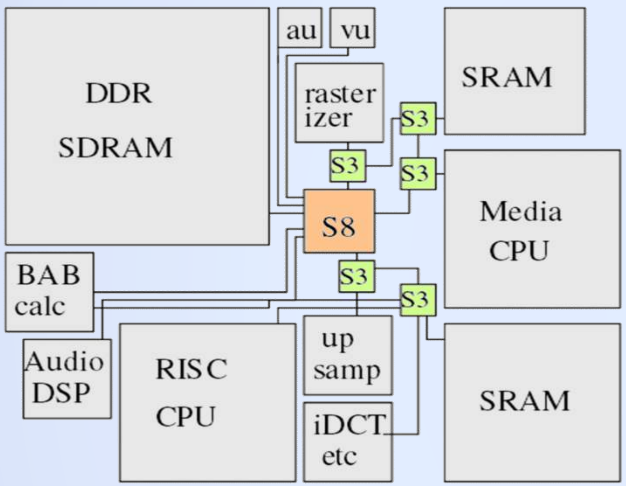
Increasing Impartiality and Robustness in High-Performance N-Way Asynchronous Arbiters

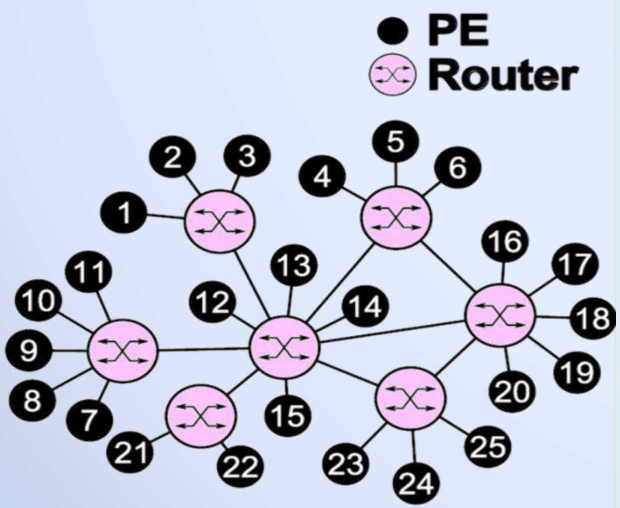
Gabriele Miorandi,
Davide Bertozzi
ENDIF
University of Ferrara
Ferrara, Italy

Steven M. Nowick
Dept. of Computer Science
Columbia University
New York, NY, USA

MOTIVATION

Arbiters are the most critical element to manage a shared resource!





CASE STUDY:
Application specific networks-on-chip
- **Irregular topologies**
- **Asymmetric NxM routers**
- **Heterogeneous routers**

Arbiters are the key elements of the router control logic.

Requirements:

- ***N:1 arbiters***
- ***N ranging from 2 to 10/15***

For larger router sizes, place and route issues make router physical synthesis overly challenging, if not unfeasible.

(A.Pullini et al., “Bringing NoCs to65nm”, IEEE Micro, 12(5):75–85, 2007)

ASYNCHRONOUS ARBITERS

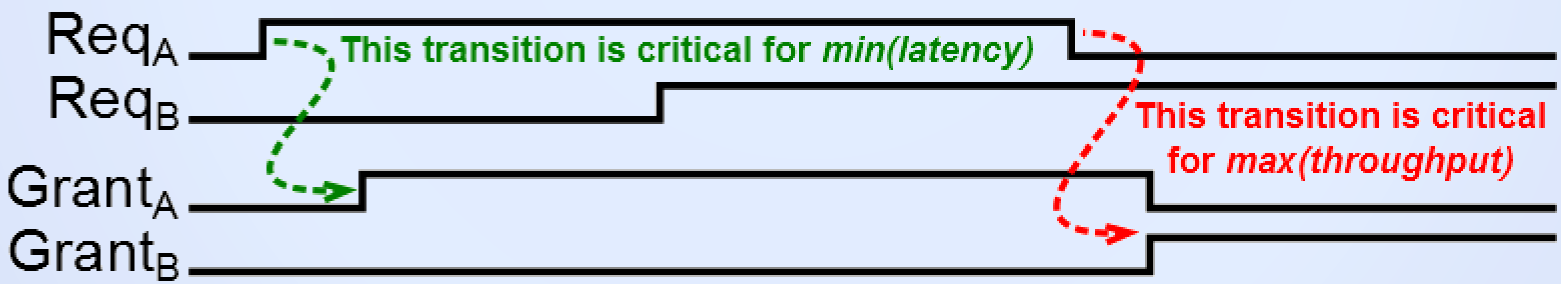
Asynchronous arbiters are more challenging to design than synchronous ones

Inputs may compete and request at arbitrary points in continuous time, unaligned to clock cycles.

METRICS TO EVALUATE AN ASYNCHRONOUS ARBITER

1. **High performance**

- **MIN (Latency) to access shared resource**
- **MAX (Throughput) when switching between active requests**



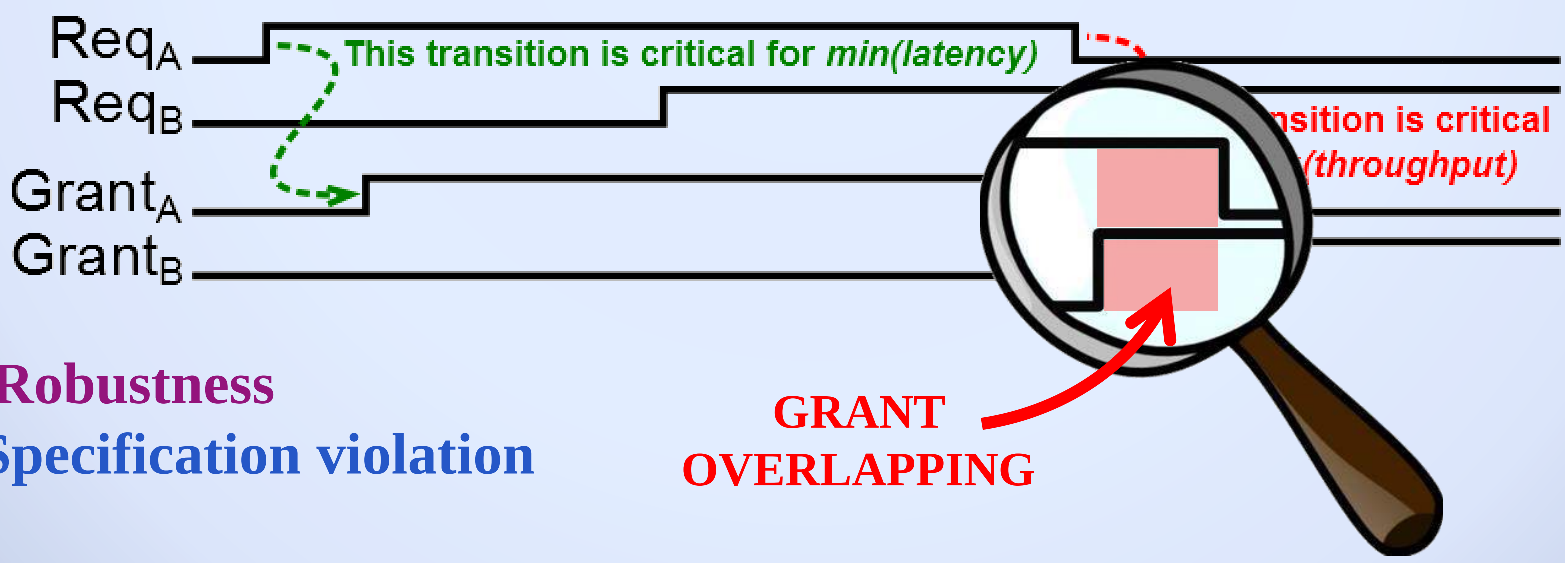
ASYNCHRONOUS ARBITERS

Asynchronous arbiters are more challenging to design than synchronous ones

Inputs may compete and request at arbitrary points in continuous time, unaligned to clock cycles.

METRICS TO EVALUATE AN ASYNCHRONOUS ARBITER

1. **High performance**
- **MIN (Latency) to access shared resource**
- **MAX (Throughput) when switching between active requests**



2. **Robustness**
- **Specification violation**

ASYNCHRONOUS ARBITERS

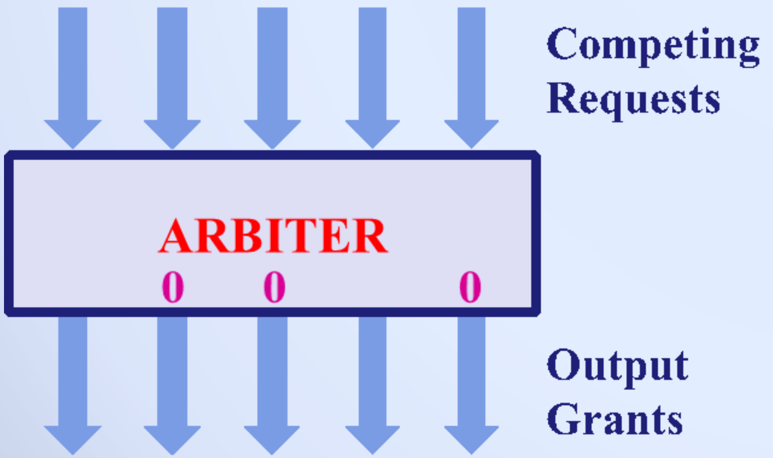
Asynchronous arbiters are more challenging to design than synchronous ones

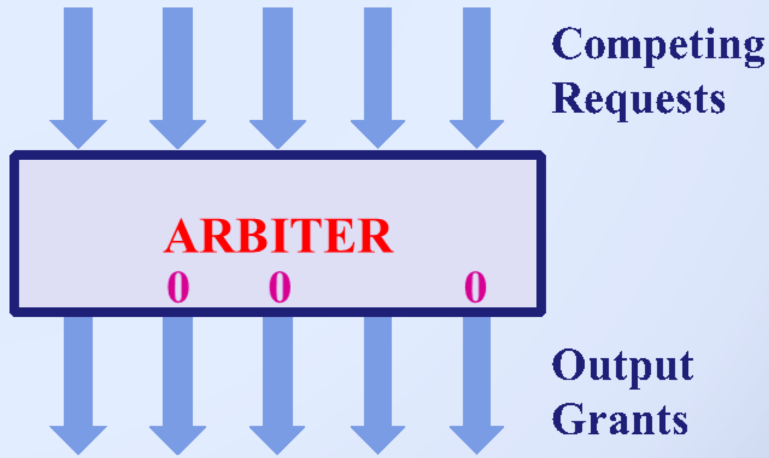
Inputs may compete and request at arbitrary points in continuous time, unaligned to clock cycles.

METRICS TO EVALUATE AN ASYNCHRONOUS ARBITER

3. Impartiality

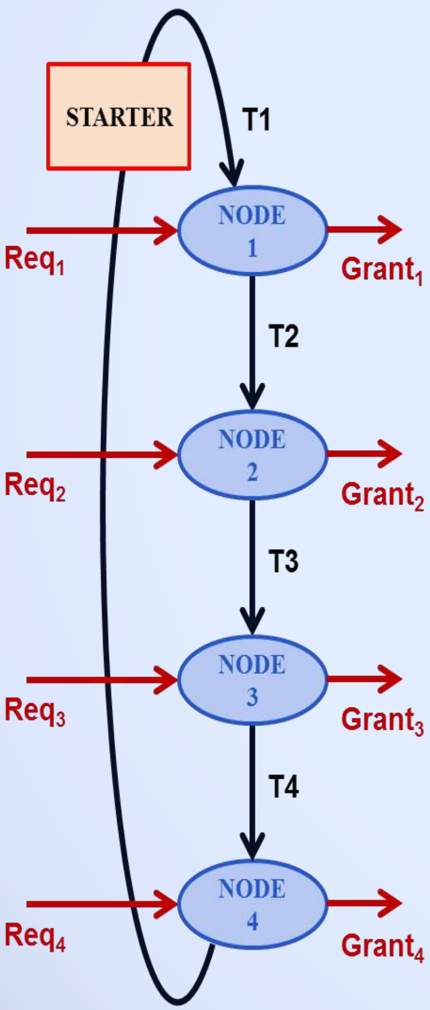
- **All requests should have the same win rate (fairness)**
- **All requests should have the same acquisition latency**





COMMON ASYNCHRONOUS ARBITERS

TOKEN RING



RELEVANT PREVIOUS WORK

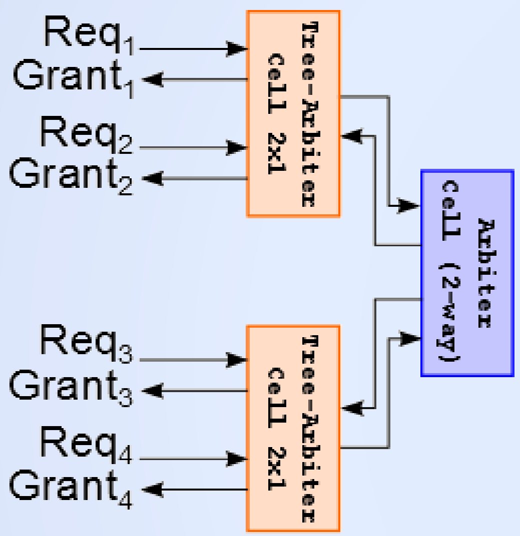
T. Singh and A. Taubin, "A highly scalable GALS crossbar using token ring arbitration" IEEE Design & Test of Computers, vol. 24:5, pp. 464-472, 2007.

- ✓ **This is the reference Round-Robin solution**
- ✓ **Scaled-up versions are easy to design**

- **Worst case latency is severe**
- **Poor performance scalability**
- **Large gap between Min/Max performance**

COMMON ASYNCHRONOUS ARBITERS

TREE



Requests pass through as few as a logarithmic number of cells in order to be granted

For performance and scalability reasons, we build our novel N-way asynchronous arbiters on top of a tree structures!

RELEVANT PREVIOUS WORK

1. A. Yakovlev, A. Petrov and L. Lavagno, "A low latency asynchronous arbitration circuit," IEEE Transactions on VLSI Systems, vol. 2:3, pp. 372-377, 1994.

   Yields robustness at the cost of performance

2. S.R. Naqvi and A. Steininger, "A tree arbiter cell for high speed resource sharing in asynchronous environments" ACM/IEEE DATE Conference, 2014.

   Optimized for throughput at the cost of latency and robustness. Has timing assumptions.

3. A. Ghiribaldi, D. Bertozzi and S.M. Nowick, "A transition-signaling bundled data NoC switch architecture for cost-efficient GALS multicore systems" ACM/IEEE DATE Conference, pp. 332-337, 2013. (this is our baseline architecture)

   Overly simple and performance-efficient design at the cost of robustness

All tree arbiters suffer from poor impartiality if number of inputs is not a power of two.

CONTRIBUTION OF THIS WORK

Most N-way asynchronous arbiters have serious drawbacks in one or more cost/reliability metrics

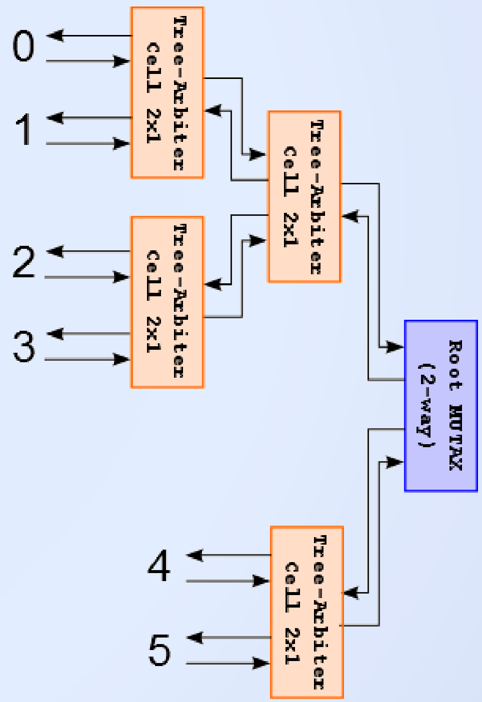
Our contribution in this context:

1. **We provide a new high-performance and scalable N-way asynchronous arbiter design, with increased robustness and impartiality in treatment of their inputs.**
   - **A novel rebalanced and flattened tree architecture.**
   - **A novel 3-way arbiter with highly equalized latency response.**
     - ✓ ***Both standalone and building block of the 3-way tree arbiter cell (3x1 TAC).***
   - **A novel 4-way tree arbiter cell (4x1 TAC), with simple recursive structure.**
2. **We present an extensive cross design evaluation of a wide range of N-way arbiters, including the newly-proposed one, across a variety of metrics, to evaluate their suitability.**
   - **Formal verification for QDI-ness has been performed using a state-of-the-art verification framework.(Workcraft, from U-Newcastle).**

THE PROBLEM

Tree arbiter are optimal only for power of 2 dimensions.

Unbalanced tree structures are affected by the following problems:



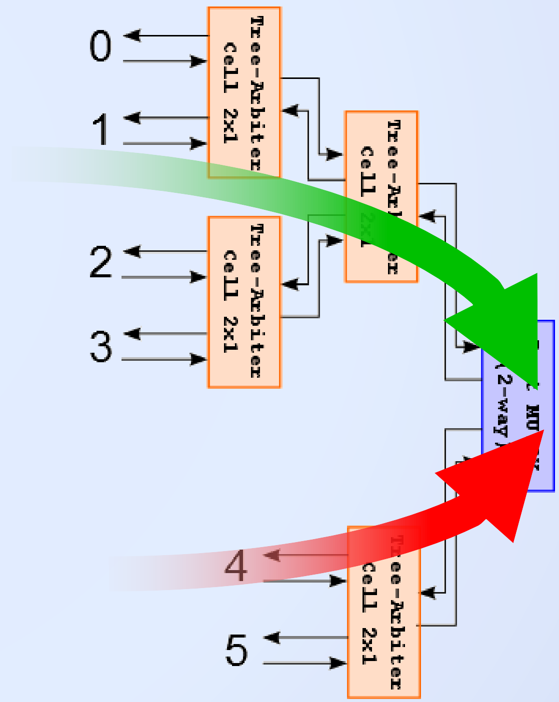
THE PROBLEM

Tree arbiter are optimal only for power of 2 dimensions.

Unbalanced tree structures are affected by the following problems:

Client inequality

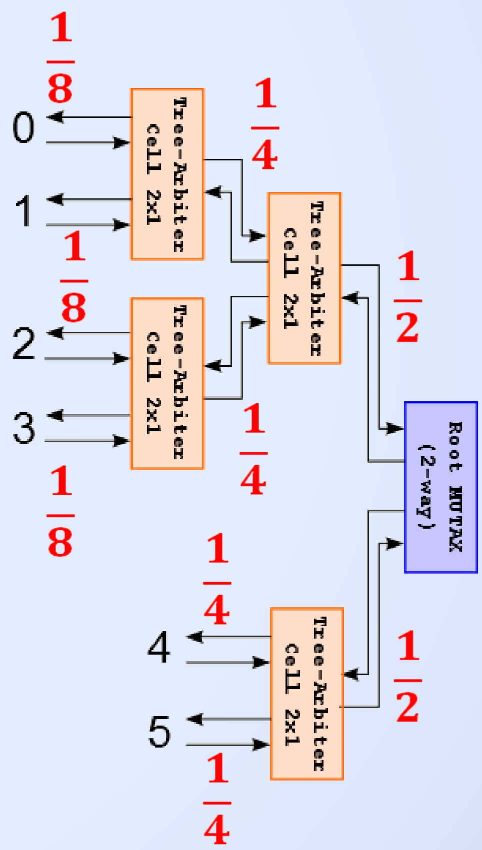
- For other dimensions, impartiality is experienced:
  - No latency equalization



THE PROBLEM

Tree arbiter are optimal only for power of 2 dimensions.

Unbalanced tree structures are affected by the following problems:



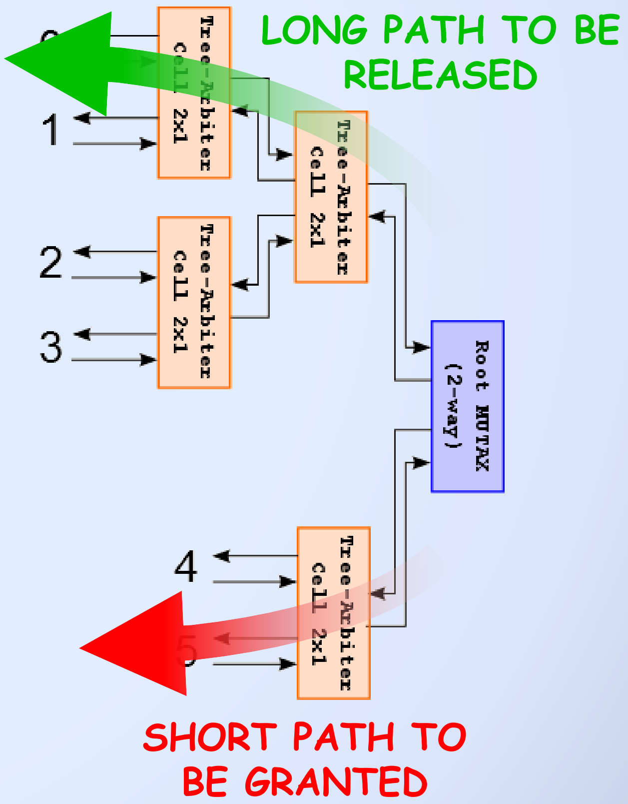
Client inequality

- **For other dimensions, impartiality is experienced:**
  - No latency equalization
  - No equal win rate

THE PROBLEM

Tree arbiter are optimal only for power of 2 dimensions.

Unbalanced tree structures are affected by the following problems:

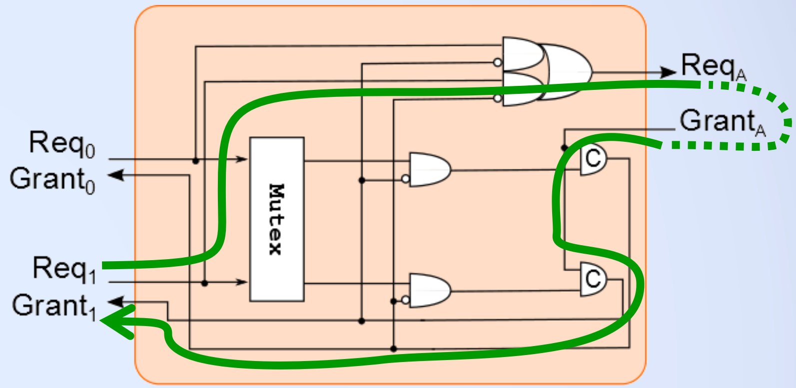


Client inequality

- For other dimensions, impartiality is experienced:
  - No latency equalization
  - No equal win rate
- For other dimensions, grant overlapping may be experienced.

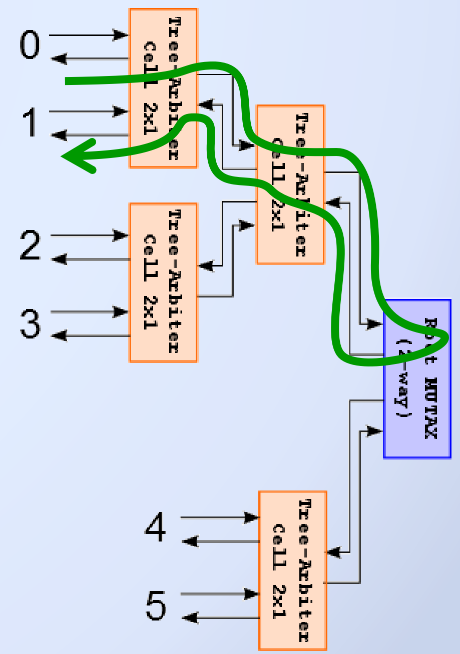
THE PROBLEM

Unbalanced tree structures are affected by the following problems:



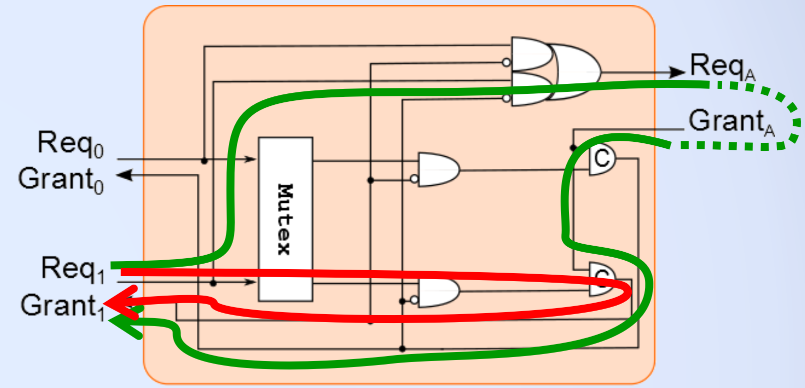
Critical path imbalance

- Performance will be driven by the **global** critical path (through the root)
- This effect gets worse for larger arbiters with many layers of TACs (global critical path gets even longer)



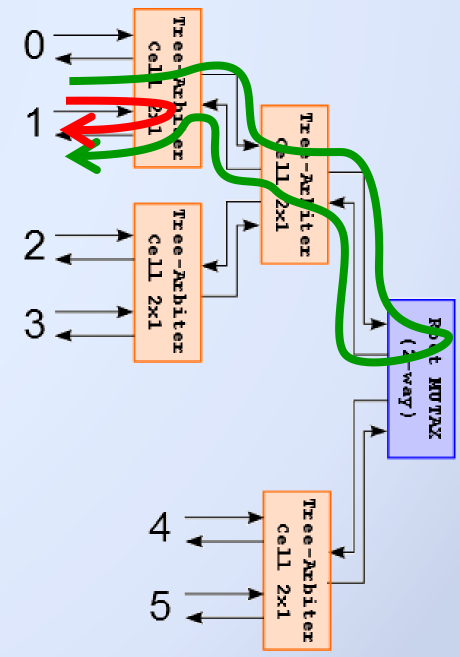
THE PROBLEM

Unbalanced tree structures are affected by the following problems:



Critical path imbalance

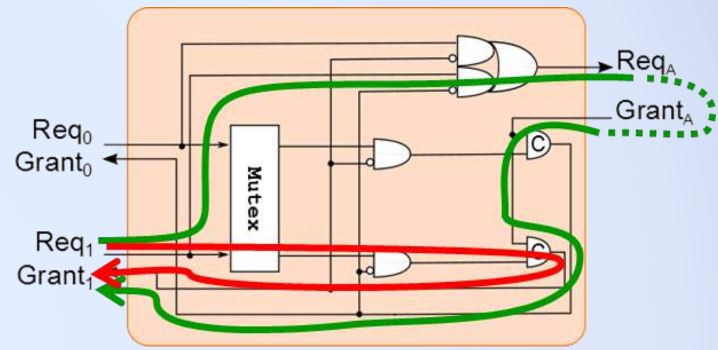
- Performance will be driven by the **global** critical path (through the root)
- This effect gets worse for larger arbiters with many layers of TACs (global critical path gets even longer)
- While the **local** critical path (within the leaf TAC) is short



IDEA

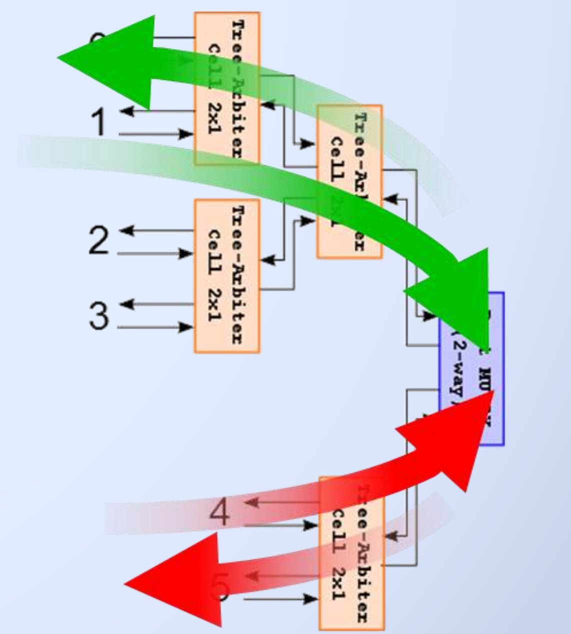
Overall, we identified some structural imbalances which lead to unfair performance and less robustness

Green dominates the worst critical path



Critical path before rebalancing

UNFAIR SYSTEM

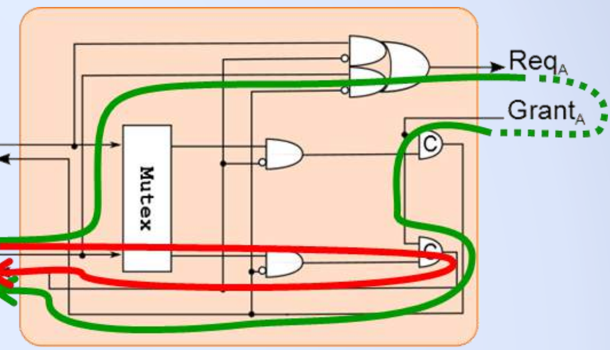


IDEA

Overall, we identified some structural imbalances which lead to unfair performance and less robustness

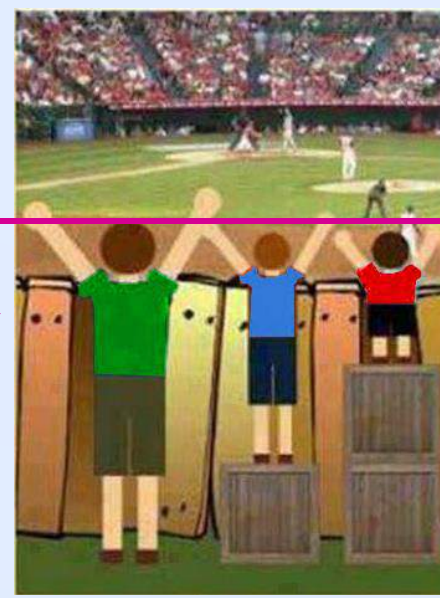
IDEA:

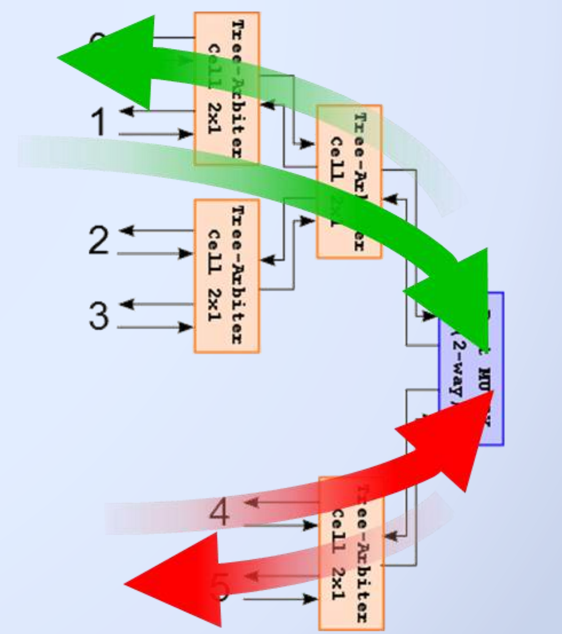
Rebalance the system, moving complexity where there is not, in order to simplify the worst critical operations!



Critical path before rebalancing

Critical path after rebalancing

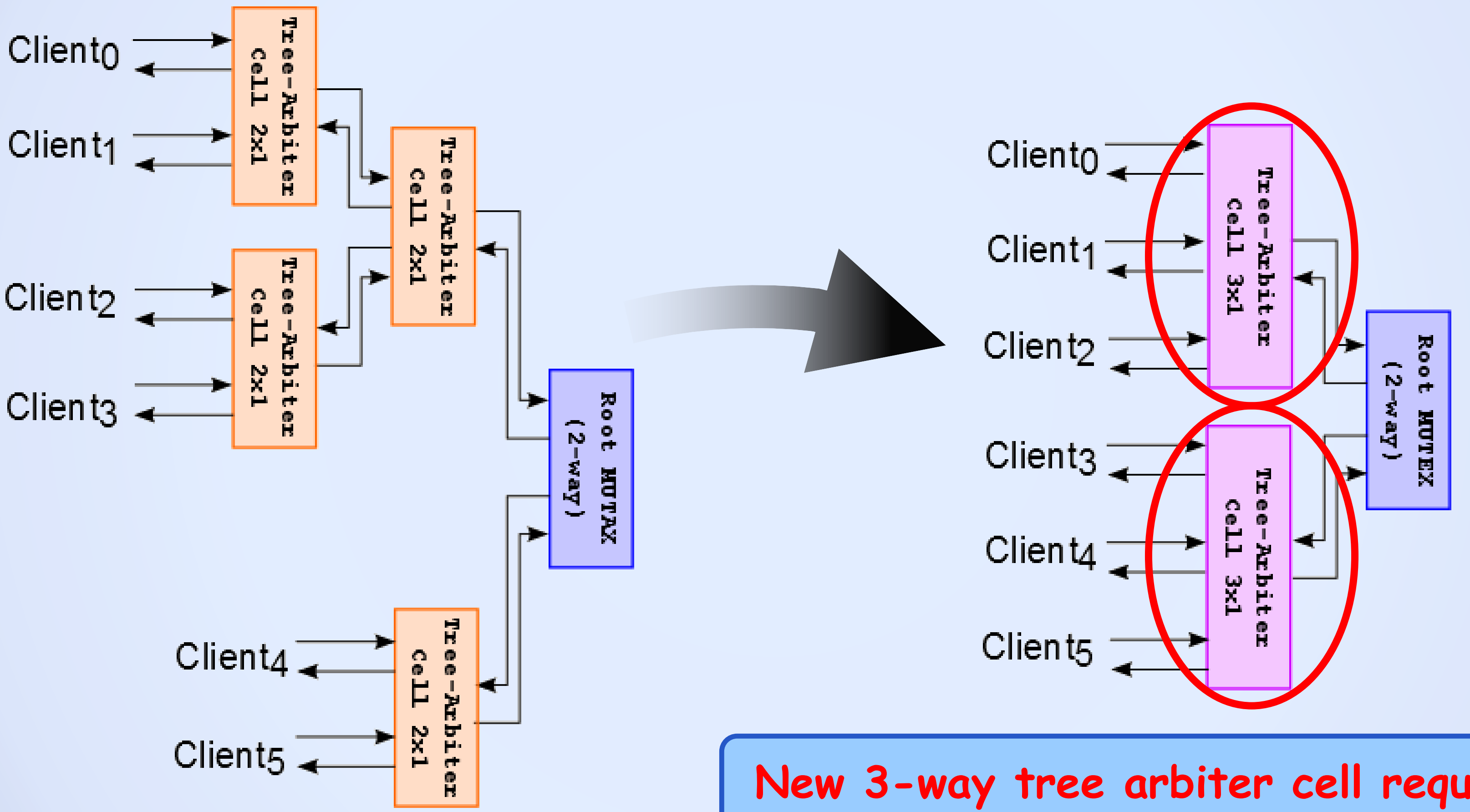




UNFAIR SYSTEM

REBALANCED SYSTEM

IDEA: REBALANCED ARCHITECTURE



New 3-way tree arbiter cell required

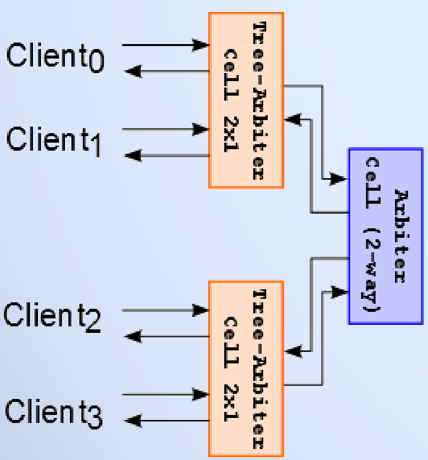
It must be fair (cannot be implemented with traditional tree structure – requires new engineering effort)

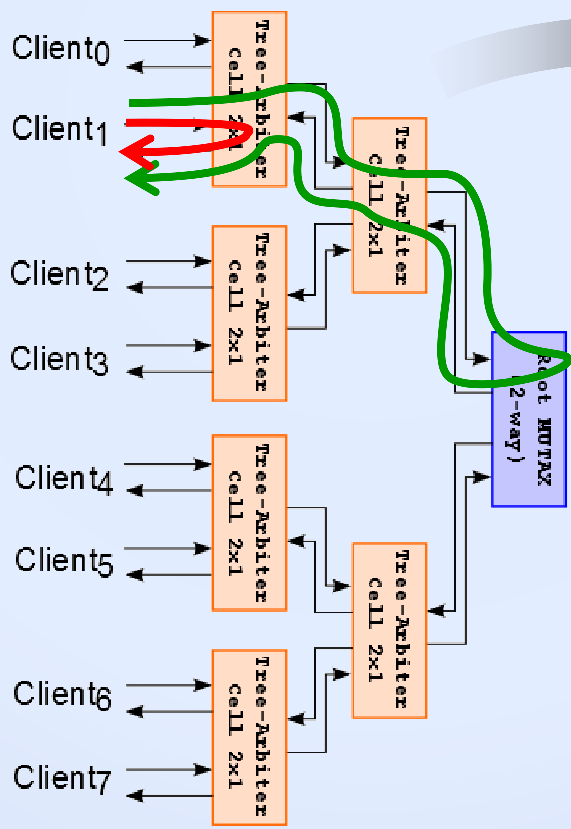
IDEA: REBALANCED ARCHITECTURE

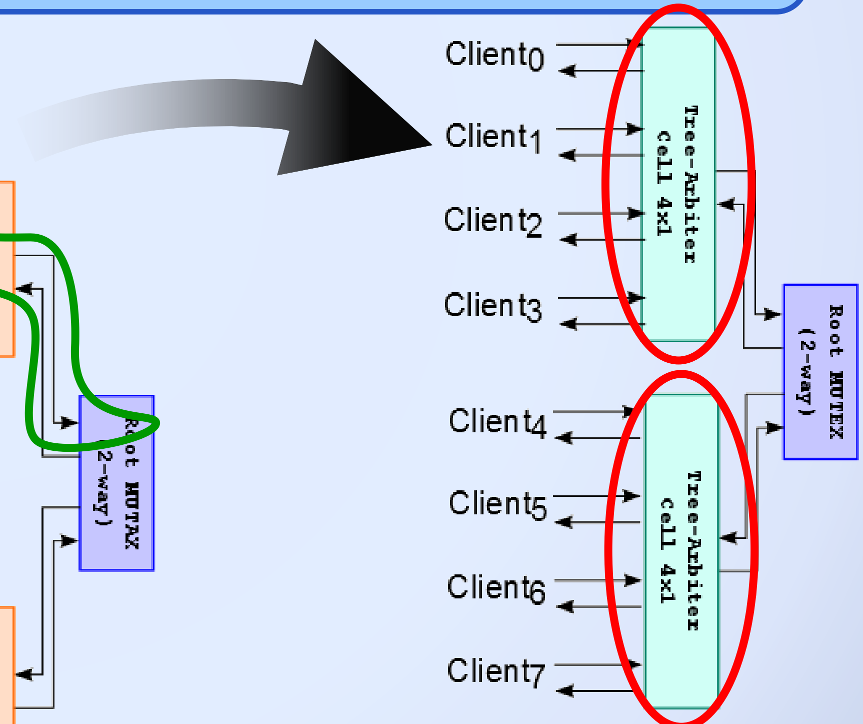
Power-of-two tree arbiters are apparently already balanced…
…from the structural viewpoint, but not from the critical path viewpoint

We can rebalance local vs. global critical path by moving complexity to the leaves

Proposed 4-way arbiter is equal to baseline.



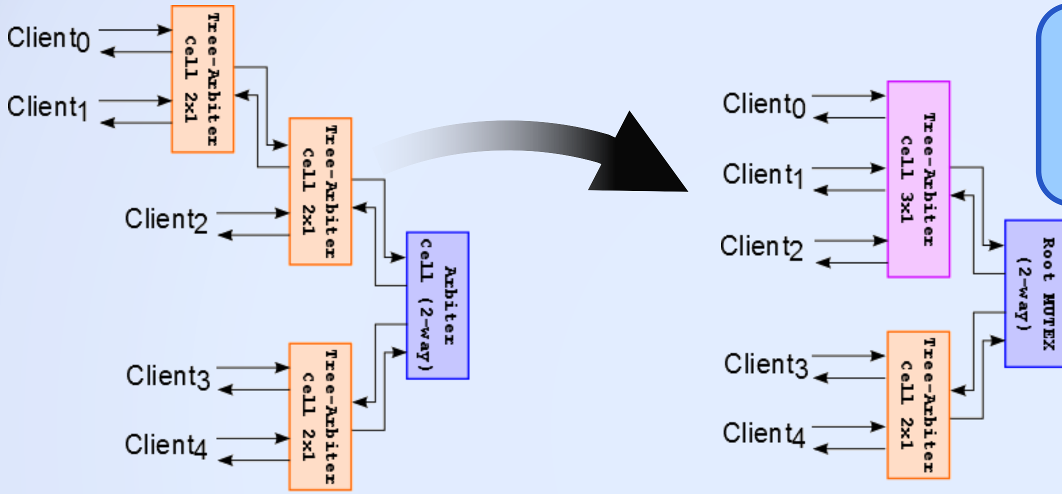




New 4-way tree arbiter cell required

IDEA: REBALANCED ARCHITECTURE

There are still suboptimal solutions (5-way and 7-way), yet…



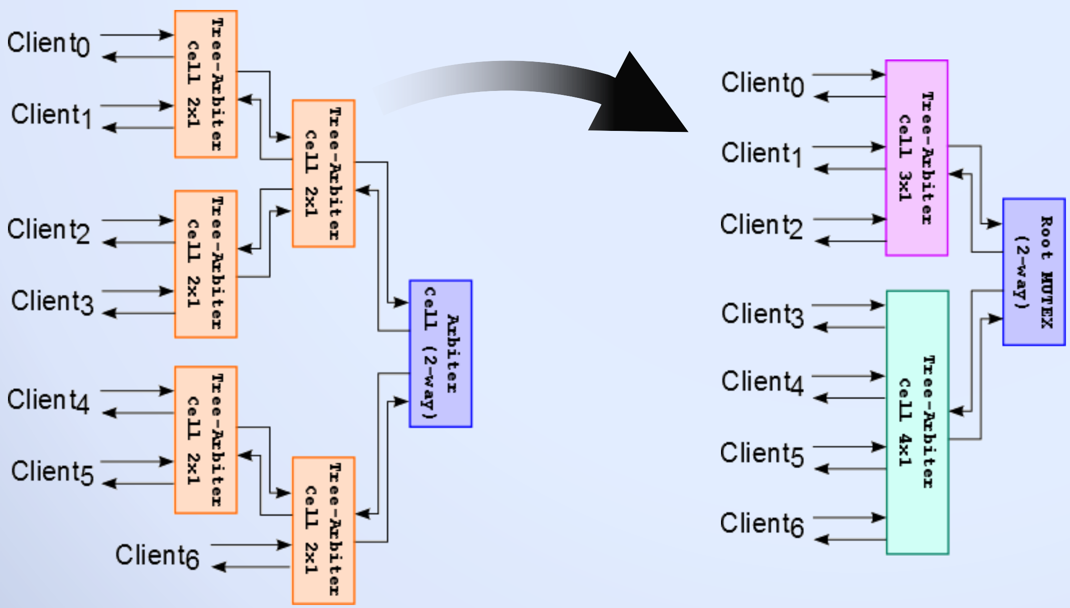
…unbalancing issues are strongly mitigated with respect to standard tree arbiters

Win rate for 5-way are

$$3 \times \frac{1}{6} + 2 \times \frac{1}{4}$$

instead of

$$2 \times \frac{1}{8} + 3 \times \frac{1}{4} \quad (\text{ideal } \frac{1}{5})$$



Win rate for 7-way are

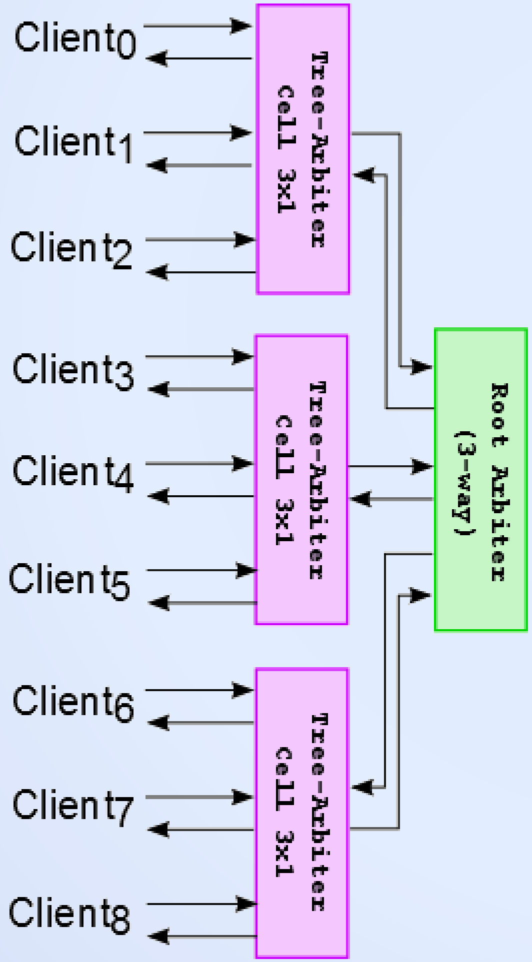
$$3 \times \frac{1}{6} + 4 \times \frac{1}{8}$$

instead of

$$6 \times \frac{1}{8} + 1 \times \frac{1}{4} \quad (\text{ideal } \frac{1}{7})$$

IDEA: REBALANCED ARCHITECTURE

An interesting hybrid solution: 9-way arbiter is perfectly balanced if it is built using 3-way arbiters only…



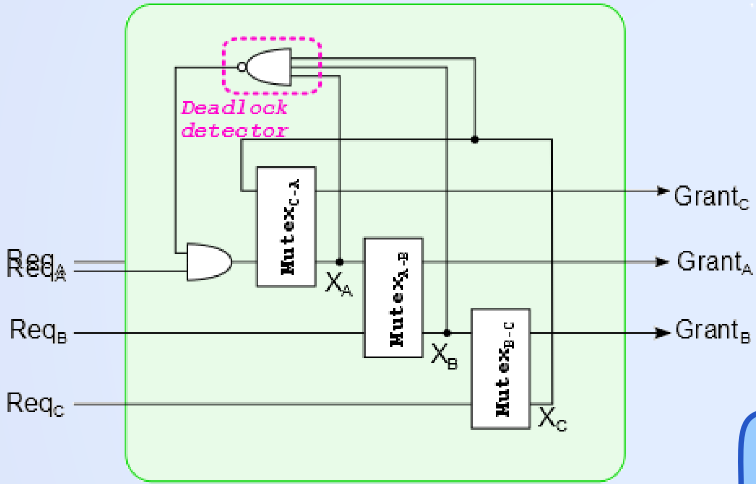
Fair 3-way arbiters are required for the root as well as for the 3x1 TACs

In this case we are using a "complex" root for the sake of rebalancing.

MISSING ITEMS: 3-WAY ARBITER

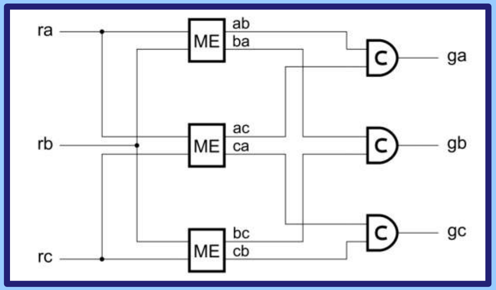
The proposed 3-way arbitration core contains three mutexes connected in a ring-like structure…

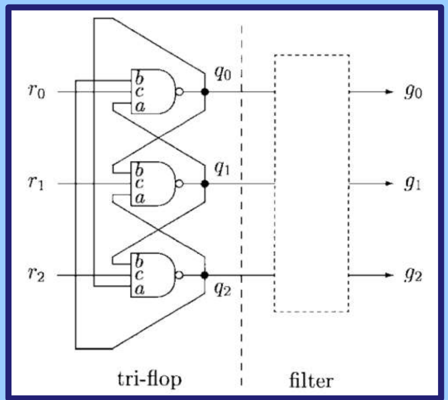
- **To be used in 3x1TACs to build up larger arbiters**



Arbiter may deadlock when three requests come and each one wins the first ME. (X_A,X_B,X_C=1)

We selectively kill one of the inputs. Latency equalization is maintained at a low implementation cost.





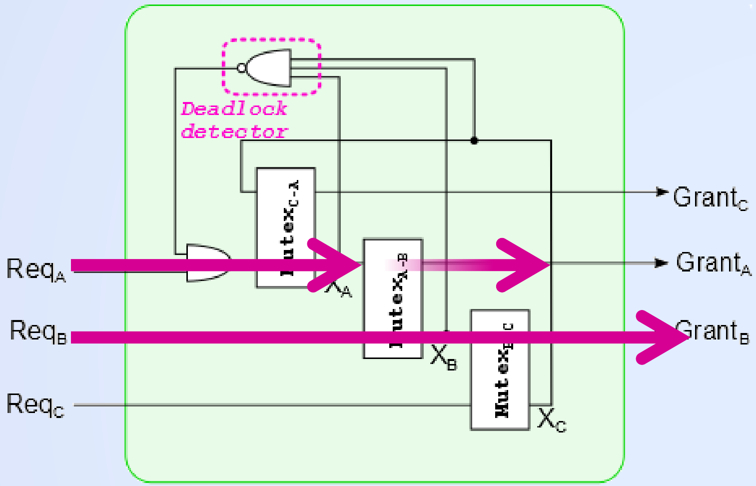
PREVIOUS 3-WAY ARBITER

Fair 3-way arbiter previously presented in the literature may deadlock during transient operation or may fail because of metastability issues.

A.Mokhov, V. Khomenko and A. Yakovlev, "Flat arbiters," Fundamenta Informat-icae, no. 1-2, pp. 63-90, 2011.
C.H. van Berkel and C.E. Molnar, "Beware the three-way arbiter," IEEE Journalof Solid-State Circuits, vol. 34:6, pp. 840-848, 1999.

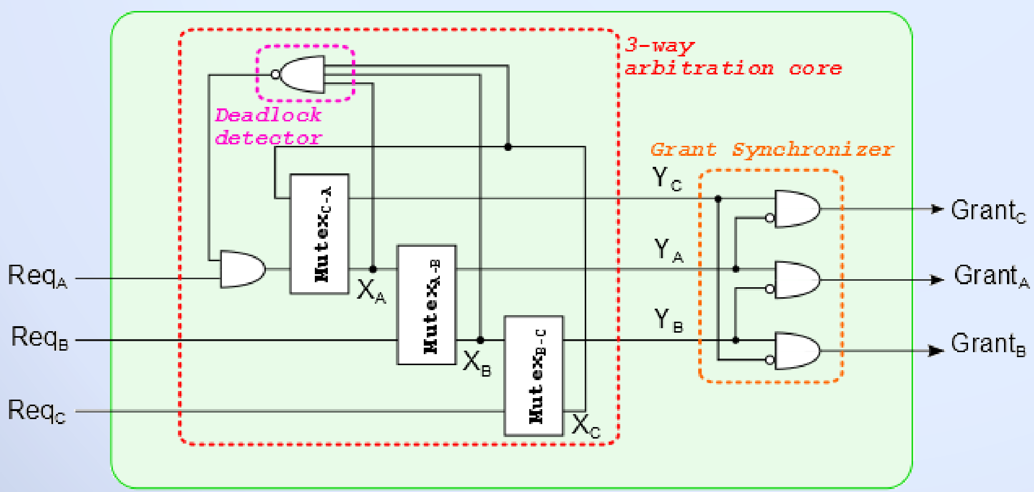
MISSING ITEMS: 3-WAY ARBITER

The same circuit cannot be used as is for standalone 3-way arbiters or for 3-way root cells… since it suffers from grant overlapping.



3-way arbitration core

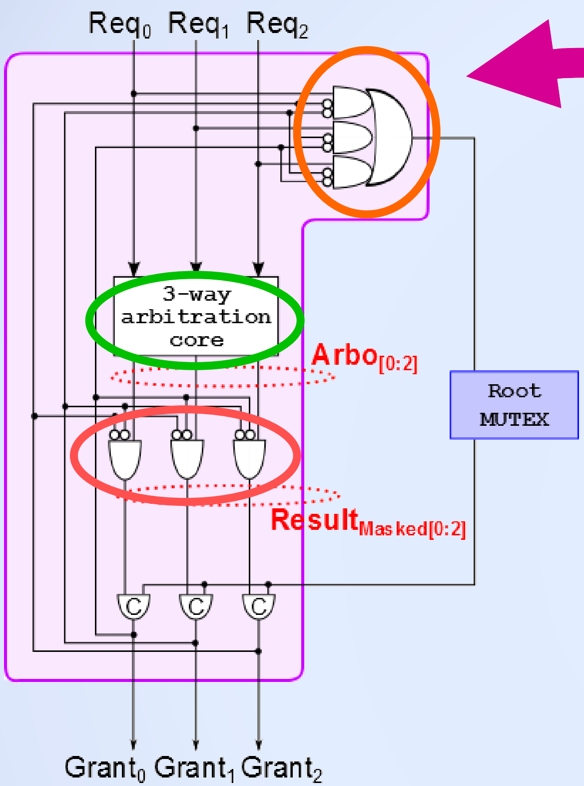
For example, in the transient while client B is released and client A is granted.



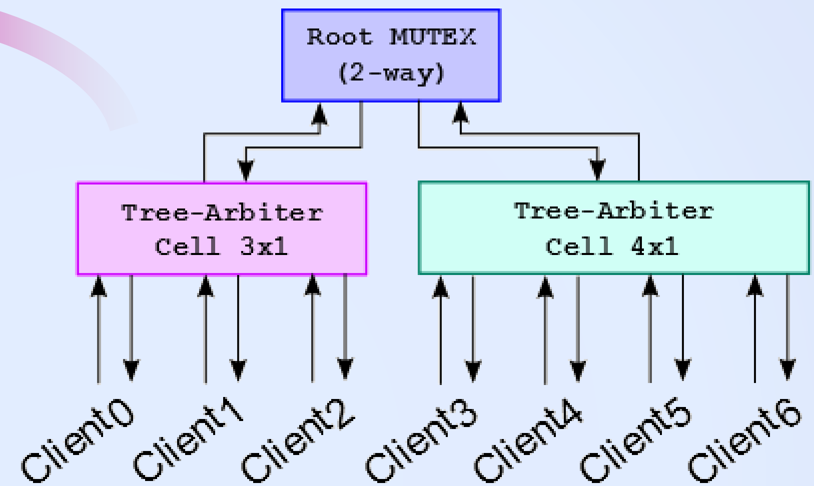
3-way standalone arbiter

The basic 3-way arbitration core has been augmented with a grant synchronizer to significantly mitigate grant overlapping.
($Y_B\downarrow$ is precondition for $Grant_A\uparrow$)

MISSING ITEM: 3x1 TAC



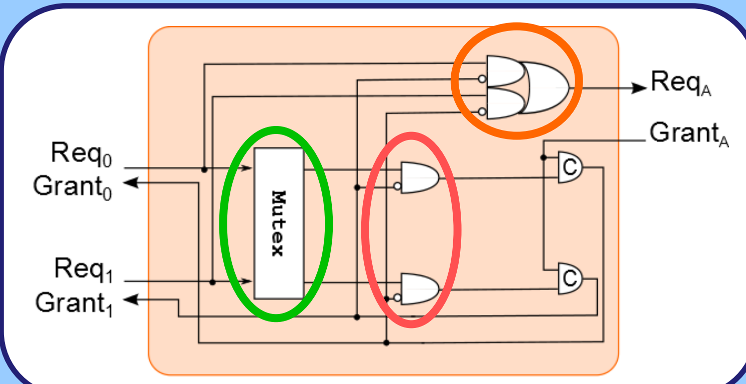
REBALANCED 7-WAY ARBITER



We proved this circuit is QDI using Workcraft tools from Univ. Newcastle

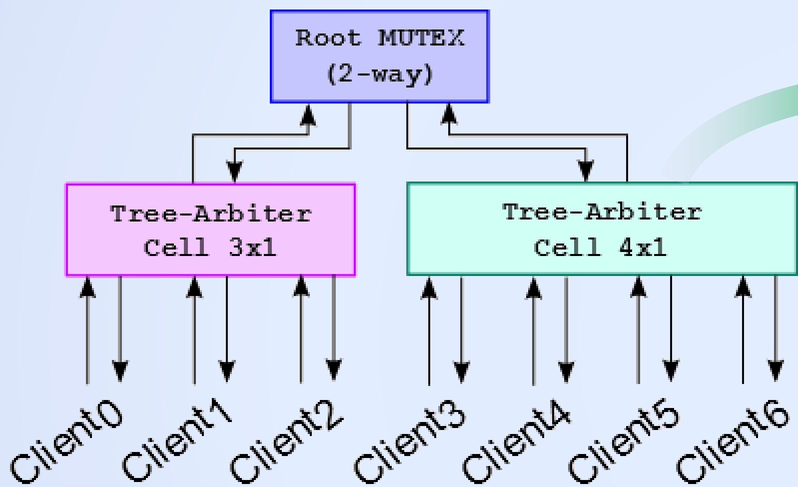
INTERNAL ARCHITECTURE IS SIMILAR TO THE BASELINE 2x1 TAC

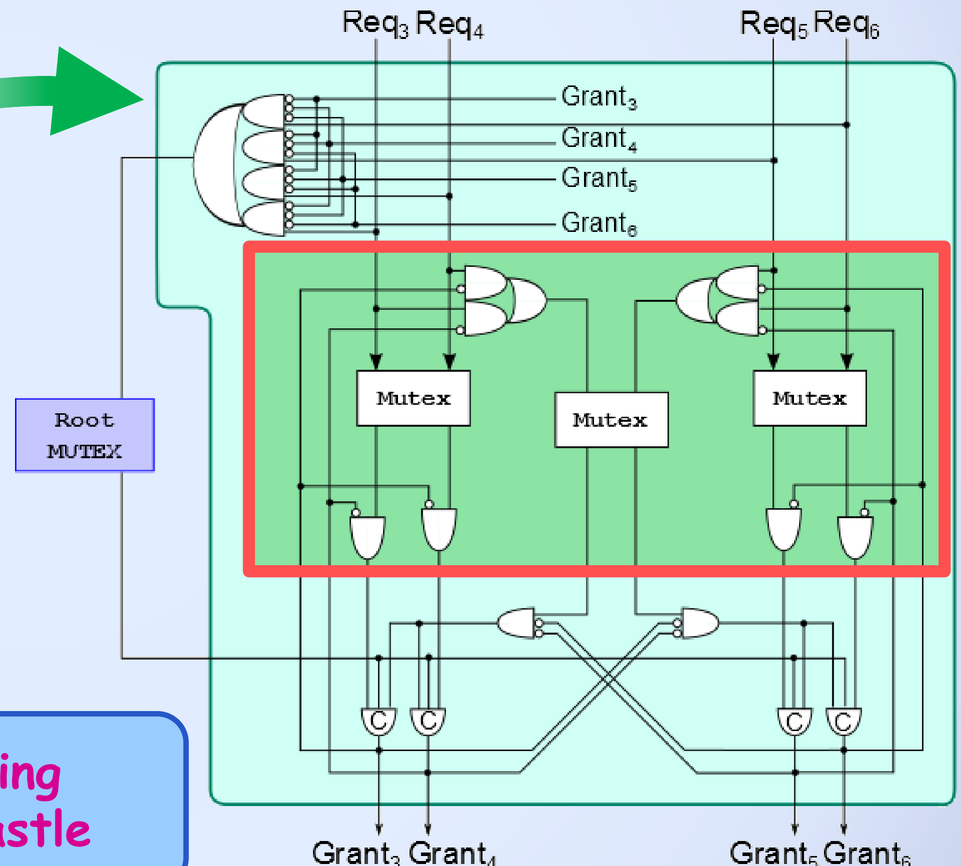
Our 3-way arbitration core is used in place of the 2-way mutex



MISSING ITEM: 4x1 TAC

REBALANCED 7-WAY ARBITER



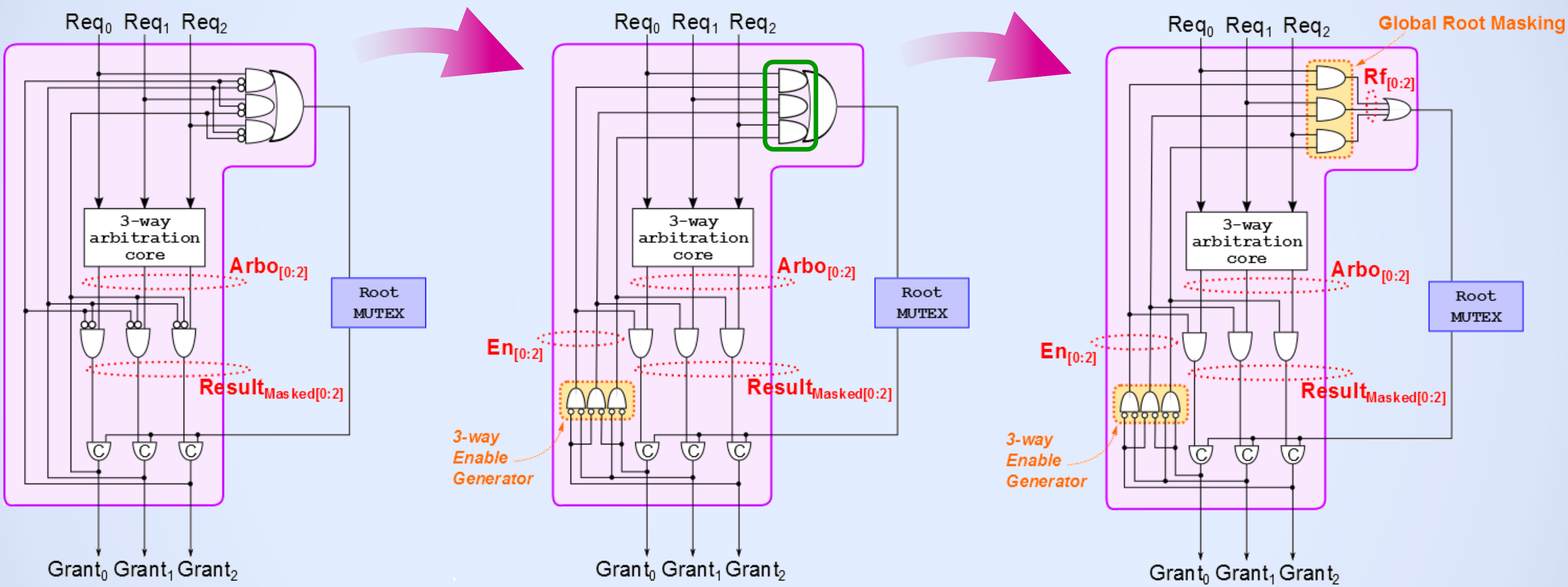


A BASELINE 4-way arbiter is used in a recursive structure.

We proved this circuit is QDI using Workcraft tools from Univ. Newcastle

GATE DECOMPOSITION

Simple gate level decomposition has been applied because the target technology library does not have such complex gates.



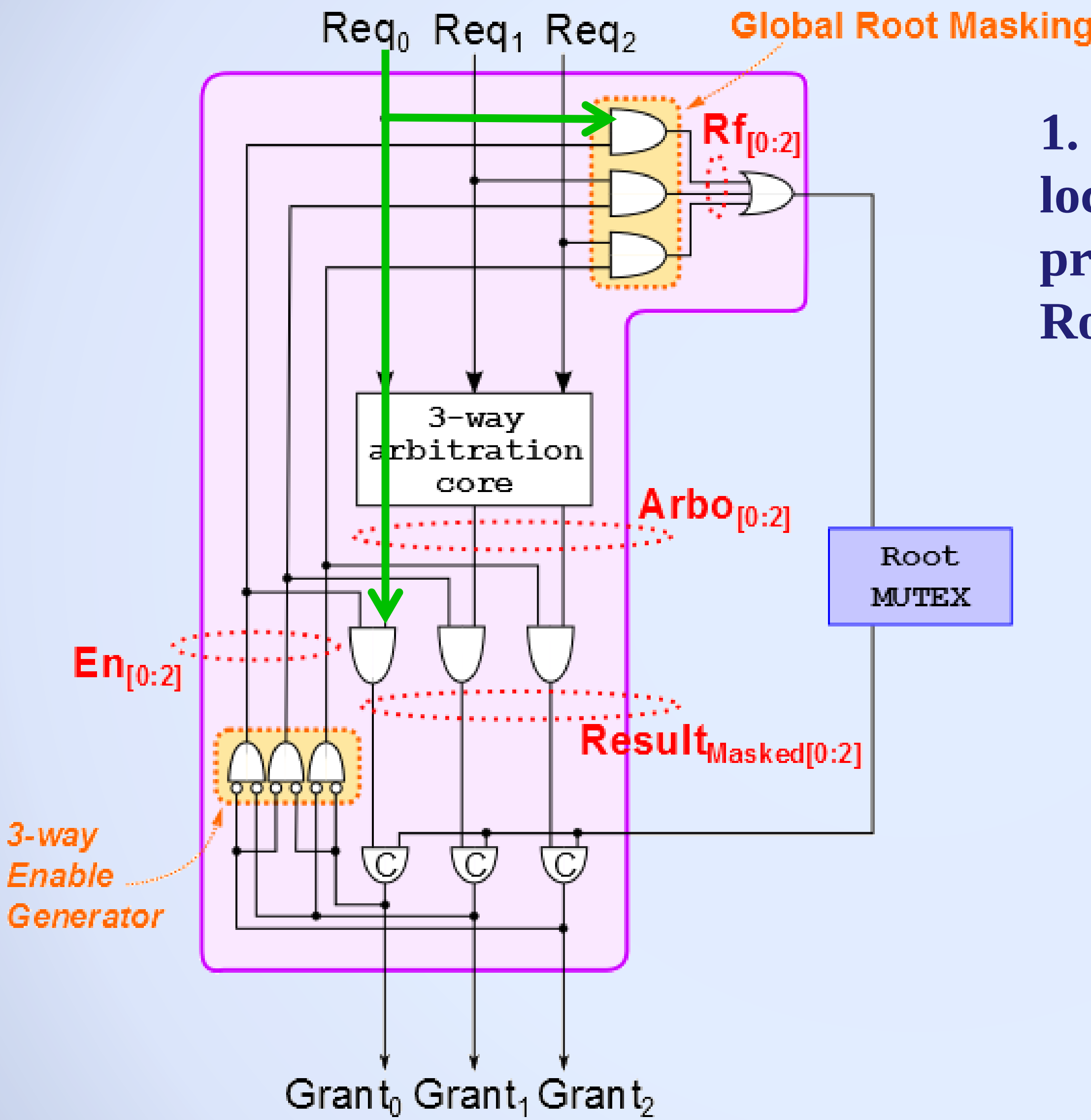
Inverted inputs are extracted into an Enable Generator (NOR gates)

Note how this fact reduces the global critical path, since 2-way AND gates are used

Complex AO gates are separated into simpler gates

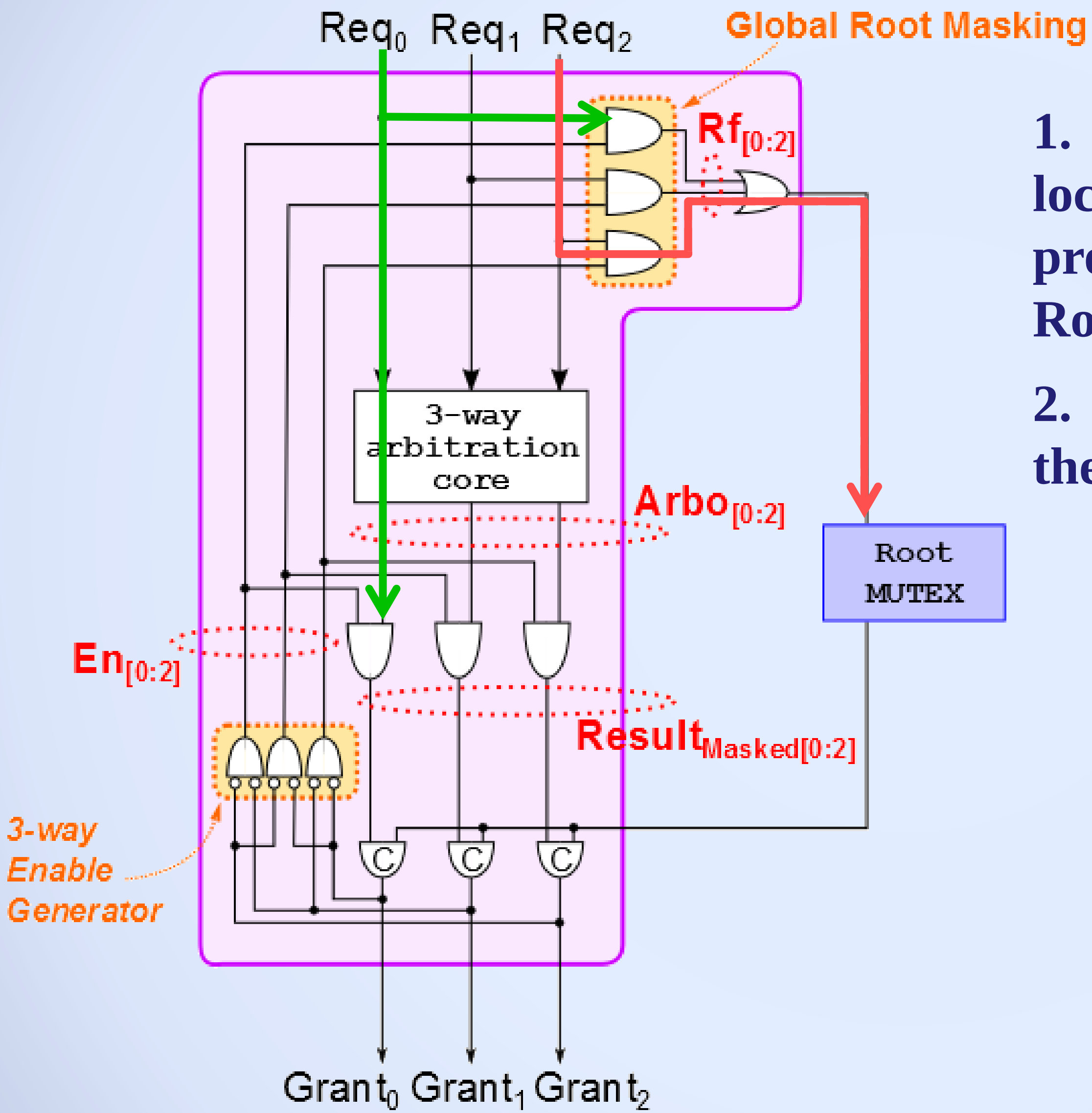
This gate level decomposition gives rise to **reasonable timing assumptions**

MAIN TIMING ASSUMPTION



1. Req_0 comes, acquires the local mutex but gets stuck while propagating through the Global Root Masking

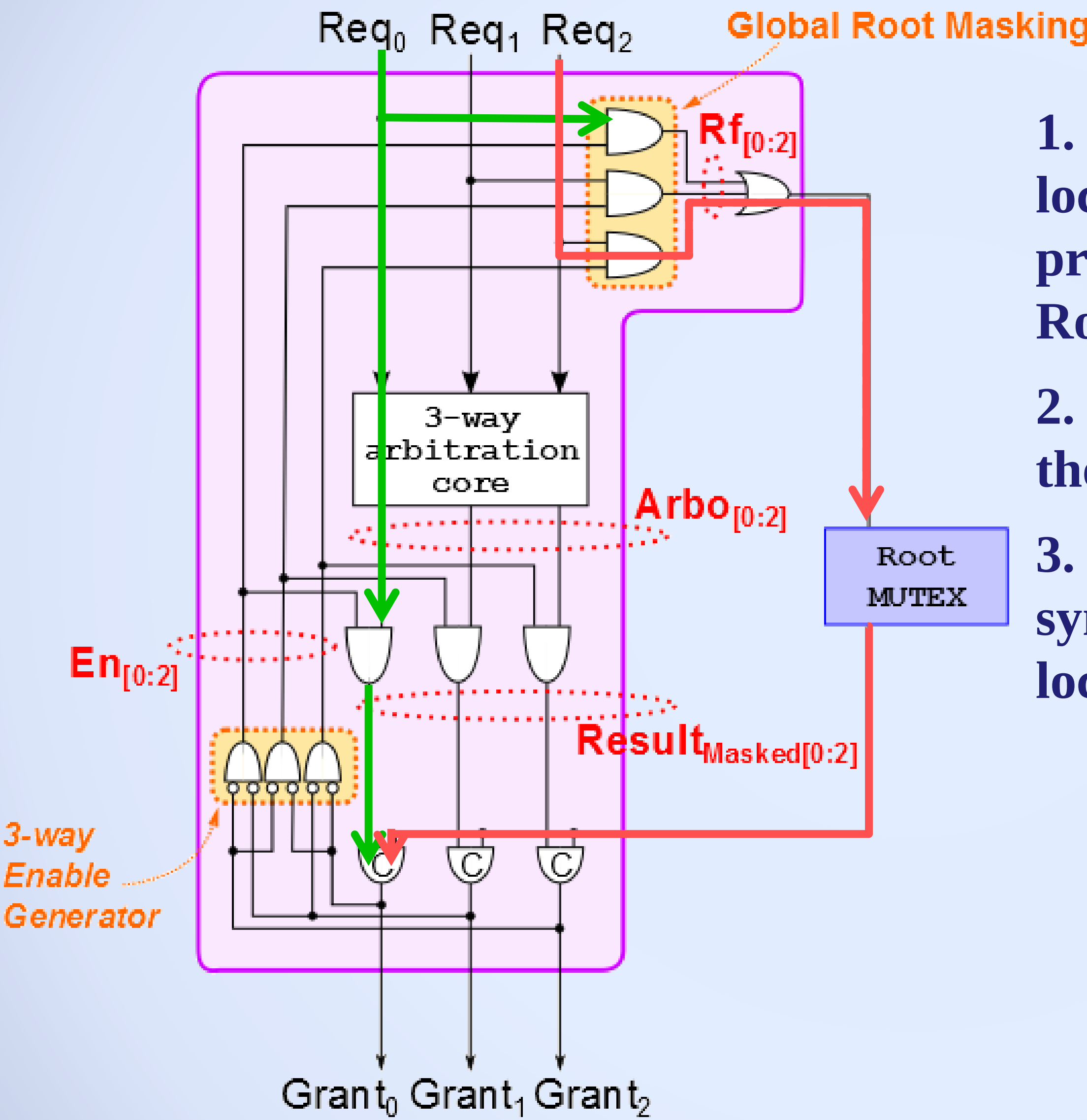
MAIN TIMING ASSUMPTION



1. Req_0 comes, acquires the local mutex but gets stuck while propagating through the Global Root Masking

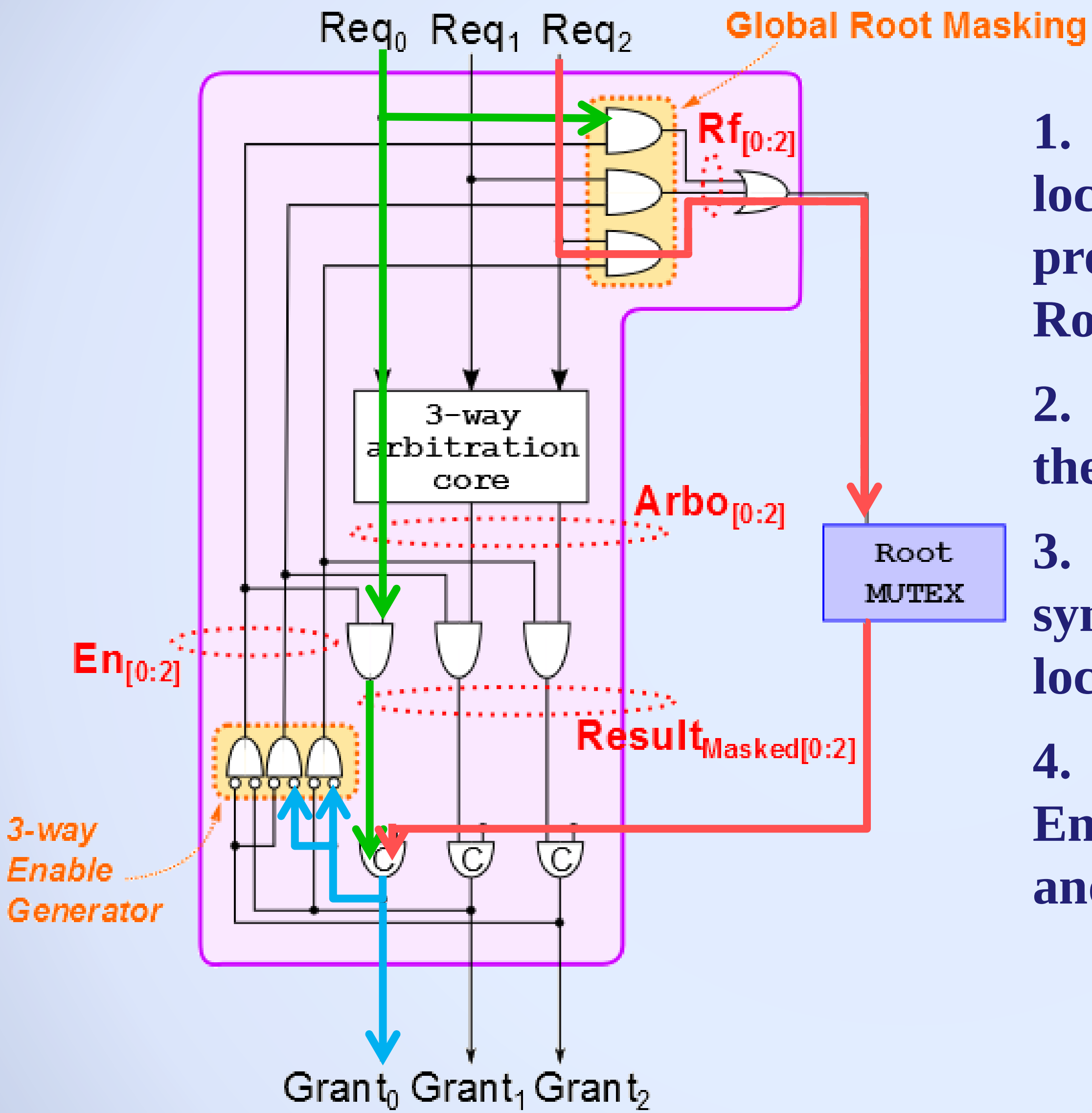
2. Req_2 comes and propagates to the root

MAIN TIMING ASSUMPTION



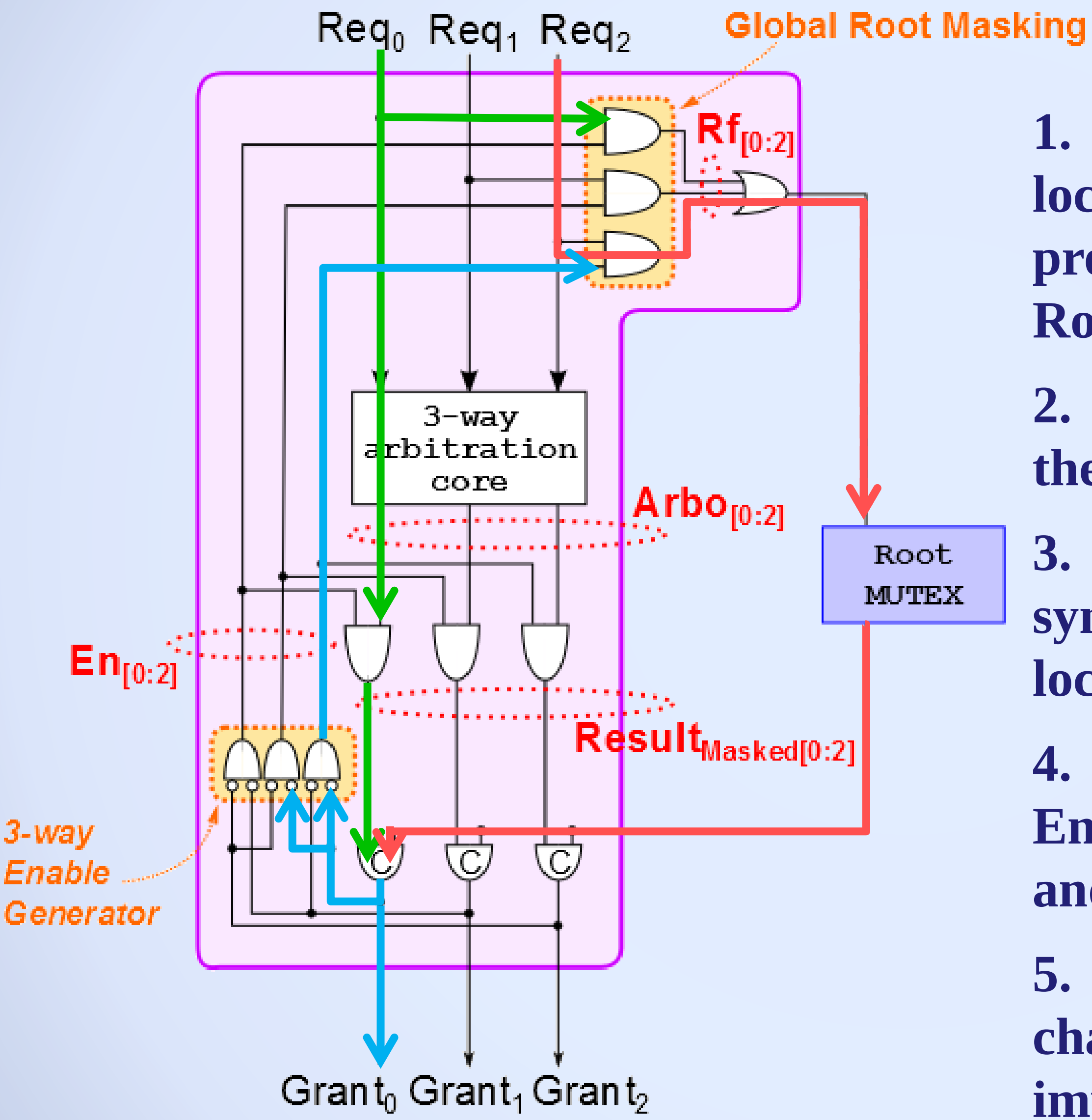
1. Req_0 comes, acquires the local mutex but gets stuck while propagating through the Global Root Masking
2. Req_2 comes and propagates to the root
3. The MullerC Element synchronizes the requests from the local and the root arbiter

MAIN TIMING ASSUMPTION



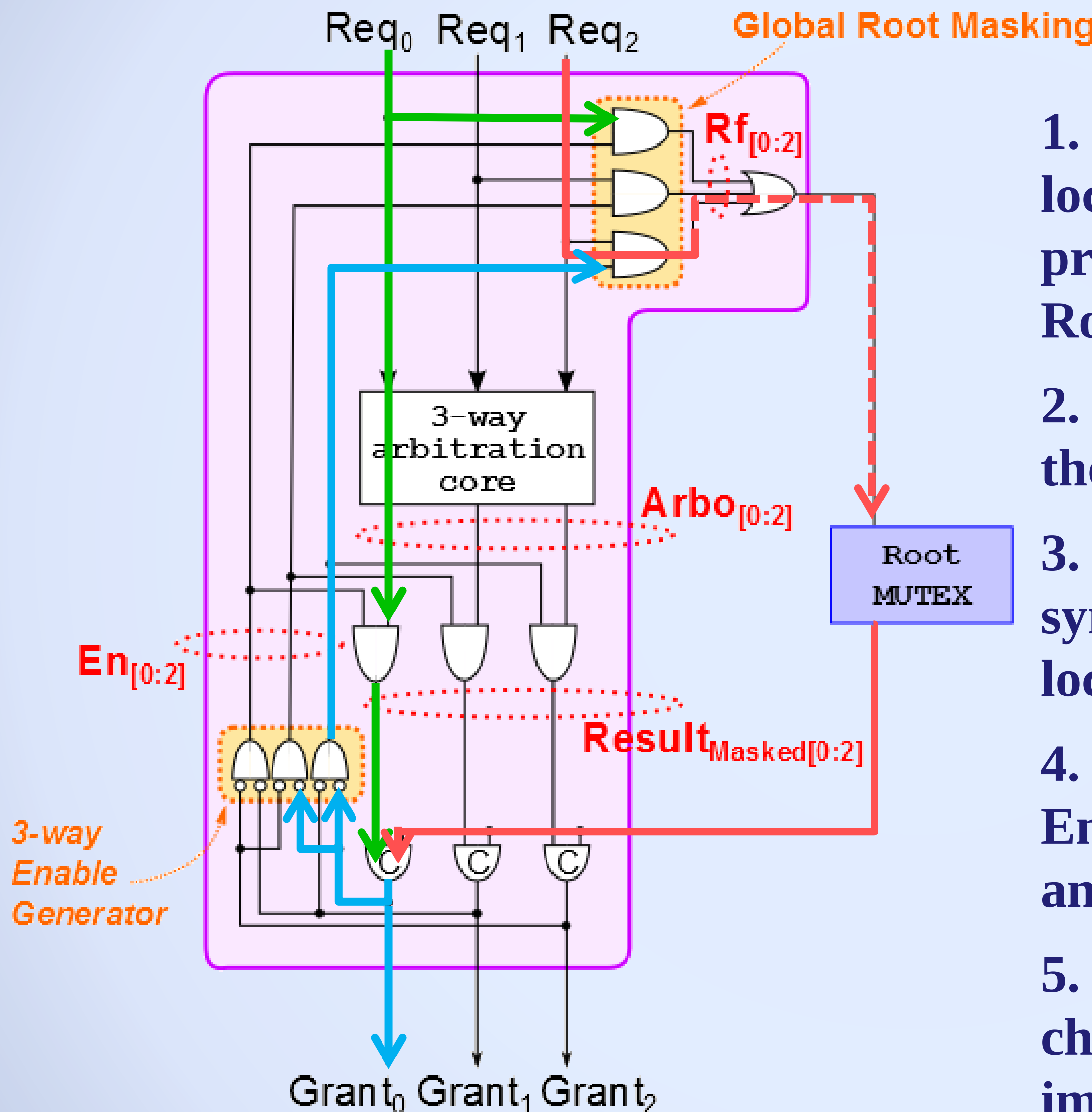
1. Req_0 comes, acquires the local mutex but gets stuck while propagating through the Global Root Masking
2. Req_2 comes and propagates to the root
3. The MullerC Element synchronizes the requests from the local and the root arbiter
4. Grant0 is asserted high, Enable generators for channel 1 and 2 are deasserted low

MAIN TIMING ASSUMPTION



1. **Req_0 comes, acquires the local mutex but gets stuck while propagating through the Global Root Masking**
2. **Req_2 comes and propagates to the root**
3. **The MullerC Element synchronizes the requests from the local and the root arbiter**
4. **Grant0 is asserted high, Enable generators for channel 1 and 2 are deasserted low**
5. **Masking is activated for channel 2 and the root is improperly released. (It can not be released until $Req_0\downarrow$)**

MAIN TIMING ASSUMPTION



$$\partial(AND_2 \uparrow) < \partial(6 - 7\ gates)$$

1. Req_0 comes, acquires the local mutex but gets stuck while propagating through the Global Root Masking
2. Req_2 comes and propagates to the root
3. The MullerC Element synchronizes the requests from the local and the root arbiter
4. Grant0 is asserted high, Enable generators for channel 1 and 2 are deasserted low
5. Masking is activated for channel 2 and the root is improperly released. (It can not be released until $Req_0\downarrow$)

EXPERIMENTAL RESULTS

We implemented post-layout models for seven different arbiter designs using a low-power standard-Vth 40nm technology library.

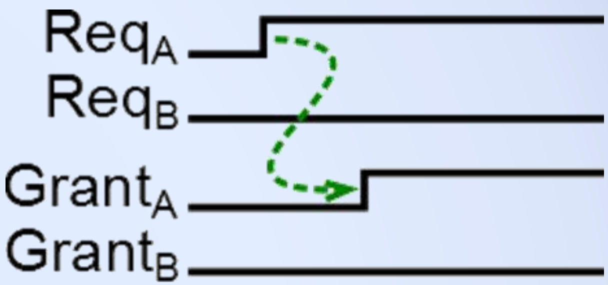
- **TREE ARBITERS: Baseline, Yakovlev ('94), Naqvi ('14) and proposed one, for dimensions from 3-way to 9-way**
- **RING ARBITER: Taubin ('07), for dimensions from 5-way to 9-way (3-way and 4-way are not feasible).**

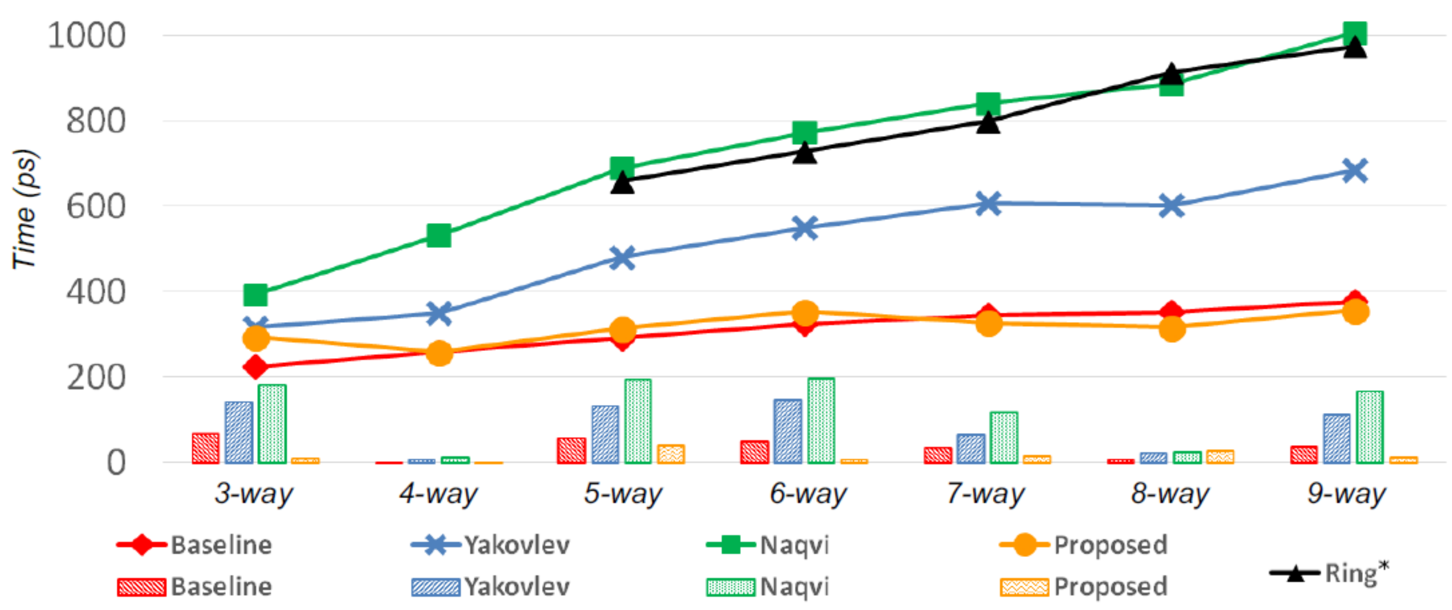
We evaluated several design metrics (performance, cost, robustness) including grant overlapping to investigate the robustness.

BASELINE: A. Ghiribaldi, D. Bertozzi and S.M. Nowick, "A transition-signaling bundled data NoC switch architecture for cost-efficient GALS multicore systems" ACM/IEEE DATE Conference, pp. 332-337, 2013.

EXPERIMENTAL RESULTS

Mean Latency and standard deviation experienced by all the design points under test in a non-competing scenario

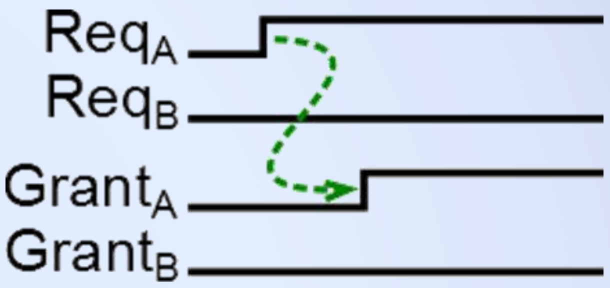




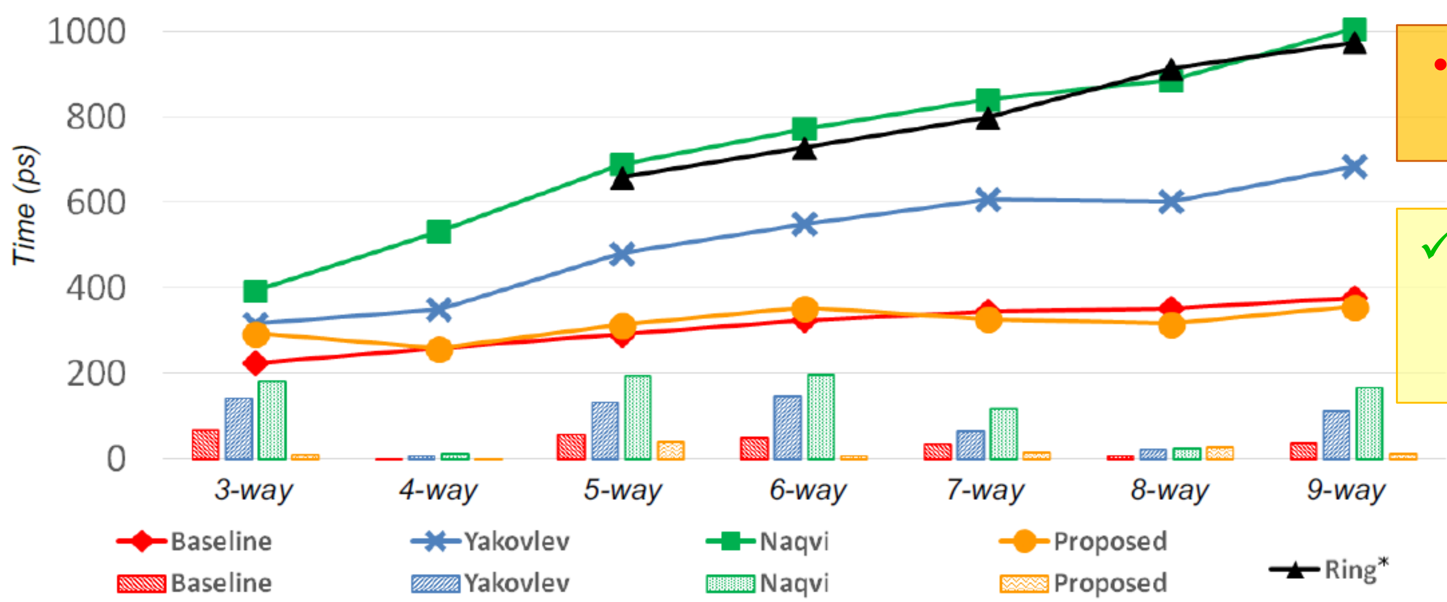
*Only average values for TokenRing were calculated under light traffic injection.

EXPERIMENTAL RESULTS

Mean Latency and standard deviation experienced by all the design points under test in a non-competing scenario



✓ **Proposed and baseline are the best overall solutions**



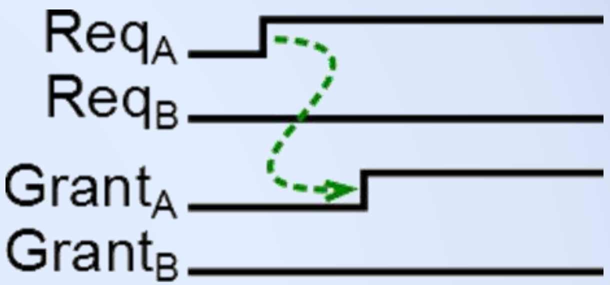
*Only average values for TokenRing were calculated under light traffic injection.

- **Other solutions scale linearly**

✓ **Nearly flat trend for baseline and proposed**

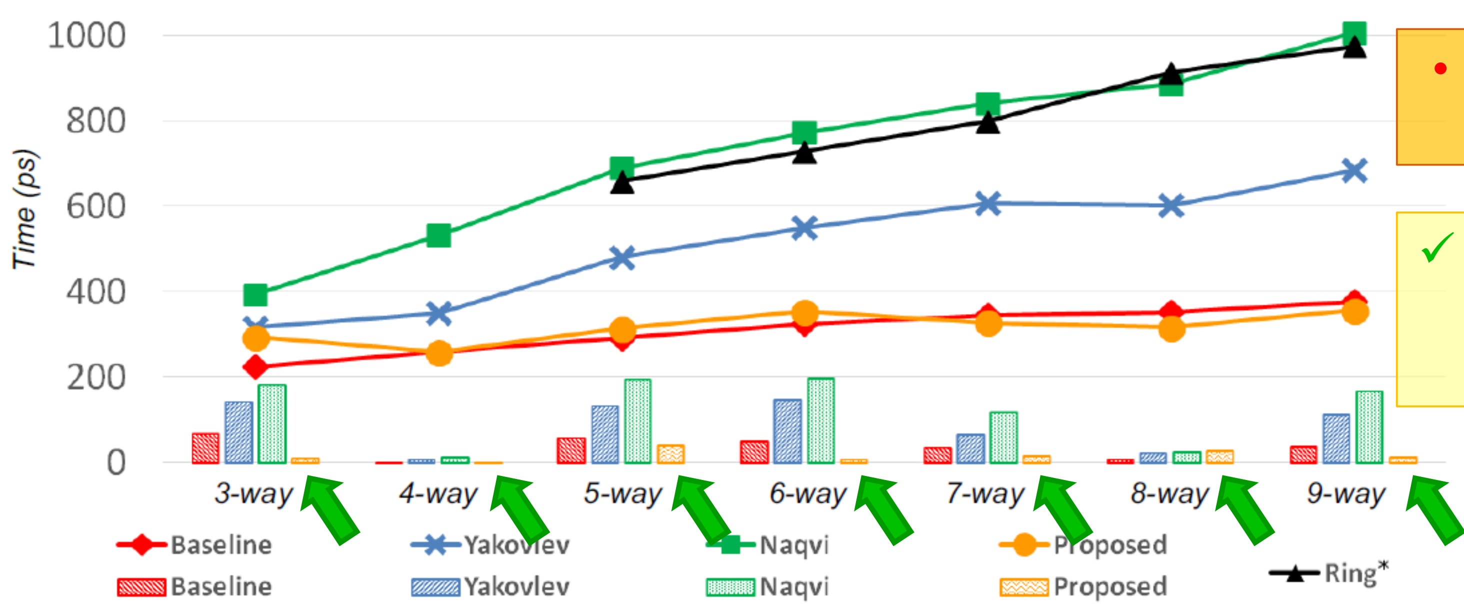
EXPERIMENTAL RESULTS

Mean Latency and standard deviation experienced by all the design points under test in a non-competing scenario



✓ **Proposed and baseline are the best overall solutions**

✓ **Proposed yields latency equalization across input requests for N**



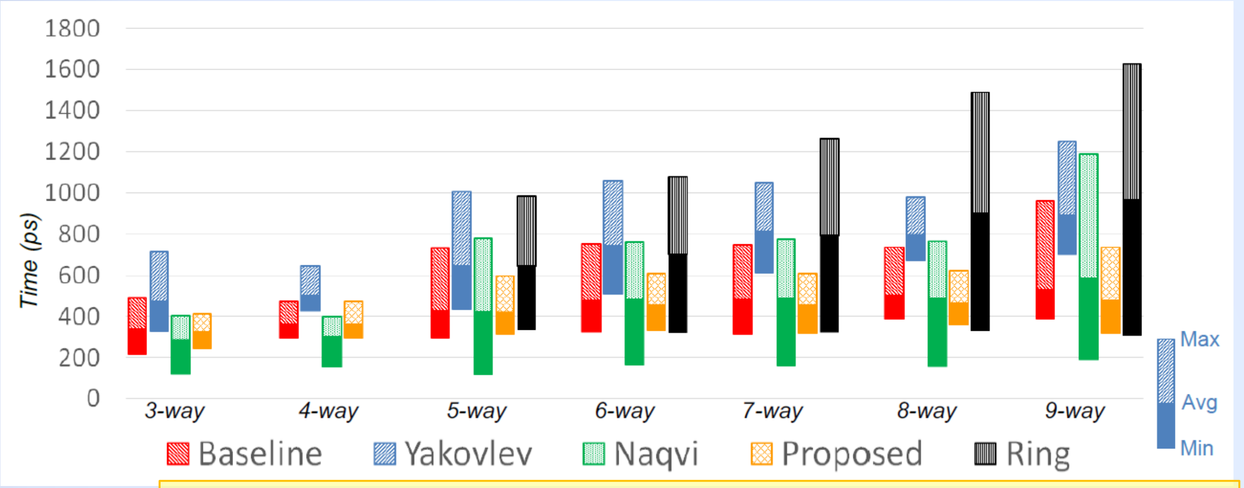
- **Other solutions scale linearly**

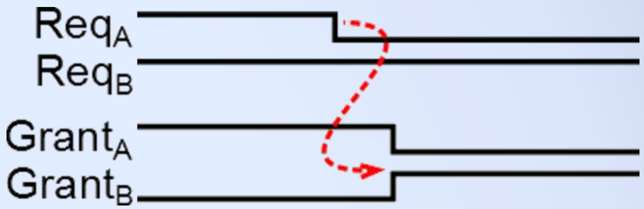
✓ **Nearly flat trend for baseline and proposed**

*Only average values for TokenRing were calculated under light traffic injection.

EXPERIMENTAL RESULTS

Multiple Channel Response Time between Req_n↓ and $Grant_m$↑ (n≠m)





✓ **Proposed, Baseline and Naqvi exhibit roughly similar mean performance**

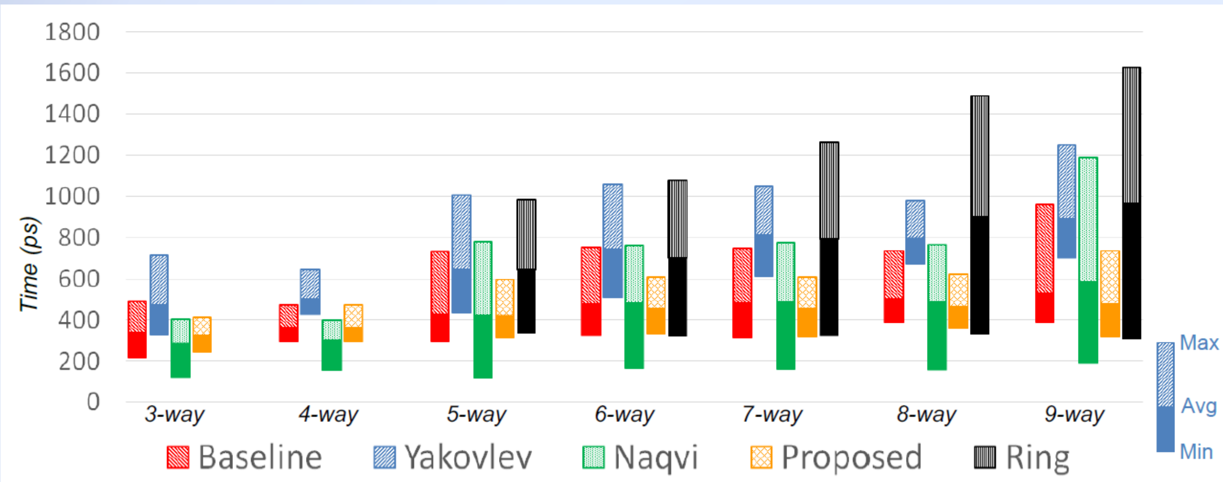
- **Naqvi , but also Baseline, exhibit larger variability as N increases**

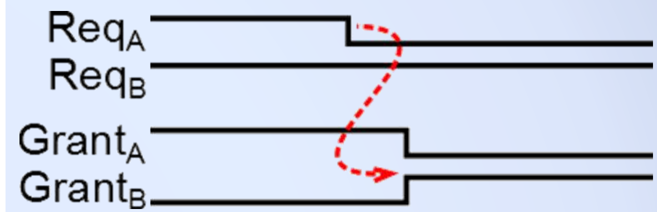
✓ **Proposed bounds the max. value quite effectively**

These results have been extracted using an *ActiveTime=400ps.* For long *ActiveTime* Naqvi becomes the best solution.

EXPERIMENTAL RESULTS

Multiple Channel Response Time between $Req_n\downarrow$ and $Grant_m\uparrow$ $(n \neq m)$





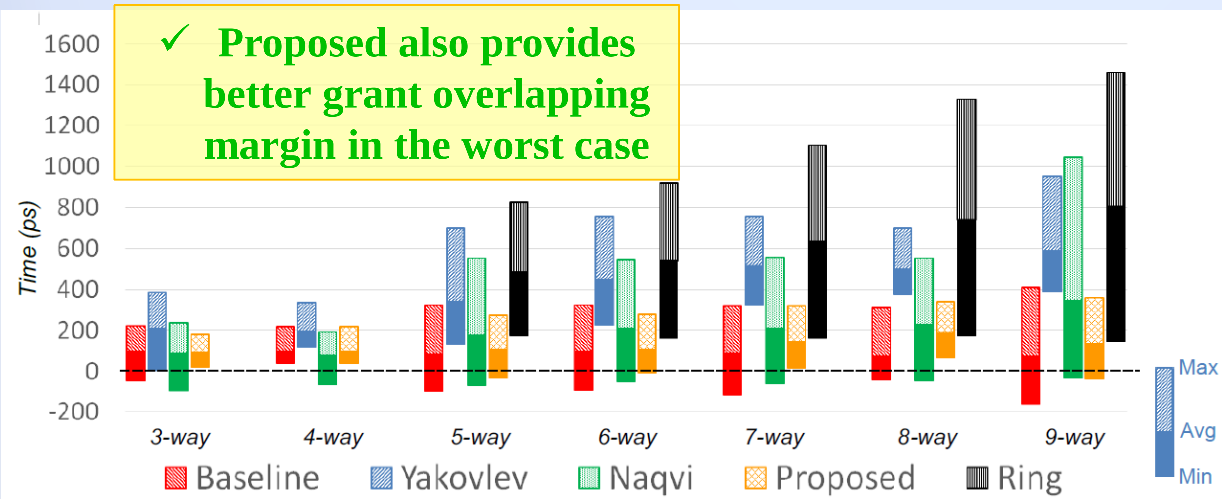
✓ Proposed, Baseline and Naqvi exhibit roughly similar mean performance

- Naqvi , but also Baseline, exhibit larger variability as N increases

✓ Proposed bounds the max. value quite effectively

Grant Overlapping Margin

✓ Proposed also provides better grant overlapping margin in the worst case

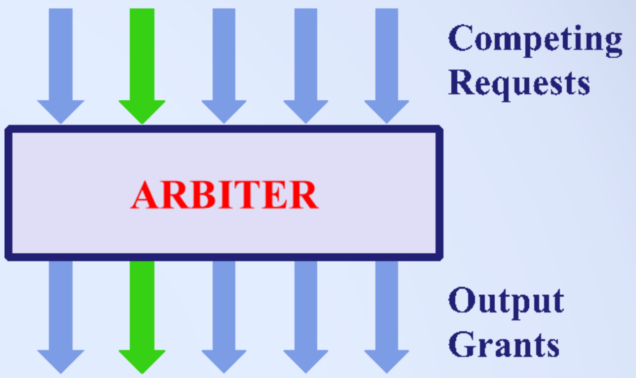


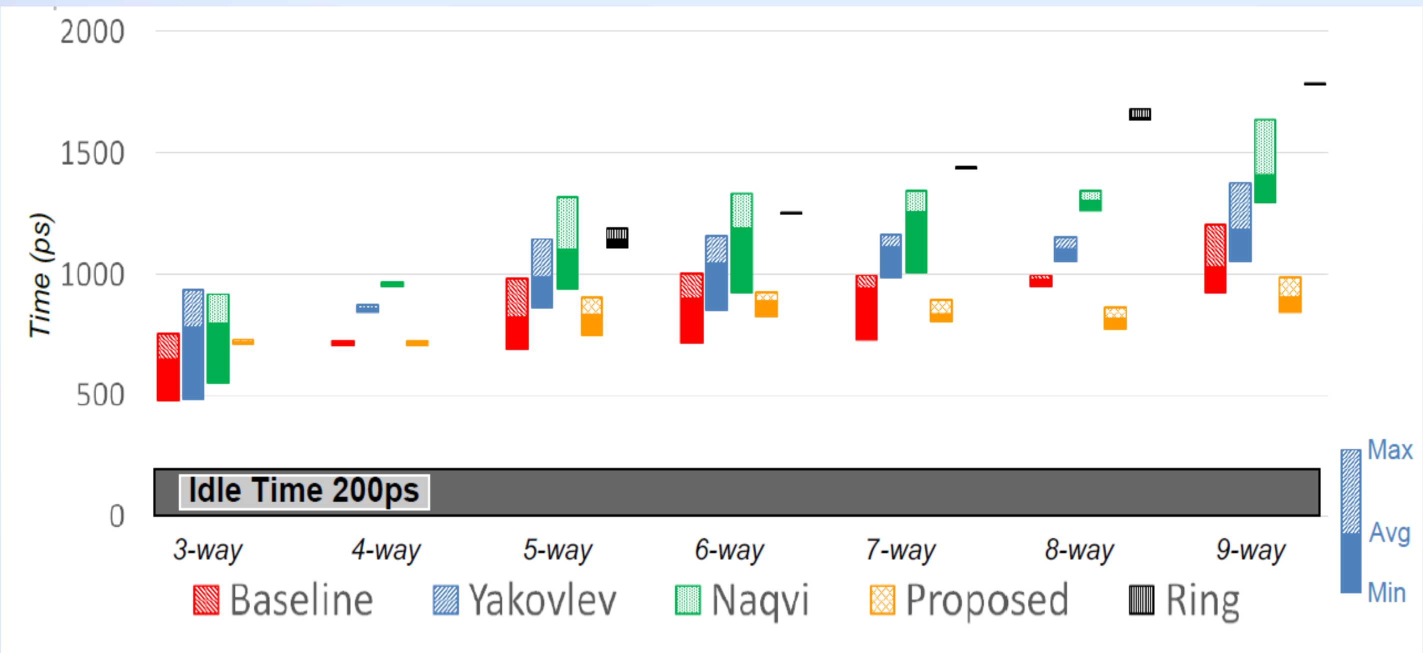
These results have been extracted using an *ActiveTime=400ps.*
For long *ActiveTime* Naqvi becomes the best solution.

EXPERIMENTAL RESULTS

Single Channel Response Time between Req_n↓ and $Grant_m$↑ (m=n) is an interesting metric to evaluate performance in case of bursty traffic from same input.

ActiveTime=400ps, IdleTime=200ps





- ✓ Proposed exhibits by far the best "worst-case" condition
- ✓ Proposed exhibits the best average performance overall
- ✓ Nearly flat trend for Proposed

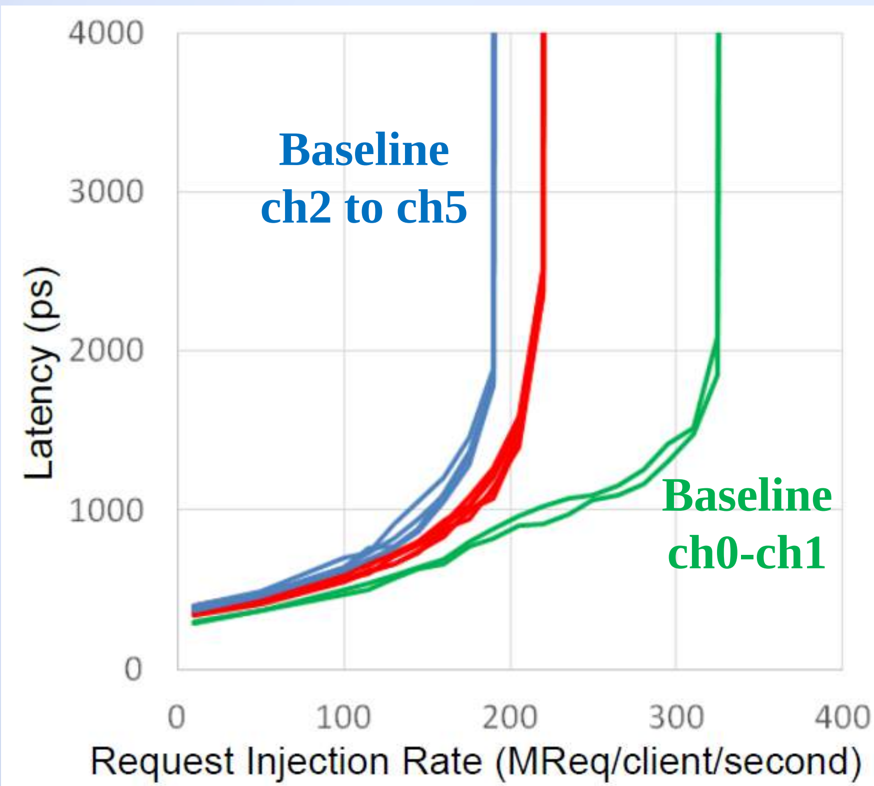
EXPERIMENTAL RESULTS

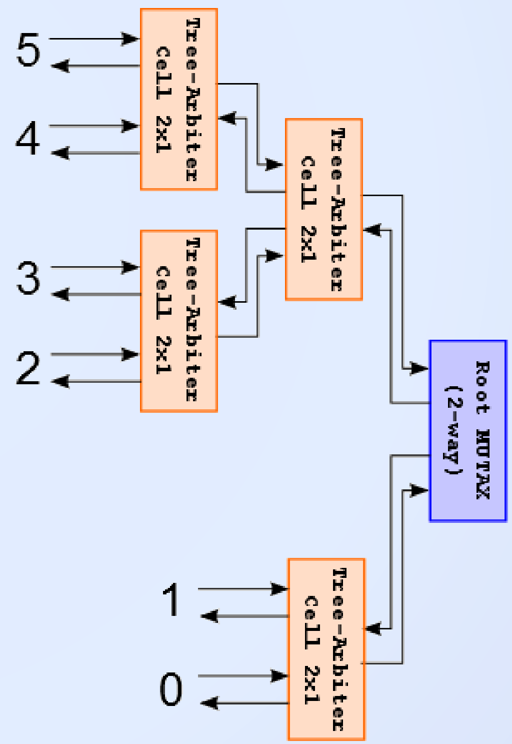
To evaluate the Impartiality of our proposed approach, we injected an uniform traffic of requests among all the clients, and we measured the acquisition time.

EXPERIMENTAL RESULTS

To evaluate the Impartiality of our proposed approach, we injected an uniform traffic of requests among all the clients, and we measured the acquisition time.

Proposed vs. Baseline (6-way)

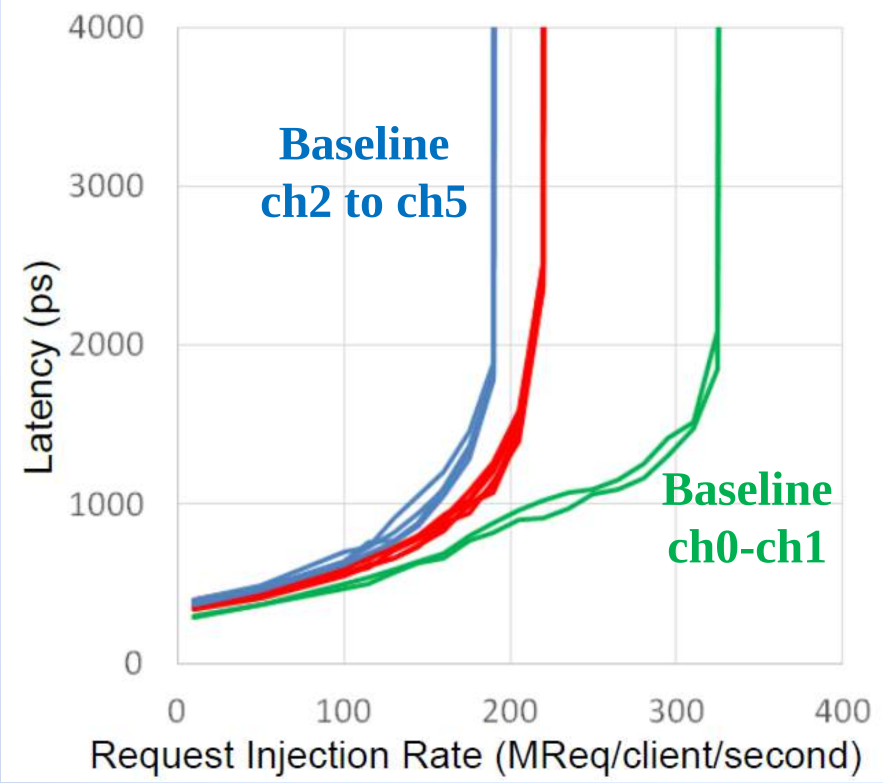




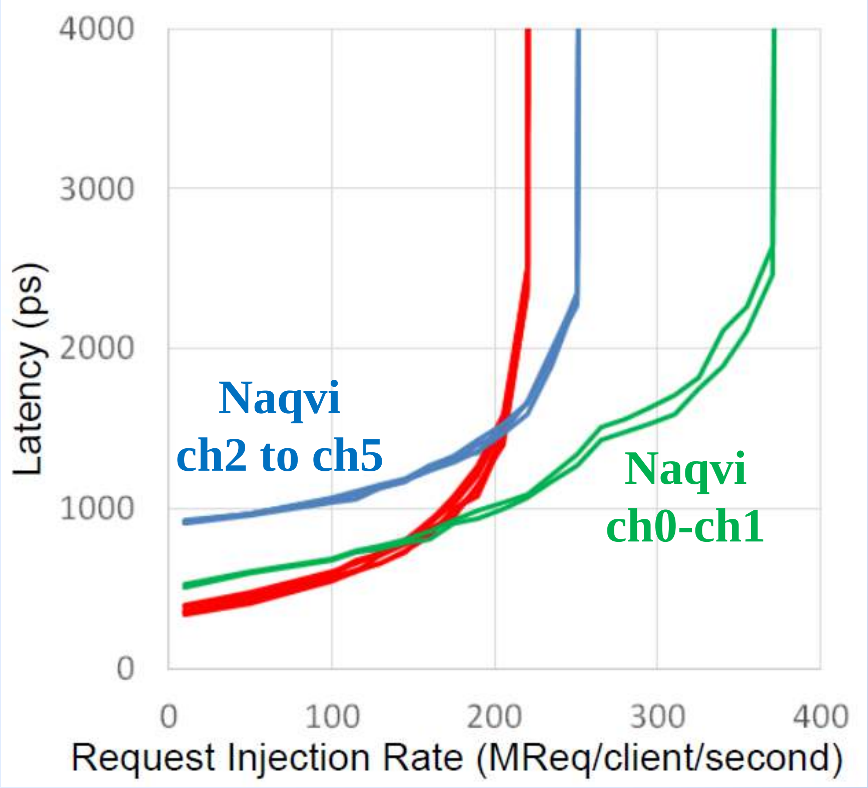
EXPERIMENTAL RESULTS

To evaluate the Impartiality of our proposed approach, we injected an uniform traffic of requests among all the clients, and we measured the acquisition time.

Proposed vs. Baseline (6-way)



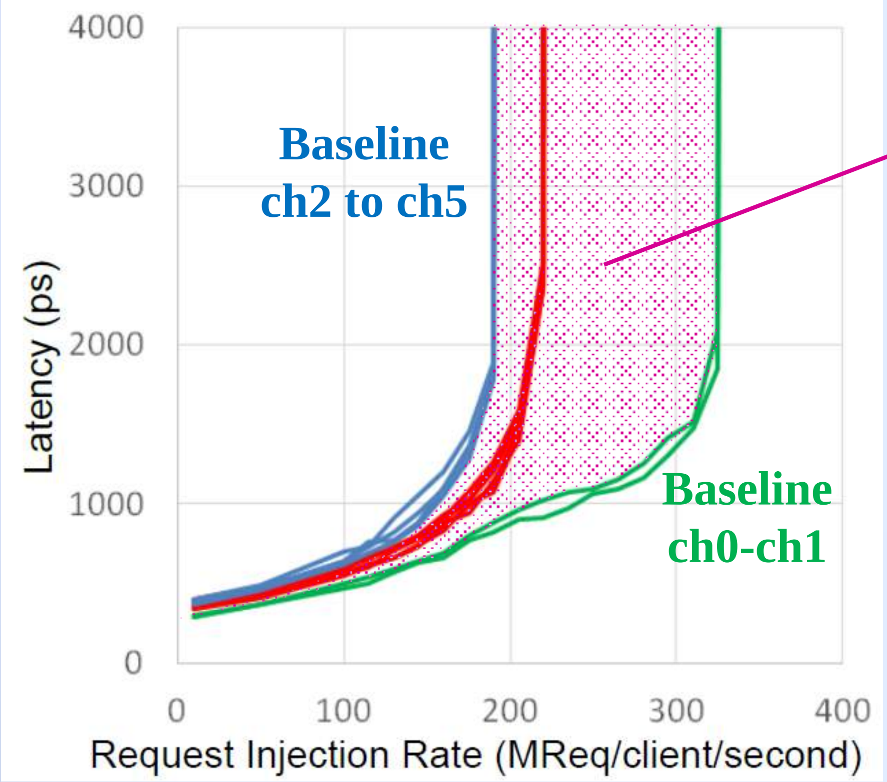
Proposed vs. Naqvi (6-way)



EXPERIMENTAL RESULTS

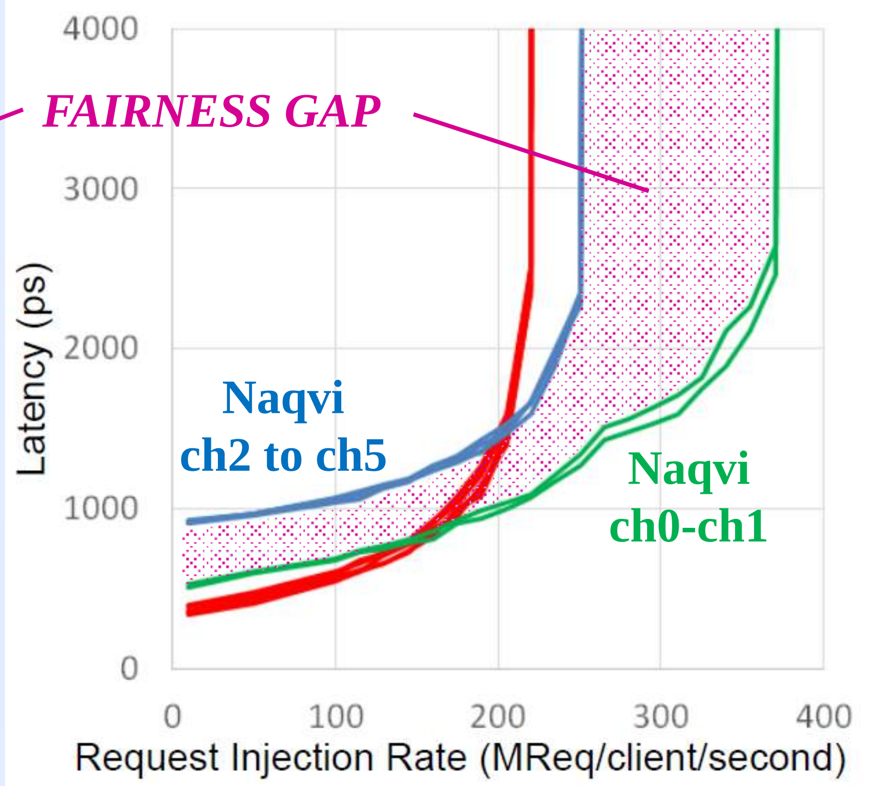
To evaluate the Impartiality of our proposed approach, we injected an uniform traffic of requests among all the clients, and we measured the acquisition time.

Proposed* vs. *Baseline (6-way)



Proposed* vs. *Naqvi (6-way)

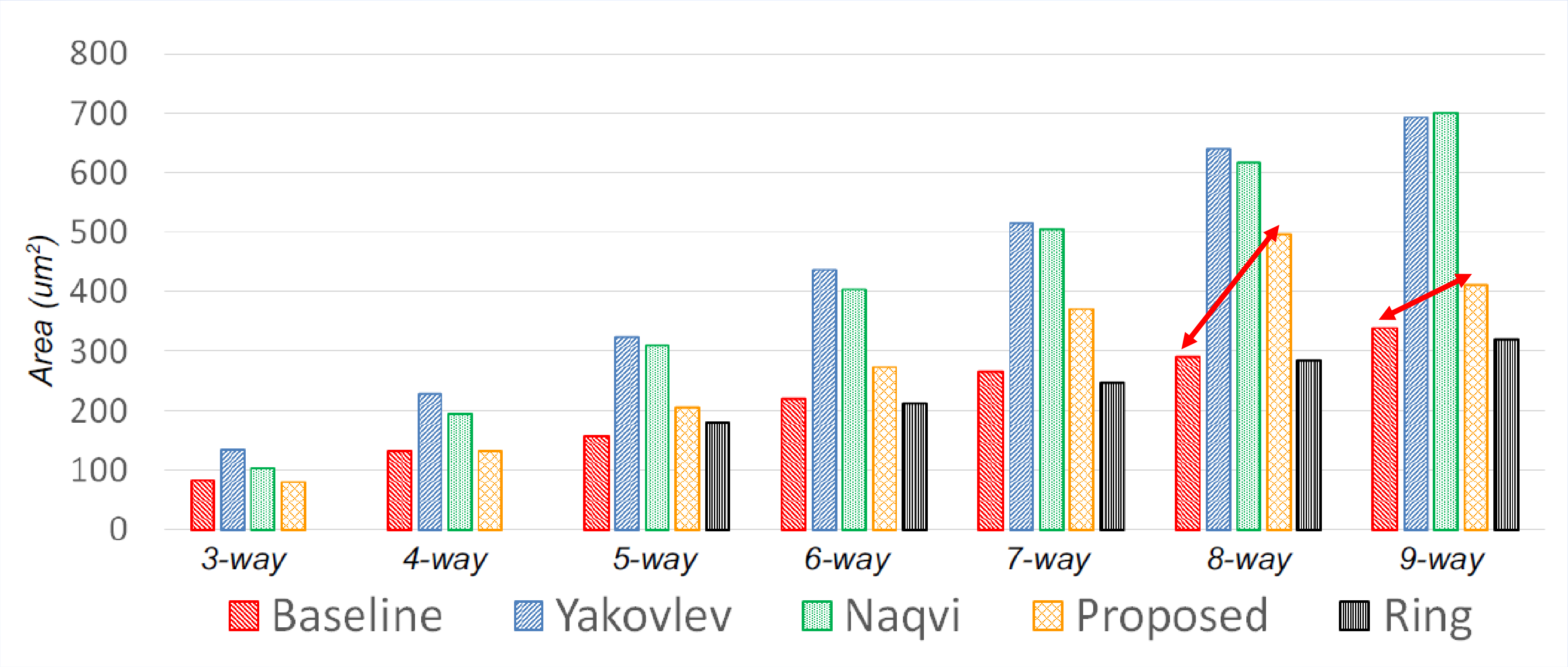
FAIRNESS GAP



For Naqvi and Baseline, only 2 of 6 clients have an optimal performance proposed exhibits equalized performance

EXPERIMENTAL RESULTS

Area Overhead



✓ **Baseline and Token ring are the simplest solutions (roughly 20-30% less than Proposed in the worst case)**

Proposed has a discontinuity (i.e., improved area efficiency) between 8 and 9-ways due to the use of 3-way roots and TACs.

With respect to Baseline, Proposed **trades area for latency and throughput equalization/scaling, and better GO margin**

CONCLUSIONS

- **Rebalancing of timing paths in asynchronous arbiters has never been addressed by previous work, despite the aggressive use of parallel protocols**

- **Effective solutions have been devised for fixed-size arbiters, while the design of scalable N-way arbiters is lagging far behind**

✓ **This work proposed a *novel rebalanced tree structure* which**
- ***materializes performance equalization across input requests***
- ***achieves the best performance scalability trends***

while yielding unprecedented multi-objective balance of cost functions with respect to existing arbiters

✓ **Robustness is part of the balance, by minimizing grant overlapping**
- ***this is a consequence of the performance equalization that has been achieved within the novel building blocks we delivered (e.g., 3x1 and 4x1 TACs).***

Our novel hierarchical recursive architecture is a promising solution to implement a scalable high-radix arbiter

Thank You!

Questions

Gabriele Miorandi (gabriele.miorandi@unife.it)