System-Level Design
(and Modeling for Embedded Systems)

Lecture 6 – System Synthesis and Exploration

Kim Grüttner kim.gruettner@dlr.de
Henning Schlender henning.schlender@dlr.de
Jörg Walter joerg.walter@offis.de

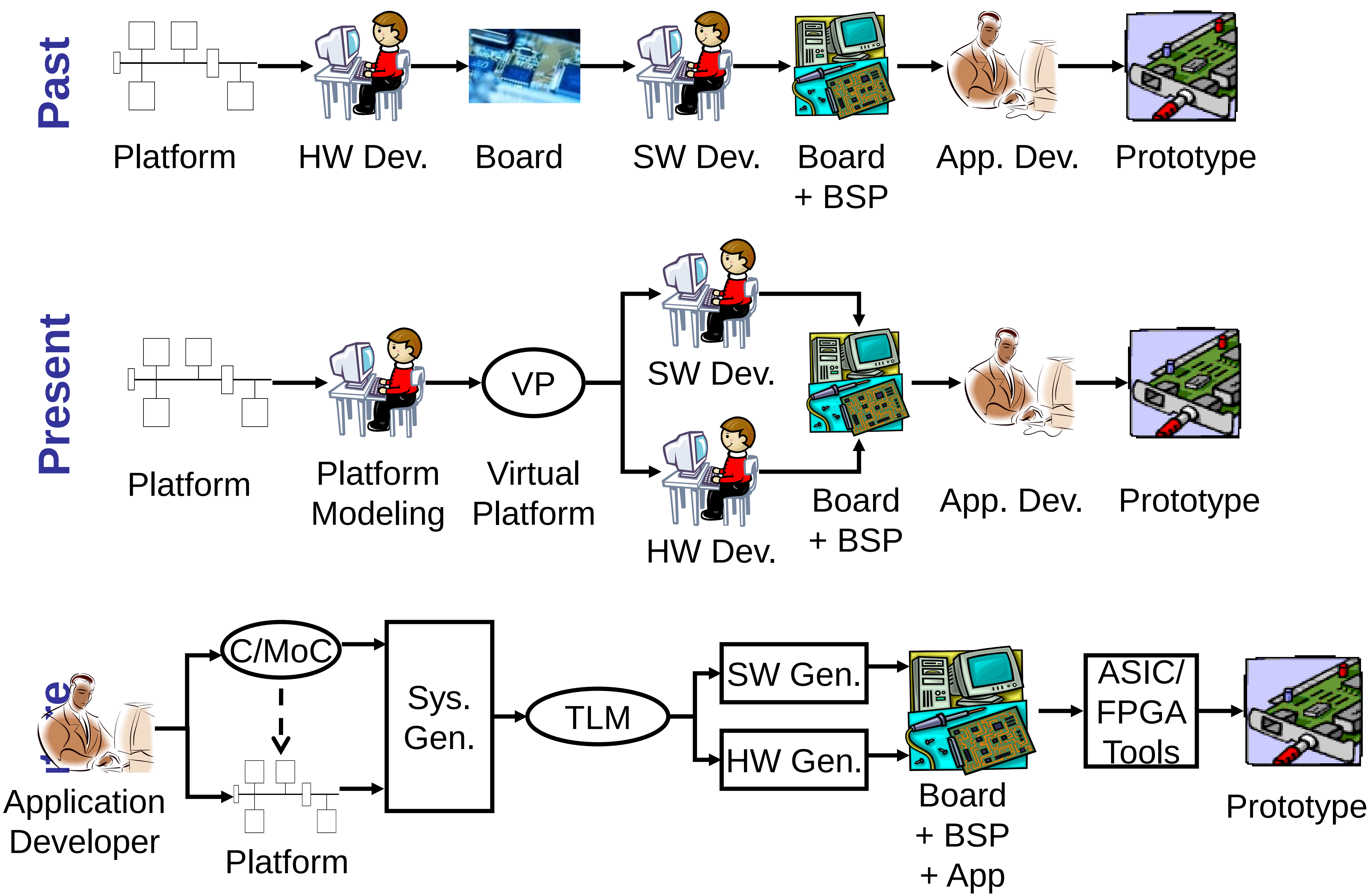
System Evolution and Operation
German Aerospace Center (DLR)
&
Distributed Computation and Communication
OFFIS

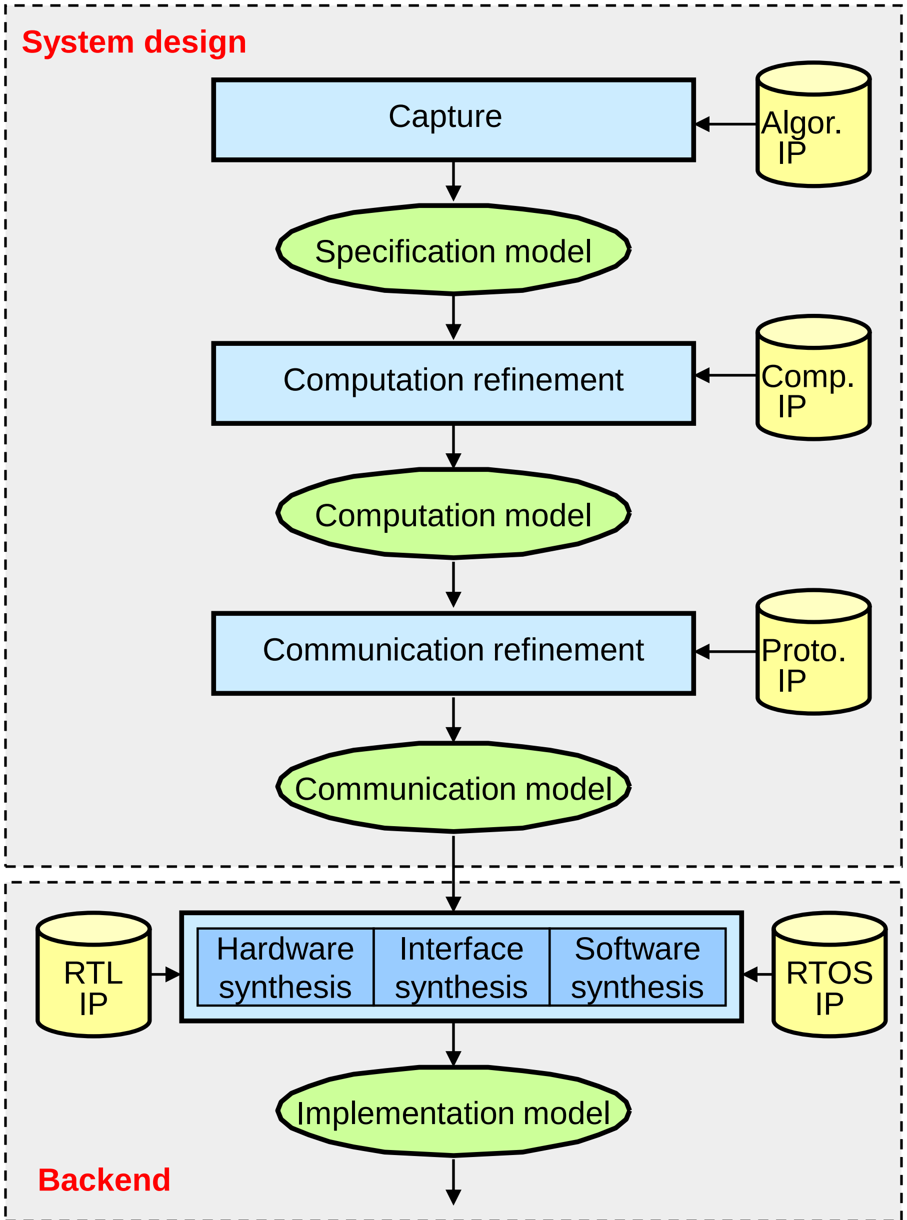
Electrical and Computer Engineering
University of Texas at Austin
gerstl@ece.utexas.edu

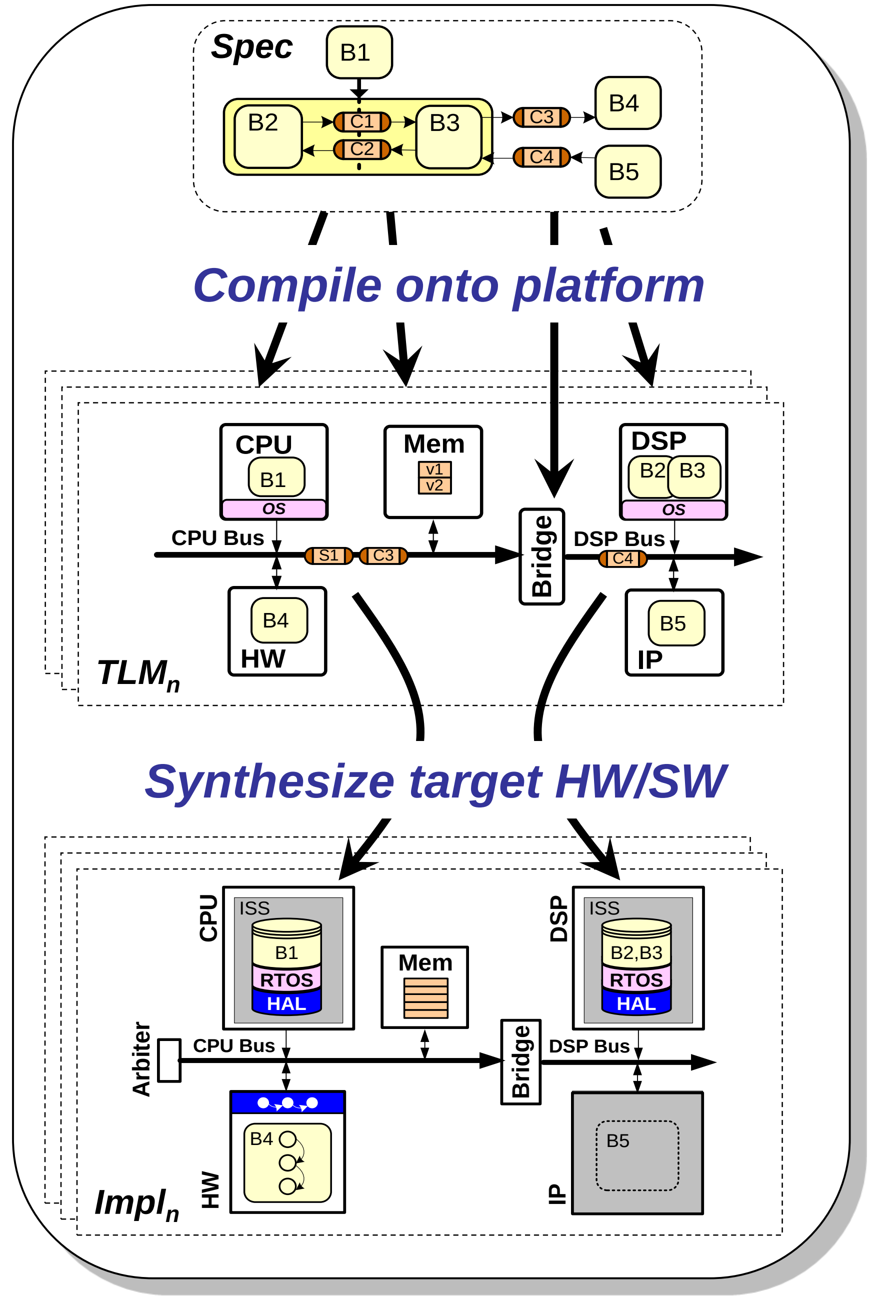
Lecture 6: Outline

- **System synthesis**
  - Synthesis process
- **Evaluation**
  - Profiling and simulation
  - Component estimation
  - Analytical and combined methods
- **Design space exploration**
  - Platform-based synthesis
  - Multi-objective optimization

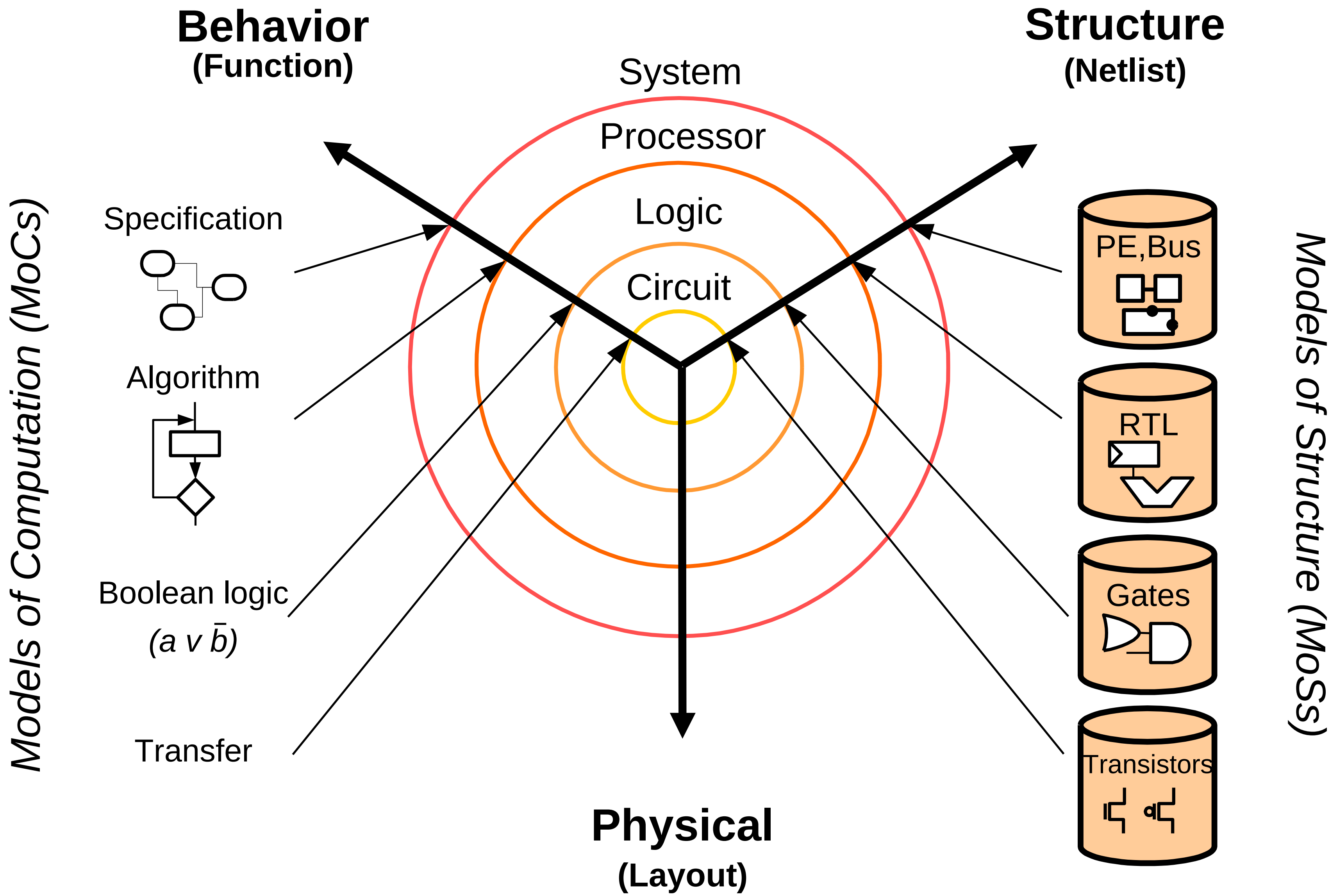
System Design Process




System design
Capture
Algor. IP
Specification model
Computation refinement
Comp. IP
Computation model
Communication refinement
Proto. IP
Communication model
RTL IP
Hardware synthesis
Interface synthesis
Software synthesis
RTOS IP
Implementation model
Backend

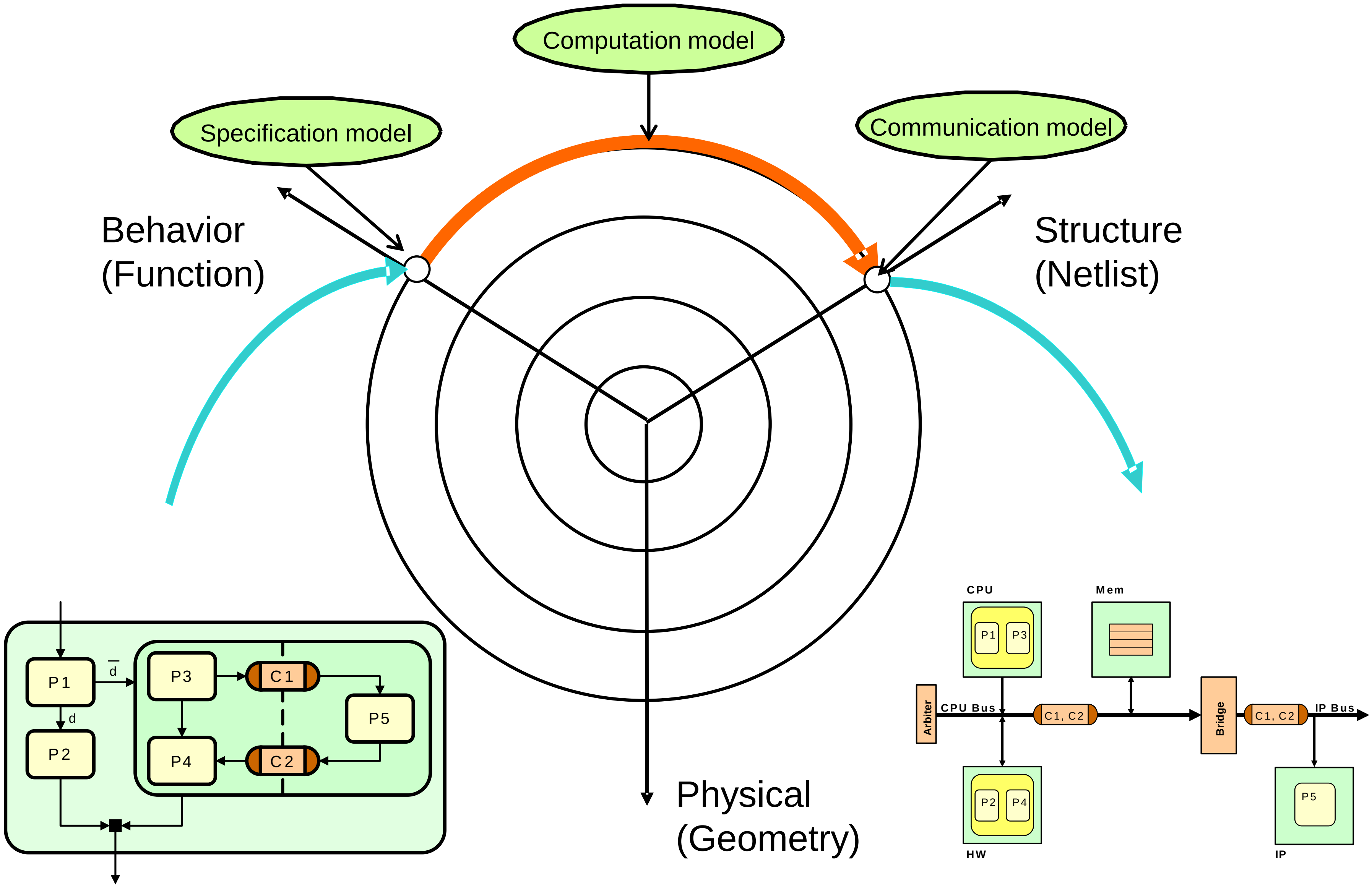

Spec
B1
B2
C1
C2
B3
C3
C4
B4
B5
Compile onto platform
CPU
B1
OS
Mem
v1
v2
DSP
B2
B3
OS
CPU Bus
S1
C3
Bridge
DSP Bus
C4
B4
HW
B5
IP
TLMn
Synthesize target HW/SW
CPU
ISS
B1
RTOS
HAL
Mem
DSP
ISS
B2,B3
RTOS
HAL
Arbiter
CPU Bus
Bridge
DSP Bus
B4
HW
B5
IP
Impln

Y-Chart



Behavior
(Function)
Structure
(Netlist)
System
Processor
Logic
Circuit
Models of Computation (MoCs)
Specification
Algorithm
Boolean logic
(a v b̄)
Transfer
Models of Structure (MoSs)
PE,Bus
RTL
Gates
Transistors
Physical
(Layout)

System Synthesis



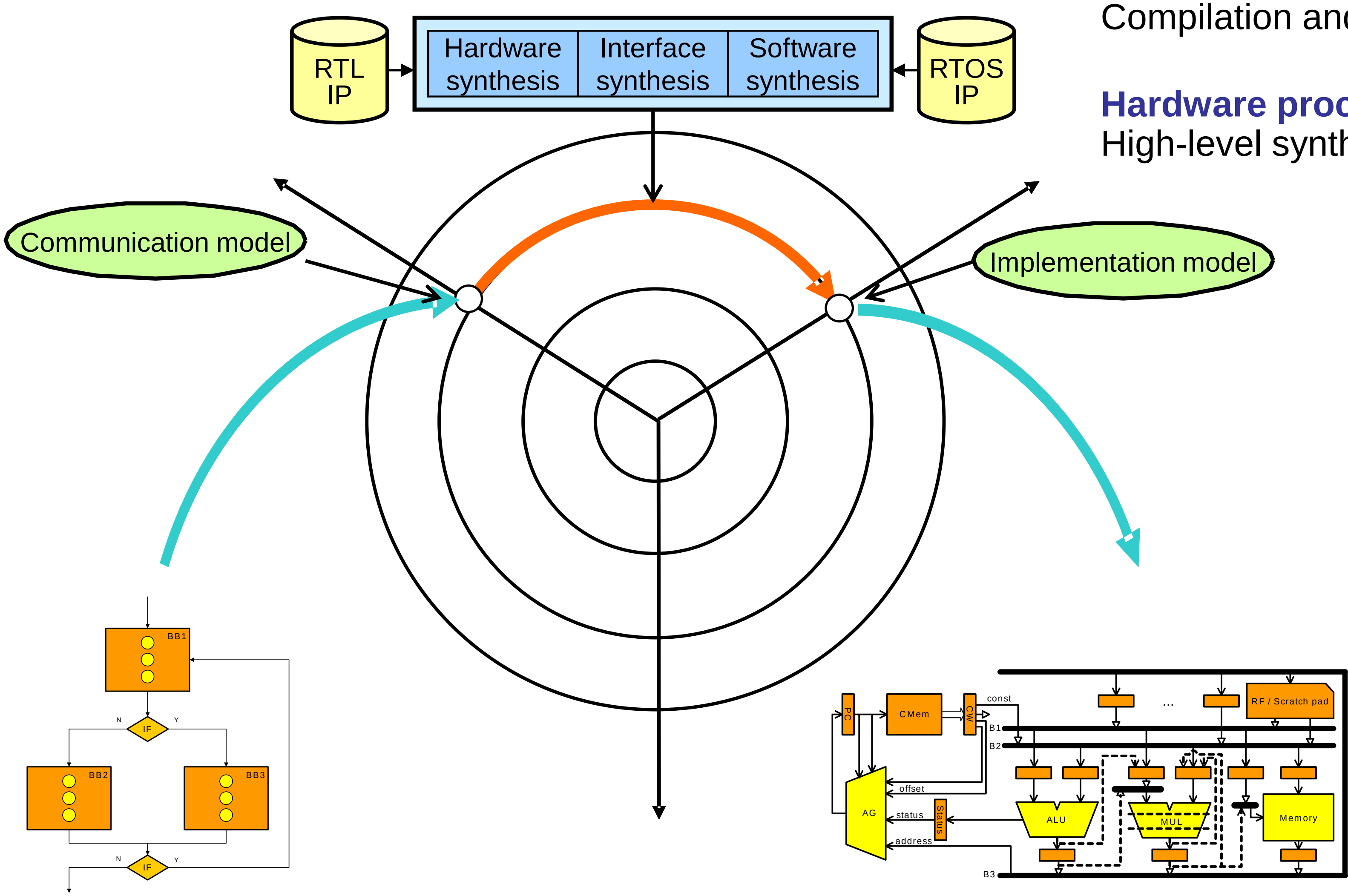
Specification model (processes)

Architecture model (TLM)

Processor Synthesis

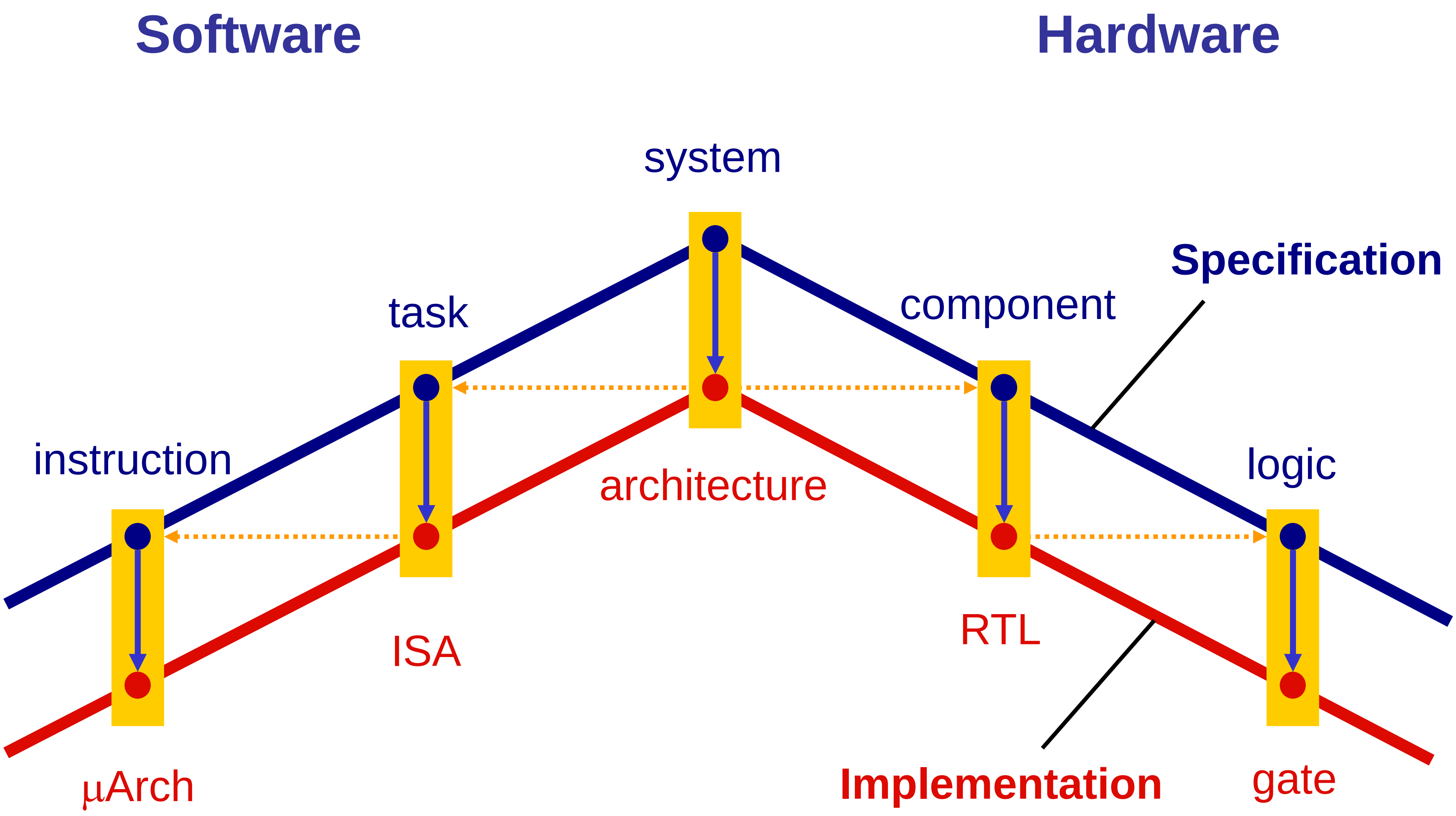
Software processor:
Compilation and linking

Hardware processor:
High-level synthesis

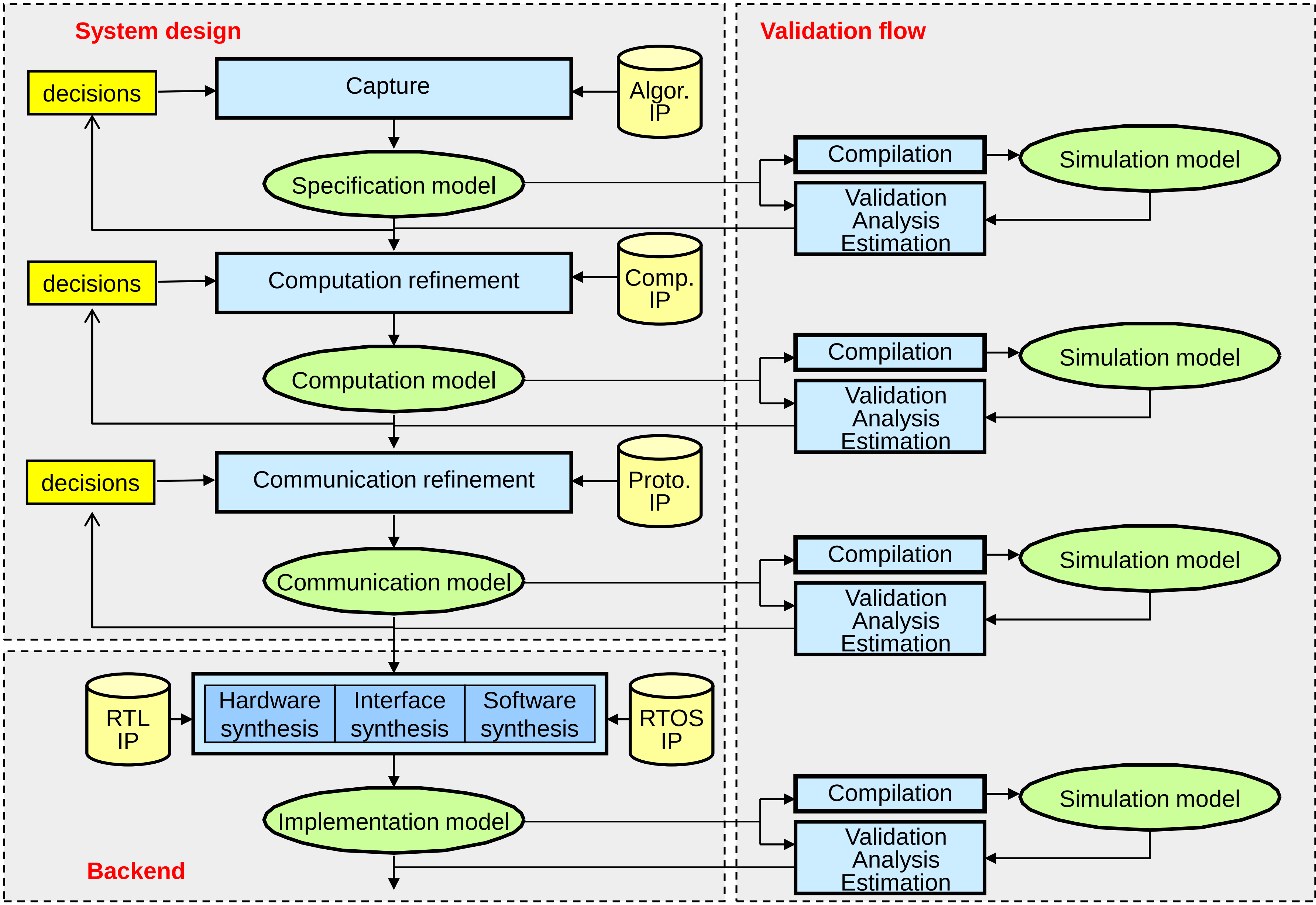


Algorithm model (program) **Microarchitecture model (RTL)**

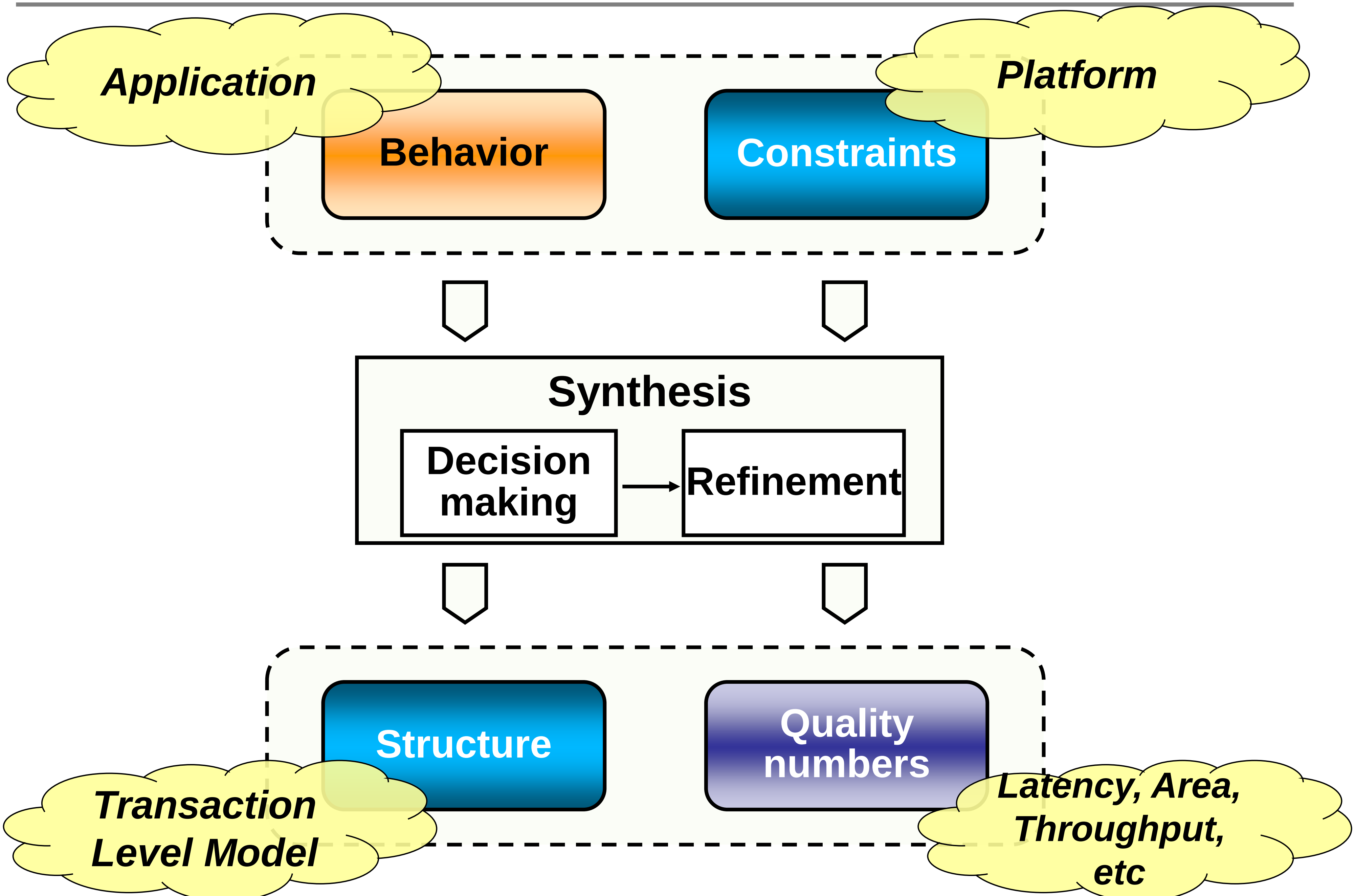
Double Roof Model



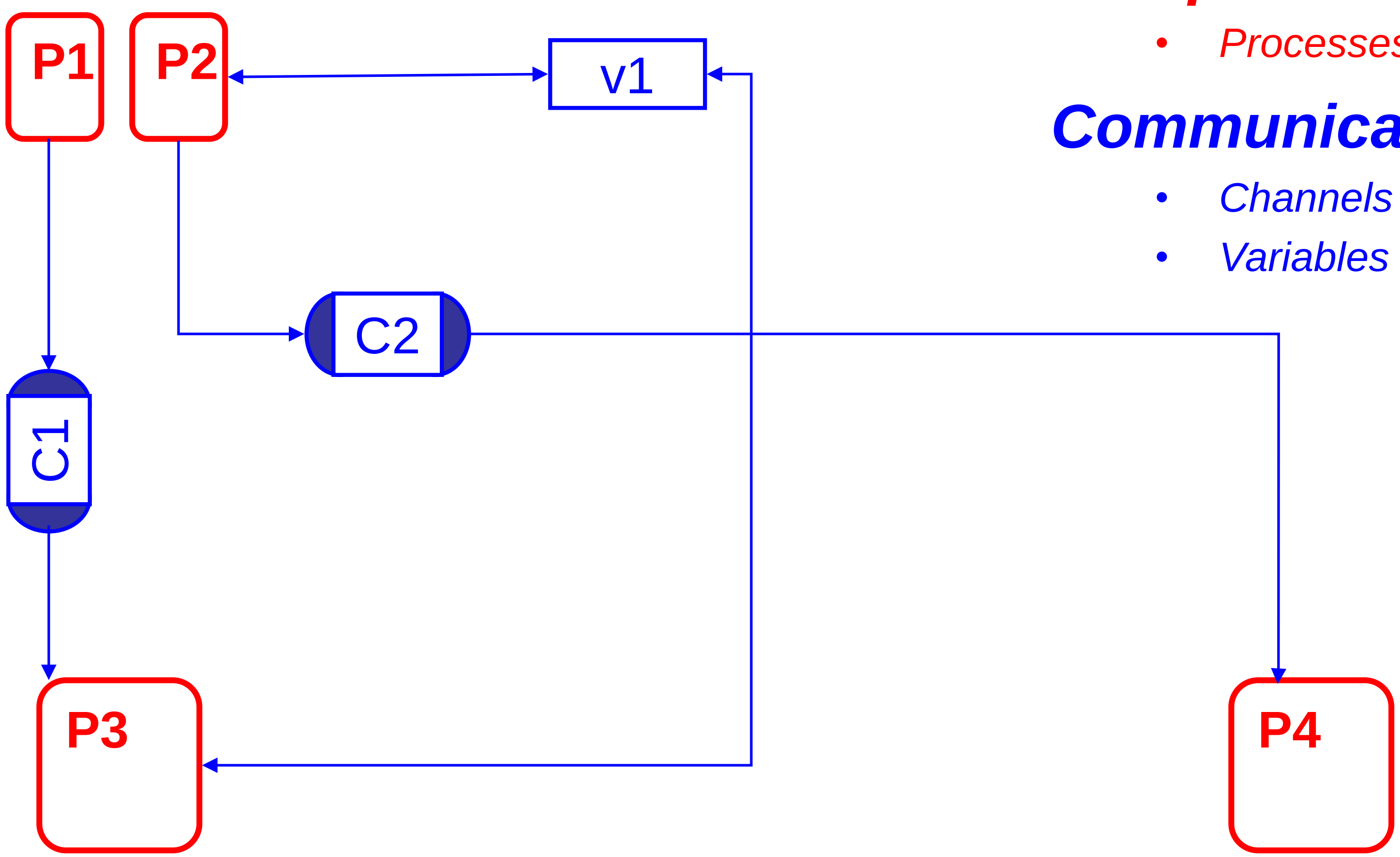
Design Methodology



X-Chart


Application
Behavior
Constraints
Platform
Synthesis
Decision making
Refinement
Structure
Quality numbers
Transaction Level Model
Latency, Area, Throughput, etc

Input: Application Behavior Specification



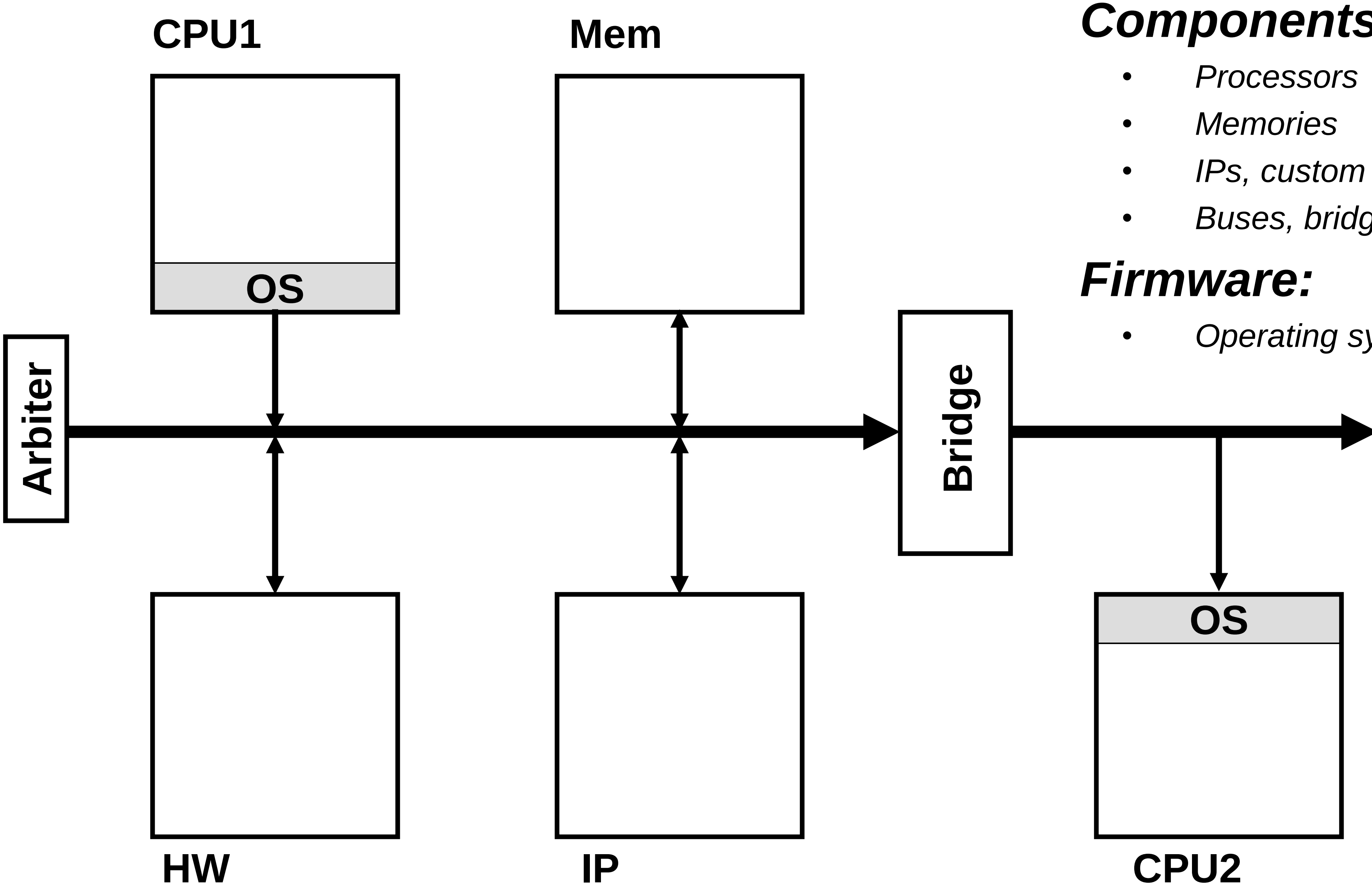
Computation

- *Processes (in C)*

Communication

- *Channels (in C)*
- *Variables (in C)*

Input: Platform Architecture Template



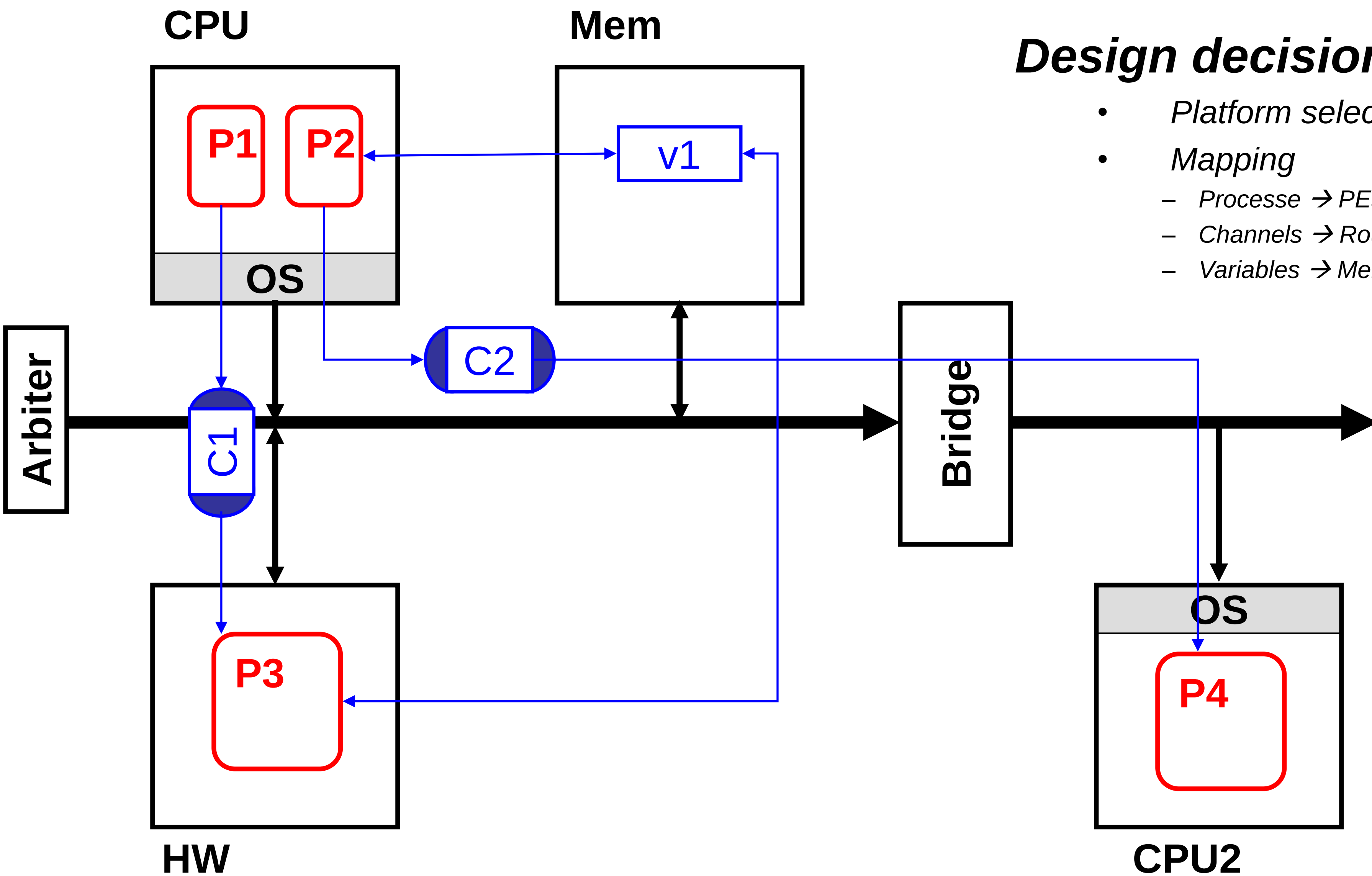
Components:

- *Processors*
- *Memories*
- *IPs, custom HW*
- *Buses, bridges*

Firmware:

- *Operating system*

System Definition

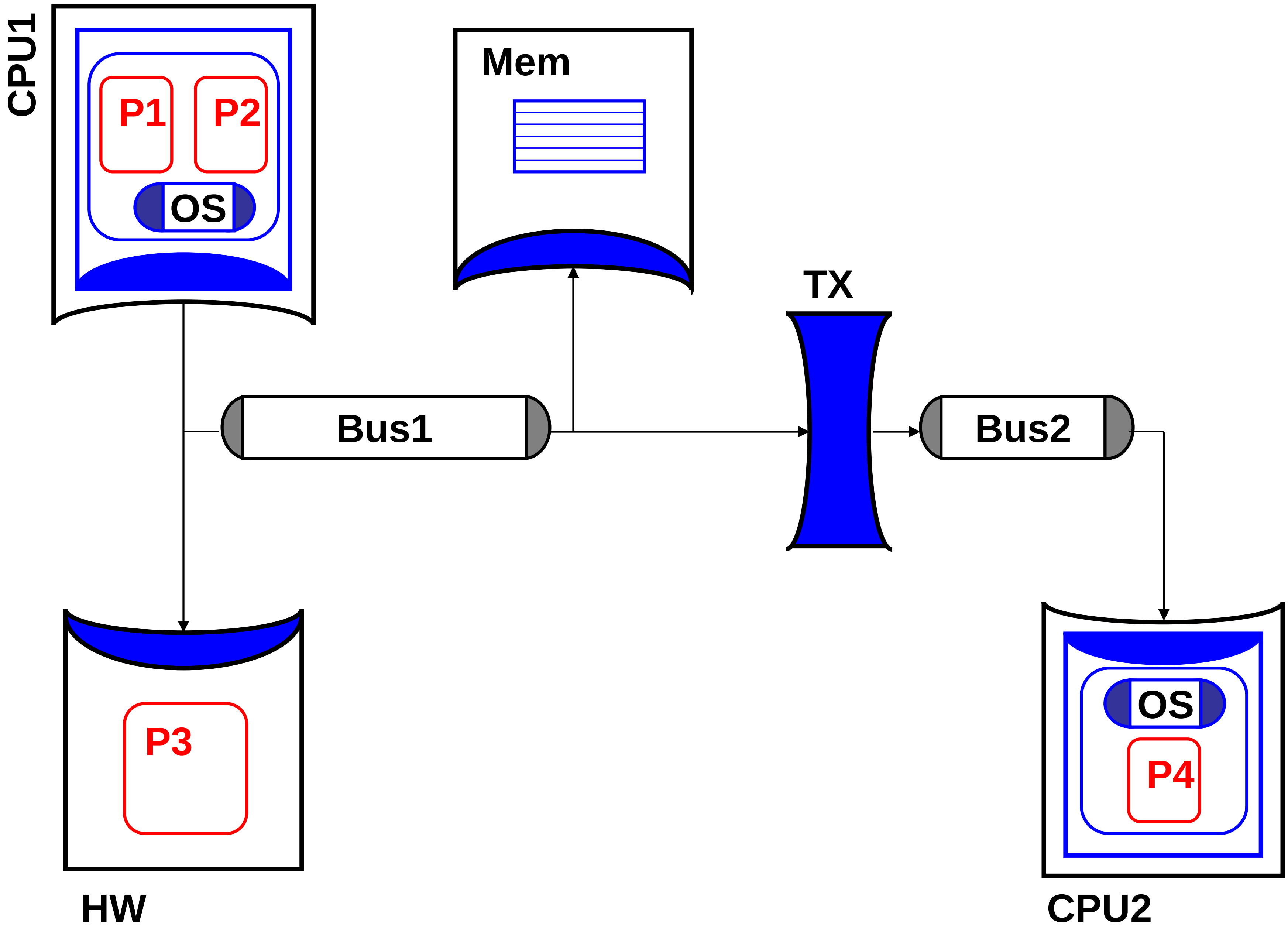


Design decisions:

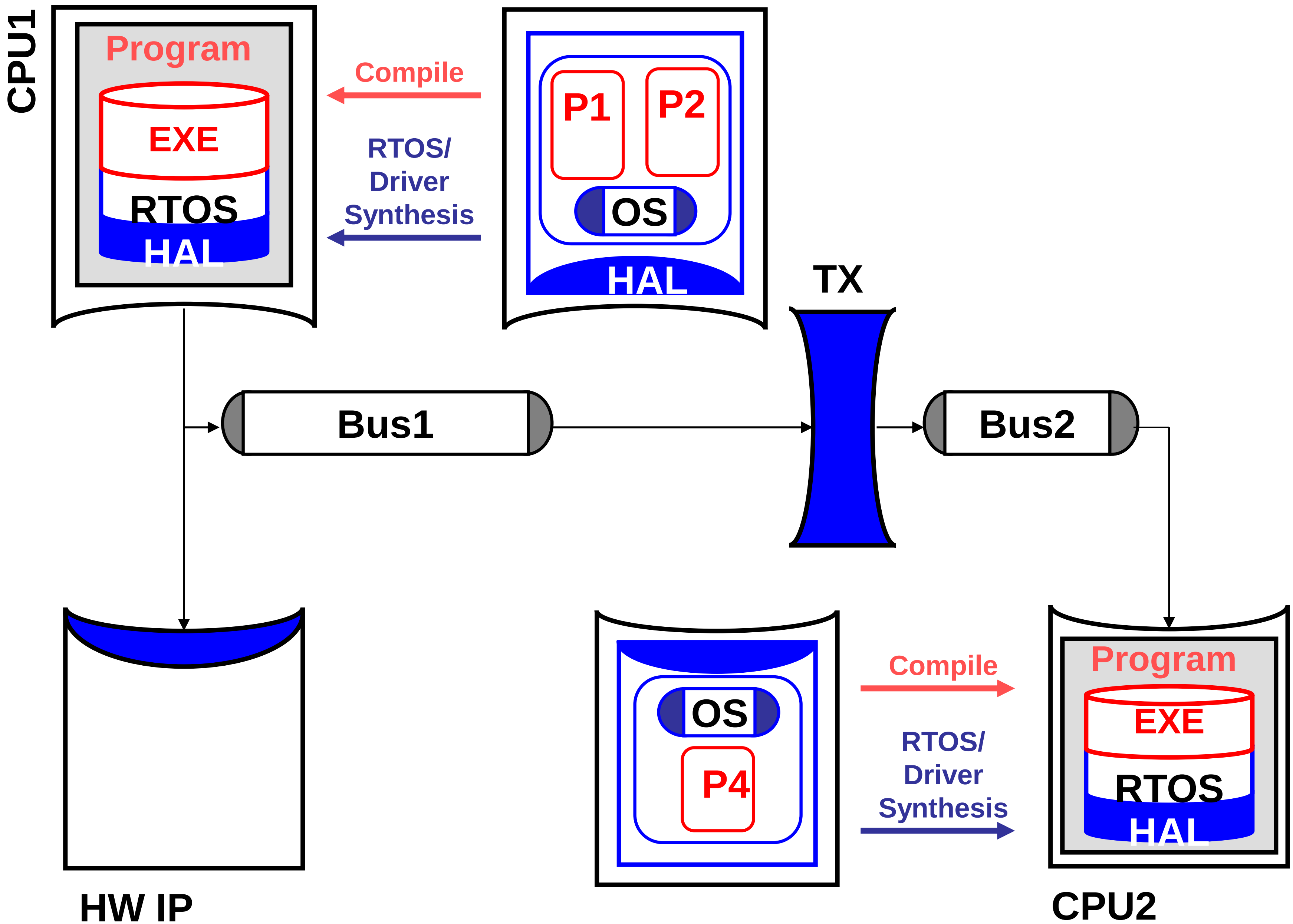
- *Platform selection*
- *Mapping*
  - *Processe → PEs*
  - *Channels → Routes*
  - *Variables → Memories*

System Definition = Application + Platform + Mapping

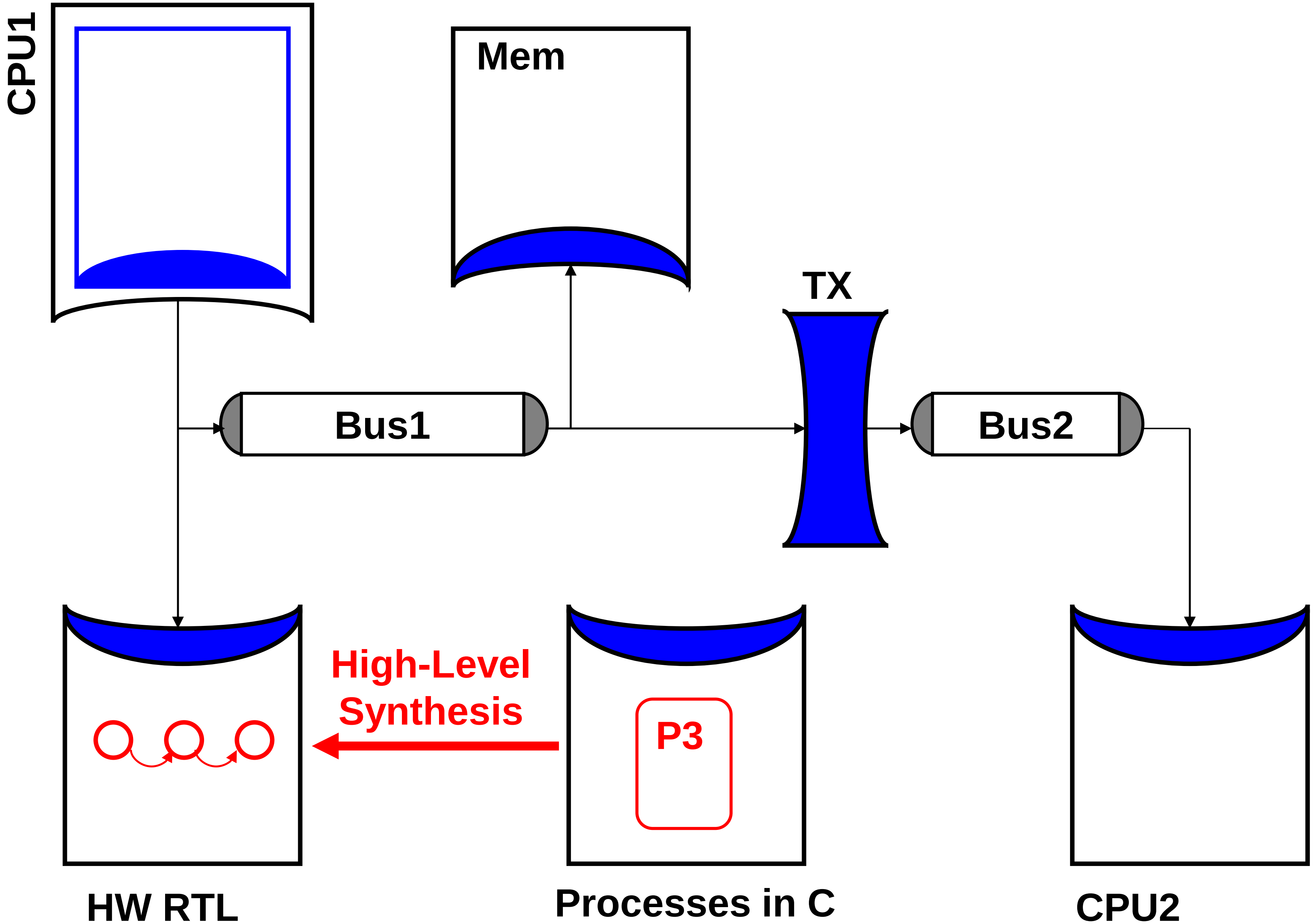
Output: Refined TLM



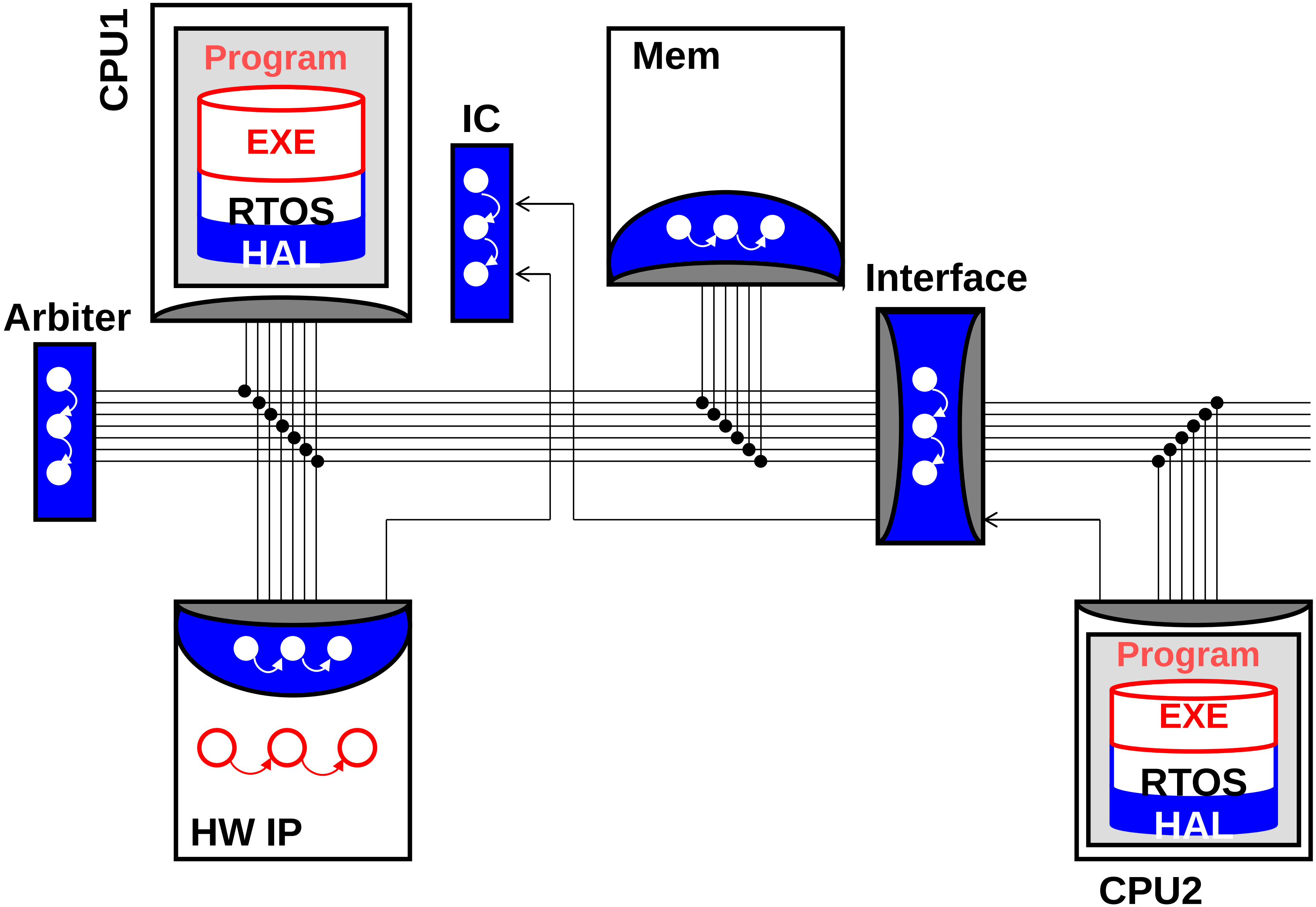
Software Synthesis



Hardware Synthesis



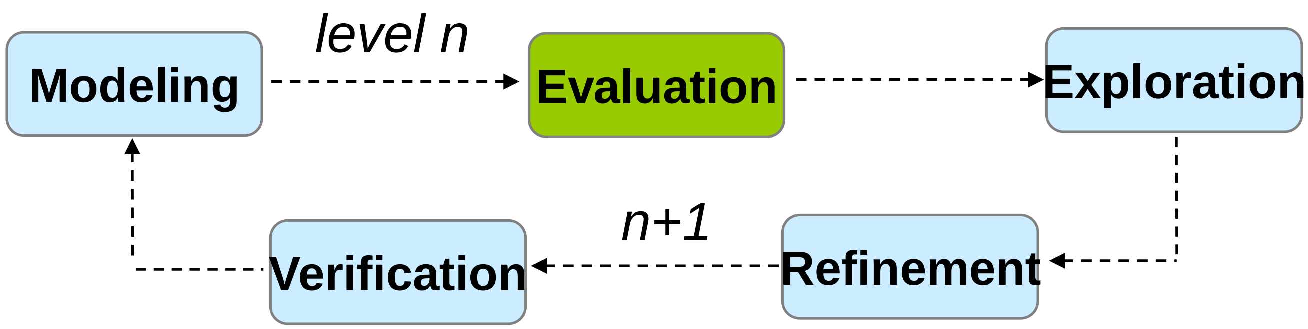
Cycle-Accurate Model



Lecture 6: Outline

- ✓ **System synthesis**
  - ✓ Synthesis process
- **Evaluation**
  - Profiling and simulation
  - Component estimation
  - Analytical and combined methods
- **Design space exploration**
  - Platform-based synthesis
  - Multi-objective optimization

System Quality Evaluation

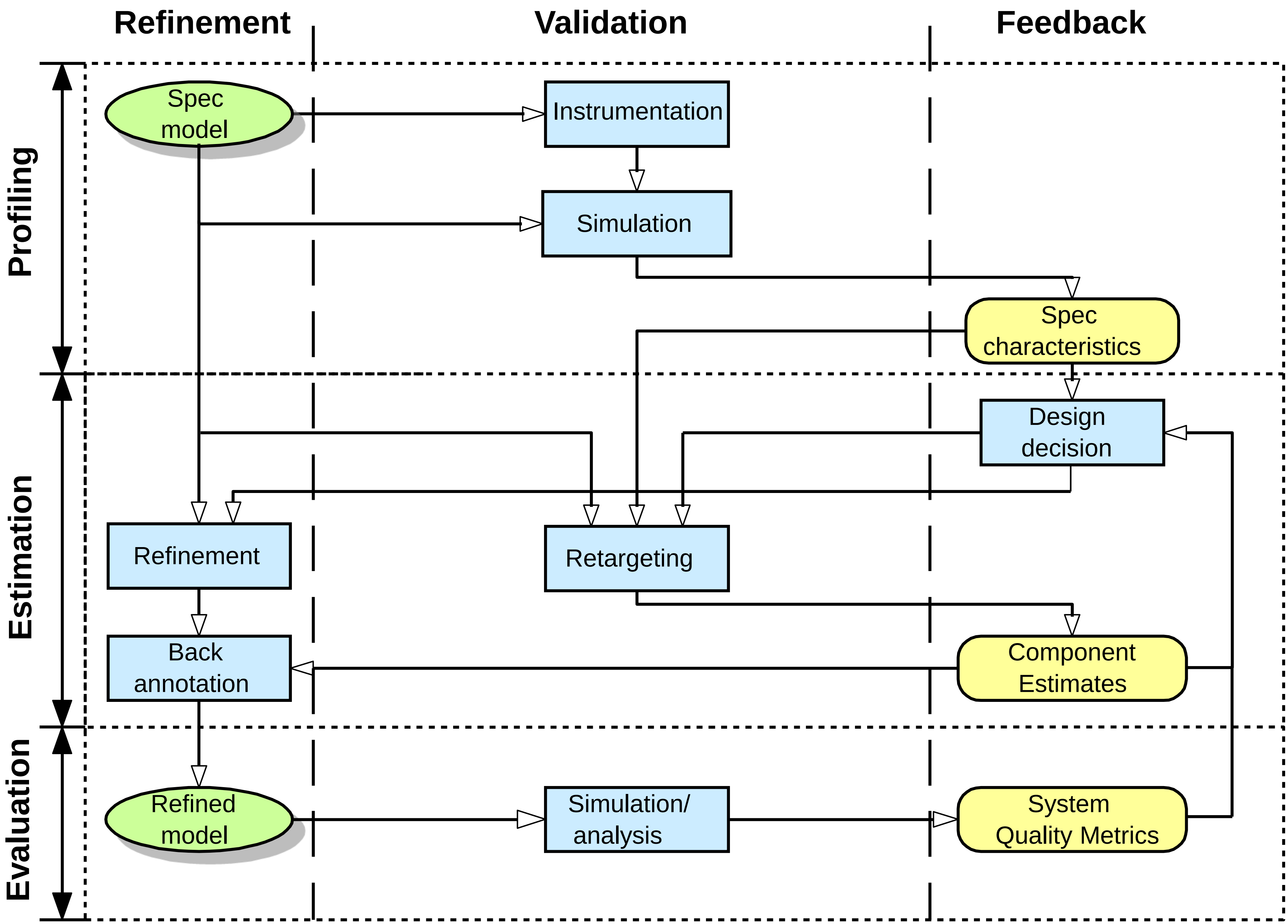


- **Runtime vs. accuracy**
  - Fast design space exploration
  - Fidelity: relative accuracy (vs. absolute accuracy)
- **Capabilities**
  - Various levels of abstraction: components, system
  - Wide range of metrics: power, timing, area, reliability
  - Wide variety of target implementations

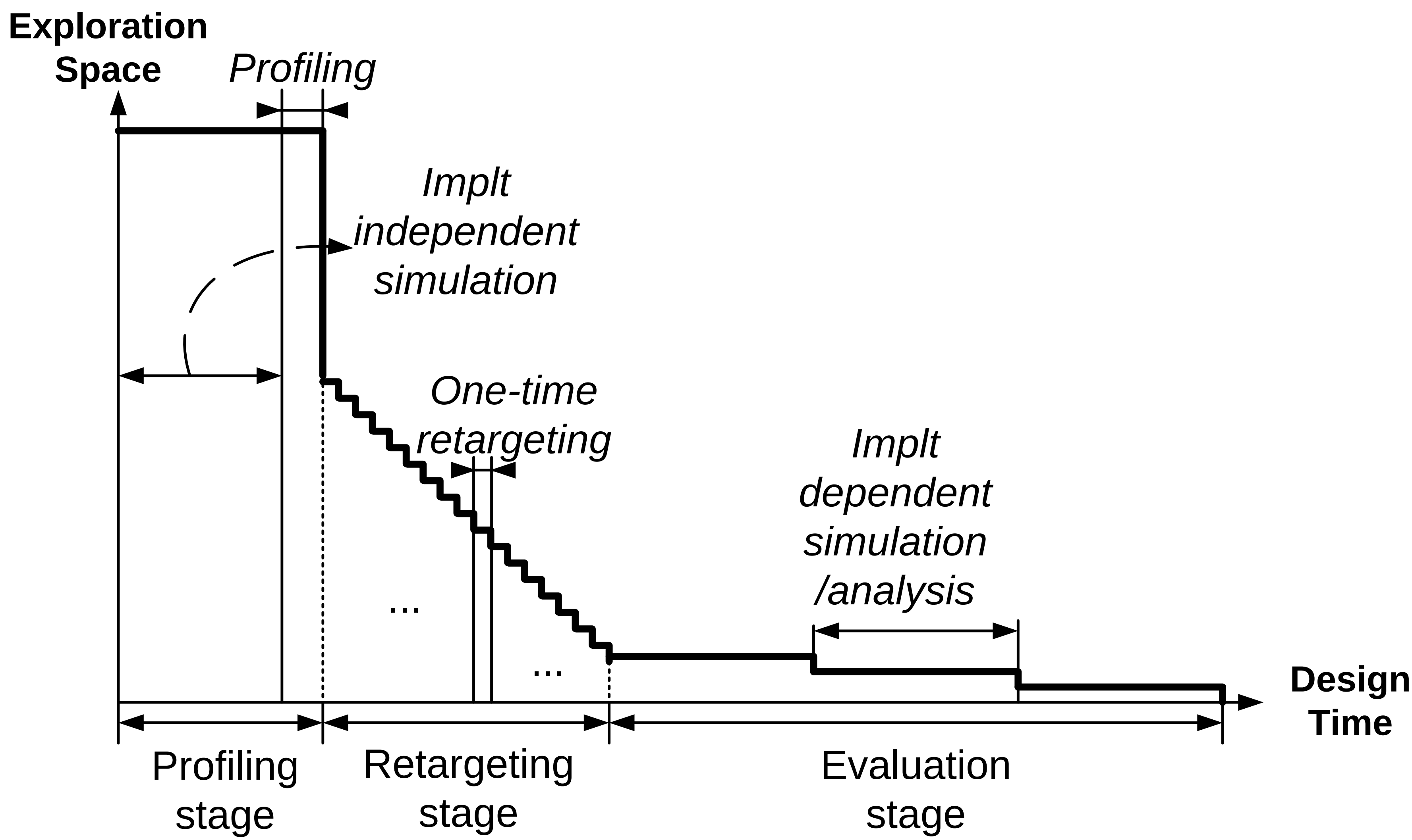
Evaluation Approaches

- **Dynamic simulation**
  - Profiling, instruction-set simulation (ISS)
  - ➢ *Long simulation times, corner cases*
  - ➢ *Target vs. host machine-dependent characteristics*
  - ➢ *Limited metrics (performance, operations)*
- **Static analysis**
  - Worst-case execution time (WCET), memory footprint, etc.
  - System cost functions, schedulability & real-time analysis
  - ➢ *Inaccurate bounds, manual interference (false paths)*
- **Combinations**
  - Host-compiled, back-annotated simulation
  - Trace-driven simulation
  - ➢ *Tradeoff between accuracy and speed*

Exploration Flow



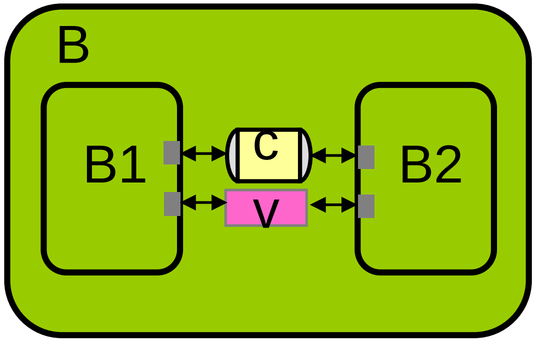
Design Space Exploration



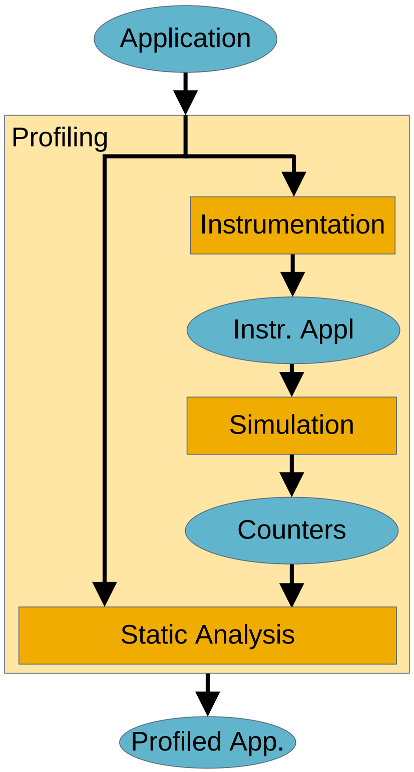
- **Explore and trim**

Profiling

- **Input specification MoC**
  - Hierarchy
  - Computation & communication



- **Multi-dimensional analysis**
  - Multi-entities
    - Behavior, channel, port, variable
  - Multi-metrics
    - Operation, traffic, storage
    - Static, dynamic
  - Multi-levels
    - Application, transaction, bus-functional



Profiling

- **Instrumentation-based profiling**
  - B_b: The execution counts of basic block b
    - Enumerate execution paths
  - $C_{b,i,d}$: No. of computed characteristics for item type i and data type d in the block b
  - Data type d: `float`, `int`, …
  - Item type i: metric-dependent

```
int b,c;
if( a = 0){
    b++;
}
else{
    b++;
    c++;
}
```

$B_1 = 1$

$C_{1,++,int} = 1$

$B_3 = 3$

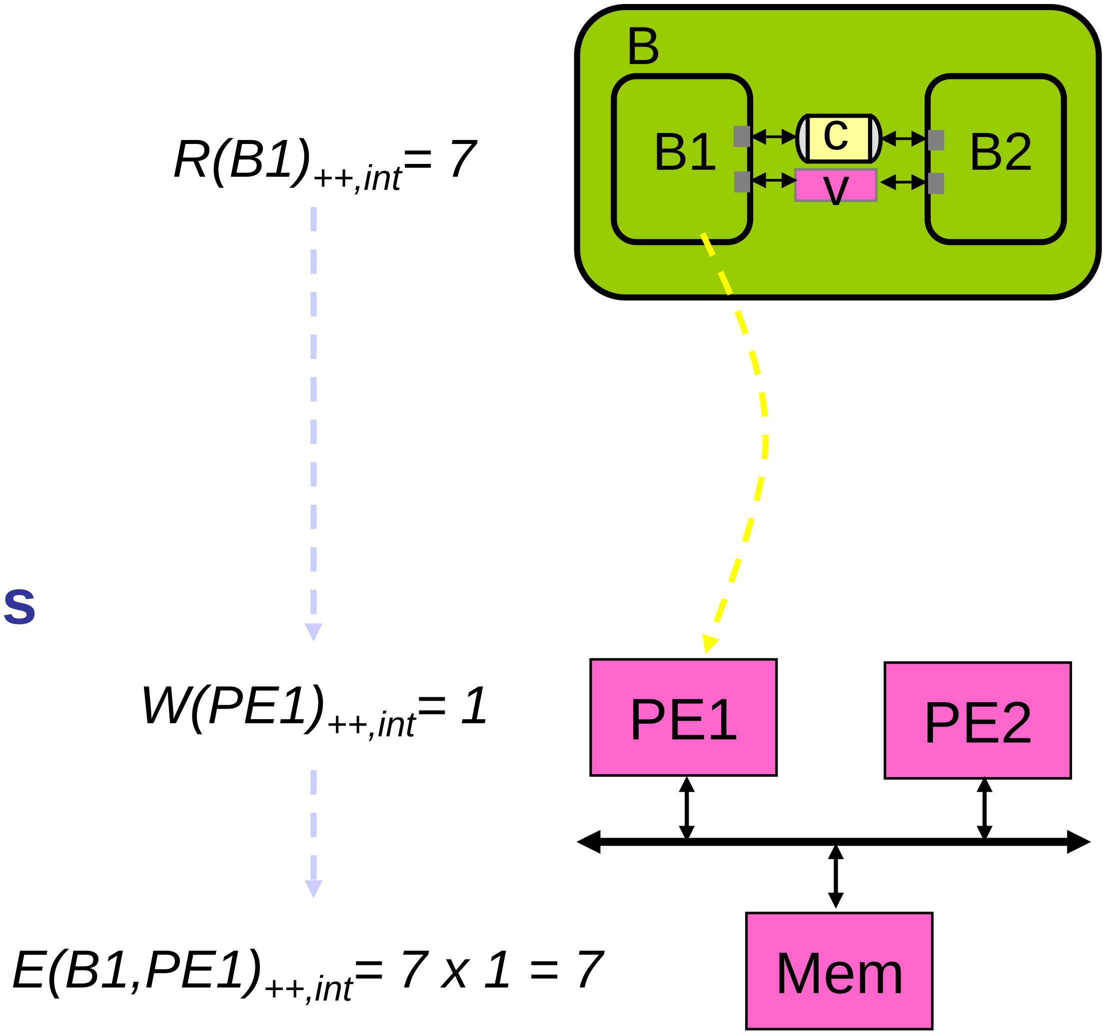
$C_{3,++,int} = 2$

- **Specification metrics**
  - $R_{i,d} = \Sigma_b C_{b,i,d}\, B_b$
  - $R = \Sigma_i \Sigma_d R_{i,d}$

$$R_{++,int} = \Sigma_i \,[\, B_i * C_{i,++,int} \,]$$
$$= 1 * 1 + 3 * 2$$
$$= 7$$

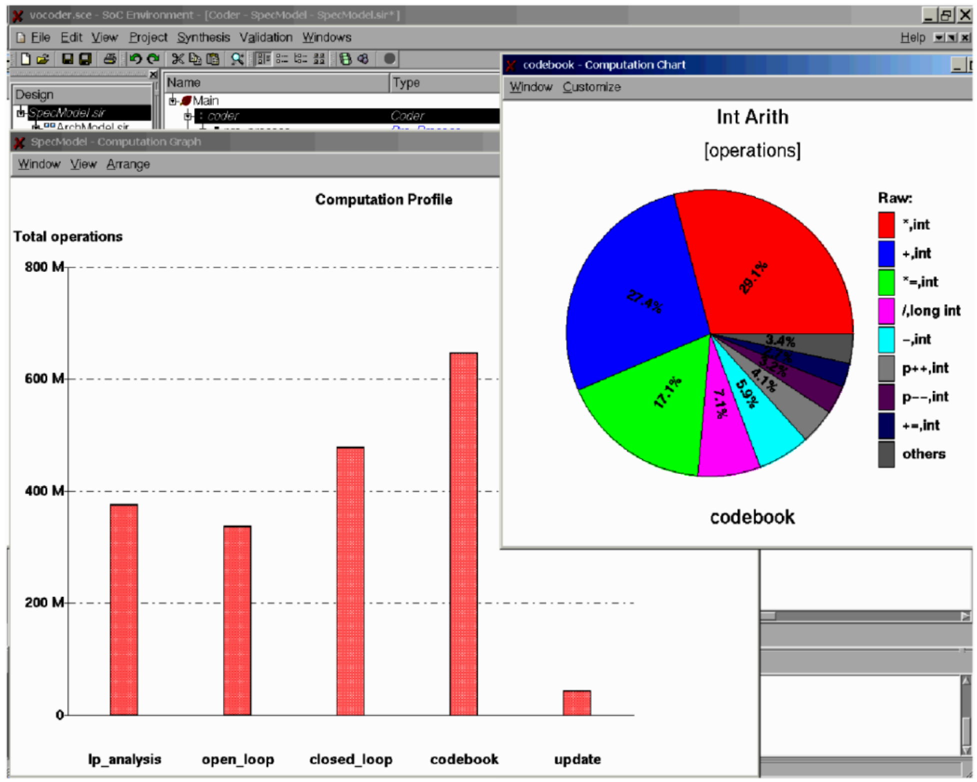
Retargeting

- **Target machine model**
  - $W_{i,d}$: weights of components which the entity mapped to
    - – Manual
    - – Simulation
    - – Complex cost function/ algorithm



➢ **Implementation estimates**
  - ➢ $E = \Sigma_i \Sigma_d (R_{i,d} * W_{i,d})$
  - ➢ Time complexity: $O(n)$

Vocoder Profiling



Floating –point not required
Dedicated hardware multipliers

HW acceleration

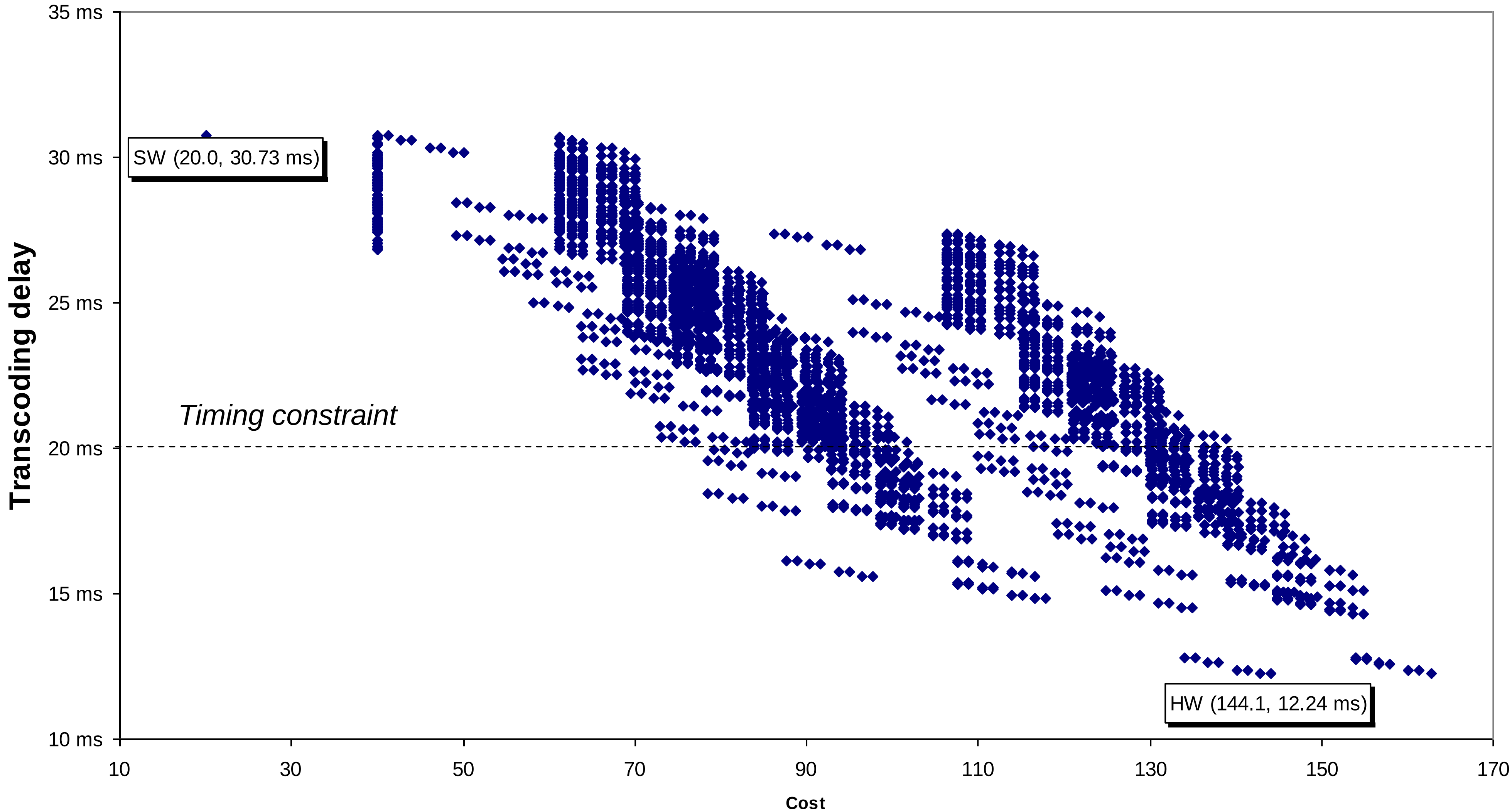
Computational complexity of top-level Vocoder behaviors:

LP_Analysis	Open_Loop	Closed_Loop	Codebook	Update
377.0 MOp	337.1MOp	478.7 MOp	646.4 MOp	43.6 MOp

Codebook operation mix:

(x, int)	(+, int)	(-, int)	(/,int)	(others,int)
46.2%	33.5%	9.1%	7.1%	4.1%

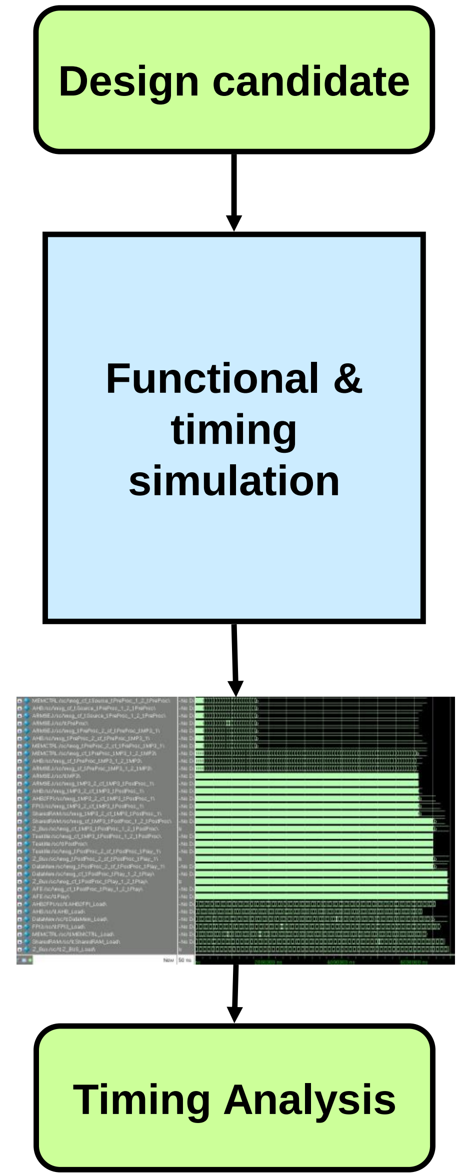
Vocoder Design Space Exploration



- Mapping of 8 top-level encoder behaviors onto ColdFire + DSP + HW
- 85:04h for 6561 alternatives (1.7s simulation + 3s refinement each)

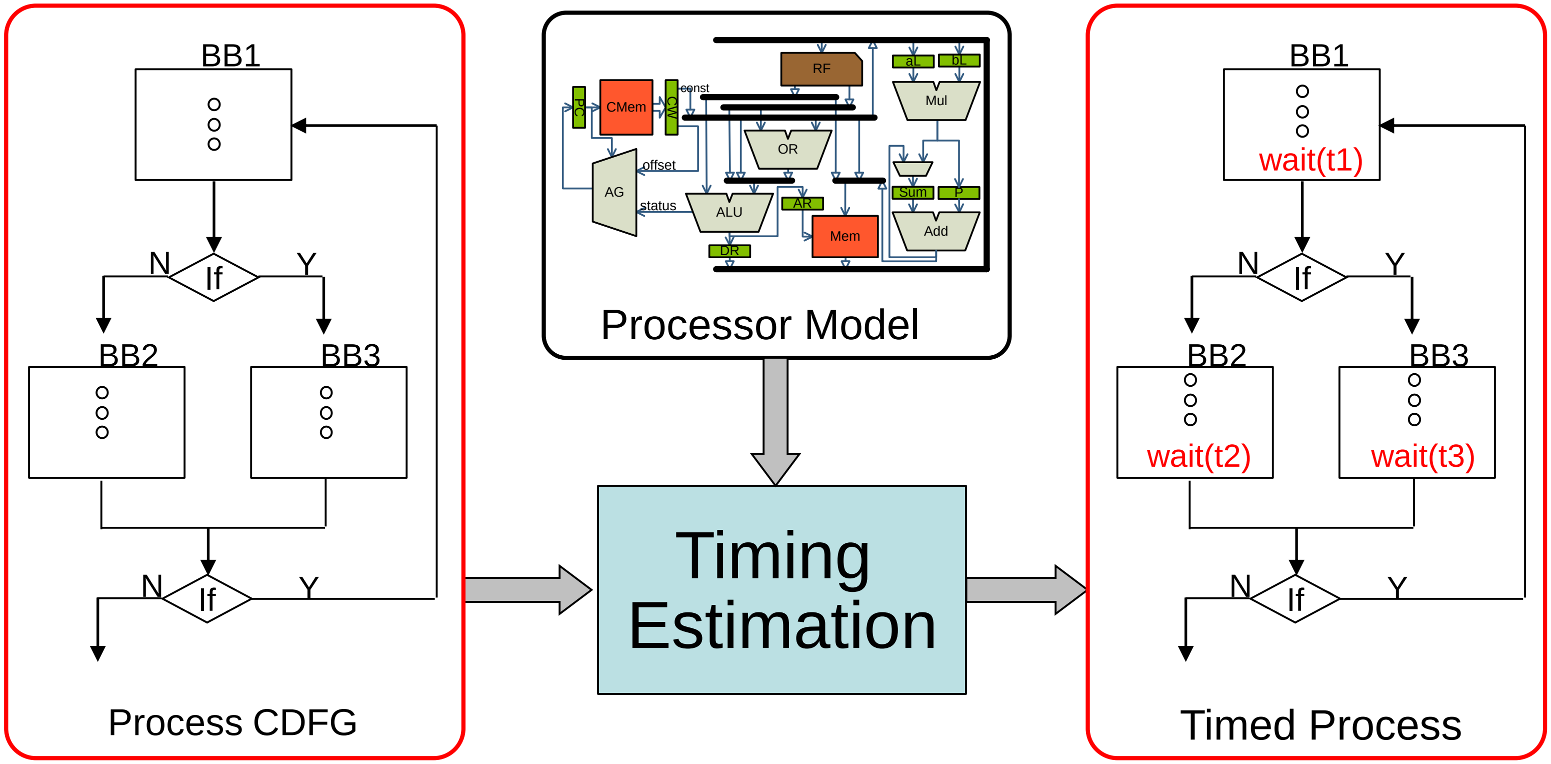
Simulation-Based Evaluation

- **Timing back-annotation**
  - Source level
  - Instructions, basic blocks or functions
  - Estimation of basic metrics
- **System simulation**
  - System description language
  - Simulation host
  - Functionality & timing
  - Generate trace
- **Timing analysis**
  - Latency, throughput, response time, etc.



Source: C. Haubelt, J. Teich

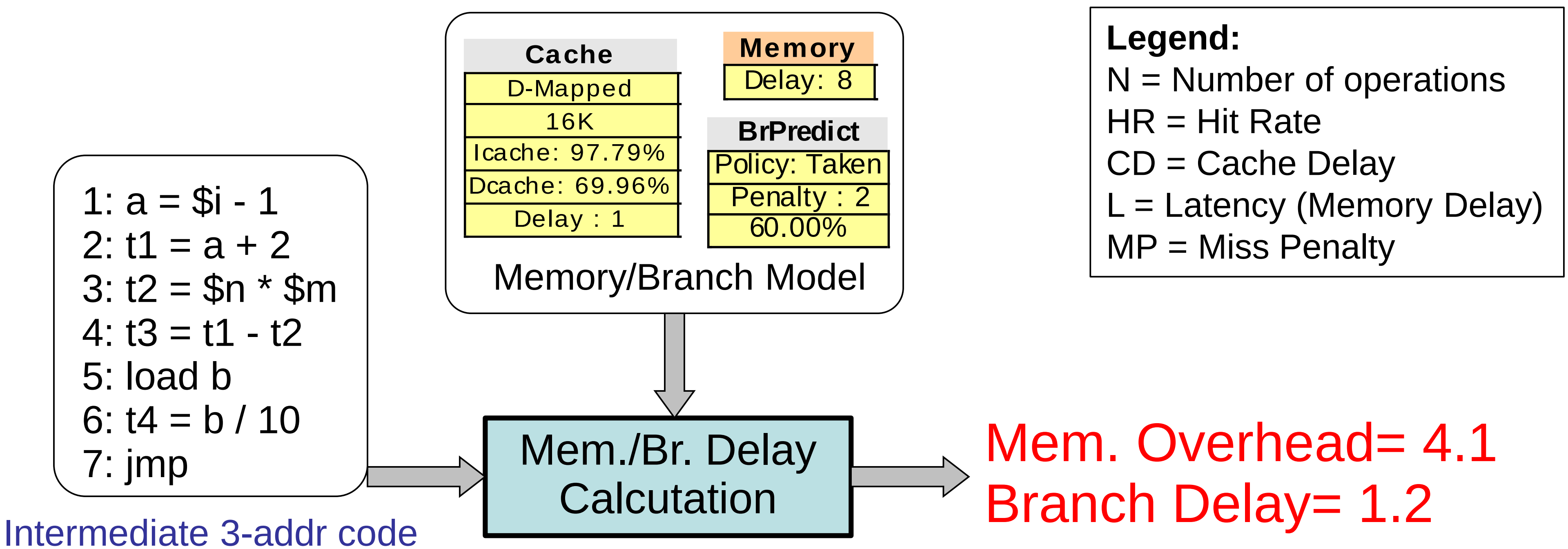
Computation Estimation



- **Execution time estimation**
  - Compiler frontend optimizations to produce CDFG
  - Processor microarchitecture model
  - DFG scheduling to compute basic block delay
  - RTOS model added for PEs with multiple processes

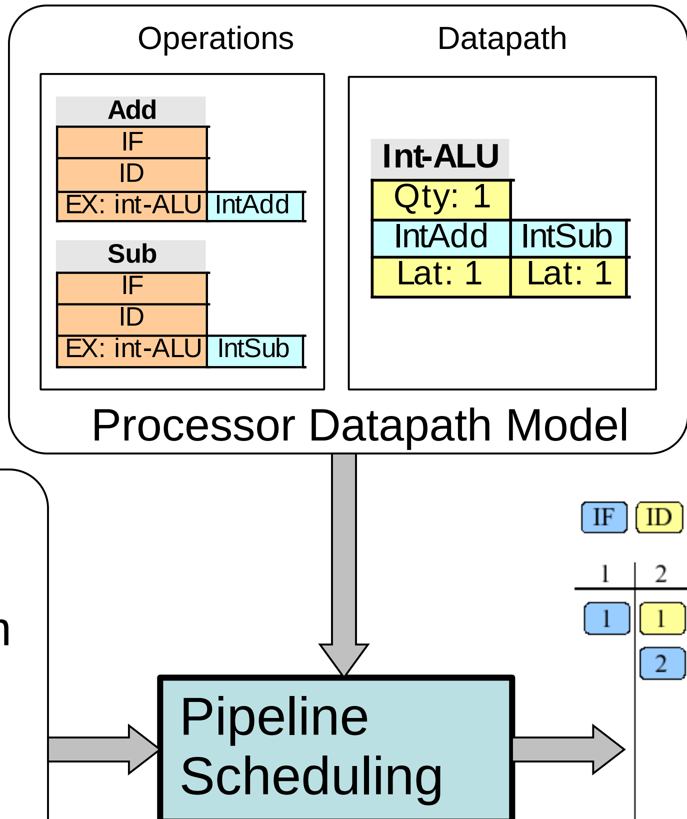
Stochastic Memory Delay Model

- **Assumption**
  - Cache and branch prediction hit rate available in data model
- **Delay Estimation**
  - Operation access overhead = N_{op} * $((1.0 – HR_i) * (CD + L_{mem}))$
  - Data access overhead = N_{ld} * $((1.0 – HR_d) * (CD + L_{mem}))$
  - Branch prediction miss penalty = MP_{rate} * Penalty



Processor Timing Estimation

- **Assumptions**
  - In-order, single issue processor
  - Optimistic during scheduling (100% cache hit)

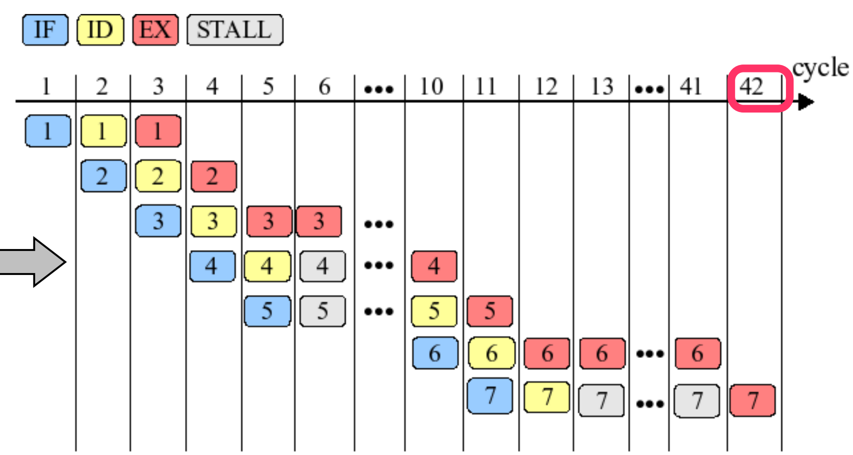


Total BB delay=
Op.+Mem.+Br. =
47.3 cycles

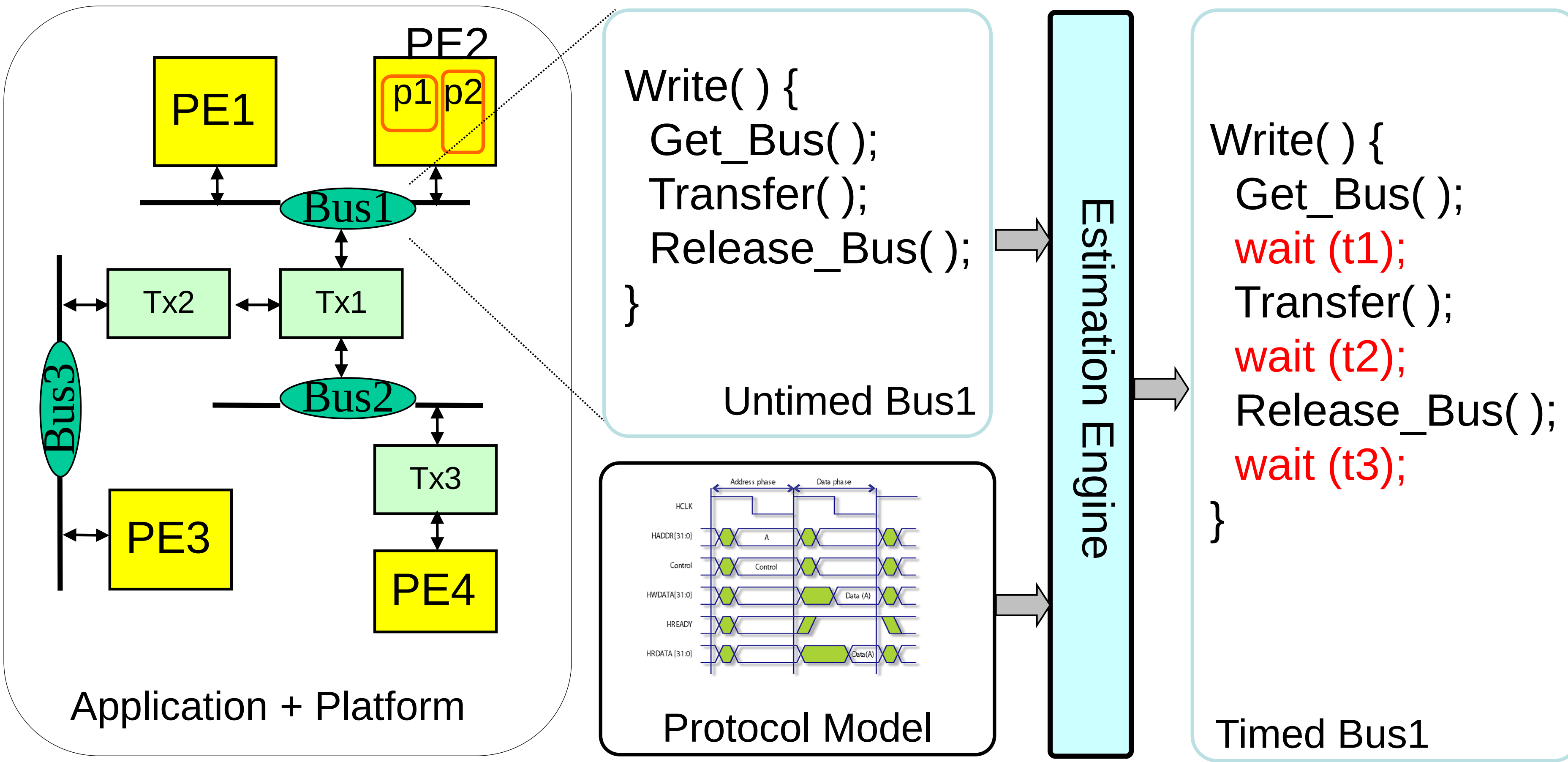
1: a = $i - 1
2: t1 = a + 2
3: t2 = $n * $m
4: t3 = t1 - t2
5: load b
6: t4 = b / 10
7: jmp
8: wait 47*CT

Intermediate 3-addr code

Operation delay= 42



Communication Estimation



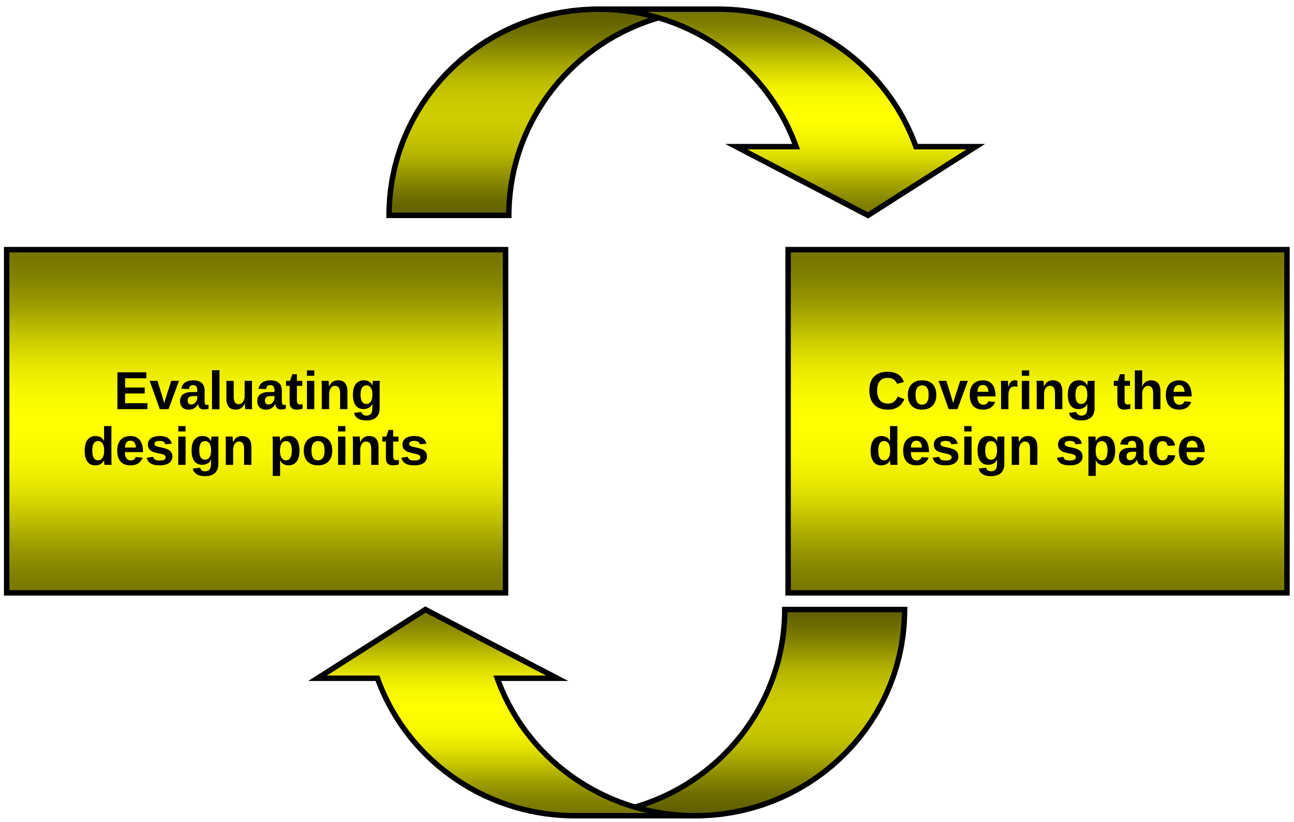
- **Communication latencies**
  - Protocol model used to estimate delays
  - Timing is annotated in bus channel

Lecture 6: Outline

- ✓ **System synthesis**
  - ✓ Synthesis process
- ✓ **Evaluation**
  - ✓ Profiling and simulation
  - ✓ Component estimation
  - ✓ Analytical and combined methods
- **Design space exploration**
  - Platform-based synthesis
  - Multi-objective optimization

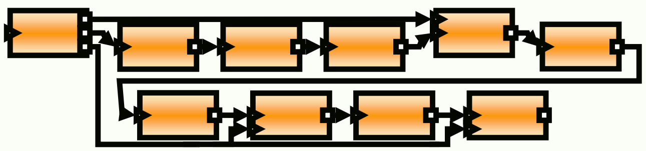
Design Space Exploration

- **Design Space Exploration is an iterative process:**
  - How can a single design point be evaluated?
  - How can the design space be covered during the exploration process

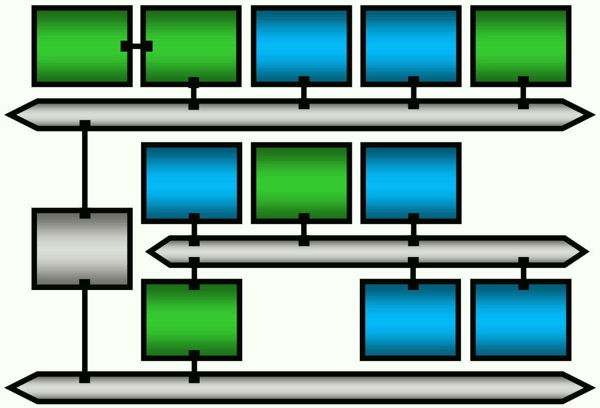


Source: C. Haubelt, J. Teich, Univ. of Erlangen-Nuremberg

Platform-Based Synthesis

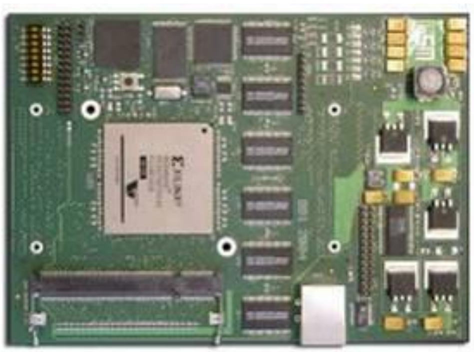


Application



Platform

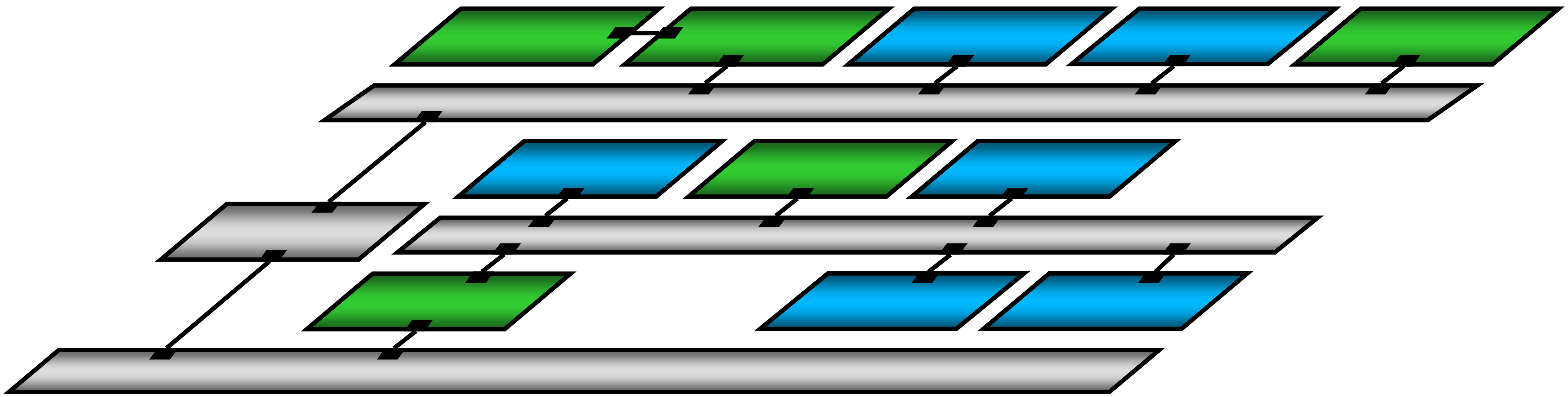
Automatic Optimal Mapping ?



Source: C. Haubelt, J. Teich

Resource Allocation

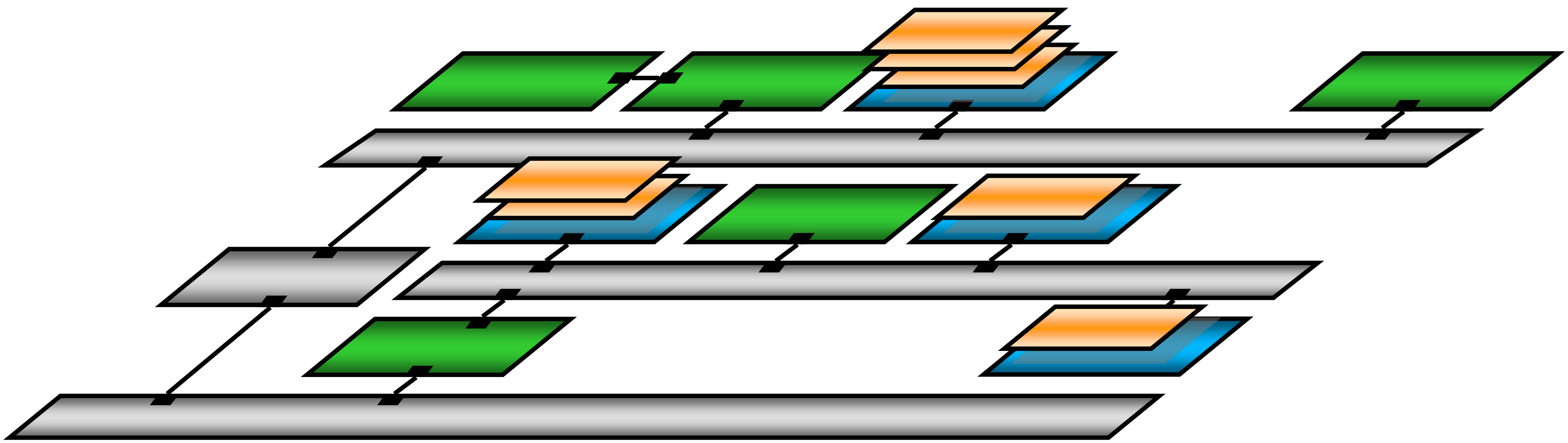
- **Resource allocation, i.e., select resources from a platform for implementing the application**



Source: C. Haubelt, J. Teich

Process Binding

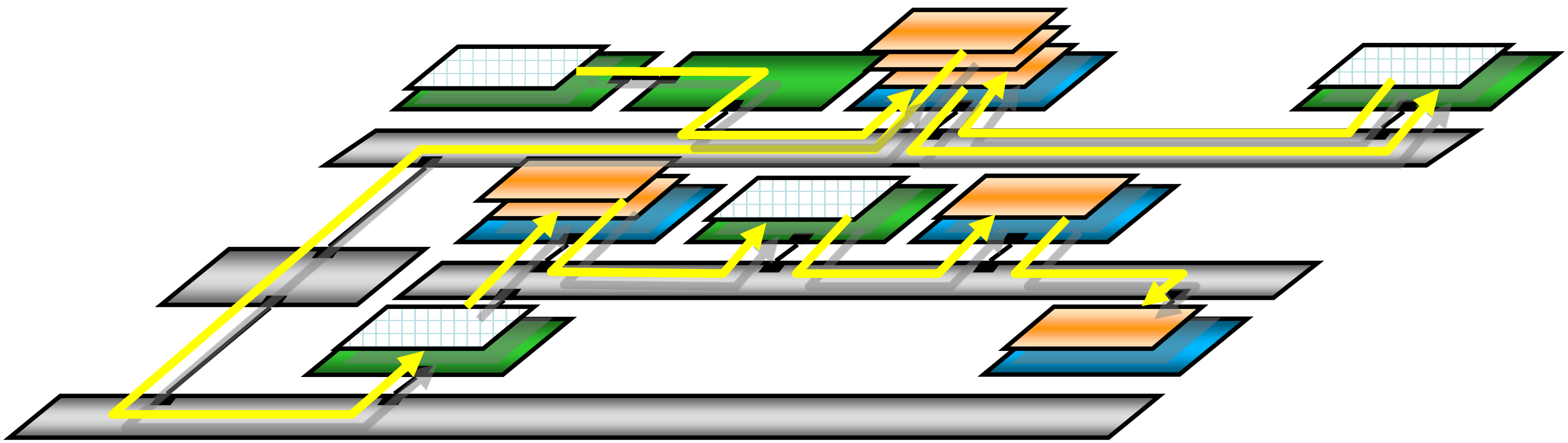
- **Process mapping, i.e., bind processes onto allocated computational resources**



Source: C. Haubelt, J. Teich

Channel Routing

- **Channel mapping, i.e., assign channels to paths over busses and address spaces**



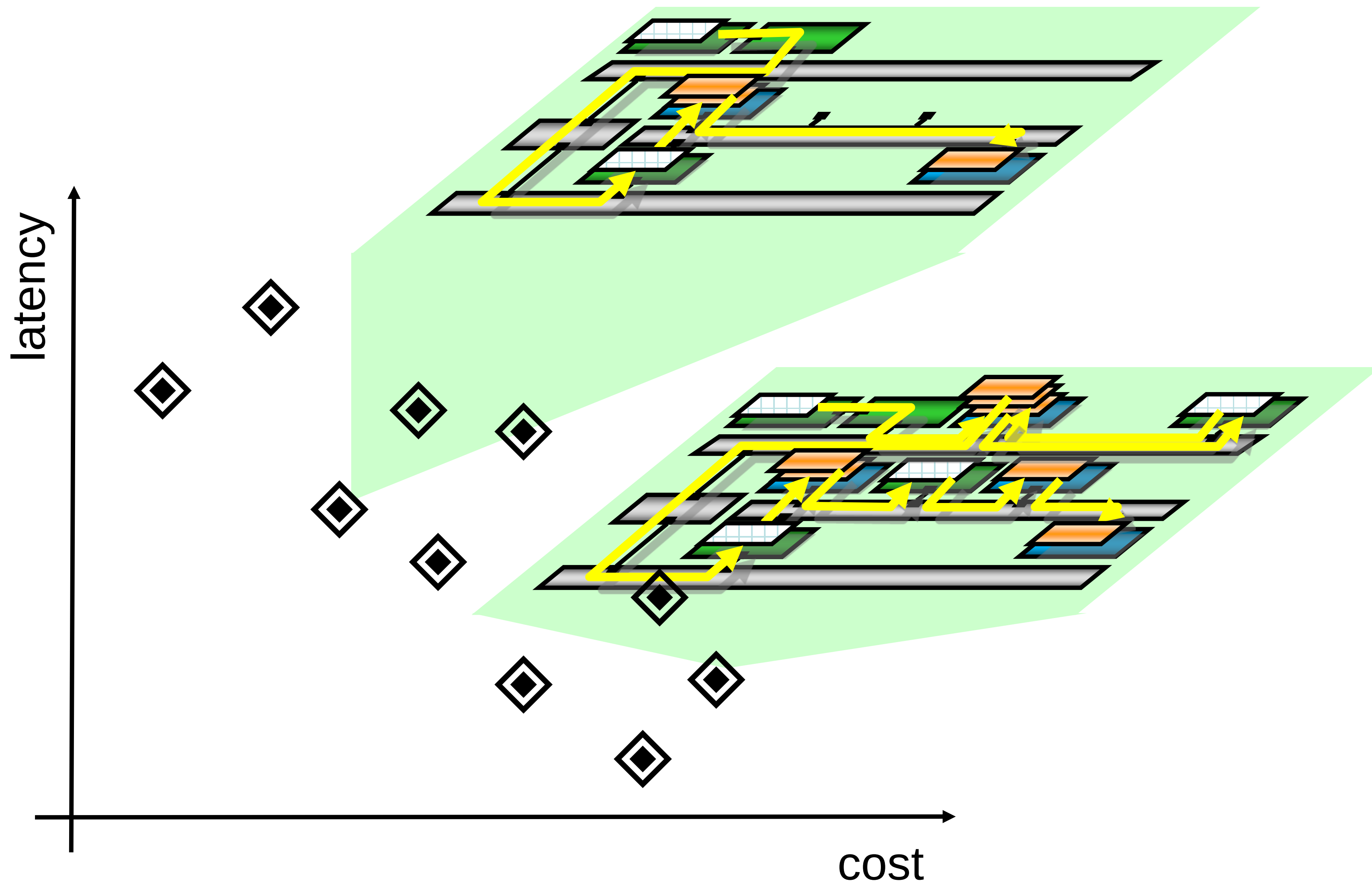
Source: C. Haubelt, J. Teich

Multi-Objective Optimization

- In general, several solutions (implementations) exist with different properties, e.g., area and power consumption, throughput, etc.
- Implementations are often optimized with respect to many (conflicting) objectives
- Finding best implementations is task of Multi-Objective Optimization

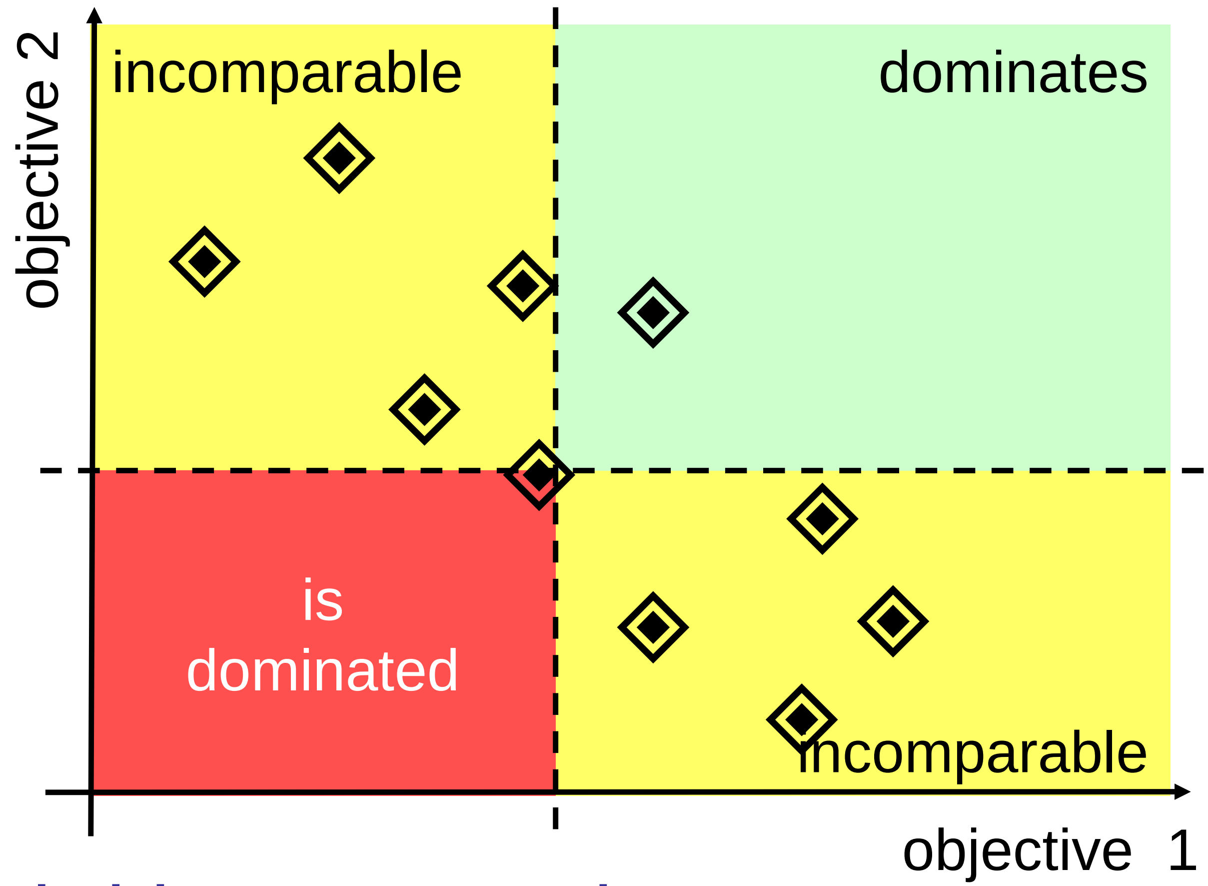
Source: C. Haubelt, J. Teich

Objective Space



Source: C. Haubelt, J. Teich

Pareto Dominance

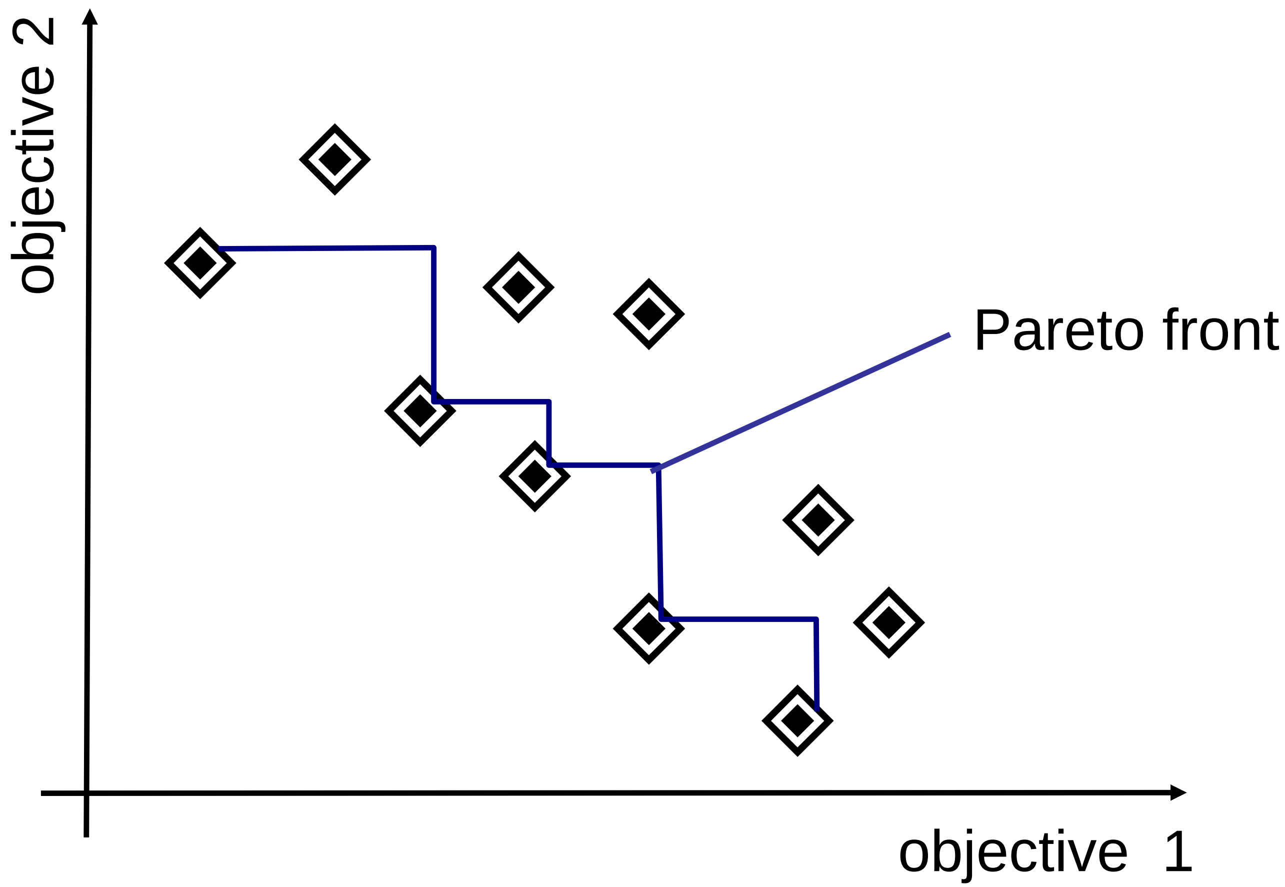


- **Given: two decision vectors x_1 and x_2**
  - $x_1 \succ\succ x_2$ (strongly dominates) if $\forall i: f_i(x_1) < f_i(x_2)$
  - $x_1 \succ x_2$ (dominates) if $\forall i: f_i(x_1) \leq f_i(x_2) \wedge \exists j: f_j(x_1) < f_j(x_2)$
  - $x_1 \sim x_2$ (indifferent) if $\forall i: f_i(x_1) = f_i(x_2)$
  - $x_1 || x_2$ (incomparable) if $\exists i,j: f_i(x_1) < f_i(x_2) \wedge f_j(x_2) < f_j(x_1)$

Source: C. Haubelt, J. Teich

Pareto Optimality

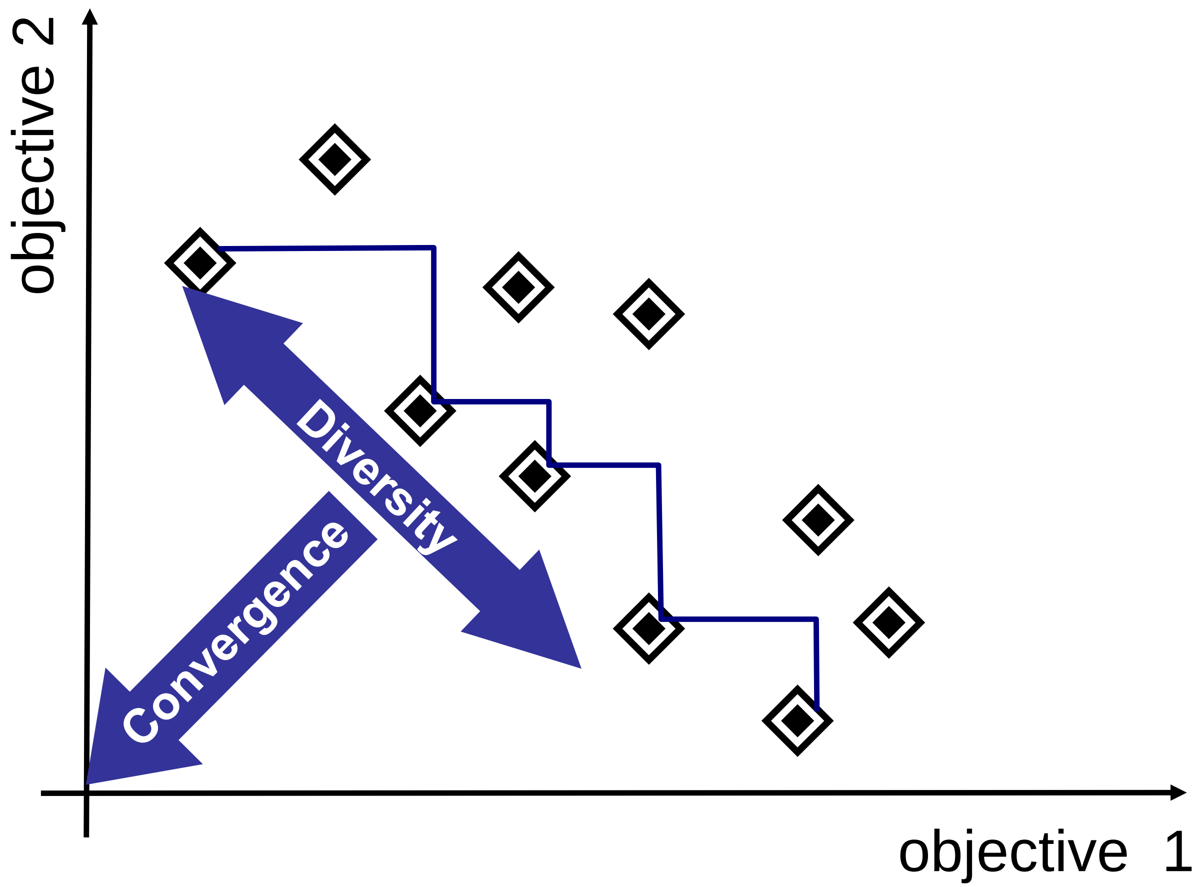
- **Set of all solutions *X***
- **A decision vector $x \in X$ is said to be *Pareto-optimal* if $\nexists y \in X: y \succ x$**



Source: C. Haubelt, J. Teich

Optimization Goals

- **Find Pareto-optimal solutions**
- **Or a good approximation (convergence, diversity)**
- **With a minimal number of iterations**



Source: C. Haubelt, J. Teich

Lecture 6: Summary

- **System synthesis**
  - Design space exploration (DSE)
    - – Decision making to cover design space
    - – Refinement to generate system model
    - – Evaluation for feedback about design quality
- **Evaluation**
  - Source-level profiling and simulation
  - Component-level estimation
  - System-level analysis, simulation or combination
- **Decision making**
  - Multi-objective optimization, Pareto optimality