Introduction and System Design Challenges

IL2206 Embedded Systems

Ingo Sander

KTH Royal Institute of Technology
Stockholm, Sweden
ingo@kth.se

Outline

1 Motivation

2 Embedded Systems

3 The Design Challenge

4 Correct-by-Construction Design

5 Course Outline

Outline

1 Motivation

2 Embedded Systems

3 The Design Challenge

4 Correct-by-Construction Design

5 Course Outline

A few introductory questions. . .

- What is the percentage of ’embedded processors’ within the total number of processors? Estimate!

A few introductory questions. . .

- What is the percentage of 'embedded processors' within the total number of processors? Estimate!
- How many processors are in a modern car?

A few introductory questions. . .

- What is the percentage of 'embedded processors' within the total number of processors? Estimate!
- How many processors are in a modern car?

- How much software (lines of code) is running in a modern car?

A few introductory questions. . .

- What is the percentage of 'embedded processors' within the total number of processors? Estimate!
- How many processors are in a modern car?
- How much software (lines of code) is running in a modern car?

Check the links on the Canvas page!

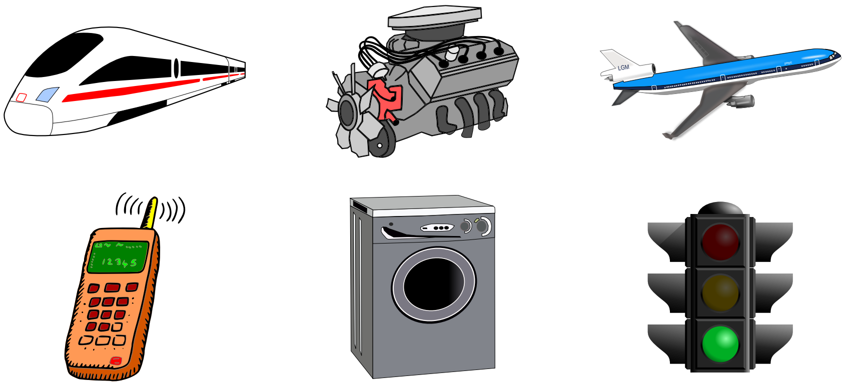
- Links to articles are provided for this lecture on Canvas.
- One interesting example: How many lines of code does it take?

Outline

1. Motivation
2. Embedded Systems
3. The Design Challenge
4. Correct-by-Construction Design
5. Course Outline

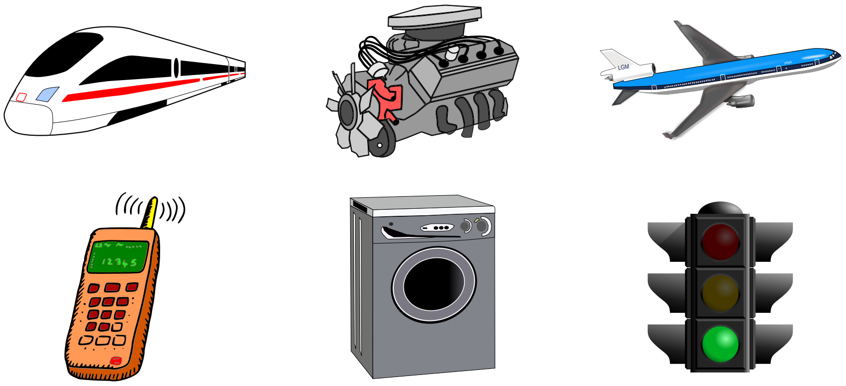
Embedded Systems are Everywhere

... and control vital functions in our daily life!



Embedded Systems are Everywhere

... and control vital functions in our daily life!



Designers have large responsibility!

Between 1985-87 several deaths and serious injuries of cancer patients were due to overdoses of radiation resulting from a race condition between concurrent tasks in the Therac-25 software (1985-87).

How can we avoid future accidents?

Embedded systems

- take over an increasing number of vital functions in our society
- include more and more functionality

How can we avoid future accidents?

Embedded systems

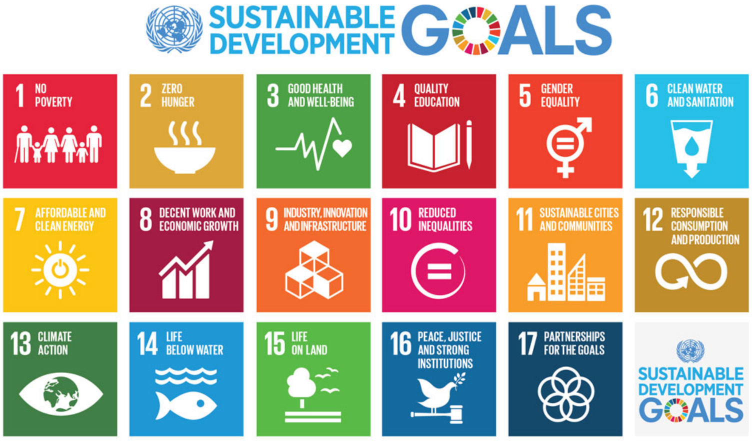
- take over an increasing number of vital functions in our society
- include more and more functionality

Increasingly complex design process

A disciplined design methodology is needed to design future embedded systems!

Embedded Systems and Sustainable Development

On September 25th 2015, countries adopted a set of goals to end poverty, protect the planet, and ensure prosperity for all as part of a new sustainable development agenda. Each goal has specific targets to be achieved over the next 15 years.


SUSTAINABLE DEVELOPMENT GOALS
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

[Source: United Nations, 2015[1]]

[1] http://www.un.org/sustainabledevelopment/sustainable-development-goals/

Embedded Systems and Sustainable Development

SUSTAINABLE DEVELOPMENT GOALS
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

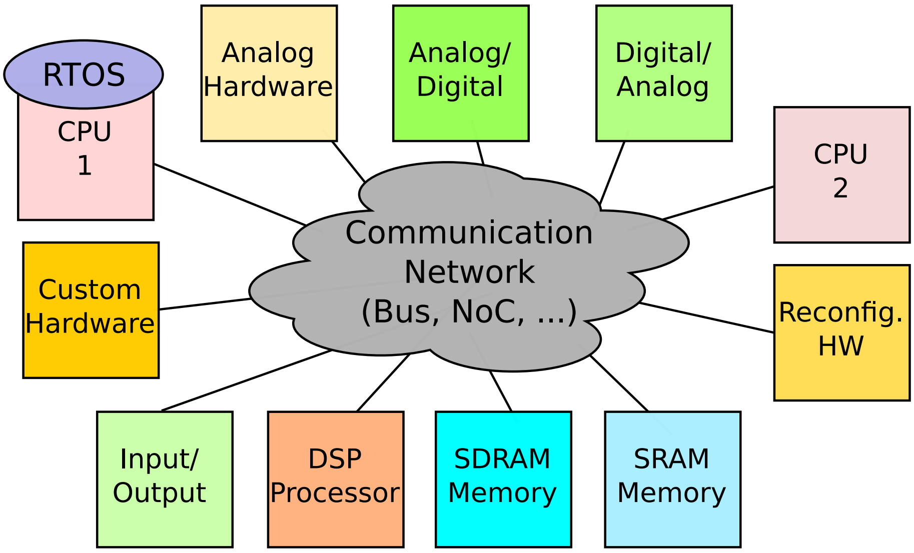
[Source: United Nations, 2015[1]]

How can embedded systems contribute to achieving these goals?

[1] http://www.un.org/sustainabledevelopment/sustainable-development-goals/

Heterogeneous Nature of Embedded Systems

- An embedded system interacts with the physical world and other embedded components
- An embedded system architecture consists of heterogeneous components



Characteristics of Embedded Systems

An embedded system

- is usually designed for one single task. Its functionality will never change.
- is often a mass product. Design cost is critical.
- interacts with the environment at the speed of the environment. Many embedded systems are safety-critical systems and have to fulfil hard real-time requirements.
- is often a hand-held device. Power-efficiency is critical.
- is often a consumer products. Time-to-market is critical.

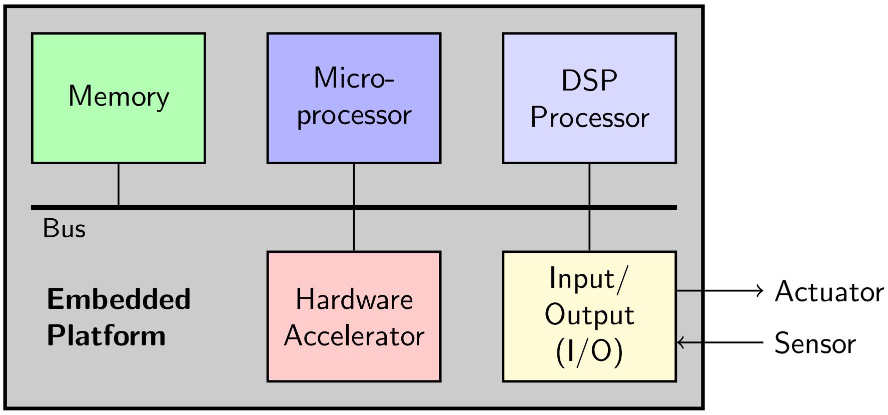
Design process for embedded systems is very different from general purpose programming!

- Embedded systems can be highly optimised
- All unneeded features are a disadvantage (cost, power)
- Design process must
  - be cost-efficient
  - ensure the correct functionality and timing of the implementation
  - be fast to ensure a short time-to-market

To be fast is not enough!

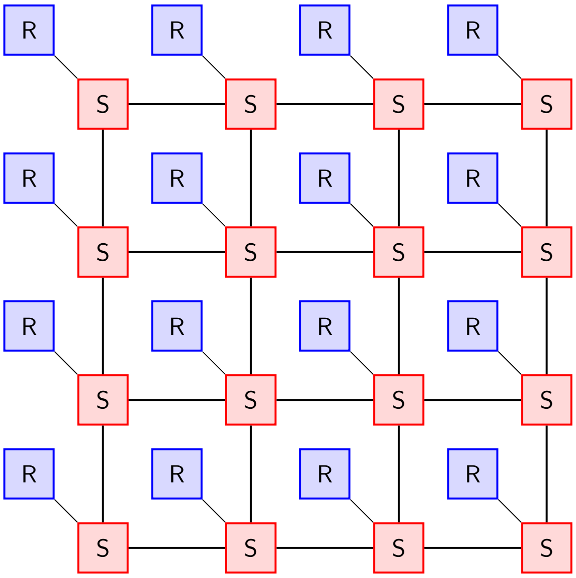
The system has to react to the environment at the right time instance, otherwise there can be fatal consequences.

Shared Memory Multiprocessor



- Several processing units share one or more memories
- Communication bottleneck due to shared resources and shared communication link

Network-on-Chip



- Resources (R) communicate via a network, that consists of switches (S) and communication links
- A resource can be a processor, a memory, a small bus-based system, custom hardware, . . .
- Scalable architecture with large communication bandwidth

Cyber-Physical Systems

- The term cyber-physical system is often used to describe a system that integrates computation with physical processes.
- Embedded systems are often used to control physical systems, so many embedded systems are part of cyber-physical systems.
- Terms are often used interchangeably, but the term embedded systems focuses more on the computation.
- The course will mainly use the terms embedded system and heterogeneous embedded system

Challenge

A successful design requires to understand the interaction between the physical components and the computational components.

Outline

1 Motivation

2 Embedded Systems

3 The Design Challenge

4 Correct-by-Construction Design

5 Course Outline

How to design an embedded system?

- Design Entry: Functional Specification
  - Overall functionality has to be understood and captured
  - Identification of system components
  - Details are not yet important
  - ⇒ System shall be modelled at a high abstraction level.

How to design an embedded system?

- Design Entry: Functional Specification
  - Overall functionality has to be understood and captured
  - Identification of system components
  - Details are not yet important

  ⇒ System shall be modelled at a high abstraction level.

  Design Objective: Final Implementation
  - Implementation details are important to fine-tune the design

  ⇒ System shall be modelled at a low abstraction level.

How to design an embedded system?

- Design Entry: Functional Specification
  - Overall functionality has to be understood and captured
  - Identification of system components
  - Details are not yet important

  ⇒ System shall be modelled at a high abstraction level.

  Design Objective: Final Implementation
  - Implementation details are important to fine-tune the design

  ⇒ System shall be modelled at a low abstraction level.

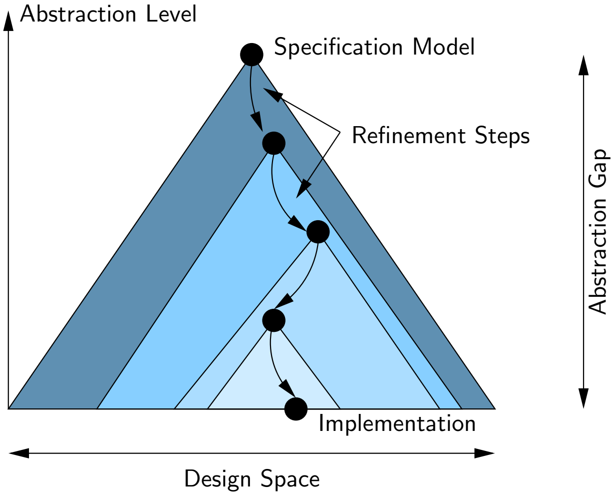
The challenge

How to bridge the abstraction gap?

Top-Down Design Process

The design specification is written at a high abstraction level

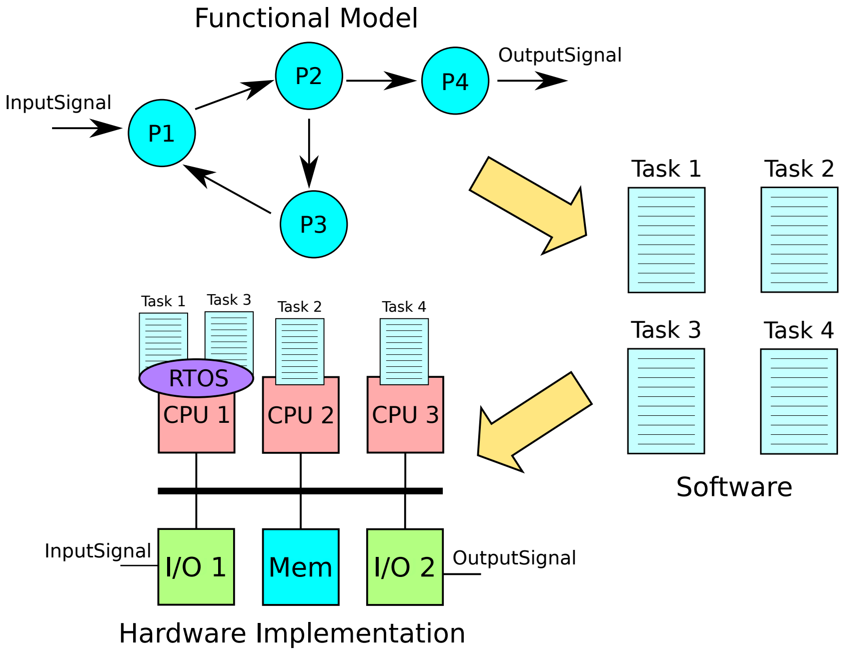
1. Many implementations comply with the specification. Together they comprise the design space.
2. During the design process the initial model is refined. The designer makes design decisions and the design space is reduced.
3. Finally the design process yields one single implementation. The implementation does not need to be optimal, but shall only fulfil the design requirements (functionality, power, speed, area, reliability, costs,. . .)



Models

The design process requires to express the system at different levels of abstraction. For each level a new model of the system is required.

Overview Design Process



Outline

1 Motivation

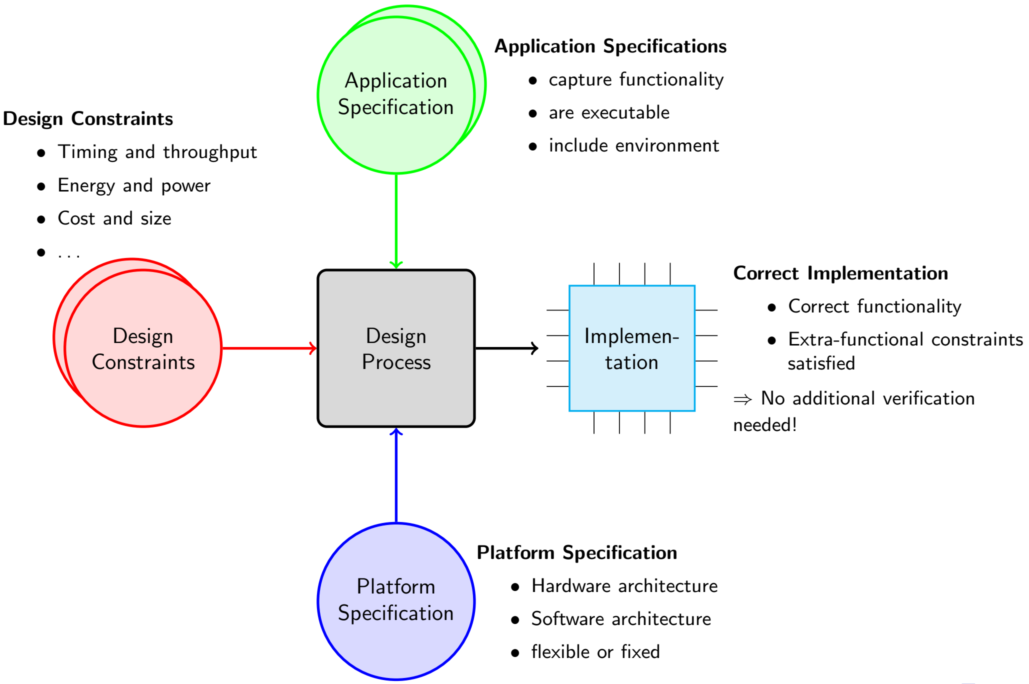
2 Embedded Systems

3 The Design Challenge

4 Correct-by-Construction Design

5 Course Outline

Correct-by-Construction Design: The Dream



Application Specification

Application Specifications

- capture functionality
- are executable
- include environment

Design Constraints

- Timing and throughput
- Energy and power
- Cost and size
- . . .

Design Constraints

Design Process

Implementation

Correct Implementation

- Correct functionality
- Extra-functional constraints satisfied

⇒ No additional verification needed!

Platform Specification

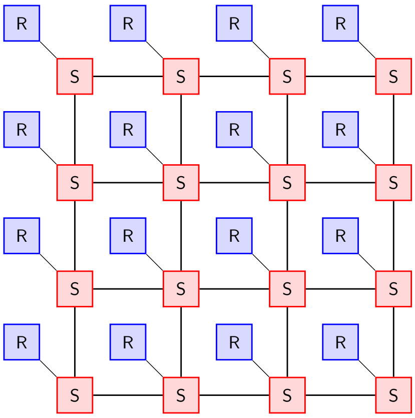
Platform Specification

- Hardware architecture
- Software architecture
- flexible or fixed

Design Challenge

Real-Time in the Many-Core Era

- Technology advances lead to
  - increasingly parallel, powerful and complex architectures
  - increasingly advanced and demanding applications
- Difficult to verify real-time requirements

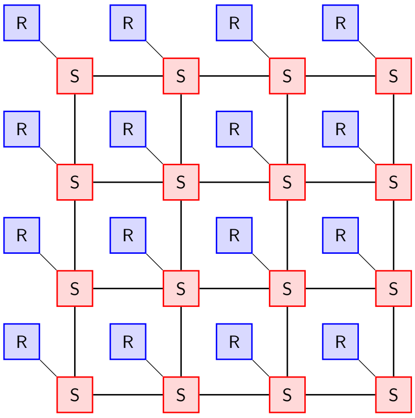


Network-on-Chip

Design Challenge

Real-Time in the Many-Core Era

- Technology advances lead to
  - increasingly parallel, powerful and complex architectures
  - increasingly advanced and demanding applications
- Difficult to verify real-time requirements



Network-on-Chip

We have already problems with complexity today! How do we design tomorrow's "sea-of-cores" real-time systems?

Embedded Real-Time Software Design: Current Situation

Very difficult to accurately estimate real-time performance

- Huge difference between average and worst-case execution time

Consequence

- New designs are rather based on old experience than on performance analysis
- Large safety-margins in form of more powerful components and extra communication bandwidth
- Verification costs are exploding

Embedded Real-Time Software Design: Current Situation

Very difficult to accurately estimate real-time performance

- Huge difference between average and worst-case execution time

Consequence

- New designs are rather based on old experience than on performance analysis
- Large safety-margins in form of more powerful components and extra communication bandwidth
- Verification costs are exploding

Current Situation

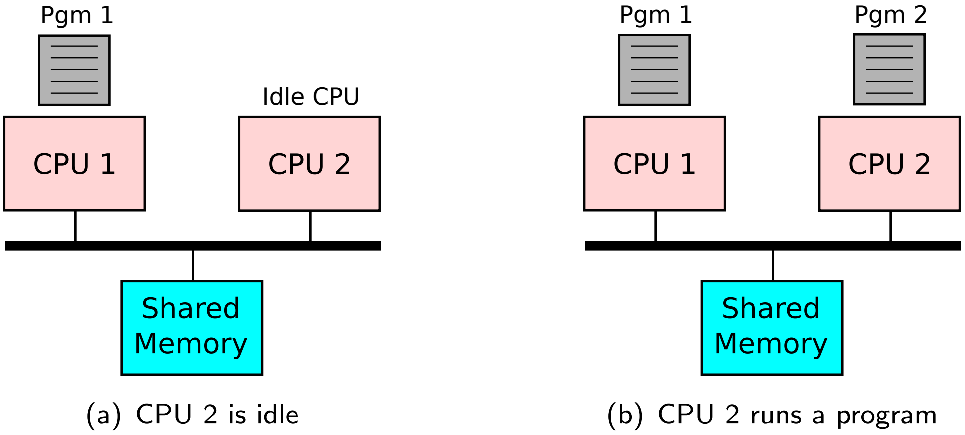
Far away from correct-by-construction!

Why is it so difficult to estimate software performance?

- Computation time is difficult to predict in advanced commercial processors due to mechanisms aimed to improve average case performance, e.g. caches, advanced pipelines
- Communication time is difficult to predict in multiprocessors due to
  - communication via shared resources, e.g. memory, communication channel
  - location dependent communication time (NUMA-architectures, NoC)
  - asynchronous communication mechanisms

Shared Memory Multiprocessor

Execution time of programs in a shared memory multiprocessor can in general not be analysed in isolation.

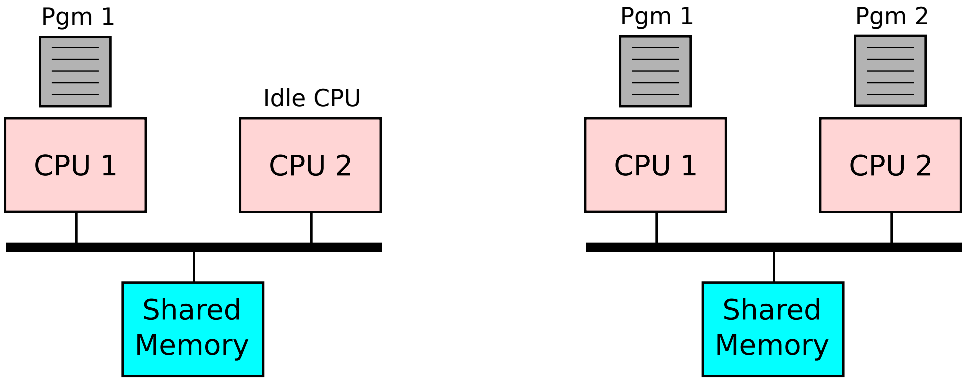


(a) CPU 2 is idle

(b) CPU 2 runs a program

Shared Memory Multiprocessor

Execution time of programs in a shared memory multiprocessor can in general not be analysed in isolation.

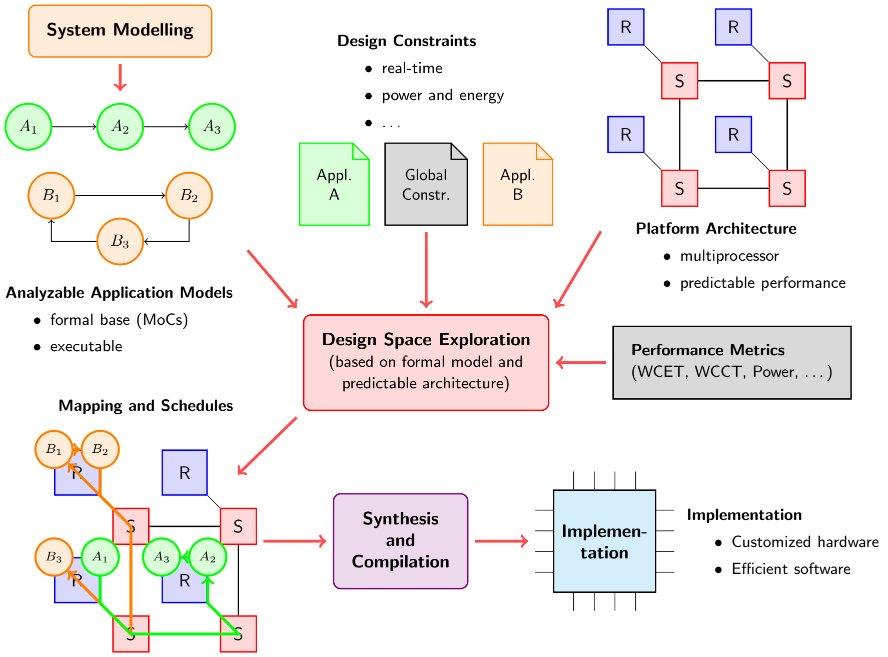


(a) CPU 2 is idle

(b) CPU 2 runs a program

Execution time for a program on one processor depends on memory access patterns of other processors ($t_{\text{pgm1,a}} \leq t_{\text{pgm1,b}}$)!

Sketch of Design Flow for Safety-Critical Embedded Systems



System Modelling

Analyzable Application Models

- formal base (MoCs)
- executable

Design Constraints

- real-time
- power and energy
- ...

Platform Architecture

- multiprocessor
- predictable performance

Design Space Exploration (based on formal model and predictable architecture)

Performance Metrics (WCET, WCCT, Power, ...)

Mapping and Schedules

Synthesis and Compilation

Implementation

- Customized hardware
- Efficient software

Outline

1 Motivation

2 Embedded Systems

3 The Design Challenge

4 Correct-by-Construction Design

5 Course Outline

Course Focus: IL2206 Embedded Systems

- The course cannot cover all aspects of the design of electronic systems
- Focus is put on the design of real-time embedded single-processor systems
- In particular, the course will also discuss techniques that can support a correct-by-construction design process can be supported
- This requires to carefully understand how
  - the different phases of the design process, and
  - the different components, hardware and software architecture,

  can fit together
- Main topics
  - Embedded computing platform
  - Introduction to real-time systems
  - Introduction to hardware-software co-design
  - Software development for real-time embedded systems