

Using Model Checking to Identify Timing Interferences on Multicore Processors

Viet-Anh Nguyen¹, **Eric Jenn**¹,
Frederic Lang², Wendelin Serwe², and Radu Mateescu²

¹ IRT Saint Exupery, Toulouse

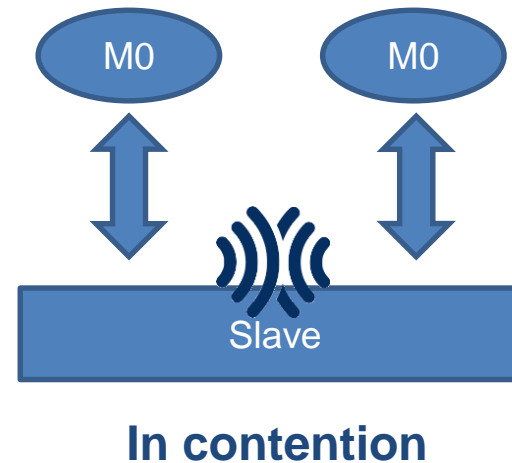
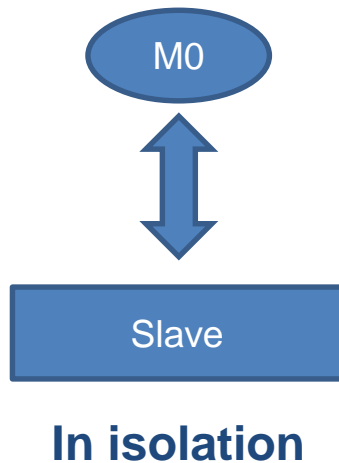
² CONVECS, INRIA / LIG, Grenoble



CAPHCA

- ❑ Temporal interferences
- ❑ From structural analysis to behavioral analysis
- ❑ Interference analysis using model checking
- ❑ Experimental results
- ❑ From behavioral analysis to behavioral timed analysis
- ❑ Conclusions

- ❑ A real-time system must (ideally) be **temporally deterministic**
- ❑ **Temporal determinism** is affected by **temporal interferences**



- ❑ Temporal interferences must be **prevented** by design or by usage, or their effects must be **estimated** and **bounded**
- ❑ Temporal interferences must be **identified**

- ❑ Most studies consider sources of interferences **to be either known** or identified by a **manual analysis** of the documentation or by measurement
- ❑ Some automated analyses are based on a **purely structural analysis**



Our proposed interference analysis

Model behavior of SW, HW, and their interactions

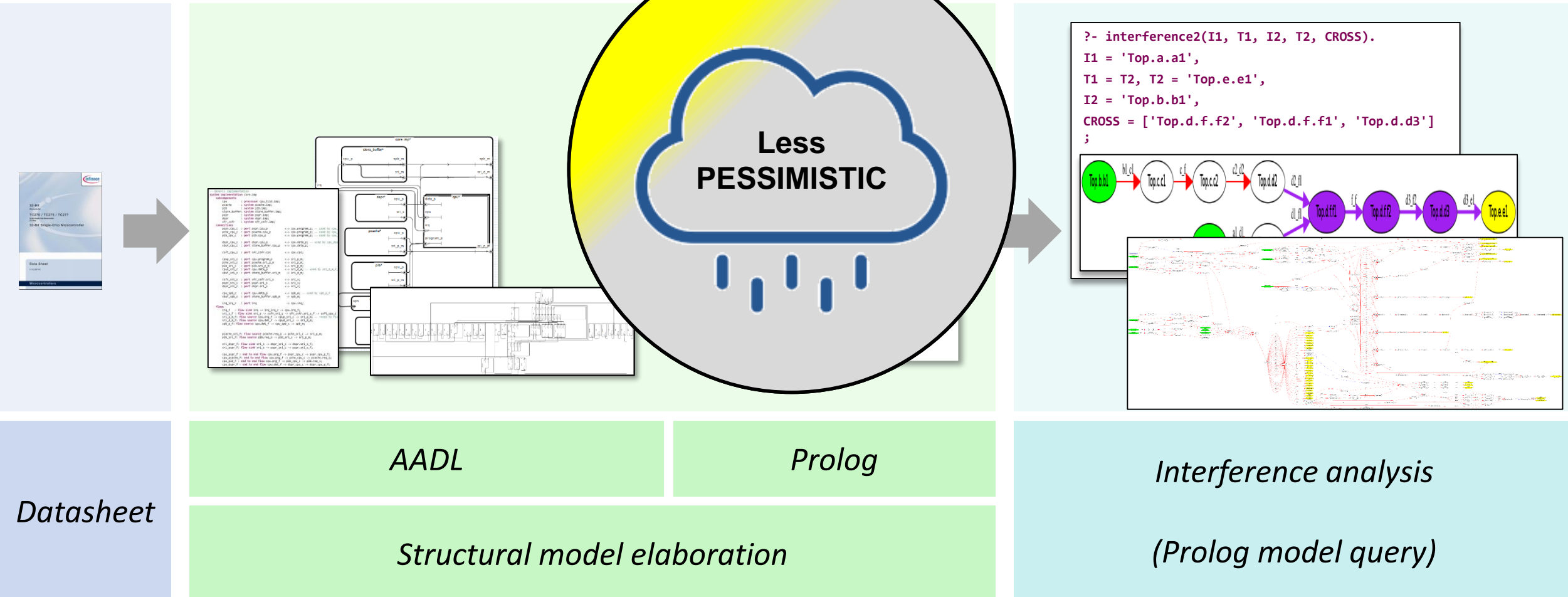


Safe

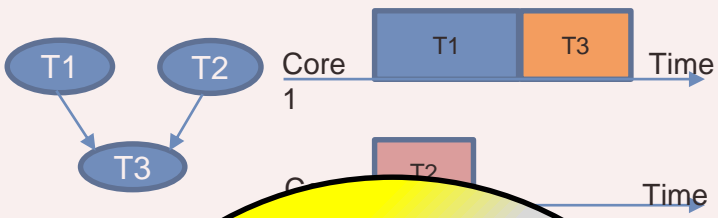


Accurate (prevent false alarms)

- ❑ Most studies consider sources of interferences **to be either known** or identified by a **manual analysis** of the documentation or by measurement
- ❑ Some automated analyses are based on **structural analysis**



Schedule model



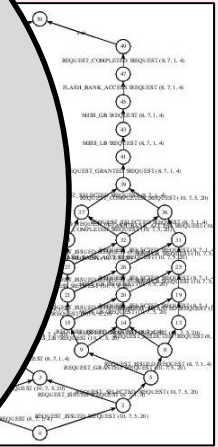
```

process ARBITER[REQ,
REQUEST_BEING_S,
BAD_VALUE, NOT_
PIPELINE_FULL: n

var register, pipe

register := EM
pipeline := EM
loop L in
  select (* Grant r
    [] (*Forward
    [] break L
  end select
end loop
end var
end process
  
```

**Less
PESSIMISTIC**



CHECKING_ARBITER_INTERFERENCE (TC275, TRUE, 8, 7, 1, 4)
 | It is potential that request id: 1 from MCI 8 to SCI 7 for address: 4 is suffered from contentions at the arbiter
PASS



Datasheet

Structural
model
elaboration
(informal)

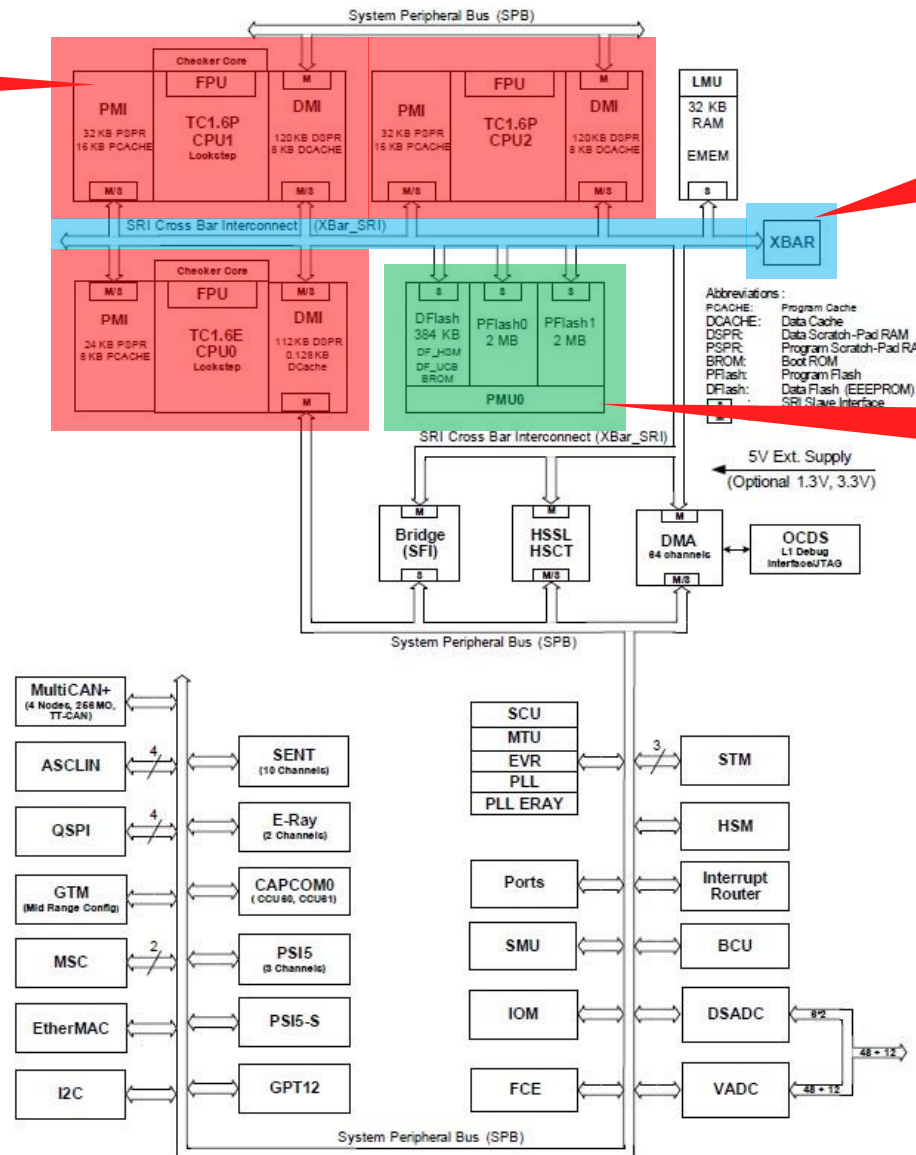
Behavioral model elaboration
(LNT)

Interference analysis
(CADP)

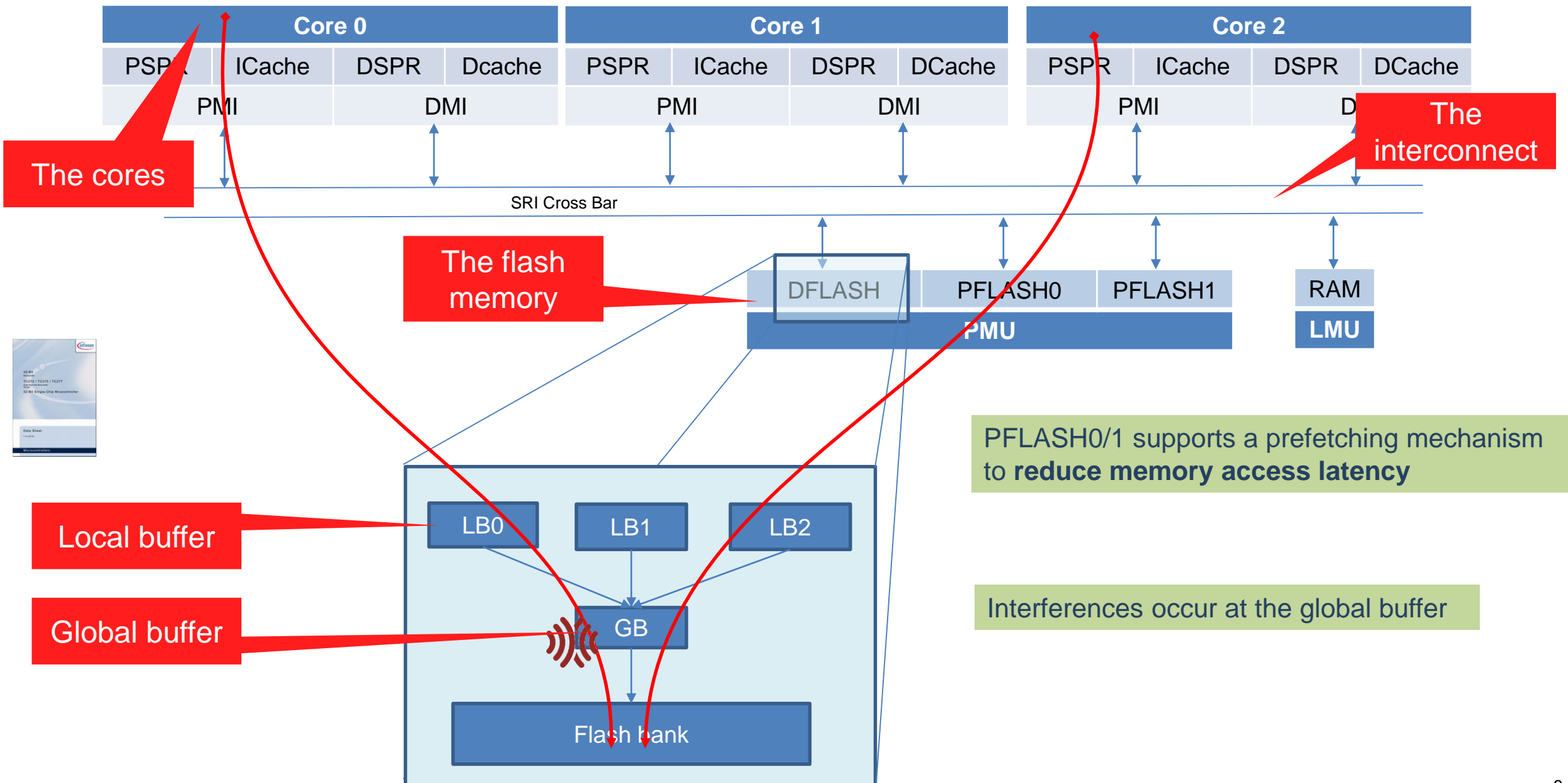
The cores

The interconnect

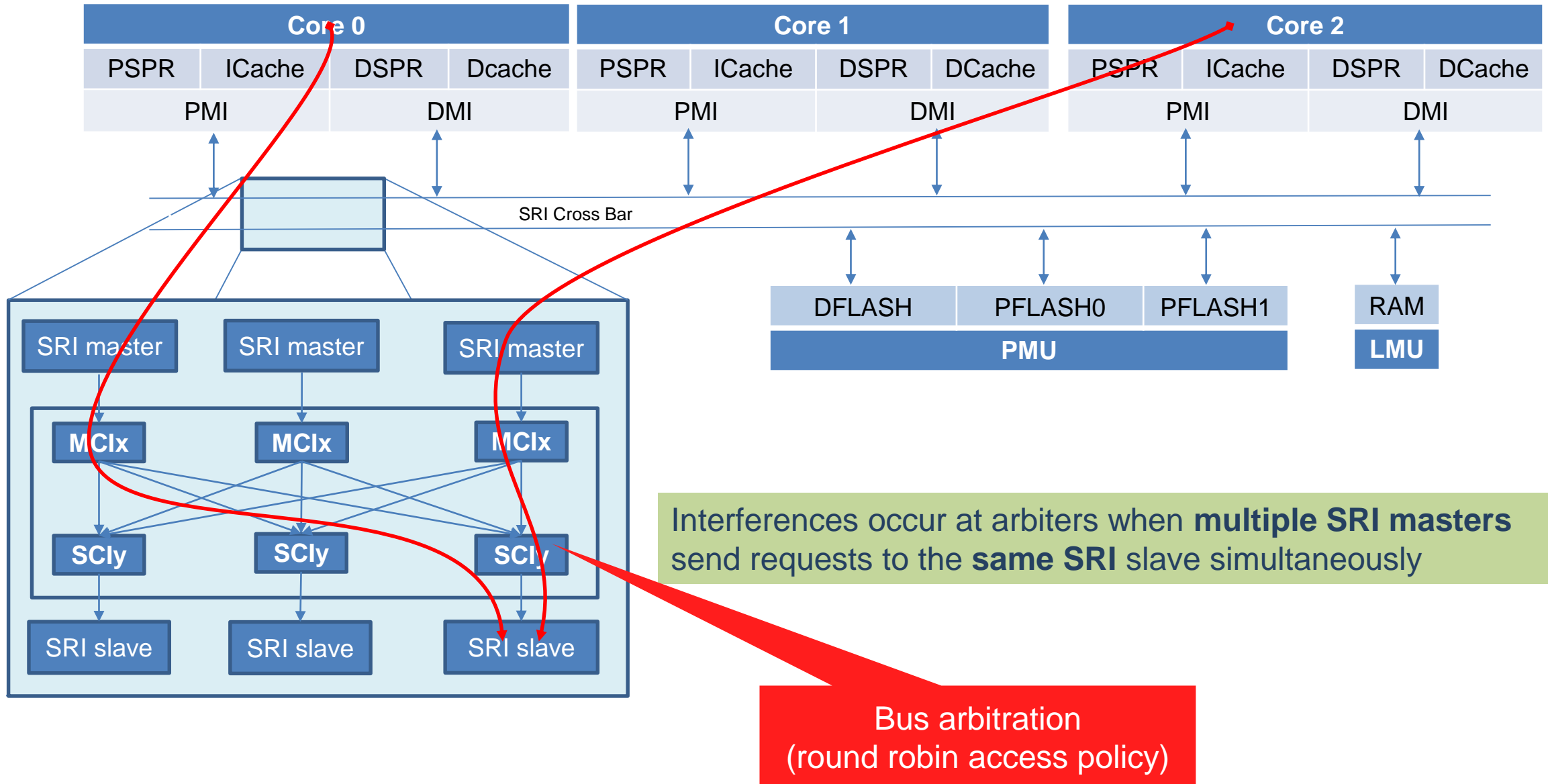
The flash memory



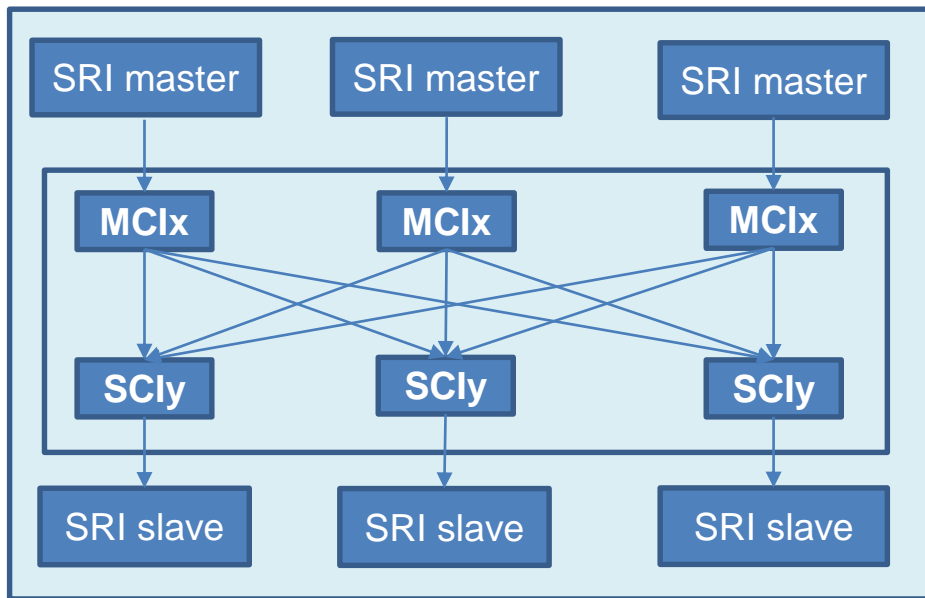
The Aurix TC275 platform



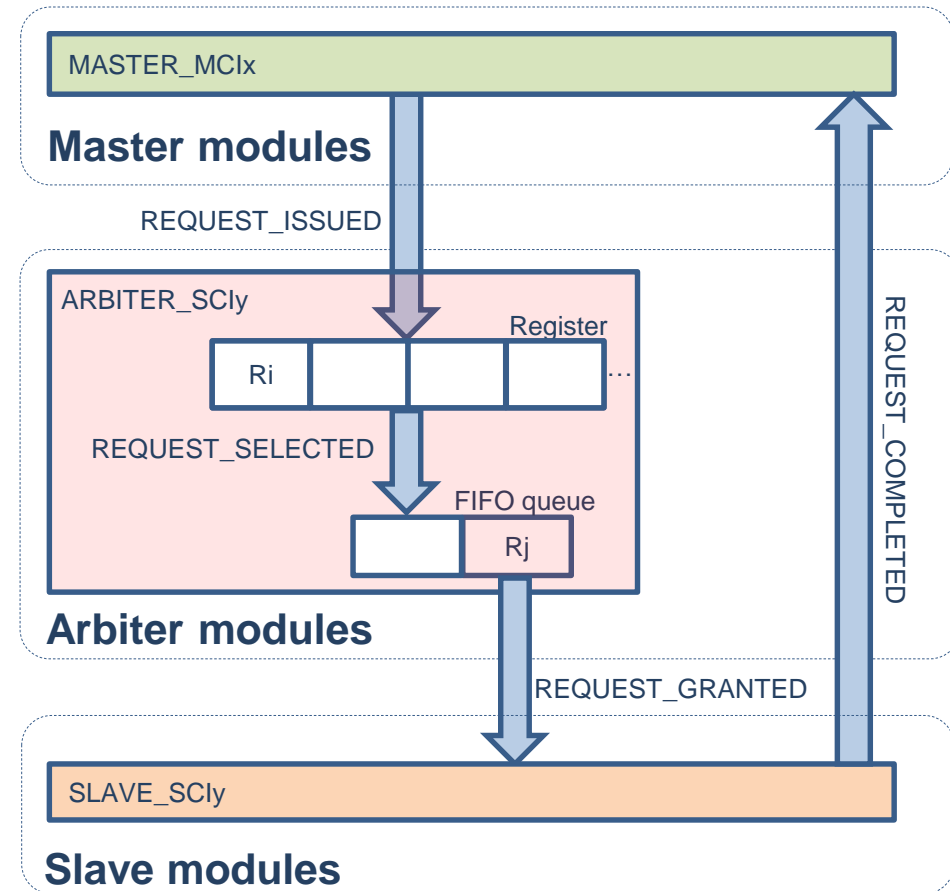
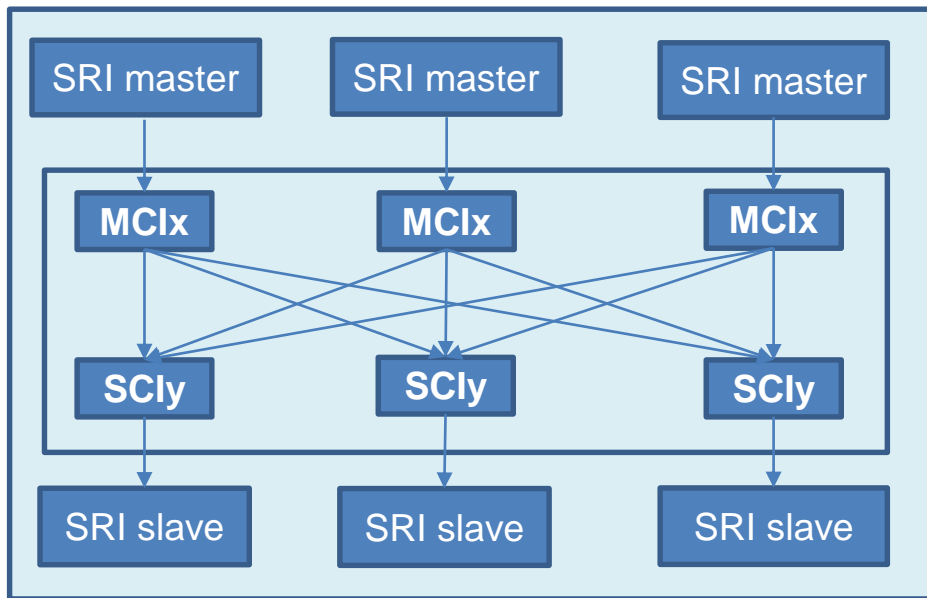
The Aurix TC275 platform



- ❑ Tasks dependencies and interactions between SW and HW are modelled by **synchronization gates**



- ❑ Tasks dependencies and interactions between SW and HW are modelled by **synchronization gates**



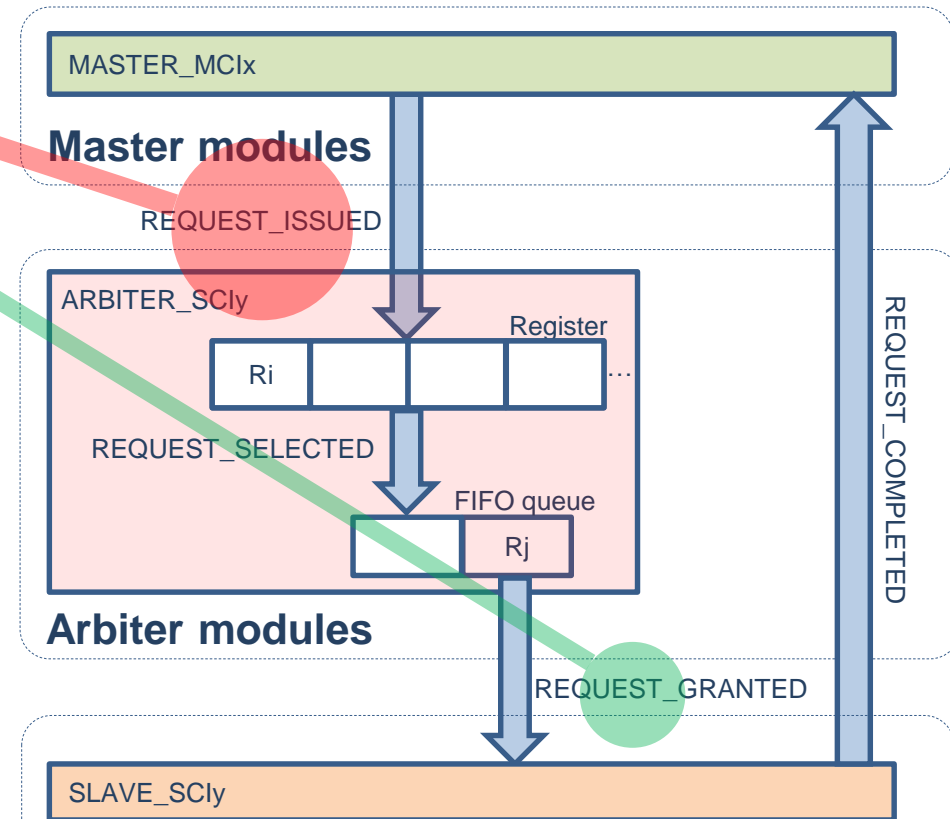
- Tasks dependencies and interactions between SW and HW are modelled by **synchronization gates**

process MASTER_MCix[REQUEST_ISSUED, REQUEST_COMPLETED: REQUEST_CHANNEL] is


```

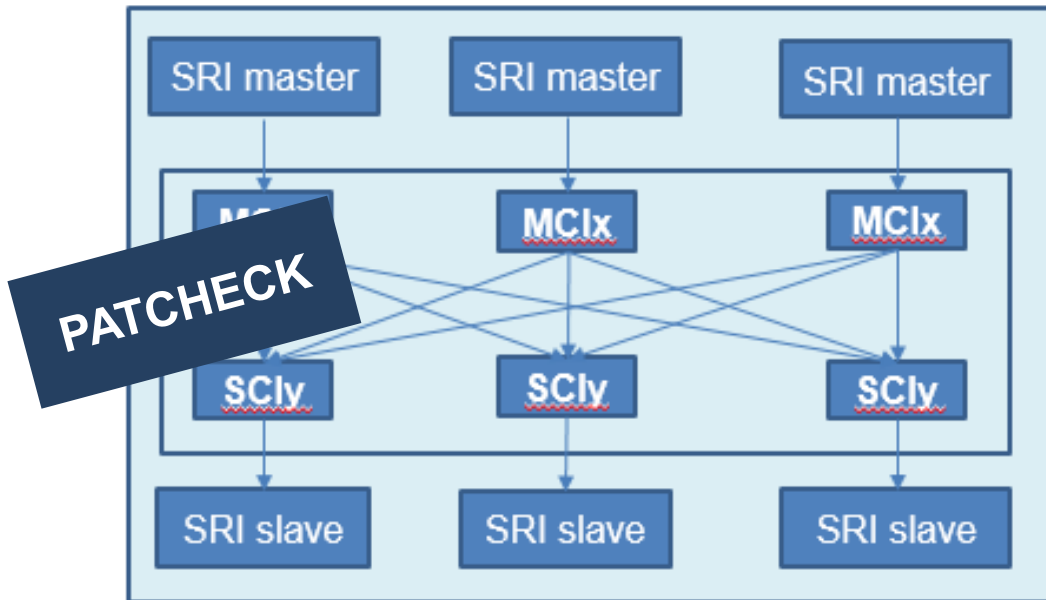
process ARBITER_SCly[REQUEST_ISSUED, REQUEST_SELECTED, REQUEST_GRANTED: REQUEST_CHANNEL, raise
BAD_VALUE, NOT_FOUND: none] is
  var register: RegSlotInfo_Array, previous_winning_MCix: Nat, fifo_queue : QUEUE in
    register := RegSlotInfo_Array(RegSlotInfo(REQUEST(0,0,0,0),false));
    fifo_queue := EMPTY_QUEUE;
    loop L in
      select
        (* Enqueue requests*)
        []
        (*Arbitrate request*)
        []
        (*Grant access to the slave*)
        []
      break L
    end select
  end loop
end var
end process
  
```

process SLAVE_SCly[REQUEST_GRANTED, REQUEST_COMPLETED, HIT_LB, MISS_LB, HIT_GB, MISS_GB, FLASH_BANK_ACCESS:REQUEST_CHANNEL] is

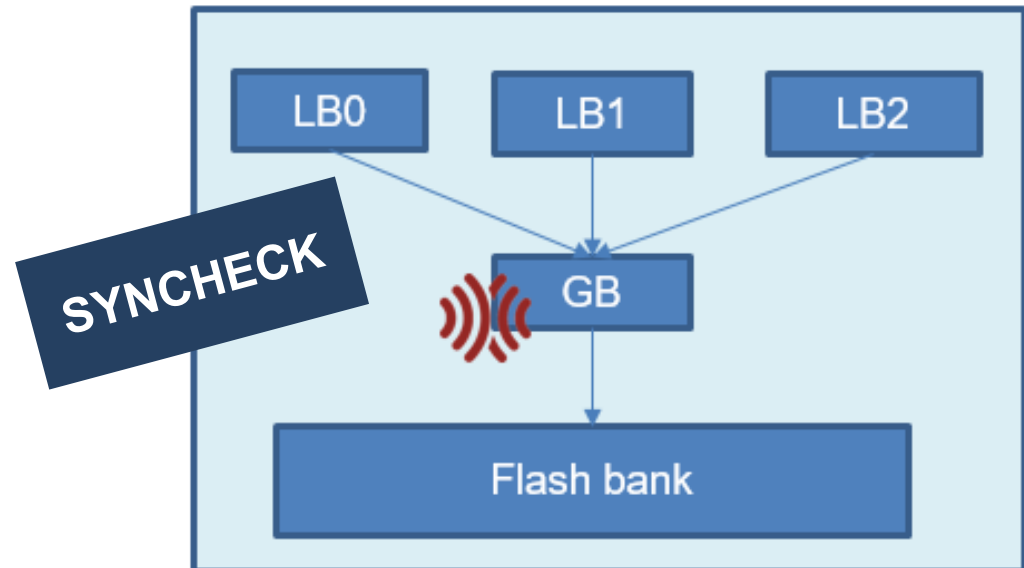


- ❑ Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

The pattern of actions is **known**  The pattern of actions is **unknown**



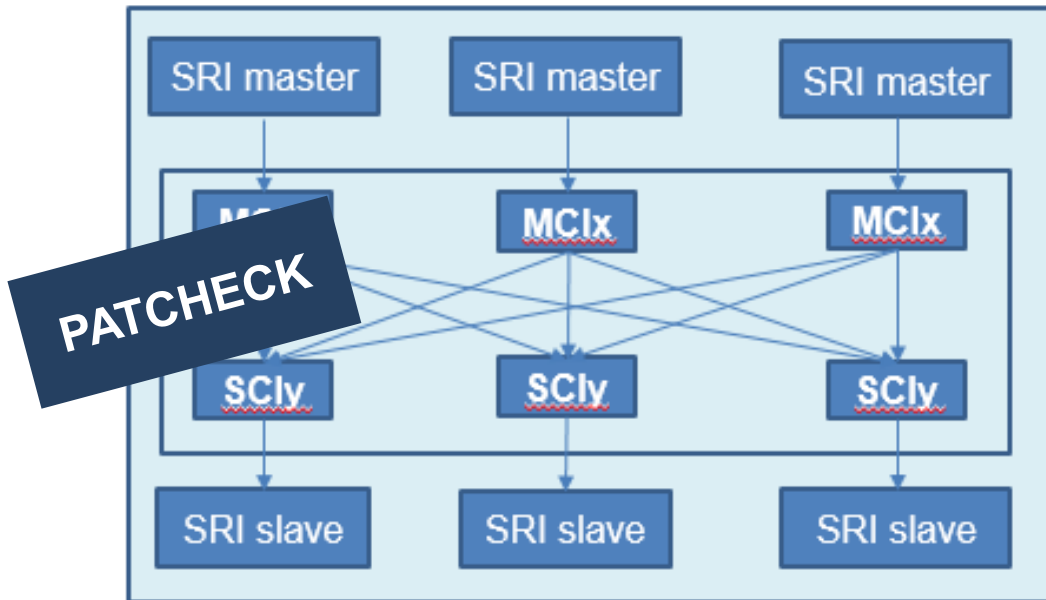
Crossbar arbitration



Flash-level prefetching

- ❑ Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

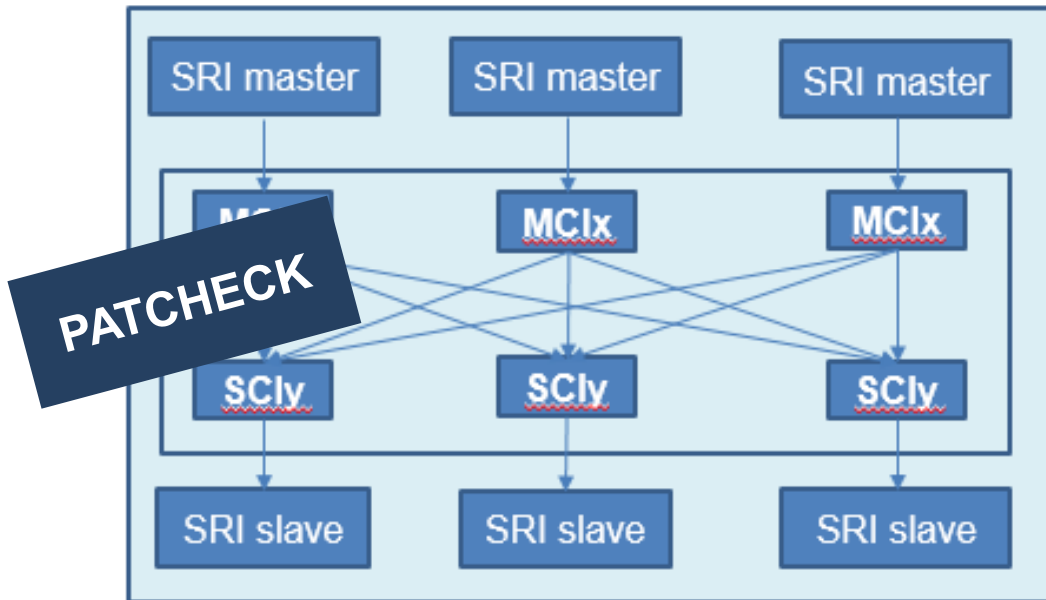
The pattern of actions is **known**



Crossbar arbitration

- ❑ Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

The pattern of actions is **known**

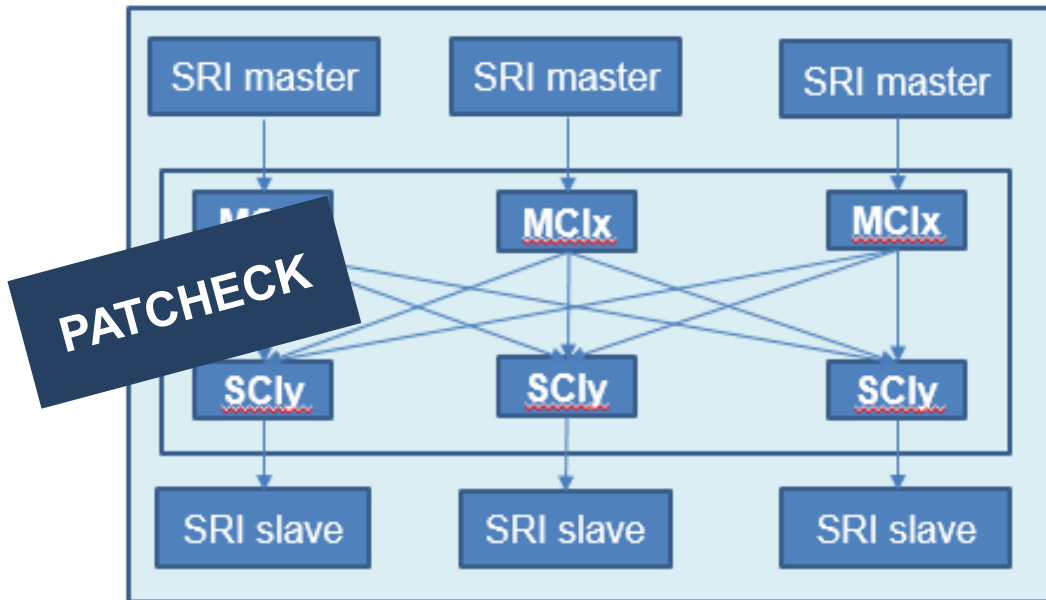


Crossbar arbitration

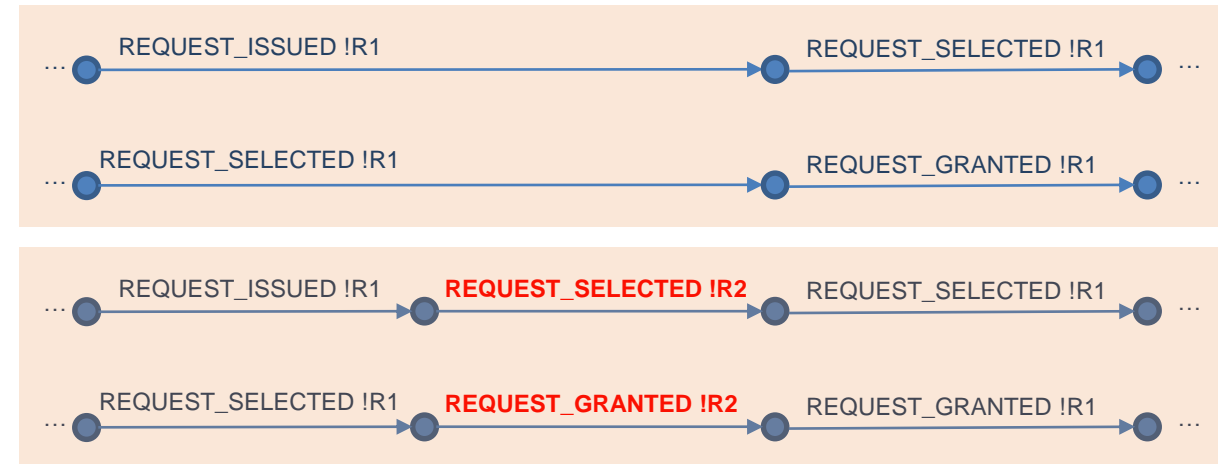


- ❑ Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

The pattern of actions is **known**

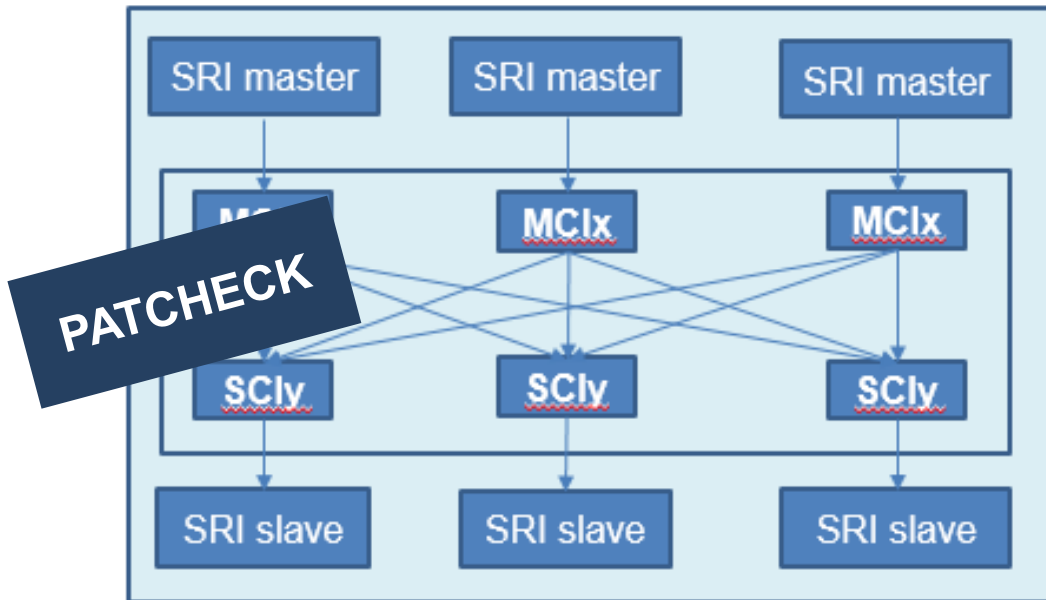


Crossbar arbitration

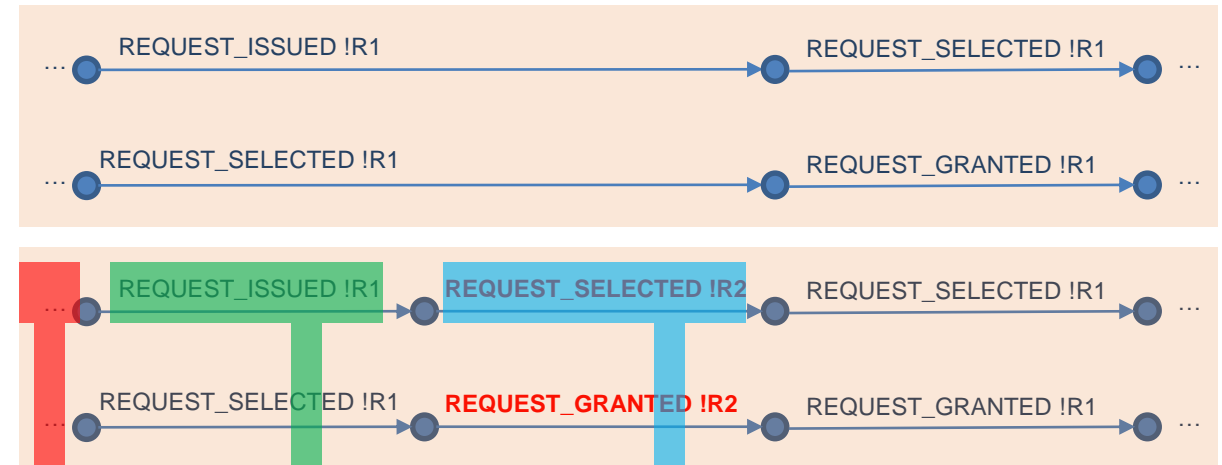


- Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

The pattern of actions is **known**



Crossbar arbitration

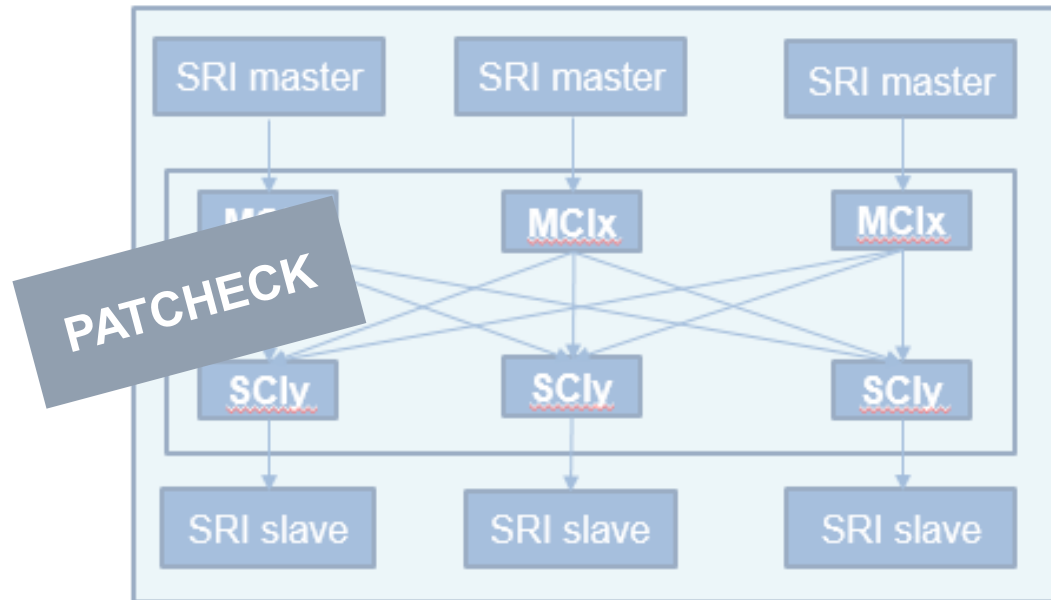


```

property CHECKING_ARBITER_INTERFERENCE(LNT_MODEL, RESULT, m, n, p, q)
  "It is potential that request id: $p from MCI $m to SCI $n for address: $q is suffered from contentions at the arbiter"
  is
    "a.bcg" = generation of "$LNT_MODEL.Int";
    "a.bcg" |=with evaluator4
    < true* "REQUEST_ISSUED IREQUEST ($m, $n, $p, $q)". (not "REQUEST_SELECTED IREQUEST ($m,
    $n, $p, $q)")* . {REQUEST_SELECTED ?R:String where R<>"REQUEST ($m, $n, $p, $q)"} >true
    or
    < true* . "REQUEST_SELECTED IREQUEST ($m, $n, $p, $q)". (not "REQUEST_GRANTED IREQUEST
    ($m, $n, $p, $q)")* . {REQUEST_GRANTED ?R:String where R<>"REQUEST ($m, $n, $p, $q)"} >true;
    expected "$RESULT"
end property
  
```

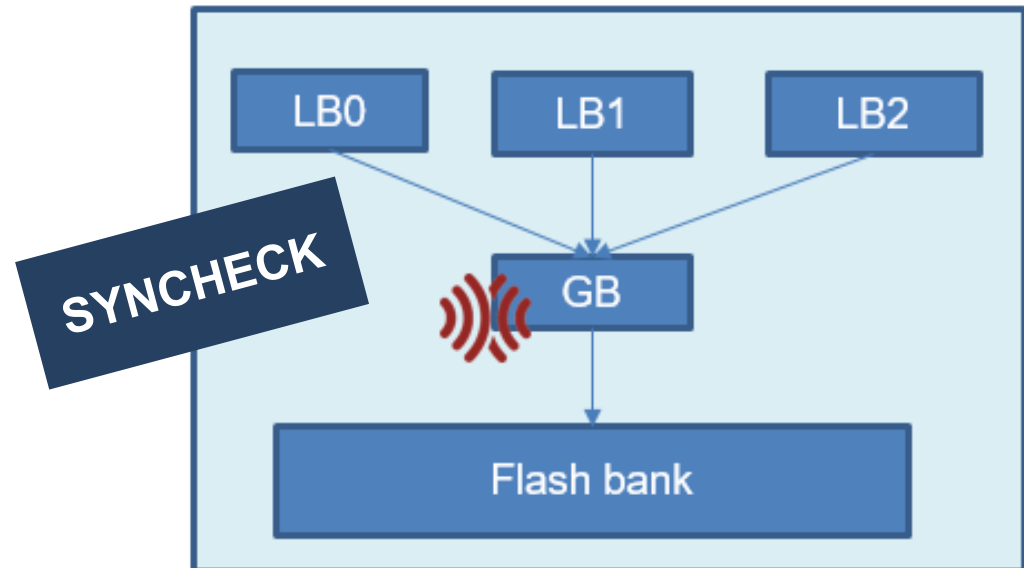
- ❑ Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

The pattern of actions is **known**



Crossbar arbitration

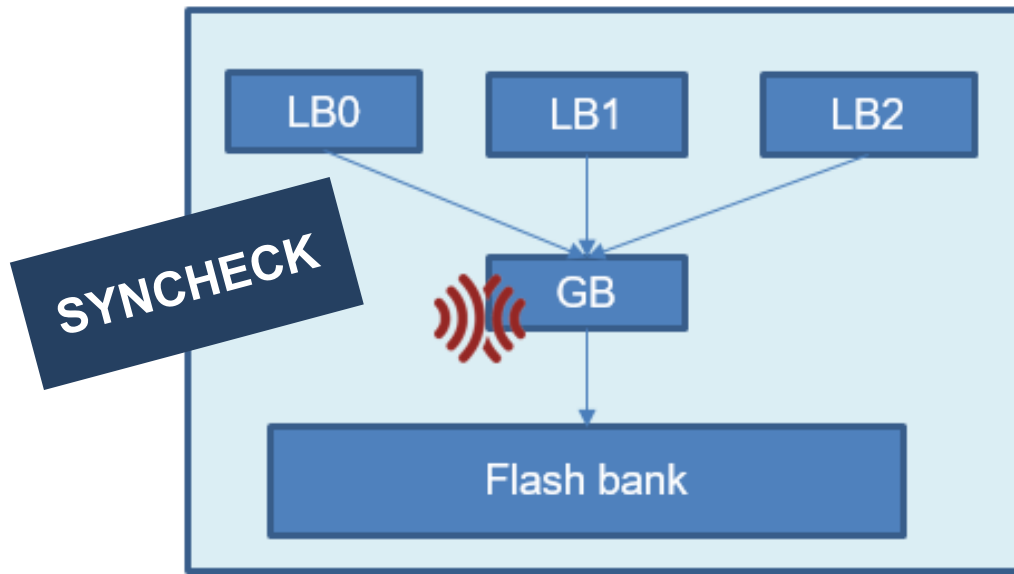
The pattern of actions is **unknown**



Flash-level prefetching

- ❑ Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

The pattern of actions is **unknown**



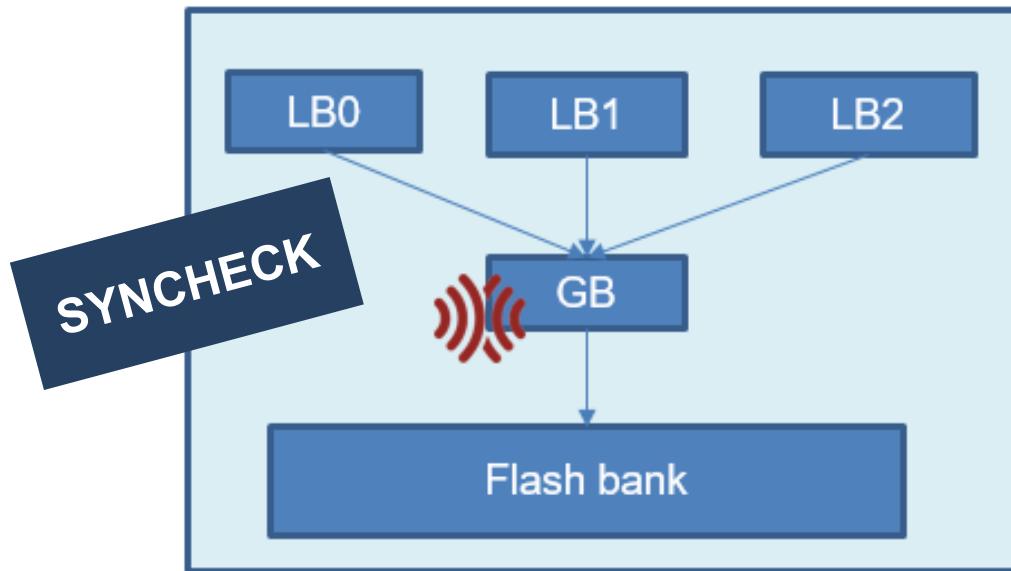
Flash-level prefetching

1st LTS: isolated model – M1



- ❑ Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

The pattern of actions is **unknown**



Flash-level prefetching

1st LTS: isolated model – M1

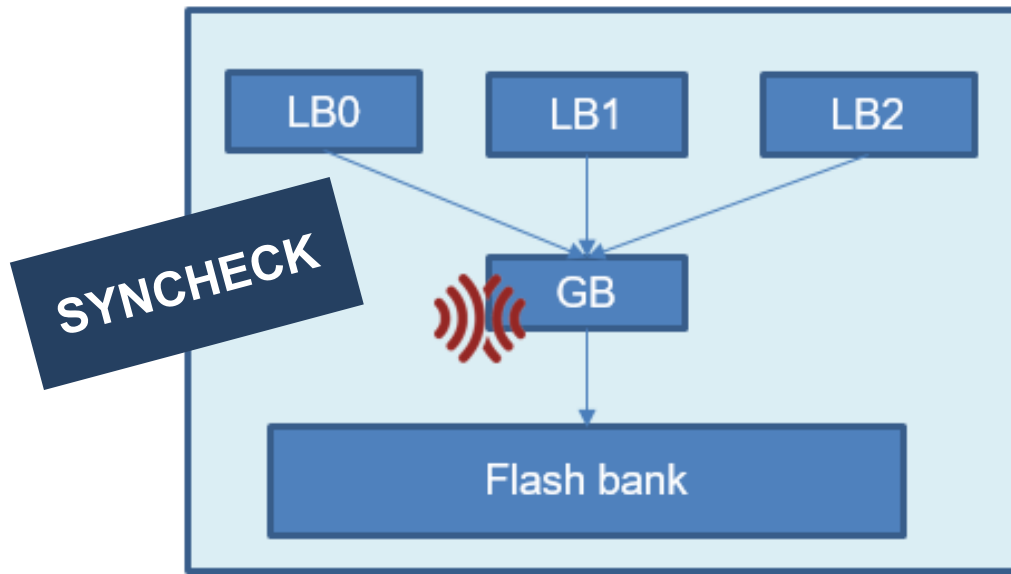


2nd LTS: non-isolated model – M2



- Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

The pattern of actions is **unknown**



Flash-level prefetching

1st LTS: isolated model – M1



2nd LTS: non-isolated model – M2

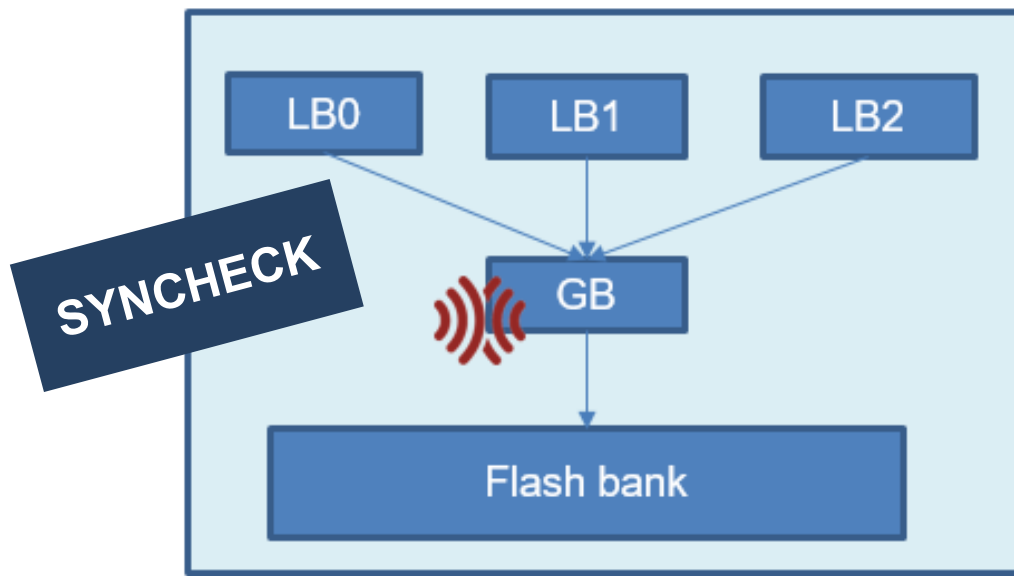


Synchronous LTS: M1 ⊗ M2



- Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

The pattern of actions is **unknown**



Flash-level prefetching

1st LTS: isolated model – M1



2nd LTS: non-isolated model – M2



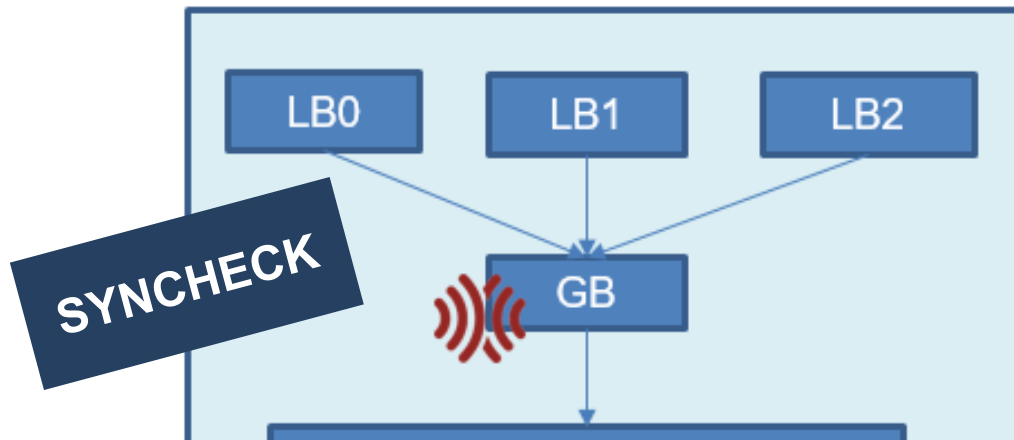
Synchronous LTS: M1 \otimes M2



`<true*. "INTF !Ri"> true`

- Interferences are reflected by the modifications of sequence of actions that have impacts on time of tasks when they execute in parallel with other tasks

The pattern of actions is **unknown**



$$\begin{aligned} & \{\tau_1 * _ \rightarrow \tau_1 \mid \tau_1 \in M1\} \cup \\ & \{\tau_1 * \tau_2 \rightarrow \text{INTF} \mid \tau_1 \in M1 \text{ and } \tau_2 \in M2 \text{ and } \tau_1 \neq \tau_2\} \cup \\ & \{_ * \tau_2 \rightarrow \tau_2 \mid \tau_2 \in M2\} \end{aligned}$$

1st LTS: isolated model – M1

... REQUEST_GRANTED !R1 → HIT_GB !R1 → ...



2nd LTS: non-isolated model – M2

... REQUEST_GRANTED !R1 → MISS_GB !R1 → ...

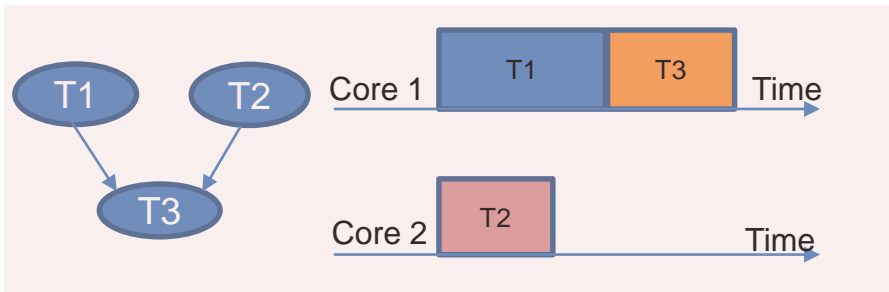
=

Asynchronous LTS: M1 ⊗ M2

REQUEST_GRANTED !R1 → INTF !R1 → ...

<true*. "INTF !Ri"> true

□ Toy task graphs

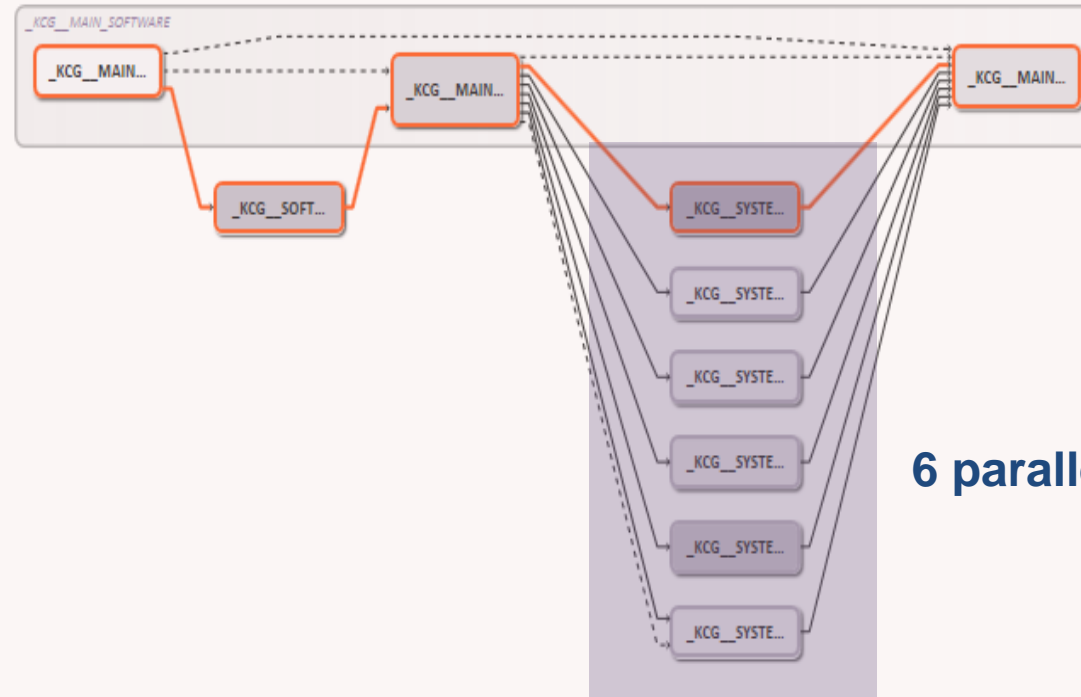


Task	Access type	Destination	Req. Addr
T1	Data	PFLASH0	4
T2	Data	PFLASH0	20
T3	Data	PFLASH0	8

Sources of interferences	Tasks	Pure structural analysis	CADP
Interferences at arbiters	T1	Yes	Yes
	T2	Yes	Yes
	T3	Yes	No
Interferences at global buffers	T1	Yes	No
	T2	Yes	No
	T3	Yes	Yes

Improvements

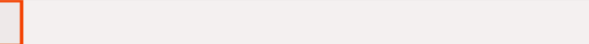
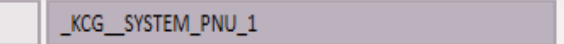



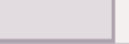
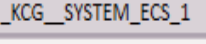
□ LIEBHERR's use case: TLS



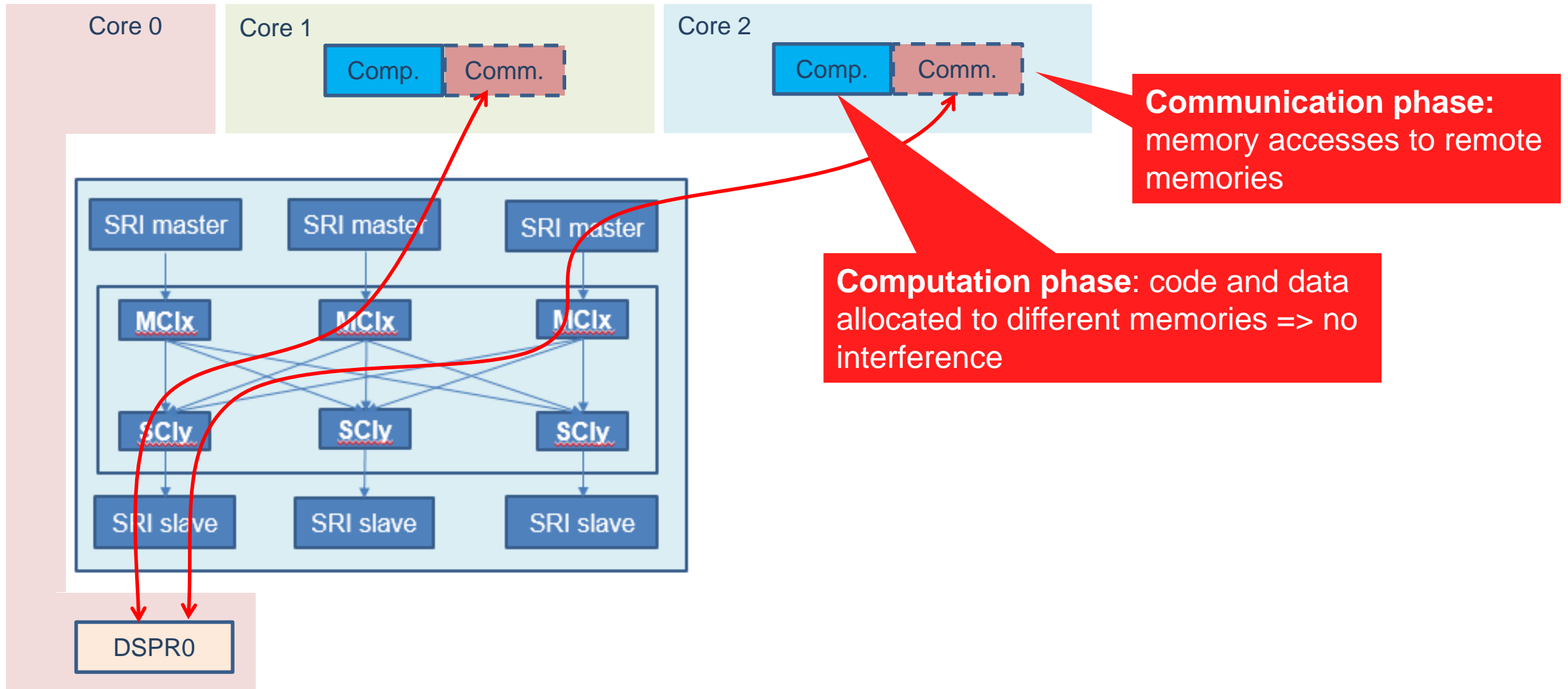
6 parallel tasks

Task graph

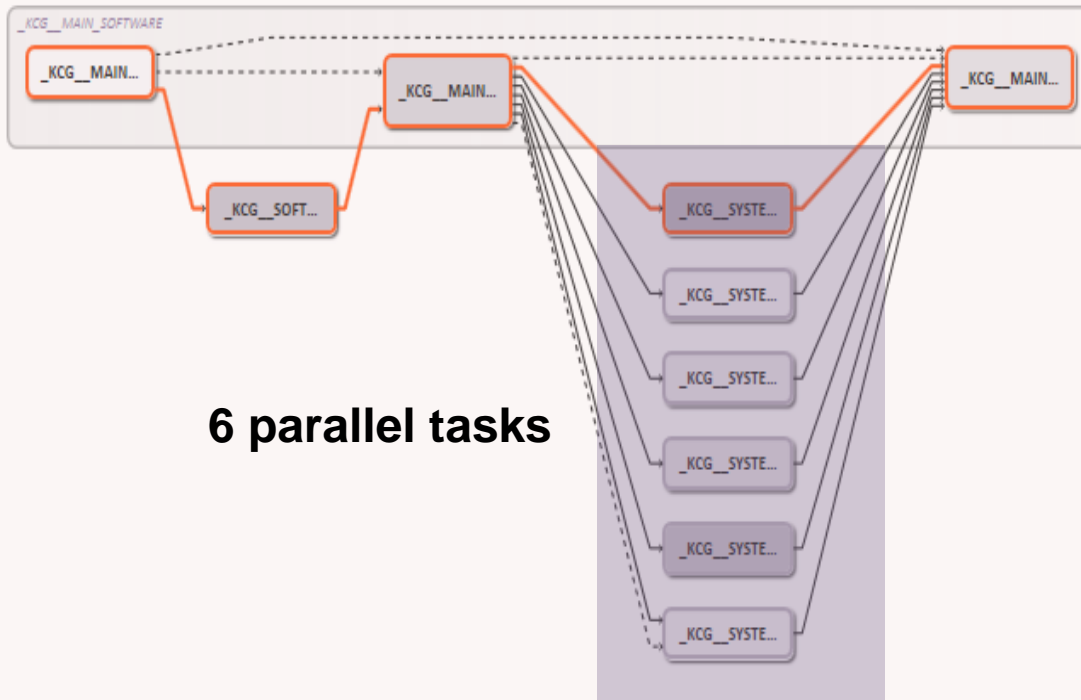


Process duration: 621 Number of allocated methods: 10/10 Average Usage: 47.24%				
0				60.2 %
1				65.7 %
2				15.8 %

Task schedule

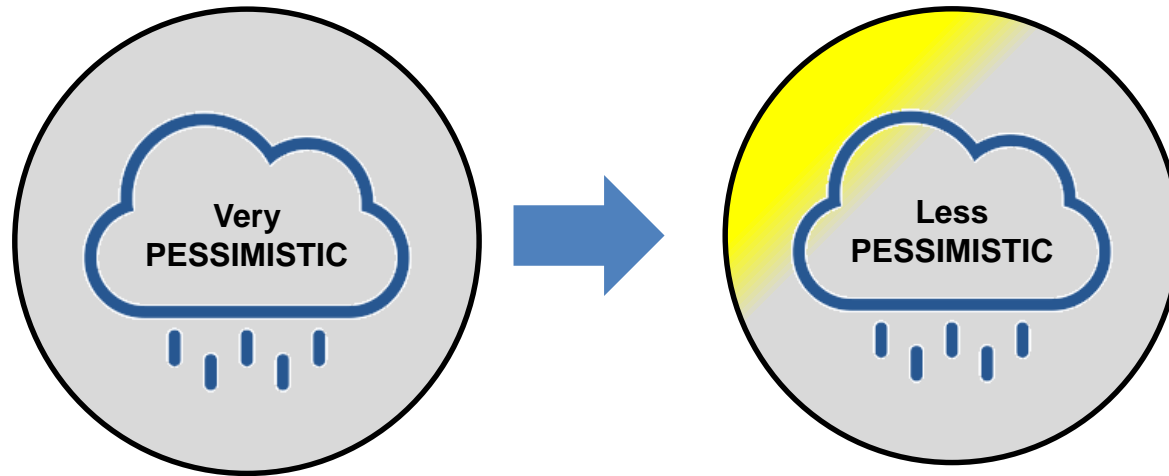


Improvements



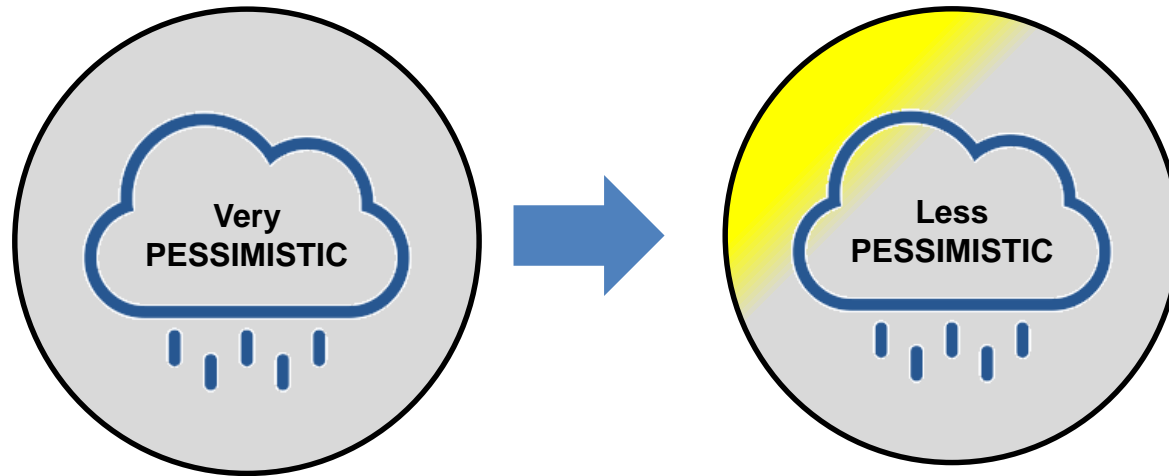
TASK	Pur structural analysis	CADP
SCADE_MAIN_SOFTWARE_1	YES	NO
SCADE_MAIN_SOFTWARE_2	YES	NO
SCADE_SOFTWARE_CPCS	YES	NO
SCADE_WAIS	YES	YES
SCADE_PNU	YES	YES
SCADE_DSHS	YES	YES
SCADE_AVS	YES	YES
SCADE_OPS	YES	YES
SCADE_ECS	YES	YES

- ❑ The solution is **safe^(*)** and **prevents reporting (some) spurious interferences**



(*) As long as the model is correct...

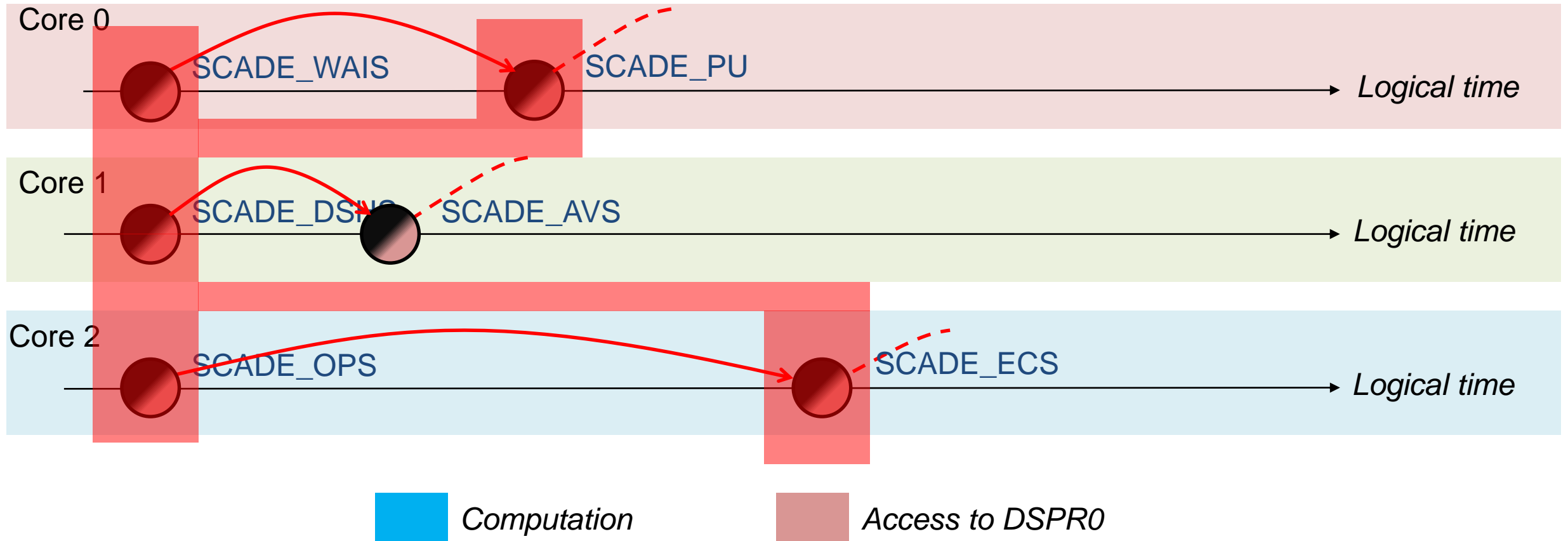
- ❑ The solution is **safe^(*)** and **prevents reporting (some) spurious interferences**



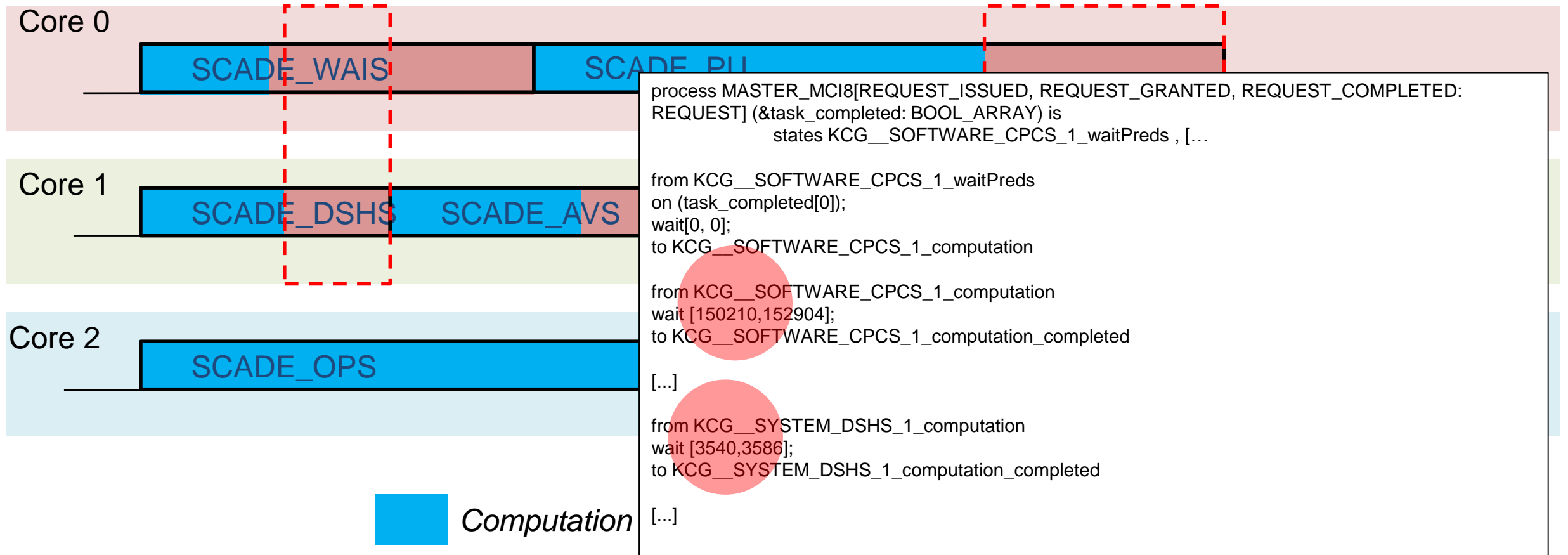
- ❑ It does not take into account physical time
 - If T1 and T2 execute in parallel, but access shared memory at different times, there won't be any interference...

(*) As long as the model is correct...

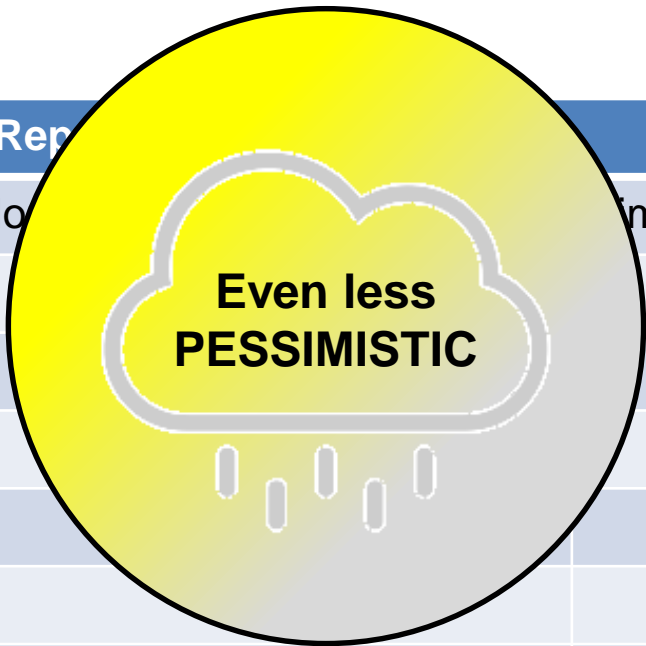
- ❑ The solution is **safe^(*)** and **prevents reporting (some) spurious interferences**



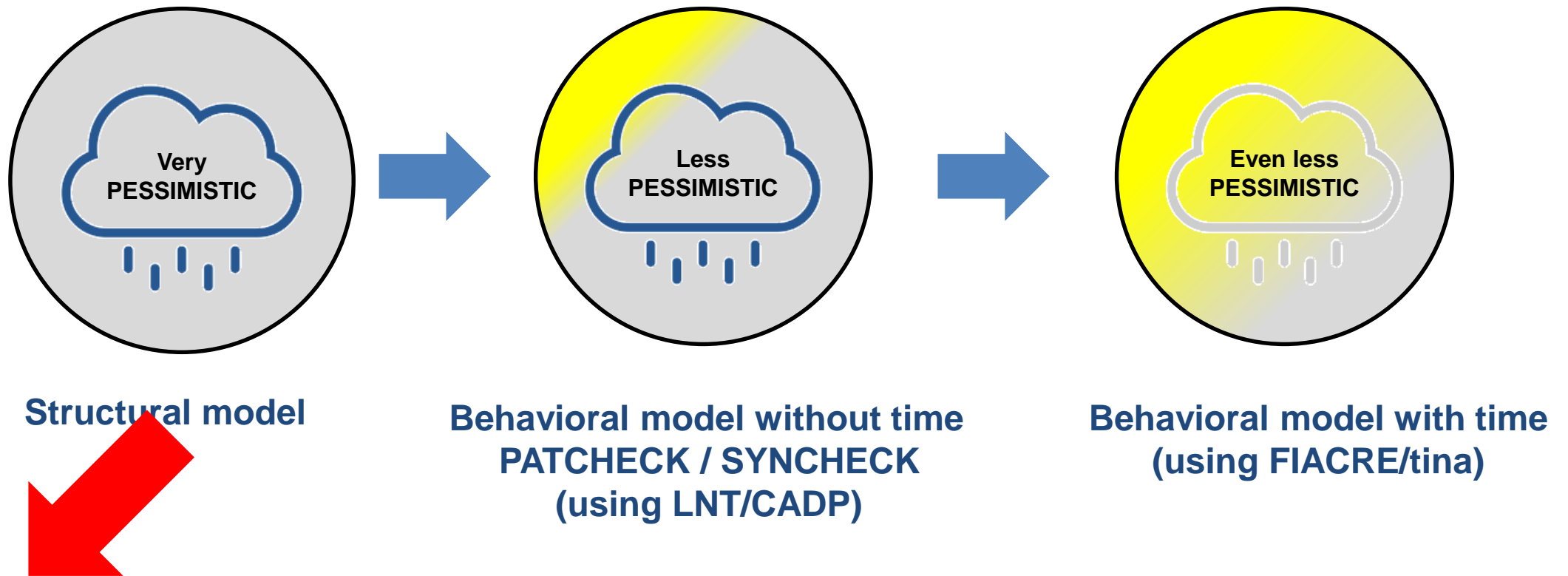
- ❑ The solution is **safe^(*)** and **prevents reporting (some) spurious interferences**



Rep		
TASK	No	med model - TINA
SCADE_MAIN_SOFTWARE_1		NO
SCADE_MAIN_SOFTWARE_2		NO
SCADE_SOFTWARE_CPCS		NO
SCADE_WAIS		YES
SCADE_PNU		YES
SCADE_DSHS	YES	YES
SCADE_AVS	YES	NO
SCADE_OPS	YES	NO
SCADE_ECS	YES	YES



- We have proposed an interference analysis based on **behavioral models** and **model checking** in order to **decrease the pessimism** of pure structural analyses



(*) As long as the model is correct...

Questions?

