Task Mapping and Partition Allocation for Mixed-Criticality Real-Time Systems

Domițian Tămaș-Selicean and Paul Pop

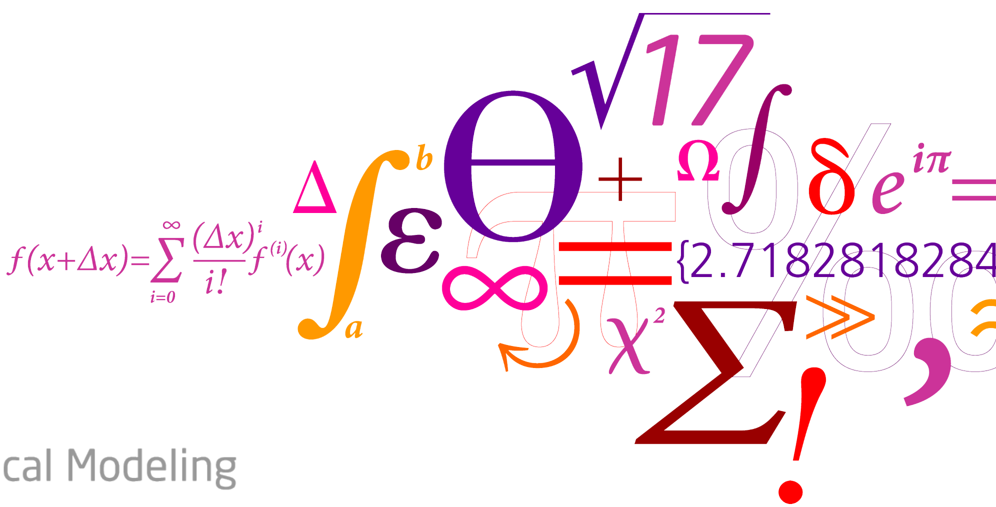
Technical University of Denmark

recomp

Reduced Certification Costs for trusted Multi-core Platforms

DTU Informatics

Department of Informatics and Mathematical Modeling



Outline

- Motivation
- System and application models
- Problem formulation and example
- Optimization strategy
- Experimental results
- Conclusions

Motivation

- **Safety** is the property of a system that will not endanger human life or the environment
- A safety-related system needs to be **certified**

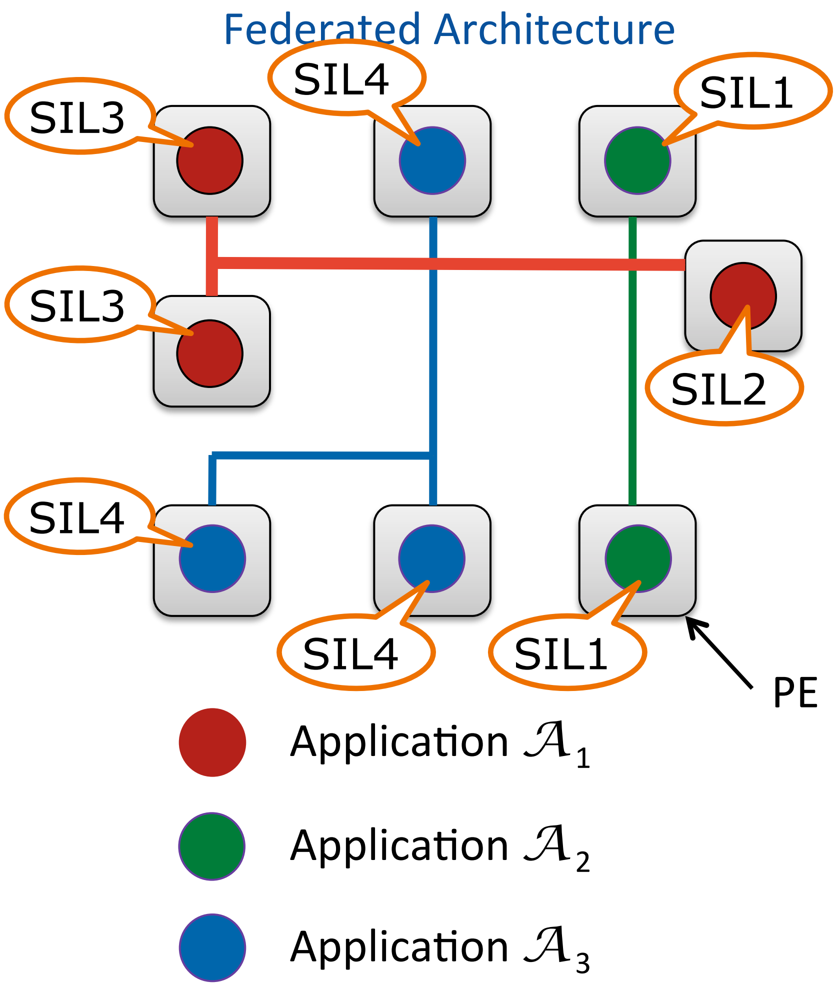
- A Safety Integrity Level (SIL) is assigned to each safety related function, depending on the required level of risk reduction
- There are 4 SILs:
  - SIL4 (most critical)
  - SIL1 (least critical)
  - SIL0 (non-critical) – not covered by standards

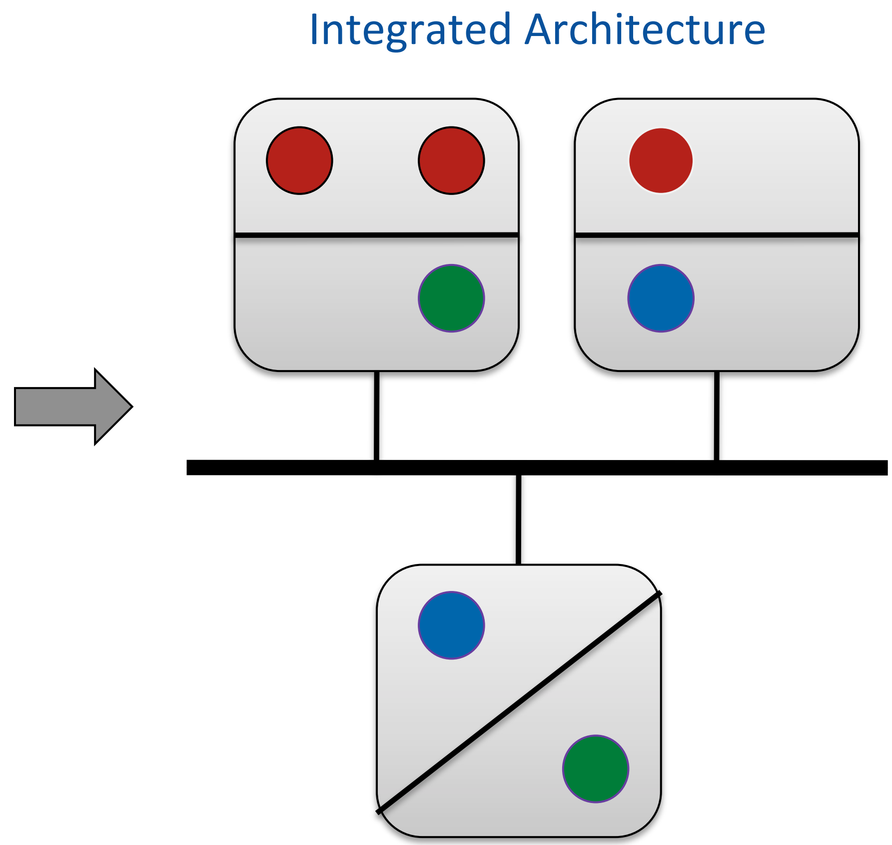
- SILs dictate the development process and certification procedures

Motivation

- Real time applications implemented using distributed systems

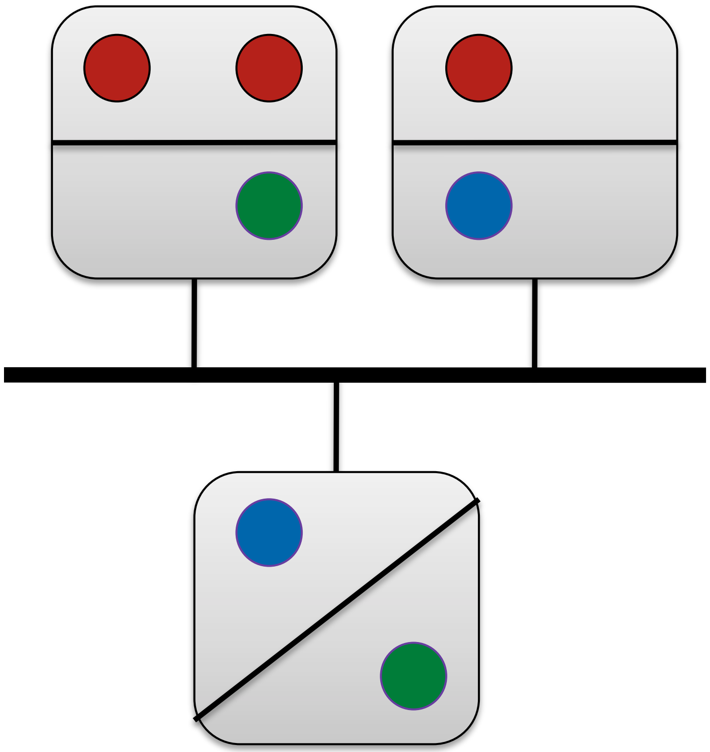
- Mixed-criticality applications share the same architecture





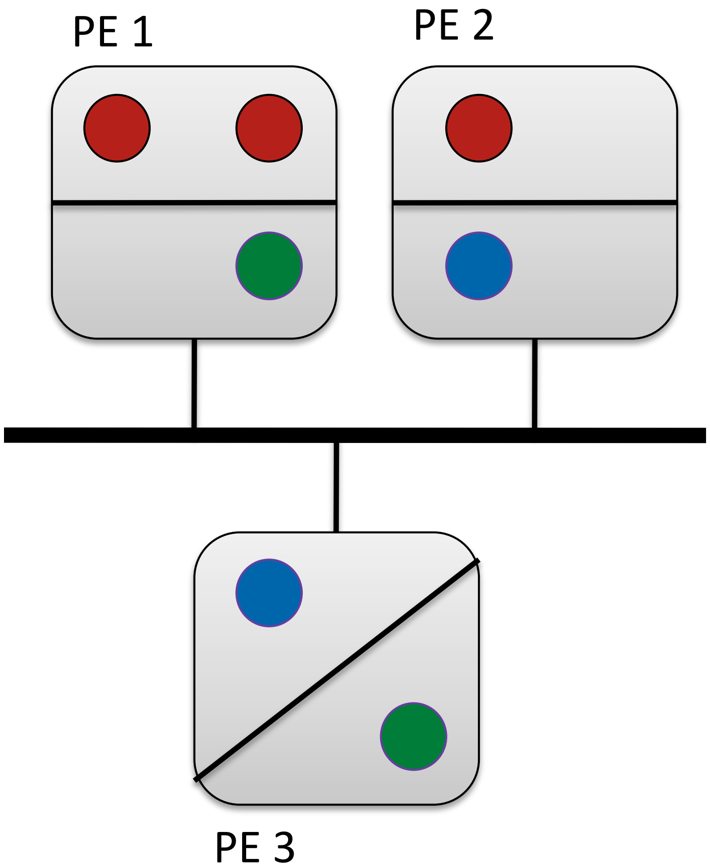
Solution: partitioned architecture

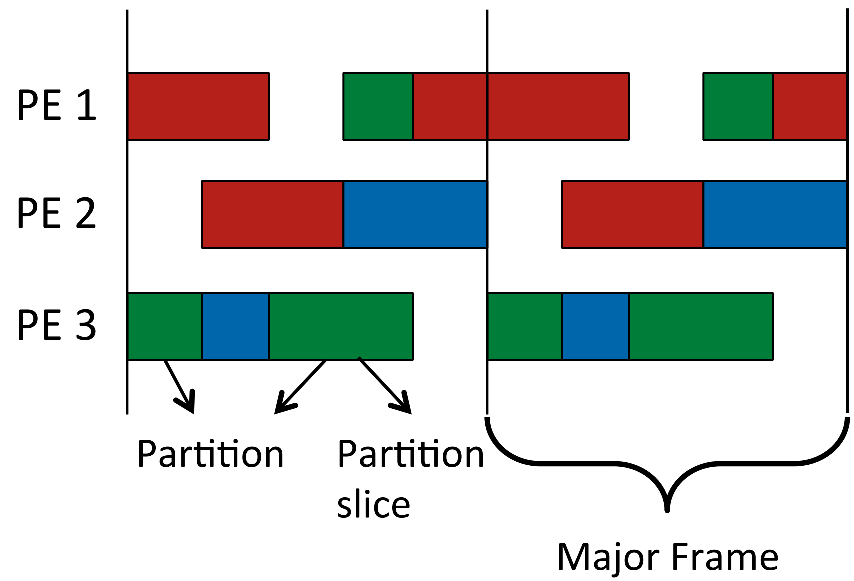
System Model



- Partition = virtual dedicated machine
- Partitioned architecture
  - Spatial partitioning
    - protects one application's memory and access to resources from another application
  - Temporal partitioning
    - partitions the CPU time among applications

System Model



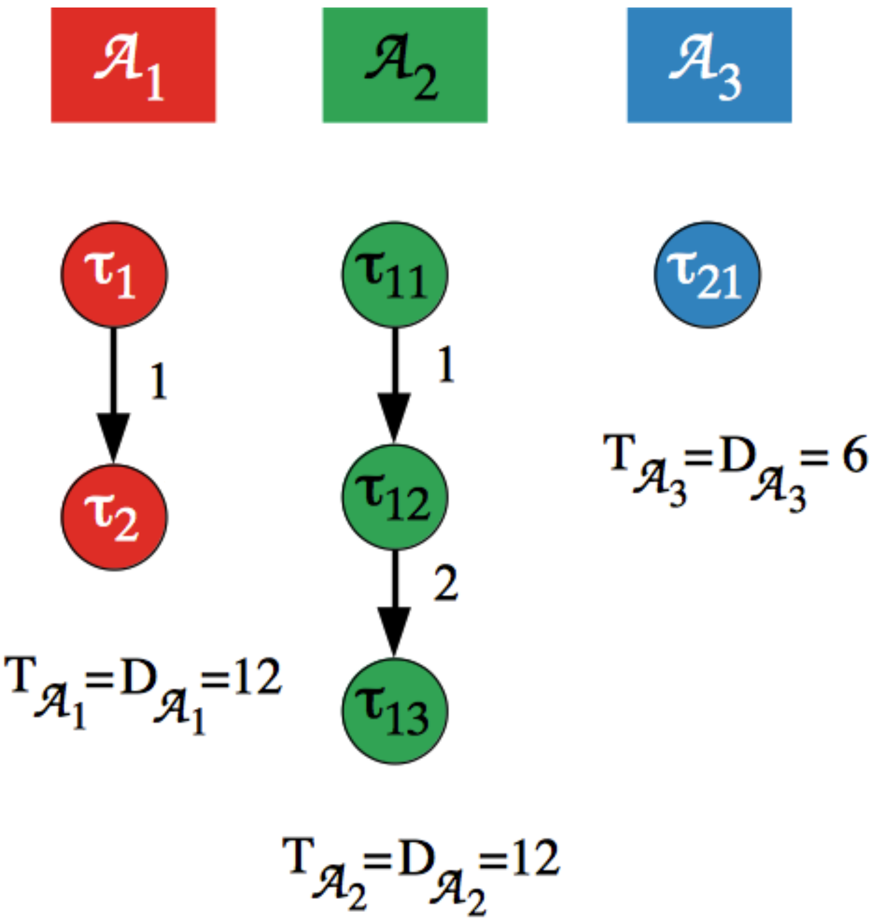


- Temporal partitioning
  - Static partiti
    - Rep … eriod MF
    - …itch overhead
    - …h partition can have its own scheduling policy
    - A partition has a certain SIL

Problem: optimize task mapping and allocation of partitions

Application Model

- Static Cyclic Scheduling



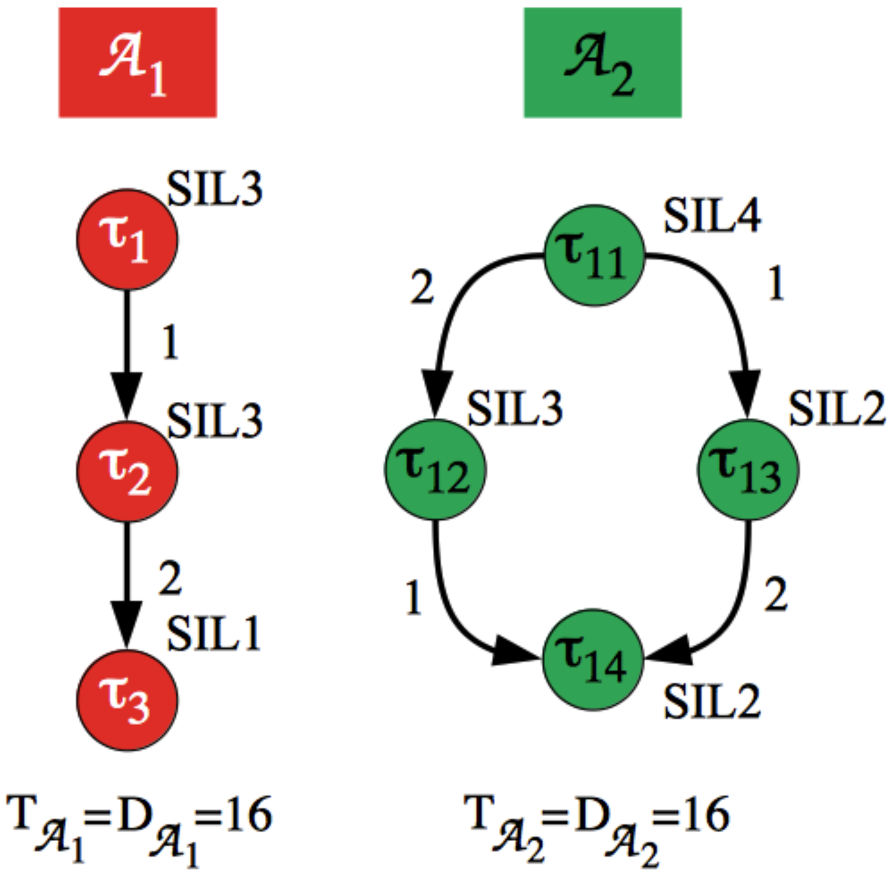
Problem formulation

- Given
  - A set of applications
  - The criticality level (or SIL) for each task
  - A set of N processing elements (PEs)
  - The size of the Major Frame and of the Application Cycle

- Determine
  - The mapping of tasks to PEs
  - The sequence and length of partition slices on each processor
  - The assignment of tasks to partitions
  - The schedule for all the tasks in the system

- Such that
  - All applications meet their deadline

Motivational Example

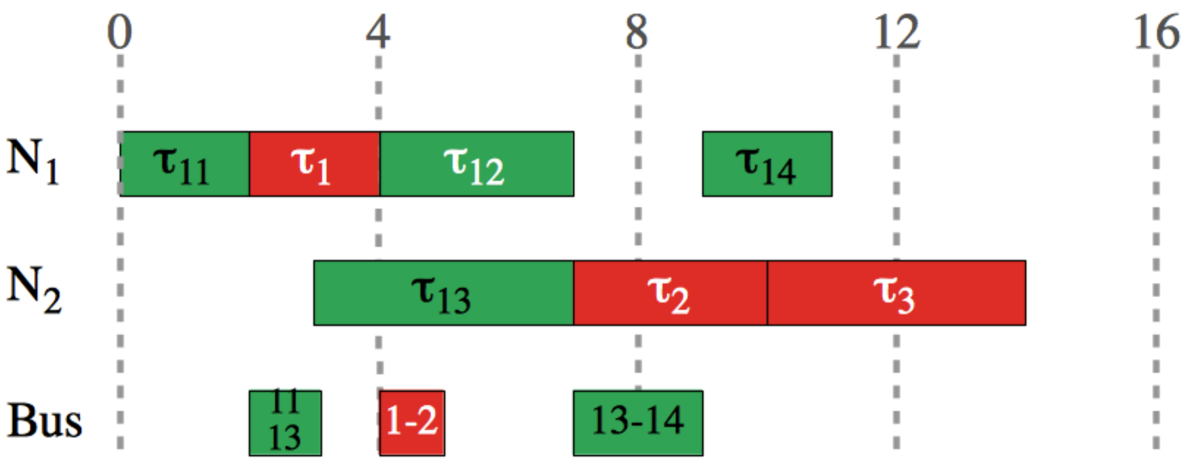


Mixed-criticality applications

		N_1	N_2
$\mathcal{A}_1$	τ_1	2	4
	τ_2	x	3
	τ_3	3	4
$\mathcal{A}_2$	τ_{11}	2	3
	τ_{12}	3	5
	τ_{13}	x	4
	τ_{14}	2	3

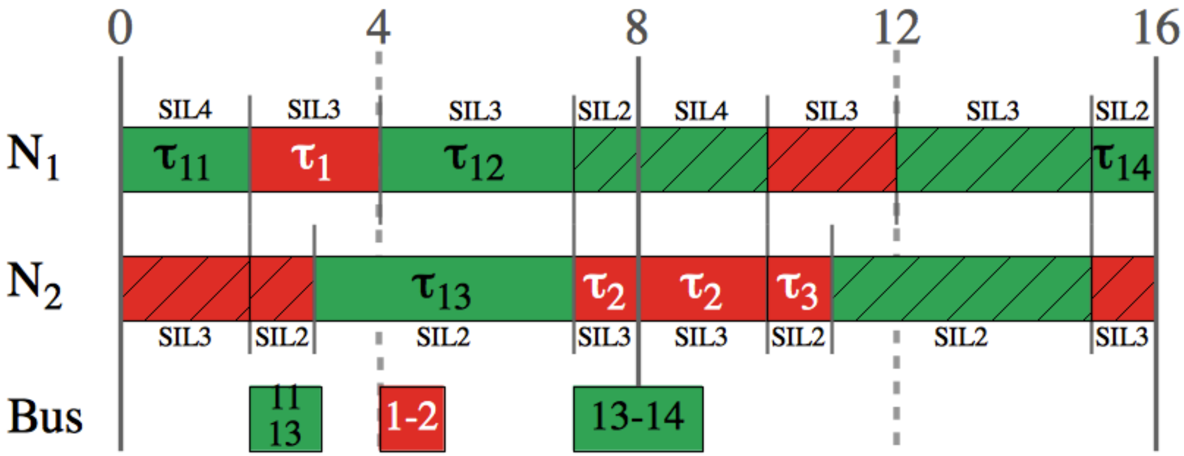
WCET and mapping restrictions

Motivational Example



Optimal mapping, without considering partitions.

Motivational Example



Partitioning, using the previously obtained mapping. τ_3 and τ_{14} miss their deadline.

Optimization Strategy

- Mapping and Time-Partitioning Optimization (MTPO) strategy:
  - Tabu Search meta-heuristic
    - The mapping of tasks to processors
    - The sequence and length of partition slices on each PE
    - The assignment of tasks to partitions
  - List scheduling
    - The schedule for the applications

- Tabu Search
  - Minimizes the cost function
  - Explores the solution space using design transformations

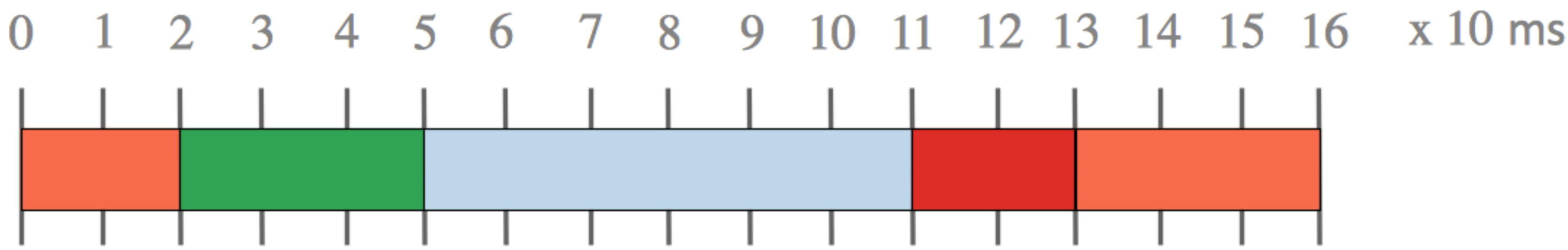
Optimization Strategy

- Degree of schedulability
  - Captures the difference between the worst-case response time and the deadline

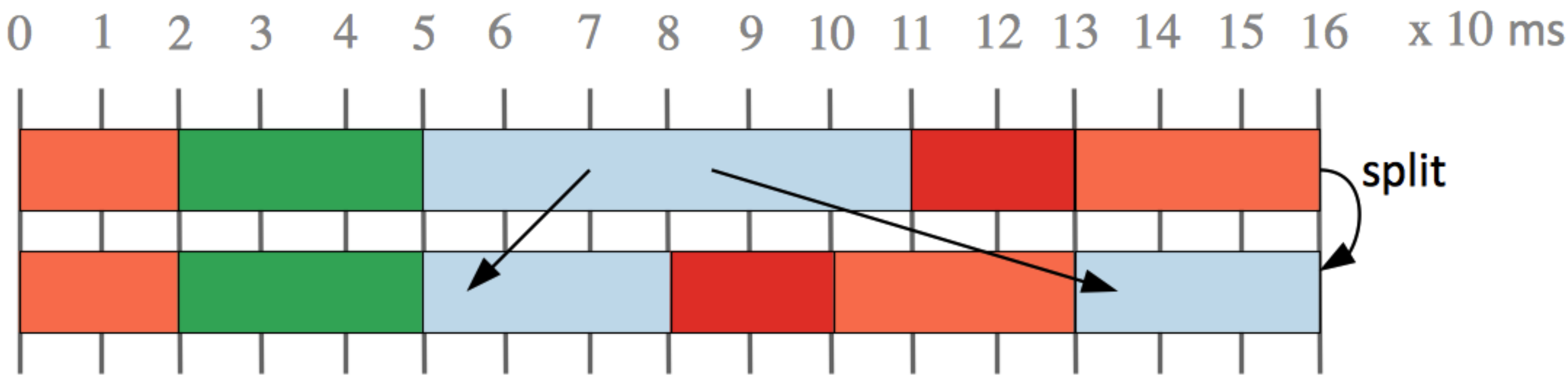
- Cost Function

$$Cost(\Psi) = \begin{cases} c_1 = \sum_{\mathcal{A}_i \in \Gamma} \max(0, R_i - D_i) & if c_1 > 0 \\ c_2 = \sum_{\mathcal{A}_i \in \Gamma} (R_i - D_i) & if c_1 = 0 \end{cases}$$

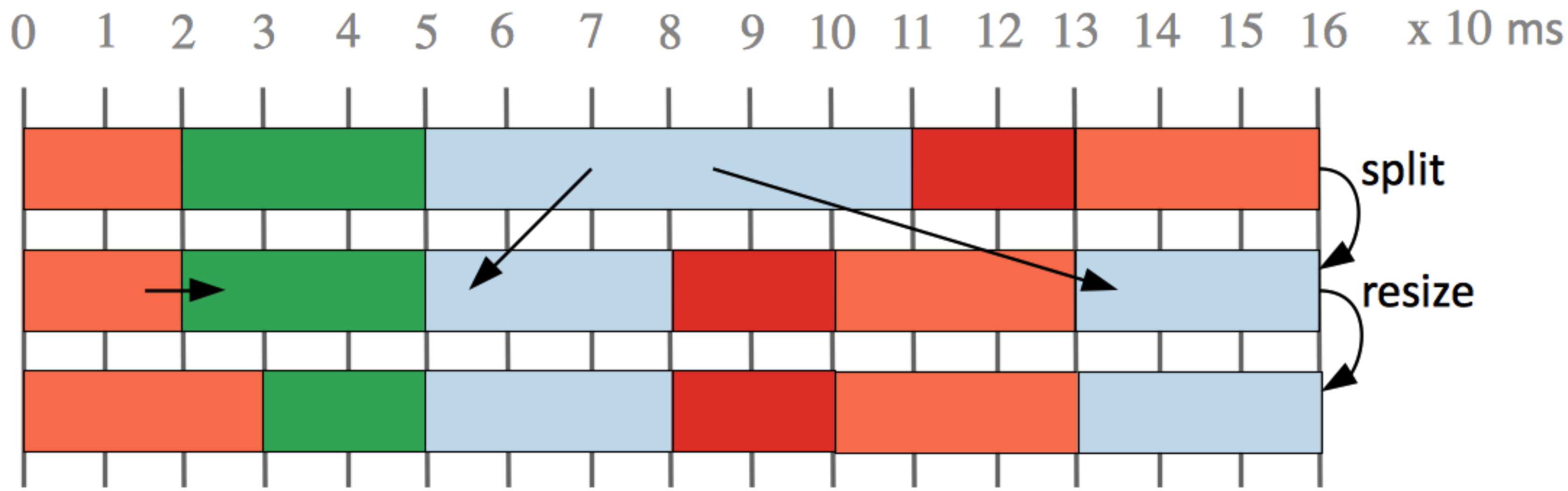
Optimization Strategy: Design Transformations



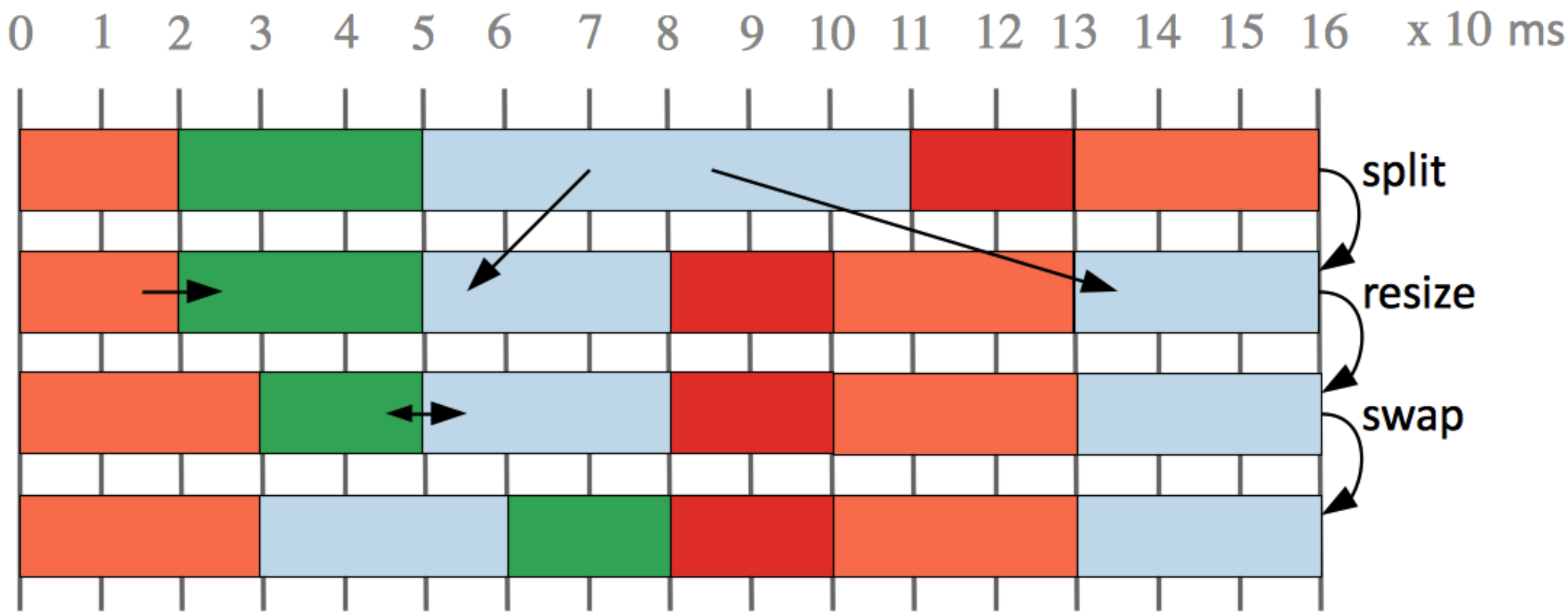
Optimization Strategy: Design Transformations



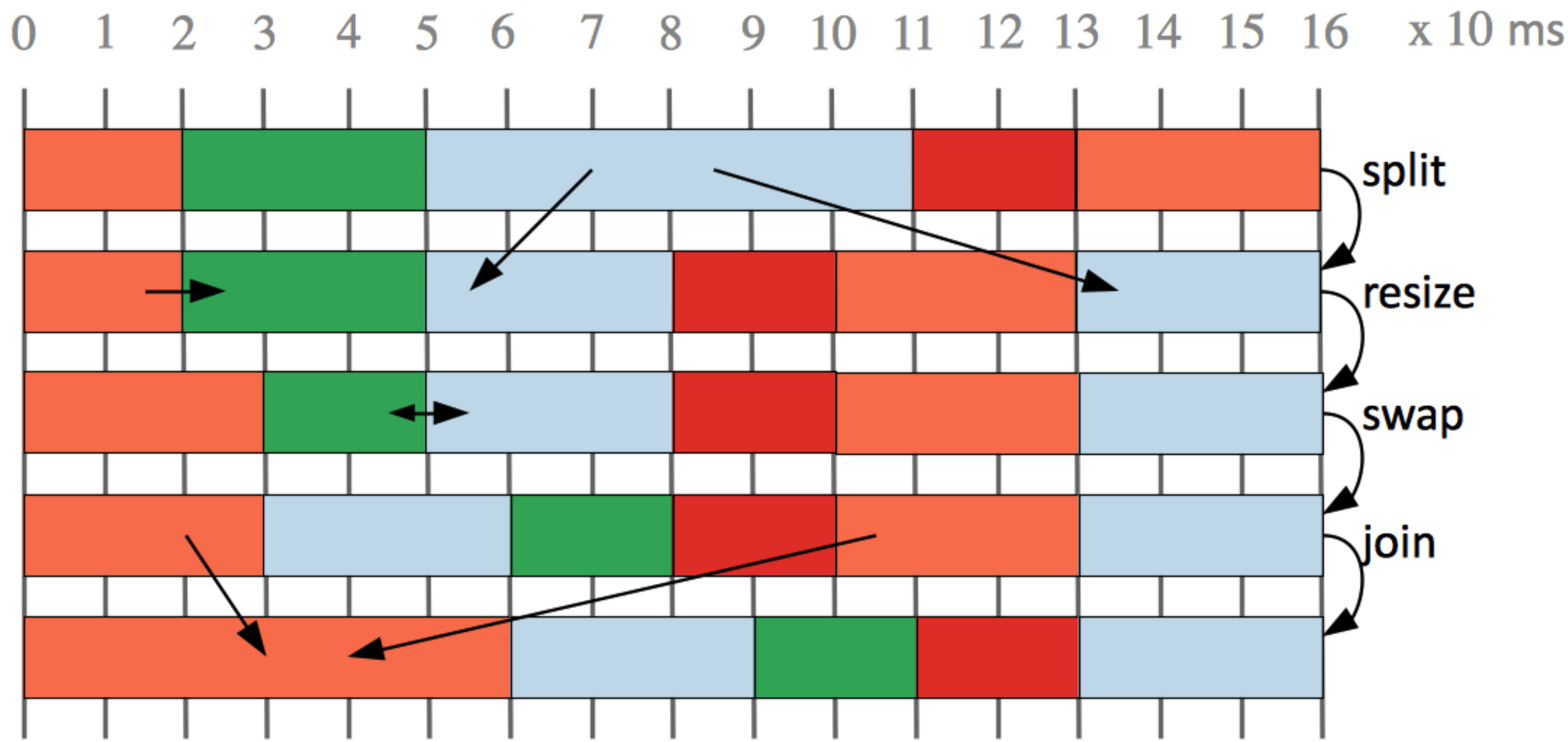
Optimization Strategy: Design Transformations



Optimization Strategy: Design Transformations



Optimization Strategy: Design Transformations



- Task re-assignment

Experimental Results

- Benchmarks
  - 5 synthetic
  - 3 real life test cases from E3S

- MTPO compared to:
  - MO+TPO
    - Optimization where first we do a mapping optimization, without considering partitioning (MO), and then we perform a partitioning optimization, considering the mapping obtained previously as fixed (TPO)

Experimental Results

Test Case	Apps	Tasks	PE	MO+TPO	MTPO	% increase
1	3	15	2	0	3	261.54
2	3	20	3	2	3	223.81
3	4	34	4	2	3	78.13
4	4	40	5	2	4	153.66
5	5	53	6	4	5	3116.67
consumer	2	12	3	2	2	19.88
networking	4	13	3	4	4	55.52
telecom	9	30	3	3	9	100.01

Conclusions

- Mixed-criticality systems, with applications of different criticalities running on the same processors, are implemented using a partitioned architecture.
- Optimizing the time partitions and the task allocation to partitions leads to schedulable solutions with improved resource utilization.
- We proposed a Tabu Search based optimization algorithm.

Thank you!

Domițian Tămaș-Selicean
dota@imm.dtu.dk