NoCs: a Short History of Success and a Long Future

Giovanni De Micheli

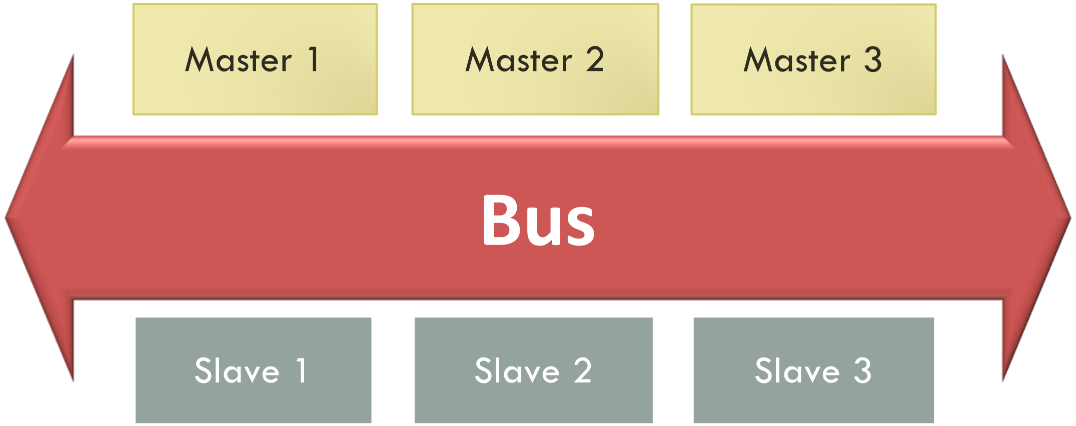
Federico Angiolini

With credits to Charles Janac

A LOOK BACK

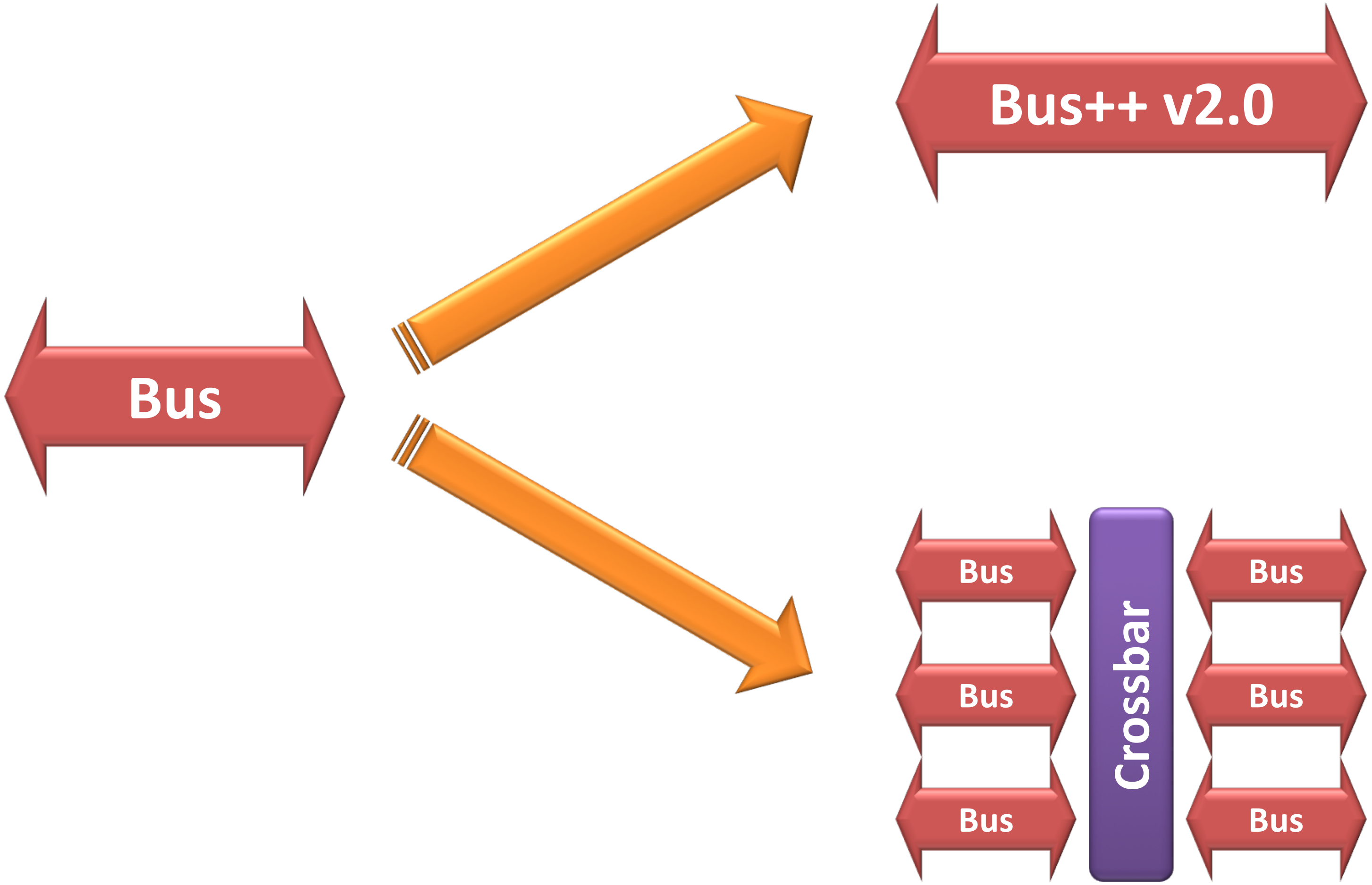
20th Century: The (Mini)bus

But buses run out of gas….



- Not enough parallelism for increasing core counts
- Power: all transactions essentially broadcast
- Zero composability
- Physical issues: timing, routing

Bus Evolution



Protocol evolution

Topology evolution

The birth of the NoC

We explain why the shared bus, which is today's dominant template, will not meet the performance requirements of tomorrow’s systems. We present an alternative interconnection in the form of switching networks.

Guerrier & Greiner, 2000

Using a network to replace global wiring has advantages of structure, performance, and modularity.

Dally & Towles, 2001

We propose borrowing models, techniques, and tools from the network design field and applying them to SoC design.

Benini & De Micheli, 2002

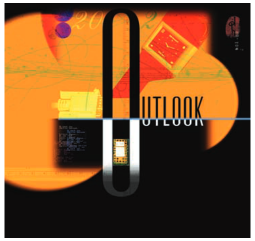
NoCs: A broad literature

SPIN: a Scalable, Packet Switched, On-chip Micro-network

Adrijean Adriahantenaina (UPMC/LIP6)
Hervé Charlery (UPMC/LIP6)

SOC DESIGNS

Networks on Chips: A New SoC Paradigm



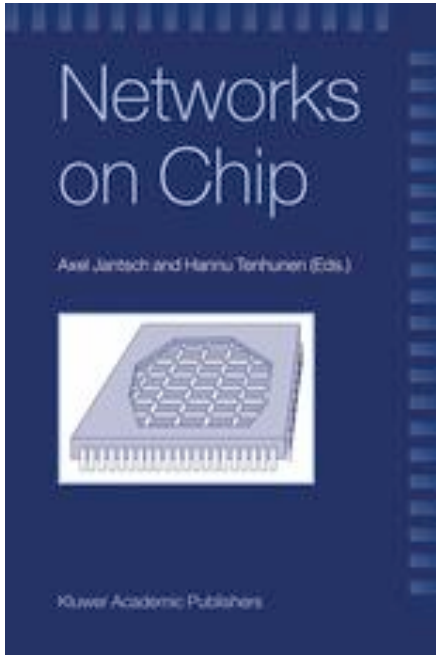
Æthereal Network on Chip: Concepts, Architectures, and Implementations

Kees Goossens, John Dielissen, and Andrei Rădulescu
Philips Research Laboratories

Route Packets, Not Wires: On-Chip Interconnection Networks

William J. Dally and Brian Towles
Computer Systems Laboratory
Stanford University
Stanford, CA 94305

TECHNOLOGY AND TOOLS
NETWORKS ON CHIPS
GIOVANNI DE MICHELI LUCA BENINI


Networks on Chip
Axel Jantsch and Hannu Tenhunen (Eds.)
Kluwer Academic Publishers

The concept in a nutshell

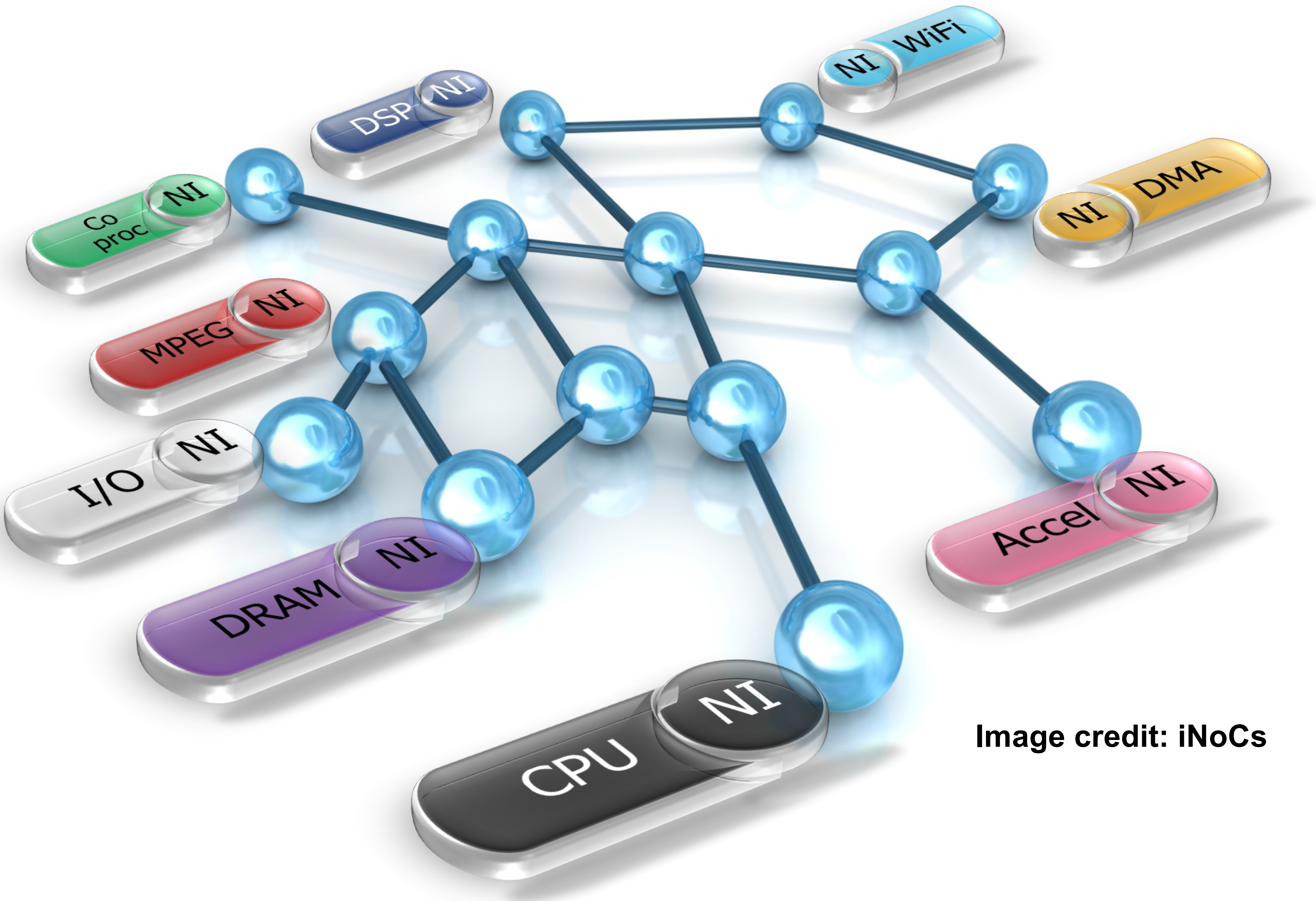
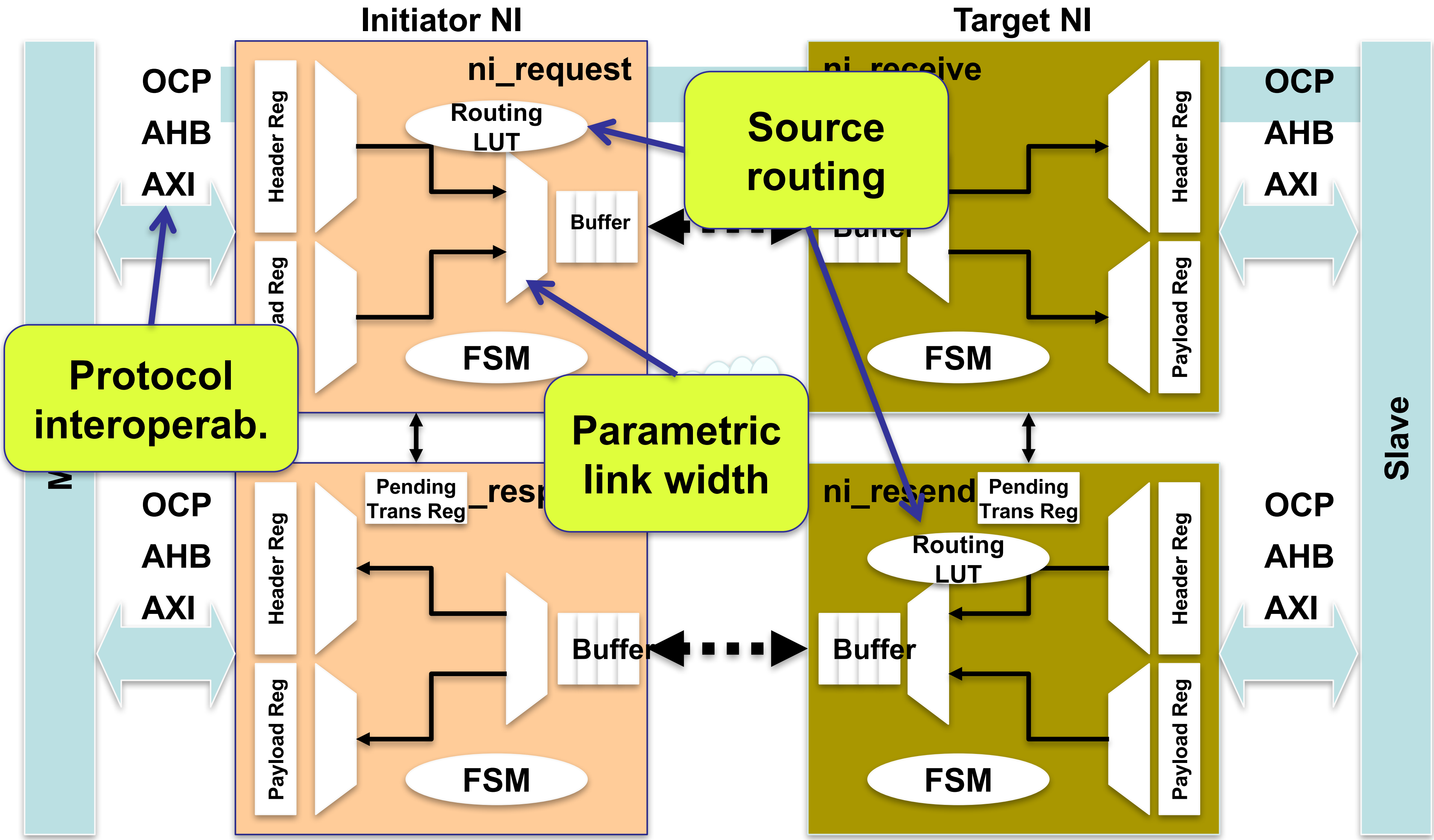


Image credit: iNoCs

The Main Promises

- *Scalability*
  - Hundreds/thousands of connected cores
- Tunable *Power/Frequency/Area*
  - With packetization has happened, links can be tuned locally:
    - wide or narrow, fast or slow, ..
- Easier *design closure*
  - Fewer, shorter, point-to-point wires
  - Decentralized nature suited to multiple clock/power domains

NoC synthesis: xPIPES



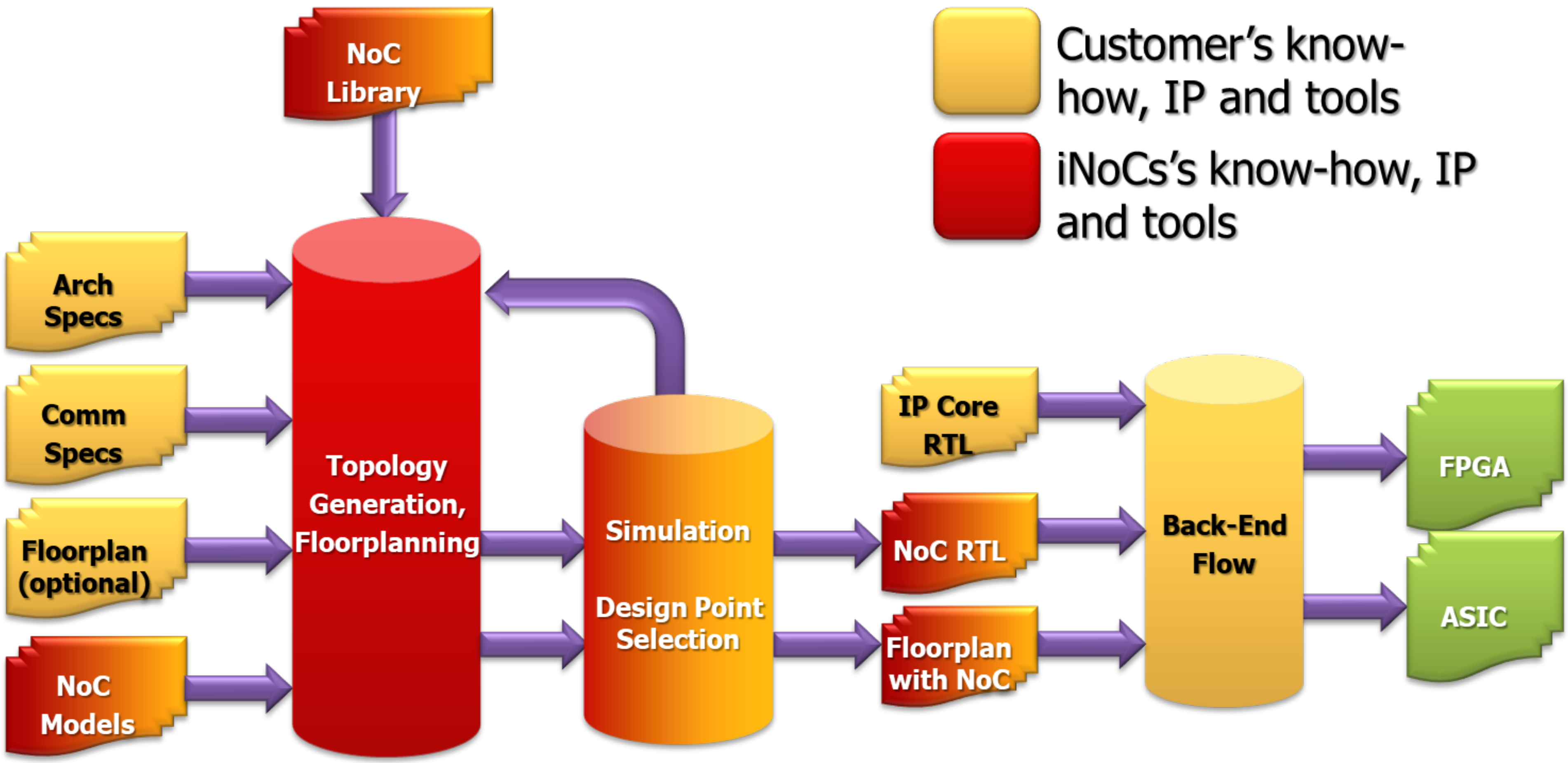
Source routing

Protocol interoperab.

Parametric link width

[Bertozzi et al. 2005]

Layout-aware NoC Synthesis



🟨 Customer's know-how, IP and tools

🟥 iNoCs's know-how, IP and tools

Credit: iNoCs, 2009

[Angiolini, Murali et al.]

Quick, Broad Academic Adoption

- Within few years, hundreds of papers every year
- Favourite subjects:
  - Topology
  - Architecture, esp. switches, buffering, …
  - Routing algorithms and implementations
  - Simulation
  - Physical implementation (signaling, asynchronous, …)
  - Fault tolerance
  - QoS, mapping
  - Design tools (EDA)

Interesting Research Trends

- Some research were strongly inspired by WANs and supercomputers, e.g.:
  - Virtual channels
  - Deeply pipelined switches
  - Hypercube topologies
  - Store-and-forward switching
  - Virtual output queuing
  - Dynamic routing

- How did this go?

Learnings Set in Quickly

- A NoC is **not** the same as a wide-area network

- In most cases, opposite tradeoffs:
  - WANs need to minimize cable count, but cable length is unlimited. Router area and power is secondary. Software on top implements much of the stack. Accepted latency: milliseconds
  - NoC wires are comparatively inexpensive but they must be short. Area and power are severely limited. Must work also without any software. Accepted latency: sometimes <1 ns!

- Led to quick adjustments and an opposite current innovation (for some types of designs)
  - Low power NoCs, bufferless routing, combinational NoCs

So What Happened in the Real World?

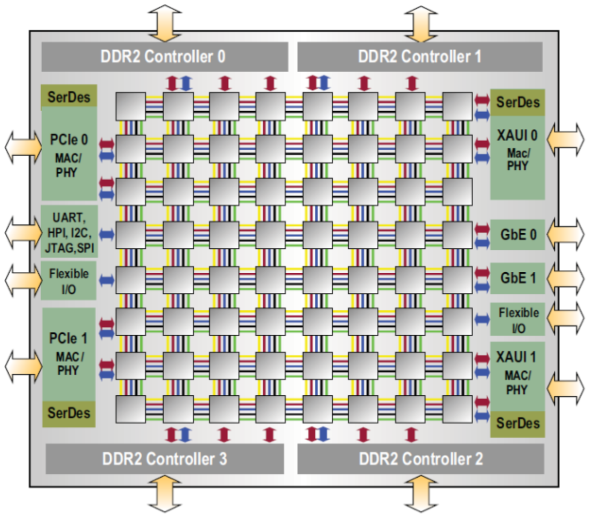
"Academia invents complex solutions to problems, and evaluates them in a simplified context.

"Industry tries to find the simplest solutions because the context is already complex."

José Duato

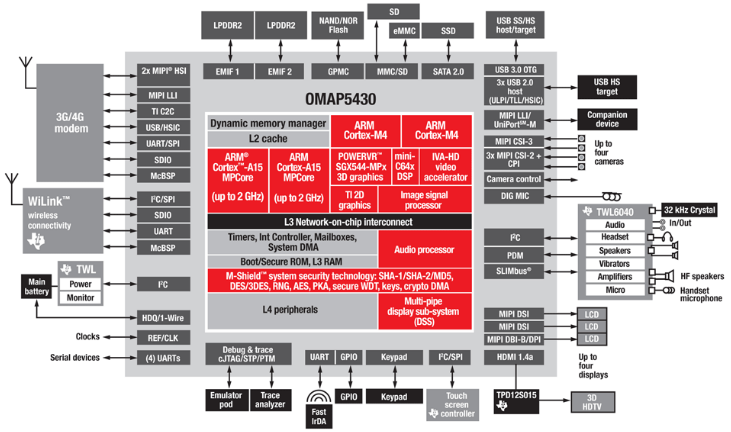
Initial Industrial Adoption

- A few designs based on the “mesh” approach for CMP and high-end computing
  - Notably, Tilera TILE64 (~2007) (64 cores, 11 W), Intel SCC (~2009) (48 x86 cores, 125 W)



- A huge number of designs based on heterogeneous MPSoCs, often low-power
  - E.g. ST (2006), TI (2008), NEC, Samsung, LG, MobilEye, Toshiba, Qualcomm…

TI OMAP5430 SoC



Who Designs These NoCs?

- Specialized vendors: Arteris, Sonics, NetSpeed (acquired by Intel)
- ARM
- Academic spinoffs: e.g. Silistix, iNoCs (IP acquired by Arteris)
- In-house teams are still the lion’s share of the market
  - Interconnect and related services are seen as key differentiator
  - Crucial for functionality, design time, performance, power
  - Specialized designs have sometimes fundamentally different and unique traffic patterns (e.g. GPUs, network processors, FPGAs…)
  - With ever-increasing complexity, this may shift

NOCS TODAY

NoC is Data Highway of the SoC

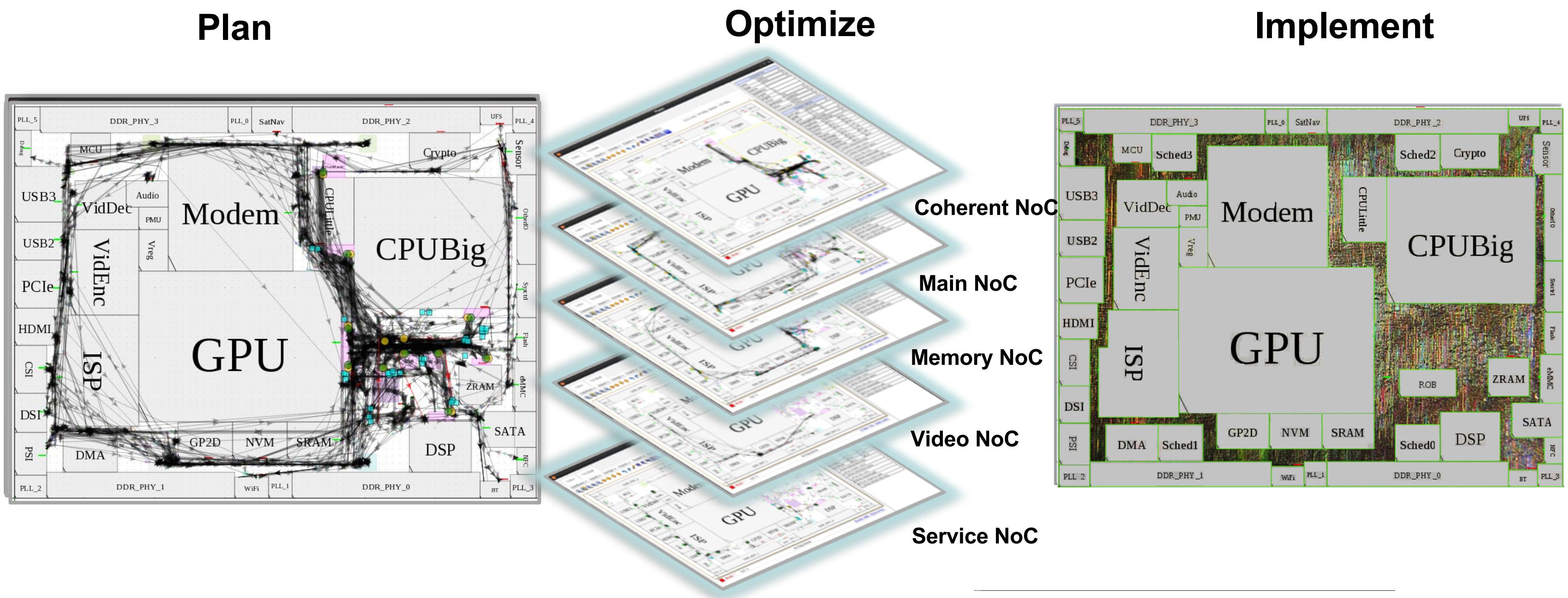
- Only IP that traverses the chip
- Changes between projects
- If it does not function properly, the SoC does not work
- Contains the longest SoC wires
- Carries most of the interesting data
- Helps define the SoC architecture
  - Must support SoC performance requirements
- Often the last IP to be frozen, has to fit into available channel space
  - When timing closure becomes schedule-critical!
- Changes multiple time in response to Architecture, Marketing ECOs

Slide courtesy ARTERIS IP

A NoC is Many NoCs

- At the very minimum, most implementations separate request and response networks (avoid deadlocks)
- NoCs may further separate message types
  - E.g. for cache coherence (see later); TILE64 has 5 networks…
- NoCs are likely to be partitioned into “subsystem NoCs”
  - either co-designed or reused
- Plus additional “interwoven” NoCs for:
  - Configuration of main NoCs
  - Performance monitoring/statistics
  - Debug
  - …

NoC Interconnect Technology Enables Better SoCs - Faster



Plan

Optimize

Implement

Coherent NoC

Main NoC

Memory NoC

Video NoC

Service NoC

- NoC technology allows isolation of individual fabrics so they can be managed quickly and easily
- Capturing both logical interconnect topologies and physical constraints
- Enabling rapid delivery of SoC interconnect for architectural, logical and layout success

Slide courtesy ARTERIS IP

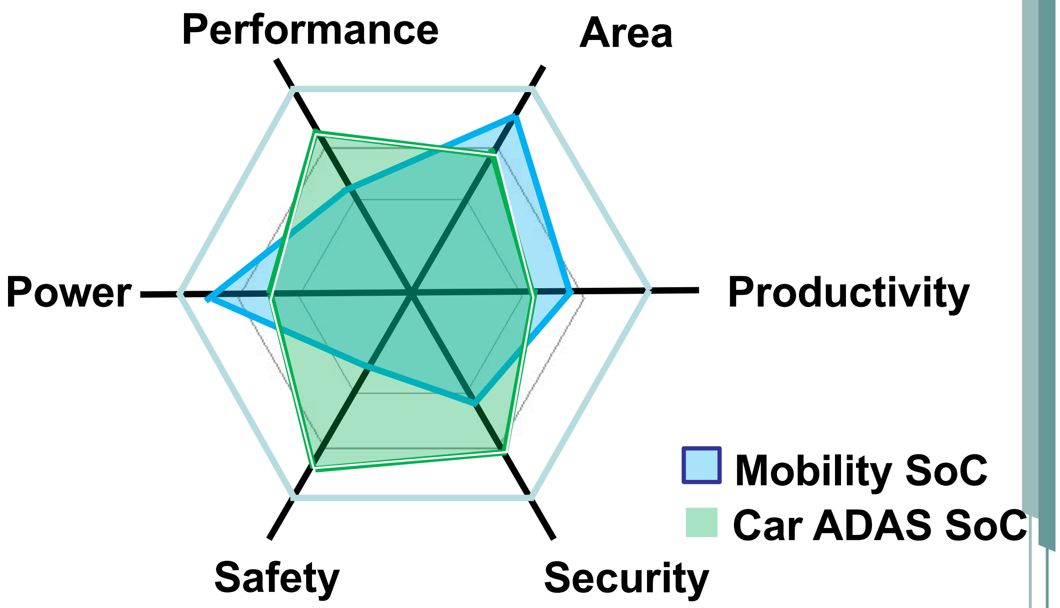
Quo vadis: two main drivers in SoC design

- Mobile communication

- Autonomous vehicles

NoCs Cover Design Space of SoC Requirements

- **Architecture Flexibility:** Unlimited topologies, Support for standard protocols & heterogeneous coherency, multiple caching levels to reduce off-chip accesses
- **Performance:** 166Mhz-2Ghz frequency @16nm, >1TBit/sec bandwidth w/512 bit links
- **Power:** <0.5mW idle power/1M gates@16nm, one cycle power domain wake up, 3-level clock gating, etc.
- **Area:** Endpoint NoC = Lower area/Interconnect function (vs hybrid buses or corner router NoCs)
- **Productivity:** Design exploration, multi-level modelling, auto test bench generation, physical awareness, design flexibility, derivative SoC NoCs can be built in 3 days
- **Safety:** Resilience – ISO26262 ASIL B-D capable, Functionally safe domains
- **Security:** Customer extensible firewalls and access controls



Quality Assumed and Vitally Important!!

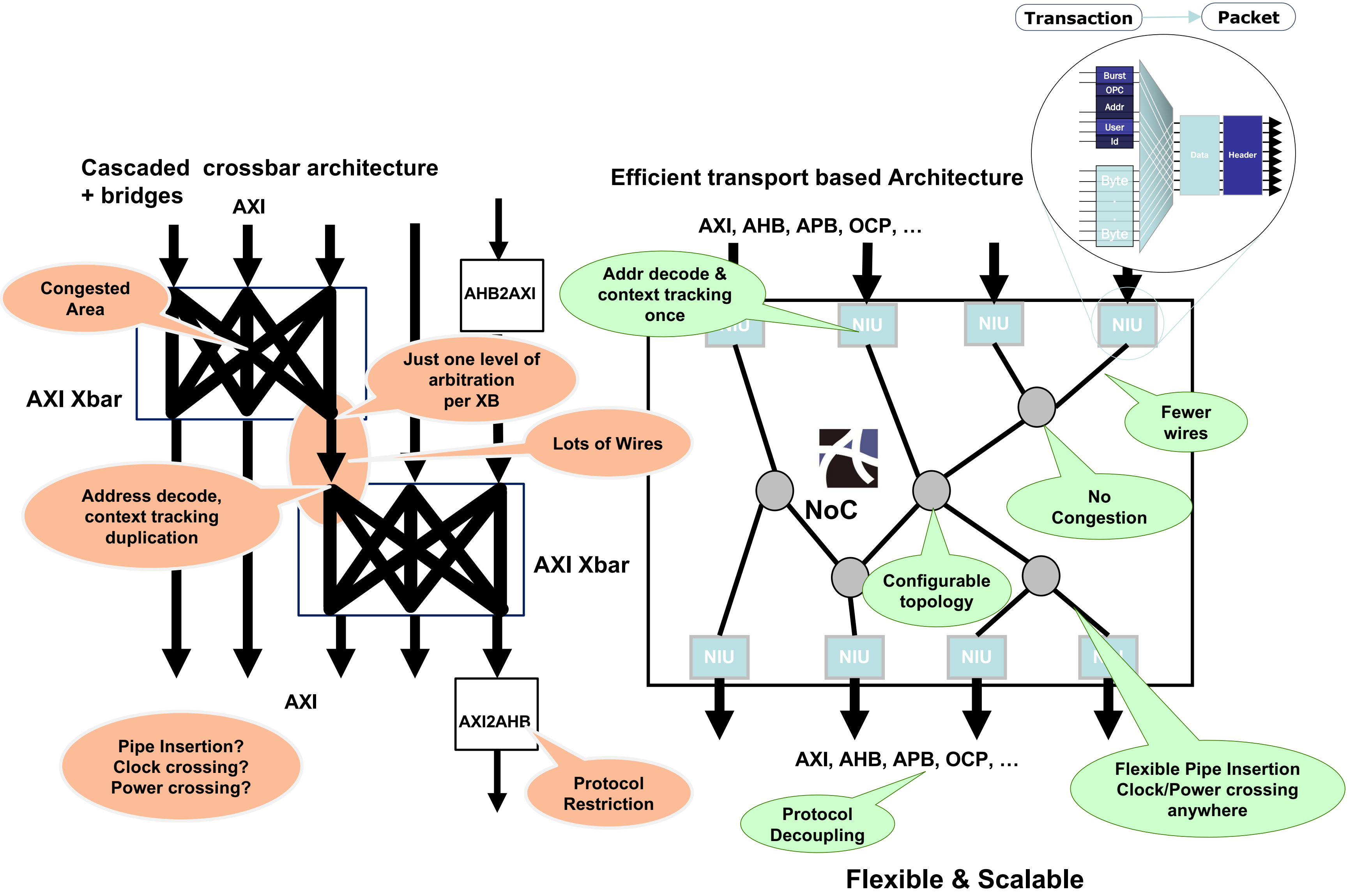
Slide courtesy ARTERIS IP

Mobility – Original Killer App for NoC Technology

- Application Processors, Modems
- Many initiators to many targets
- Required NoC due to needs for:
  - Low power for battery life
  - Efficient area for cost
  - Performance for response time
  - Productivity due to short SoC cycles
  - Multiple power domain flexibility
- All at the same time!

Slide courtesy ARTERIS IP

NoCs Replaced Cascaded Crossbars


Cascaded crossbar architecture + bridges
AXI
Congested Area
AHB2AXI
AXI Xbar
Just one level of arbitration per XB
Lots of Wires
Address decode, context tracking duplication
AXI Xbar
AXI
AXI2AHB
Pipe Insertion? Clock crossing? Power crossing?
Protocol Restriction
Transaction
Packet
Burst
OPC
Addr
User
Id
Byte
Byte
Data
Header
Efficient transport based Architecture
AXI, AHB, APB, OCP, …
Addr decode & context tracking once
NIU
NIU
NIU
NIU
Fewer wires
NoC
No Congestion
Configurable topology
NIU
NIU
NIU
NIU
AXI, AHB, APB, OCP, …
Flexible Pipe Insertion Clock/Power crossing anywhere
Protocol Decoupling
Flexible & Scalable

Slide courtesy ARTERIS IP

Automotive: the new driver for NoCs

- Electrification + ECU Consolidation + Automated Driving = Disruption

Vision Camera	(4-16) (6)
ADAS / Machine Learning	(1-4) (2)
Dashboard / HUD	(2)
Infotainment	(1)
Radar and/or LIDAR	(4-6) (6)
Chassis Control	(4)
Engine Control	(2)
V2X / V2I / WAN Modem	(2-3) (3)

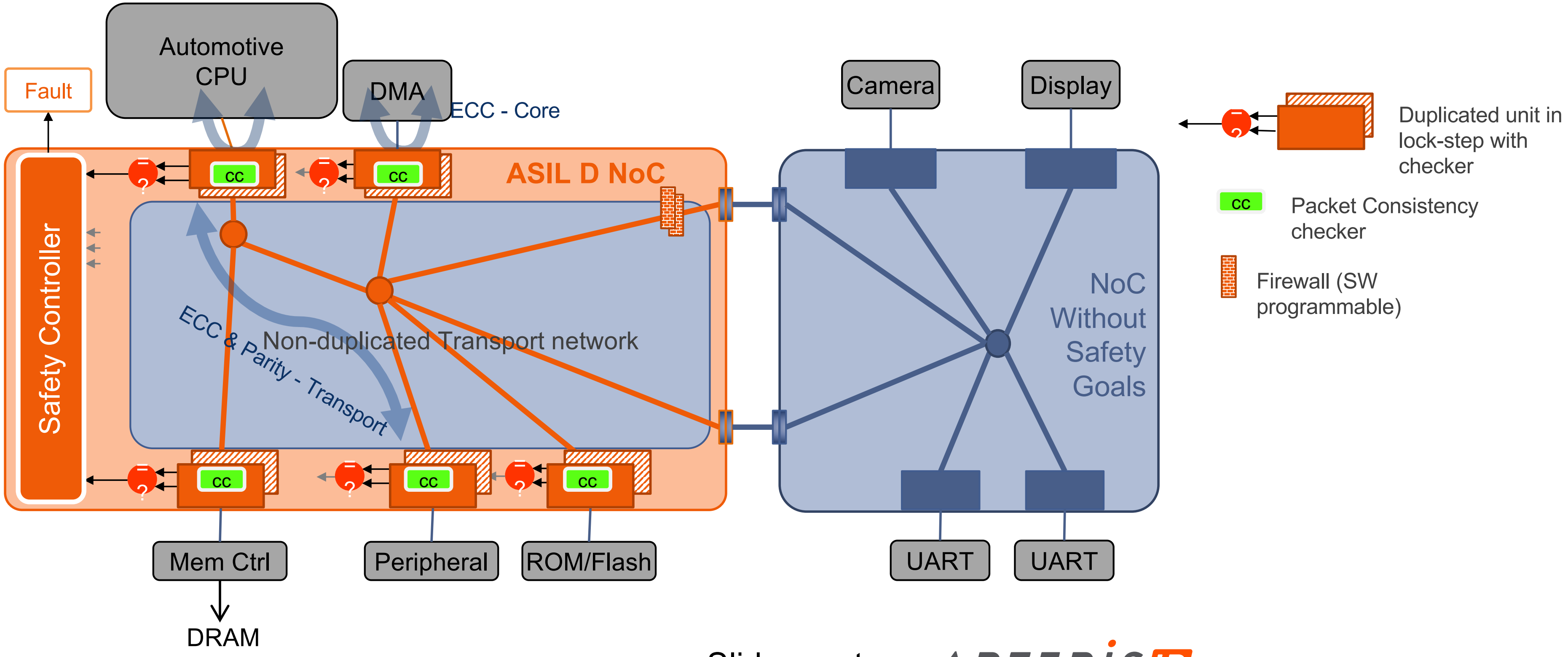
- Total SoCs/car = 24 (avg.)
- 60M electronically enabled cars by 2025
- → ~1.4B SoCs per year (plus electronic infrastructure SoCs)

Slide courtesy ARTERIS IP

End to End Resilience for ISO26262 Compliance

- Unit duplication - fault detection
- ECC at interface & in-transport
- Packet Consistency checkers

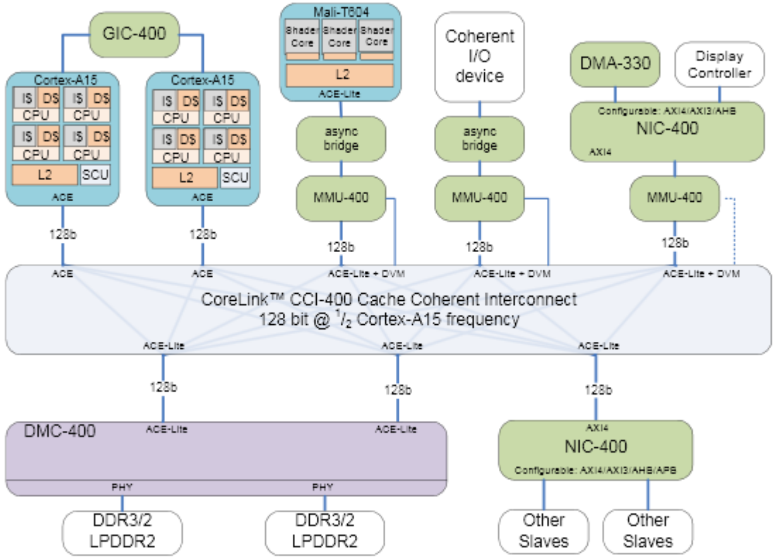
- Safety Controller
- Fault reporting logic BIST
- Multi ASIL Level Support
- ARM Cortex® R5/R7 support



Slide courtesy ARTERIS IP

Cache Coherence

- Many SoC designs using some form of cache coherence
  - ARM ACE, CHI sockets
- Synch. among processors;
  - also for CPU/GPU heterogeneous computing



- Often coherent & non-coherent islands

Credit: ARM

- Adds significant requirements to the NoC
  - In-NoC directory services and related management
  - Multiple networks support the cache coherence messages with sufficient performance and without deadlocks

Physical Design

- Increasingly critical in recent nodes
- Well-known issue: global wire delay is worsening
  - NoC has to feature configurable pipelining
  - Design QoR/PPA depends significantly on input floorplan
- Wire routing also problematic
  - Wire-intensive, but must fit in remaining channel space
  - Useful to narrow links as much as possible, but only where latency is expendable
- Interplay with domain partitioning and power management
  - NoCs must be partitioned in domains, individually power-managed (status and control signals to flush the packets in flight)

Quality of Service

- Expectations of fine-grained control
- Especially CPU ↔ memory traffic should have “zero latency”
  - Extremely complex combinations of high-bandwidth, low-latency requirements
  - Combined with complex memory mappings
    - interleaving, address spaces, …
- Requirement for multiple HW tuning knobs
  - Buffers, arbitration, priorities, …
- QoS encompasses notions like “security” (from attackers) and “safety” (from faults)

UPCOMING CHALLENGES

Automotive: Super Computer on Wheels

- Level 4 ADAS will require 30-40 Tops of processing
  - Functional safety versus performance
  - Power consumption versus performance
  - Security versus performance
- Sensor fusion (Cameras, Lidars, Radars, Ultrasonics) need
  1-2 Terabits/second bandwidth
  - Need highly sophisticated yet functionally safe NoC
- Near real time latency
  - On-chip cache hierarchy to minimize off-chip DRAM accesses
- Architectural flexibility
  - CPU subsystem, vision subsystem, deep learning, power management, cache coherency islands and high bandwidth memory on a single SoC
  - Interconnect needs to support both 2D implementations and 3D approaches using chiplets
- Power Management – car have only 300W budget for ~80 chips,
  less than 12 watts for air cooling

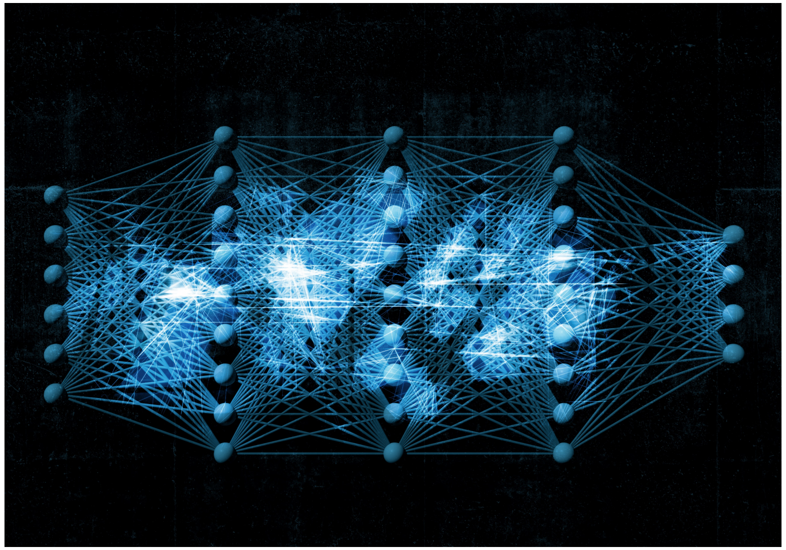
Slide courtesy ARTERIS IP

IoT: Working in Milliwatts

- The IoT market is still evolving and killer applications are still emerging
- We already know that IoT applications will require fast computing, but in a challenging power and area budget
  - Sensor networks, wearables, embedded
- IoT applications will soon need NoCs on a massive scale
  - Power management?
  - Extreme resource conservation?

Emerging Artificial Intelligence Applications

- New algorithms & large data sets
  → hardware architecture evolution

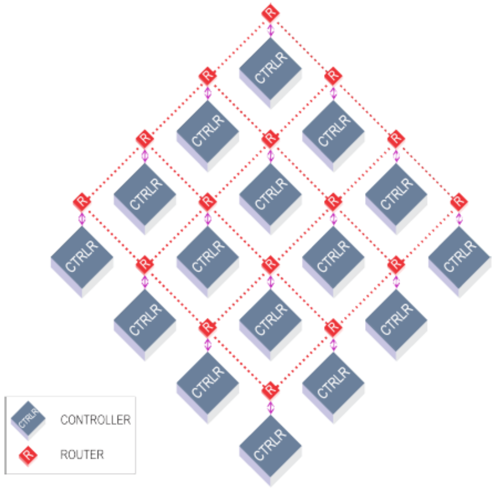


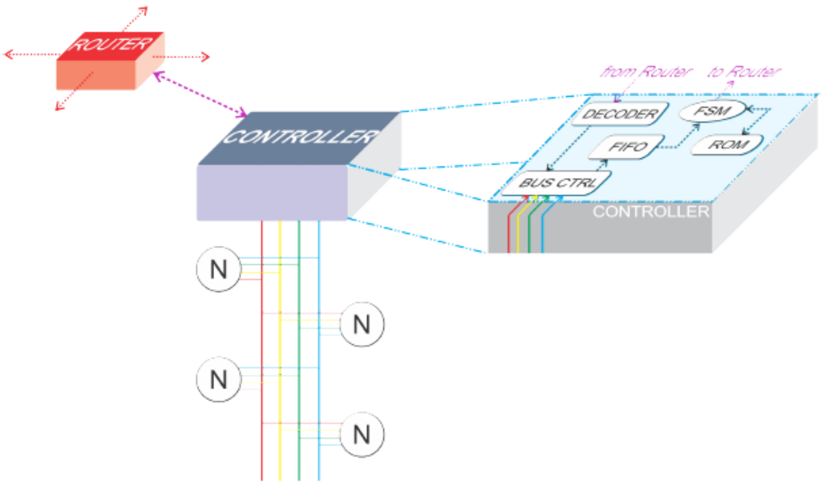
- How you move the data determines
  - Performance
  - Power
  - Scalability

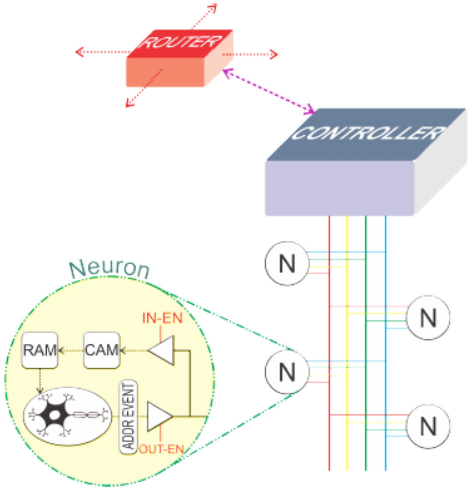
- Regular arrays, suited for e.g. meshes
  - Switches + buffers → “corner routers”
  - Assisted mesh generation, editing
  - Dedicated optimizations (routing, buffering)?
  - Broadcast for updating of CNN weights

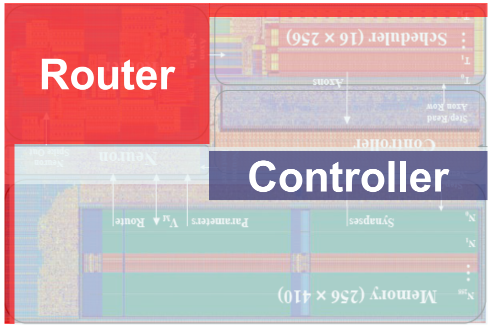
Slide courtesy ARTERIS IP

AI/Mesh Background: Array of Processors





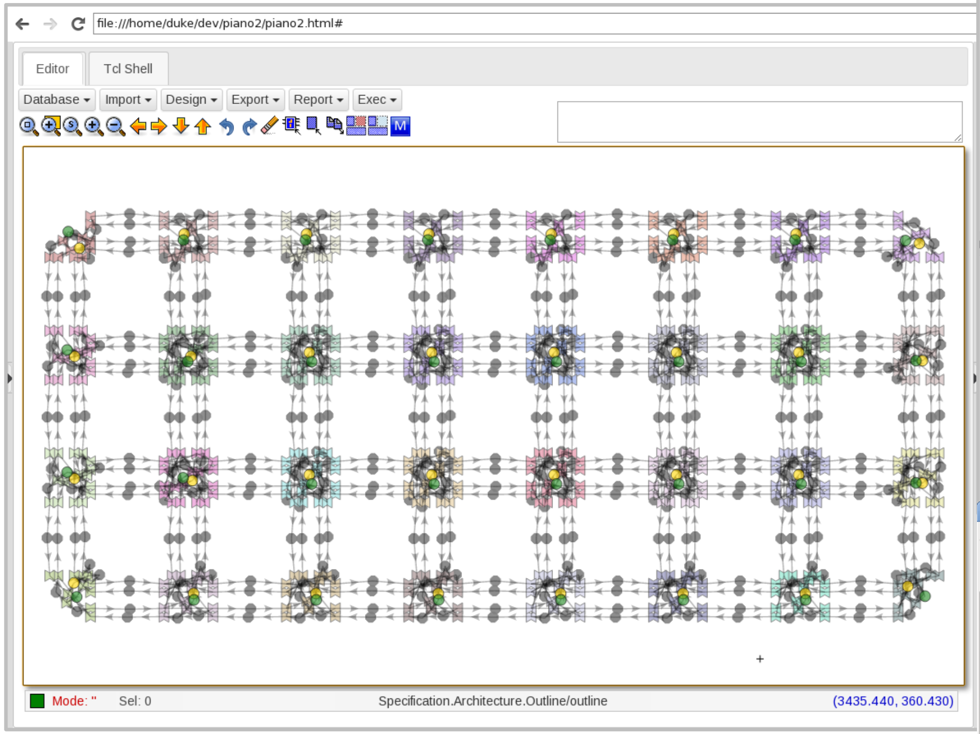




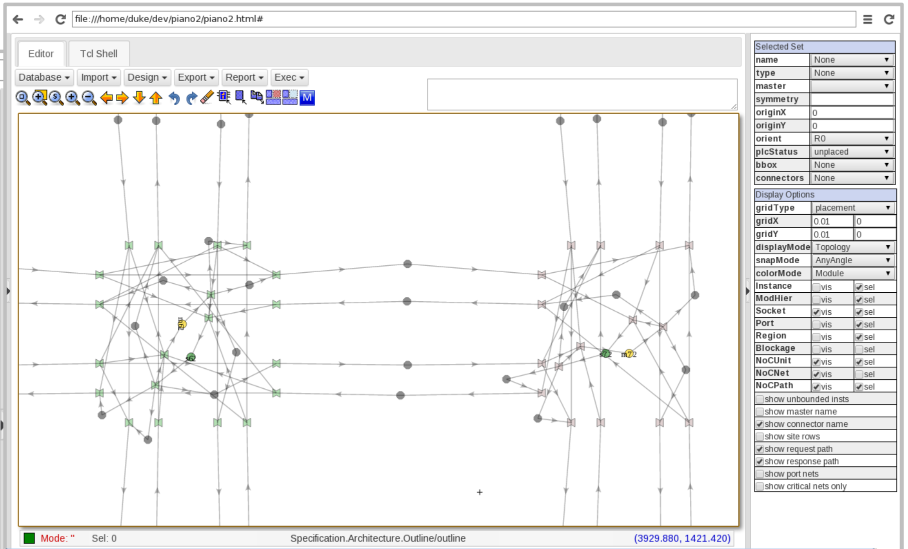
- Each node has a router and controller
  - Controller provides access to one or more processing units
  - Controller provide interfaces (sockets) to the corner router
  - Processing units give access to Controller/Router
- Note: Router may include integrated controller

Slide courtesy ARTERIS IP

AI Mesh Generation



Mode: '' Sel: 0 Specification.Architecture.Outline/outline (3435.440, 360.430)

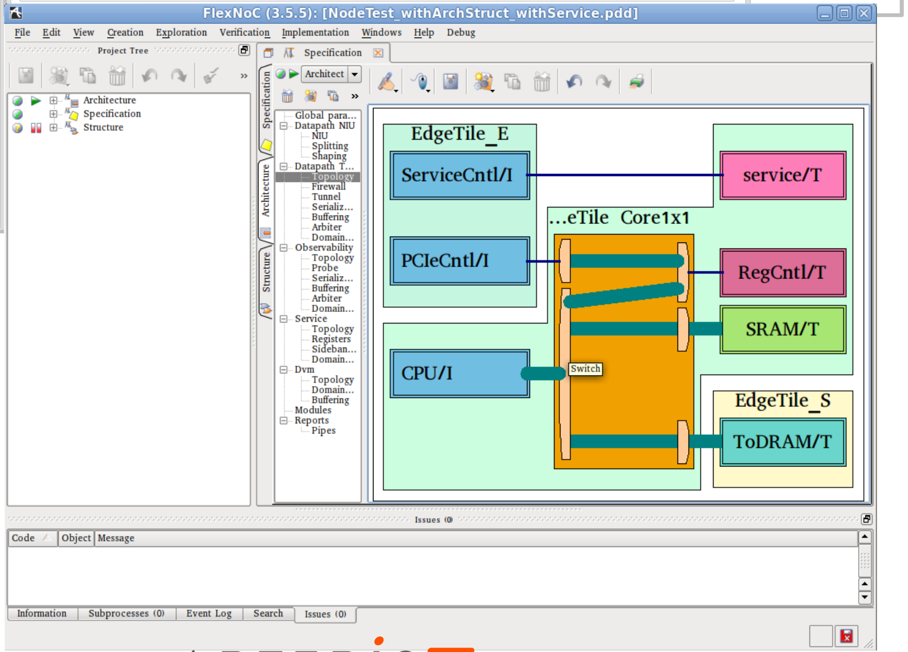


Selected Set	
name	None
type	None
master	
symmetry	
originX	0
originY	0
orient	R0
plcStatus	unplaced
bbox	None
connectors	None

Display Options		
gridType	placement	
gridX	0.01	0
gridY	0.01	0
displayMode	Topology	
snapMode	AnyAngle	
colorMode	Module	
Instance	□ vis	☑ sel
ModHier	□ vis	☑ sel
Socket	☑ vis	☑ sel
Port	□ vis	☑ sel
Region	□ vis	☑ sel
Blockage	□ vis	□ sel
NoCUnit	☑ vis	☑ sel
NoCNet	☑ vis	□ sel
NoCPath	☑ vis	☑ sel

□ show unbounded insts
□ show master name
☑ show connector name
□ show site rows
☑ show request path
☑ show response path
□ show port nets
□ show critical nets only

Mode: '' Sel: 0 Specification.Architecture.Outline/outline (3929.880, 1421.420)


FlexNoC (3.5.5): [NodeTest_withArchStruct_withService.pdd]

Slide courtesy ARTERIS IP

AI Challenges (Hard Stuff) – Predictability, Safety & Reliability

- How do you verify a deep learning system?
- How do you debug the Neural Network black box?
- What are the ethics and biases of these systems?
- What does it mean to make a Neural Network “safe”?
- No one wants a stale AI algorithm → hardware evolution

Slide courtesy ARTERIS IP

ONE LAST THING…
…WE STILL HAVE TO GET RIGHT

What About NoC Design Software

- NoC design flows are extremely complex
  - Rich specifications (sockets, connectivity, address maps, …)
  - Design entry
  - Schematic editing
  - Floorplan viewing and tuning (placement, pipelining)
  - Reporting
  - Iterations and ECOs
  - Interaction with simulators, back-annotation from placement/routing, etc.
  - Generation of RTL and many collaterals
    - Scripts, IP-XACT, documentation, verification…

- …how much is design automation supporting NoC design?

Distilling Lessons to Do Better

1. The market was not ready ten years ago (but may be now)
   - Automation is distrusted until necessary, i.e. on really large designs
2. The problem is very, very complex
   - Large set of requirements that are often conflicting:
     latency, bandwidth, power, area, wirelength, frequency, traffic priorities, resilience, cache coherence, deadlock freedom…
3. Solution domains are extremely different
   - NetSpeed offers a heavily assisted flow to generate mesh-like NoCs, but few types of chips look like meshes.
     Domain-specific tools and algorithms needed
4. Push-button automation is the wrong goal
   - Real-world flows are iterative and often demand last-minute ECOs.
     Automation must be piecemeal and users must be able to override it at any stage

CONCLUSIONS

Outlook

- NoCs have been with us for almost 20 years
- An unmitigated success in the academia and, with few years of delay, also in the industry
- Different research avenues all turned out relevant
  - New promising avenues are optical and wireless NoCs
- We are not done yet:
  - Chip complexity keeps scaling up and new challenges (e.g. resilience, coherence) become prominent
  - NoCs must cover extremely-high-performance as well as extremely-low-power designs, seamlessly
- We must remove EDA bottlenecks to enable the next generations of designs

Thank you