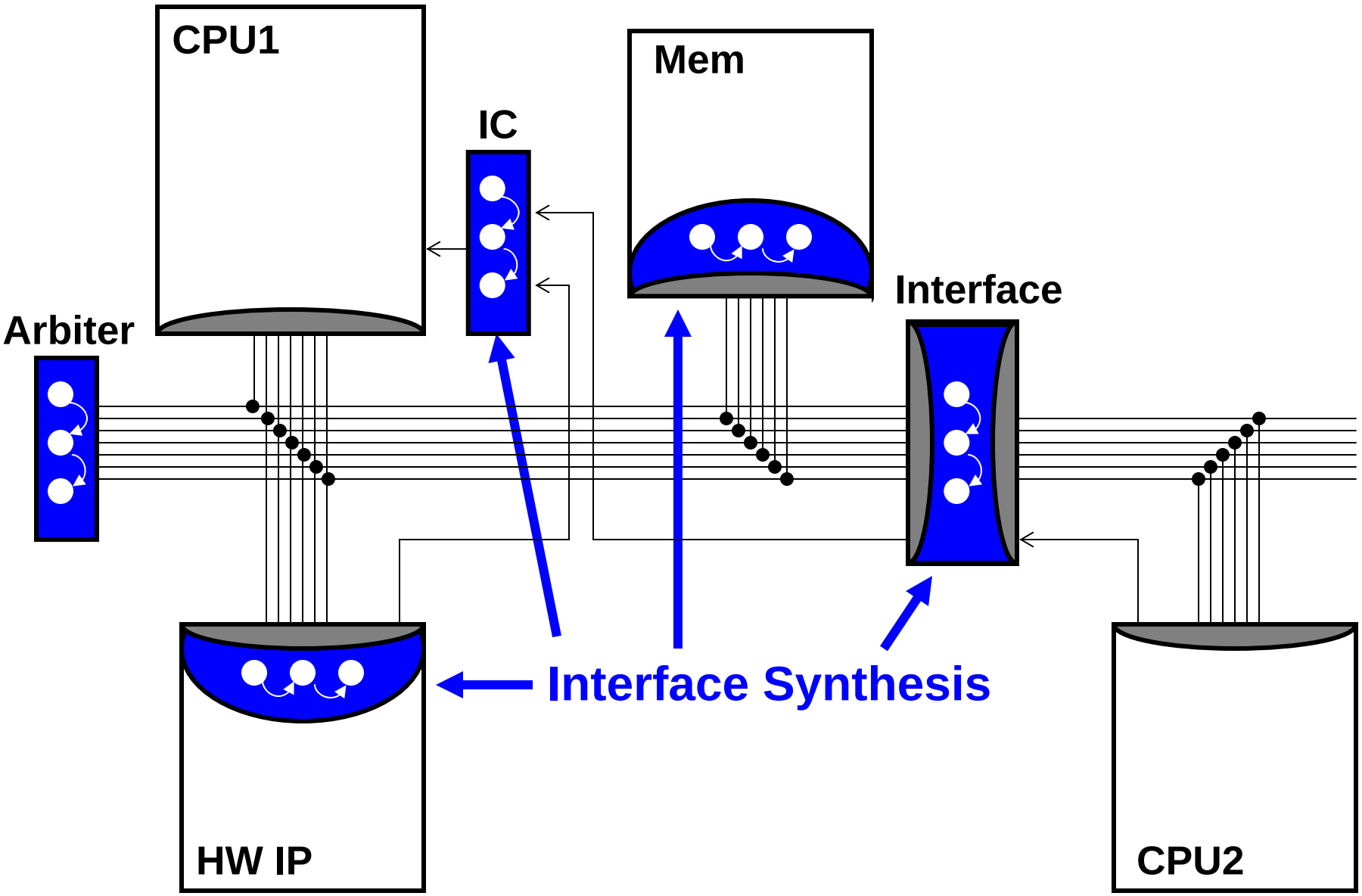
Cycle-Accurate Software Synthesis



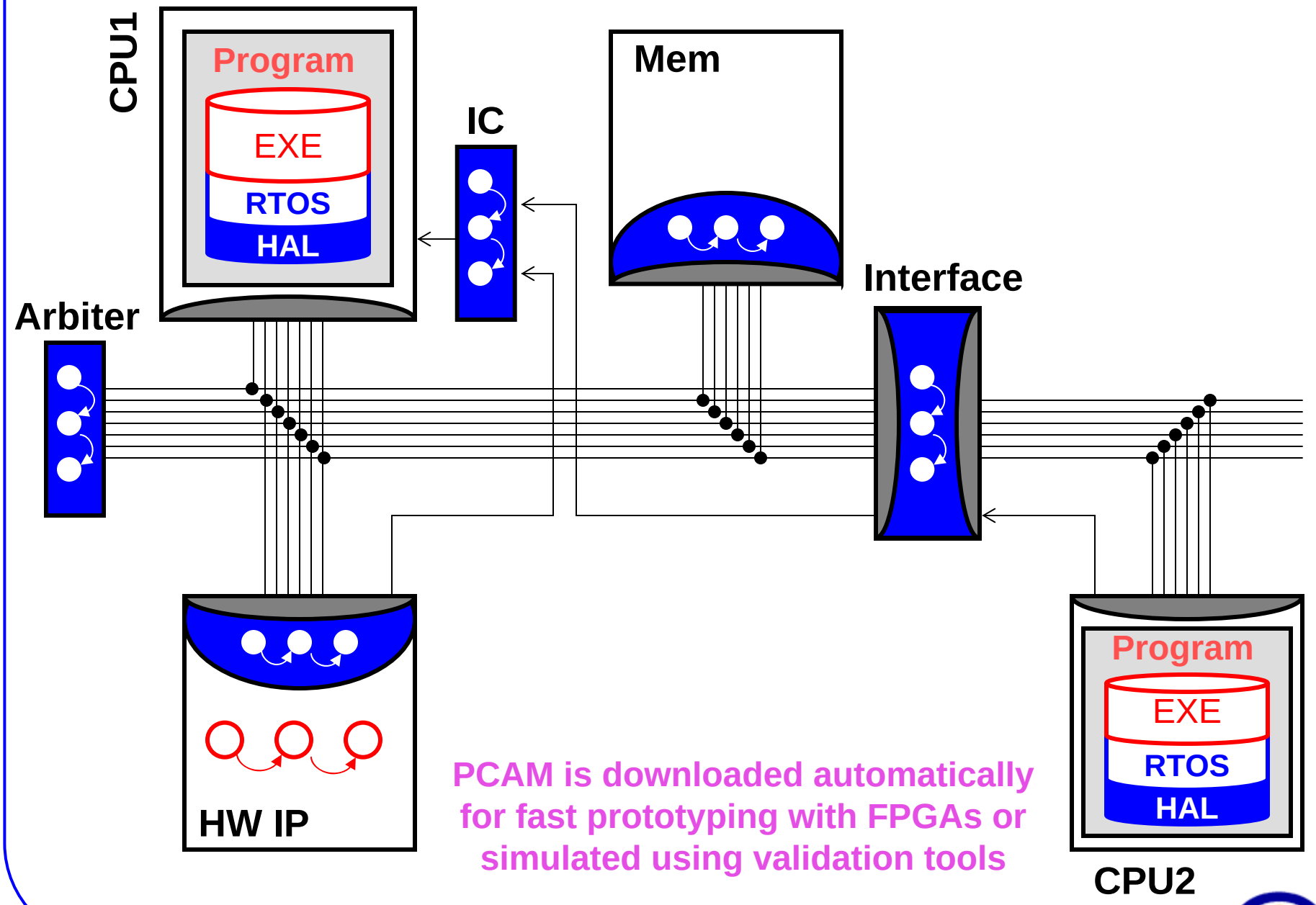
Cycle-Accurate Hardware Synthesis



Cycle-Accurate Interface Synthesis



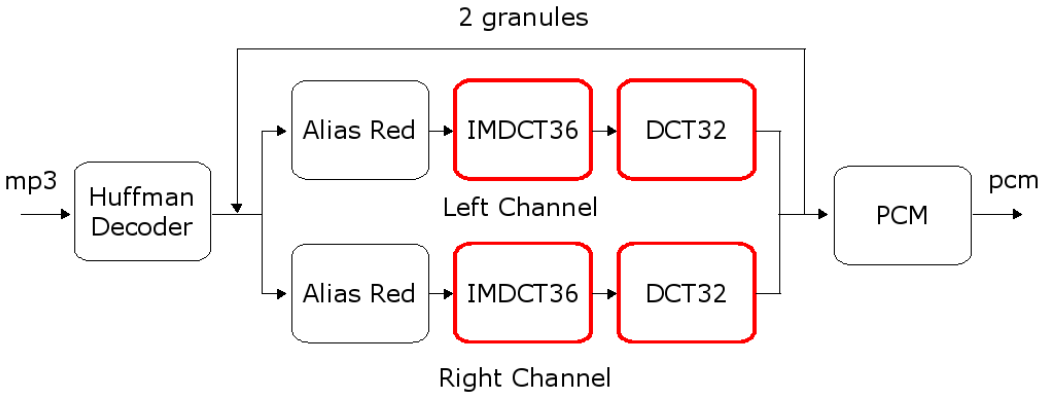
Pin/Cycle-Accurate Model

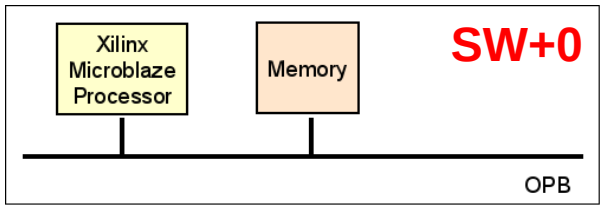


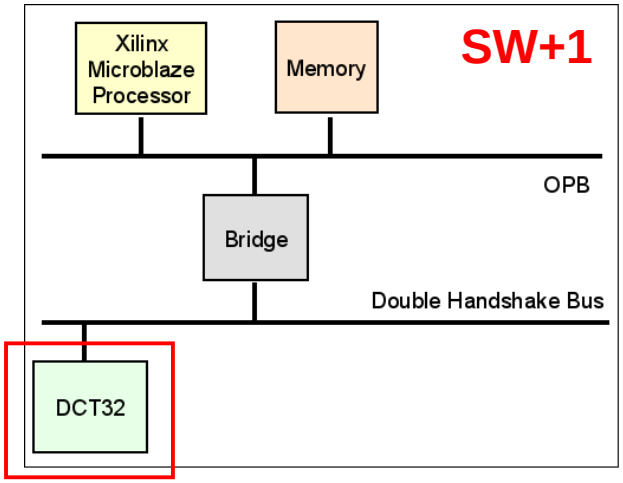
PCAM is downloaded automatically for fast prototyping with FPGAs or simulated using validation tools

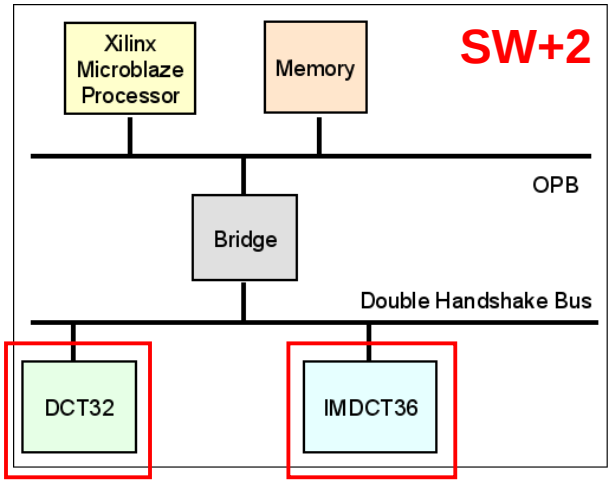
MP3 Design with ESE

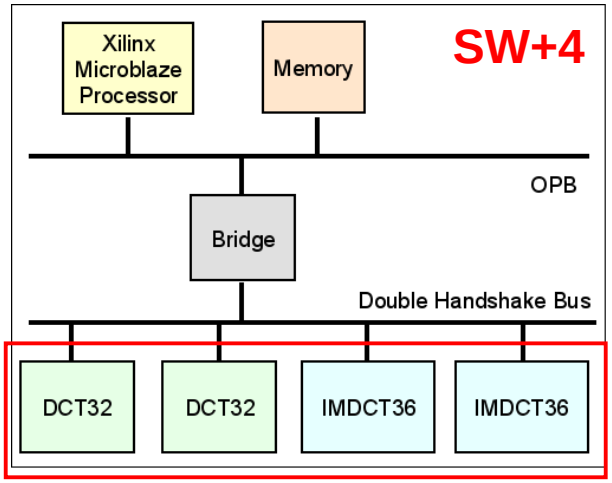
- **MP3 Decoder(>12K LOC) on Xilinx Multimedia FPGA board**
  - Microblaze soft-core with 0/1/2/4 HW components



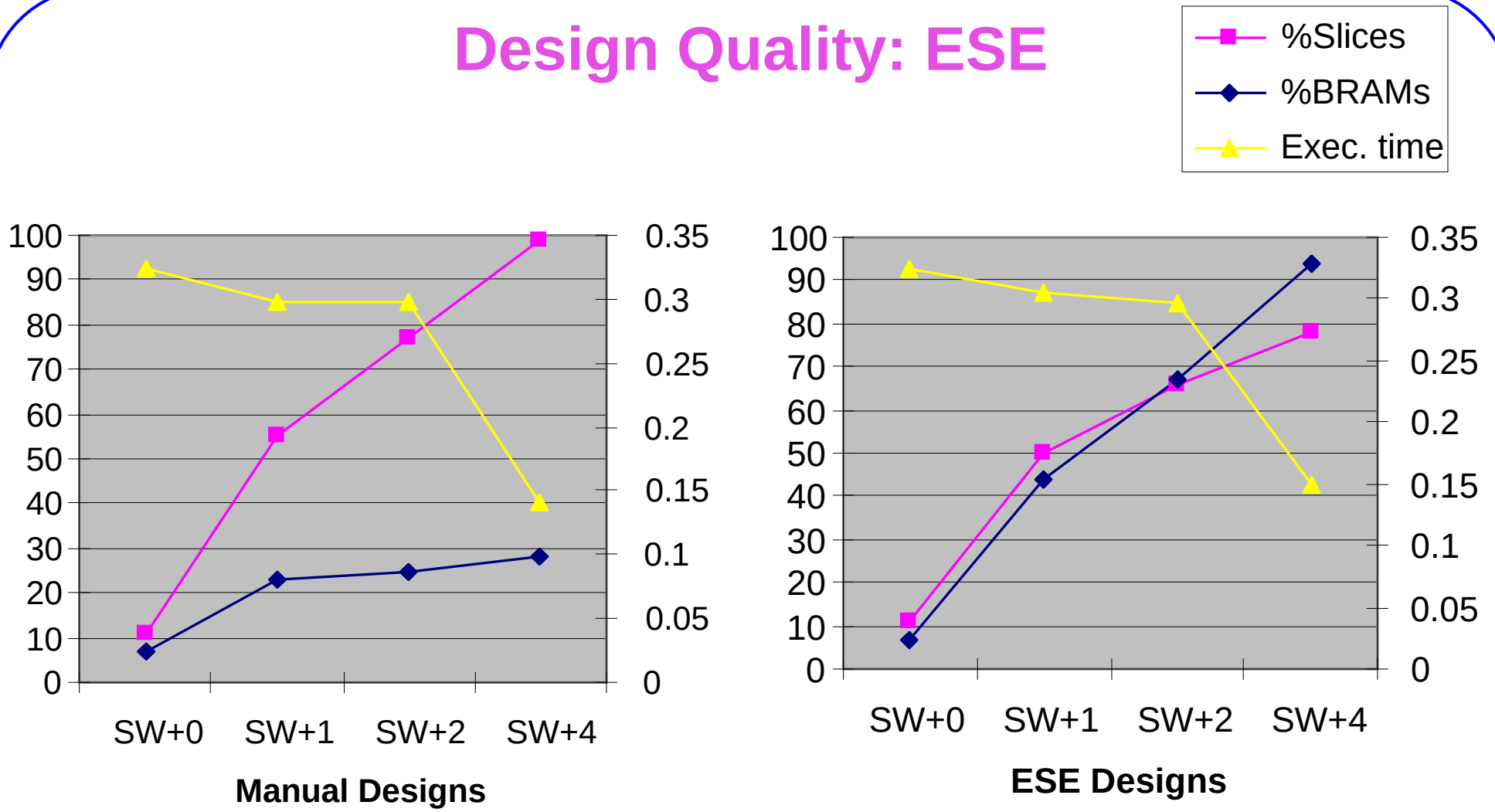






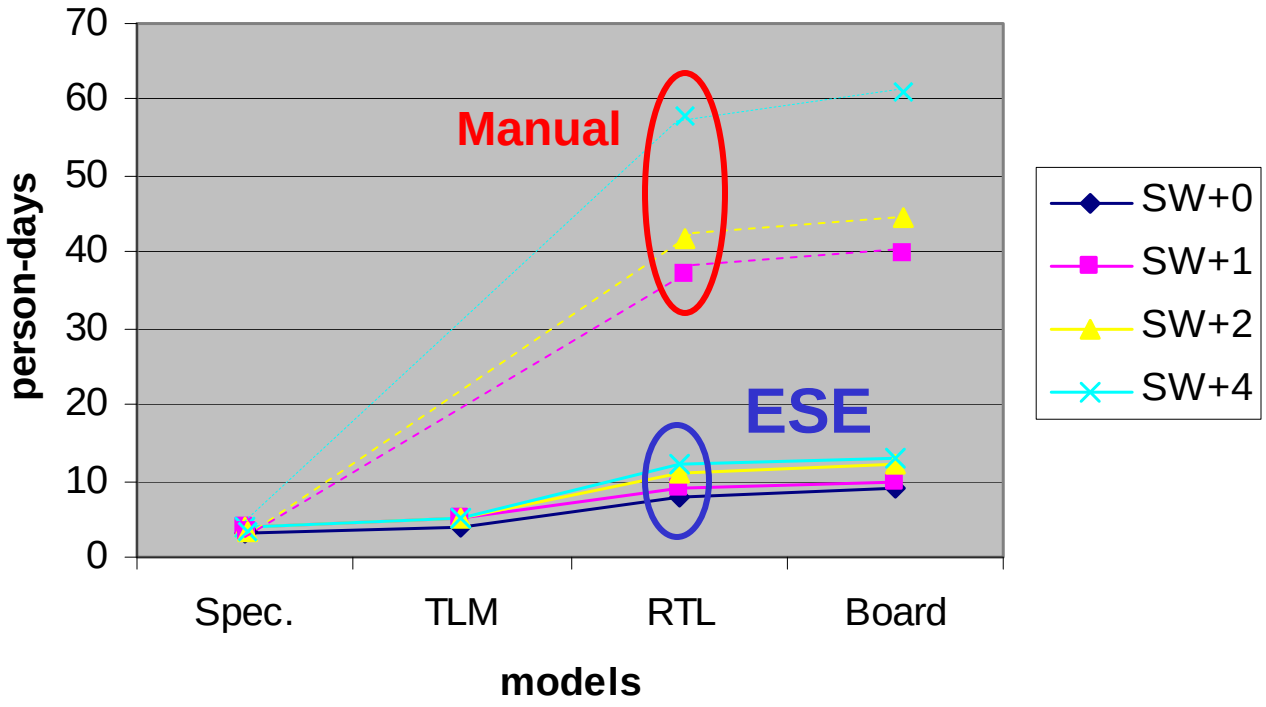


Design Quality: ESE



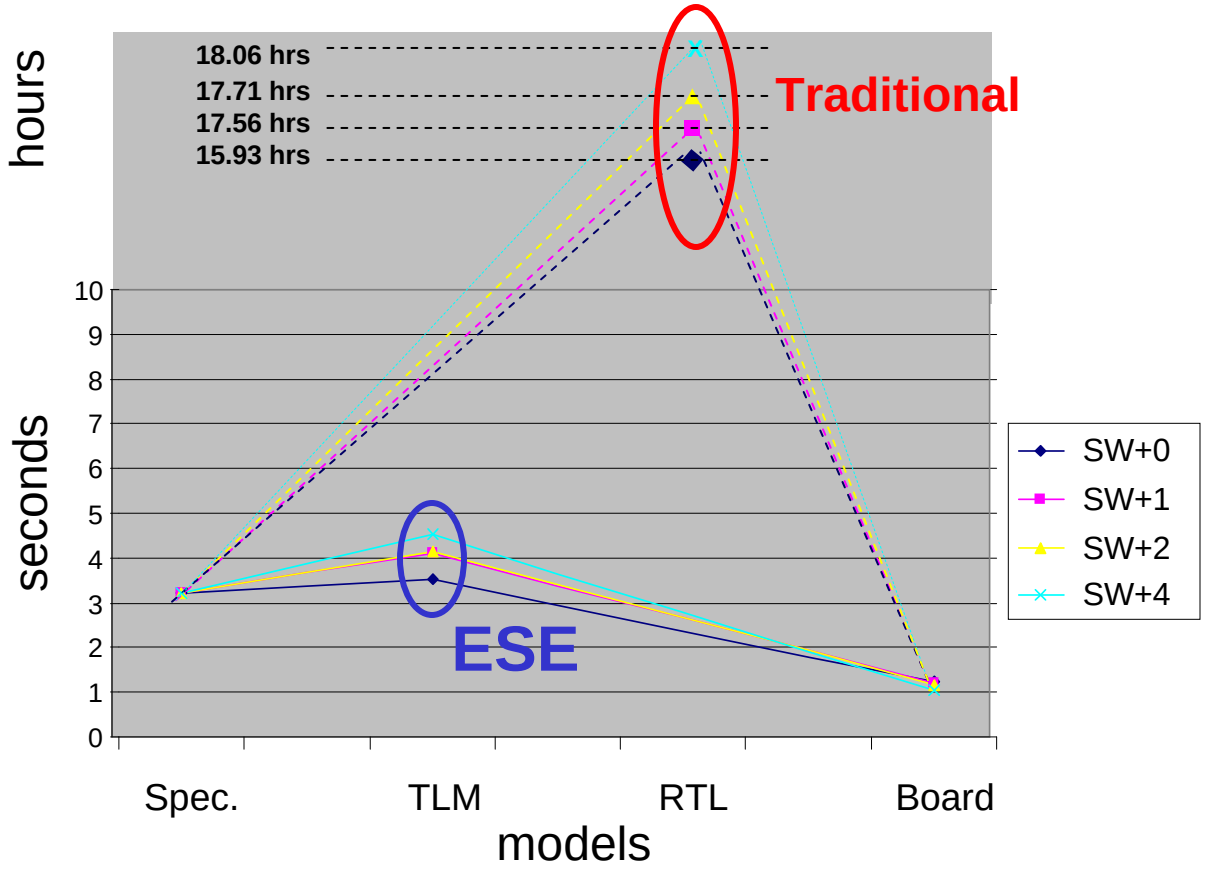
- **Area**
  - ESE designs use fewer FPGA slices and more BRAMs than manual HW
- **Performance**
  - ESE designs execute at same speed as manual designs

Development Time: ESE vs. Manual



- **ESE drastically cuts RTL and Board development time**
  - Manual development includes months of RTL coding
  - Models can be developed at Spec level with ESE
  - TLM, RTL and Board models are generated automatically by ESE

Validation Time: ESE vs. Traditional



- **ESE cuts validation time from hours to seconds**
  - No need to verify RTL models for every design change
  - Designers can perform high speed validation with TLM and board

Technology Summary

- **C based application input**
  - Supports model based design and legacy reuse
- **Automatic functional and timed TLM generation**
  - Enables early design validation and reliable estimation
- **Automatic SW synthesis**
  - Provides modular, verifiable, platform specific SW code
- **Automatic interface synthesis**
  - Allows rapid implementation of heterogeneous networks
- **FPGA and C/HDL export**
  - Generates standard input for commercial prototyping and CA validation tools

Business Benefits

- Huge productivity gain
- Easy change and upgrade management
- Model-based design and manufacturing
  - Executable specification for suppliers
  - Easy verification of supplied parts
  - Easy integration, assembly, repair
- New platform exploration and definition possibilities
- In-house early prototyping possibilities
- Early evaluation and implementation of standards
- Huge competitive edge

Acknowledgement

I would like to thank ESE team at CECS that has contributed many ideas through numerous discussions and made ESE tool work:

Samar Abdi

Lochi Yu

Yonghyun Hwang

Hansu Cho

and

Gunar Schirner

Thank You