From multi to many-core: network on chip challenges

Marc Boyer

ONERA – The French Aerospace Lab

24 mars 2016
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
Outline

Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC ?

Conclusion
Outline

Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC ?

Conclusion
Outline

Many-cores are arriving

Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC ?

Conclusion
Some trends [TMC+15]

- IBM’s high-frequency 8-core, 16-thread System z mainframe processor in 22nm SOI with 64MB of eDRAM L3 cache and 4MB/core eDRAM L2 cache.
- Oracle’s SPARC M7 processor implements 32 S4 cores, a 1.6TB/s bandwidth 64MB L3 Cache and a 0.5TB/s data bandwidth on-chip network (OCN) to deliver more than 3.0x throughput compared to its predecessor.
- Intel’s next generation Xeon processor supports 18 dual-threaded 64b Haswell cores, 45MB L3 cache, 4 DDR4-2133MHz memory channels, 40 8GT/s PCIe lanes, and 60 9.6GT/s QPI lanes.

The maximum core clock frequency seems to have saturated in the range of 5-6GHz, primarily limited by thermal considerations. The nominal operating frequency of the power-limited processors this year is around 3.5GHz. Core counts per die are typically above 10, with increases appearing to slow in recent years. Cache size growth continues, with modern chips incorporating tens of MB on-die.

The trend towards digital phase-locked loops (PLL) and delay locked loops (DLL) to better exploit nanometer feature scaling, and reduce power and area continues. Through use of highly innovative architectural and circuit design techniques, the features of these digital PLLs and DLLs have improved significantly over the recent past. Another new trend evident this year is towards fully digital PLLs being synthesizable and operated with non-LC oscillators. The diagram below shows the jitter performance vs. energy cost for PLLs and multiplying DLLs (MDLL).
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

Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
Some limits [BC11, HM08]

**Moore’s law**

The transistor density doubles every generation.

**Pollack’s rules**

Performance (of a single core) is roughly proportional to \( \sqrt{\text{number of transistors}} \).
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

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

(1)
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

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

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 [BC11, HM08]

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

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

(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

\[ W = CV^2f \]

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

<table>
<thead>
<tr>
<th></th>
<th>Power</th>
<th>GFlop/Watt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mono/Quadri-Core</td>
<td>150W</td>
<td>7GF/W</td>
</tr>
<tr>
<td>GPGPU</td>
<td>300W</td>
<td>10GF/W</td>
</tr>
<tr>
<td>Many-core</td>
<td>30W</td>
<td>70GF/W</td>
</tr>
</tbody>
</table>
Outline

Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
Symmetric multicores?

Forecast [HM08, BC11]: 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)
Outline

Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
Outline

Many-cores are arriving
- Trends
- Theoretical limits
- Architecture impact

Network on Chip
- Overview
- Routing
- Contention
- Other solutions
- Tiles

Four selected solutions
- Intel SCC
- Spidergon STNoC
- The Kalray MPPA
- Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
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

Bus: shared resource
Pt-to-pt: does not scale
NoC:
  - set of shared resources
  - allow parallel communications
A common 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* (≈ 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* ($\approx 64$ bits)
  - also flow control
Topologies

One simple way to distinguish different regular topologies is in terms of dimensions (Figure 9), first described by Dally [1990] for multicomputer networks.

Formally, the k-ary n-cube is a network topology that can be laid out on a chip surface (a 2-dimensional plane) for a larger, scalable network. For more complex structures such as trees, finding the optimal layout is a challenge on its own right.

Tree-based topologies are useful for exploiting locality of traffic. Besides the form, the nature of links adds an additional aspect to the topology. In k-ary tree networks, popular NoC topologies based on the nature of link are the mesh and the fat tree are two alternate regular forms of network topology that can be laid out on a chip surface (a 2-dimensional plane) dictably for increasing size of regular forms of topology. Most NoCs implement regular forms of topology. Generally, mesh topology makes better use of links (utilization), while that it has longer delays between routing nodes. Figure 9 shows examples of regular forms of topology.

Irregular forms of topologies are derived by mixing different forms in a hierarchical, hybrid, or asymmetric fashion as seen in Figure 10. Irregular forms of topologies scale predictably with regard to area and power. Examples are (a) 4-ary 2-cube mesh, (b) 4-ary 2-cube torus and (c) binary k-ary 2-cube, commonly known as grid-based topologies. The Octagon NoC for example, k-ary n-cube (grid-type), where

The network area and power consumption scales predictably for increasing size of regular forms of topology. Most NoCs implement regular forms of topology. Generally, mesh topology makes better use of links (utilization), while that it has longer delays between routing nodes. Figure 9 shows examples of regular forms of topology.
Outline

Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
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?

Routing:
Routing:

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

- **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: sending the same data to several cores

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

Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC ?

Conclusion
Router: managing contention

They always are contentions

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

- large set of solutions
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
  ⇒ link flow control
- allow large messages / packets
- implies blocking, and even dead-locks
Wormhole illustration: single flow

Time: 0

a b c d e f
Wormhole illustration: single flow
Wormhole illustration: single flow
Wormhole illustration: single flow

Time: 3
Wormhole illustration: single flow
Wormhole illustration: single flow

Diagram showing a wormhole with points A, B, C, D, E, F, G, and H. The diagram includes labels a, b, c, d, e, and f with a time marker at 5.
Wormhole illustration: single flow

Time: 6
Wormhole illustration: single flow

Time: 7
Wormhole illustration: single flow
Wormhole illustration: interfering flows
Wormhole illustration: interfering flows
Wormhole illustration: interfering flows
Wormhole illustration: interfering flows
Wormhole illustration: interfering flows
Wormhole illustration: interfering flows
Wormhole illustration: interfering flows

Time: 6
Wormhole illustration: interfering flows
Wormhole illustration: interfering flows
Wormhole illustration: interfering flows

![Diagram of wormhole with interfering flows]
Wormhole illustration: interfering flows
Wormhole illustration: deadlock
Wormhole illustration: deadlock
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

<table>
<thead>
<tr>
<th>Forwarding</th>
<th>Per node cost</th>
<th>Latency</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>store &amp; forward</td>
<td>packet</td>
<td>packet</td>
<td>Common in networks</td>
</tr>
<tr>
<td>wormhole</td>
<td>header</td>
<td>header</td>
<td>Blocking</td>
</tr>
<tr>
<td>virtual cut-through</td>
<td>header</td>
<td>packet</td>
<td>A trade-off (?)</td>
</tr>
</tbody>
</table>

- 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

Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention

Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC ?

Conclusion
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

Many-cores are arriving
   Trends
   Theoretical limits
   Architecture impact

Network on Chip
   Overview
   Routing
   Contention
   Other solutions

Tiles

Four selected solutions
   Intel SCC
   Spidergon STNoC
   The Kalray MPPA
   Intel Xeon Phi coprocessor

A real-time NoC ?

Conclusion
Tile-based solutions

- Initial architecture: [Tay07], MIT, 2007
- Tile:
  - local multi-core
  - DRAM, I/O...
- NoC between tiles
- Hierarchical design
  ⇒ multi-core interferences + NoC interferences
Outline

Many-cores are arriving
Trends
Theoretical limits
Architecture impact

Network on Chip
Overview
Routing
Contention
Other solutions
Tiles

Four selected solutions
Intel SCC
Spidergon STNoC
The Kalray MPPA
Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
Outline

Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
Intel SCC architecture

- experimental processor [SJJ+11]
- 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

![Diagram of Intel SCC router](image)
Outline

Many-cores are arriving
- Trends
- Theoretical limits
- Architecture impact

Network on Chip
- Overview
- Routing
- Contention
- Other solutions
- Tiles

Four selected solutions
- Intel SCC
- Spidergon STNoC
- The Kalray MPPA
- Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
A patented topology: Spidergon
- logical view: ring + diameters links
- less links than full mesh

classical switching: wormhole + VC

A NoC for SoC (?) [CGL+08, PMS+07, MSVOK07]

also application specific topology (subset of Spidergon links)

Time-Triggered scheduling soon
Outline

- Many-cores are arriving
  - Trends
  - Theoretical limits
  - Architecture impact

- Network on Chip
  - Overview
  - Routing
  - Contention
  - Other solutions
  - Tiles

Four selected solutions
- Intel SCC
- Spidergon STNoC
- The Kalray MPPA
- Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
Kalray architecture

- A 256-cores chip [dDdML+13]
- torus topology
- 16 tiles
- 16 “simple” cores per tile
Kalray Network Adapter

- 8 channels [DdDDvAG14]
- explicit communications
- per channel traffic limiter

⇒ HW support for latency computation
virtual cut-through forwarding

round-robin arbitration
Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
A different approach

- A co-processor
- > 50 cores
- 8GB GDDR5 memory
- Ring NoC
- Coherent caches
The interconnection ring

Interconnect

Core
L2
TD
Core
L2
TD
Core
L2
TD
Core
L2
TD
BL - 64 Bytes
AD
AK
Command and Address
Coherence and Credits
BL – 64 Bytes

Copyright © 2012 Intel Corporation. All rights reserved.
Outline

Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC ?

Conclusion
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)
  \[ \Rightarrow \text{bound NoC Worst Case Traversal Time (WCTT)} \]

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)

⇒ 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
Many-cores are arriving
  Trends
  Theoretical limits
  Architecture impact

Network on Chip
  Overview
  Routing
  Contention
  Other solutions
  Tiles

Four selected solutions
  Intel SCC
  Spidergon STNoC
  The Kalray MPPA
  Intel Xeon Phi coprocessor

A real-time NoC?

Conclusion
The bad news

The good news
Conclusion

The bad news
- no large real-time processor market

The good news

⇒ less implicit communication

The times they are changing from shared memory to message passing
Conclusion

The bad news

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

The good news

⇒ less implicit communication

The times they are changing from shared memory to message passing
Conclusion

The bad news

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

The good news

⇒ less implicit communication

The times they are changing from shared memory to message passing
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
  ⇒ 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
  ⇒ less implicit communication

The times they are changing

- from shared memory to message passing


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


