# ECE382N.23: Embedded System Design and Modeling

## Lecture 3 – System-Level Synthesis

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

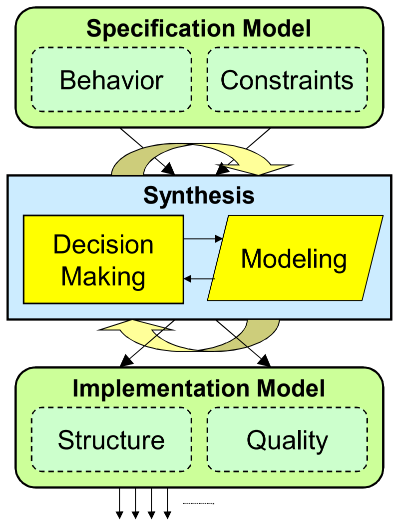
The University of Texas at Austin
Chandra Department of Electrical and Computer Engineering
Cockrell School of Engineering

## Lecture 3: Outline

- **System synthesis**
  - Overview
- **Synthesis process**
  - Allocation
  - Partitioning
  - Scheduling
- **Optimization & exploration**
  - Optimization formulation
  - Optimization methods

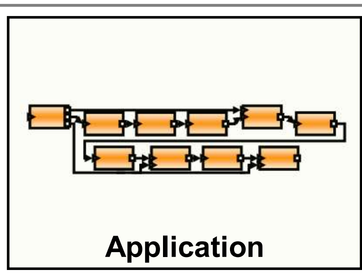
## System Synthesis

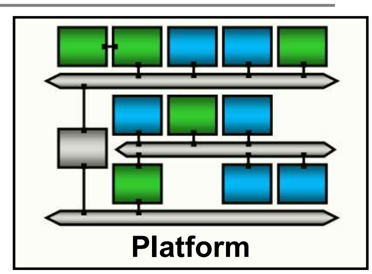
- **Decision making**
  - Allocation
  - Binding
  - Scheduling
- **Modeling**
  - Create and evaluate architecture & cost/quality models



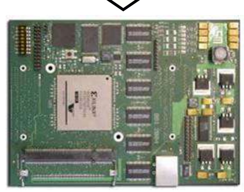



## System Synthesis Process



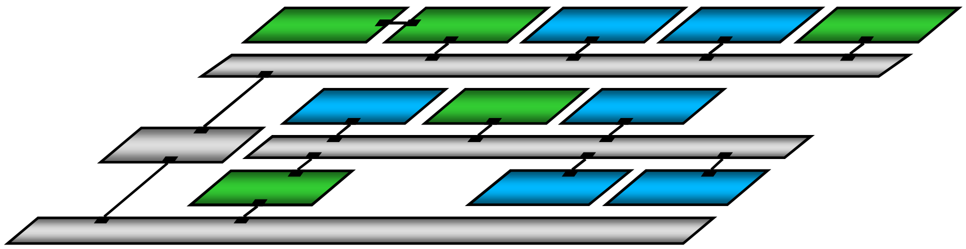

**Automatic Optimal Mapping ?**



*Source: C. Haubelt, J. Teich*

# Resource Allocation

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

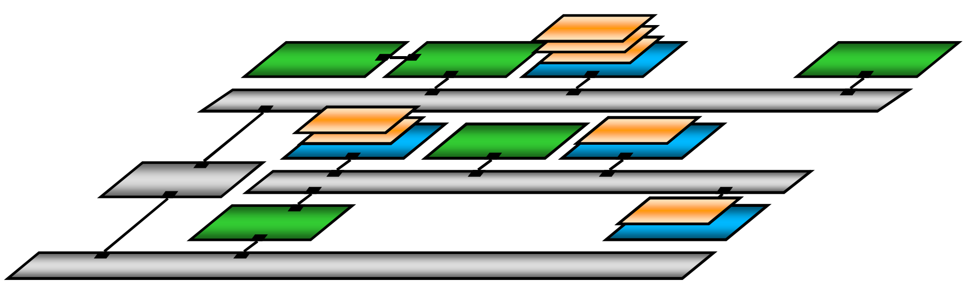


*Source: C. Haubelt, J. Teich*



# Process Binding

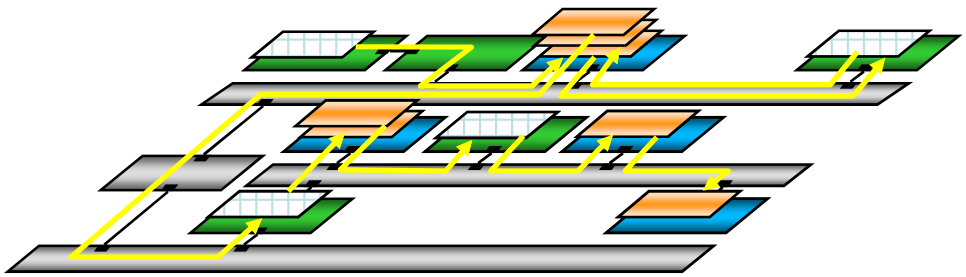
- **Process mapping, i.e., bind processes onto allocated computational resources**



*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*



## Automated Decision Making

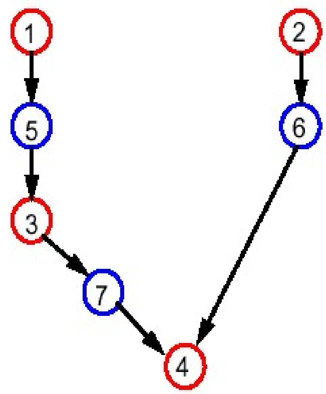
- **Map specification onto architecture**
  - Functionality + constraints ⇨ structure + metrics
- **Synthesis tasks**
  - Allocation
    - Select resources from a platform/architecture template (database)
  - Binding
    - Map processes onto allocated computational resources
    - Map variables onto allocated storage units
    - Route channels over busses, gateways and address spaces
  - Scheduling
    - Determine order of processes bound to the same resource
    - Determine order of transaction routed over the same (arbitration)
  - ➢ Partitioning = (allocation +) binding
  - ➢ Mapping = (allocation +) binding + scheduling

➢ **Formalize & automate the decision making process**

## Example (1)

- **Basic model with a task graph input and static scheduling**
  - Task graph = homogeneous, acyclic SDF

Application task graph $G_P(V_P, E_P)$




Interpretation:

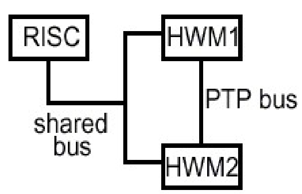
- $V_P$ consists of functional nodes $V_P^f$ (task, procedure) and communication nodes $V_P^c$ .
- $E_P$ represent data dependencies

*Source: L. Thiele*

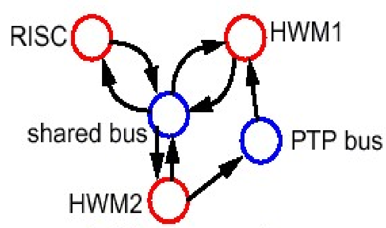


## Example (2)

Architecture graph $G_A(V_A,E_A)$:




Architecture



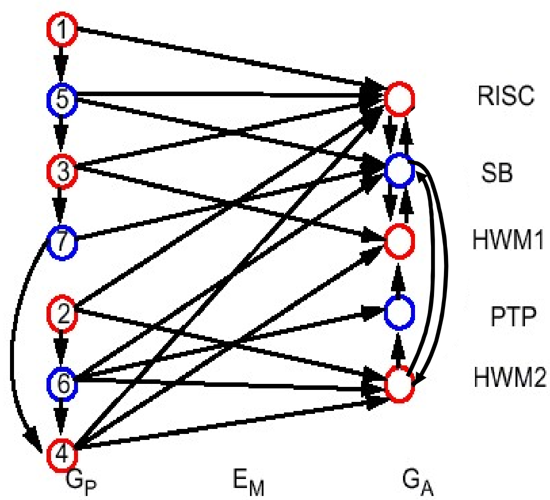

Architecture graph

- $V_A$ consists of functional resources $V_A^f$ (RISC, ASIC) and bus resources $V_A^c$. These components are potentially allocatable.
- $E_A$ model directed communication.

*Source: L. Thiele*

## Example (3)

Definition: A specification graph is a graph $G_S=(V_S,E_S)$ consisting of a problem graph $G_P$, an architecture graph $G_A$, and edges $E_M$. In particular, $V_S=V_P \cup V_A$, $E_S=E_P \cup E_A \cup E_M$




(mapping constraint)

Source: L. Thiele



## Example (4)

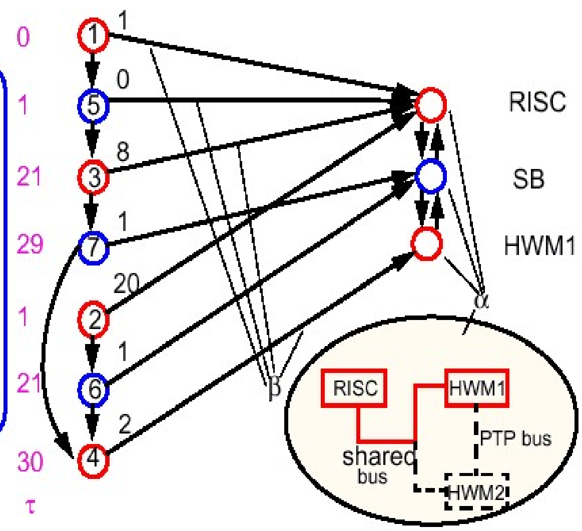
Three main tasks of synthesis:

- Allocation $\alpha$ is a subset of $V_A$.
- Binding $\beta$ is a subset of $E_M$, i.e., a mapping of functional nodes of $V_P$ onto resource nodes of $V_A$.
- Schedule $\tau$ is a function that assigns a number (start time) to each functional node.

Source: L. Thiele

## Example (5)

Definition: Given a specification graph $G_S$ an implementation is a triple ($\alpha$,$\beta$,$\tau$), where $\alpha$ is a feasible allocation, $\beta$ is a feasible binding, and $\tau$ is a schedule.




Source: L. Thiele



## Optimization Problems

- **Decision making under optimization objectives**
  - Single- vs. multi-objective optimization
  - Couple with refinement for full synthesis
- **General optimization formulation**
  - Decision variables: $x \in$ *Domain*
  - Constraints: $g_i(x) \leq G_i$, $h_j(x) = H_j$
  - Objective function: $f(x)$: *Domain* $\rightarrow \mathbb{R}$
  - Single-objective optimization problem:

$$\min_x f(x) \text{ subject to } g_i(x) \leq G_i, h_j(x) = H_j$$

- **System-level optimization**
  - Allocation ($\alpha$), binding ($\beta$), scheduling ($\tau$) decisions
  - Under functional and non-functional constraints/objectives
    - Architecture & mapping constraints ($G_A$, $E_m$)
    - Design quality constraints & objectives

## Cost / Objective Functions

- **Measure quality of a design point vs. optimization objective**
  - May include $C$ … system cost in [$]
    $L$ … latency in [sec]
    $P$… power consumption in [W]
- **Example: linear weighted cost function with penalty**

$$f(C, L, P) = k_1 \cdot h_C(C, C_{max}) + k_2 \cdot h_L(L, L_{max}) + k_3 \cdot h_P(P, P_{max})$$

  - $h_C$ , $h_L$ , $h_P$ … denote how strong $C, L, P$ violate the design constraints $C_{max}, L_{max}, P_{max}$
  - $k_1$ , $k_2$ , $k_3$ … weighting and normalization
- **Requires *estimation and/or evaluation* to find $C, L, P$**
  - Refinement + simulation (evaluation)
  - Analytical quality/cost models (estimation)
  - ML-based quality/cost models (prediction)

*Source: L. Thiele*



## Optimization & Exploration Methods

- **Exact (optimal) methods**
  - Enumeration, exhaustive search
  - Convex optimizations
  - (Integer) linear programming
  - ➢ Prohibitive for intractable problems (large design spaces)
- **Heuristics (non-optimal)**
  - Constructive
    - Random assignment, list schedulers
  - Iterative
    - Random search, simulated annealing
  - Set-based iterative
    - Evolutionary/genetic Algorithms (EA/GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO)
    - ➢ Multi-objective optimization (MOO), Design space exploration (DSE)

➢ **Exact & constructive methods imply analytical cost models**

*Source: C. Haubelt, J. Teich*

## Lecture 3: Summary

- **System-level synthesis**
  - Automatic decision making + modeling
- **Decision making**
  - Allocation
  - Partitioning
  - Scheduling
- **Optimization & exploration**
  - Decision variables, objectives and constraints
  - Single- vs. multi-objective optimization
  - Cost functions to quantify impact of decisions