

ECE382N.23: Embedded System Design and Modeling

Lecture 1 – Introduction

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

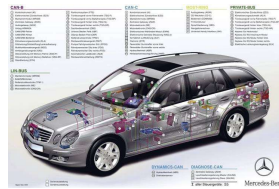
Lecture 1: Outline

- **Introduction**
 - Embedded systems
 - Design challenges
 - Formal methods and models
 - System-level design
- **Course information**
 - Topics
 - Logistics
 - Projects
- **System-level design process**
 - Models and methodologies
 - Machine learning (ML) for system-level design

Embedded Systems

- **System-in-a-system**

- Application-specific
 - Not general purpose
 - Known a priori
- Tightly constrained
 - Correctness, safety
 - Real time/performance, power, cost, reliability, security, ...



- **Ubiquitous**

- Far bigger market than general-purpose computing (PCs, servers)
 - 98% of all processors sold [Turley02, embedded.com]

- **Growing complexities**

- Application demands & technological advances
- Increasingly networked and programmable
 - Cyber-Physical Systems (CPS), Internet of Things (IoT)
 - Autonomous systems, edge intelligence



Embedded System Design

- **Correctly implement a specific set of functions**

- Hardware and software

- **While satisfying constraints**

- Real-time, cost, energy, power, thermal, ...

- **Application-/domain-specific & resource-constrained**

- Opportunity and need to optimize
- Choice of *system architecture* and *application mapping*
 - Large design spaces, and growing

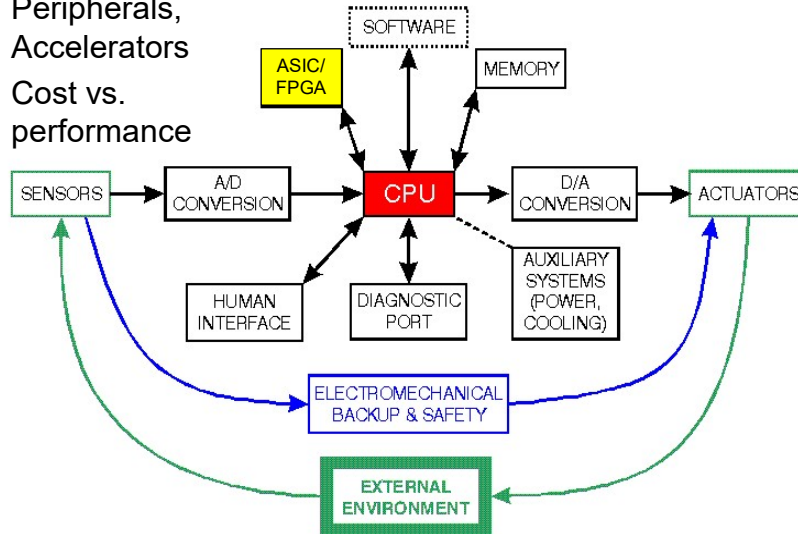
- **General-purpose computing seeing similar needs**

- Physical limits of scaling w/ power, thermal, ... constraints
- Application/architecture specialization & optimization
 - The two worlds are merging...

Traditional Embedded System

- CPU-centric design

- Peripherals, Accelerators
- Cost vs. performance



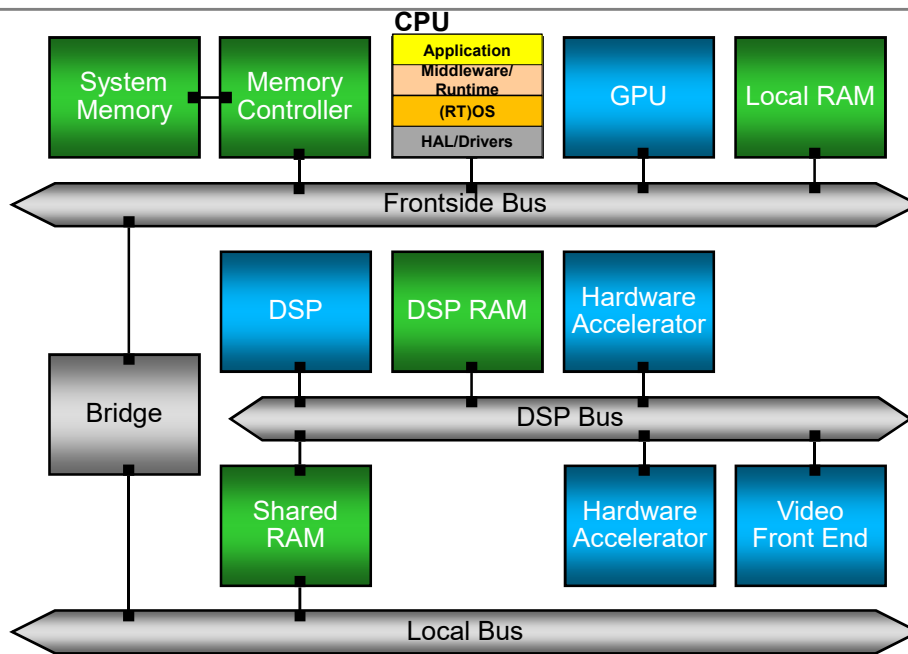
Source: M. Jacome, UT Austin

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

5

Multi-Processor System-on-Chip (MPSoC)



Source: C. Haubelt, Univ. of Rostock

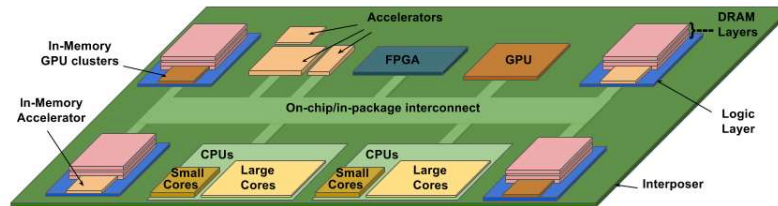
ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

6

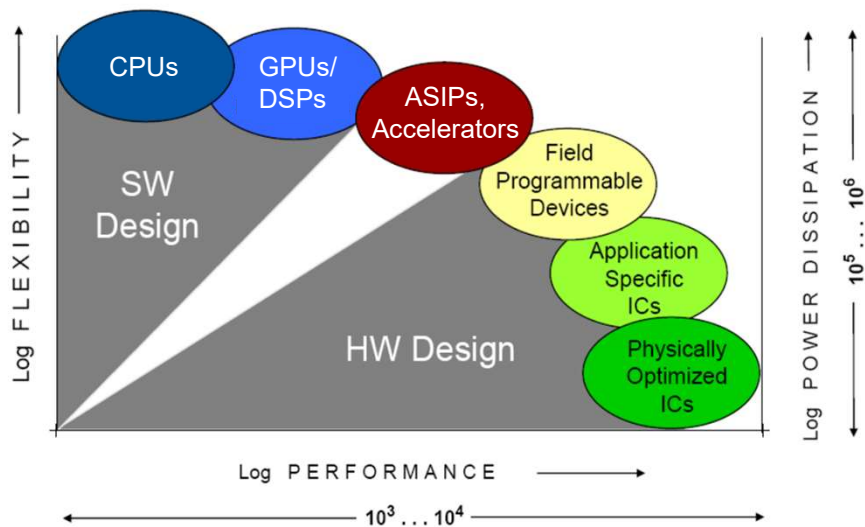
“General-Purpose” Computing

- **Heterogeneous, accelerator-rich architectures**
 - CPUs, GPUs, TPUs, custom hardware accelerators
 - In-/near-memory computing
 - Chiplet-based system-in-package (SiP) integration



- **Domain-specific computing**
 - Complexity & heterogeneity

Implementation Options



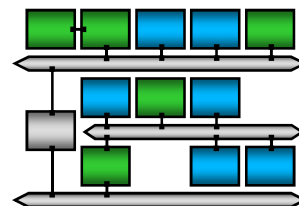
Source: T. Noll, RWTH Aachen, via R. Leupers, "From ASIP to MPSoC", Computer Engineering Colloquium, TU Delft, 2006

Design Challenges

- **Complexity**
 - High degree of parallelism
 - Large design spaces
 - Multiple optimization objectives & tight design constraints
 - Cost, performance, power, ...
 - Reliability, safety
- **Heterogeneity**
 - Of components
 - Processors, memories, busses
 - Of applications
 - Workload variation
 - Of design tasks
 - Architecture design, application mapping



Programming Model?



Domain-specific system-on-chip (DSSoC)

Source: C. Haubelt, Univ. of Rostock

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

9

Automated Design Methods

- **Managing complexity, heterogeneity, correctness challenges**
 - Mix of hardware design with software design
 - Mixes design styles within each of these categories
 - Mix of abstraction/detail/specificity
- **Automated analysis, synthesis, design space exploration (DSE)**
 - Systematic specification, modeling and design techniques
- **Formal methods for analysis, synthesis & exploration are key**
 - Application of optimization algorithms
 - Correctness verification and correct-by-construction synthesis



Key need ⇒ Abstractions into formal models to capture/express the various types of behavior at different abstraction levels, and how those diverse formal models interact and can be analyzed and synthesized.

Source: M. Jacome, UT Austin

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

10

Design Automation Methodology

- Design should be based on the use of one or more *formal models* to describe the desired *behavior* of the *system* at a high level of abstraction
 - Such behavior should be captured on an unbiased way, that is, before a decision on its decomposition into hardware and software components is taken
- The final implementation of the system should be generated as much as possible using *automatic synthesis* from this high level of abstraction
 - To ensure implementations that are “correct by construction”
- Validation (through *simulation* or *verification*) should be done as much as possible at the higher levels of abstraction

➤ Model-based design

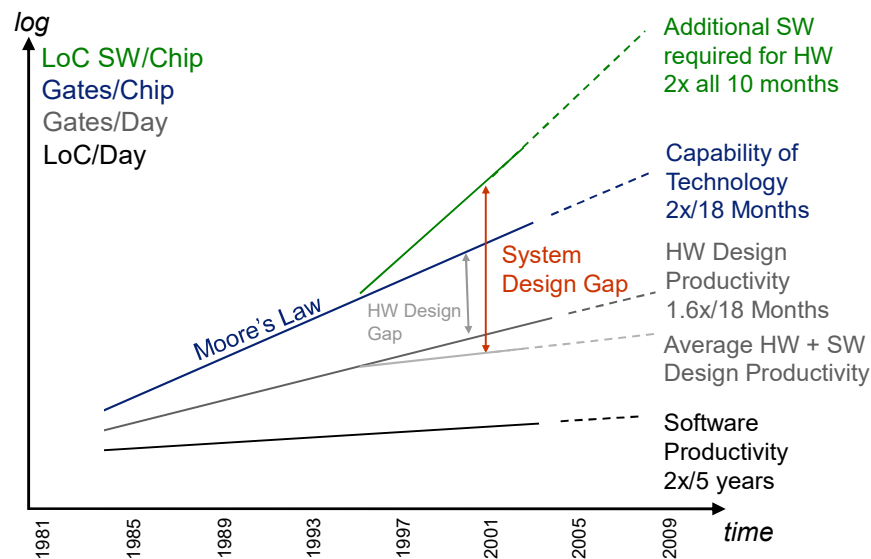
Source: M. Jacome, UT Austin

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

11

Productivity Gaps



Source: W. Ecker, W. Müller, R. Dömer, *Hardware-dependent Software - Principles and Practice*, Springer 2009.

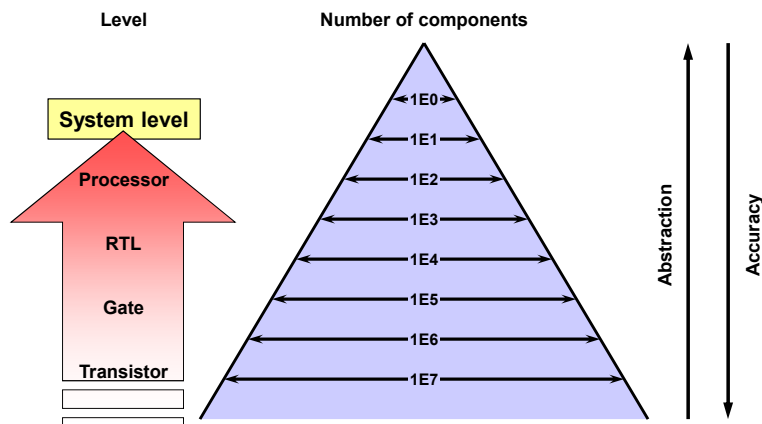
ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

12

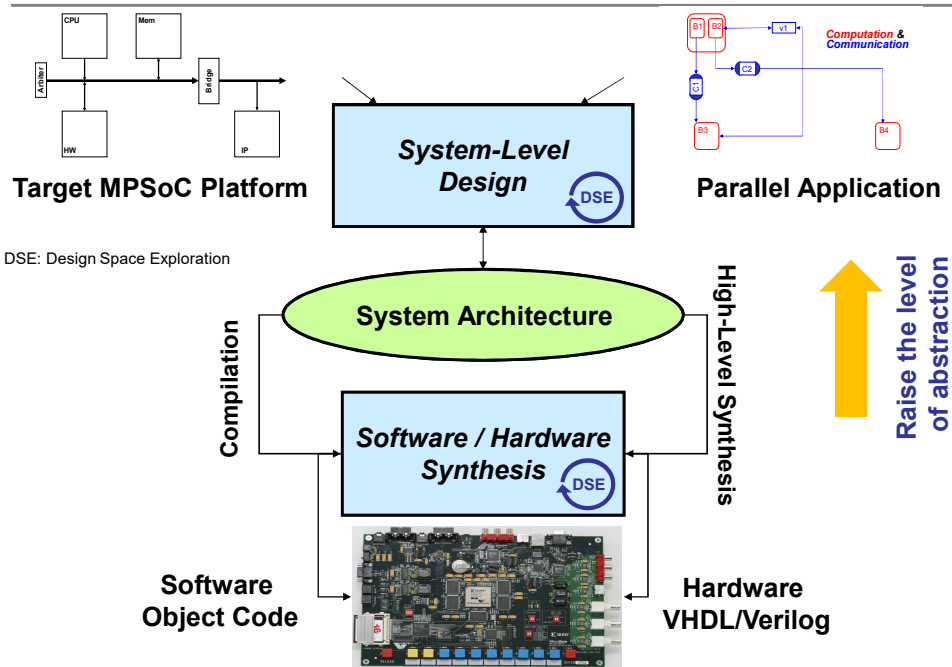
Abstraction Levels

- Move to higher levels of abstraction [ITRS07, itrs.net]
 - System-level design

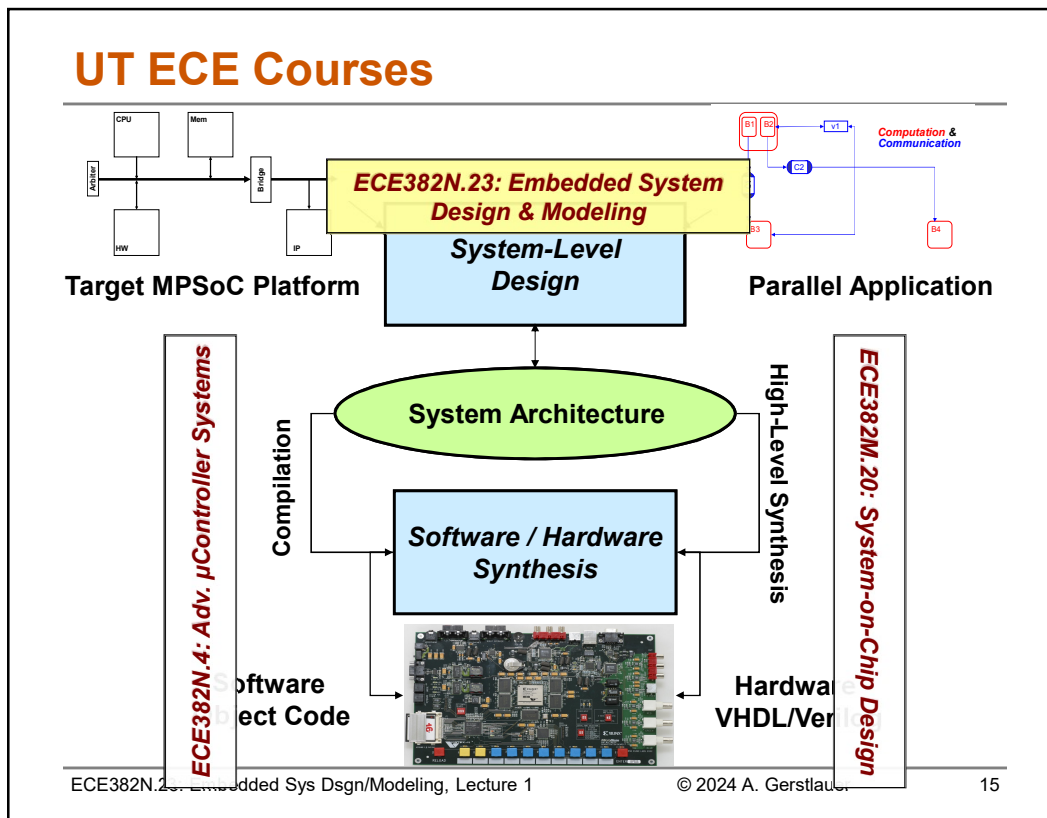


Source: R. Doemer, UC Irvine

System-Level Design



DSE: Design Space Exploration



Lecture 1: Outline

✓ Introduction

- ✓ Embedded systems
- ✓ Design challenges
- ✓ Formal methods and models
- ✓ System-level design

• Course information

- Topics
- Logistics
- Projects

• System-level design process

- Models and methodologies
- Machine learning (ML) for system-level design

Course Topics

➤ System-level design of application-/domain-specific SoCs

1. System application specification

- Application & programming models
 - Models of computation, process networks, dataflow, task graphs, FSMs
 - Workload modeling and prediction

2. System architecture modeling

- Energy/performance/cost modeling & estimation
 - Analytical and simulation-based modeling
 - Machine learning (ML)-based predictive models

3. System synthesis

- Design space exploration (DSE) and optimization
 - Exploration & optimization heuristics and meta-heuristics
 - Reinforcement and other learning-based DSE methods

➤ Prerequisites

- Software: C/C++ (algorithms and data structures)
- Hardware: VHDL/Verilog (digital design)
- Embedded systems and embedded software

Class Administration

• Schedule

- Lectures: TTh 12:30-2:00pm, ECJ 1.318

• Instructor

- Prof. Andreas Gerstlauer <gerstl@ece.utexas.edu>
 - Office hours: EER 5.882, TTh 11am-noon, or after class/by appointment

• Information

- Web page: http://www.ece.utexas.edu/~gerstl/ece382n_f24
- Announcements, assignments, grades: Canvas
- Questions, discussions: Ed Discussion

• Dates (tentative)

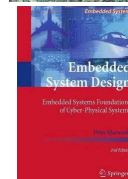
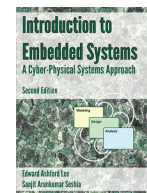
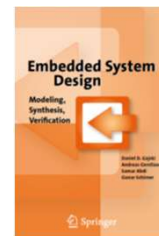
- Project proposal: September 12 (Canvas)
- Project review presentations: October 15 & 17 (in class)
- Final project presentations: December 3 & 5 (in class)
- Final project report: December 10 (Canvas)

Policies

- **Grading**
 - Homework and reading assignments: 35%
 - Class participation: 15%
 - Class project: 50%
 - No late submissions
- **Academic dishonesty**
 - Homeworks are independent
 - Discuss questions and problems with others
 - Turn in own, independently developed solution
 - Project is teamwork
 - Teams of up to 3 students
 - One report and presentation per team

Textbooks

- **None required, recommended background:**
 - D. Gajski, S. Abdi, A. Gerstlauer, G. Schirner, *Embedded System Design: Modeling, Synthesis, Verification*, Springer, 2009 (“orange book”)
 - <http://www.cecs.uci.edu/embedded-system-design-book/>
 - E. A Lee, S. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, 2nd ed., 2017
 - Available for download at <http://leeseshia.org>
 - P. Marwedel, *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*, 4th ed., Springer, 2021
 - <https://link.springer.com/book/10.1007/978-3-030-60910-8>



Project

- **Two options**
 - Research project
 - System-level design research problem
 - Implementation project
 - Non-trivial embedded system design example/case study
- **Project timeline (tentative)**
 - Proposal: September 12 (Canvas)
 - Single page with title, team members, abstract
 - Review: October 15 & 17 (in class)
 - Research project: literature survey/related work, preliminary results
 - Implementation project: system specification, initial analysis
 - Presentations: December 3 & 5 (in class)
 - Final project documentation & results
 - Final Report: December 10 (Canvas)
 - Research project: research paper in publishable quality
 - Implementation project: working system (in simulation or physical)

Some Possible Projects

- **Design projects**
 - (Embedded) system design example
 - Specify, model, explore, and implement
 - » Embedded machine learning, TinyML, IoT, e.g. in health care, robotics, smart X...
 - » Your choice of tools and target hardware/software platform
- **Research projects**
 - Modeling
 - Application modeling
 - » Application analysis and ML-based prediction
 - Energy/performance modeling
 - » High-level analytical power/performance models or simulators
 - » ML-based power/performance/.... prediction
 - » Parallel or FPGA-based simulation of hardware/software/network systems
 - Synthesis & exploration
 - Pick an optimization/exploration problem and solve it
 - » ML-based design space exploration and optimization methods
 - » Hardware or software synthesis for different optimization targets

Successful Past Class Projects

- **Modeling**
 - A. K. Ananda Kumar, "Learning-Based CPU Power Modeling," MLCAD'19.
 - K. Punniyamurthy, B. Boroujerdian, "GATSim: Abstract Timing Simulation of GPUs," DATE'17.
 - X. Zheng, "Learning-Based Analytical Cross-Platform Performance Prediction," SAMOS'15 (**best paper award**)
 - A. Abdel-Hadi, J. Michel, "Real-Time Optimization of Video Transmission in a Network of AAVs," VTC'11.
 - A. Pedram, C. Craven, T. Amimeur, "Modeling Cache Effects at the Transaction Level," IESS'09 (**best paper runner-up**)
 - A. Banerjee, "Transaction Level Modeling of Best Effort Channels for Networked Embedded Devices", IESS'09.
- **Exploration and synthesis**
 - K. Mirzazad, Z. Zhao, "Quality/Latency-Aware Real-time Scheduling of Distributed Streaming IoT Applications," ACM TECS 2019.
 - S. Lee, K. Saleem, J. Li, "Fine Grain Word Length Optimization for Dynamic Precision Scaling in DSP Systems," VLSI-SoC'13 (**best paper candidate**)
 - J. Lin, A. Srivatsa, "Heterogeneous Multiprocessor Mapping for Real-Time Streaming Systems," ICASSP'11.

Lecture 1: Outline

- ✓ **Introduction**
 - ✓ Embedded systems
 - ✓ Design challenges
 - ✓ Formal methods and models
 - ✓ System-level design
- ✓ **Course information**
 - ✓ Topics
 - ✓ Logistics
 - ✓ Projects
- **System-level design process**
 - Models and methodologies
 - Machine learning (ML) for system-level design

Design Process

- **Sequence of steps that transforms a set of requirements described informally into a detailed description that can be used for manufacturing**
 - Intermediate steps with transformation from a more abstract description to a more detailed one (*refinement*)
- **A designer can perform step-by-step refinement**
 - The “input” description is a *specification*
 - The final description of the design is an *implementation*
- **Take a model of the design at a level of abstraction and refine it to a lower one (level of detail ↑).**
 - Ensure that the **properties at the lower level of abstraction are verified**, and that the **performance indices are satisfactory**
 - Thus, refinement process involves **mapping constraints, performance indices and properties to the lower level**, so that they can be computed for the next level down

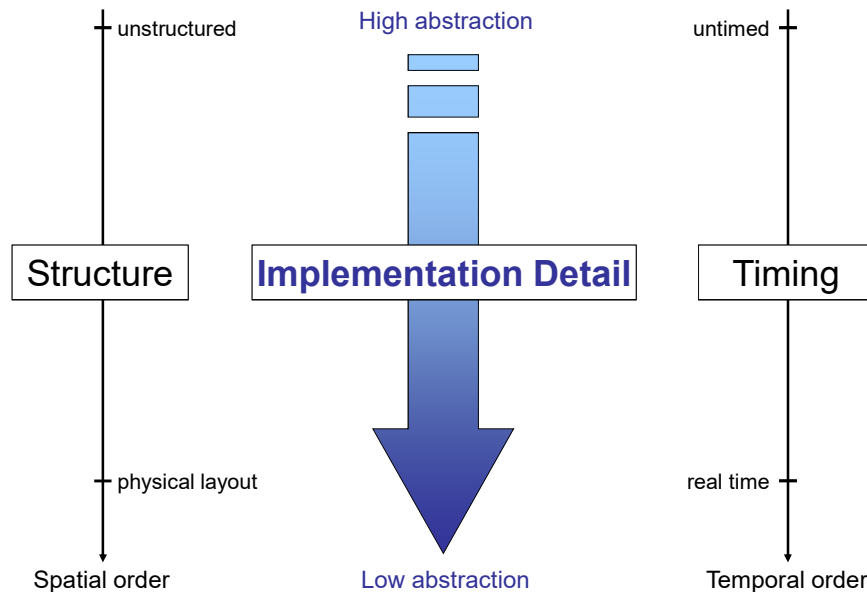
Source: M. Jacome, UT Austin

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

25

Abstraction Levels



ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

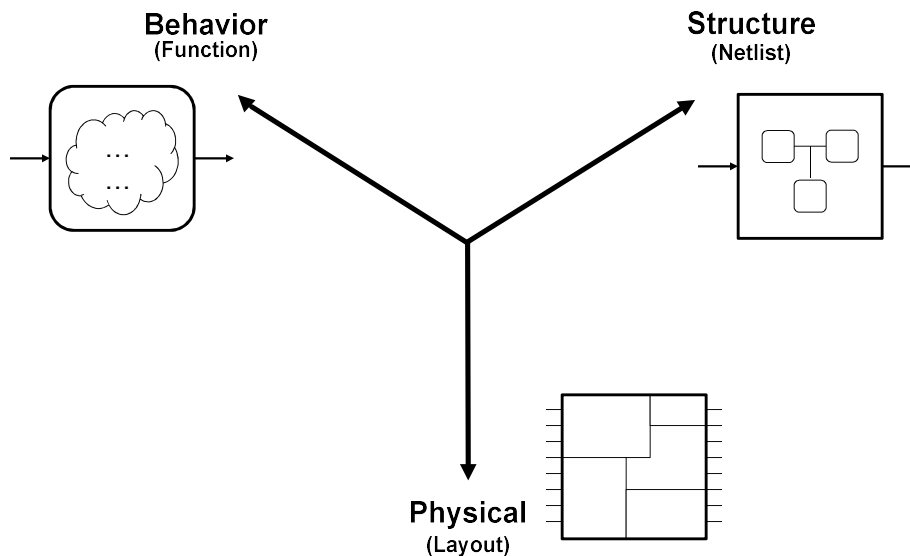
26

Design Methodology

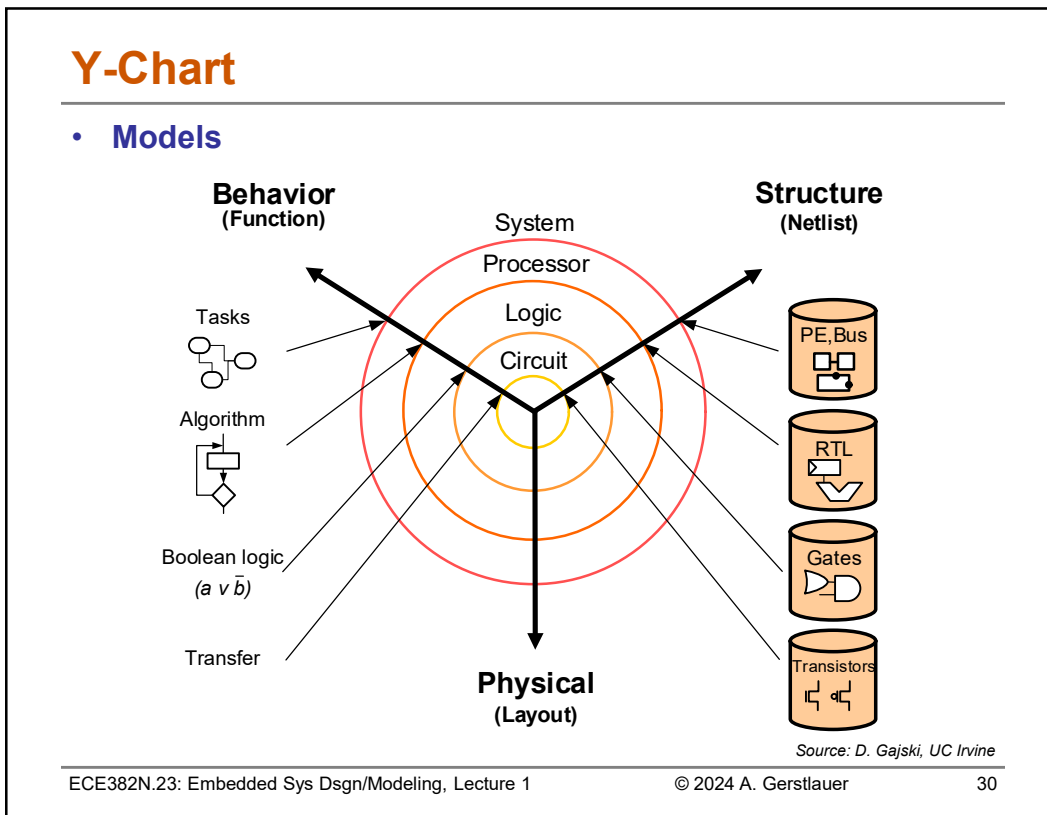
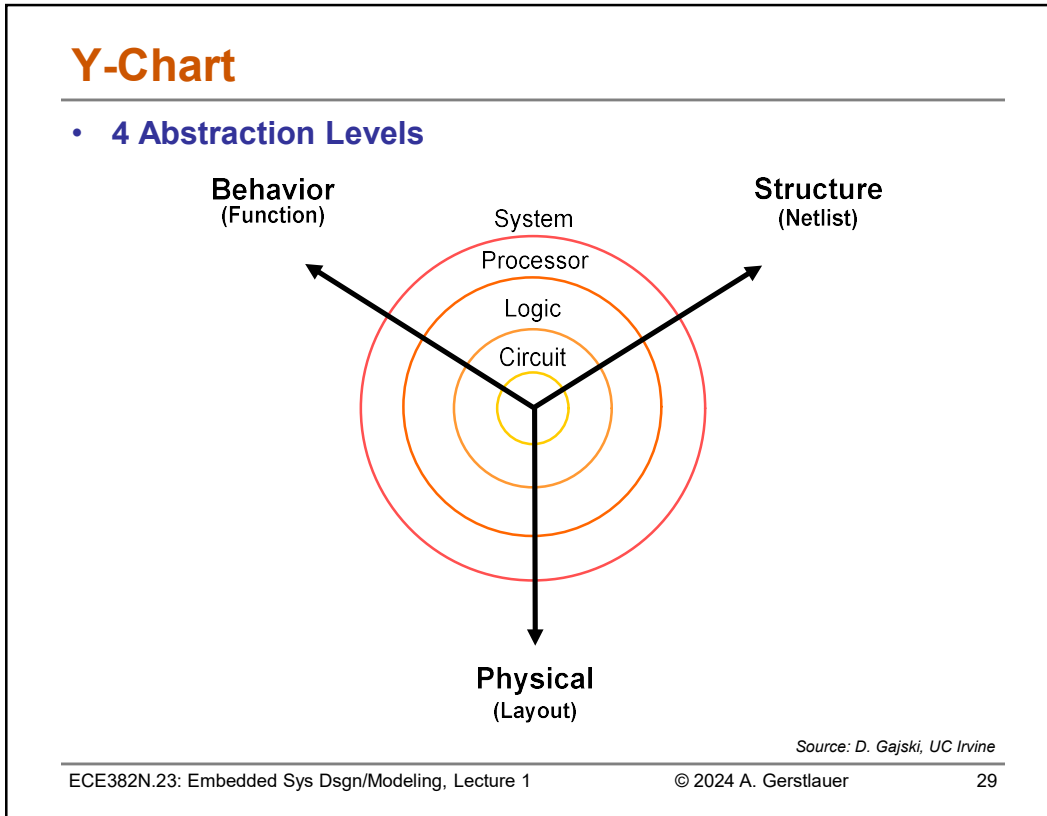
- **Set of Models**
 - Design representations
 - Specification and documentation at interface between steps
- **Set of Transformations**
 - Design decisions and design steps
 - Refine input model into an output model reflecting decisions
- **Formalization of a design flow**
 - Break into well-defined, repeatable steps
 - Automate

Y-Chart

- **3 Design Views**

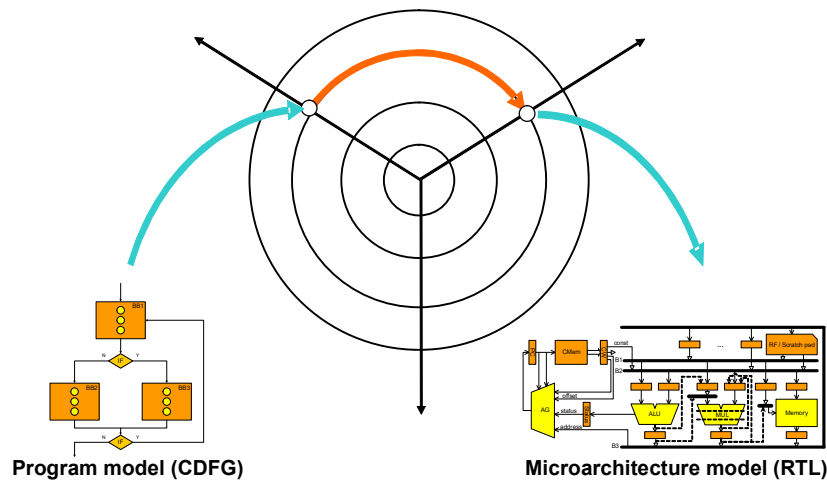


Source: D. Gajski, UC Irvine



Processor Synthesis

- **Software processor**
 - Compilation and linking
- **Hardware processor**
 - High-level synthesis



Source: D. Gajski, UC Irvine

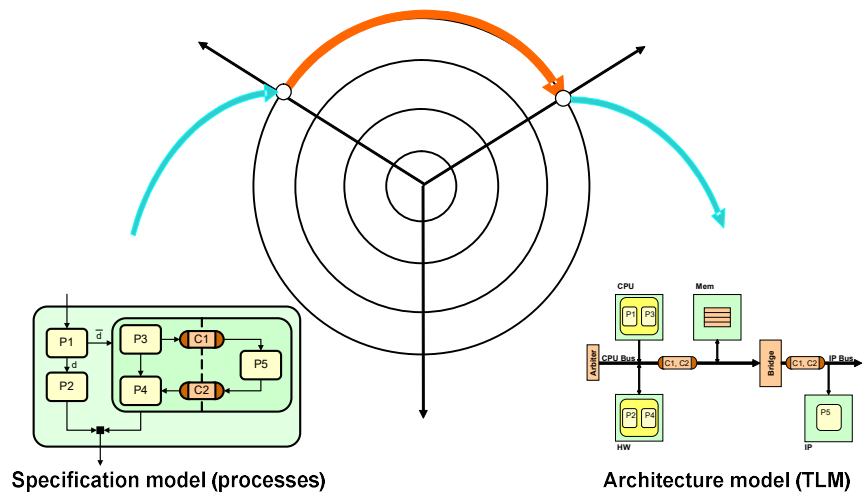
ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

31

System Synthesis

- **Structure**
 - Partitioning, mapping
- **Timing**
 - Scheduling



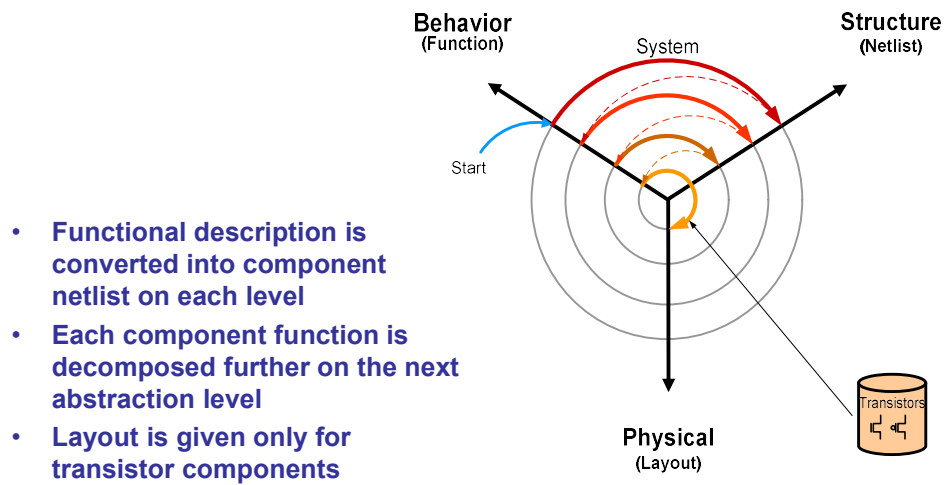
Source: D. Gajski, UC Irvine

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

32

Top-down Methodology



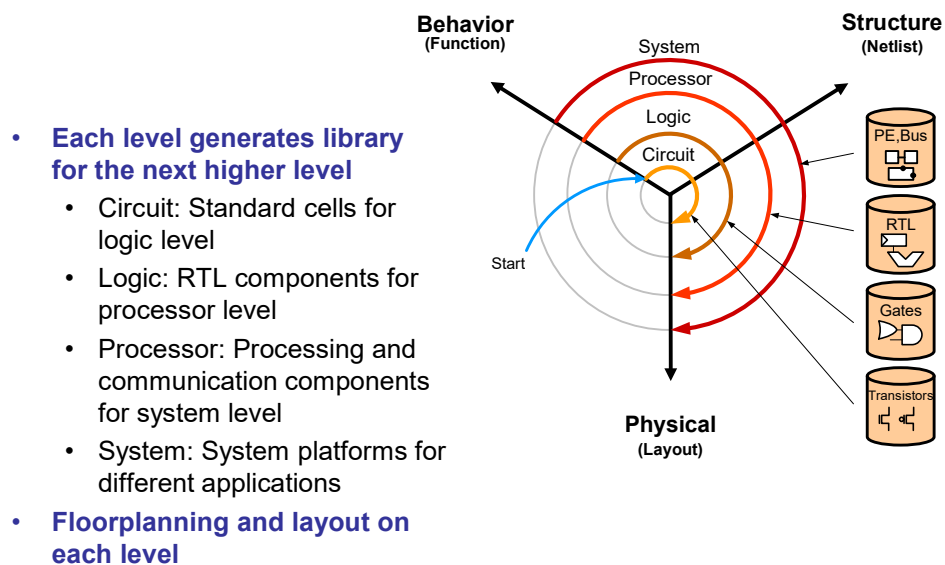
Source: D. Gajski, UC Irvine

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

33

Bottom-Up Methodology



Source: D. Gajski, UC Irvine

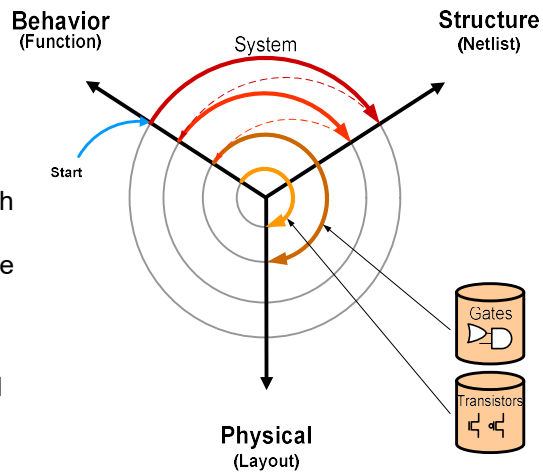
ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

34

Meet-in-the-Middle Methodology

- **Gate netlist is hand-off**
- **Three levels of synthesis**
 - System is synthesized with processor components
 - Processor components are synthesized with RTL library
 - RTL components are synthesized with standard cells
- **Two levels of layout**
 - System layout is performed with standard cells
 - Standard cells layout with transistors



Source: D. Gajski, UC Irvine

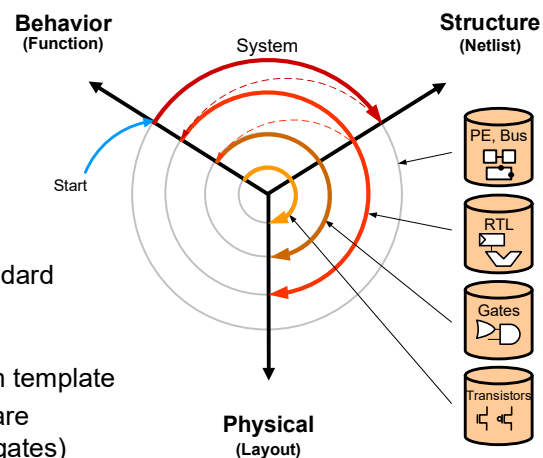
ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

35

Platform-Based Design

- **Meet-in-the-middle at the system level**
- **RTL is hand-off**
 - System platform with standard components
 - System synthesis to map specification onto platform template
 - Some custom processor are synthesized (to RTL and gates)
 - Other (programmable) processors are pre-synthesized and just need software compilation
 - Layout and floorplanning at the SoC level

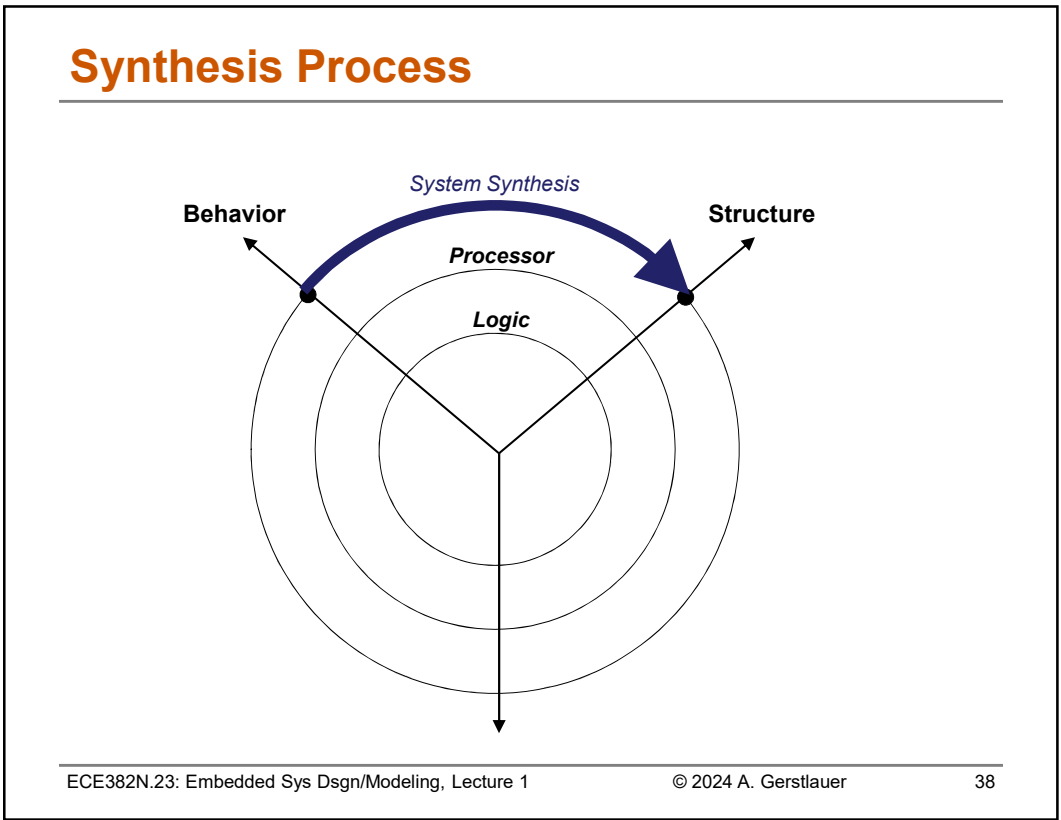
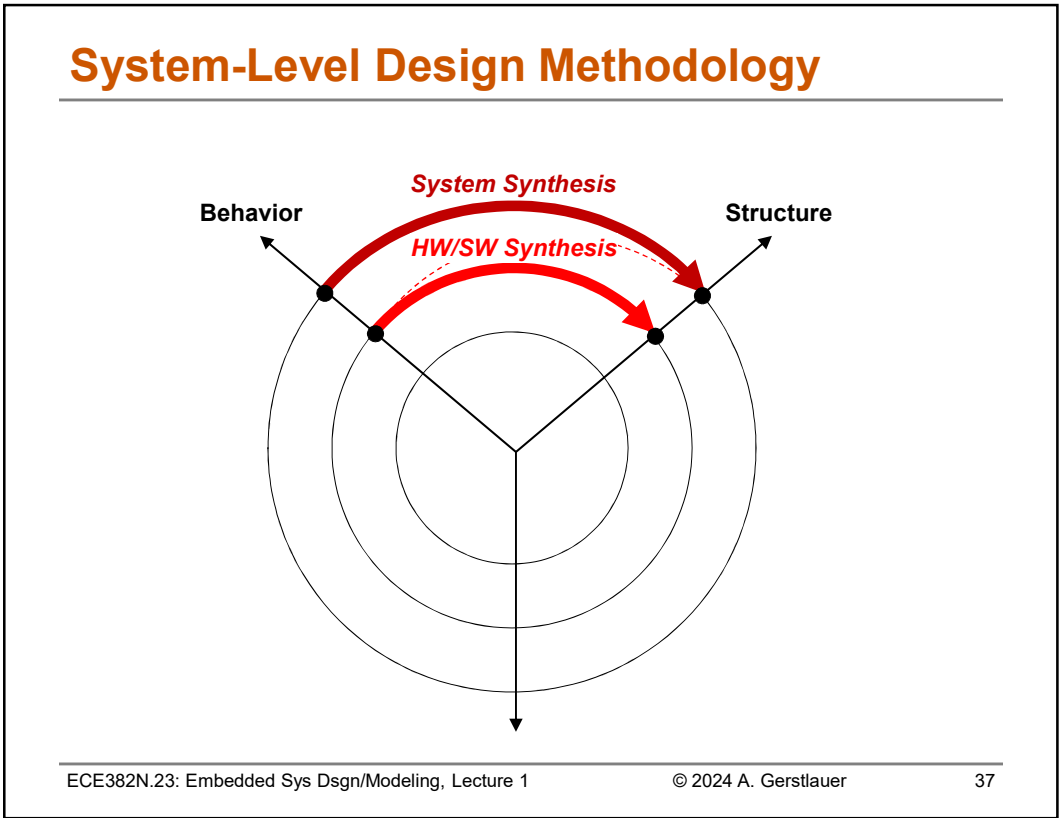


Source: D. Gajski, UC Irvine

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

36



Synthesis Process

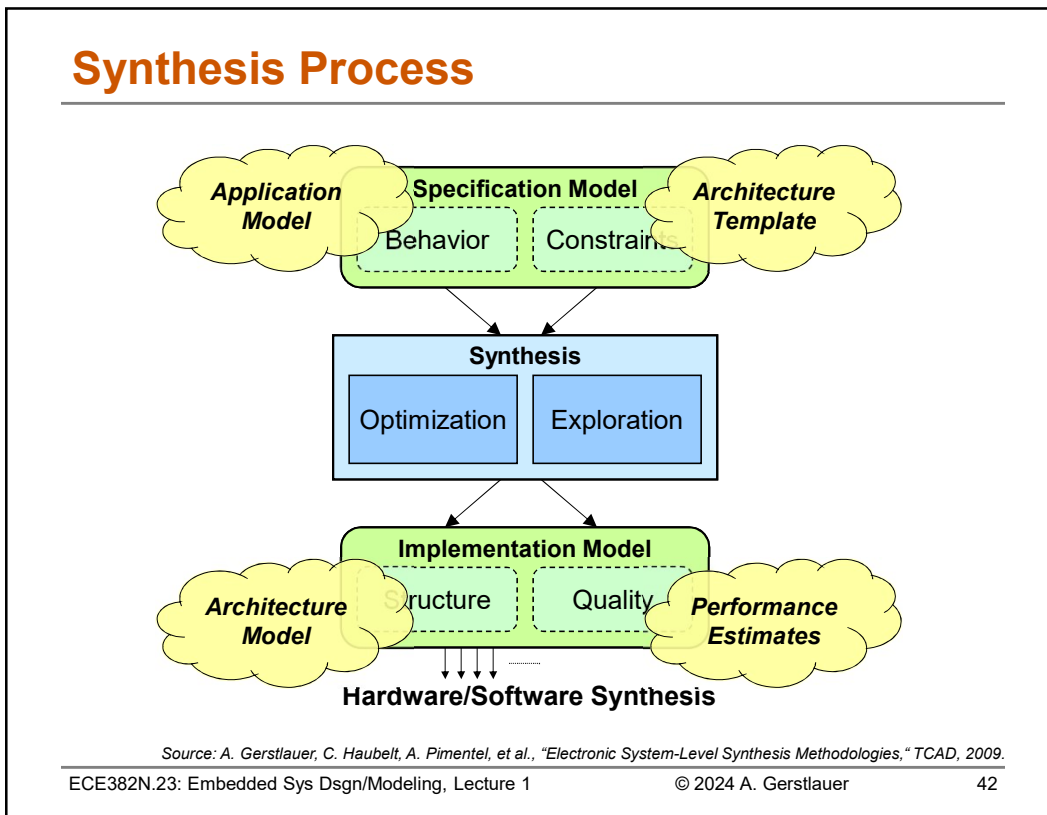
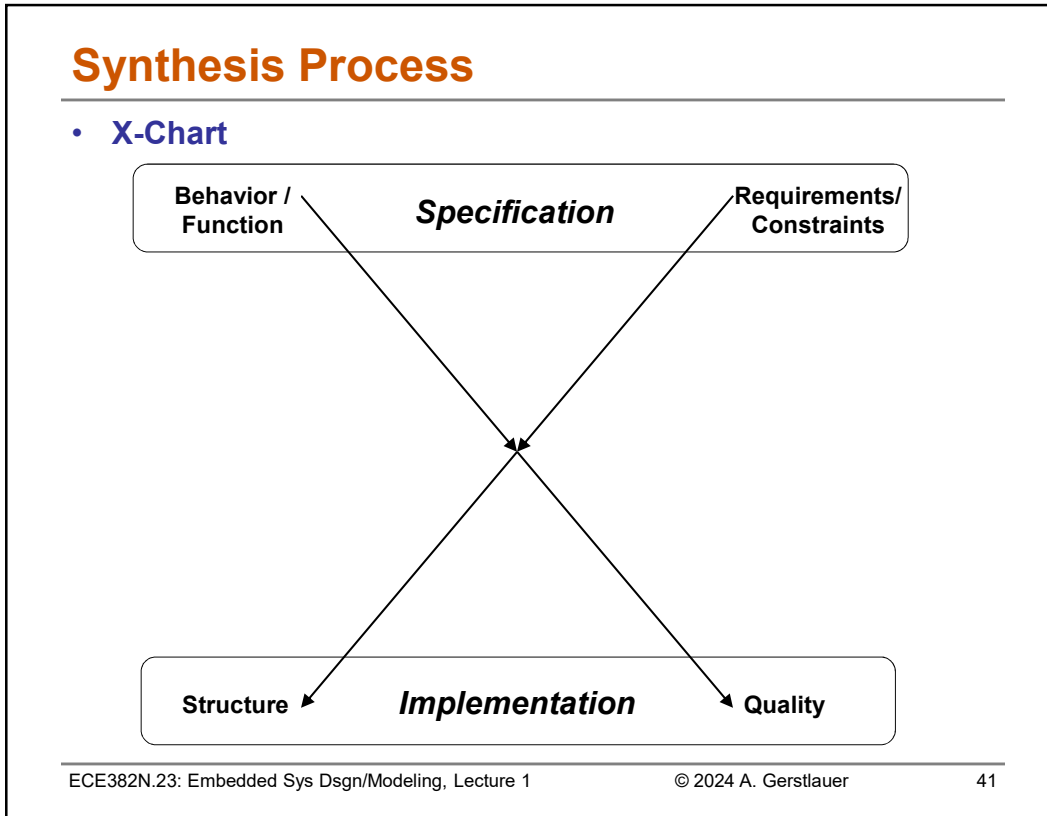
- **Y-Chart**

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1 © 2024 A. Gerstlauer 39

Synthesis Process

- **Platform-based design**

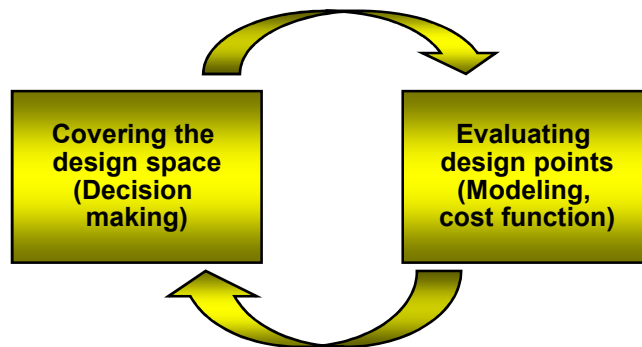
ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1 © 2024 A. Gerstlauer 40



Optimization & Exploration

➤ Design space exploration (DSE)

- Exact: exhaustive search
- Constructive/incremental: gradually construct solution
- Iterative: gradually refine and improve solution



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

ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

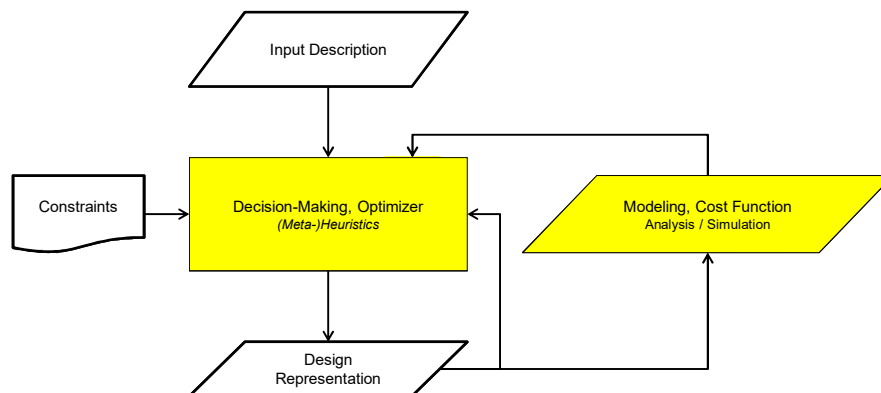
© 2024 A. Gerstlauer

43

Traditional Design Flow

• Manually derived heuristics & models

- Analytical & simulation-based models
- Optimization & exploration (meta-)heuristics



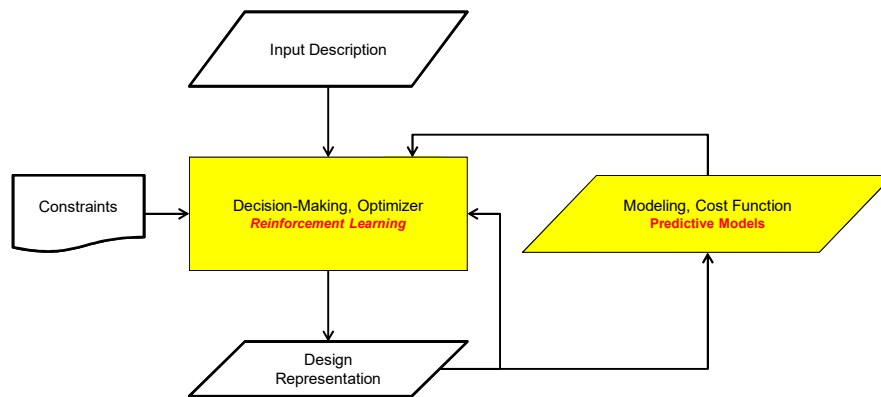
ECE382N.23: Embedded Sys Dsgn/Modeling, Lecture 1

© 2024 A. Gerstlauer

44

Machine Learning (ML)-Based Design Flow

- Apply advanced ML methods for modeling & optimization
 - Predictive modeling
 - Reinforcement or other learning for optimization



Lecture 1: Outline

✓ Introduction

- ✓ Embedded systems
- ✓ Design challenges
- ✓ Formal methods and models
- ✓ System-level design

✓ Course information

- ✓ Topics
- ✓ Logistics
- ✓ Projects

• System-level design process

- Models and methodologies
- Machine learning (ML) for system-level design