



**SAAB**

# Applying Constraint Programming for Design Space Exploration in Avionics

*Rodolfo Jordão<sup>1</sup> Tage Mohammadat<sup>1</sup> George Ungureanu<sup>1</sup> Ingemar Söderquist<sup>2</sup> Mats Ekman<sup>2</sup>  
Ingo Sander<sup>1</sup>*

<sup>1</sup>School of EECS, KTH Royal Institute of Technology, Stockholm, Sweden

<sup>2</sup>Business Area Aeronautics, Saab AB, Linköping, Sweden

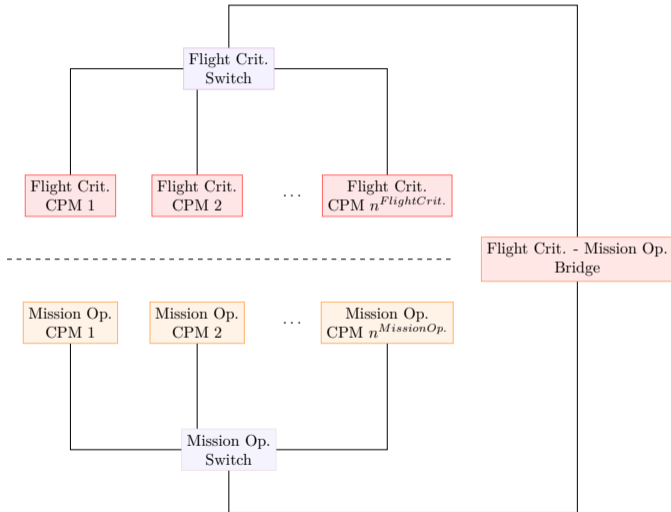
October 9, 2019

**VINNOVA**  
Sweden's innovation agency



# Motivation

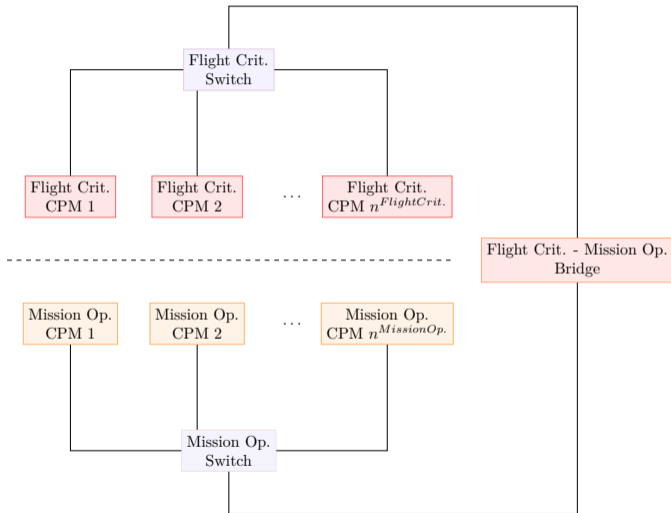
## NFFP7 CORRECT Case Study : Platform



# Motivation

## NFFP7 CORRECT Case Study : Platform

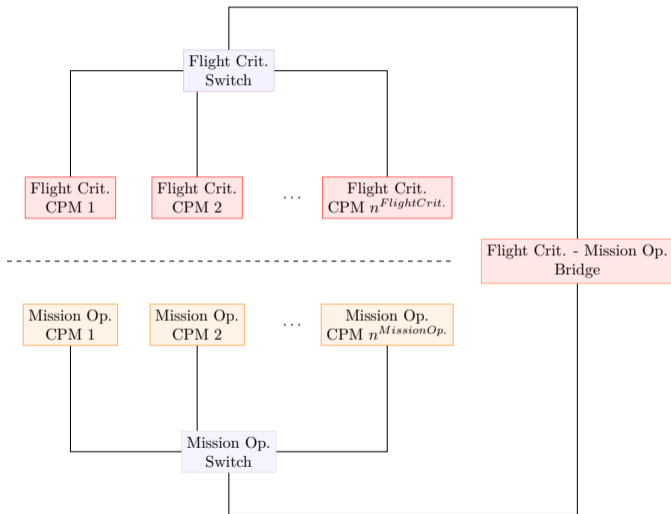
- ▶ CPM: Processing Module (CPU and Mem.)



# Motivation

## NFFP7 CORRECT Case Study : Platform

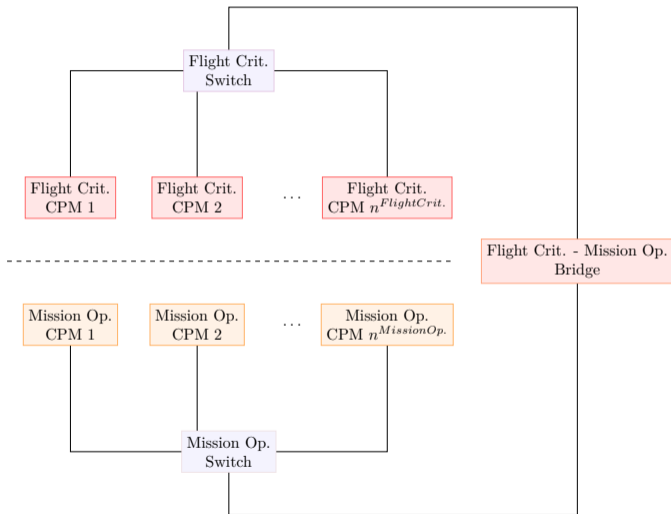
- ▶ CPM: Processing Module (CPU and Mem.)
- ▶ CPM's execute real time operating systems compliant with standard ARINC 653



# Motivation

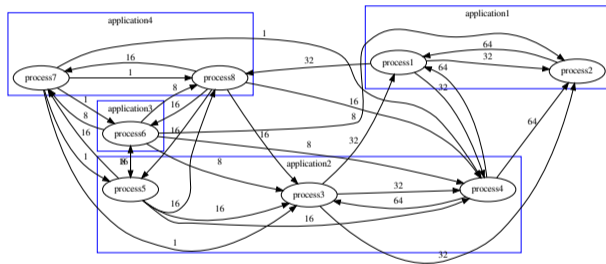
## NFFP7 CORRECT Case Study : Platform

- ▶ CPM: Processing Module (CPU and Mem.)
- ▶ CPM's execute real time operating systems compliant with standard ARINC 653
- ▶ Switch communication is predictable (Flight Crit. Based on standard SAE AS6003 and Mission Op. Based on standard ARINC 664)



# Motivation

## NFFP7 CORRECT Case Study : Applications I

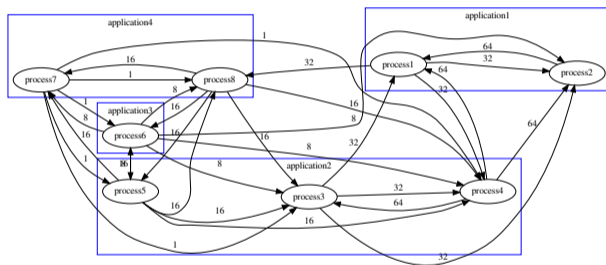




# Motivation

## NFFP7 CORRECT Case Study : Applications I

- ▶ Independent periodic tasks with communication
- ▶ Different applications may need to be spatially isolated depending upon criticalities





# Motivation

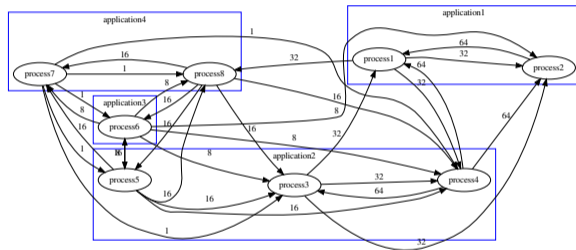
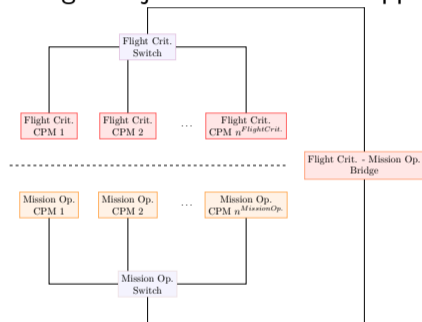
## NFFP7 CORRECT Case Study : Sub-problem

- ▶ Designer's job: decide where applications go to satisfy specifications.

# Motivation

## NFFP7 CORRECT Case Study : Sub-problem

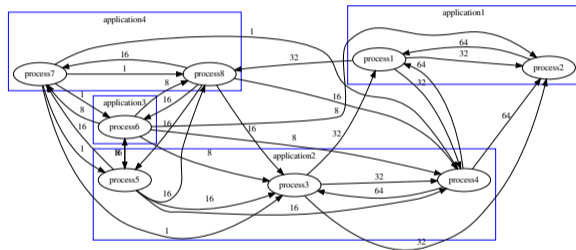
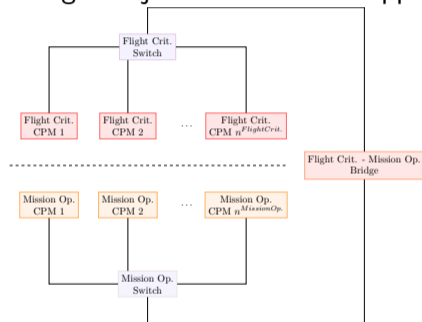
- Designer's job: decide where applications go to satisfy specifications.



# Motivation

## NFFP7 CORRECT Case Study : Sub-problem

- ▶ Designer's job: decide where applications go to satisfy specifications.

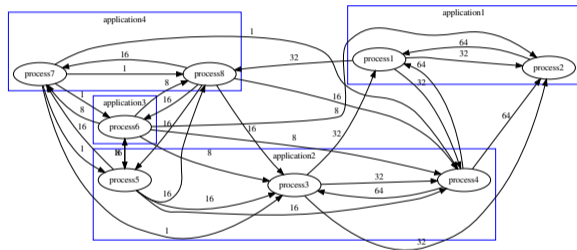
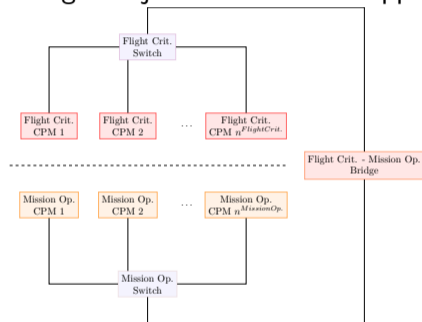


- ▶ The set of the possibilities is called the Design Space.

# Motivation

## NFFP7 CORRECT Case Study : Sub-problem

- ▶ Designer's job: decide where applications go to satisfy specifications.



- ▶ The set of the possibilities is called the Design Space.
- ▶ Design Space Exploration (DSE) is the process of finding the decisions that meet the specifications.

# Motivation

## NFFP7 CORRECT Case Study : Applications II

- ▶ How hard can DSE get?

# Motivation

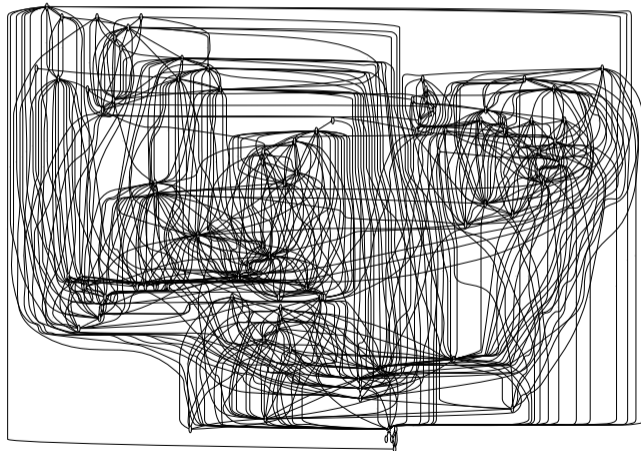
## NFFP7 CORRECT Case Study : Applications II

- ▶ How hard can DSE get? Check the Flight Critical part

# Motivation

## NFFP7 CORRECT Case Study : Applications II

- ▶ How hard can DSE get? Check the Flight Critical part (91 processes)



# Motivation

NFFP7 CORRECT Case Study : Total problem

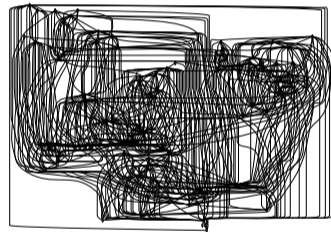
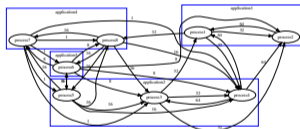
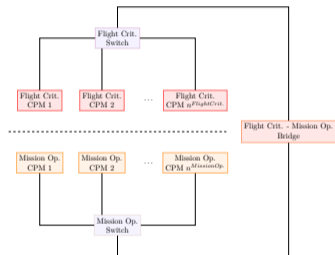
- ▶ Current total DSE problem:



# Motivation

## NFFP7 CORRECT Case Study : Total problem

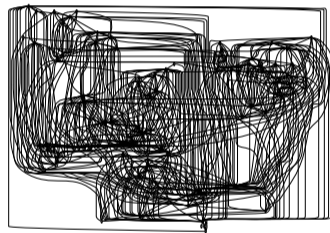
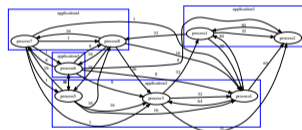
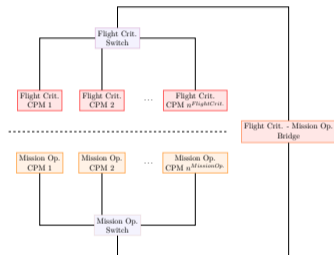
### ► Current total DSE problem:



# Motivation

## NFFP7 CORRECT Case Study : Total problem

- ▶ Current total DSE problem:

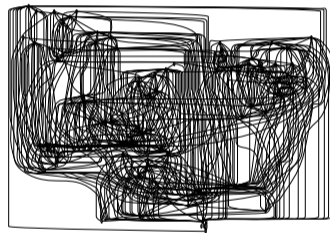
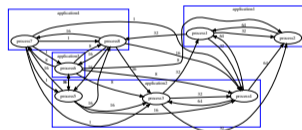
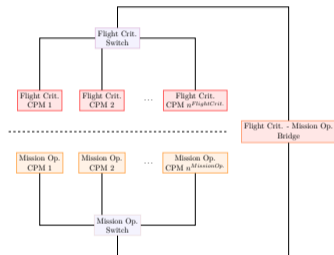


- ▶ Find a mapping that meets all deadlines and respects spatial constraints.

# Motivation

## NFFP7 CORRECT Case Study : Total problem

- ▶ Current total DSE problem:



- ▶ Find a mapping that meets all deadlines and respects spatial constraints.
- ▶ Also try to minimize the number of CPMs used!

# Motivation

NFFP7 CORRECT Case Study : To consider

- ▶ ARINC 653 Partitions

# Motivation

NFFP7 CORRECT Case Study : To consider

- ▶ ARINC 653 Partitions
- ▶ Communications between Flight Critical and Mission Operational parts

# Motivation

NFFP7 CORRECT Case Study : To consider

- ▶ ARINC 653 Partitions
- ▶ Communications between Flight Critical and Mission Operational parts
- ▶ Temporal Isolation of applications

# Motivation

NFFP7 CORRECT Case Study : To consider

- ▶ ARINC 653 Partitions
- ▶ Communications between Flight Critical and Mission Operational parts
- ▶ Temporal Isolation of applications
- ▶ ...

# Motivation

NFFP7 CORRECT Case Study : To consider

- ▶ ARINC 653 Partitions
- ▶ Communications between Flight Critical and Mission Operational parts
- ▶ Temporal Isolation of applications
- ▶ ...
  
- ▶ Complexity of the DSE problem ever increasing!



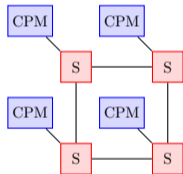
# Motivation

NFFP7 CORRECT Case Study : To consider

- ▶ ARINC 653 Partitions
- ▶ Communications between Flight Critical and Mission Operational parts
- ▶ Temporal Isolation of applications
- ▶ ...
  
- ▶ Complexity of the DSE problem ever increasing!
- ▶ ...but the complexity and scale of the problems can be lowered with automated DSE methods.

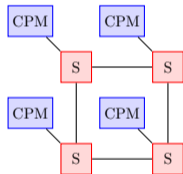
# Putting it together

Predictable platform  
(Execution times, comm. times, ...)

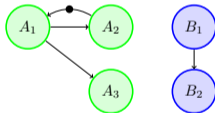


# Putting it together

Predictable platform  
(Execution times, comm. times, ...)



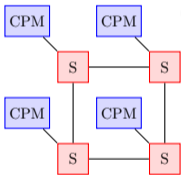
Analyzable applications  
(Termination guarantess, buffers, ...)



# Putting it together

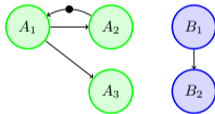
Predictable platform

(Execution times, comm. times, ...)



Analyzable applications

(Termination guarantess, buffers, ...)

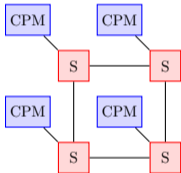


Local and global design constraints

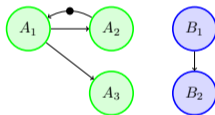


# Putting it together

Predictable platform  
(Execution times, comm. times, ...)



Analyzable applications  
(Termination guarantess, buffers, ...)



Local and global design constraints



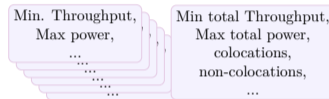
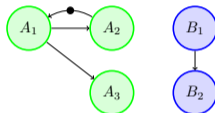
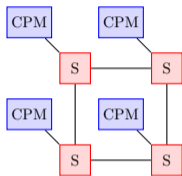
Exploration of map and schedules of all applications in the platform respecting design constraints and applications behaviours.  
Optionally with optimization goals.

# Putting it together

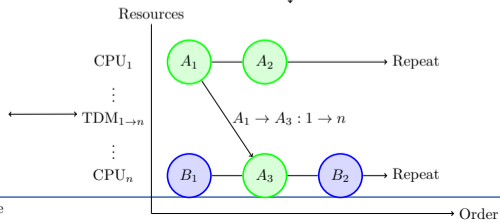
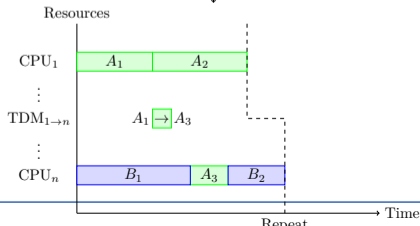
Predictable platform  
(Execution times, comm. times, ...)

Analyzable applications  
(Termination guarantess, buffers, ...)

Local and global design constraints



Exploration of map and schedules of all applications in the platform respecting design constraints and applications behaviours.  
Optionally with optimization goals.



# Constraint Programming

Why choose it?

- ▶ Additional specifications directly reflect additional variables and constraints.

# Constraint Programming

Why choose it?

- ▶ Additional specifications directly reflect additional variables and constraints.

Example:

1. Two processes  $P_1$  and  $P_2$  cannot be co-located.
2. Variables  $map(P_1)$  and  $map(P_2)$  represent their locations.
3. Then  $map(P_1) \neq map(P_2)$  reflects the co-location constraint.



# Constraint Programming

Why choose it?

- ▶ Additional specifications directly reflect additional variables and constraints.  
Example:
  1. Two processes  $P_1$  and  $P_2$  cannot be co-located.
  2. Variables  $map(P_1)$  and  $map(P_2)$  represent their locations.
  3. Then  $map(P_1) \neq map(P_2)$  reflects the co-location constraint.
- ▶ The problem description is separated from the solution method: different solvers may be used.

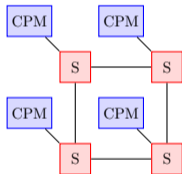
# Constraint Programming

## Why choose it?

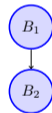
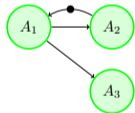
- ▶ Additional specifications directly reflect additional variables and constraints.  
Example:
  1. Two processes  $P_1$  and  $P_2$  cannot be co-located.
  2. Variables  $map(P_1)$  and  $map(P_2)$  represent their locations.
  3. Then  $map(P_1) \neq map(P_2)$  reflects the co-location constraint.
- ▶ The problem description is separated from the solution method: different solvers may be used.
- ▶ Constraint programming solution frameworks are complete: guarantees of feasibility and optimality available.

# Putting it together

Predictable platform  
(Execution times, comm. times, ...)

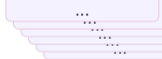


Analyzable applications  
(Termination guarantess, buffers, ...)



Local and global design constraints

Min. Throughput,  
Max power,  
...



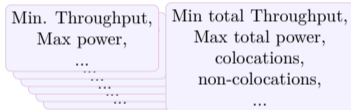
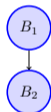
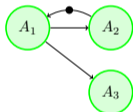
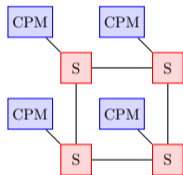
Min total Throughput,  
Max total power,  
colocations,  
non-colocations,  
...

# Putting it together

Predictable platform  
(Execution times, comm. times, ...)

Analyzable applications  
(Termination guarantess, buffers, ...)

Local and global design constraints

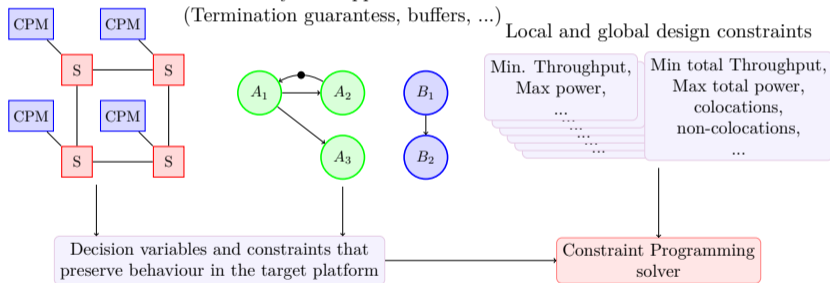


Decision variables and constraints that preserve behaviour in the target platform

# Putting it together

Predictable platform  
(Execution times, comm. times, ...)

Analyzable applications  
(Termination guarantess, buffers, ...)

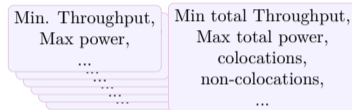
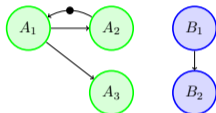
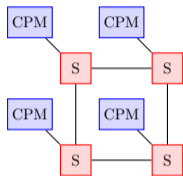


# Putting it together

Predictable platform  
(Execution times, comm. times, ...)

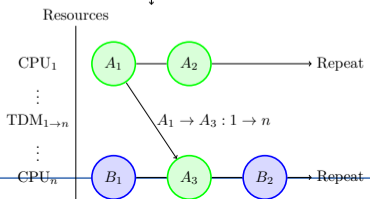
Analyzable applications  
(Termination guarantess, buffers, ...)

Local and global design constraints



Decision variables and constraints that preserve behaviour in the target platform

Constraint Programming solver

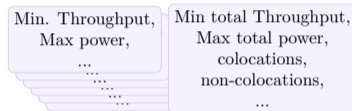
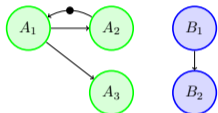
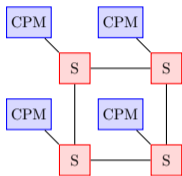


# Putting it together

Predictable platform  
(Execution times, comm. times, ...)

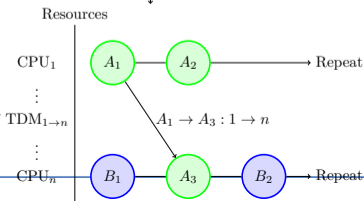
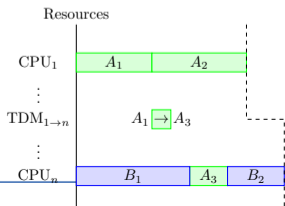
Analyzable applications  
(Termination guarantess, buffers, ...)

Local and global design constraints



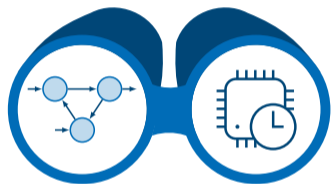
Decision variables and constraints that preserve behaviour in the target platform

Constraint Programming solver



# Putting it together

DeSyDe



ForSyDe's logo

- ▶ This DSE flow is implemented in DeSyDe<sup>1</sup>, the DSE tool of the ForSyDe<sup>2</sup> methodology and tooling ecosystem.

---

<sup>1</sup><https://github.com/forsyde/DeSyDe>

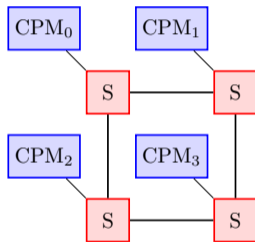
<sup>2</sup><https://forsyde.github.io/>



# DSE example

## Test Case I

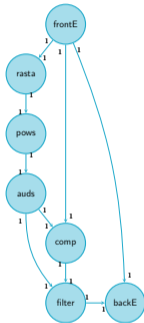
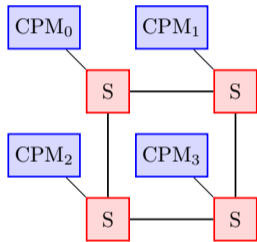
- ▶ Given the platform,



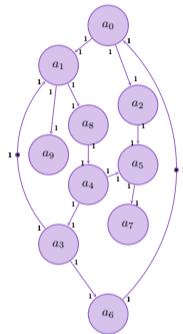
# DSE example

## Test Case I

- ▶ Given the platform, applications



RASTA-PLP

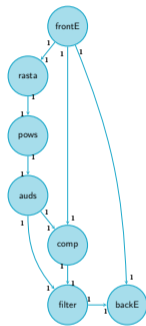
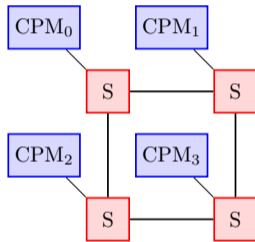


Synthetic Cyclic Graph

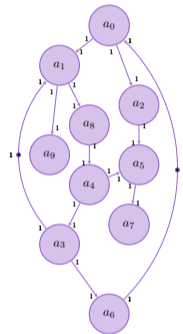
# DSE example

## Test Case I

- ▶ Given the platform, applications and the constraints



RASTA-PLP



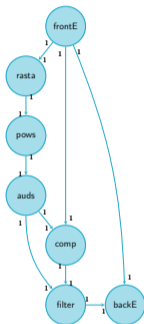
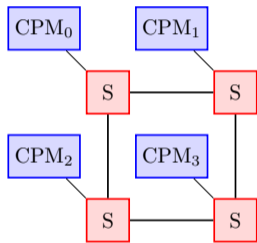
Synthetic Cyclic Graph

- ▶ Rasta and Synth must both execute an actor in parallel at least once.

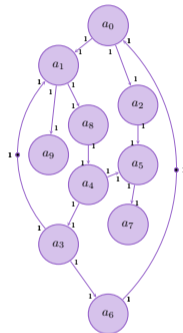
# DSE example

## Test Case I

- ▶ Given the platform, applications and the constraints



RASTA-PLP



Synthetic Cyclic Graph

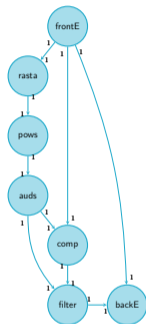
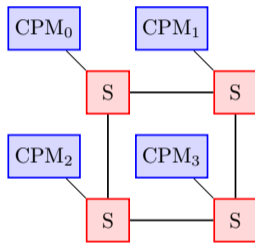
- ▶ Rasta and Synth must both execute an actor in parallel at least once.

- ▶ Find a mapping and schedule that minimizes power.

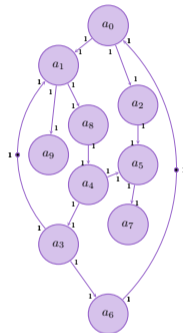
# DSE example

## Test Case I

- ▶ Given the platform, applications and the constraints



RASTA-PLP



Synthetic Cyclic Graph

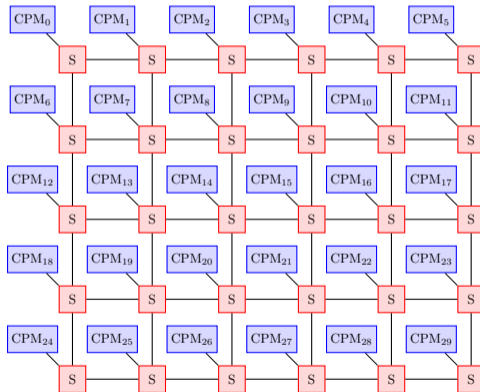
- ▶ Rasta and Synth must both execute an actor in parallel at least once.

- ▶ Find a mapping and schedule that minimizes power.
- ▶ DeSyDe finds a mapping and schedule using 3 CPMs after 10 minutes and proves its optimality after 44 minutes.

# DSE example

## Test Case II - Platform

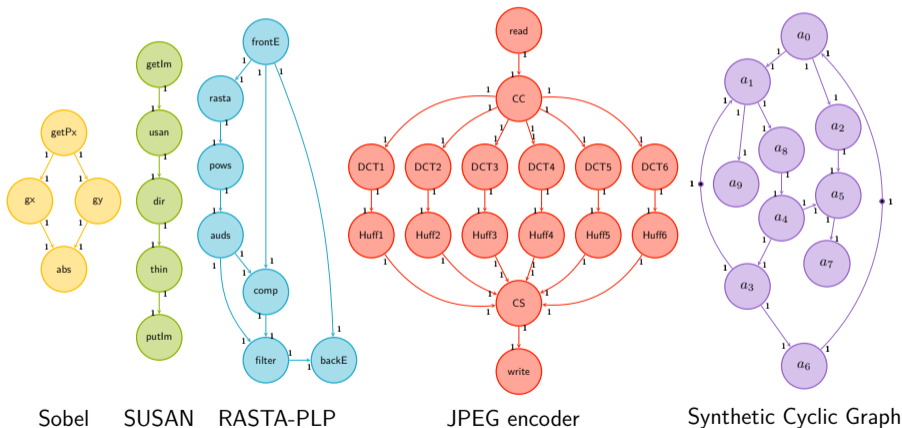
- ▶ 5 by 6 predictable NoC Platform.



# DSE example

## Test Case II - Applications

- ▶ 5 streaming applications.



# DSE example

## Test Case II

- ▶ There are no local or global constraints.



# DSE example

## Test Case II

- ▶ There are no local or global constraints.
- ▶ The goal is to minimize power.

# DSE example

## Test Case II

- ▶ There are no local or global constraints.
- ▶ The goal is to minimize power.
- ▶ ...After 4 hours DeSyDe does not find any solution:

# DSE example

## Test Case II

- ▶ There are no local or global constraints.
- ▶ The goal is to minimize power.
- ▶ ...After 4 hours DeSyDe does not find any solution:
- ▶ design space is too big! Many degrees of freedom to explore.

# DSE example

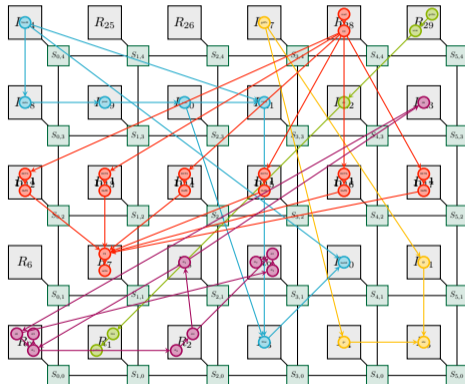
## Test Case II - Workaround

- ▶ Workaround: incorporate designer's knowledge via a given mapping.

# DSE example

## Test Case II - Workaround

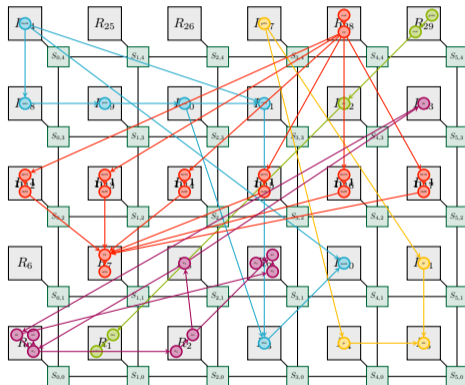
- ▶ Workaround: incorporate designer's knowledge via a given mapping.



# DSE example

## Test Case II - Workaround

- ▶ Workaround: incorporate designer's knowledge via a given mapping.



- ▶ DeSyDe now finds a schedule after 5 seconds and prove its optimality after 20 minutes.

## Ongoing research and Future plans

- ▶ Use the knowledge gained with streaming application models to reach DSE problems with periodic tasks formulations, akin to the motivation.
- ▶ Incorporating designer's knowledge in DeSyDe as smart searches, other than fixed mappings.
- ▶ Use virtualization (partitioning) concepts to mix different application models into the same platform.
- ▶ Incorporate platform exploration into the DSE framework.

## References



K. Rosvall, T. Mohammadat, G. Ungureanu, J. Öberg, and I. Sander.  
Exploring Power and Throughput for Dataflow Applications on Predictable NoC Multiprocessors.

*In 2018 21st Euromicro Conference on Digital System Design (DSD)*, pages 719–726, August 2018.



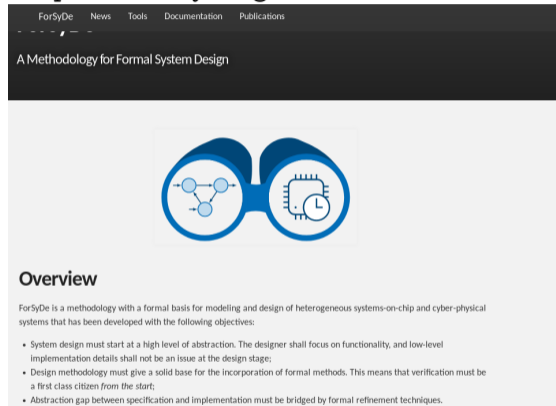
Kathrin Rosvall and Ingo Sander.

Flexible and Tradeoff-Aware Constraint-Based Design Space Exploration for Streaming Applications on Heterogeneous Platforms.

*ACM Trans. Des. Autom. Electron. Syst.*, 23(2):21:1–21:26, November 2017.



(much) more in the ForSyDe webpage!  
<https://forsyde.github.io/>




The screenshot shows the top navigation bar of the ForSyDe website with links for ForSyDe, News, Tools, Documentation, and Publications. Below the navigation bar is a dark header with the text "A Methodology for Formal System Design". The main content area features a logo consisting of a pair of blue glasses. The left lens contains a state transition diagram with three nodes and arrows, while the right lens contains a microchip icon with a clock symbol. Below the logo is the section title "Overview" and a paragraph describing the methodology's objectives, followed by a bulleted list of three key objectives.

ForSyDe

News Tools Documentation Publications

A Methodology for Formal System Design



### Overview

ForSyDe is a methodology with a formal basis for modeling and design of heterogeneous systems-on-chip and cyber-physical systems that has been developed with the following objectives:

- System design must start at a high level of abstraction. The designer shall focus on functionality, and low-level implementation details shall not be an issue at the design stage;
- Design methodology must give a solid base for the incorporation of formal methods. This means that verification must be a first class citizen *from the start*;
- Abstraction gap between specification and implementation must be bridged by formal refinement techniques.

Thank you!