From multi to many-core: network on chip challenges

On-board software and multicores : opportunities and issues
CCT SIL-IRE, Mardi 16 Juin 2015

Marc Boyer

ONERA – The French Aerospace Lab

June 16, 2015

ONERA
THE FRENCH AEROSPACE LAB

retour sur innovation

Disclaimer

- exact information are not always public
- the meaning of words sometimes (...) differs
- personal experience on limited set of chips

A good overview: "ISSCC TRENDS" `http://isscc.org/trends`

This and other related topics have been discussed at length at ISSCC 2015, the foremost global forum for new developments in the integrated-circuit industry.
Next session: Jan. 31 – Feb. 4 2016, San Francisco, CA
`http://isscc.org`

ONERA experience I

ONERA has been involved in several many-core projects

- MARC (2009-2012): Intel Many-Core Application Research Community academic program
- MACSIMA (2013-2017): ONERA internal research project
  - topic: radar, planification and control-command application on the same many-core
- DREAMS (2013-2017): FP7 European project
  - topic: mixed-criticality on multi-core
  - partners: Univ. Siegen, Ikerlan s. coop, Thales, ONERA, RealTime-At-Work, TTTech, Technische Univ. Kaiserslautern, Fortiss, Sintef, Alstom, STMicroelectronics, Virtual Open Systems, Technological Educational Institute of Crete, Polytechnic Univ. of Valencia, Fentiss, TUV
- CAPACITÉS (2014-2018): LEOC (Logiciel Embarqué et Objets Connectés) french-founded project,

ONERA experience II

- topic: multicore for critical embedded systems
- partners: Airbus, Airbus Helicopter, ARMINES, Dassault Aviation, Inria, IRISA, IRIT, IS2T, Kalray, MBDA, ONERA, OpenWide, Probayes, Real-Time At Work, Safran, Supersonic Imagine, TIMA, Verimag

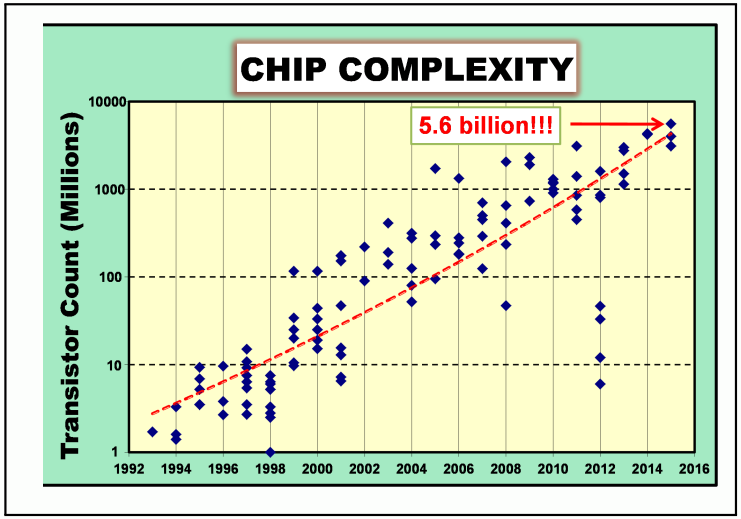
15 publications (2011-2015):
[16, 19, 23, 17, 20, 15, 21, 3, 1, 8, 6, 7, 11, 13, 14]

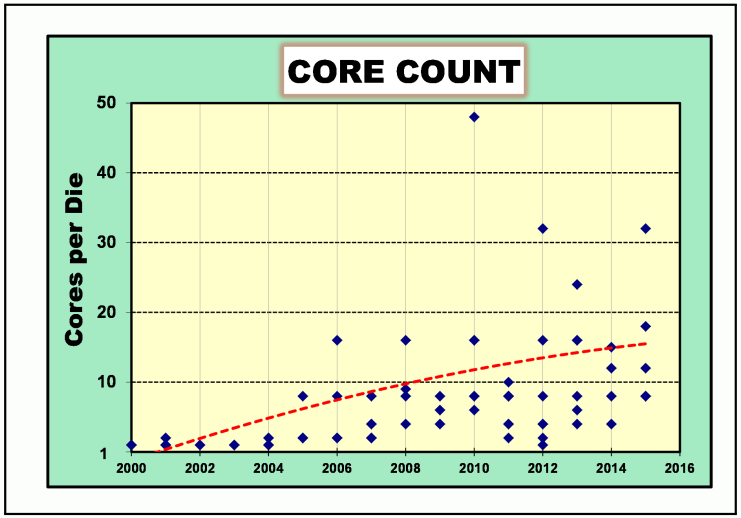
Outline

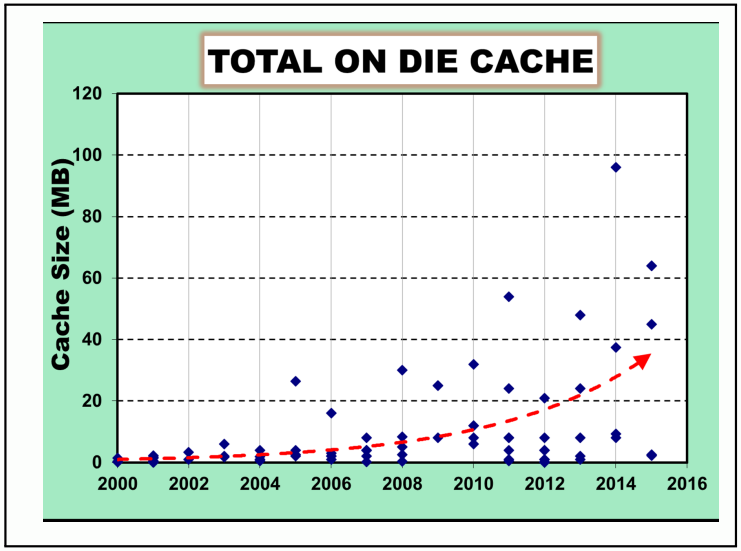
Outline

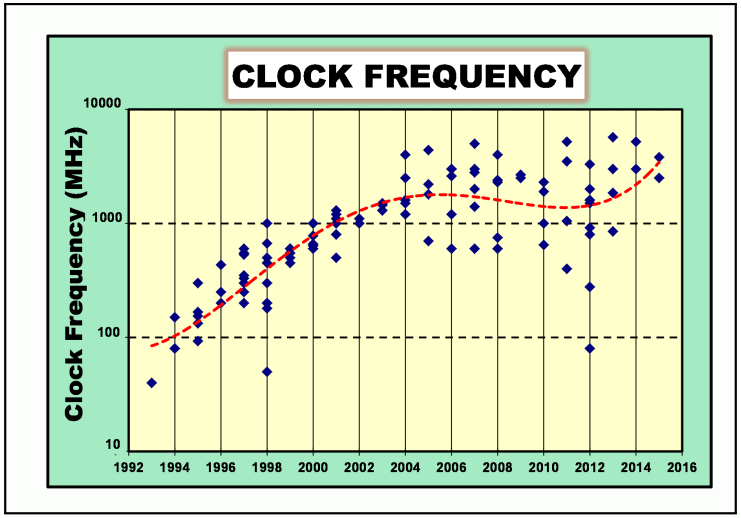
Outline

Some trends [25]









Cache memory

More cores, and more cache

- cache consumes few energy
- cache is efficient

But...

Cache memory

More cores, and more cache

- cache consumes few energy
- cache is efficient

But...

- how to ensure cache coherency with 32 cores ?

Cache memory

More cores, and more cache

- cache consumes few energy
- cache is efficient

But...

- how to ensure cache coherency with 32 cores ?
- why ?

Cache memory

More cores, and more cache

- cache consumes few energy
- cache is efficient

But...

- how to ensure cache coherency with 32 cores ?
- why ?
- local cache or local memory ?

Cache memory

More cores, and more cache

- cache consumes few energy
- cache is efficient

But...

- how to ensure cache coherency with 32 cores ?
- why ?
- local cache or local memory ?
- implicit or explicit communications ?
  - message passing vs shared memory

Cache memory

More cores, and more cache

- cache consumes few energy
- cache is efficient

But...

- how to ensure cache coherency with 32 cores ?
- why ?
- local cache or local memory ?
- implicit or explicit communications ?
  - message passing vs shared memory
- an old/new programming way

Outline

Outline

Network on chip

- how to connect chip elements ?
- NoC for SoC vs NoC for multicore
  - homogeneous vs heterogeneous system
  - access to main memory and IO
- different approaches depending on manufacturer
- less information than on cores

From bus to NoC

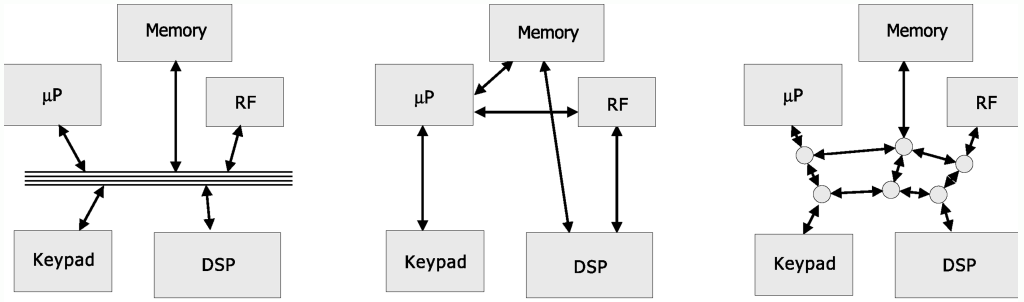


Figure : From bus to NoC [2]

- Bus: shared resource
- Pt-to-pt: does not scale
- NoC:
  - set of shared resources
  - allow parallel communications

A common vocabulary

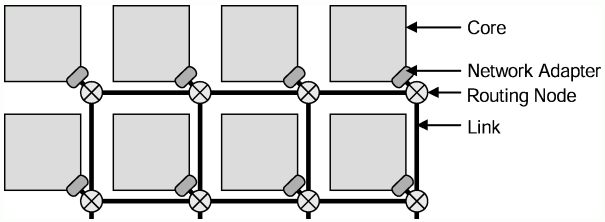


Figure : Architecture elements[2]

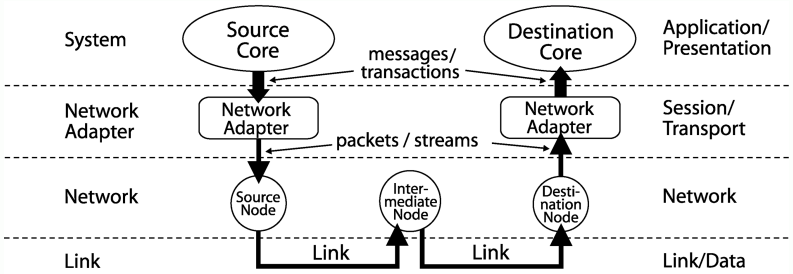
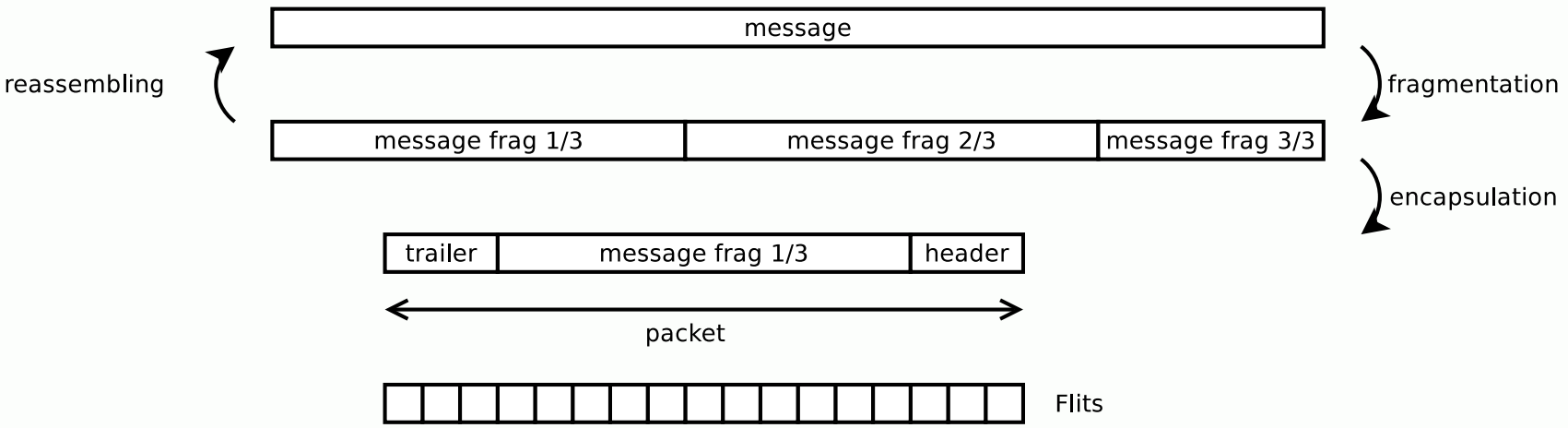


Figure : Network layers [2]

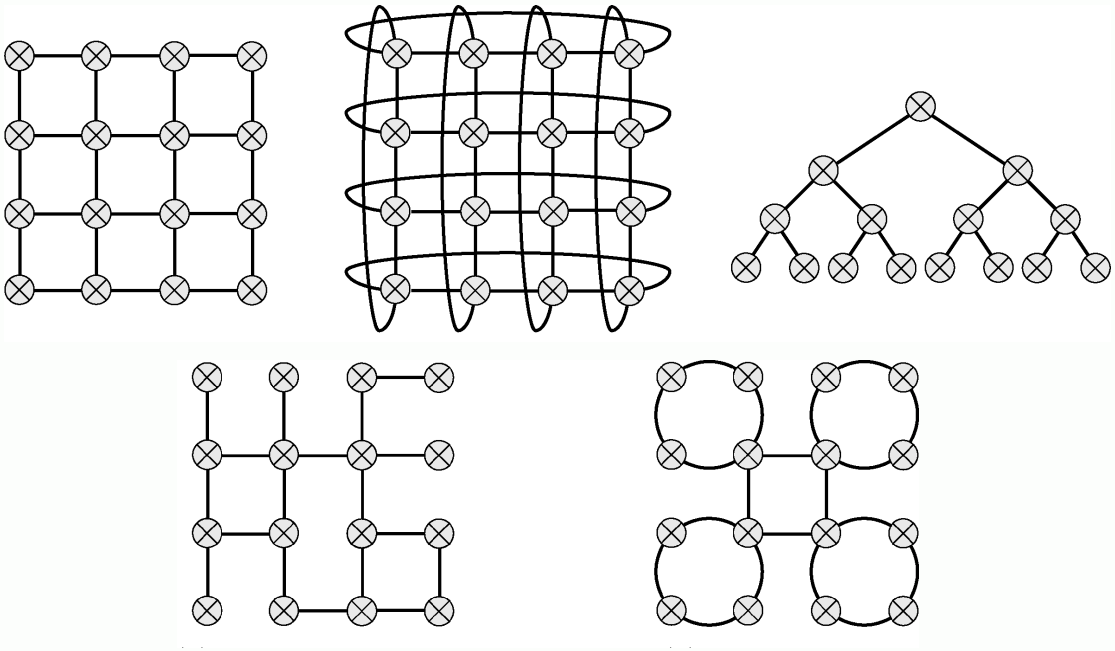
- Core/tile: could be also IO/RAM
  - write/read messages
- Network adapter
  - fragment/reassemble messages into packets
  - send/receive packets
  - flow control
- Routing node: commutation element
  - send/receive *flits* ($\approx$ 64bits)
  - also flow control

Data vocabulary



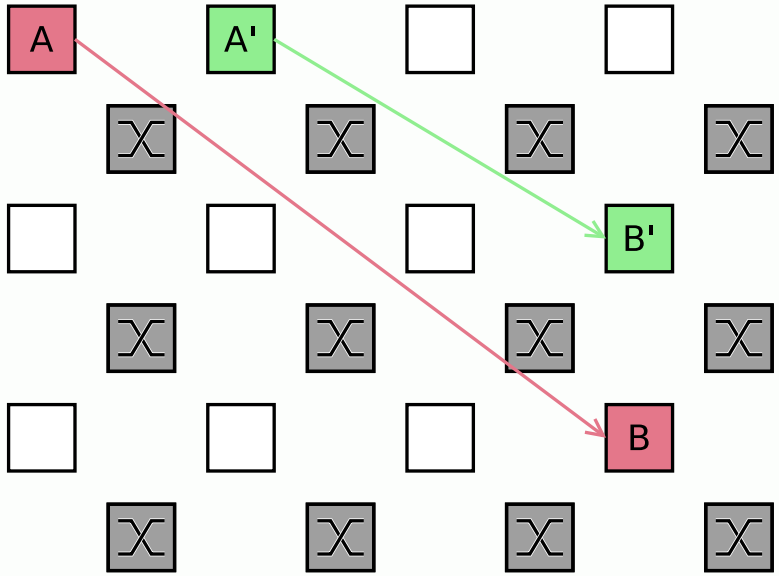
- Core/tile: could be also IO/RAM
  - write/read messages
- Network adapter
  - fragment/reassemble messages into packets
  - send/receive packets
  - flow control
- Routing node: commutation element
  - send/receive *flits* ($\approx$ 64bits)
  - also flow control

Topologies



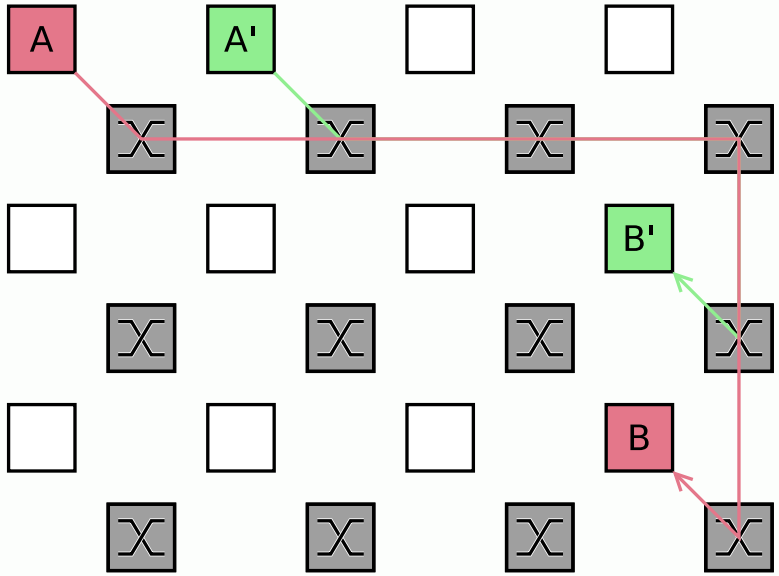
Outline

Routing



Routing:

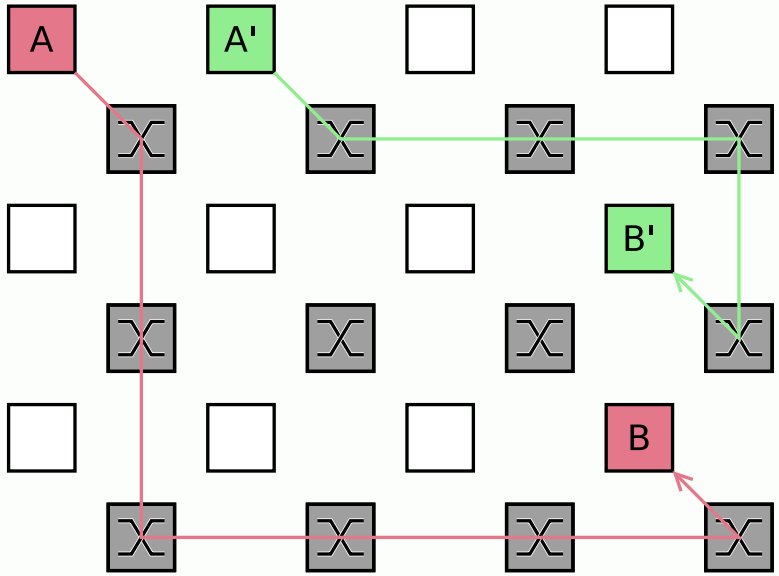
Routing



Routing:

- XY: follows the row first, then moves along the column
  Note: reverse communication uses another path

Routing



Routing:

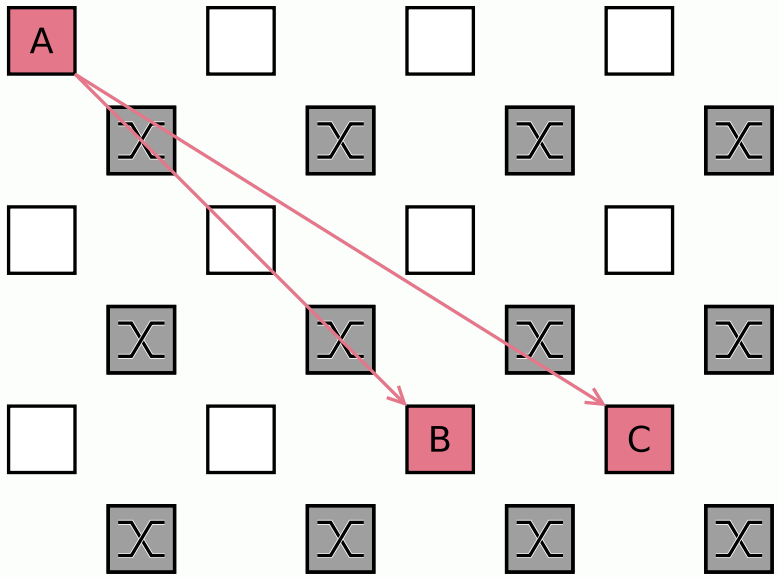
- XY: follows the row first, then moves along the column
  Note: reverse communication uses another path
- Source routing: source set the path in the header

Routing

Routing:

- XY: follows the row first, then moves along the column
  Note: reverse communication uses another path
- Source routing: source set the path in the header
- Adaptative:
  - route computed "on the fly"
  - minimize link/router load
  - research only ?

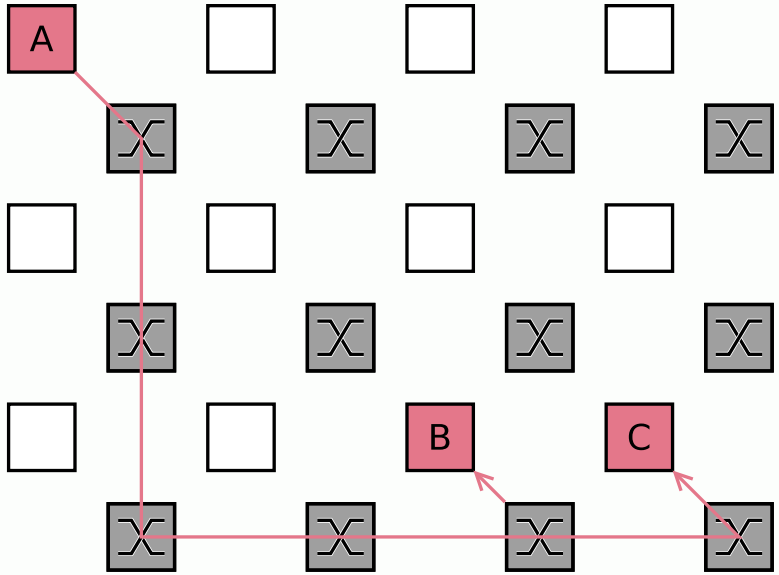
Multicast



Multicast: sending the same data to several cores

- multicast NoC: data send once, path sharing
- non multicast NoC: data sent several times, path competition

Multicast

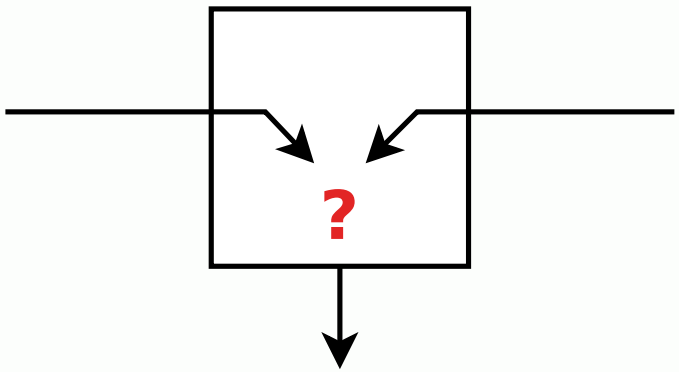


Multicast: sending the same data to several cores

- multicast NoC: data send once, path sharing
- non multicast NoC: data sent several times, path competition

Outline

Router: managing contention



They always are contentions

- arbitration is needed
  - $\Rightarrow$ link scheduling policy
  - $\Rightarrow$ storage is needed
    - $\Rightarrow$ memory allocation policy
      such memory is expensive

$\Rightarrow$ large set of solutions

Forwarding

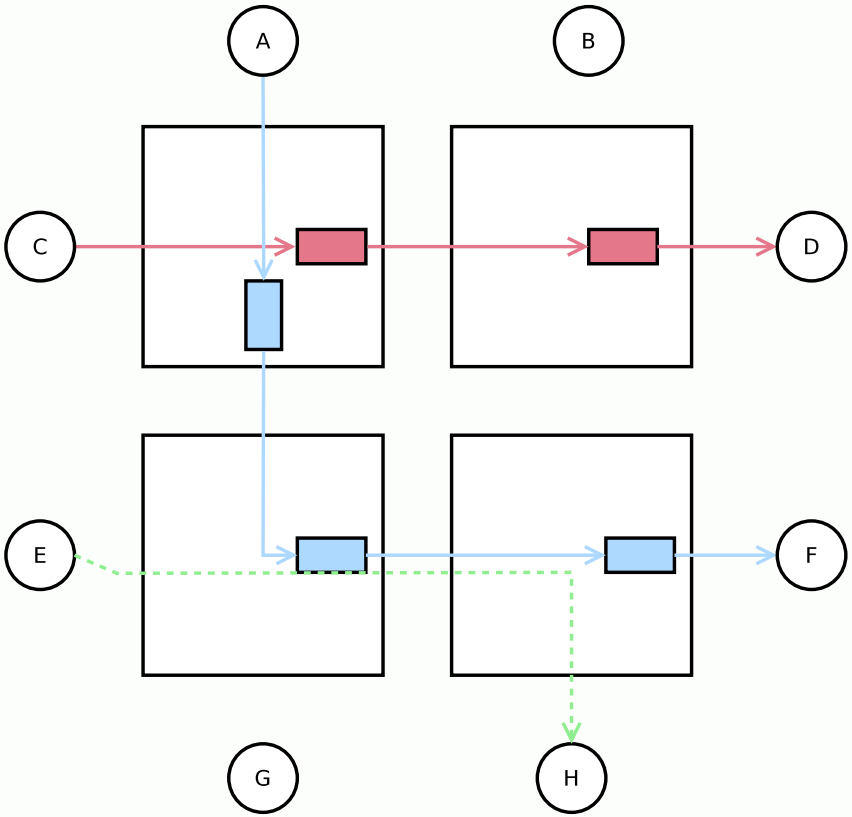
Forwarding: how to transmit data?

- store & forward
- wormhole
- virtual circuit
- virtual cut-through

Store and forward

- common policy in network
- send and store full packets in routers
- require buffer size many times larger than packet size

Wormhole forwarding

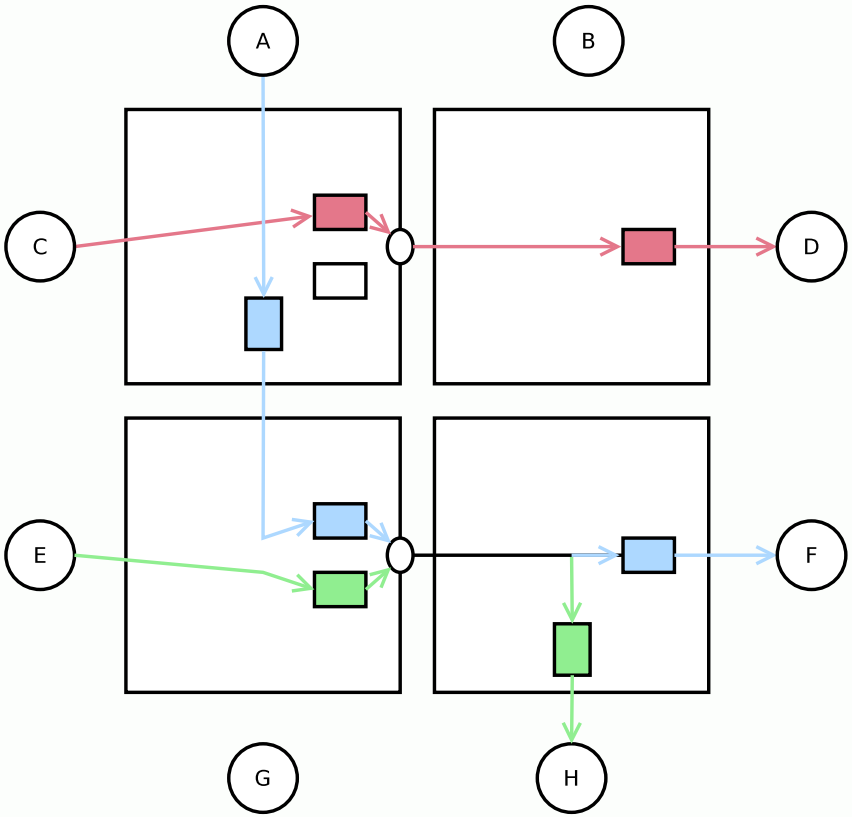


- like Spacewire
- forward flits even while receiving
- send flits up to blocking

$\Rightarrow$ link flow control

- allow large messages / packets
- implies blocking, and even dead-locks

Virtual circuit enhancement



- reduces blocking
- require more logic
  - flit tagging
  - VC id allocation
  - memory sharing
  - arbitration policy
- allows per VC QoS

Virtual cut-through forwarding

- looks like wormhole restriction

Virtual cut-through forwarding

- looks like wormhole restriction
- or store&forward enhancement

Virtual cut-through forwarding

- looks like wormhole restriction
- or store&forward enhancement
- send packet only if enough space in next router

Virtual cut-through forwarding

- looks like wormhole restriction
- or store&forward enhancement
- send packet only if enough space in next router

$\Rightarrow$ require storage of full packet

Forwarding: feedback

Forwarding	Per node cost		
	Latency	Memory	
store & forward	*packet*	*packet*	Common in networks
wormhole	*header*	*header*	Blocking
virtual cut-through	*header*	*packet*	A trade-off (?)

- wormhole has good mean performances

... but can lead to dead-locks

- virtual circuit has less blocking

... but require more memory and logic

- store& forward / virtual cut-through are well known

... but require more memory or small messages

Outline

The time-triggered approach

Build a global TDMA schedule

+ avoid any contention
+ small network delays
- require periodic tasks/communications
- does it scale ?

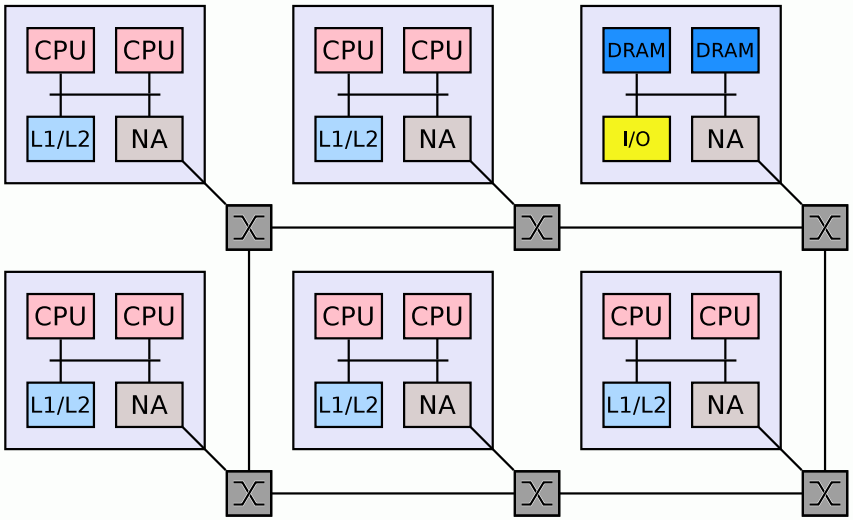
Connection oriented

Avoid contention in network, by establishing core-to-core connection, and resource reservation.

- increases latency (connection establishment)
- increases logic
- research only ?

Outline

Tile-based solutions



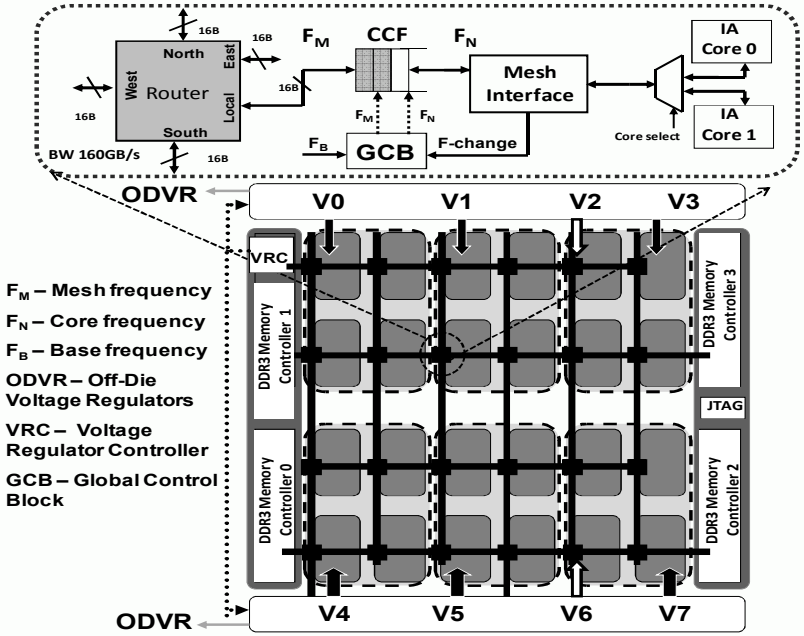
- Initial architecture: [24], MIT, 2007
- Tile:
  - local multi-core
  - DRAM, I/O...
- NoC between tiles
- Hierarchical design

  $\Rightarrow$ multi-core interferences + NoC interferences

Outline

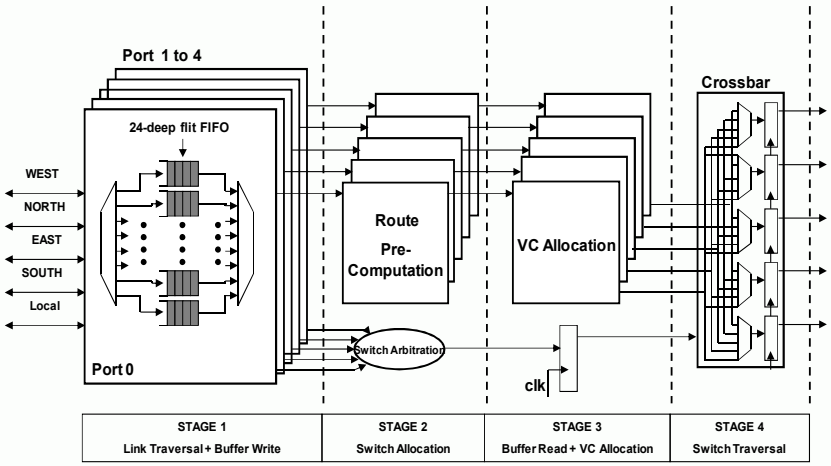
Outline

Intel SCC architecture



- experimental processor [22]
- 24 tiles
- 2 cores per tile
- 2Tb/s bisection bandwidth
- explicit message passing (but virtual global addressing)

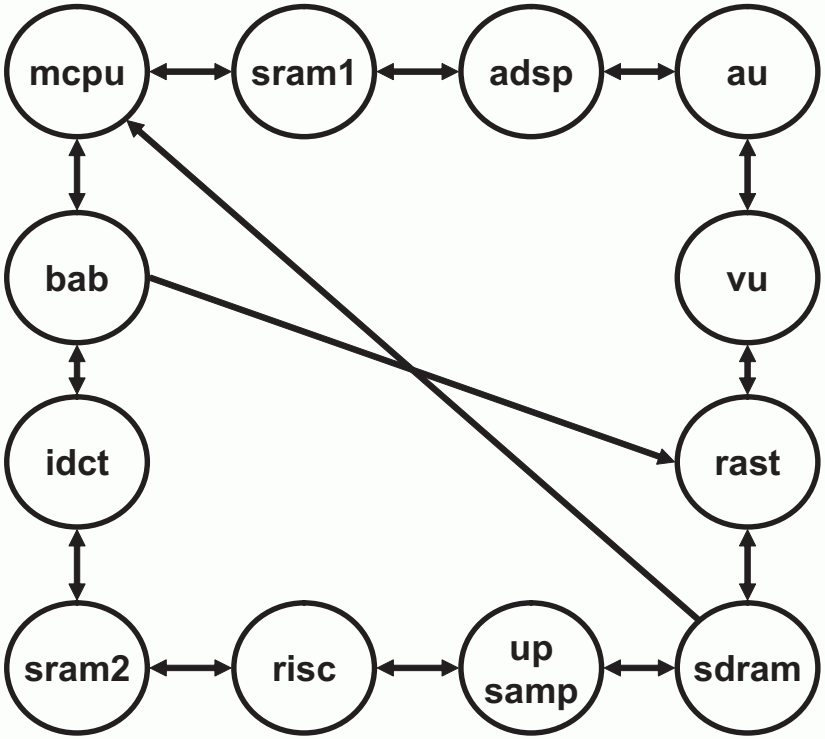
Intel SCC router



- Virtual Circuit forwarding
- 8 VC for the whole router
- crossbar output

Outline

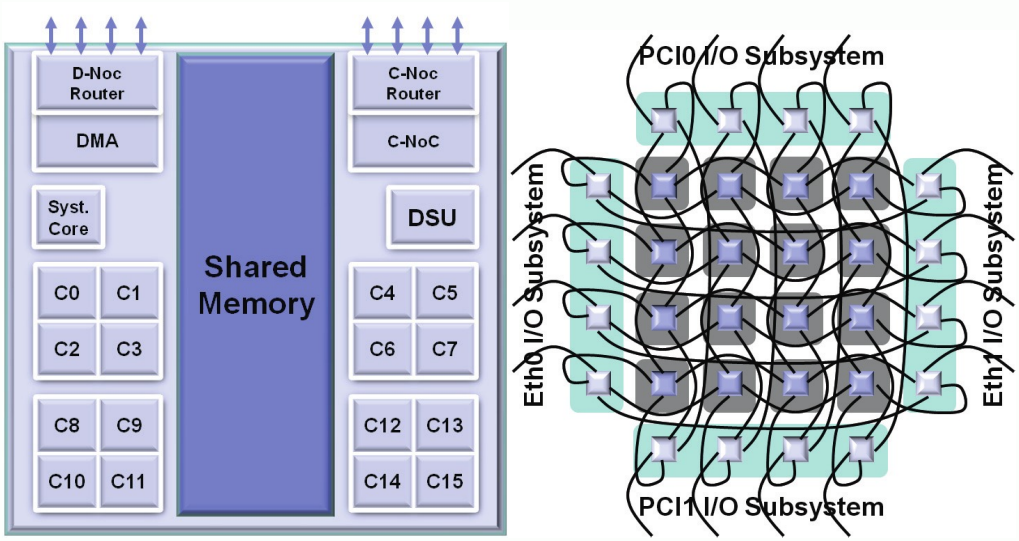
Spidergon STNoC



- A NoC for SoC (?) [5, 18]
- application specific topology
- Time-Triggered scheduling soon

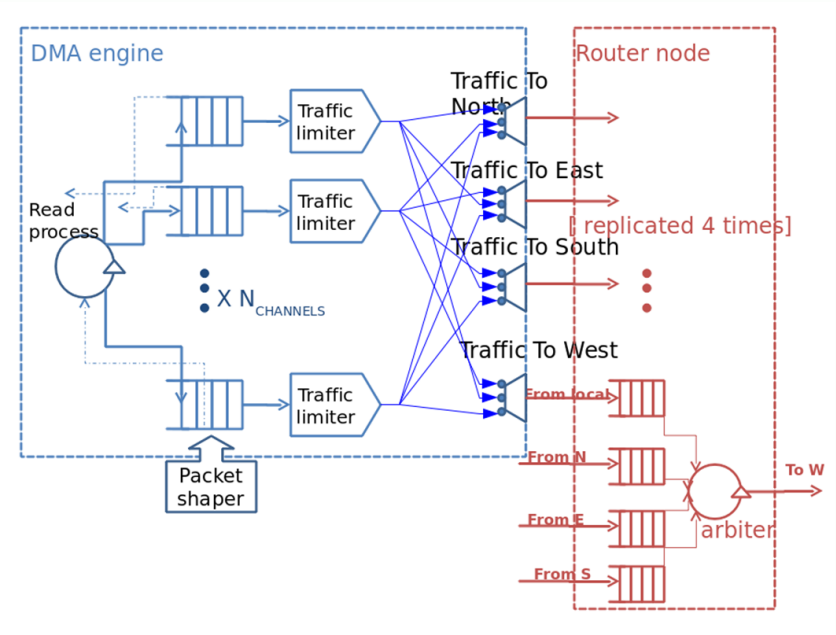
Outline

Kalray architecture



- A 256-cores chip [9]
- torus topology
- 16 tiles
- 16 “simple” cores per tile

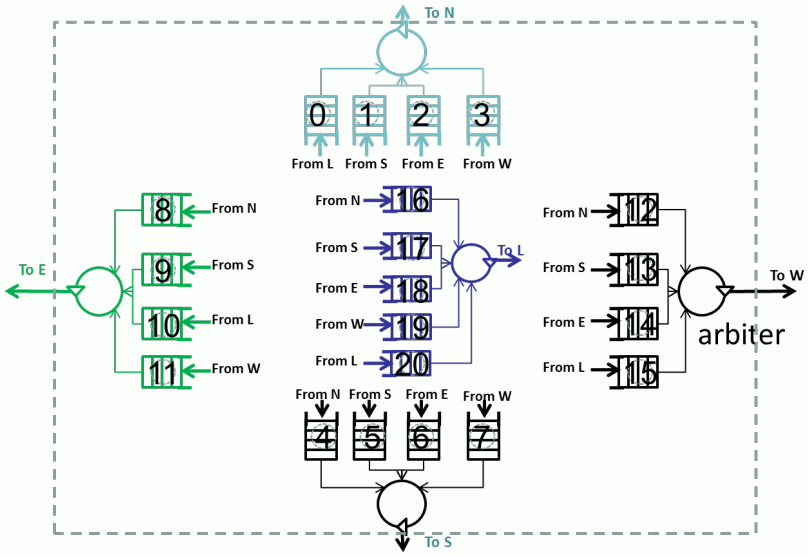
Kalray Network Adapter



- 8 channels [10]
- explicit communications
- per channel traffic limiter

$\Rightarrow$ HW support for latency computation

Kalray Network router



- virtual cut-through forwarding
- round-robin arbitration

Outline

Real-time with NoC?

Challenge:

- bound Worst case Interference Time (WCIT)

$\Rightarrow$ bound NoC Worst Case Traversal Time (WCTT)

Real-time with NoC?

Challenge:

- bound Worst case Interference Time (WCIT)

$\Rightarrow$ bound NoC Worst Case Traversal Time (WCTT)

Real-Time NoC:

- there will be one: TT extension of STMicro Spidergon STNoC
- there are some HW mechanisms
  - deactivation of cache coherency
  - bandwidth limiters

Real-time with NoC?

Challenge:

- bound Worst case Interference Time (WCIT)

$\Rightarrow$ bound NoC Worst Case Traversal Time (WCTT)

Real-Time NoC:

- there will be one: TT extension of STMicro Spidergon STNoC
- there are some HW mechanisms
  - deactivation of cache coherency
  - bandwidth limiters

Solutions:

- execution model
- analyse methods

Outline

Conclusion

The bad news

The good news

Conclusion

The bad news

- no large real-time processor market

The good news

Conclusion

The bad news

- no large real-time processor market
- neither NoC one

The good news

Conclusion

The bad news

- no large real-time processor market
- neither NoC one
- the NoC is a common shared resource

The good news

Conclusion

The bad news

- no large real-time processor market
- neither NoC one
- the NoC is a common shared resource

The good news

- cache coherence is no more affordable

$\Rightarrow$ less implicit communication

Conclusion

The bad news

- no large real-time processor market
- neither NoC one
- the NoC is a common shared resource

The good news

- cache coherence is no more affordable

$\Rightarrow$ less implicit communication

The times they are changing

- from shared memory to message passing

Outline

Outline

Some limits [4, 12]

Moore's law

The transistor density doubles every generation.

Pollack's rules

Performance (of a single core) is roughly proportional to *square root* of the number of transistors.

More limits [4, 12]

Amdahl's law

Given a program, with fraction $f \in [0, 1]$ that can be executed in parallel. Then, the speed-up with n cores is bounded by

$$\frac{1}{(1-f) + \frac{f}{n}} \tag{1}$$

More limits [4, 12]

Amdahl's law

Given a program, with fraction $f \in [0, 1]$ that can be executed in parallel. Then, the speed-up with n cores is bounded by

$$\frac{1}{(1-f)+\frac{f}{n}} \tag{1}$$

Gustafson's law

Given a program, with fraction $f \in [0, 1]$ that can be executed in parallel, n processors allows to handle a problem $\frac{n}{n+f(1-n)}$ larger.

More limits [4, 12]

Amdahl's law

Given a program, with fraction $f \in [0, 1]$ that can be executed in parallel. Then, the speed-up with n cores is bounded by

$$\frac{1}{(1-f) + \frac{f}{n}} \tag{1}$$

Gustafson's law

Given a program, with fraction $f \in [0, 1]$ that can be executed in parallel, n processors allows to handle a problem $\frac{n}{n+f(1-n)}$ larger.

Corollary

With more processors, programmers find more parallelism in problems...

The power limit

Power consumption

$$W = CV^2 f$$

The capacity (C) is technology dependant.
Cache memory uses few energy.

	Power	GFlop/Watt
Mono/Quadri-Core	150W	7GF/W
GPGPU	300W	10GF/W
Many-core	30W	70GF/W

Outline

Symmetric multicores ?

Forecast [12, 4]: next chips will have

- a few “large” cores for sequential part
- several “small” cores for parallel part
- on the same chip ? (multi-core vs GPGPU)

Bibliographie I


Julie Baro, Frédéric Boniol, Mikel Cordovilla, Eric Noulard, and Claire Pagetti.
Off-line (optimal) multiprocessor scheduling of dependent periodic tasks.
In *Proceedings of the 27th ACM Symposium on Applied Computing (SAC'2012)*, March 2012.
http://www.acm.org/conferences/sac/sac2012/.


Tobias Bjerregaard and Shankar Mahadevan.
A survey of research and practices of network-on-chip.
ACM Computing Surveys (CSUR), 38(1):1, 2006.


Frédéric Boniol, Hugues Cassé, Eric Noulard, and Claire Pagetti.
Deterministic execution model on cots hardware.
In *Architecture of Computing Systems - ARCS2012*, Lecture Notes in Computer Science. Springer, 2012.
http://www.arcs2012.tum.de/.


Shekhar Borkar and Andrew A. Chien.
The future of microprocessors.
Commun. ACM, 54:67–77, May 2011.

Bibliographie II

Marcello Coppola, Miltos D. Grammatikakis, Riccardo Locatelli, Giuseppe Maruccia, and Lorenzo Pieralisi.
Design of Cost-Efficient Interconnect Processing Units: Spidergon STNoC.
CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2008.

Mikel Cordovilla, Frédéric Boniol, Julien Forget, Eric Noulard, and Claire Pagetti.

Developing critical embedded systems on multicore architectures: the prelude-schedmcore toolset.
In *19th International Conference on Real-Time and Network Systems (RTNS 2011)*, Nantes, France, September 29-30 2011. IRCCyN lab, IRCCyN lab.
`http://hal.inria.fr/inria-00618587/en/`.

Mikel CORDOVILLA, Frédéric BONIOL, Eric NOULARD, and Claire PAGETTI.
Multiprocessor schedulability analyzer.
In *26th ACM Symposium of Applied Computing (Real-Time Systems Track)*, March 2011.

Bruno d'Ausbourg, Marc Boyer, Eric Noulard, and Claire Pagetti.
Deterministic execution on many-core platforms: application to the scc.
Technical report, Hasso-Planner Institute, December 2011.
In 4th MARC Symposium, Potsdam, Germany.

Bibliographie III

Benoît Dupont de Dinechin, Pierre Guironnet de Massas, Guillaume Lager, Clément Léger, Benjamin Orgogozo, Jérôme Reybert, and Thierry Strudel.
A distributed run-time environment for the kalray mppa®-256 integrated manycore processor.
Procedia Computer Science, 18(0):1654 – 1663, 2013.
2013 International Conference on Computational Science.

Benoît Dupont de Dinechin, Yves Durand, Duco van Amstel, and Alexandre Ghiti.
Guaranteed services of the noc of a manycore processor.
In *Proc. of the 7th Int. Workshop on Network on Chip Architectures (NoCArc'14)*, Cambridge, United Kingdom, December 2014.

Guy Durrieu, Madeleine FaugÃ¨re, Sylvain Girbal, Daniel Gracia PÃ©rez, Claire Pagetti, and Wolfgang Puffitsch.
Predictable flight management system implementation on a multicore processor.
In *Proceedings of the 7th Conference on Embedded Real Time Software and Systems (ERTS'14)*, 2014.

Mark D. Hill and Michael R. Marty.
Amdahl's law in the multicore era.
Computer, 41(7):30–38, July 2008.

Bibliographie IV

Angeliki Kritikakou, Olivier Baldellon, Claire Pagetti, Christine Rochange, and Matthieu Roy.
Run-time control to increase task parallelism in mixed-critical systems.
In *26th Euromicro Conference on Real-Time Systems (ECRTS'14)*, pages 119–128, July 2014.

Angeliki Kritikakou, Claire Pagetti, Christine Rochange, Matthieu Roy, Madeleine Faugère, Sylvain Girbal, and Daniel Gracia Pérez.
Distributed run-time wcet controller for concurrent critical tasks in mixed-critical systems.
In *Proceedings of the 22th International Conference on Real-Time and Network Systems (RTNS'14)*, pages 139–148, 2014.

Alessandra Melani, Eric Noulard, and Luca Santinelli.
Learning from probabilities: Dependences within real-time systems.
In *Proceedings of the 18th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'2013)*, September 2013.
`http://www.etfa2013.org/`.

Bibliographie V

Eric Noulard.
Real-time scheduling on multi/many-processor: from specification to real-time execution, August 2015.
Invited presentation to ETR'21015, http://etr2015.irisa.fr.

Claire Pagetti, David Saussié, Romain Gratia, Eric Noulard, and Pierre Siron.
The ROSACE case study: From simulink specification to multi/many-core execution.
In *Proceedings of the 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'2014)*, April 2014.
`http://2014.rtas.org/`.

Gianluca Palermo, Giovanni Mariani, Cristina Silvano, Riccardo Locatelli, and Marcello Coppola.
Mapping and topology customization approaches for application-specific stnoc designs.
In *Application-specific Systems, Architectures and Processors, 2007. ASAP. IEEE International Conf. on*, pages 61–68. IEEE, 2007.

Wolfgang Puffitsch, Eric Noulard, and Claire Pagetti.
Off-line mapping of multi-rate dependent task sets to many-core platforms.
submitted to journal (currently reviewed).

Bibliographie VI


Wolfgang Puffitsch, Eric Noulard, and Claire Pagetti.
Explicit precedence constraints in safety-critical java.
In *Proceedings of the 11th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES'2013)*, October 2013.
`http://jtres2013.atego.com/`.


Wolfgang Puffitsch, Eric Noulard, and Claire Pagetti.
Mapping a multi-rate synchronous language to a many-core processor.
In *Proceedings of the 19th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'2013)*, April 2013.
`http://www.cister.isep.ipp.pt/rtas2013/`.


Praveen Salihundam, Shailendra Jain, Tiju Jacob, Shasi Kumar, Vasantha Erraguntla, Yatin Hoskote, Sriram Vangal, Gregory Ruhl, and Nitin Borkar.
A 2 Tb/s 6x4 mesh network for a single-chip cloud computer with DVFS in 45 nm CMOS.
IEEE journal of solid-state circuits, 46(4):757–766, April 2011.

Bibliographie VII

Luca Santinelli, FrÃ©dÃ©ric Boniol, Eric Noulard, Claire Pagetti, and Woflgang Puffitsch.
Integrated development framework for safety-critical embedded systems.
In *Proceedings of the 1st Formal Methods for Timing Verification Workshop (FMTV'2014)*, May 2014.
http://www.merge-project.eu/fmtv14-workshop.

Michael Bedford Taylor.
Tiled Microprocessors.
PhD thesis, Massachusetts Institute of Technology, 2007.

Axel Thomsen, Boris Murmann, Andreia Cathelin, Aarno Pärssinen, Daniel Friedman, T. J. Watson, Stephen Kosonocky, Stefaon Rusu, Joo Sun Choi, Makoto Ikeda, and Eugenio Cantatore.
Isscc 2015 trends.
Technical report, IEEE international solid-state circuits conference, 2015.