

ECE382N.23: Embedded System Design and Modeling

Lecture 4 – System Modeling

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



The University of Texas at Austin
Chandra Department of Electrical
and Computer Engineering
Cockrell School of Engineering

Lecture 4: Outline

- **Evaluation and estimation**
 - Methods
- **Simulation**
 - Simulation methods
- **Analysis**
 - Component- and system-level estimation
- **Hybrid approaches**
 - Semi-analytical methods
 - Machine learning-based prediction methods

System Modeling

- System definition**
 - Platform netlist
 - Mapping
- System quality**
 - Performance, power, ...

The diagram illustrates the System Modeling process. It starts with a **Specification Model** (green box) containing **Behavior** and **Constraints**. This feeds into a **Synthesis** stage (blue box) which includes **Decision Making** and **Modeling**. The **Synthesis** stage then feeds into an **Implementation Model** (green box) containing **Structure** and **Quality**. A cloud labeled **Cost models** is shown connected to the Implementation Model. Below the Specification Model, there is a hardware architecture diagram showing components like P1-P2, V1, OS, C1-C2, and P3-P4, connected via a Bridge to a CPU2. Arrows indicate data flow between these components.

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 4 © 2024 A. Gerstlauer 3

Evaluation and Estimation Methods

The diagram compares Worst-Case and Best-Case evaluation methods. It shows a vertical axis with horizontal bars representing different methods. The bars are color-coded: blue for Real System, orange for Measurement, pink for Simulation, and green for (Formal) Analysis. The vertical axis has two points: "Worst-Case" at the top and "Best-Case" at the bottom. A vertical line connects the "Real System" bar to the "Worst-Case" point, and another vertical line connects the "Real System" bar to the "Best-Case" point.

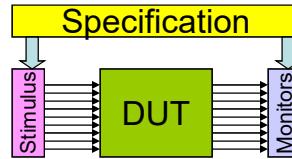
Source: L. Thiele

- Measurement**
 - Fast (real time), exhaustive?
 - Requires physical implementation
- Simulation**
 - Speed vs. accuracy tradeoffs
 - Quality of testbench, corner cases?
- Analysis**
 - Worst-case/best-case assumptions
 - Tightness of upper/lower bounds? Dynamic effects?

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 4 © 2024 A. Gerstlauer 4

Simulation Methods

- Create stimuli and simulate model



- Inputs

- Specification
 - Used to create interesting stimuli and monitors (golden output)
- Model of DUT
 - Typically written in HDL or C or both

- Output

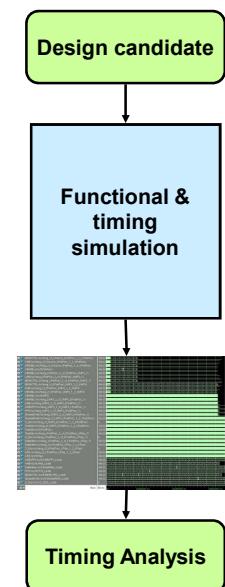
- Failed test vectors (validation)
- Quality metrics (evaluation)

- Speed vs. accuracy

- Fundamental tradeoff

Co-Simulation

- Component simulation models
 - Functional model
 - Timing, energy, ... models
- Co-simulation
 - System description language & model
 - Generates a trace
- Simulation trace analysis
 - Check functional results
 - Extract metrics from trace
 - Latency, throughput, etc.



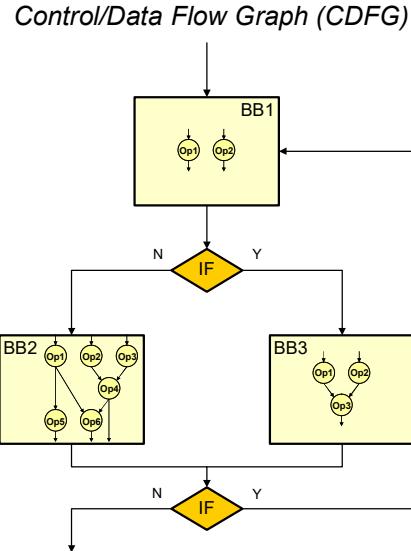
Source: C. Haubelt, J. Teich

Analysis Methods

- **Static analysis**
 - Symbolic, mathematical models for avg/best/worst case
 - Worst-case execution time analysis (WCET) of single task
 - Real-time scheduling of single processor
 - Best-case roofline models of system
- **Probabilistic analysis**
 - Statistical models, distributions for “average” case
 - Queuing theory for computer systems
- **Deterministic dynamic analysis**
 - Min-plus/max-plus algebra, upper/lower bounds over time
 - Network calculus, real-time calculus
 - Modular Performance Analysis (MPA) of parallel systems

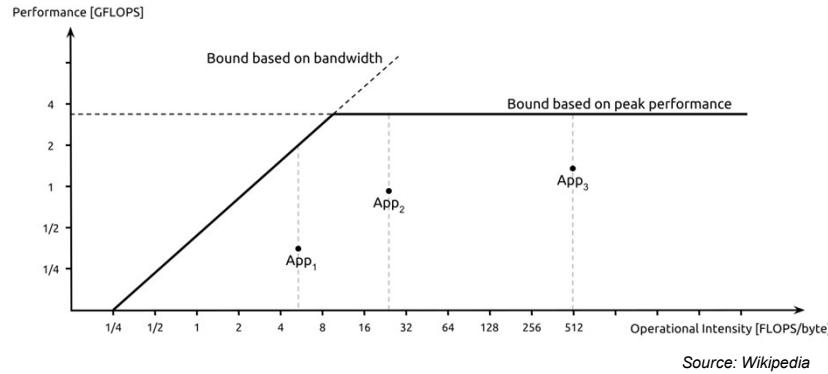
Worst-Case Analysis

- **Example: worst-case execution time (WCET) analysis of single task**
 - Micro-architecture analysis
 - Compute bounds for each basic execution block
 - Symbolically simulate statements on processor model (pipeline)
 - Conservative assumptions for dynamic effects (caches, predictors)
 - Path analysis
 - Enumerate possible paths and take maximum of block sequence
 - Possible paths often highly dynamic (loop bounds, false paths)
 - Basis for back-annotation or static system analysis
 - Combine static code analysis with dynamic system simulation
 - Static or dynamic model of inter-process cross-dependencies



Best-Case Analysis

- Example: roofline modeling
 - Bottleneck analysis
 - Application arithmetic intensity
 - Architecture peak bandwidth and operation throughput
 - Memory- vs. compute-bound regions



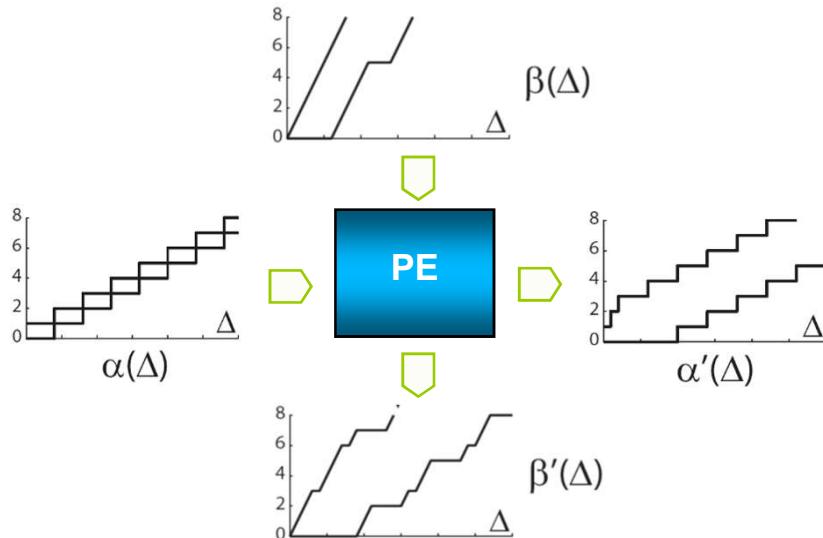
ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 4

© 2024 A. Gerstlauer

9

System-Level Dynamic Analysis

- Modular Performance Analysis (MPA)
 - Real-time calculus (RTC)

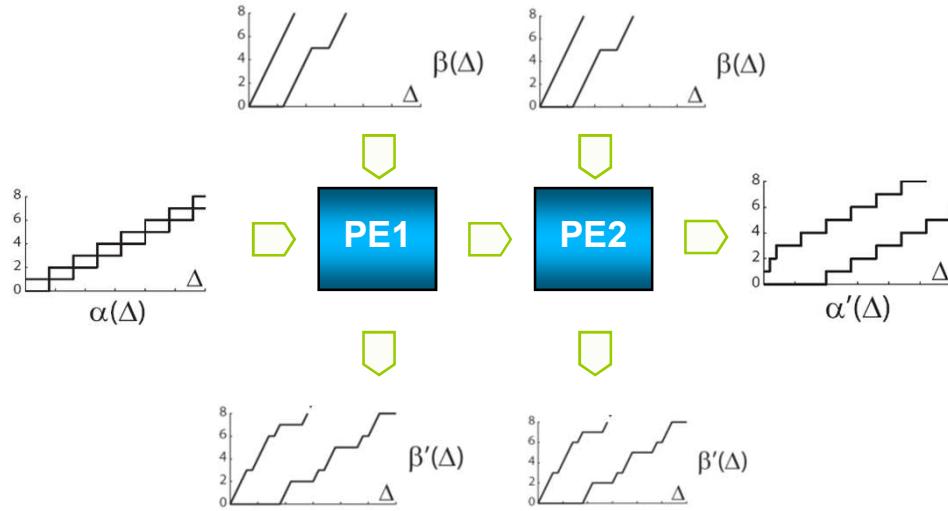


ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 4

© 2024 A. Gerstlauer

10

MPSOC Analysis with MPA



Source: C. Haubelt, J. Teich, DATE '09 Tutorial

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 4

© 2024 A. Gerstlauer

11

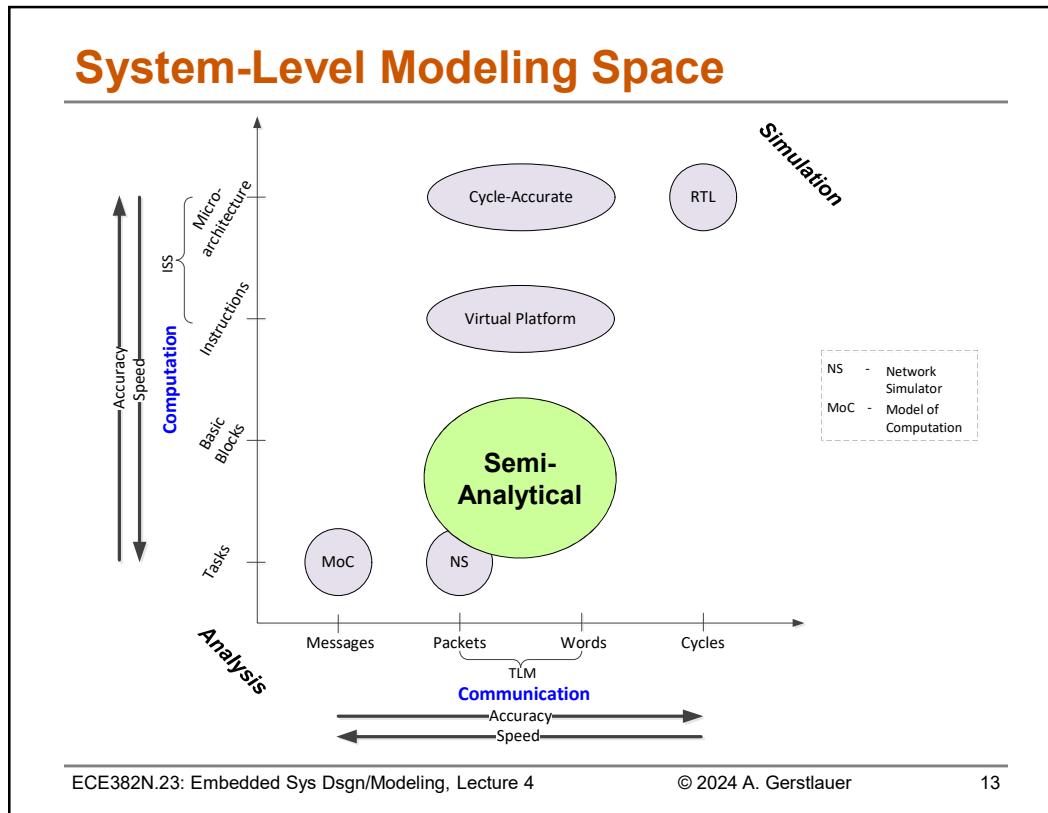
Lecture 4: Outline

- ✓ Evaluation and estimation
 - ✓ Methods
- ✓ Simulation
 - ✓ Simulation methods
- ✓ Analysis
 - ✓ Component- and system-level estimation
- Hybrid approaches
 - Semi-analytical methods
 - Machine learning-based methods

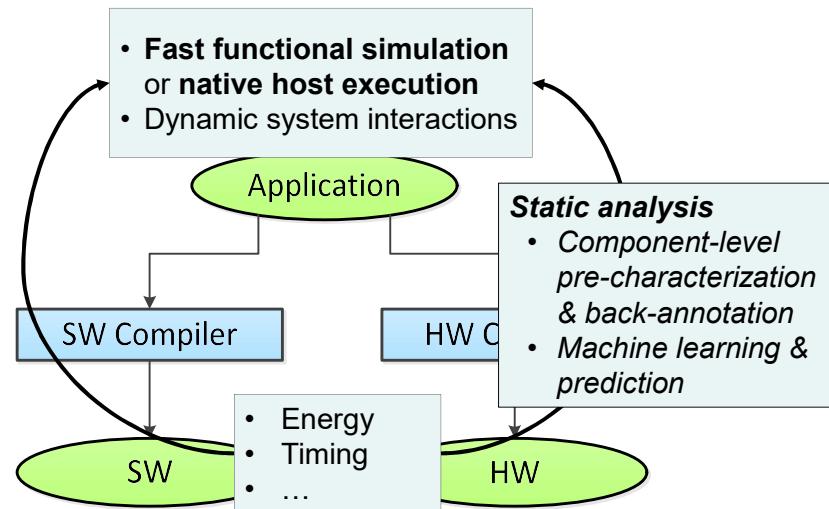
ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 4

© 2024 A. Gerstlauer

12



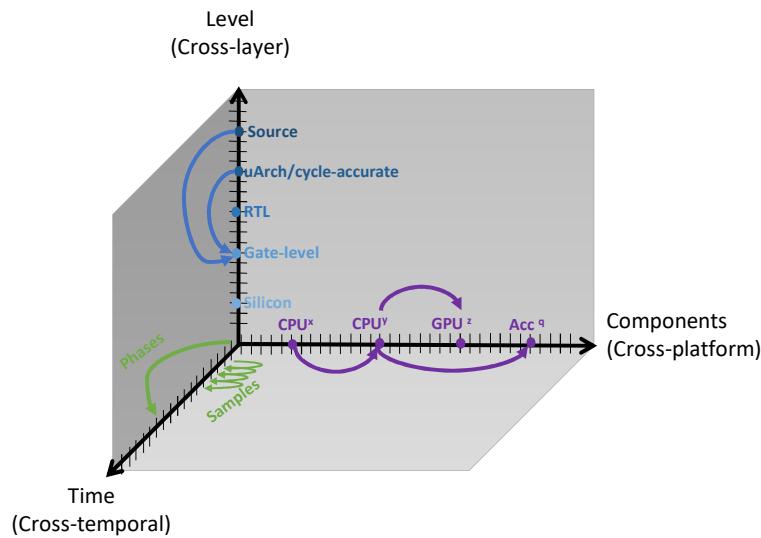
Semi-Analytical Modeling



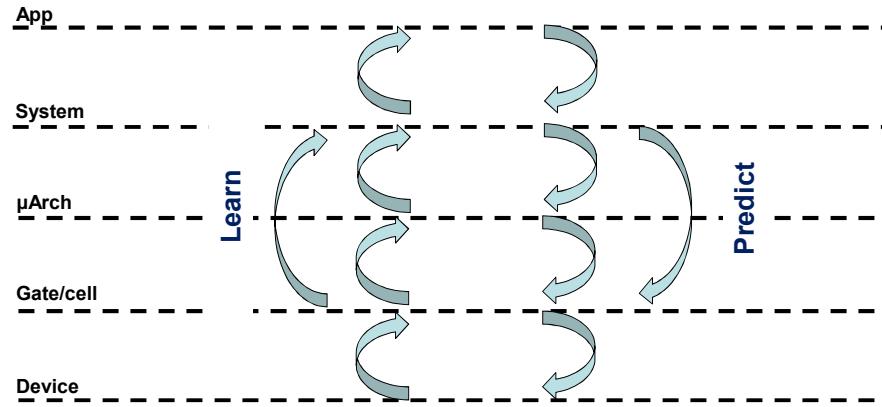
ML for System-Level Modeling

- **Learn rather than construct models**
 - Learn an abstract model from detailed observations
 - **Predict rather than simulate**
 - Replace detailed simulations with predictions
- **Supervised regression formulations**
- Interpolate/extrapolate (complex) behavior from (a few) samples
 - Linear regression: most problems are not linear
 - Non-linear regression: advanced non-linear models
 - Deep learning: often infeasible amount of training

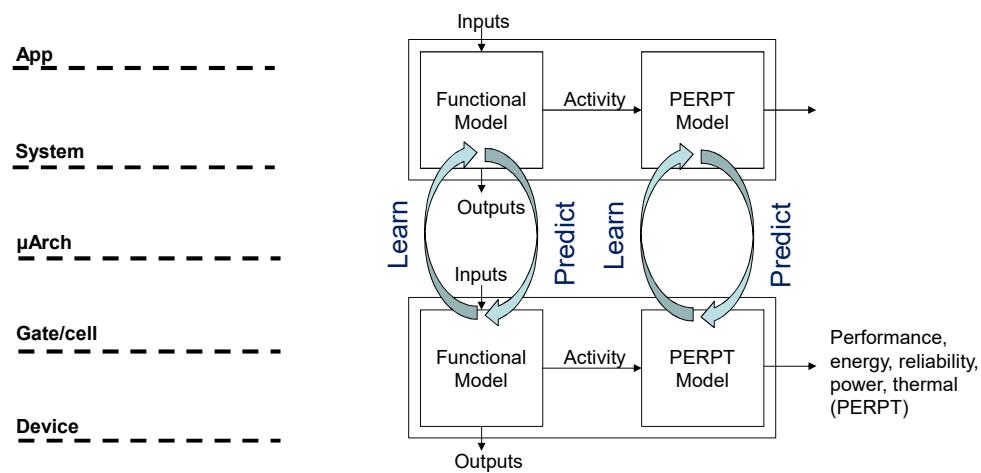
System-Level Predictive Modeling



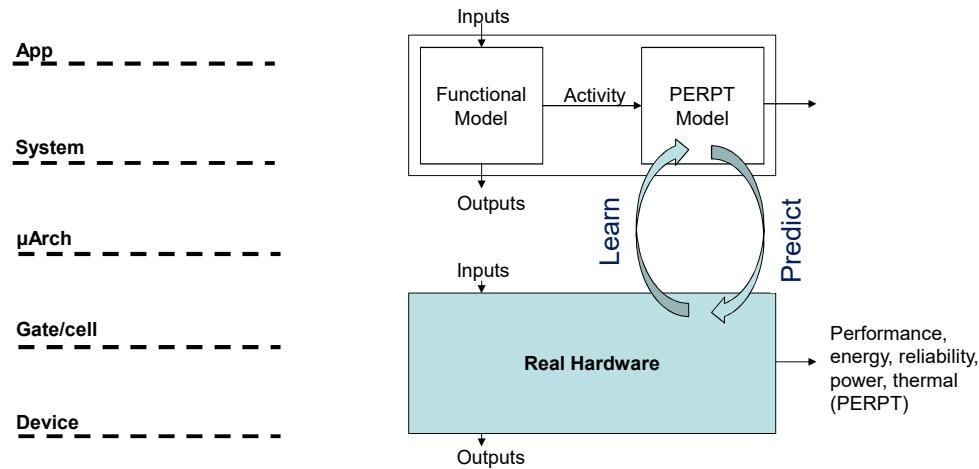
Cross-Layer Prediction



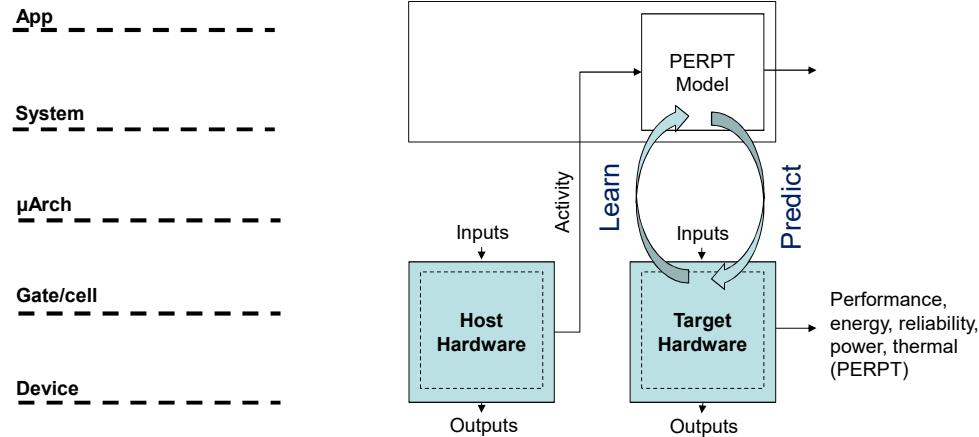
Cross-Layer Prediction



Cross-Layer Prediction



Cross-Platform Prediction



Lecture 4: Summary

- **Simulation**

- Detailed system simulation
- Trace-based simulation

- **Static analysis**

- Worst/best/average case bounds
- Execution time analysis of single task
- Real-time calculus for concurrent systems

- **Hybrid approaches**

- Semi-analytical modeling
- Machine learning-based prediction