

MOVES



A Tool for Modeling and Verification of Embedded Systems

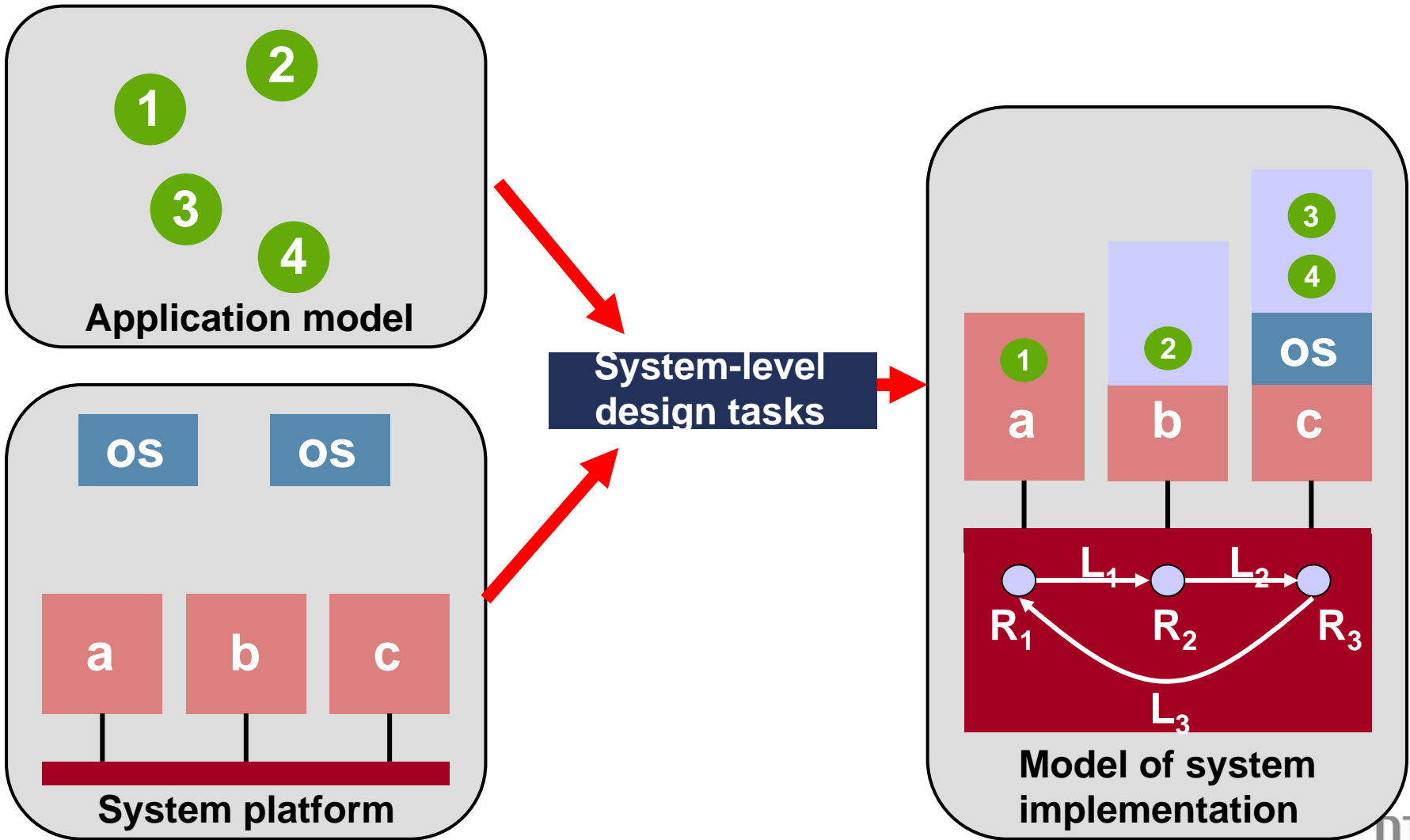
**Jan Madsen, Michael R. Hansen
Aske Brekling, Jens Ellebæk, Kristian S. Knudsen**

Embedded Systems Engineering Group

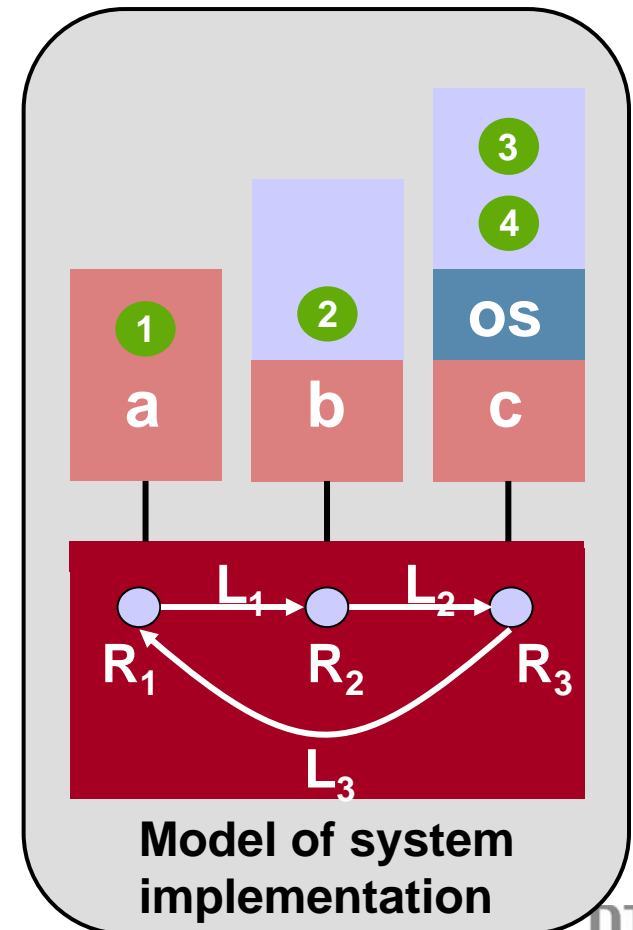
Informatics and Mathematical Modeling
Technical University of Denmark



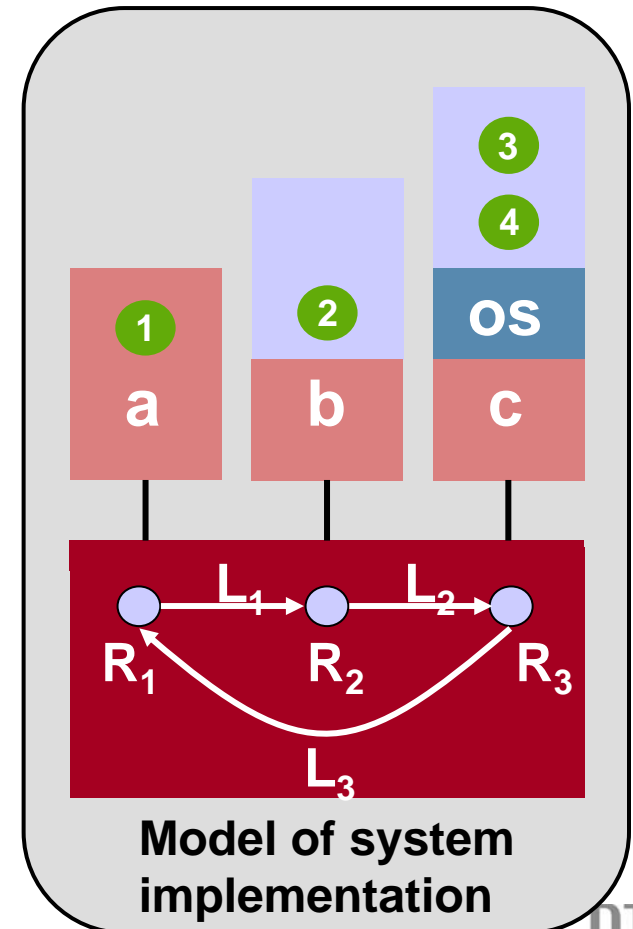
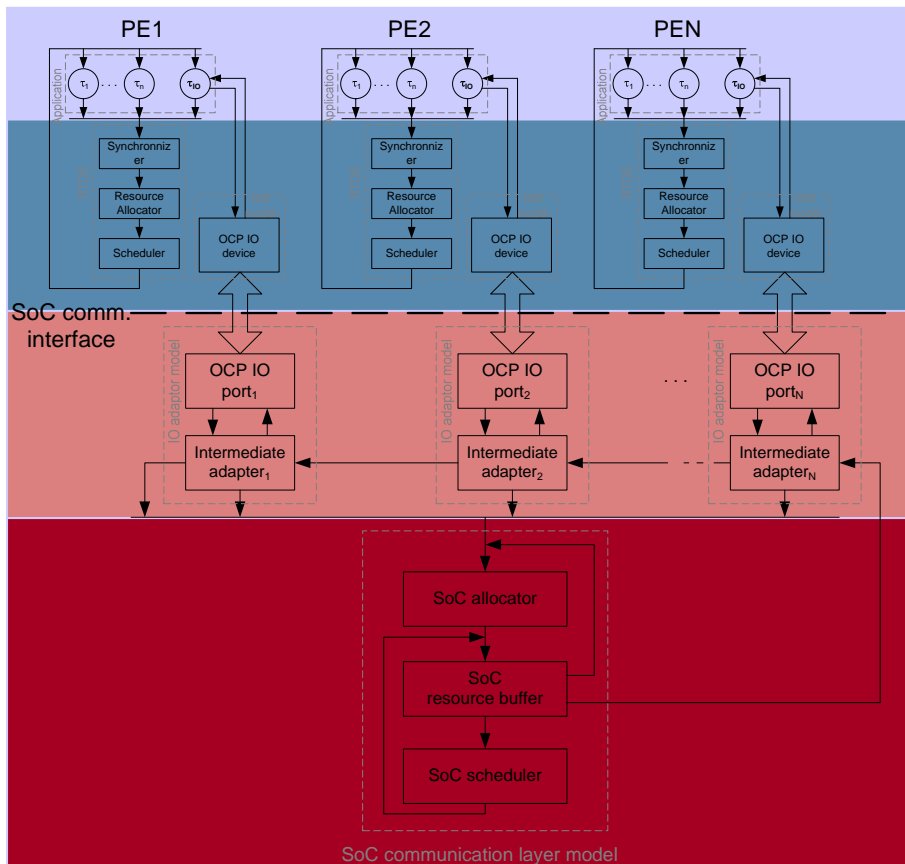
Motivation

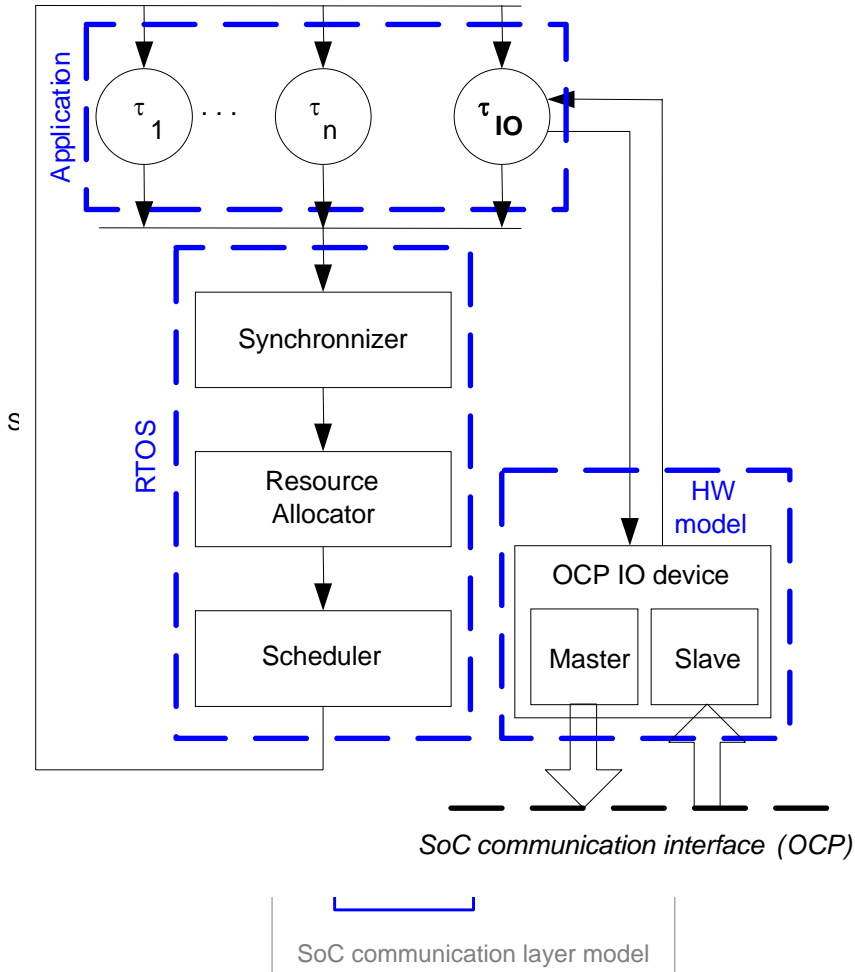


- The **ARTS** framework
- Using **ARTS**
- A TA model based on **ARTS**
- **MOVES**
- Demo of **MOVES**



ARTS model in SystemC





- **ARTS** Simulation framework based on SystemC
- **ARTS** PE module:
 - Application
 - OS
 - IO ports (OCP 2.0 interface)
 - IO device drivers
- **ARTS** Communication module:
 - Network topology and protocol
 - Network adapters
 - IO ports (OCP 2.0 interface)
- Applications of **ARTS**:
 - MPSoC (NoC exploration)
 - Wireless sensor networks
 - Automotive systems (TT vs. ET)
 - Dynamic reconfiguration

- System-level modeling framework
 - Bridging,
 - Application
 - RTOS
 - Execution platform
 - Processing elements
 - NoC
 - Supporting
 - System-level analysis
 - Early design space exploration
- Cross-layer optimization*

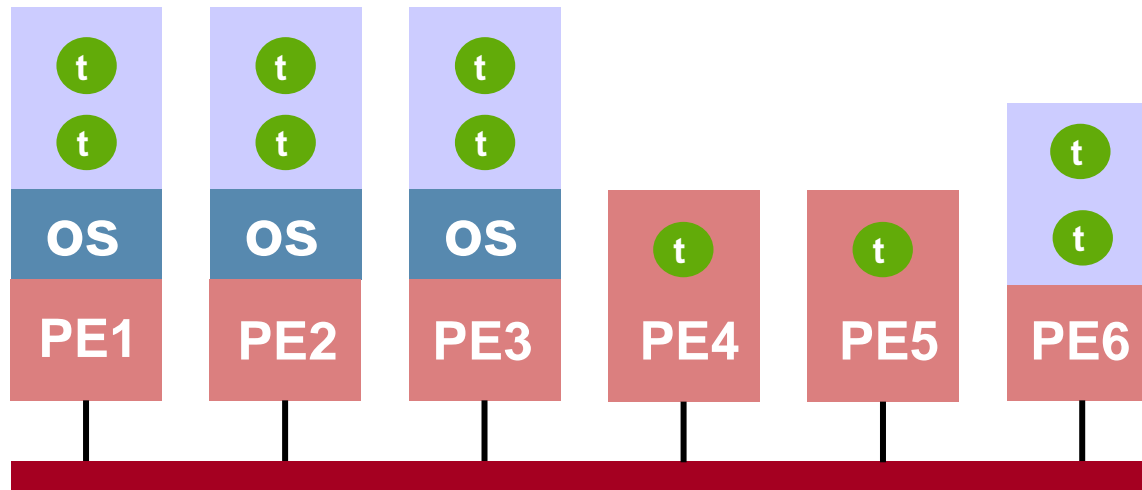
- Consequences of
 - **RTOS** selection
 - scheduling, synchronization and resource allocation policies
 - **Processor** selection
 - General purpose, semicustom or dedicated hardware
 - **Network** selection
 - topologies and communication protocols.
 - **Task mapping** to processors
 - software or hardware
- On
 - Performance
 - Power
 - Memory profile
 - Area
 - ...

Components



| <i>PE</i> type | freq. (MHz) | cost (\$) | Mem/chip size (Bytes)/(μ^2 or TLBs) | static power (μ W) | comm. power (μ W) |
|--|----------------|--------------|---|----------------------------|---------------------------|
| General Purpose Processors (ARM Family): | | | | | |
| <i>PE_{GPP0}</i> | 25.0 | 100 | 1024000 Bytes | 800 | 2000 |
| <i>PE_{GPP1}</i> | 10.0 | 50 | 1024000 Bytes | 800 | 2000 |
| <i>PE_{GPP2}</i> | 6.6 | 50 | 1024000 Bytes | 800 | 18000 |
| <i>PE_{GPP3}</i> | 10.0 | 50 | 1024000 Bytes | 880 | 2000 |
| ASIC (0.35 μ): | | | | | |
| <i>PE_{ASIC0}</i> | 2.5 | 400 | 18374 μ^2 | 10 | 21 |
| <i>PE_{ASIC1}</i> | 2.5 | 300 | 50000 μ^2 | 18 | 37 |
| <i>PE_{ASIC2}</i> | 2.5 | 500 | 24274 μ^2 | 10 | 21 |
| <i>PE_{ASIC3}</i> | 2.5 | 300 | 50000 μ^2 | 20 | 37 |
| FPGA (Xilinx Family) | | | | | |
| <i>PE_{FPGA}</i> | 2.5 | 250 | 1220 TLBs | 127 | 42.79 |
| Communication links (buses) | | | | | |
| <i>l_{BUS0}</i> | 66.0 | 65 | NA | 775 | 4881.648 |
| <i>l_{BUS1}</i> | 33.0 | 48 | NA | 941 | 4781.648 |

Execution platform



| PE | GPP0 | GPP1 | GPP2 | FPGA | ASIC |
|----|------|------|------|------|------|
| | 25 | 10 | 6.6 | 2.5 | 2.5 |

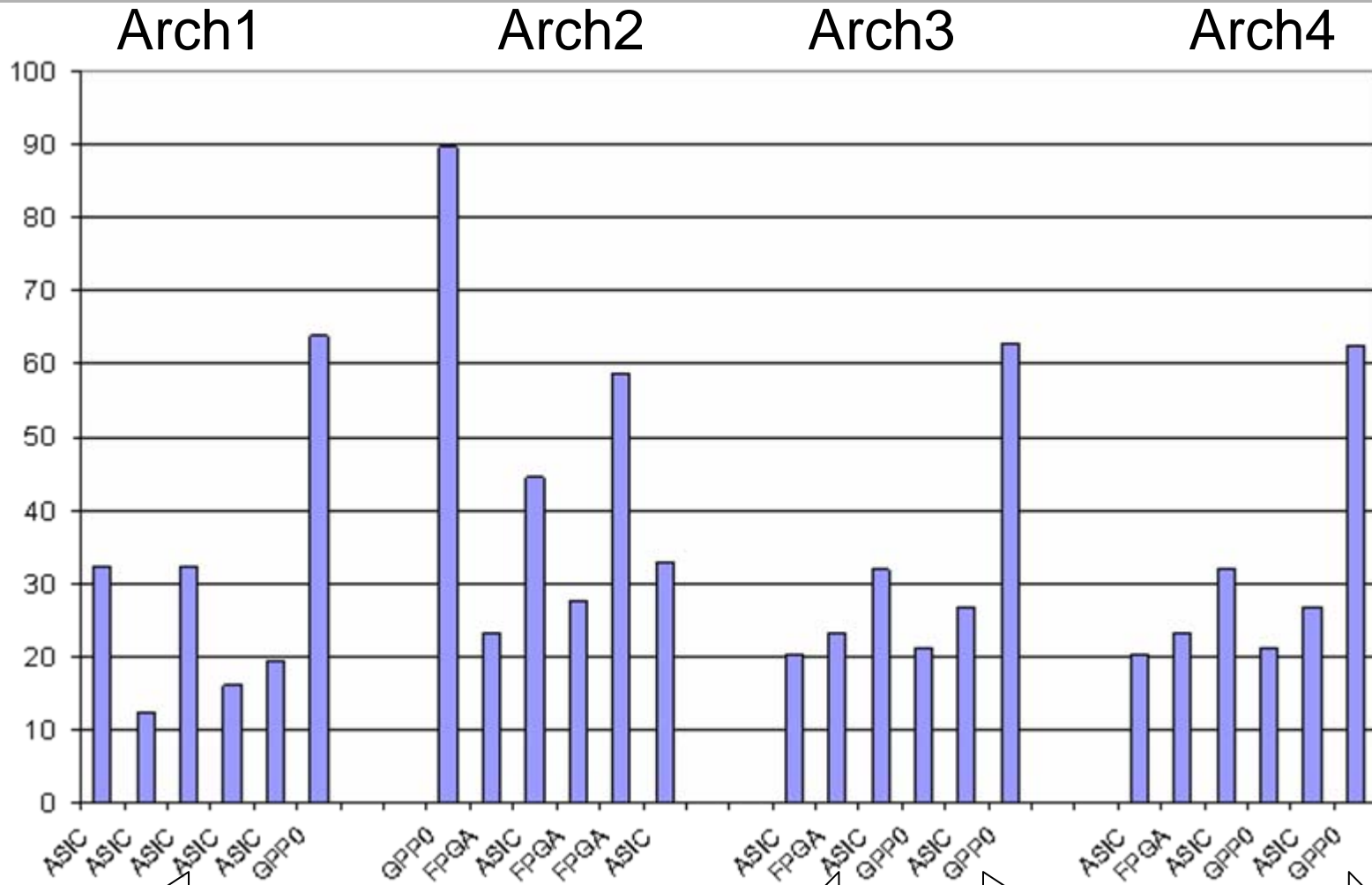
| OS | RM (Rate Monotonic) | EDF (Early Deadline First) |
|----|---------------------|----------------------------|
| | | |

Case study: Architectures



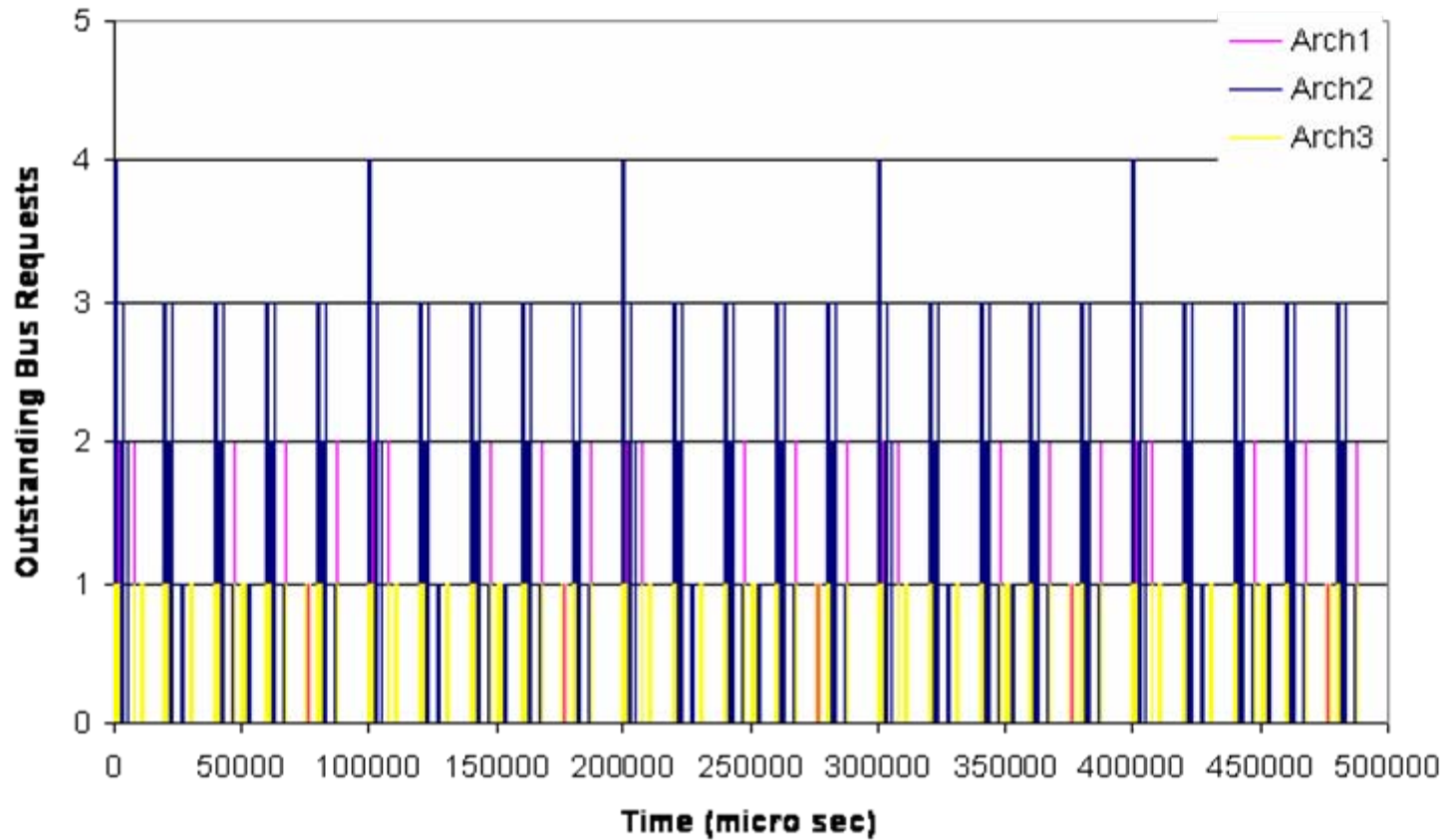
| | PE1 | PE2 | PE3 | PE4 | PE5 | PE6 |
|----------------|------|------|------|------|------|------|
| Arch1 | | | | | | |
| IP Type | ASIC | ASIC | ASIC | ASIC | ASIC | GPP0 |
| OS | - | - | - | - | - | RM |
| Tasks | 21 | 15 | 15 | 18 | 13 | 32 |
| Arch2 | | | | | | |
| IP Type | GPP0 | FPGA | ASIC | FPGA | FPGA | ASIC |
| OS | RM | RM | - | RM | RM | - |
| Tasks | 18 | 15 | 21 | 18 | 18 | 24 |
| Arch3 | | | | | | |
| IP Type | ASIC | FPGA | ASIC | GPP0 | ASIC | GPP0 |
| OS | - | RM | - | RM | - | RM |
| Tasks | 15 | 15 | 15 | 18 | 19 | 32 |
| Arch4 | | | | | | |
| IP Type | ASIC | FPGA | ASIC | GPP0 | ASIC | GPP0 |
| OS | - | EDF | - | EDF | - | EDF |
| Tasks | 15 | 15 | 15 | 18 | 19 | 32 |

Processor Utilization



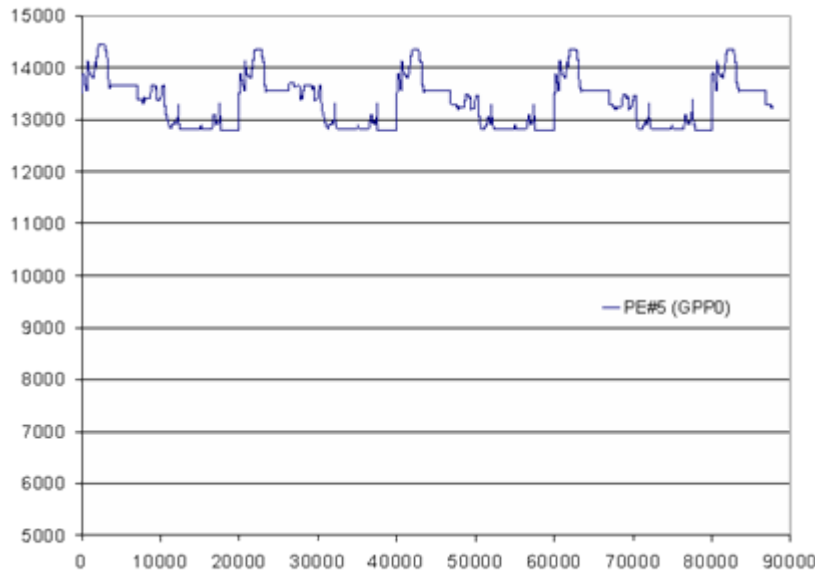
Different Partition and Architecture using RM
Different OS

Bus Contention



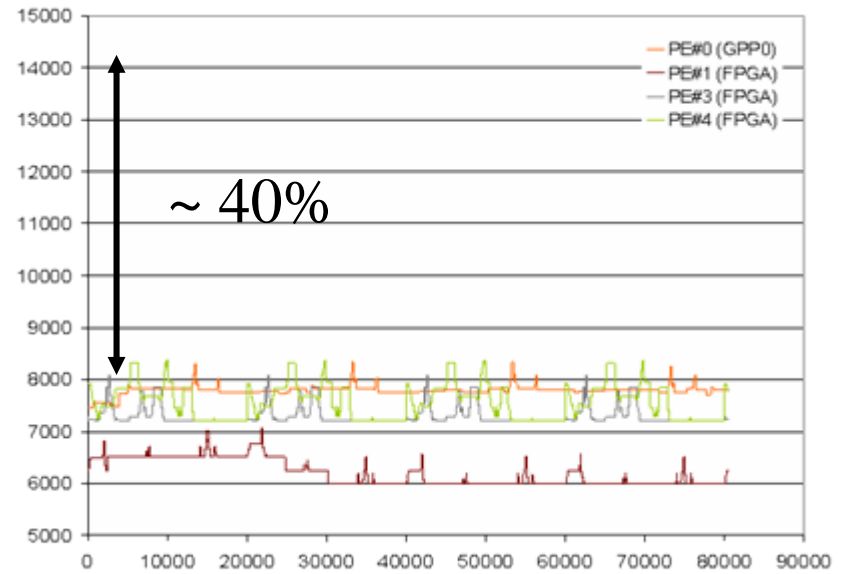
32-bit Words (Instruction + Dynamic Memory)

Arch1



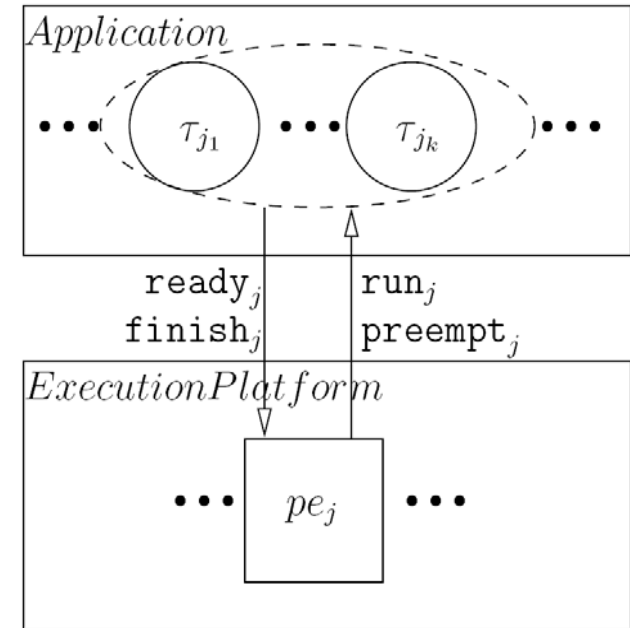
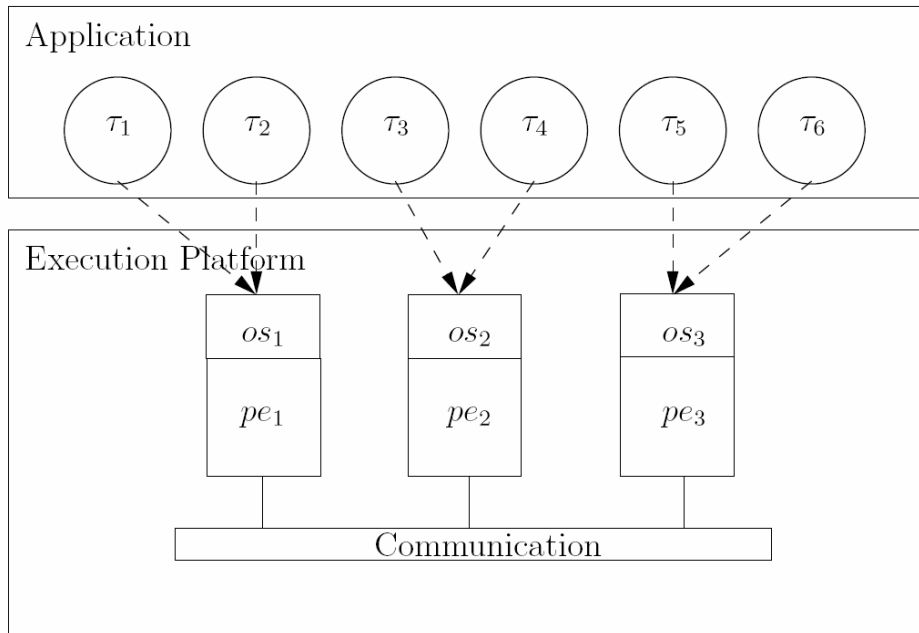
Time (μ s)

Arch2



Time (μ s)

❖ A timed-automata model



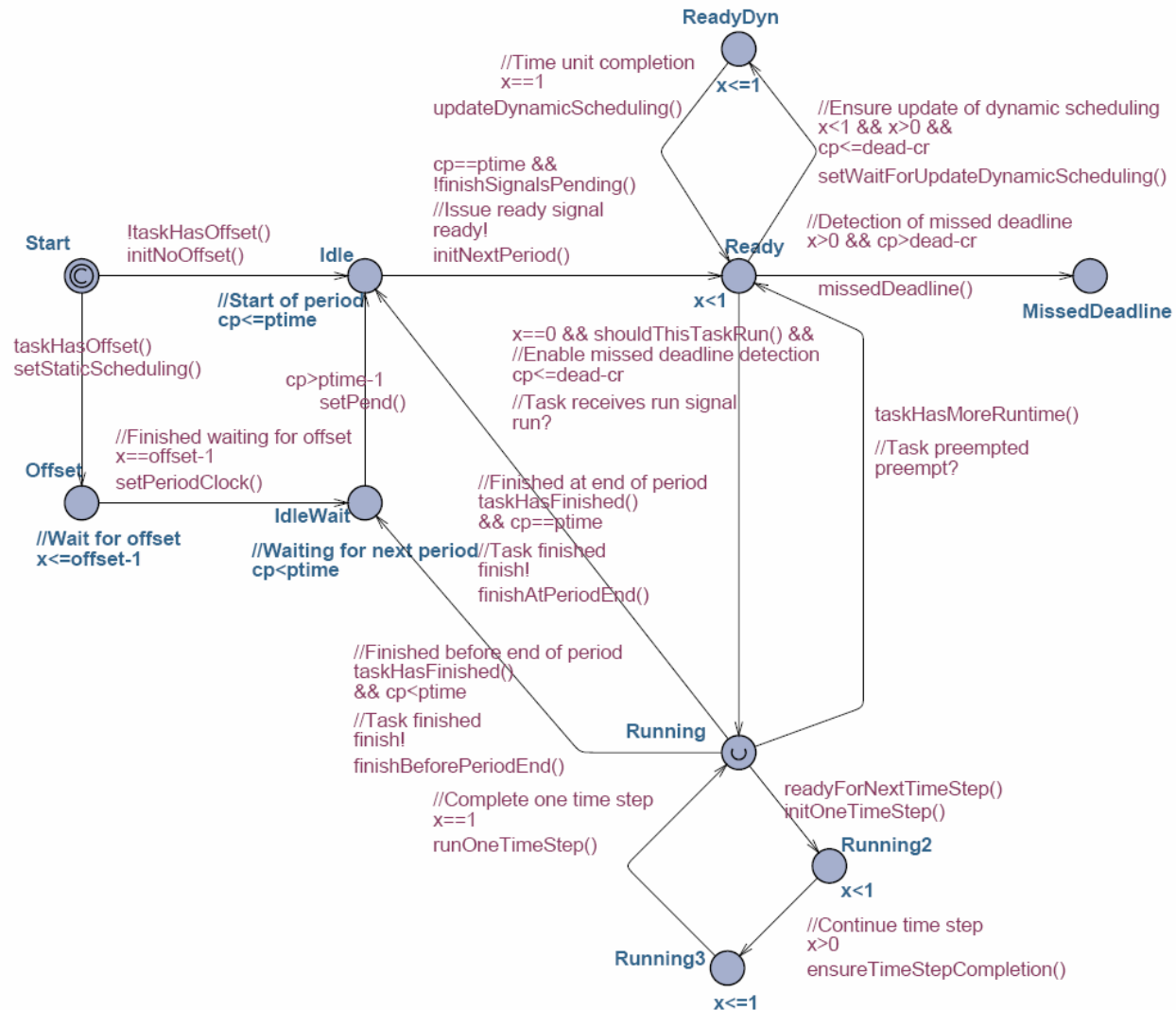
$$\text{System} = \text{Application} \parallel \text{ExecutionPlatform}$$

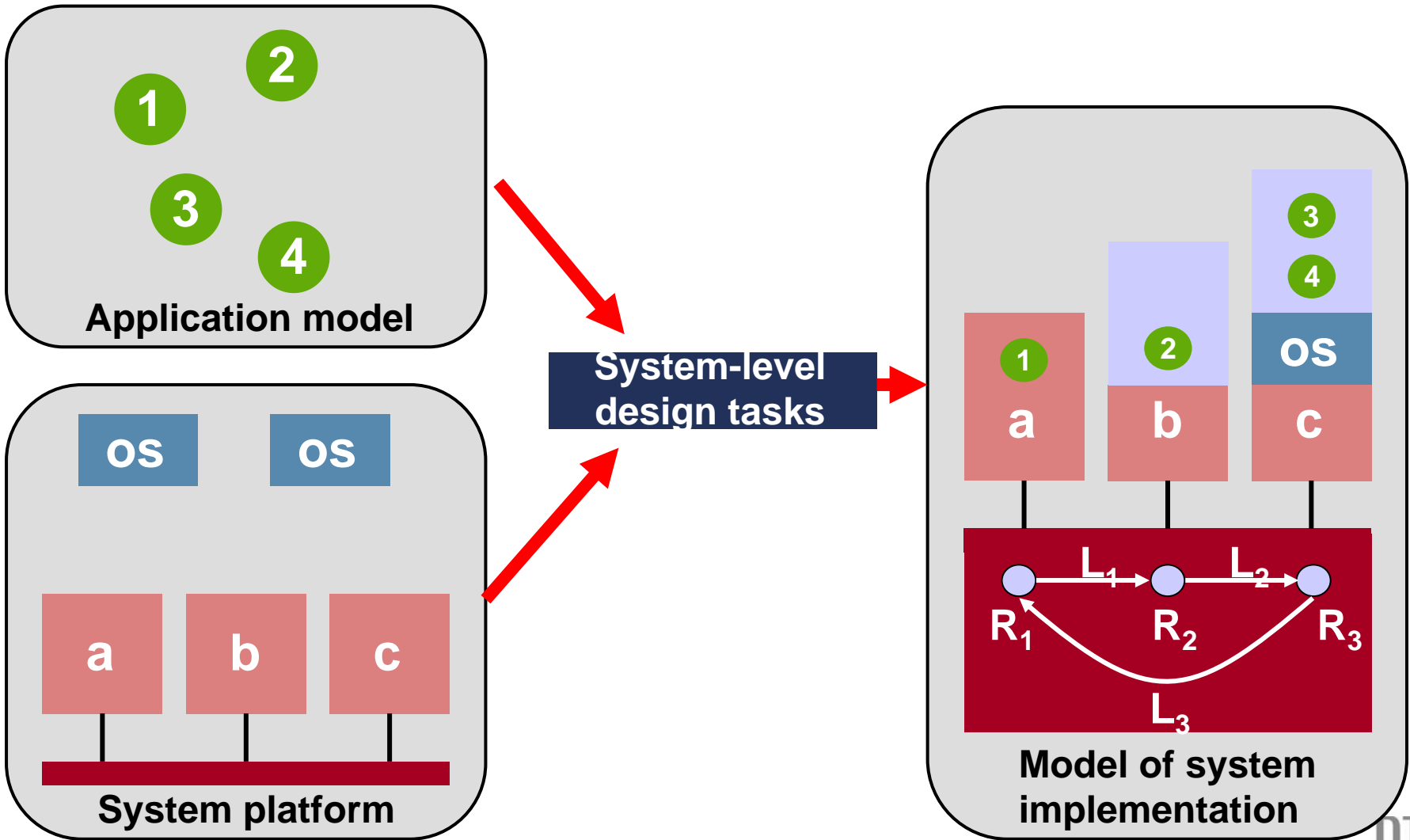
$$\text{Application} = \parallel_{i=1}^n \tau_i$$

$$\text{ExecutionPlatform} = \parallel_{j=1}^m pe_j$$



Automaton for a task





$E \langle \rangle \text{missedDeadline}$

$E \langle \rangle \text{totalCostUsed (Memory)} \geq 23$

$E \langle \rangle \text{totalCostUsed (Energy)} \geq 15$

Required specification

Model checking

1

a

2

b

3

4

OS

c

R_1

R_2

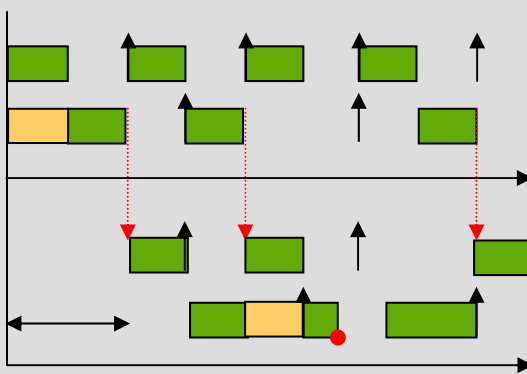
R_3

L_1

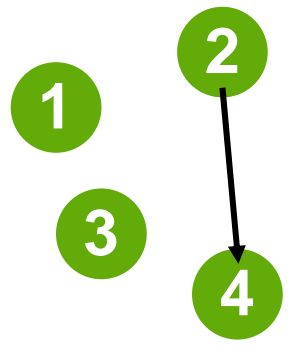
L_2

L_3

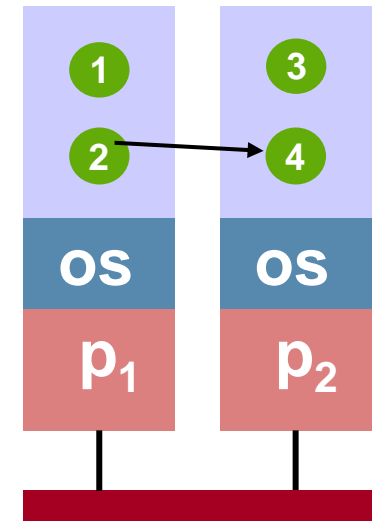
Model of system implementation



Example



| | o_i | T_i | e_i |
|---|-------|-------|-------|
| 1 | 0 | 4 | 2 |
| 2 | 0 | 6 | 2 |
| 3 | 0 | 6 | 2 |
| 4 | 4 | 6 | 3 |
| → | 0 | 6 | 1 |



Example



```
public class system {
    public Application apps;
    public Platform pl;
    public system(int M, int N, int granularity) {
        Resource r1 = new Resource();
        //Defines tasks (Execution time, Deadline, offset, period, FP)
        Task t1 = new Task(2, 2, 4, 0, 4, 1);
        Task t2 = new Task(1, 1, 6, 0, 6, 2);
        Task t3 = new Task(3, 3, 6, 4, 6, 3);
        Task t4 = new Task(2, 2, 6, 0, 6, 4);
        Task tm = new Task(1, 1, 6, 0, 6, 5);

        //Defines processors
        Processor p1 = new Processor(1, Processor.RM);
        Processor p2 = new Processor(1, Processor.EDF);
        Processor pm = new Processor(1, Processor.RM);

        //Assigns tasks to processors
        Task[][] tasks = {{t1,t2},{t3,t4},{tm}};

        //Adds the processors to the system
        Processor[] ps = {p1,p2,pm};
        //t2.softDeadline();
    }
}
```

Example



```
Cost memory = new Cost(tasks);  
Cost power = new Cost(tasks);  
Cost[] ca = {memory, power};
```

Static memory

Dynamic memory

```
    form(ps);  
    Application(task...
```

```
memory.set(t1,1,0,0,3);  
memory.set(t2,1,0,0,7);  
memory.set(t3,1,0,0,9);  
memory.set(t4,2,0,0,6);  
memory.share(t2,t4,5);
```

Data to be transferred

```
power.set(t1,0,0,0,5);  
power.set(t2,0,0,0,10);  
power.set(t3,0,0,0,5);  
power.set(t4,0,0,0,10);
```

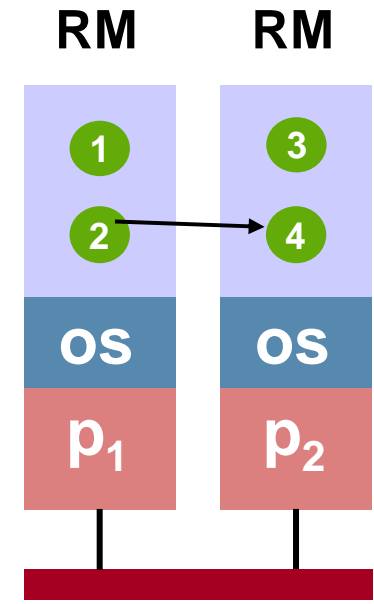
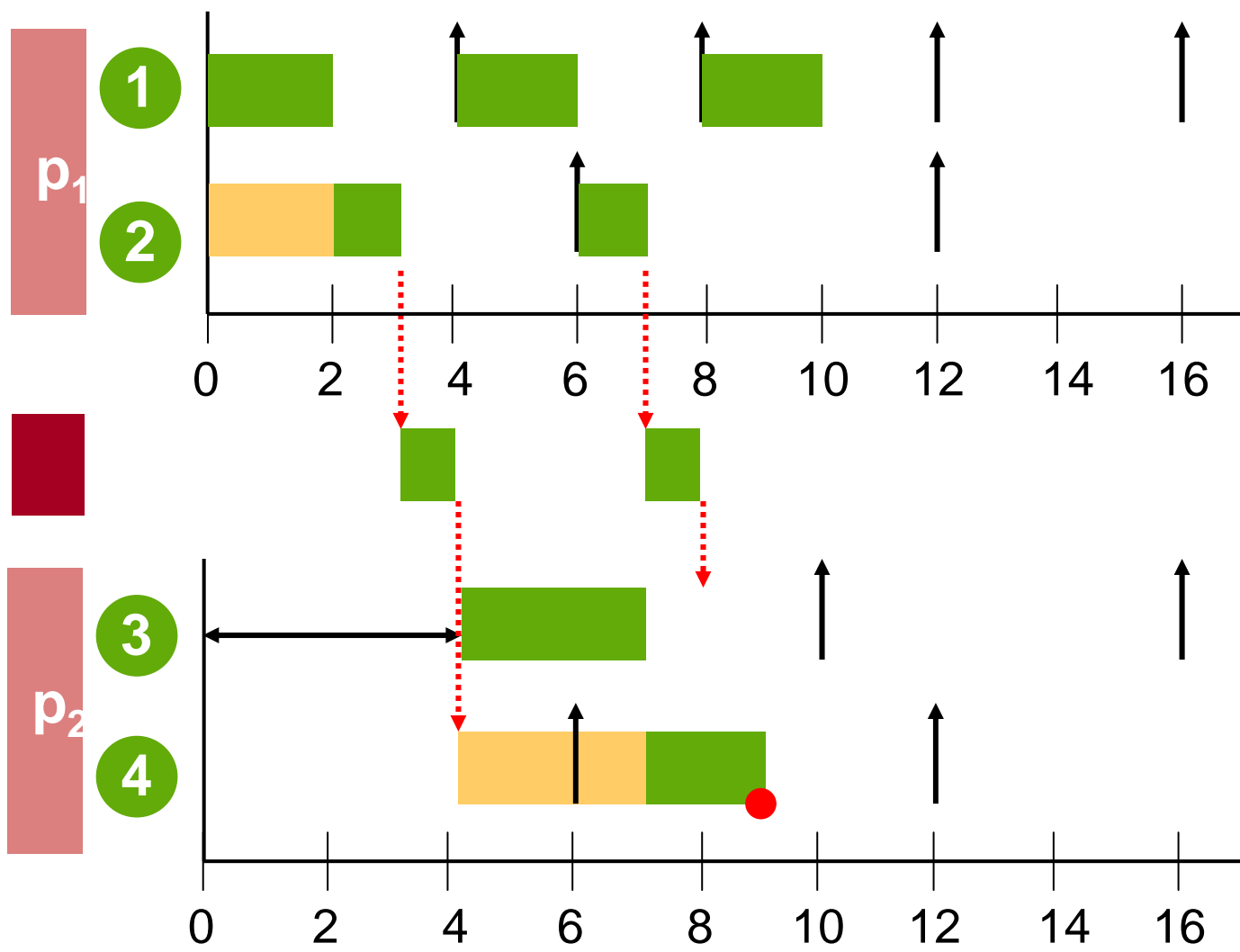
Bus is a shared resource

```
//Add resources  
apps.useResource(tm,r1);  
  
//Add dependencies to the system.  
apps.addDep(t2,tm);  
apps.addDep(tm,t4);
```



MOVES demo

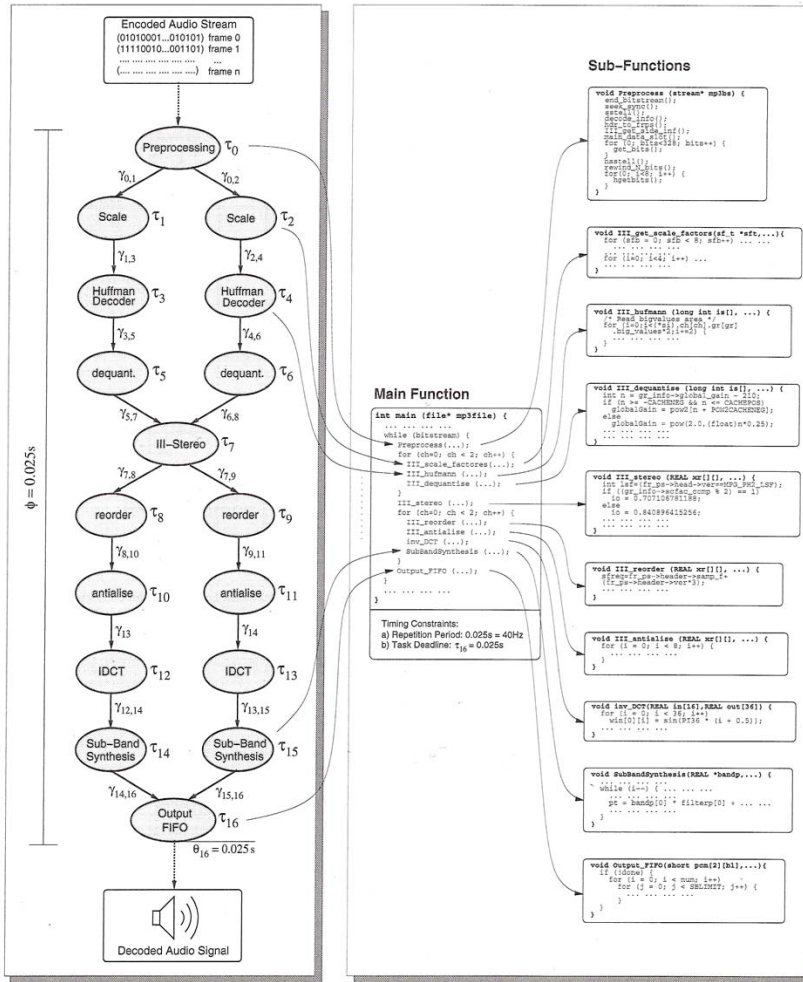
Why did it fail?





MOVES demo

MP3 decoder



(a)

(b)

[Application from Marcus Schmitz, TU Linköping]





■ Acknowledgements

MOVES

- Michael R. Hansen
- Aske Brekling
- Jens Ellebæk
- Kristian S. Knudsen

ARTS

- Shankar Mahadevan
- Kehuai Wu
- Kashif Virk
- Mercury Gonzalez

Questions?