Networks on Chip (NOC)

Ran Ginosar

Prof. EE & CS
Technion-Israel Institute of Technology
ran@ee.technion.ac.il
www.ee.technion.ac.il/~ran

CEO
Ramon Chips, Ltd.
ran@ramon-chips.com

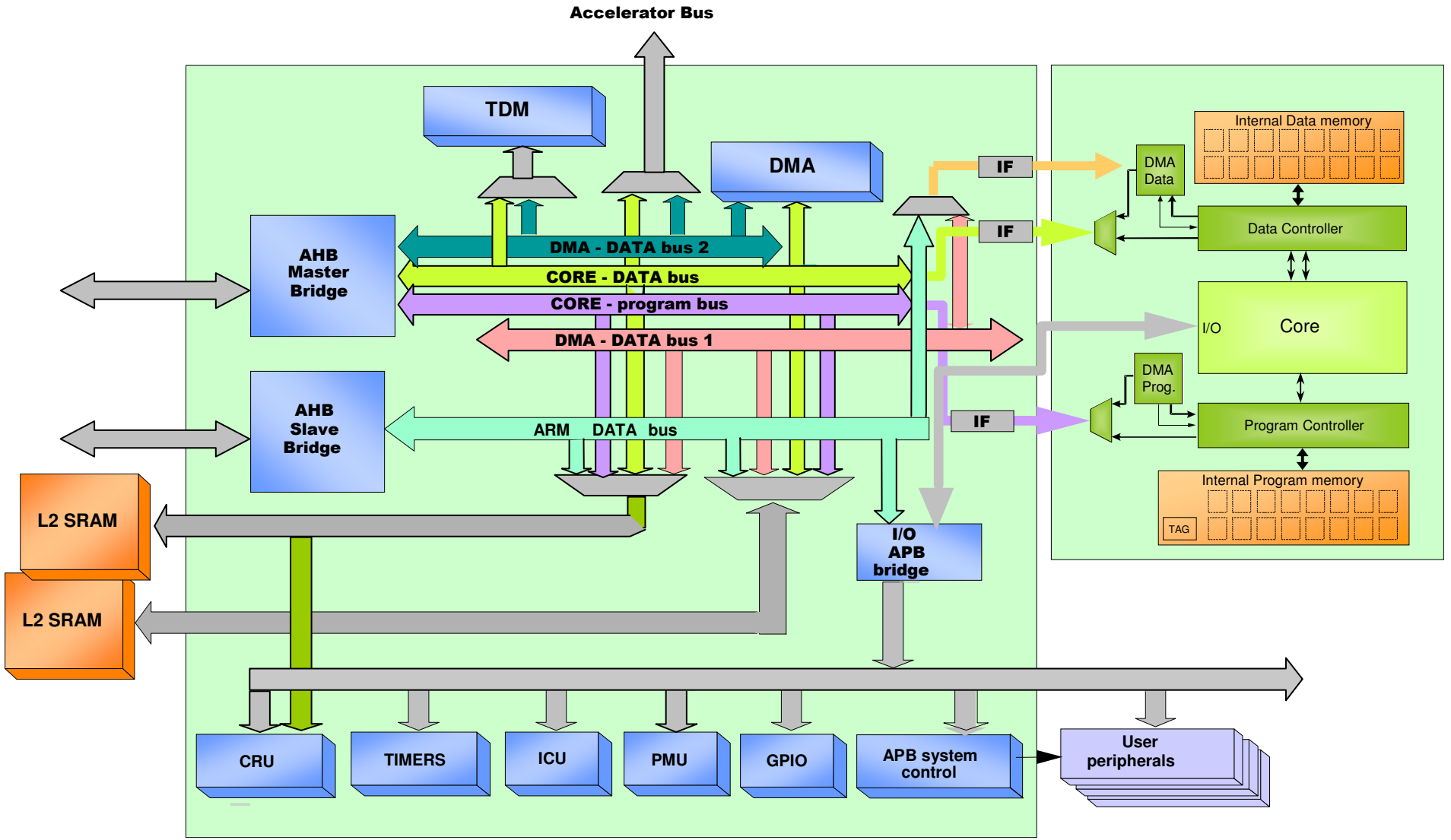
A longer tutorial on NOC: http://circuit.ucsd.edu/~nocs2009/tutorials.php

Acknowledgement

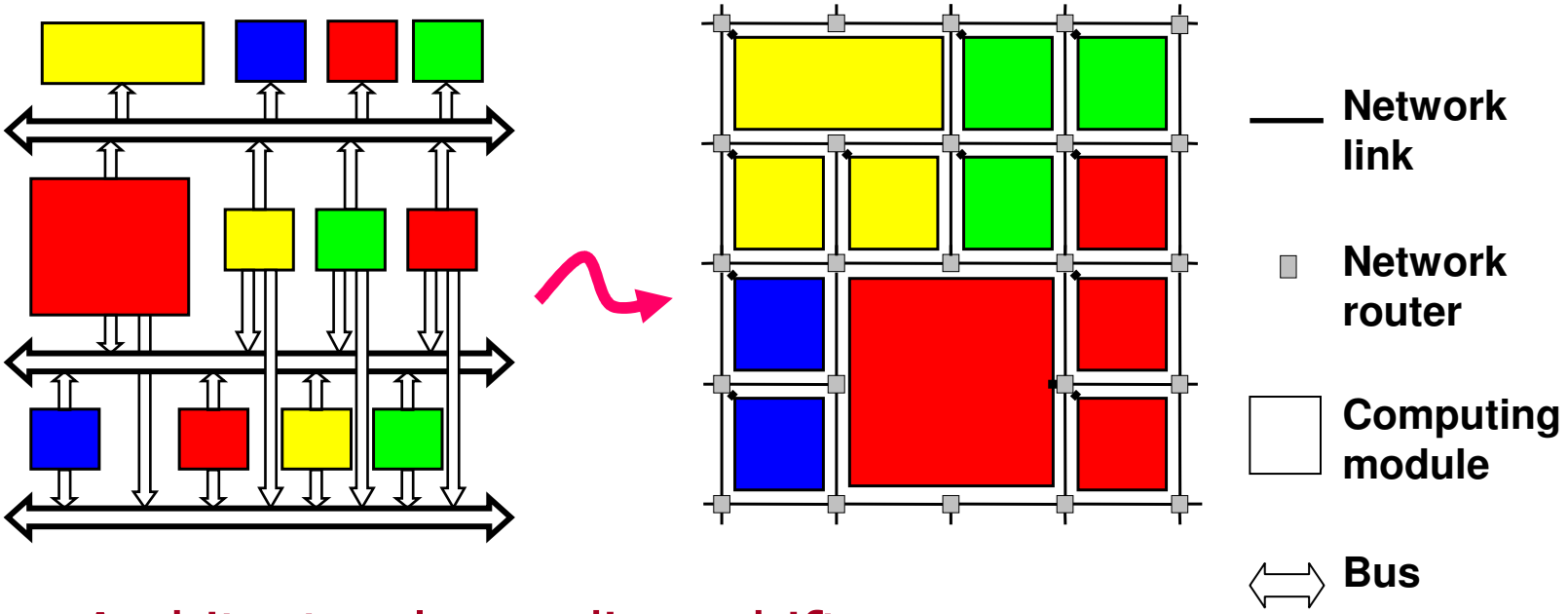
- Large research group at the Technion
  - Israel Cidon (Networking)
  - Ran Ginosar (VLSI)
  - Idit Keidar (Distributed Systems)
  - Isaac Kesslassy (Networking)
  - Avinoam Kolodny (VLSI)
  - Uri Weiser (Architecture)
- Past and present graduate students
  - Evgeny Bolotin, Roman Gindin, Reuven Dobkin, Zvika Guz, Ran Manevich, Arkadiy Morgenshtein, Zigi Walter, Asaf Baron, Dmitry Vainbrand
- Support by companies and funding agencies
  - Intel, Freescale, CEVA, Zoran, Israel government, EU FP7, USA SRC

- Large international research community since ~2000
  - Lots of literature
- Several companies making + using NoC

Buses are becoming spaghetti

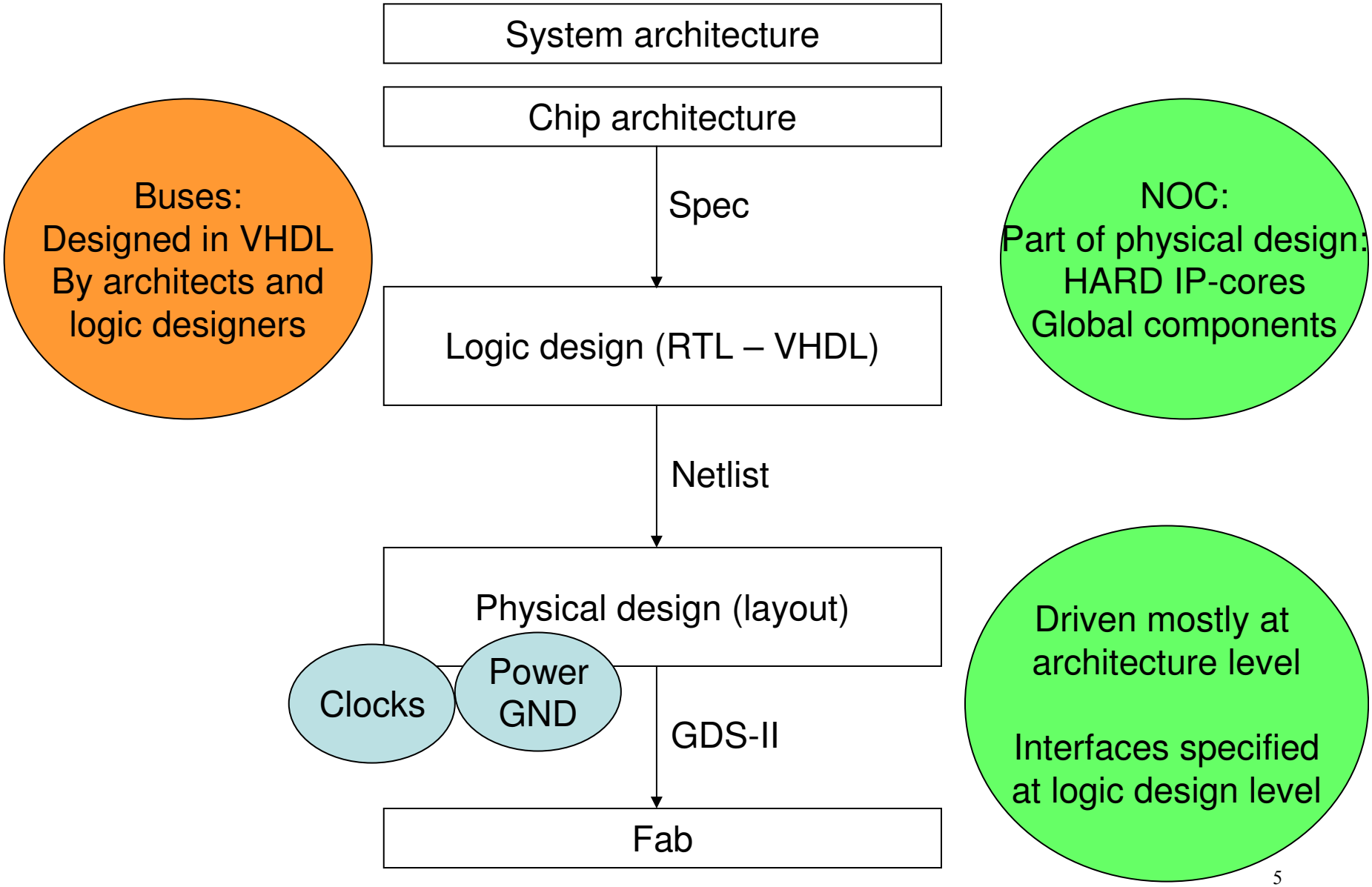


The NoC Paradigm Shift

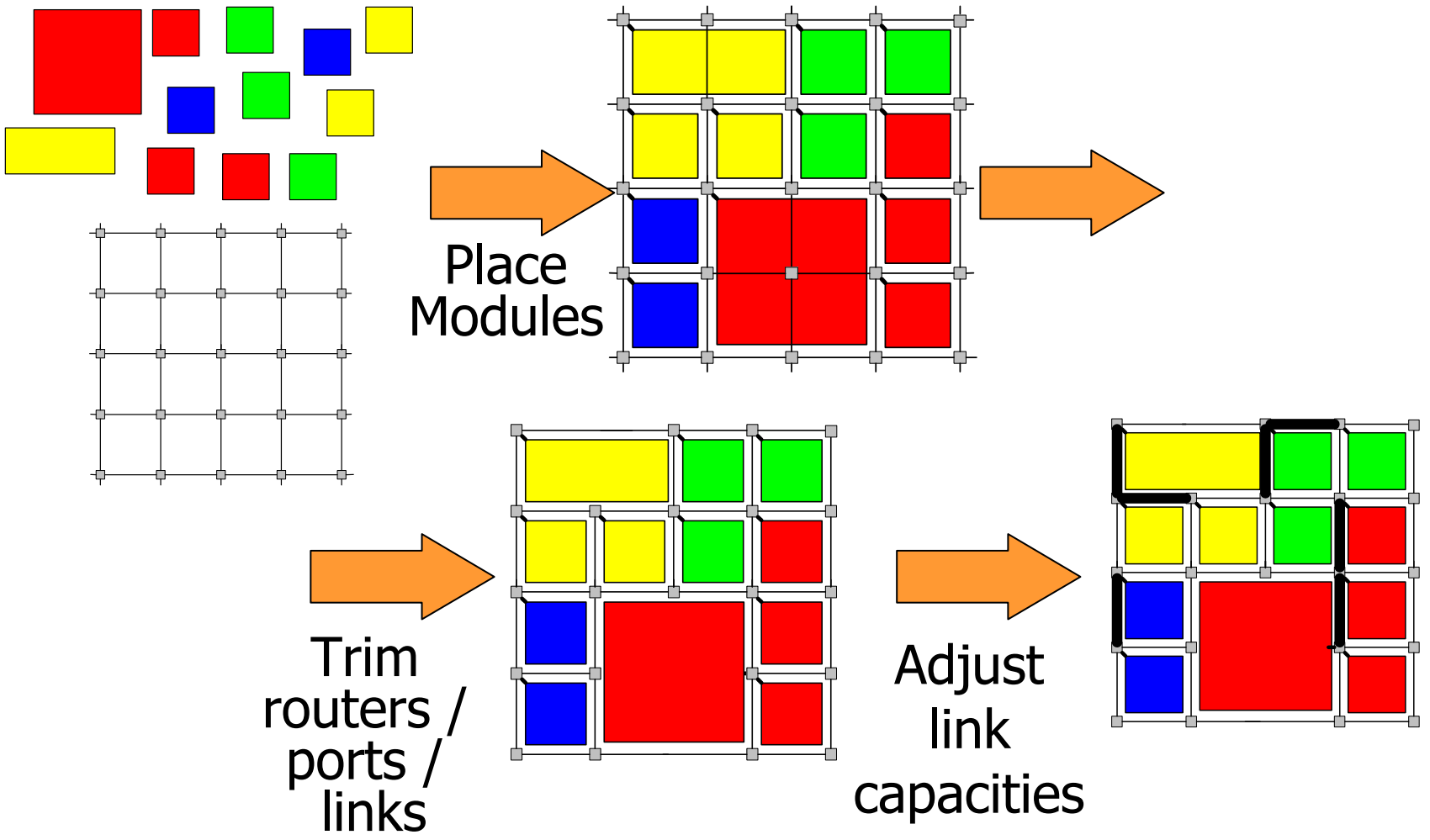


- Architectural paradigm shift
  - Replace the spaghetti by a customized network
- Usage paradigm shift
  - Pack everything in packets
- Organizational paradigm shift
  - Confiscate communications from logic designers
  - Move it to physical design

Organizational Paradigm Shift

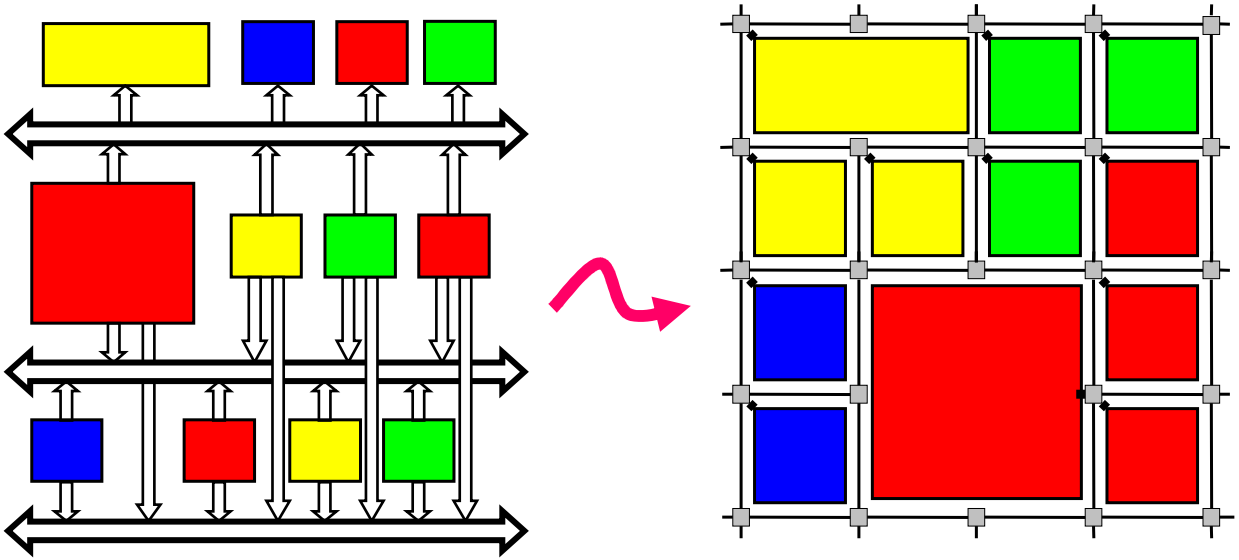


How is it designed?



- 3-way collaboration: Architects, logic designers, backend
- Requires novel special CAD !

Why go there?

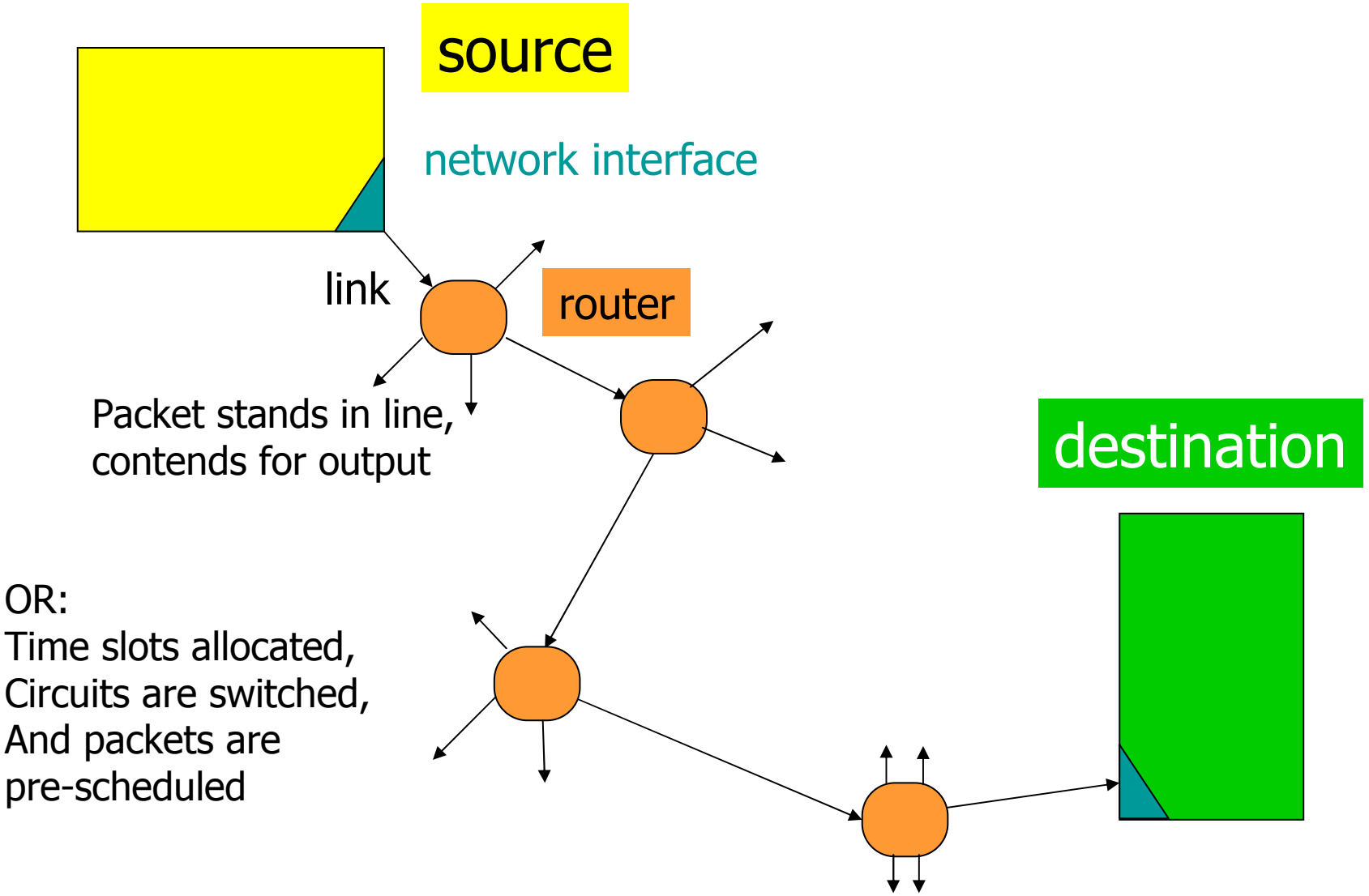


- Efficient sharing of wires
- Lower area / lower power / faster operation
- Shorter design time, lower design effort
- Scalability
- Enable using custom circuits for comm

NoC is already here!

- Companies use (try) it
  - Freescale, NXP, STM, Infineon, Intel, …
- Companies sell it
  - Sonics (USA), Arteris (France), Silistix (UK), …
- Annual IEEE Conference
  - NOSC 2007: Princeton, USA
  - NOCS 2008: Newcastle, UK
  - NOCS 2009: San Diego, USA
  - NOCS 2010: Grenoble, France
  - NOCS 2011: Pittsburgh, USA

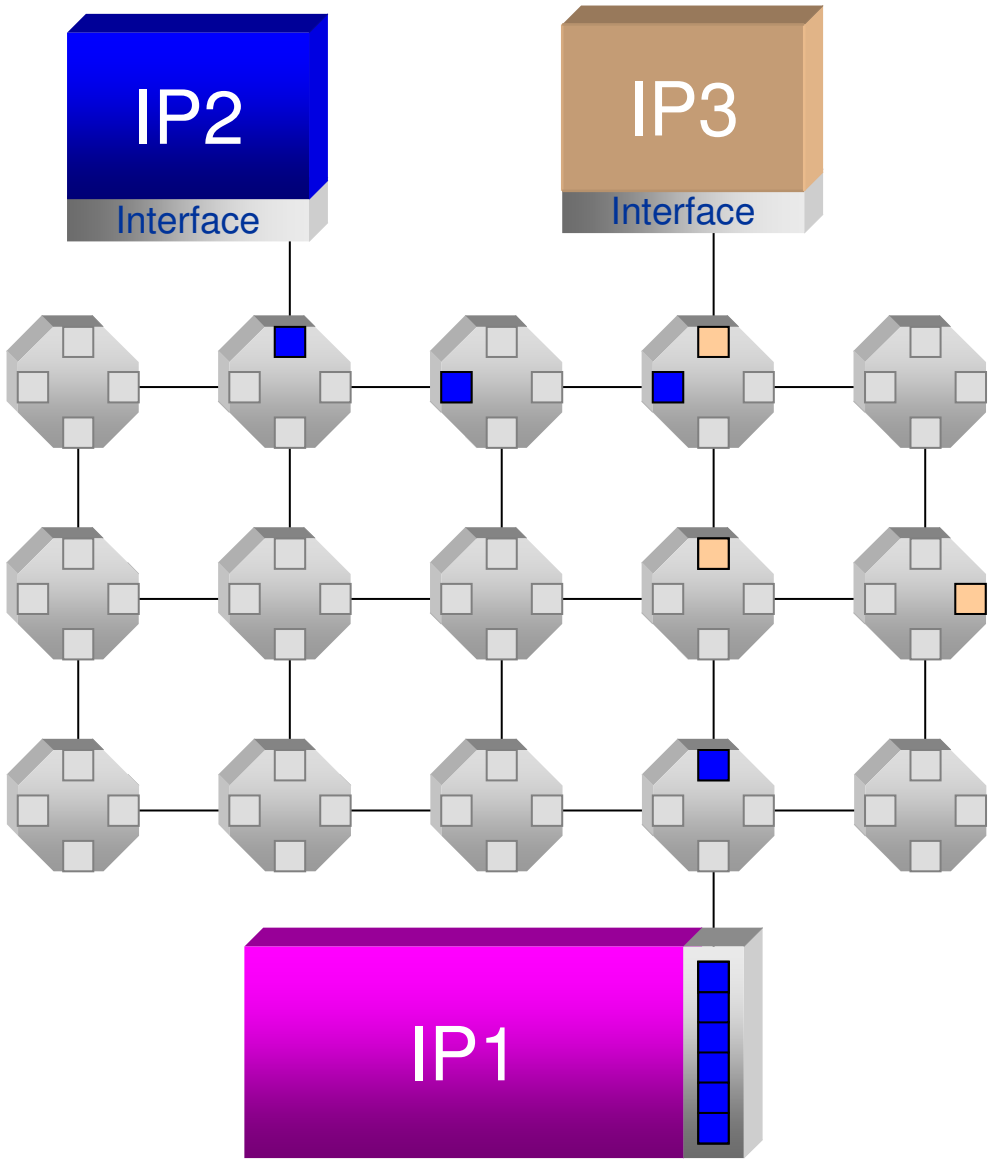
What's in the NoC?



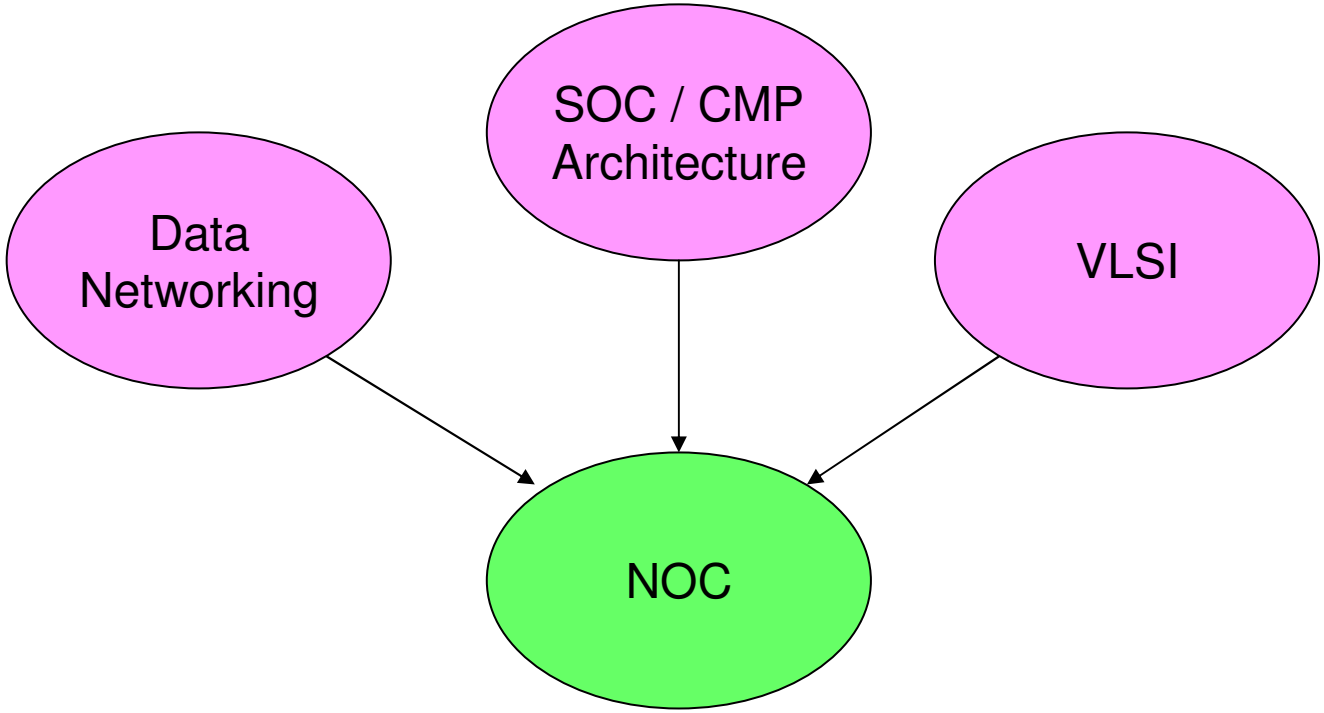
What flows in the NoC?

- Basic unit exchanged by end-points is the PACKET
- Packets broken into many FLITs
  - "flow control unit"
  - Typically # bits = # wires in each link (variations)
  - Typically contains some ID bits, needed by each switch along the path:
    - Head / body / tail
    - VC #
    - SL #
- FLITs typically sent in a sequence, making a "worm" going through wormhole.
- Unlike live worms, FLITs of different packets may interleave on same link
  - Routers know who's who

FLIT interleaving



Merging of disciplines

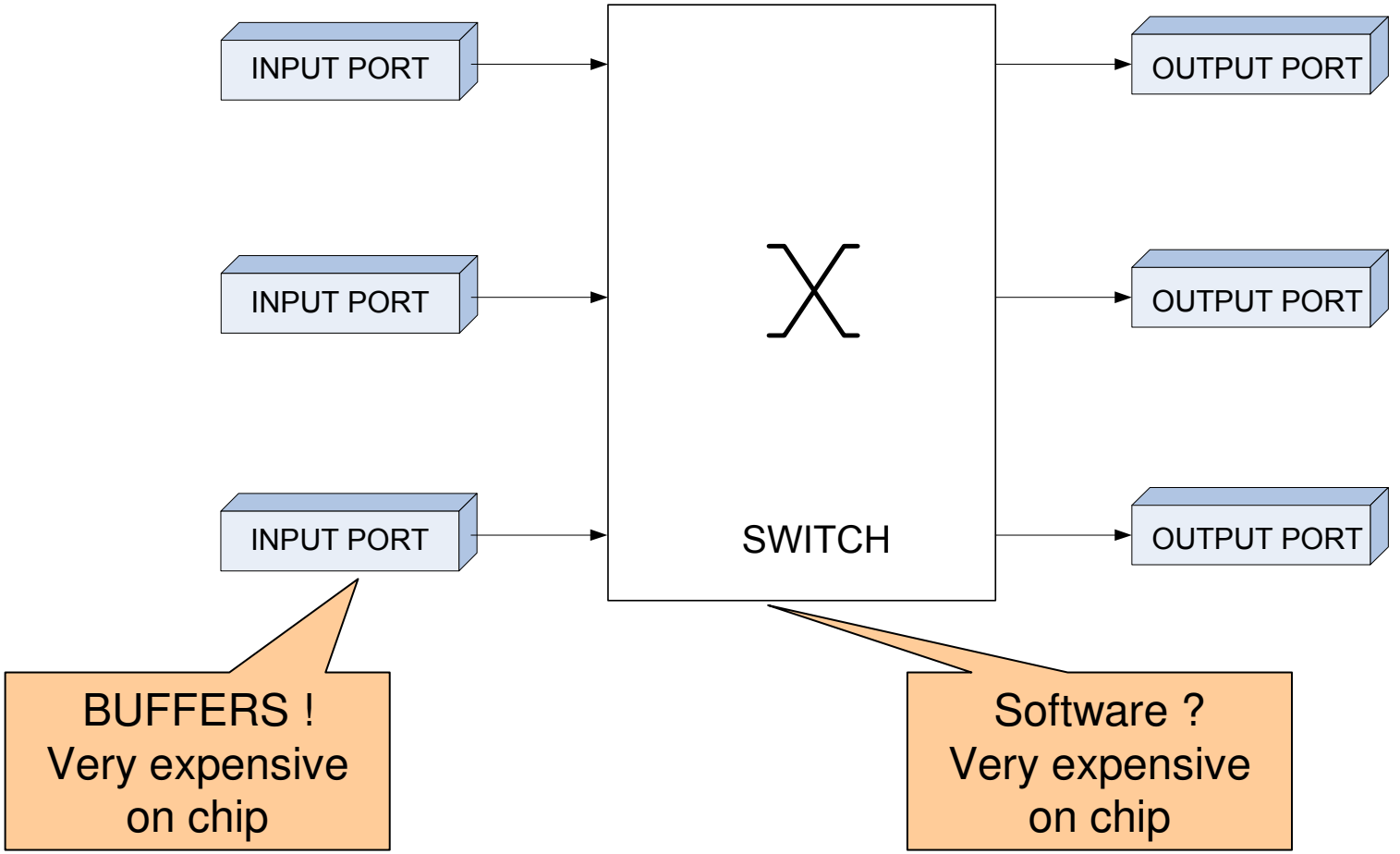


➔ confusion of terminology

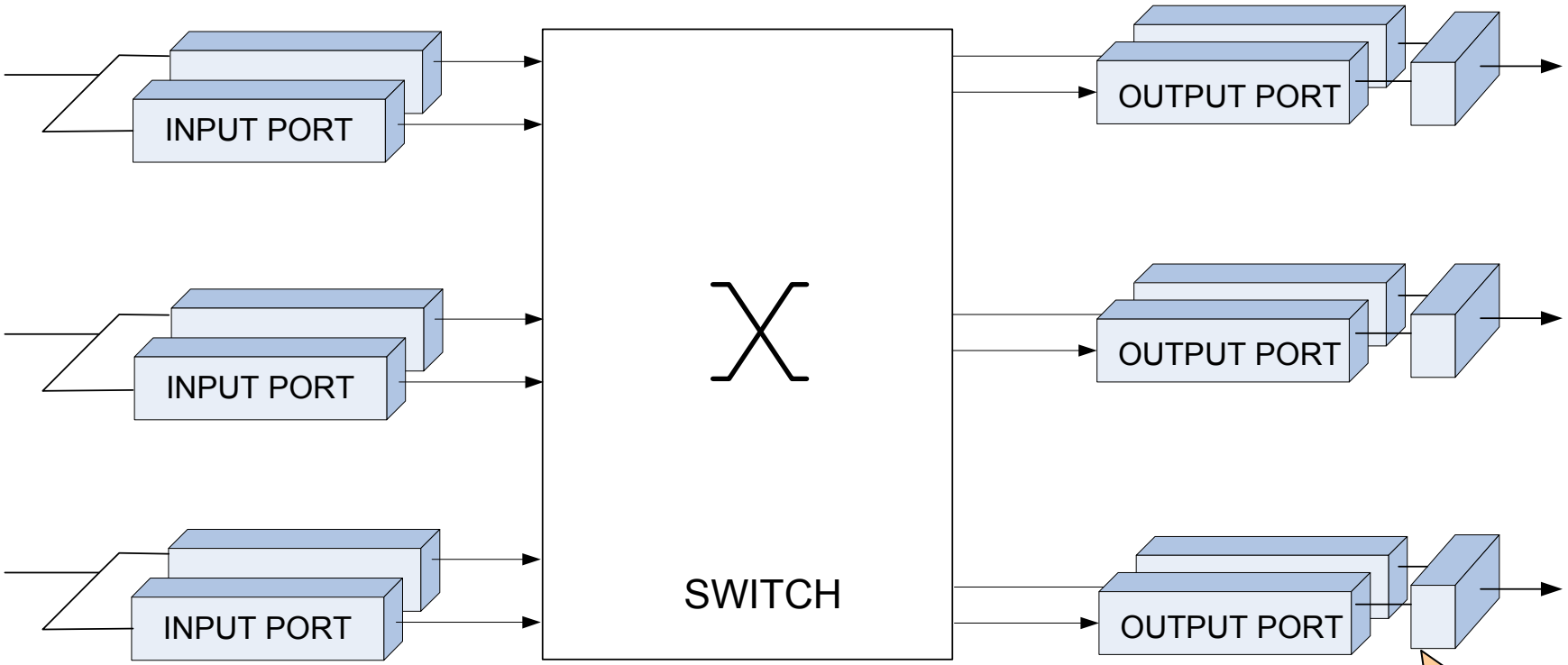
NoC vs. Off-chip Networks

NoC	Off-Chip Networks
• Main costs power & area • Wires are relatively cheap • Prefer simple hardware • Latency is critical • Traffic may be known a-priori • Design time specialization • Custom NoCs are possible • No faults, no changes	• Power and area negligible • Cost is in the links • Uses complex software • Latency is tolerable • Traffic/applications unknown • Changes at runtime • Adherence to standards • Faults and changes

Simplest NoC router: Single level

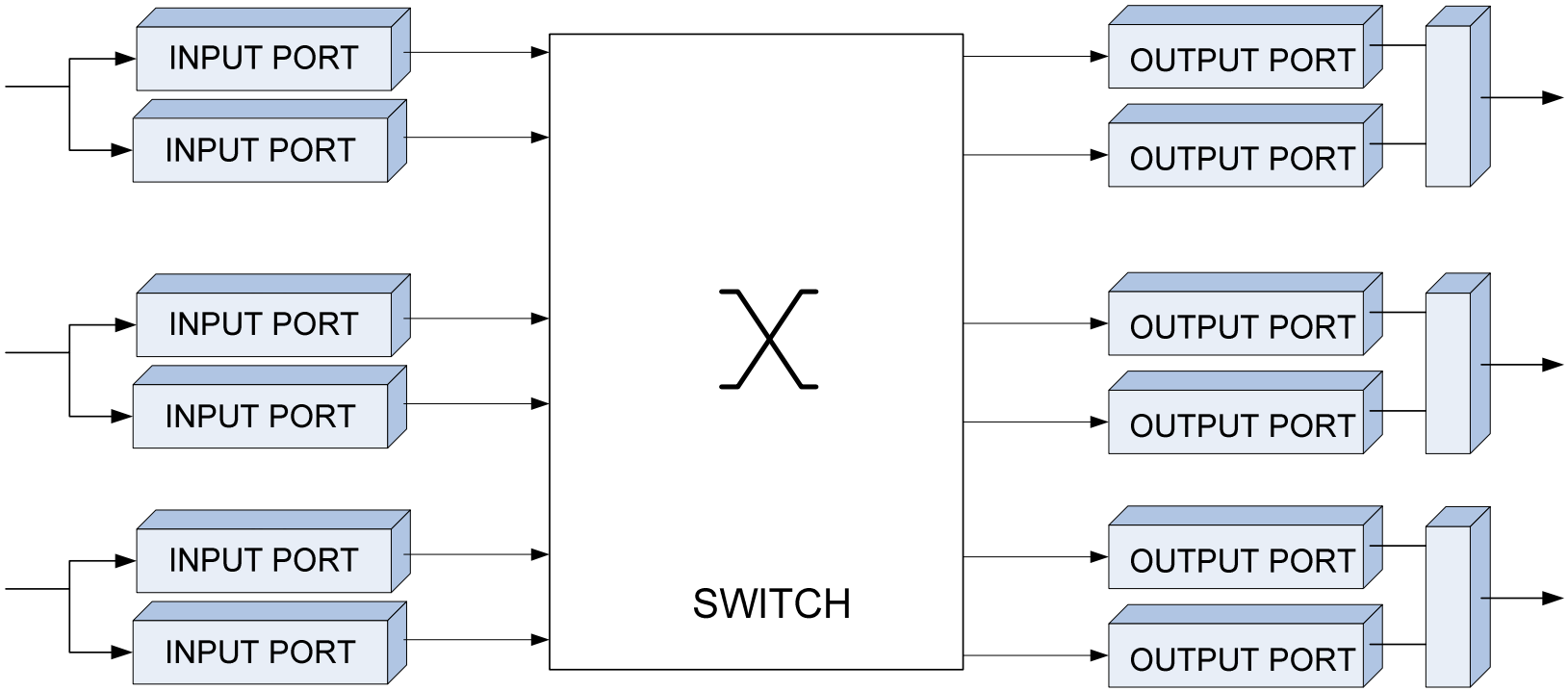


Virtual Channels (VC): Multiple same-priority levels



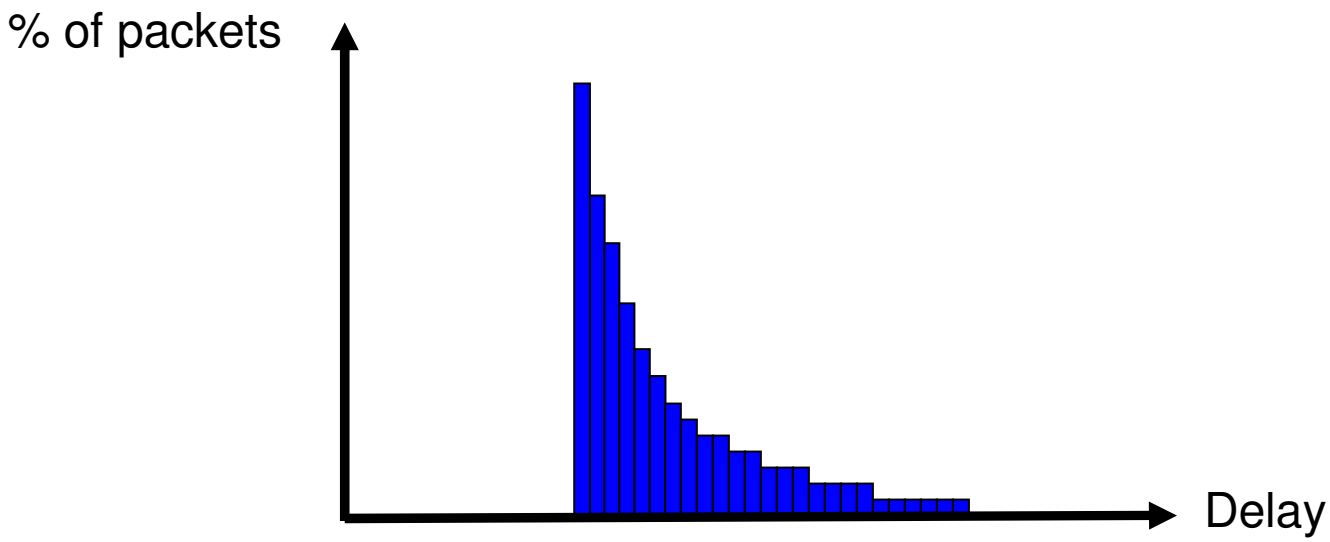
Both VC flits traverse the SAME wires

Service Levels (a.k.a. VC….): Multiple priority levels



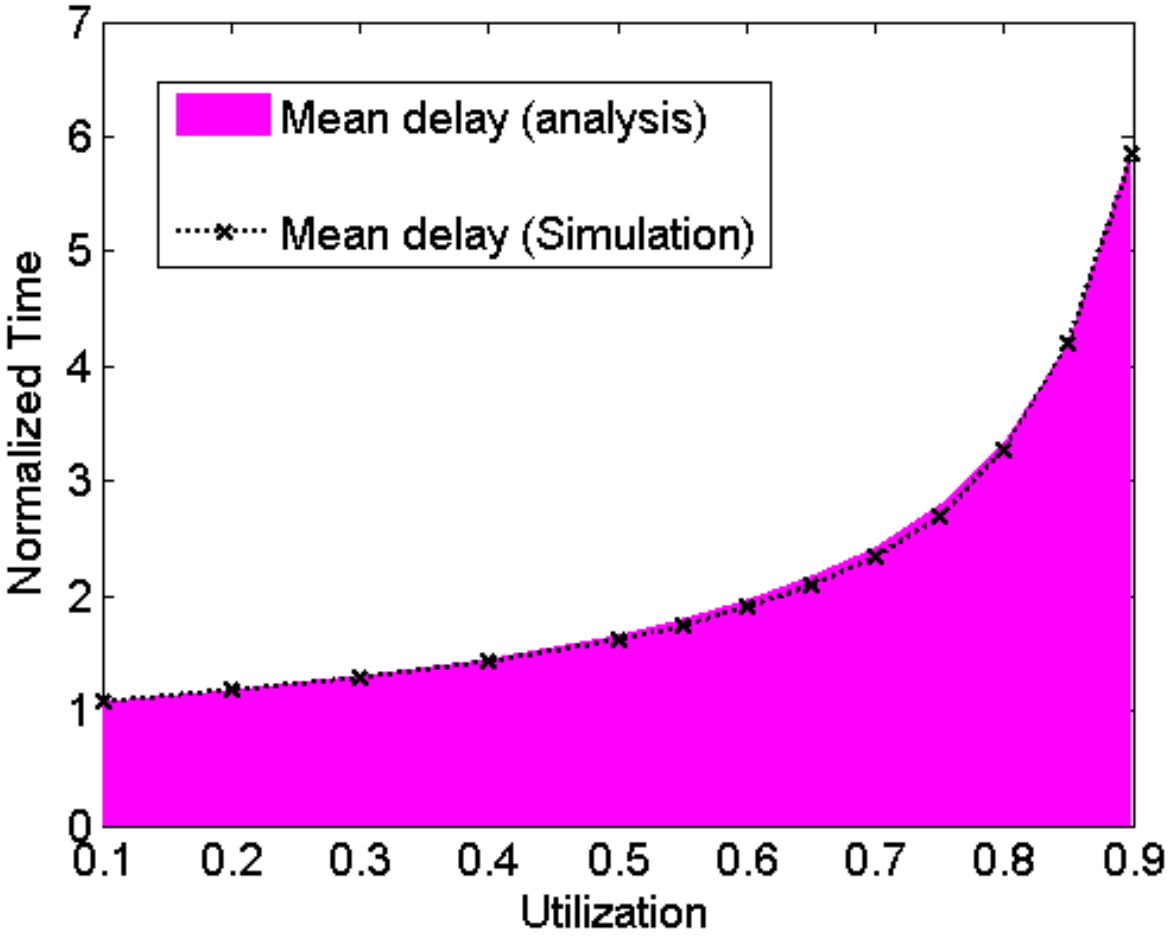
Imagine: Both VC and SL (two dimensions)

Statistical network delay



- Some packets delayed longer than others
  - Due to blocking
- Guaranteed service NoC can eliminate the uncertainty

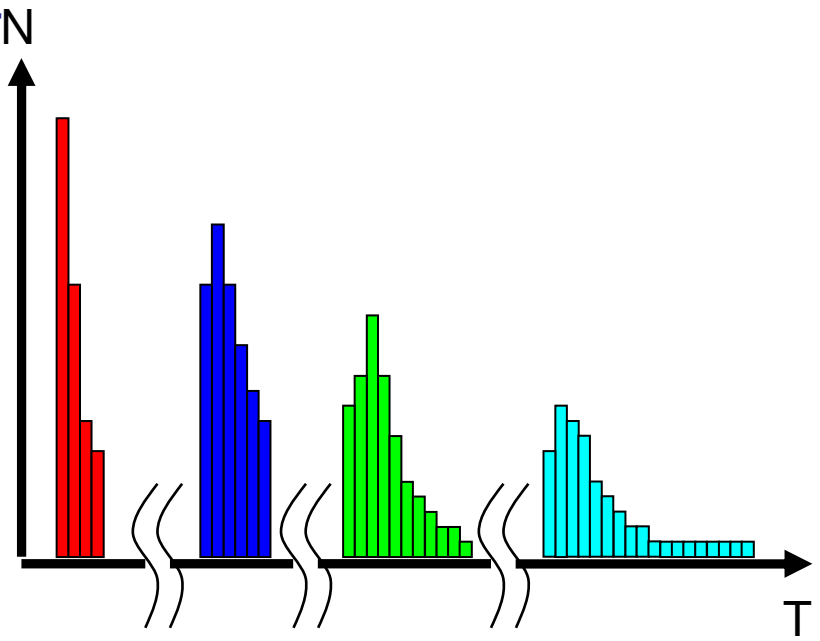
Average delay depends on load



Fully loaded networks crash ! → Plan for <50%

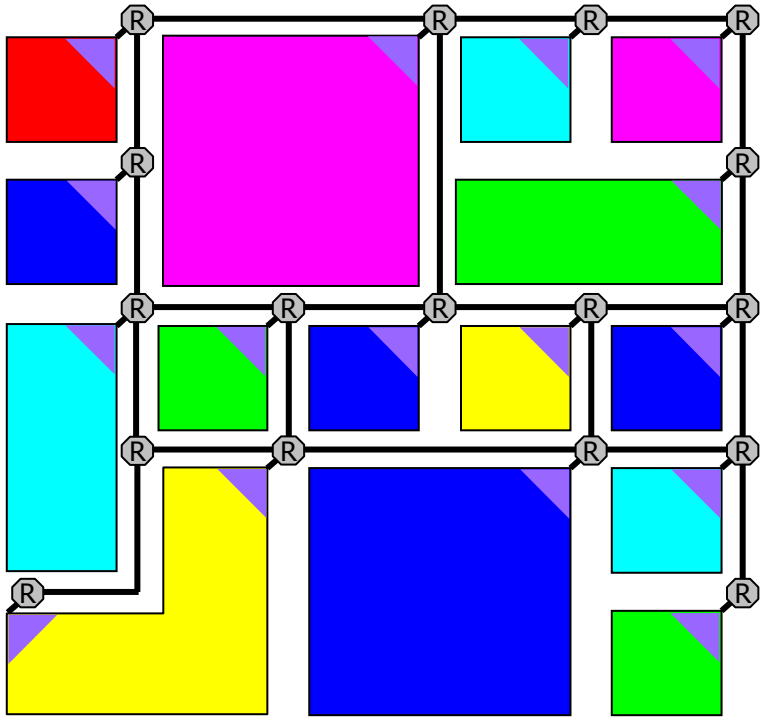
Quality-of-Service in NoC

- Multiple priority (service) levels
  - Define latency / throughput
  - Example:
    - **Signaling**
    - **Real Time Stream**
    - **Read-Write**
    - **DMA Block Transfer**
  - Preemptive
- Best Effort performance
  - E.g. 0.01% arrive later than required



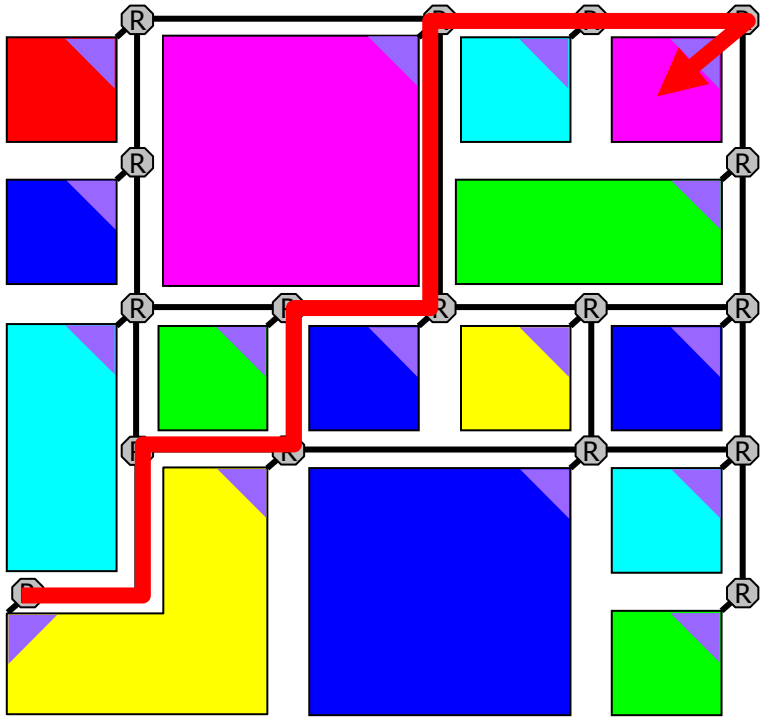
* E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny., **"QNoC: QoS architecture and design process for Network on Chip",** JSA special issue on NOC, 2004.

SoC with NoC

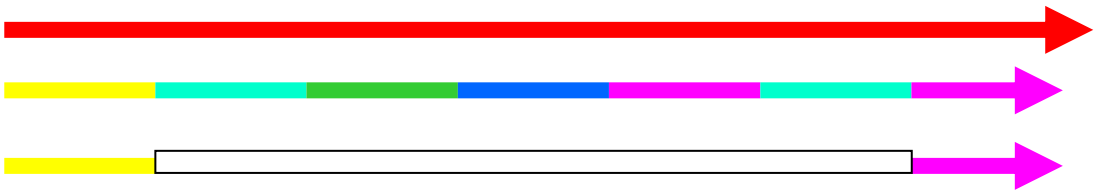


- Each color is a separate clock domain

SoC with NoC



- What clock for the interconnect?
  - Fastest?
  - Opportunistic?
  - None?



The case for Async NoC and hard IP cores

- NOCs are for large SOCs
- Large SOCs = multiple clock domains

→ NOCs in large SOCs should be asynchronous

- Two complementary research areas:
  - Asynchronous routers
    - simplify design, low power
  - Asynchronous interconnect
    - high bandwidth, low power
- Problem: need special CAD, special methodology
  - Solutions:
    - deliver and use as "configurable hard IP core"
    - use only at physical design phase
    - deliver as predesigned infrastructure (FPGA, SOPC)

NoC: Three Levels

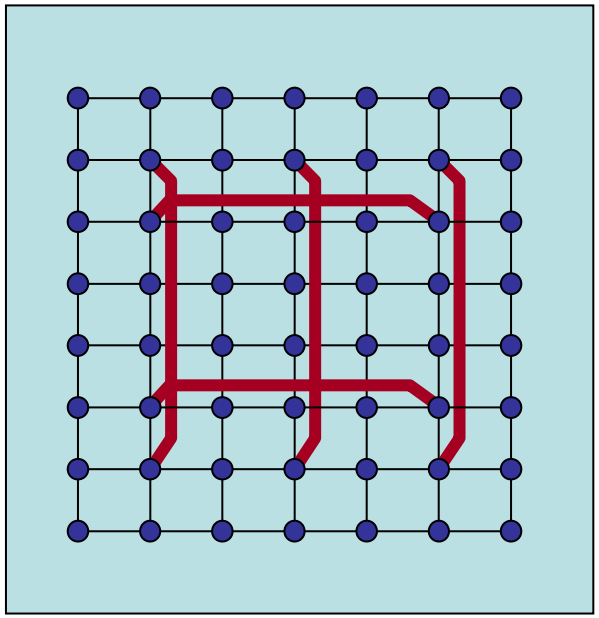
- Circuits
  - Wires, Buffers, Routers, NI
- Network
  - Topology, routing, flow-control
- Architecture
  - Everything is packets
  - Traffic must be characterized
  - NoC can extend to other chips

Circuit Issues

- Power challenge
  - Possible power sorting: Modules > NI > Switching > buffers > wires
  - Network interface (NI)
    - Buffer, request and allocate, convert, synchronize
  - Switches: X-bar or mux, arbitrated or pre-configured
  - Buffers: Enabled SRAM vs. FF
  - Wires: Parallel vs. serial, low voltage, fast wires
- Area challenge (a.k.a. leakage power)
- Latency challenge
- Design challenge
  - These circuits are not in your typical library !
- EDA challenge
  - Flow? Algorithms? NoC compiler?
- Who is the user?
  - Logic design vs. back-end
    - Not fit for simple HDL synthesis. Needs customized circuits

Networking Issues

- Topology: Regular mesh or custom?
  - ASIC are irregular
- Topology: Flat or hierarchical?



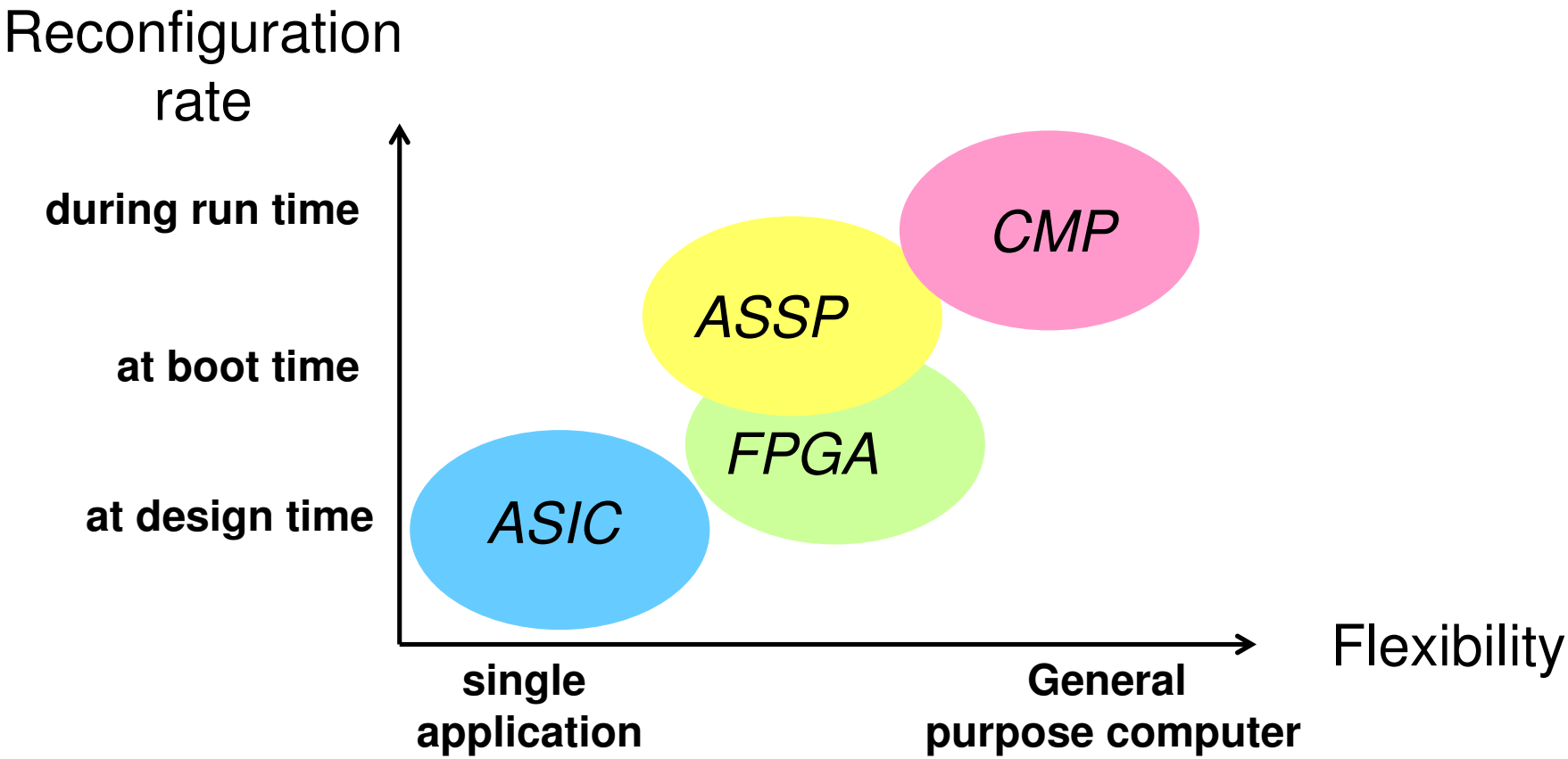
Networking Issues (cont.)

- Topology: Low or high radix?
  - Higher radix nets provide fewer hops (lower dynamic power)
  - But use more wires and more drivers / receivers (higher static power)
- How many buffers?
  - They are expensive (dynamic and static power)

Networking Issues (cont.)

- Guaranteed Service or Best Effort?
  - GS easy to verify performance
  - GS employs no buffers (only muxes): faster, lower power
  - But GS good only for precise traffic patterns
  - Philips (NXP) combined GS and BE
- Routing: Flexible or simple?
  - Flexible routing bypasses faults and congestions
  - Multiple routes may require re-ordering (expensive)
  - Fixed, simple single-path routing saves energy and area
- Multiple priorities and virtual channels
  - Effective but cost buffers

One size does not fits all!



- Even within each class several NOCS may be needed

I. Cidon and K. Goosses, in ***"Networks on Chips"*** , G. De Micheli and L. Benini, Morgan Kaufmann, 2006

NoC for CMP / Many-core chips

- Support known traffic patterns
  - CPUs to Shared Cache
  - Cache to external memory
  - Special I/O traffic: Graphic, wireless / wired comm, ??
- Support unexpected traffic patterns
- Provide new services
  - Provide cache coherency?
  - Manage the shared cache?
  - Schedule tasks / processes / threads?
  - Support OS?
  - Support other memory models ?
    - More distributed ? More tightly coupled ?
  - Manage I/O?
- One NoC may not be enough…

Other Dimensions

- ASIC vs FPGA
  - In FPGA, NoC by vendor or user?
- ASYNC vs SYNC
- One chip vs Multiple chips
  - 3D, multi-chip systems
- HW vs SW
- Fixed vs Reconfigurable SoC/NoC

NoC for Testing SoC

- Certain test methods seek repeatable cycle-accurate patterns on chip I/O pins
- But systems are not cycle-accurate
  - Multiple clock domains, synchronizers, statistical behavior
- NoC facilitate cycle-accurate testing of each component inside the SoC
  - Enabling controllability and observability on module pins
    - Instead of chip pins
- Can be extended to space
  - Decomposed testing and b-scan in mission
  - Useful together with reconfiguration

Some rules were made to be broken..

Beware the Net !

- Adopting just any off-chip net feature to NoC may be a mistake
  - You can create an elegant regular topology
    - But ASICs are often irregular
  - You can create a non-blocking network
    - But hot spots can block networks of infinite capacity
  - You can guarantee service (it's easy to verify)
    - But extremely hard to configure. Best Effort is simpler
  - You can use lots of buffers
    - And dissipate lots of power
  - You can create complex routing
    - Fixed, simple single-path routing saves energy and area
  - You can try to balance traffic
    - Single-path routing works better with links of uneven capacity
  - You can make packets conflict with each other
    - Better use priority levels and pre-emption

Where do the NoC-RT talks fit?

- OLD RULES
  - Fabien Clermidy (LETI), Abbas Sheibanyrad (TIMA)
    - Async NOC supporting reconfigurations and DVFS
  - Geir Åge Noven (Kongsberg), Eberhard Schuler (PACT), Kees Goossens (NXP)
    - TDM circuit-switching NOC supporting guaranteed service
  - Domique Houzet (INP)
    - NoC supporting parallel programming constructs
  - Laurence Pierre (TIMA), Constantin Papadas (ISD)
    - Formal verification and modeling of NOC
  - Souyri+Coldefy+Koebel+Lefftz (Astrium)
    - NoC supporting system integration (HW+SW)
- NEW RULES
  - Axel Jantsch (KTH), Riccardo Locatelli (STM)
    - hw+sw programmable NOC
  - Gerard Rauwerda (Recore Systems)
    - NoC for reconfigurable many-core
  - Bjorn Osterloh (Braunschweig), Steve Parkes (Dundee)
    - SpW-NOC for reliability, reconfiguration
  - Claudia Rusu (TIMA), Martin Radetzki (Stuttgart)
    - Faults and fault tolerance

Summary

- Interesting area!
  - Complex
  - Multi-disciplinary
  - Many open issues, but already useful
  - Many design decisions to take
- Space application will require special types of NoC
  - Faults, reconfiguration, ??

Network on Chip