



SHAPING THE NEXT GENERATION OF ELECTRONICS

**JUNE 23-27, 2024**

MOSCONE WEST CENTER  
SAN FRANCISCO, CA, USA



**JUNE 23-27, 2024**

MOSCONE WEST CENTER  
SAN FRANCISCO, CA, USA

# Advancing Multi-Core Systems: Networks-on-Chip Evolution

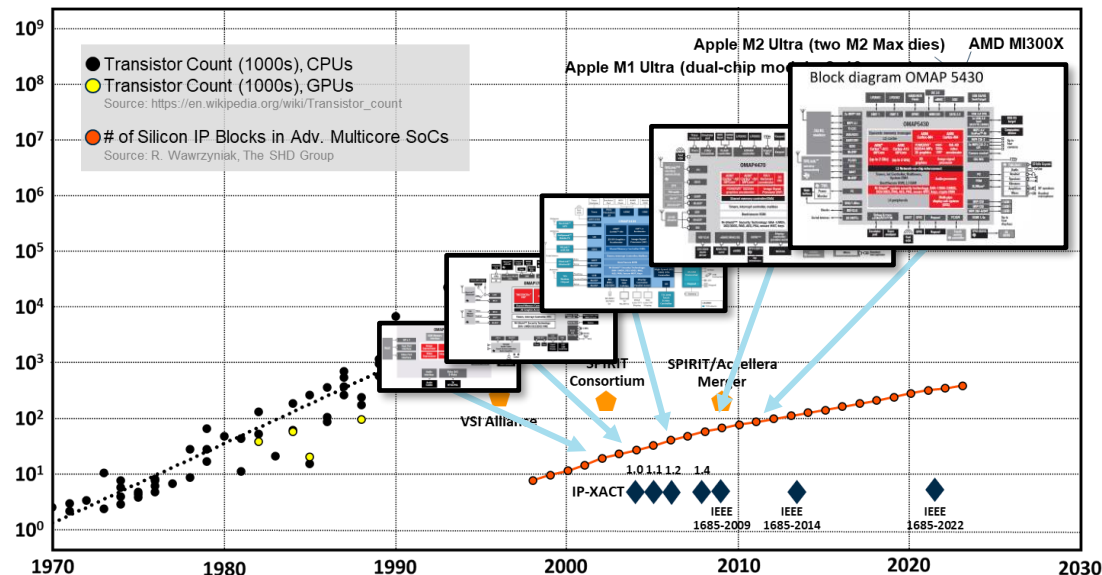
Guillaume Boillet

[guillaume.boillet@arteris.com](mailto:guillaume.boillet@arteris.com)



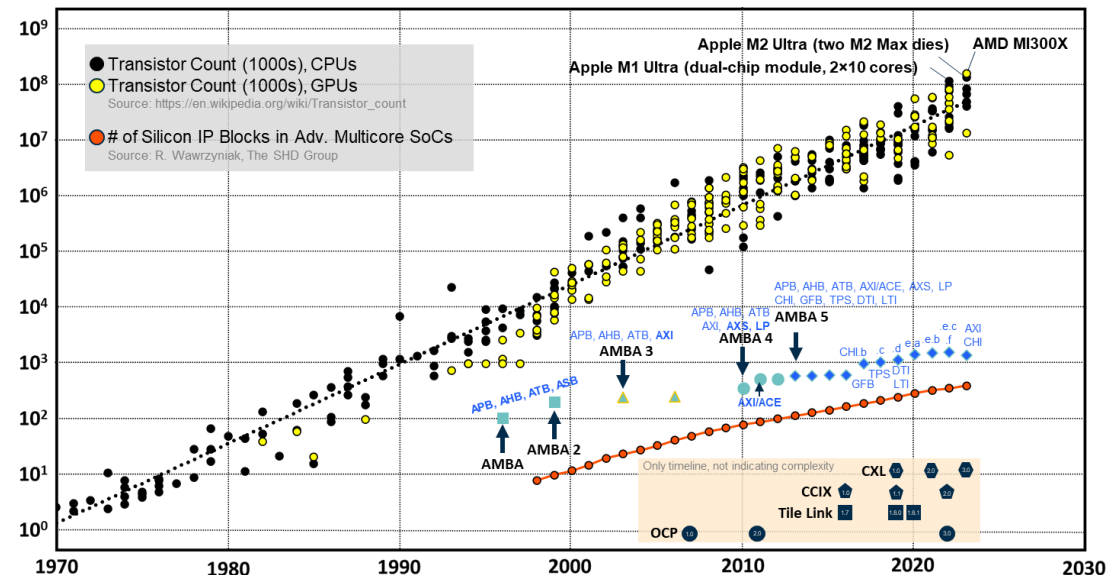
# Semiconductor Complexity Continues to Grow Exponentially

## Growing SoC Integration Complexity



- **# of IP Blocks in SoC has grown from 10s to 100s**
- **Disaggregation offers design version variety**
- **Standards like IP-XACT need to extend too**

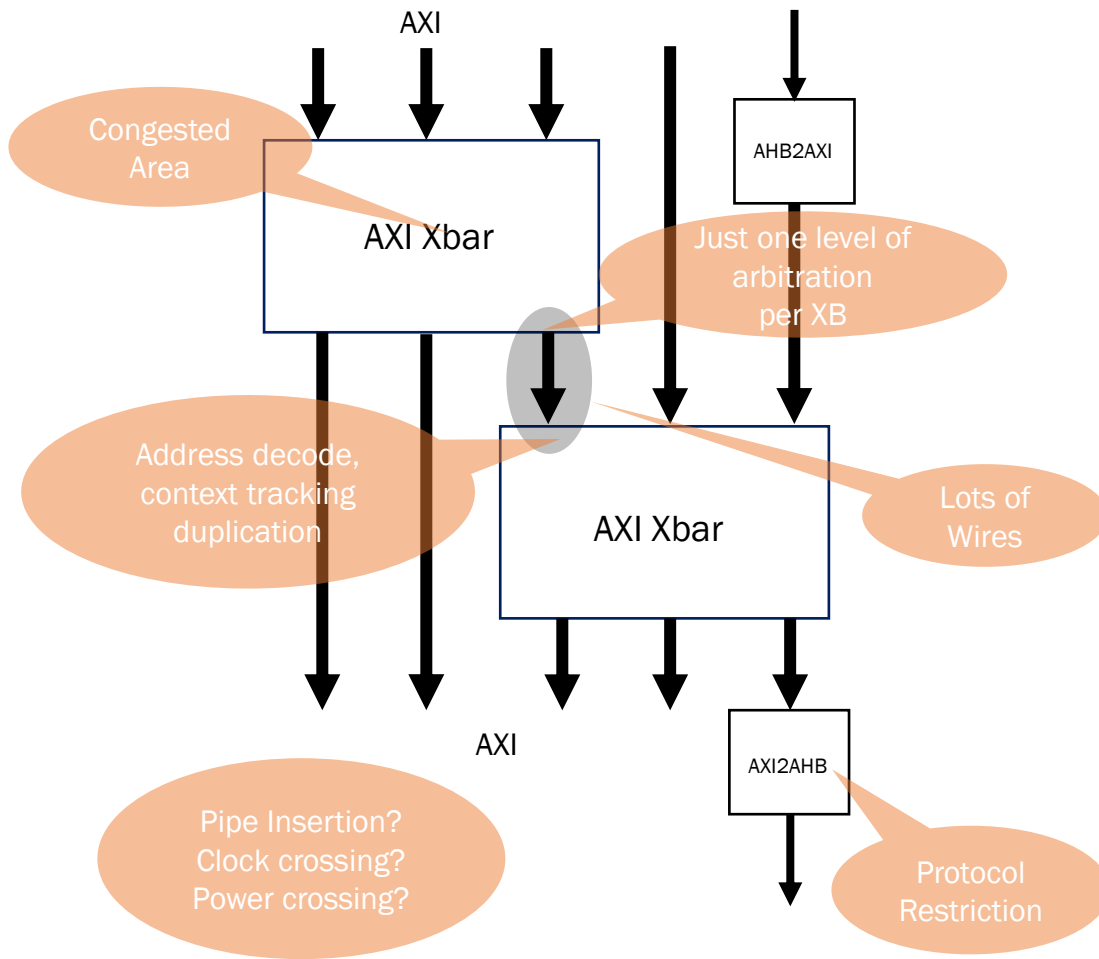
## Growing Network-on-Chip Complexity



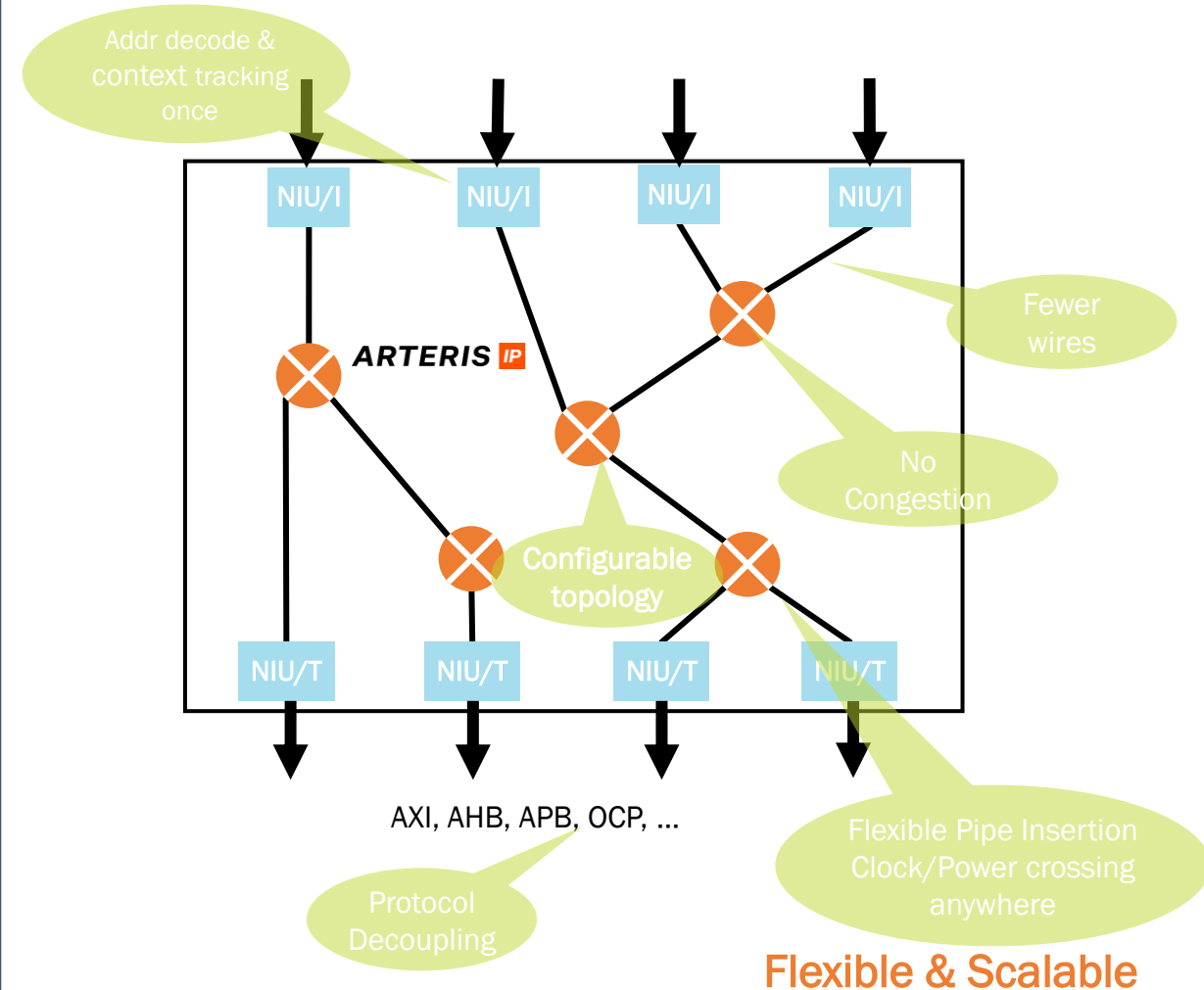
- **Complexity of NoC protocols have grown 10x (# of pages)**
- **Variety of NoC protocols has grown**
- **NoCs evolve into Super-NoCs when split across chiplets**

# Nocs Are Better Than Cascaded Crossbars

Cascaded crossbar architecture + bridges



NoC transport-based architecture

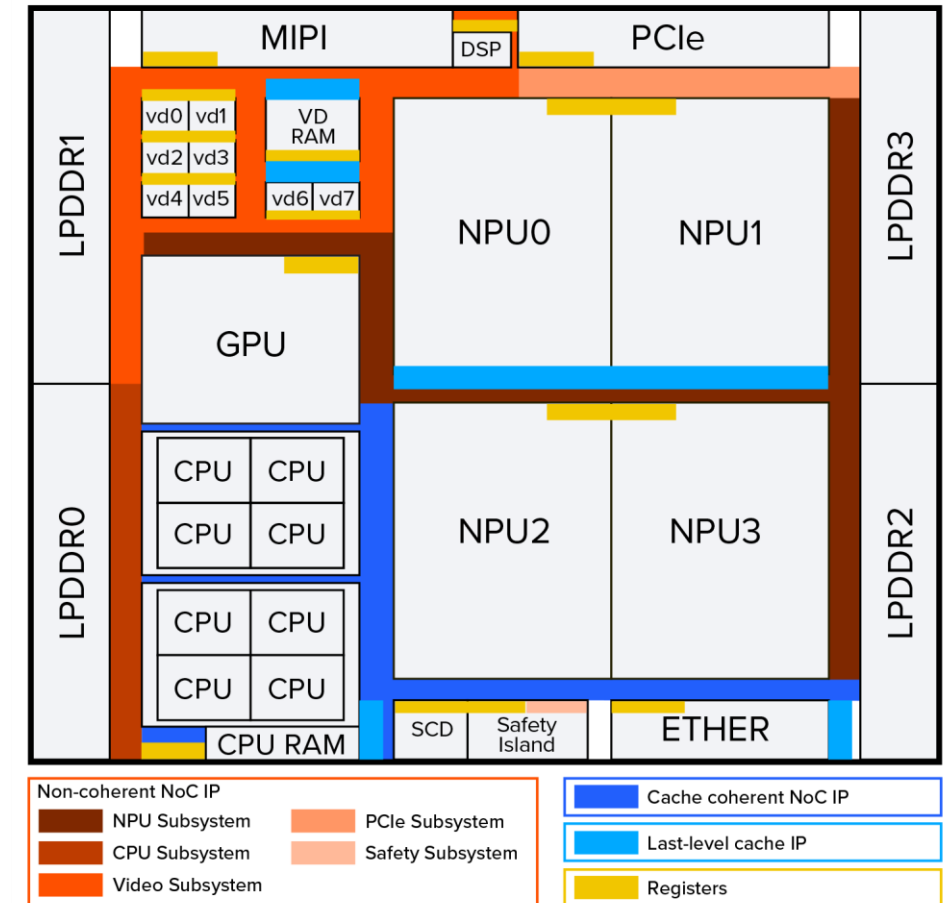


# Bigger, Faster CPUs, GPUs, and More Offload to Accelerators

Optimized Soc Interconnects Are Critical to Performant Soc Designs

SoC interconnects are the connections between all IPs

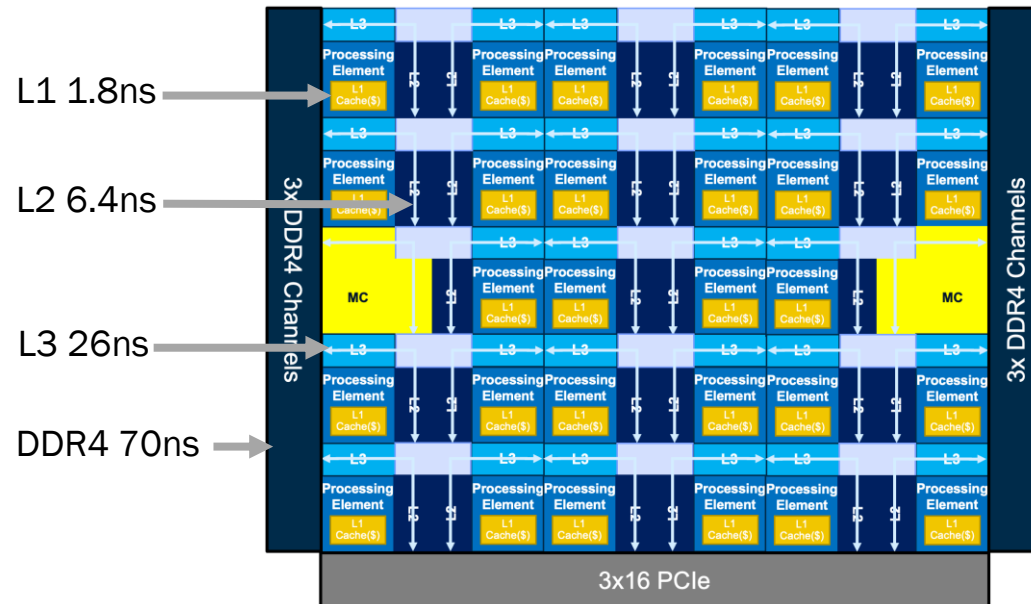
- Interconnects:
  - Traverse the SoC, unlike any other IP
  - Have the longest wires of any IPs
    - Span long distances
    - Congestion points
  - Carry most of the interesting data
  - Change in response to architecture ECOs
  - Must not be the bottleneck of the SoC performance
  - Are the last IP to be frozen – time to backend closure becomes schedule-critical



**CHALLENGE:** How to design the interconnect just right, without overdesign?

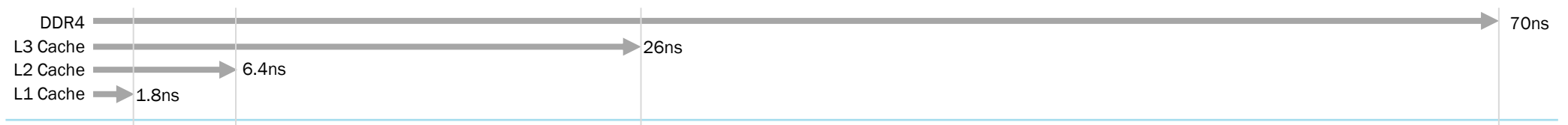
# Using Caches to Boost CPU Efficiency and Data Availability

Architects Need to Balance the Cost of Keeping Data on the SoC



- Memory latency significantly impacts compute performance:
  - Cache minimizes impact
    - Average execution before cache miss
      - ~20 cycles running embedded code
  - DRAM access remains critical for SoC performance
  - DRAM bandwidth must supply caches with vast amounts of data

CPU Freq = 3.1GHz



CPU cycles lost

20

81

217

CPU efficiency

~50%

~20%

~8%!

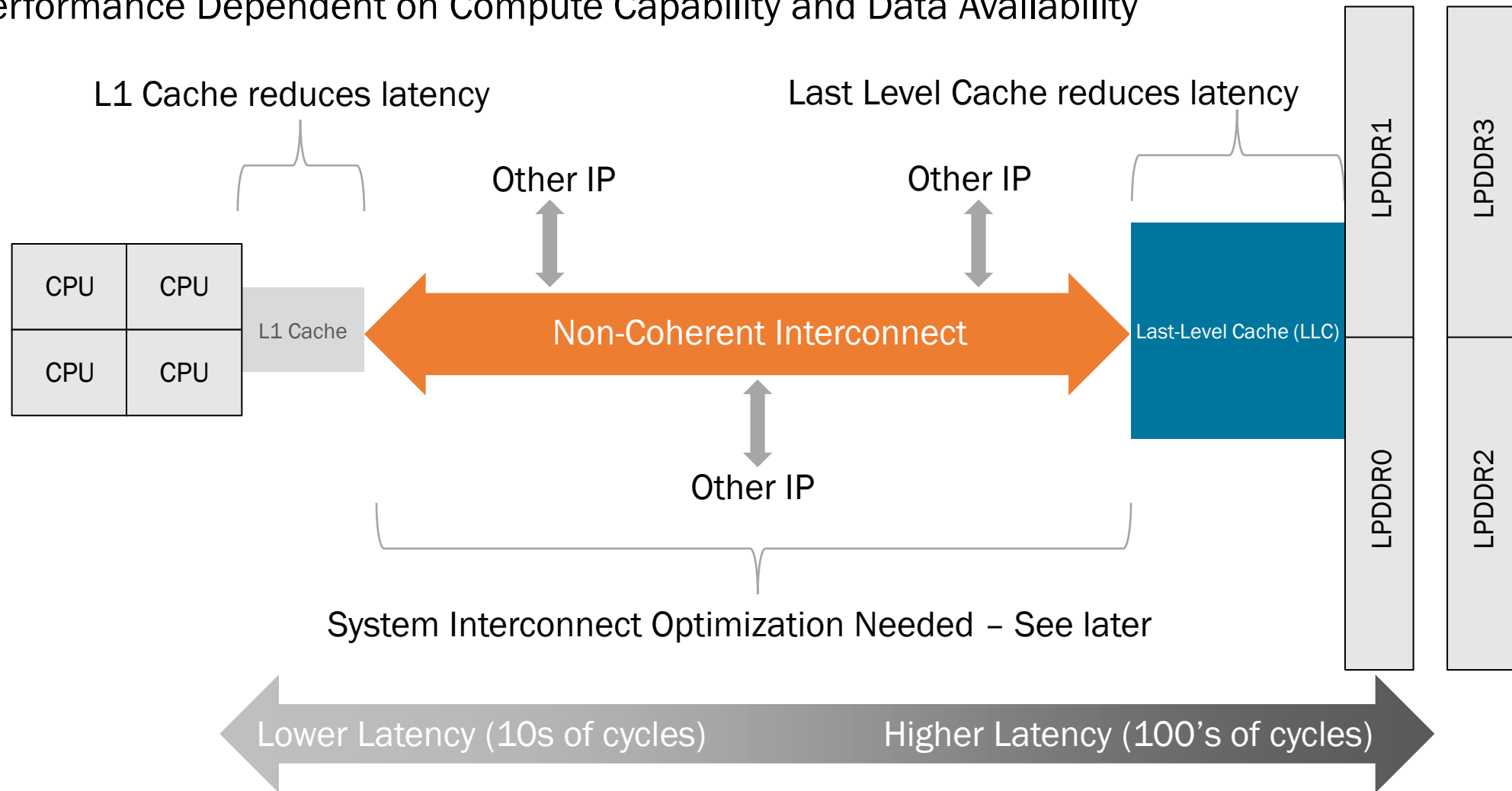
Assuming L1+ L2 Cache

Assuming L1+L2+L3 Cache

Assuming no Cache

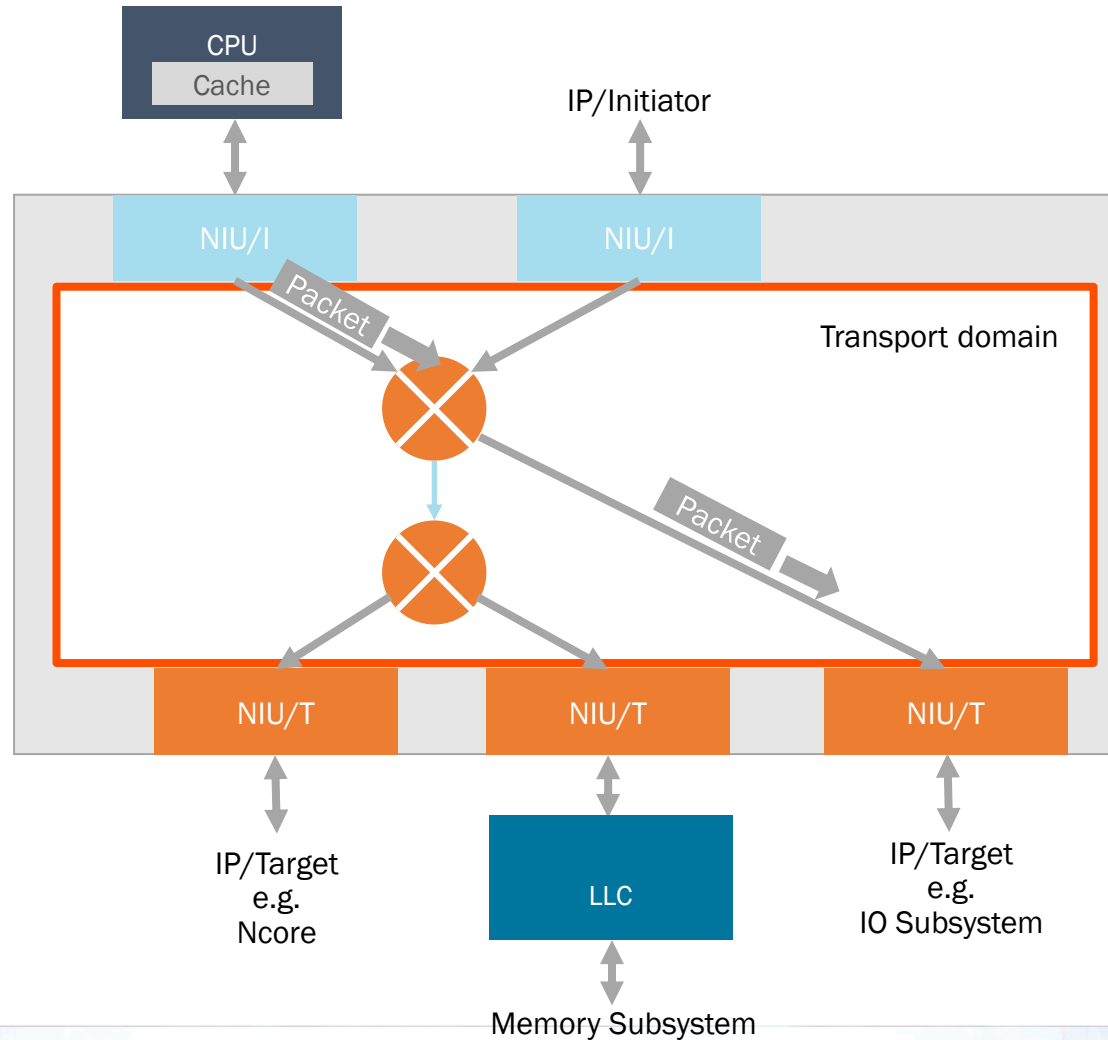
# Maintaining CPU Performance (Non-coherent Caches)

CPU Performance Dependent on Compute Capability and Data Availability



# Non-coherent network-on-chip (NoC) with a last-level cache (LLC)

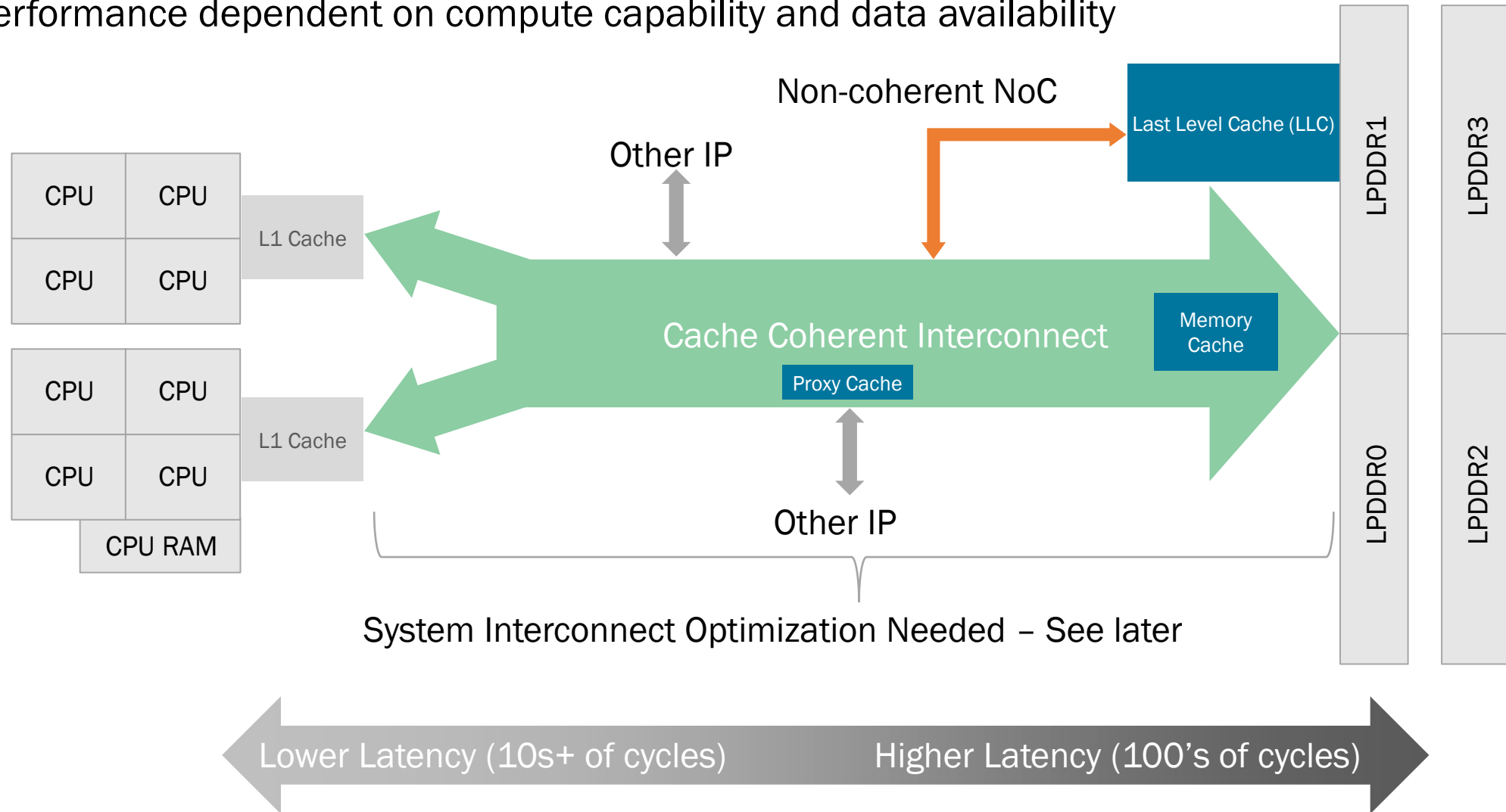
Efficient transport of data through the SoC



- Network-on-chip function
  - Transports **requests** from **initiators** to **targets** and respond back
  - Initiators: CPU, GPU, DMA, ...
  - Targets: SRAM, LLC→DDR, registers...
- Component breakdown
  - **NIU** or Network Interface Units are either Initiators or Target end-points in the network
  - **Transport** is built using **switches**
  - Switches arrangement is the **topology**
  - Transport uses its own **protocol**
    - Fine-tuned for distance spanning, efficiency
  - Transport atoms are called **packets**

# Maintaining CPU performance (Coherent caches)

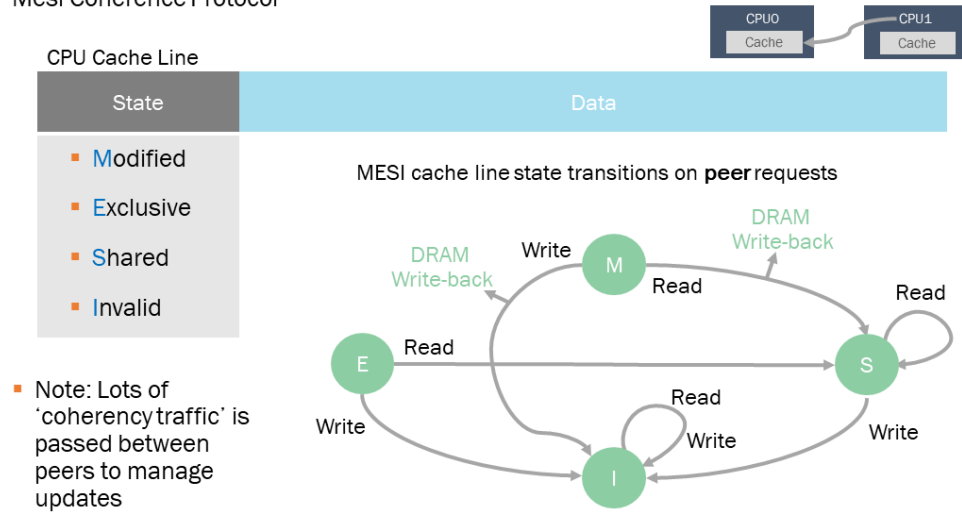
CPU performance dependent on compute capability and data availability



# Cache Protocol & Snoop Directories

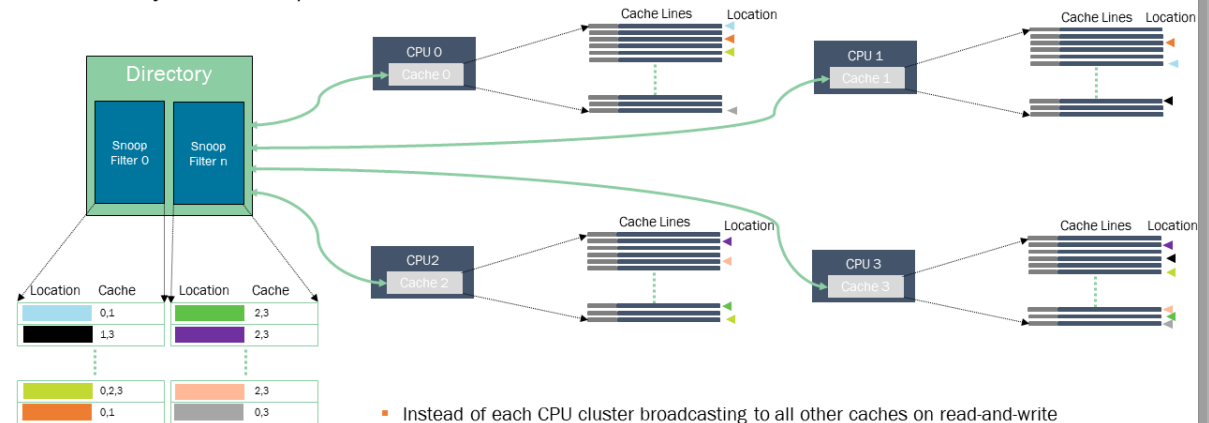
## Cache Coherency Peer CPU Requests

Mesi Coherence Protocol



## Snoop Filters Are Vital to Reduce Coherency Message Overhead

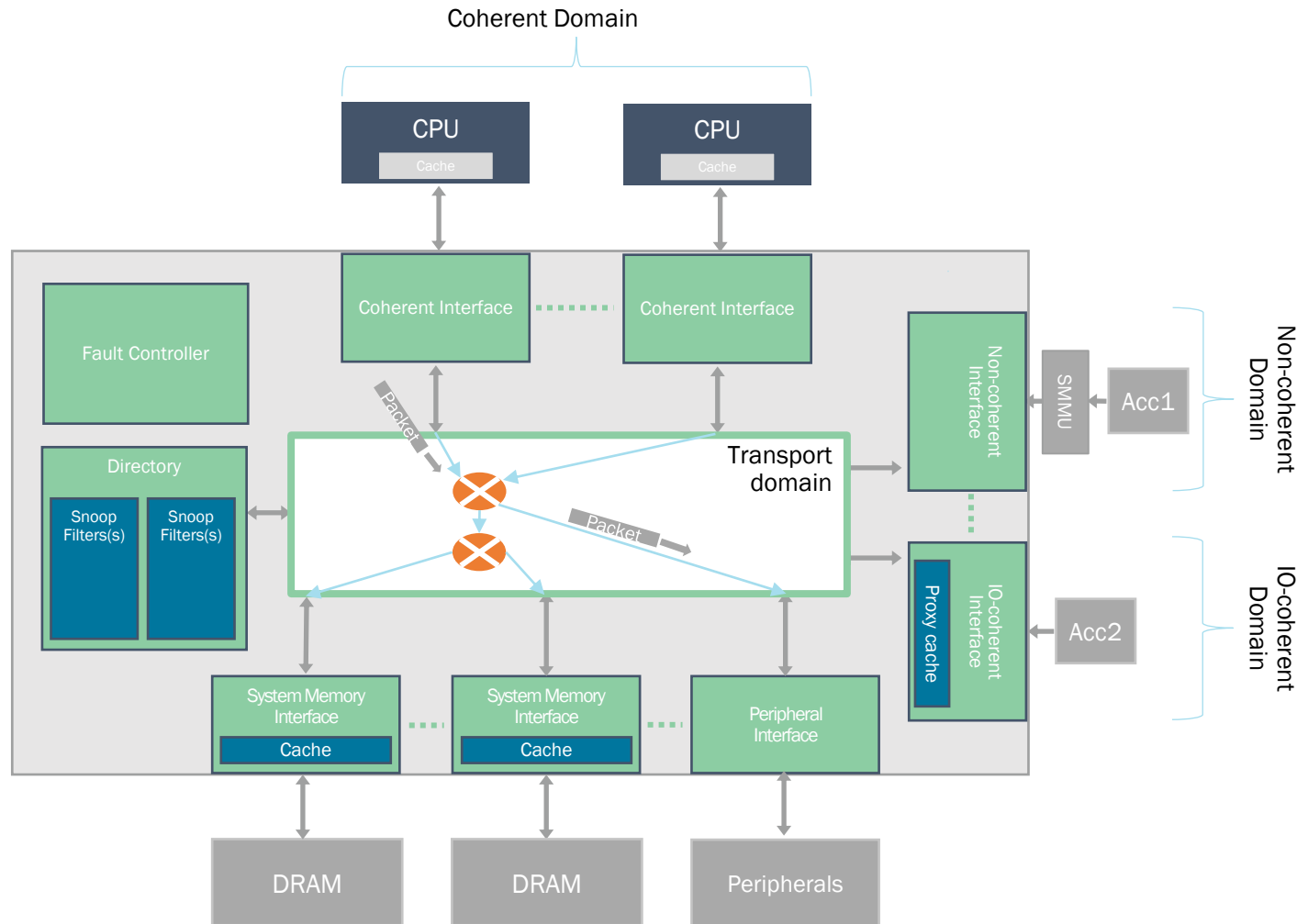
Directory-Based Snoop Filters



- Instead of each CPU cluster broadcasting to all other caches on read-and-write operations, a snoop directory tracks shared cache lines.
- Multiple snoop filters can be used to optimize snoop traffic further.

# Cache-Coherent Network-On-Chip Manages Coherency in Hardware

## Efficient Transport of Coherent Data Through the Soc

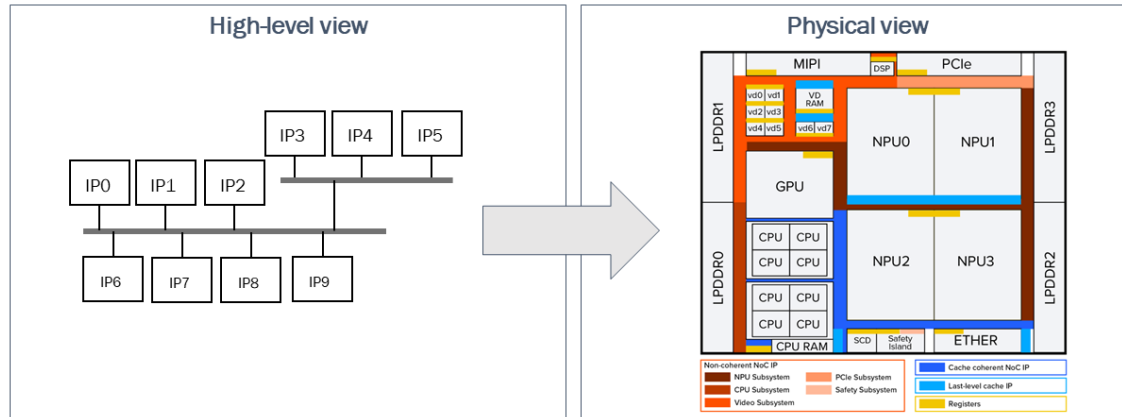


- Network-on-Chip function
  - Same as non-coherent NoC...
  - Hardware coherency is transparent to s/w
- Component breakdown
  - The **transport** is built using **switches**
  - Coherent Interfaces
    - CHI-E, CHI-B, ACE
  - IO Coherent Interfaces with Proxy Cache
    - AXI\*, ACE-Lite
  - Peripheral Interface for complex IO
  - Directory
    - Programmable snoop filters
  - Fault controller
    - For safety option

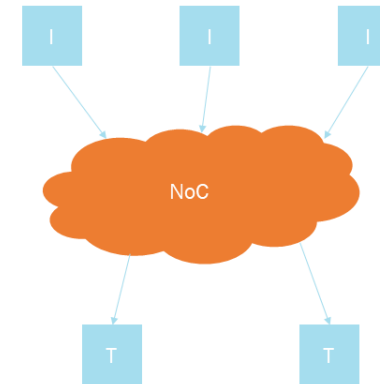
# NoC Implementation: Many Degrees of Freedom

## Noc Interconnects Help Solve Soc Implementation Challenges

SoCs present **unique implementation challenges** for the communication between IP modules and **network-on-chip technology addresses all of them**



## How Many Ways to Implement an Noc?

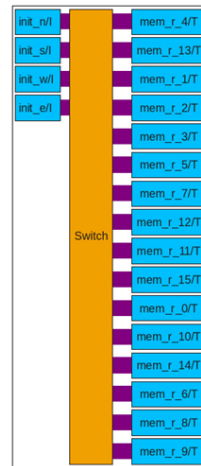


- The problem:
  - Connect initiators (I) to targets (T) using a set of switches and links
  - All I's need to communicate with all T's
  - Pre-requisite knowledge for optimization:
    - What is the amount of traffic between each I and each T?
    - What is the location on the chip of all I and all T?
  - Optimal interconnect implementation means minimizing:
    - Amount of logic in switches
    - Amount of wiring between switches
- Solving this problem is finding an optimal topology!

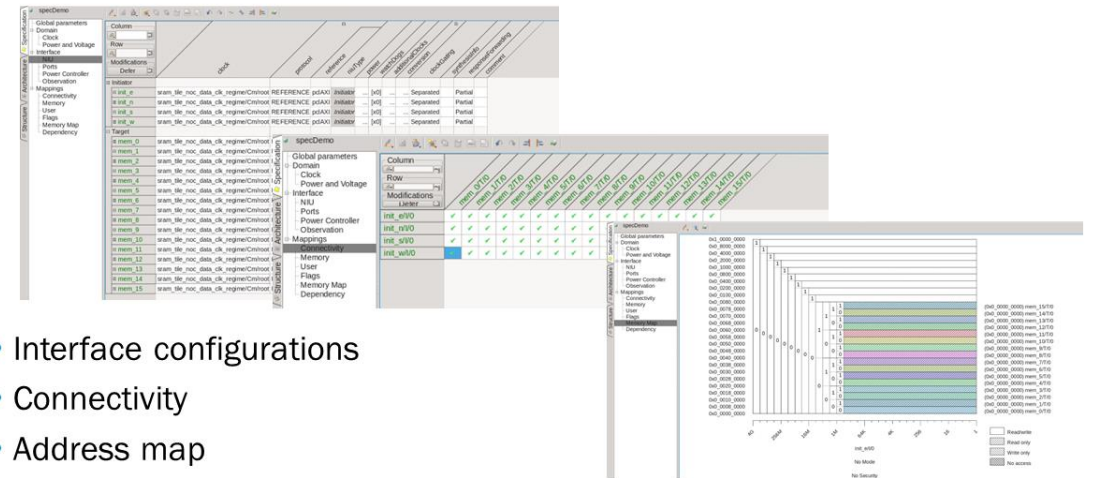
# Going Through an Example

## Signal Processing Design Example

- The use case is signal processing as part of a larger Communications SoC
- The design includes x4 DMAs that need to access 'scratchpad' SRAMs
- A crossbar approach is a good starting point



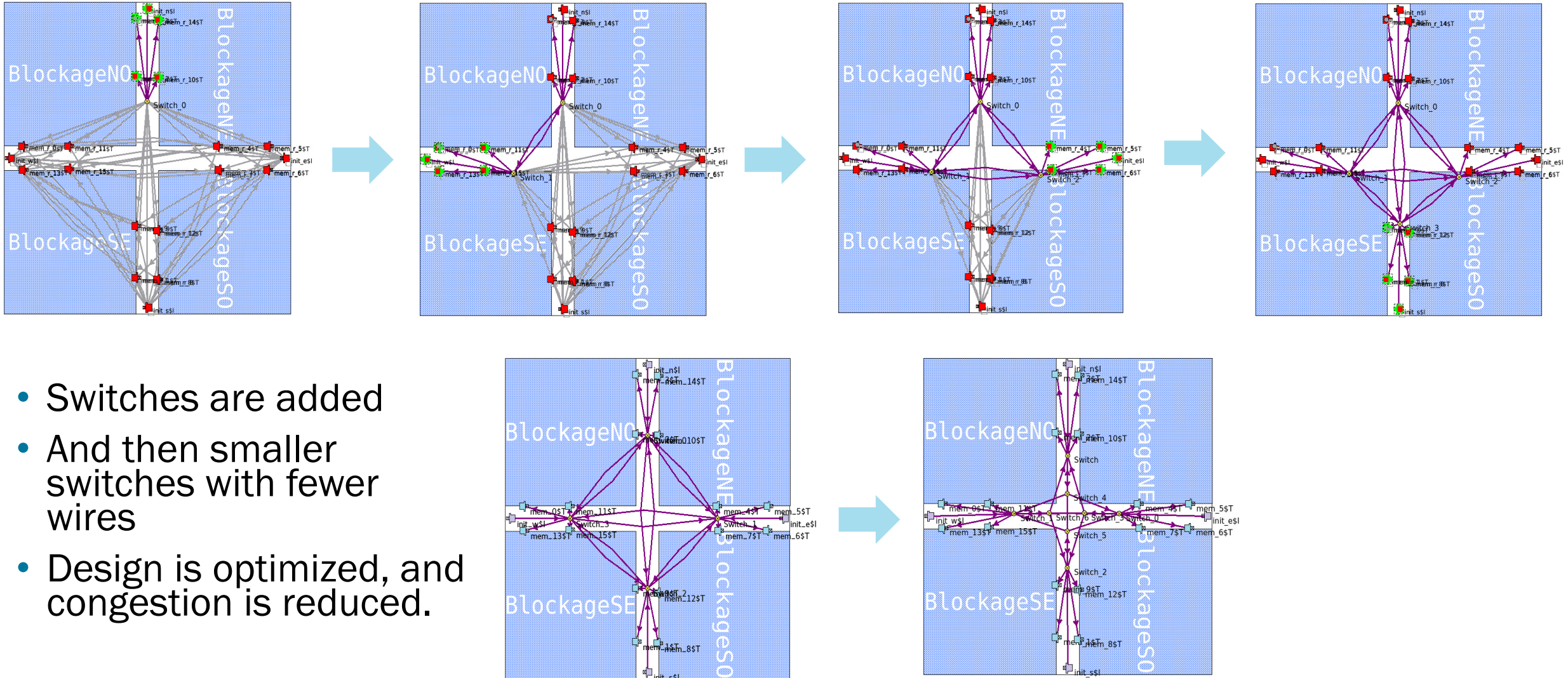
## NoC Specification Capture



- Interface configurations
- Connectivity
- Address map



# More Optimal Version Created using the Physical Layout View

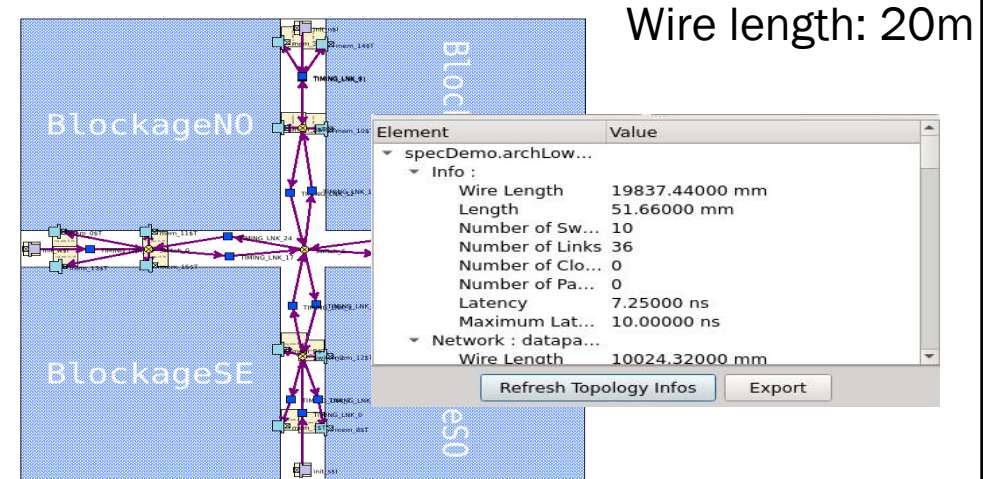
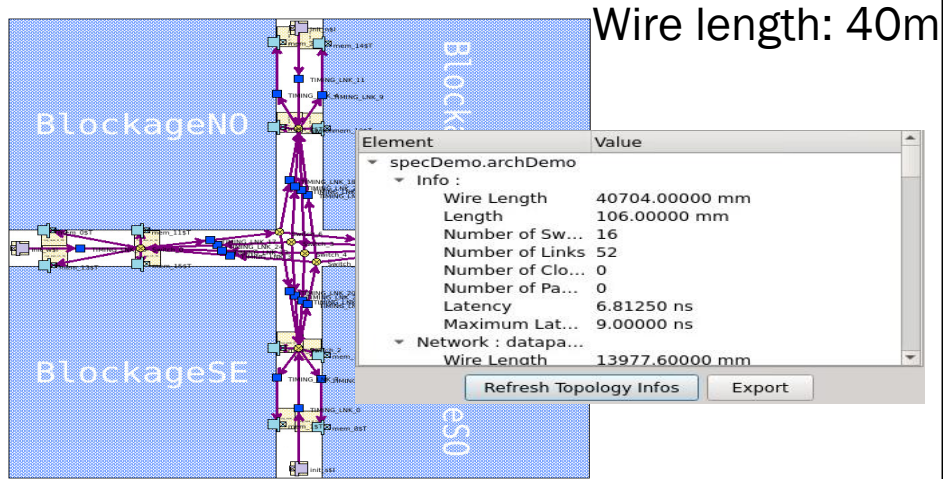


# Design Optimization using Physical and Performance Co-design

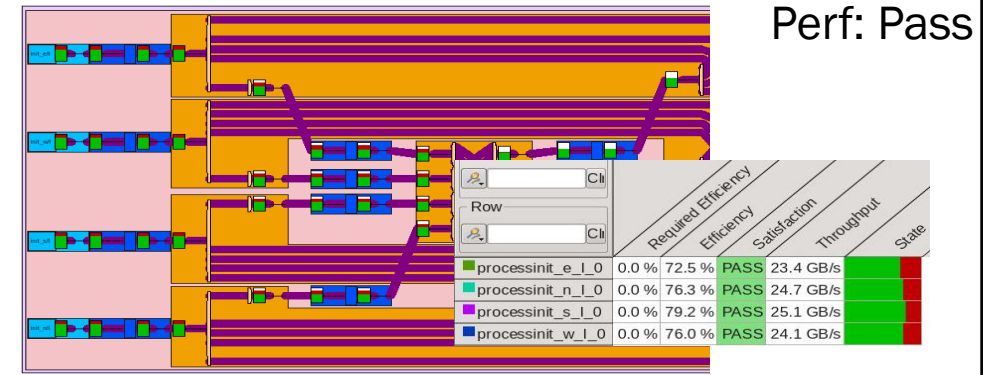
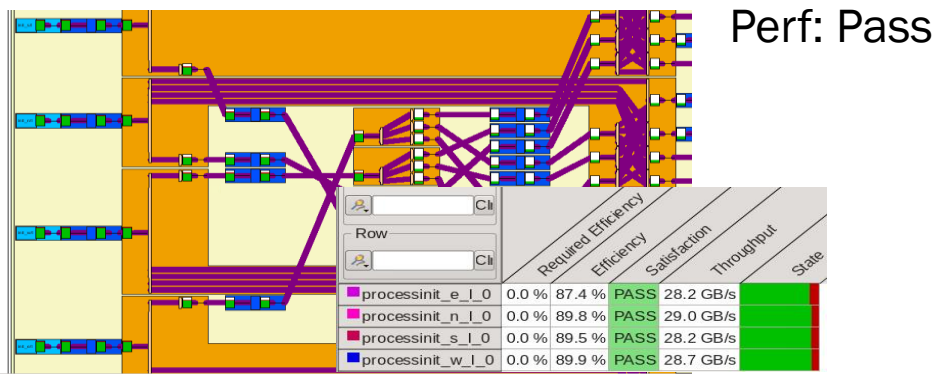
- Before

- After

Physical  
Design  
View



Performance  
Modelling  
View



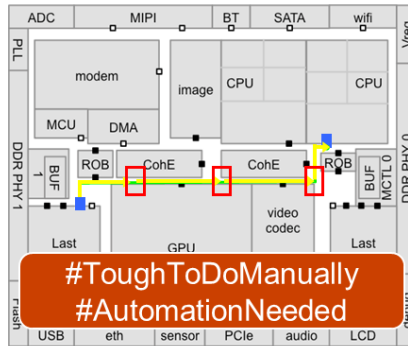
# Pipeline Insertion

## Timing Issues: Can't Cross Advanced Node SoCs in One Clock Cycle

Physical distance impacts the number of pipeline stages



Clock Cycles

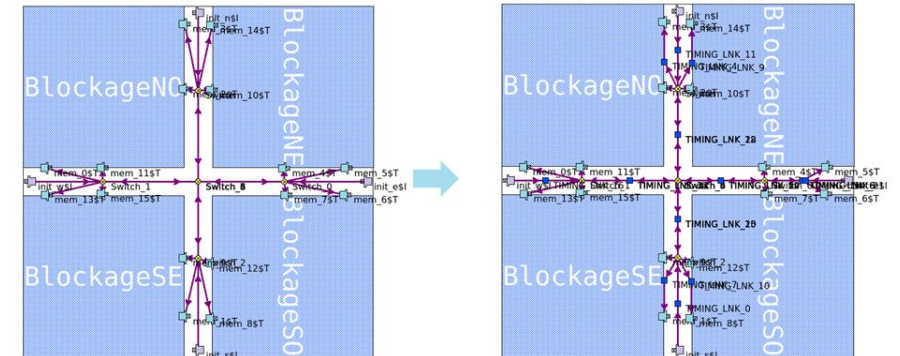


- Endpoint (NIU)
- Single NoC cycle path
- Pipeline

#ToughToDoManually  
#AutomationNeeded

Transport delay =  $F$  (foundry, routing stack, type of driving cell, process voltage, temperature,...)

## Pipeline Placeholders are Added to Assist Timing Closure



E.g. 1GHz design, 5mm<sup>2</sup>, 1ns per mm propagation speed

# Revolutionizing SoC Performance With a NoC

Your SoC, your (NoC) Topology!

- SoC complexity is growing as more IP is integrated on chip
- Complexity puts additional demand on the interconnects to handle the on-chip communication between all IP
- NoC interconnects manage the complexity and enable designers to optimize the topology to the specific physical and performance constraints of their SoC
  - Non-coherent NoCs are used for the transport of data efficiently on and off the chip
  - Cache-coherent NoCs are used in concert with non-coherent NoCs for managing coherency domains
- **We've just scratched the surface on the power of NoCs** – connect with us at Arteris to learn more!



SHAPING THE NEXT GENERATION OF ELECTRONICS

**JUNE 23-27, 2024**

MOSCONE WEST CENTER  
SAN FRANCISCO, CA, USA

# Thank You!

[guillaume.boillet@arteris.com](mailto:guillaume.boillet@arteris.com)

