Monitoring and Controlling Interconnect Contention in Critical Real-Time Systems

PhD Candidate: Jordi Cardona[1,2]

Thesis Advisors: Carles Hernández[2,3], Enrico Mezzetti[2], Jaume Abella[2] and Francisco J. Cazorla[2]

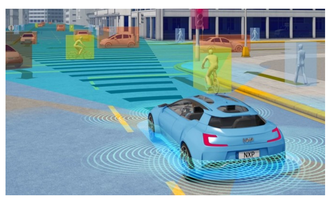
[1]Universitat Politècnica de Catalunya (UPC) Barcelona, Spain

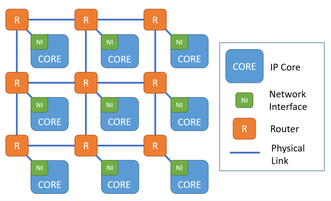
[2]Barcelona Supercomputing Center (BSC) Barcelona, Spain

[3]Universitat Politècnica de València València, Spain

Motivation

- **Critical Real-Time Systems (CRTS) industry is shifting towards complex hardware** (e.g. multi/many-cores) fueled by the increasing computing power required by the new high-performance software applications.
- Complex high-performance hardware complicates the **Validation and Verification (V&V)** required in CRTS systems:
  - **Timing Correctness** (Worst-Case Execution Time (WCET) estimation).
  - Functional Correctness
- **Interconnects and specially Networks-on-chip (NoCs) are one of the main shared resources in which contention arises.** The reduction and accounting of NoCs contention keeps currently being challenging due to their distributed nature (focus wormhole NoCs (wNoCs)).

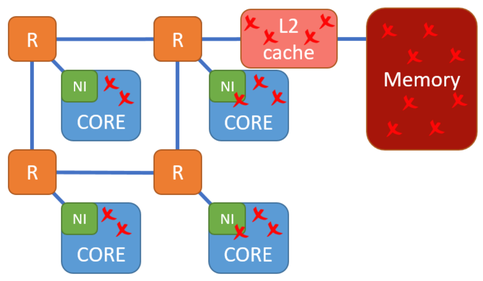


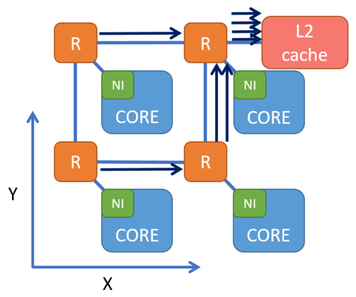


Background

Hardware & Software solutions:

- Small CRTS market
- We focus on using **Commercial Off The Shelf (COTS) hardware**





Reducing WCET estimates in NoCs:

- Several proposals show how to **compute WCET estimates under some restrictions** in the NoC (e.g. deterministic routing, bandwidth allocation, …).
- **WCET in wNoC** systems is computed accounting for the Worst-case Contention Delay (WCD) a packet can suffer from the other packets in the NoC (**very pessimistic**).
- **wNoCs offer several software-controllable parameters** that allow to reduce WCD of packets (e.g. mapping, routing and bandwidth allocation to routers) that are very interrelated. Many solutions **optimize different NoC parameters separately** or combining more than one parameter.

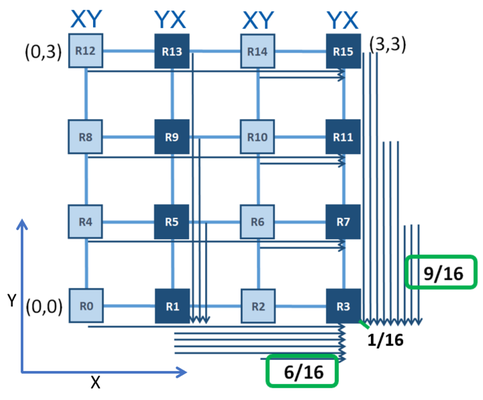
Monitor and control contention in Interconnects:

- **WCET estimations are normally enforced and monitored** during applications execution limiting the amount of contention a task can cause to the others.
- Hardware and software solutions have been proposed.
- Determining the sources of contention in distributed NoCs (e.g. 2DMesh) is even more **challenging** than in centralized interconnects (e.g. BUS).

Contributions

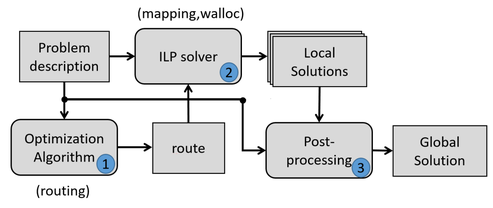
EOmesh: Combined Flow Balancing for Reduced WCET estimates[2]:

- WCD reduction by optimizing routing + bandwidth distribution together.
- Distributed meshes (e.g. 2DMeshes)
- Weighted meshes[1] have limitations
- Theoretic bandwidth distribution is not really fulfilled (imbalance).
- XY + YX deterministic routing solution
- WCD and WCET reductions in the range of 5% -28% at low hardware cost.



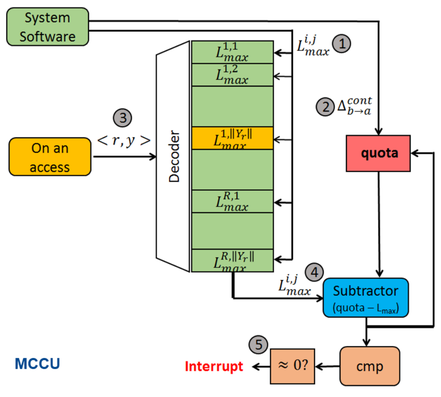
NoCo: ILP-based Worst-Case Contention Estimation for Mesh[3]:

- WCD and WCET reduction by optimizing mapping, routing and bandwidth allocation all together.
- Using a stochastic-ILP model we provide the best wNoC setup that minimizes WCET of tasks
- Analyze the WCET reduction when optimizing combinations of parameters.
- WCET is reduced 40% - 50% w.r.t XY-RR(3x3)



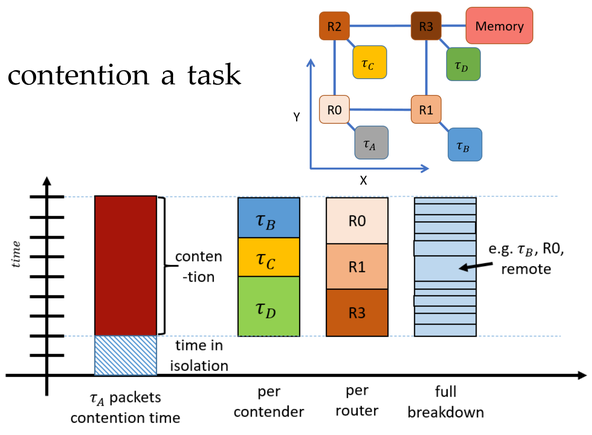
Maximum Contention Control Unit (MCCU)[4]:

- Bounds contention in multiple shared resources of the system (e.g. BUS, L2)
- Little intrusive hardware/software approach that builds on QME approaches.
- Monitors contention using the Processor Monitoring Counters (PMCs).
- Provides fine grain contention control and performance guarantees per task.
- Reduces quota interrupts.



Classification Contention Metric:

- Metric that accounts and classifies the contention a task suffers from their co-runner tasks.
- Distributed wNoCs (e.g. 2DMeshes)
- Uses NoC tracing tools reading the routers' traversing packet counters.
- Provides per task multi-break down contention information.
- Useful for time violations detection and correction.



Evaluation

Hardware simulators:

- **SoCLib-based** performance simulator with NGMP multicore architecture.
- **gNoCsim** simulator standalone and integrated in SoCLib.

NGMP information:

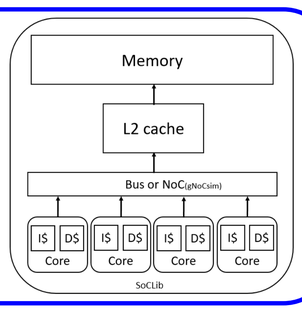
- LEON4 cores (in-order)
- L1: 4-way 16KB write-through
- BUS: on-chip round-robin policy
- L2: 4-way 256KB shared write-back

NoCs information:

- Ideal and ARM CoreLink CMN600 based NoCs.
- 2x2—7x7 2DMesh wNoCs size
- Pipelined routers
- Multiple buffer sizes

Benchmarks:

- EEMBC automotive
- MediaBench suit
- Synthetic traffic and self-created resource stressing kernels



Conclusions & Future Work

In this Thesis we have seen a wide range of the CRTS challenges, and we have proposed some solutions based in monitoring and controlling contention in multi and many-core wNoC systems. We are still working on the part of measuring/classifying the contention in NoCs as it is specially challenging when working with a very reduced amount of information to trace. As a future work, we also expect to have the opportunity to implement our solutions in real hardware in the context of the European Processor Initiative (EPI) project in the automotive part.

Acknowledgments

This work has been partially supported by the European Research Council (ERC), the European Union's Horizon 2020 research and innovation program (grant agreement No. 772773 and No. 826647(EPI) respectively) and the HiPEAC Network of Excellence. MINECO partially supported Jaume Abella under Ramon y Cajal postdoctoral fellowship (RYC-2013-14717).

References

[1] M. Panic et al. "Improving performance guarantees in wormhole mesh NoC designs". In 2016 Design, Automation Test in Europe Conference Exhibition (DATE), pages 1485–1488, March 2016.

[2] J. Cardona et al. "EOmesh: Combined flow balancing and deterministic routing for reduced wcet estimates in embedded real-time systems". IEEE TCAD, 2018.

[3] J. Cardona et al. "NoCo: ILP-based worst-case contention estimation for mesh real-time manycores". In 2018 IEEE RTSS.

[4] J. Cardona et al. "Maximum-contention control unit (MCCU): Resource access count and contention time enforcement." In 2019 DATE.

[5] SoClib. 2003-2012. http://www.soclib.fr/trac/dev

[6] NaNoC Project, https://sites.google.com/site/nanocproject/