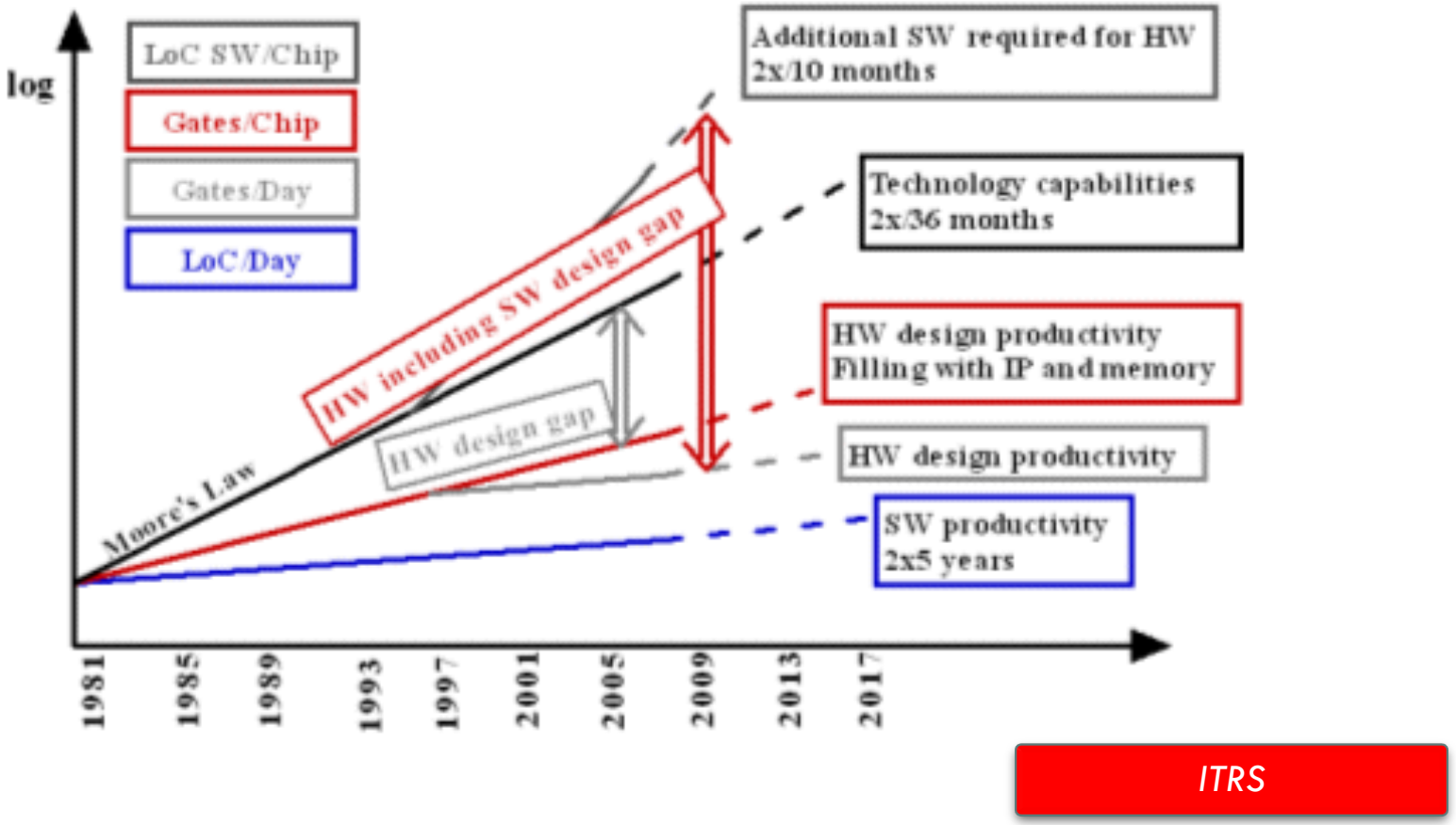
NETWORKS-ON-CHIP AND NOC SYNTHESIS

Federico ANGIOLINI
LSI EPFL
2018

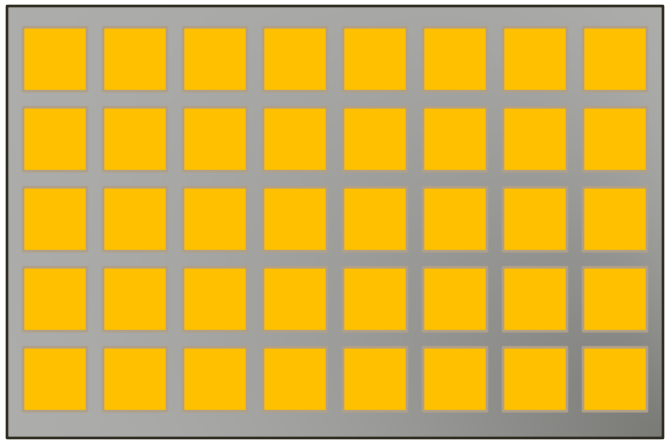
CORE-BASED DESIGN

Federico ANGIOLINI
LSI EPFL
2018

A WIDENING PRODUCTIVITY GAP

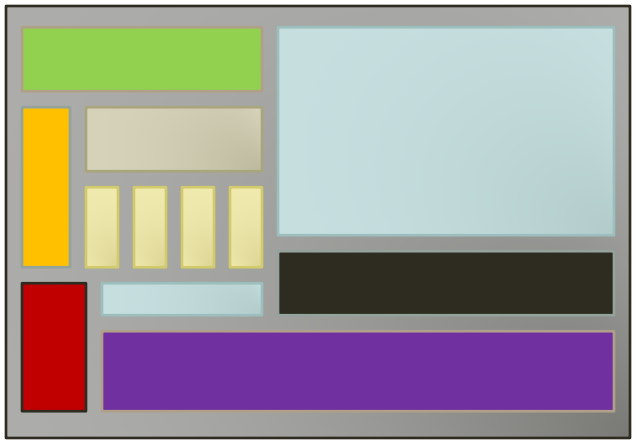


TWO TRENDS: CMPS AND MPSOCS



Chip MultiProcessor (CMP)

Intel, Tilera, STM



MultiProcessor System-on-Chip (MPSoC)

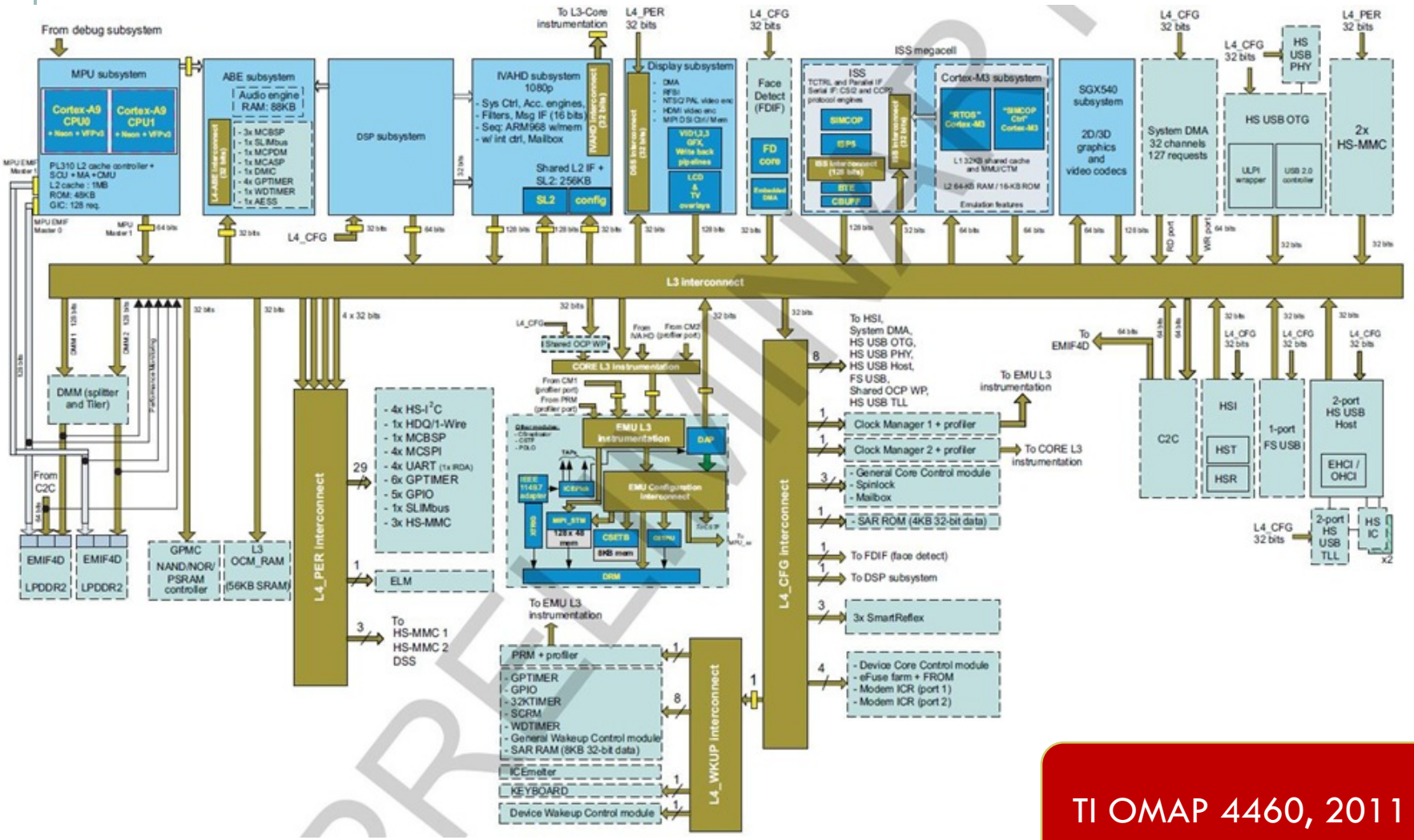
Qualcomm, Samsung, TI, Apple

Blurred boundaries…

CMPS VS. MPSOCS

CMP	MPSoC
Fully homogeneous	Very heterogeneous
Fully programmable cores	Mostly fixed-function cores
Tile-based	Subsystem-based
Regular chip layout	Heavily customized layout
Emphasis on ease of programming	Emphasis on power/performance/cost
For: general processing, programmable accelerators	For: embedded applications, optimized accelerators

MPSOC SYSTEMS BECOME VERY COMPLEX



TI OMAP 4460, 2011

INTEGRATION ISSUE ARISES

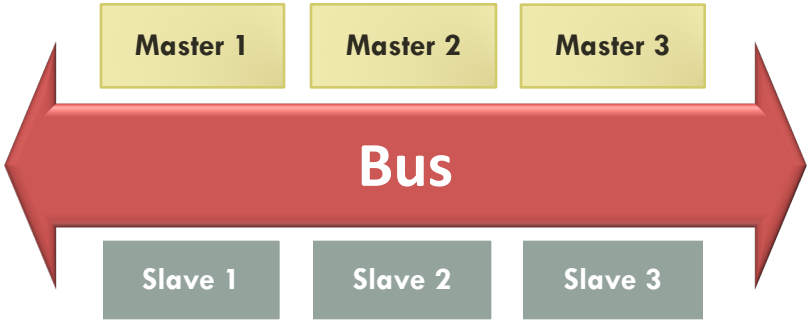
- Many cores
- If MPSoC, heterogeneous cores
  - Different pinouts
  - Different clocking (+DVFS)
  - Different physical sizes
  - Different power/temperature budgets
  - Different communication needs
- How to integrate, verify?
- System interconnect

ON-CHIP INTERCONNECTS

Federico ANGIOLINI
LSI EPFL
2018

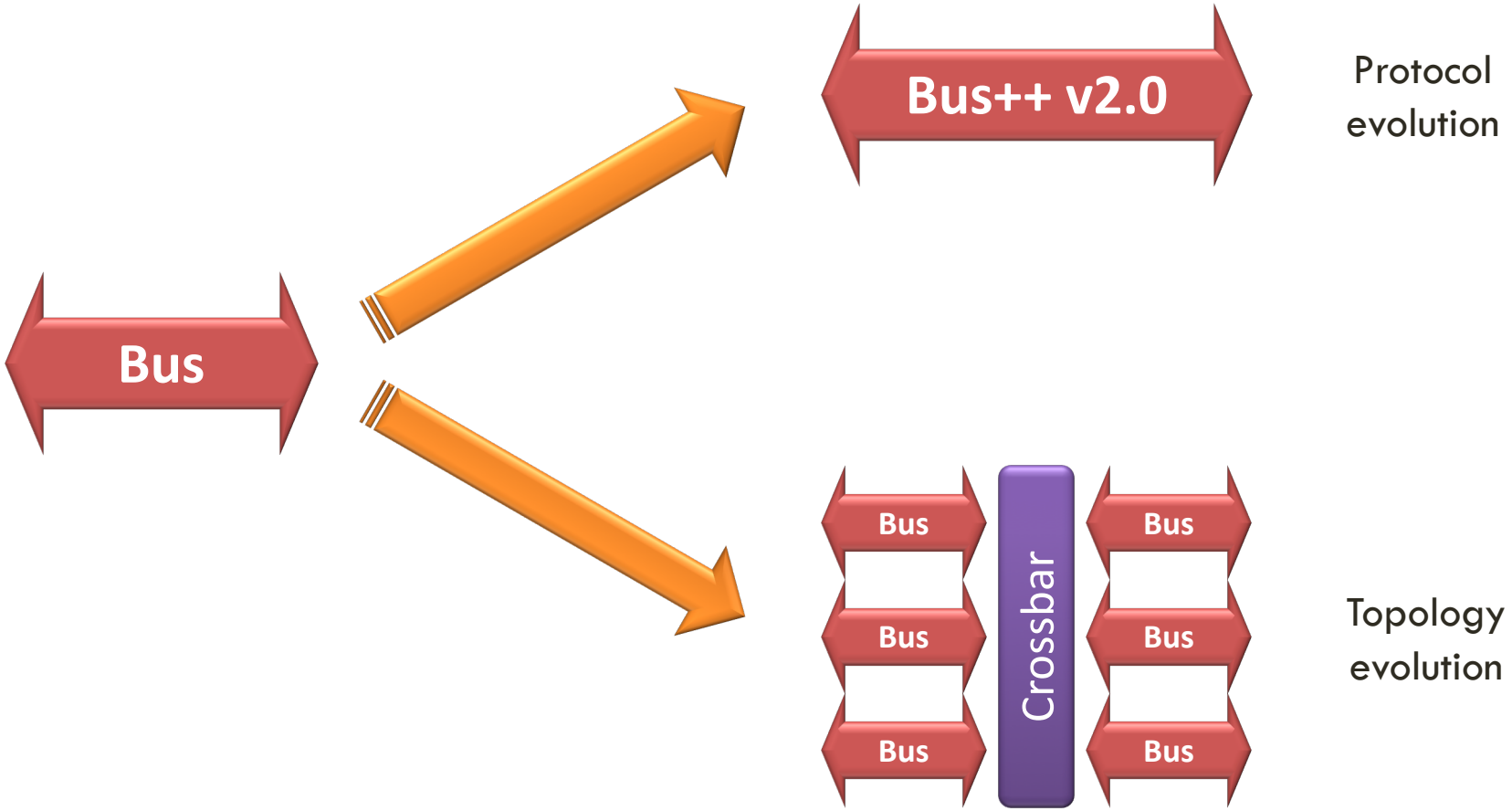
TRADITIONAL ANSWER: BUSES

- Shared bus topology
- Aimed at simple, cost-effective integration



- Typical example: ARM AMBA AHB
  - Arbitration among multiple masters
  - Single outstanding transaction allowed
  - If wait states are needed, everybody waits
  - Slow!

BUS EVOLUTION OCCURRED



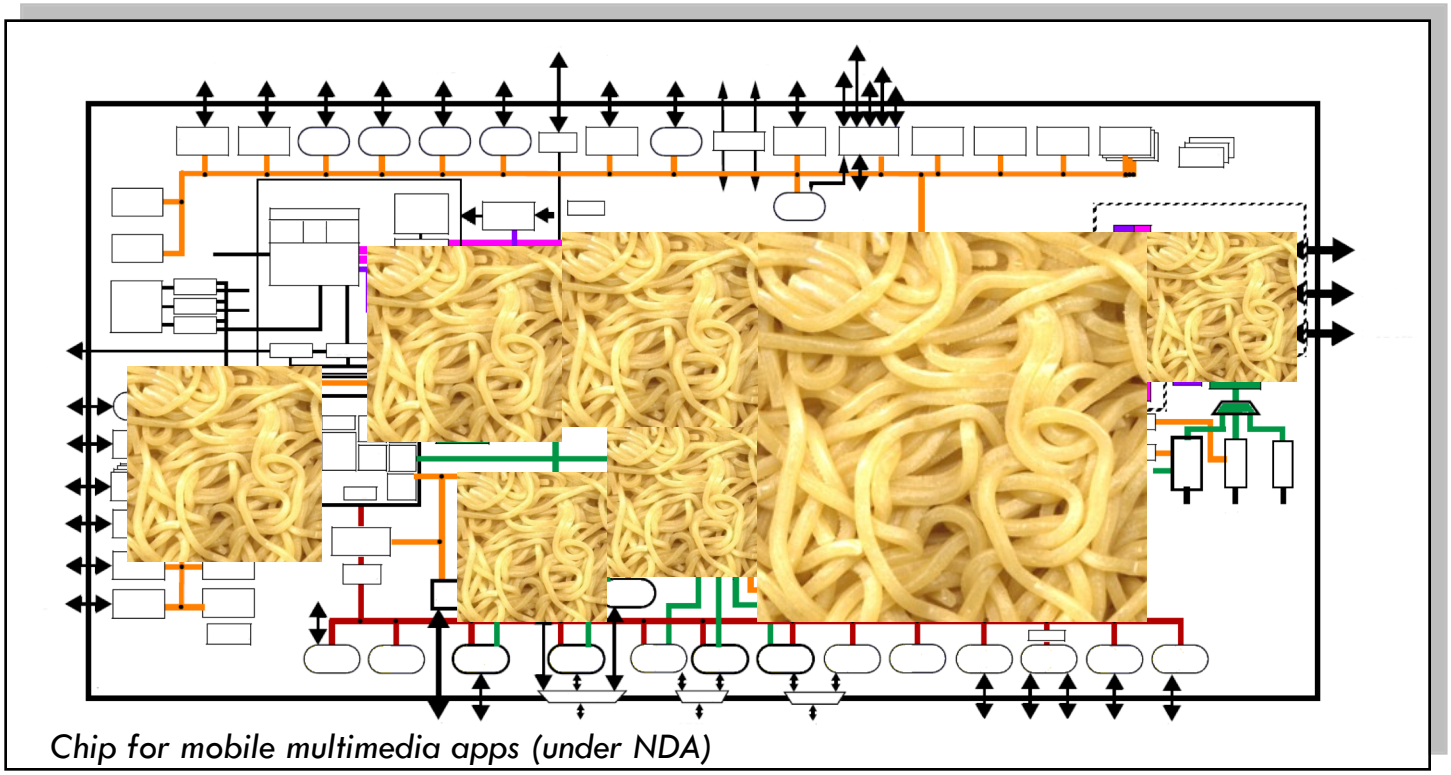
STILL OUTSTANDING PROBLEMS

- Performance scalability
- Ease of design and verification
  - ~2 years time to market (89% late!)

Numetrics Management Systems

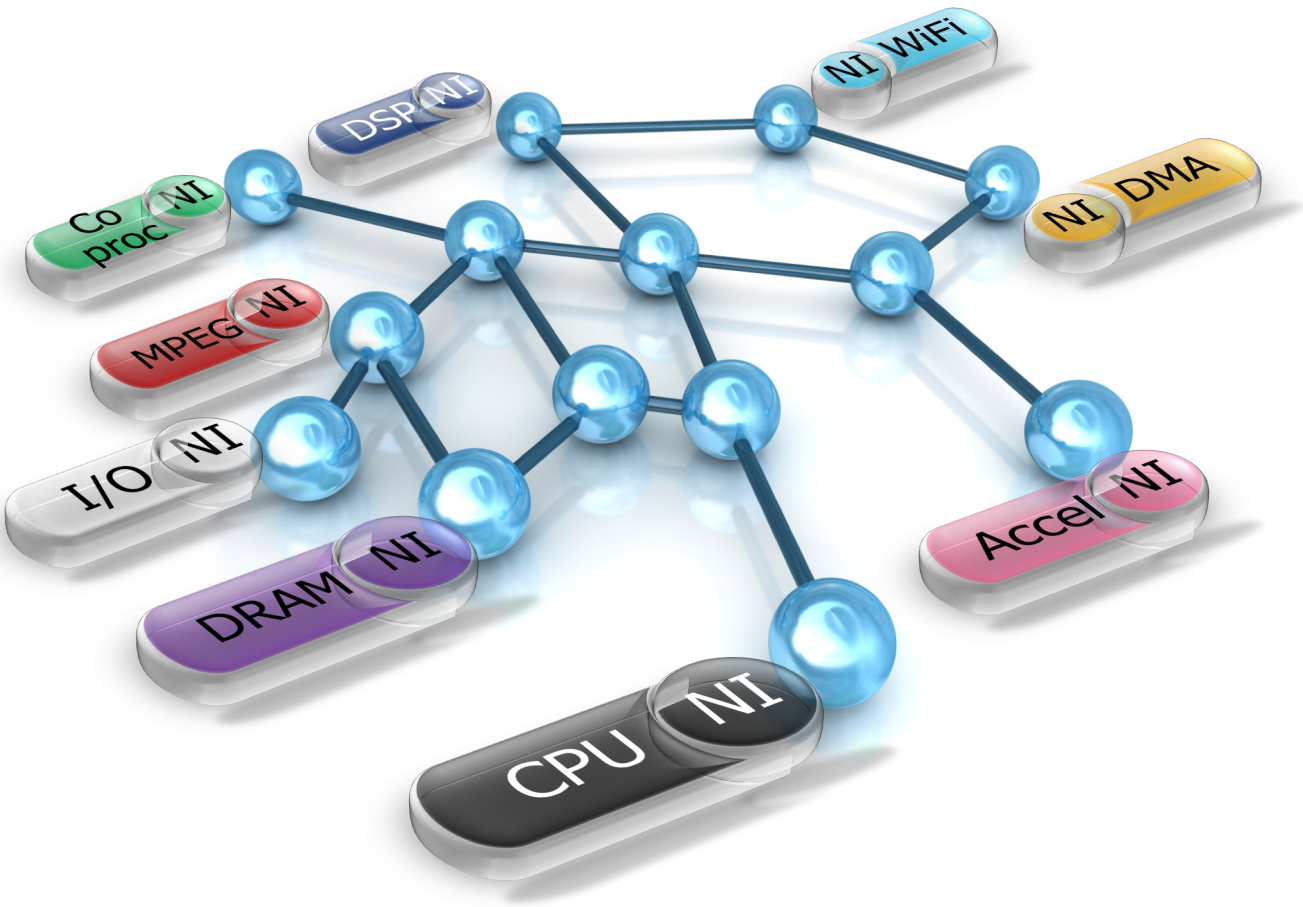
- Physical design issues
  - Routing congestion
  - Wire propagation delay

AND OVERALL, THE DESIGN LOOKS LIKE…



Chip for mobile multimedia apps (under NDA)

THE RISE OF NETWORKS-ON-CHIP (NOCS)

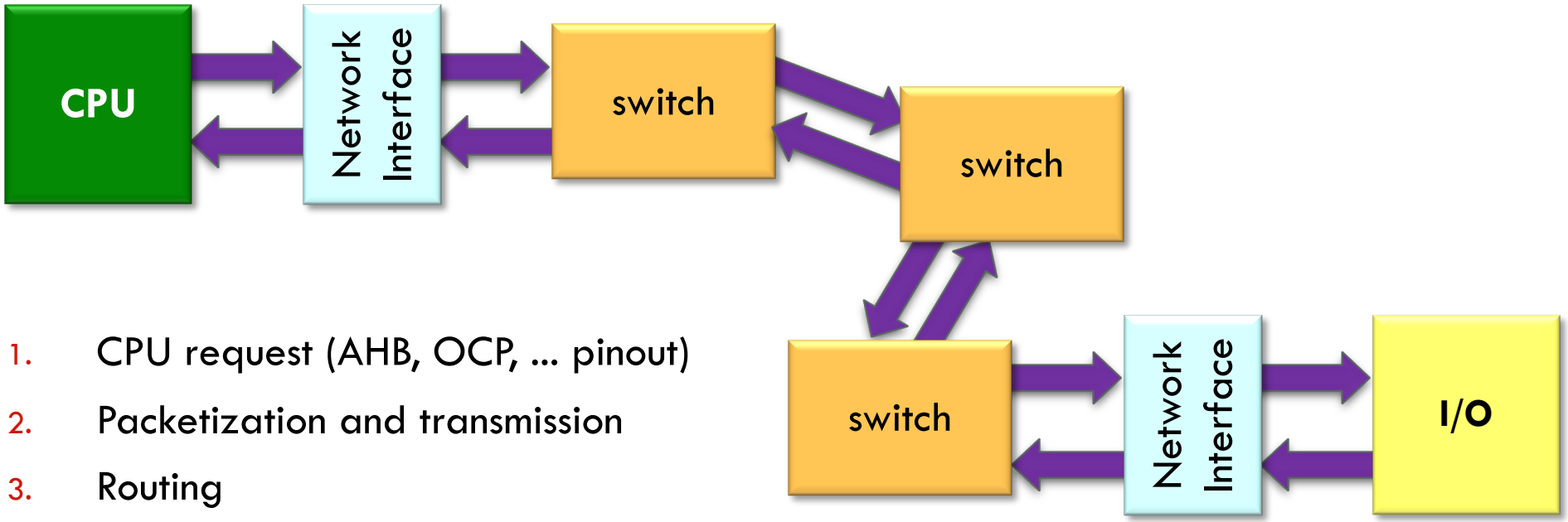


- Packet-based communication
- NIs packetize transactions
- Switches route transactions across

Dally, 2001

Benini-De Micheli, 2002

NOC OPERATION EXAMPLE



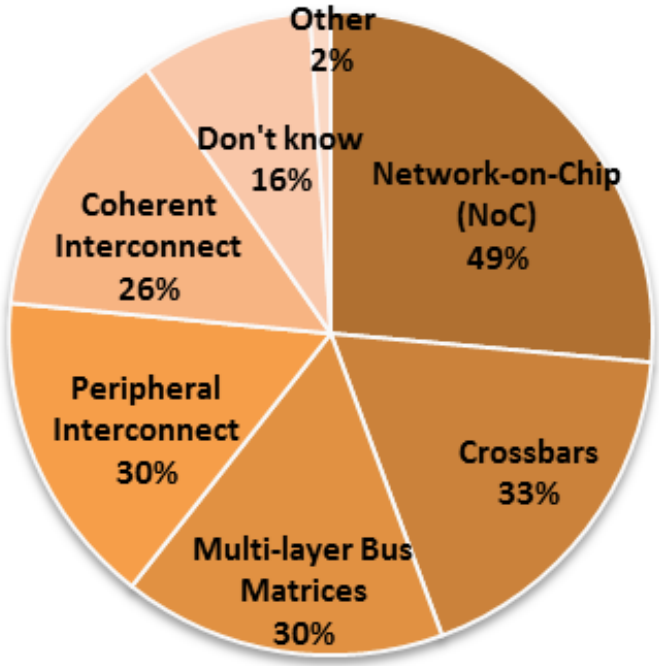
1. CPU request (AHB, OCP, … pinout)
2. Packetization and transmission
3. Routing
4. Receipt and unpacketization (AHB, OCP, … pinout)
5. Device response (if needed)
6. Packetization and transmission
7. Routing
8. Receipt and unpacketization

NOC HERITAGE: A LOGICAL STEP

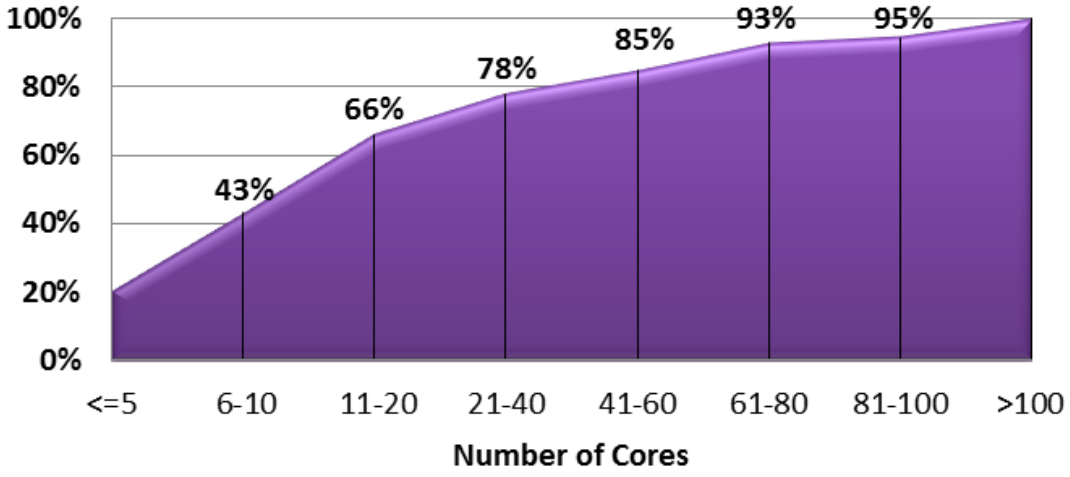
- On-chip version of large area networks
  - WAN, MAN, LAN, PAN, …
- Next logical step in interconnect evolution
- Scalable, robust, efficient, well-understood
- Different trade-offs compared to large area networks
  - Wire parallelism is cheap vs. cables
  - Latency, power, area must be orders of magnitude lower
  - Buffers must be extremely small

NOC ADOPTION TRENDS

Topologies considered for next On-chip communications network



Number of Cores when a Commercial NoCs become important consideration over internal development



Sonics Whitepaper, 2012

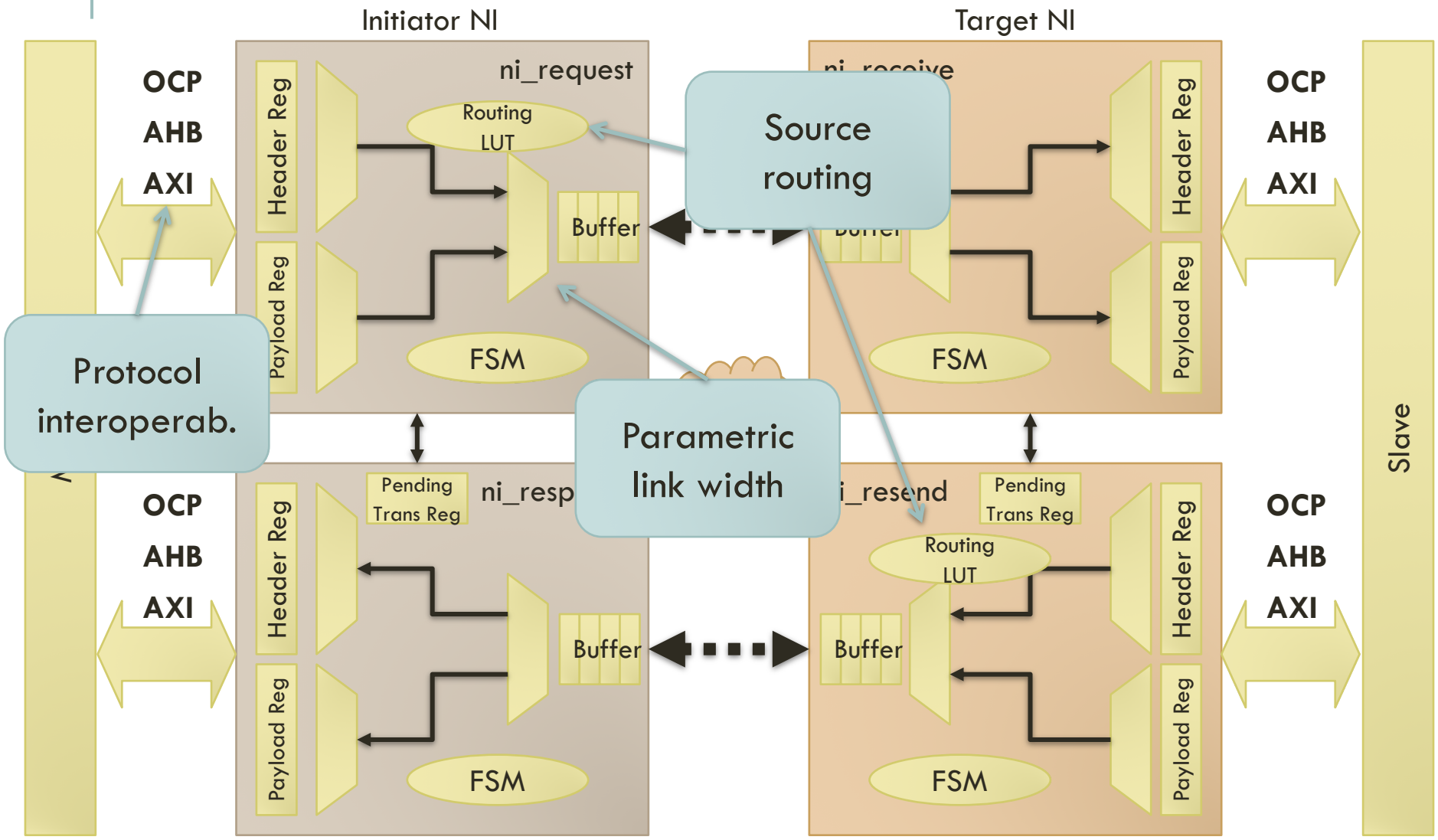
HOW TO DESIGN NOCS

Federico ANGIOLINI
LSI EPFL
2018

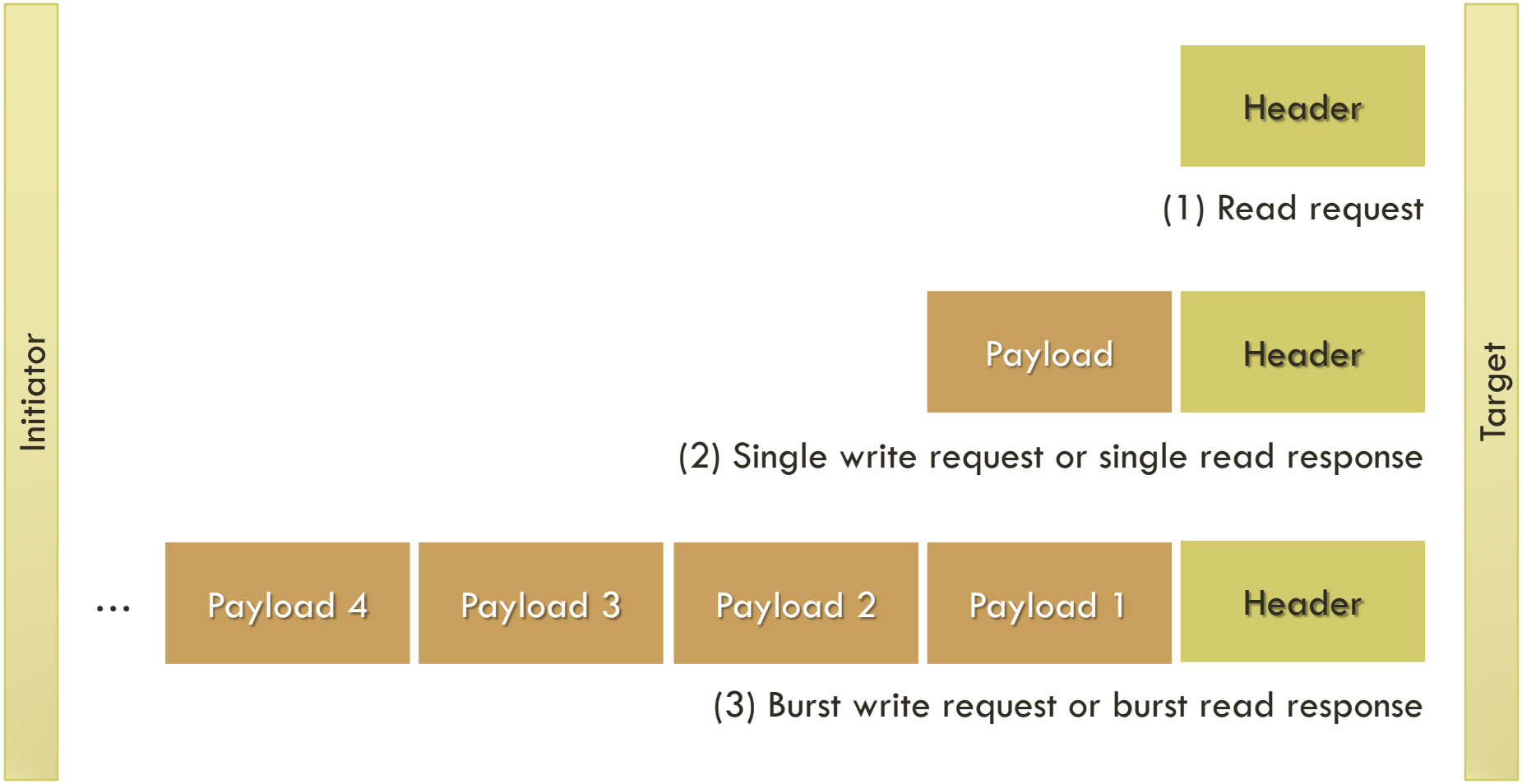
NOCS HAVE MANY DESIGN AXES

- How to connect things: topology
- How to inject and eject messages: network interface architecture (incl. packetization policy)
- How to pass packets along: router architecture (incl. switching policy, flow control policy)
- Where to send packets: routing policy

NETWORK INTERFACE ARCHITECTURE (XPIPES)



PACKET TRANSMISSION



INTEROPERABLE PACKET FORMAT

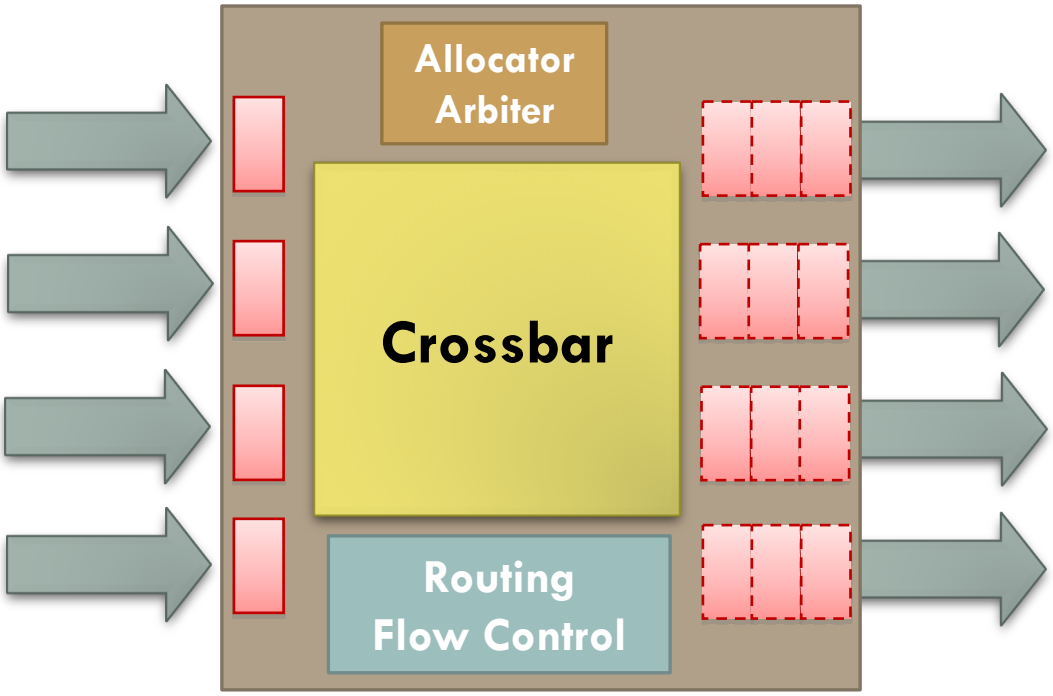
request header

request payload

response header

response payload

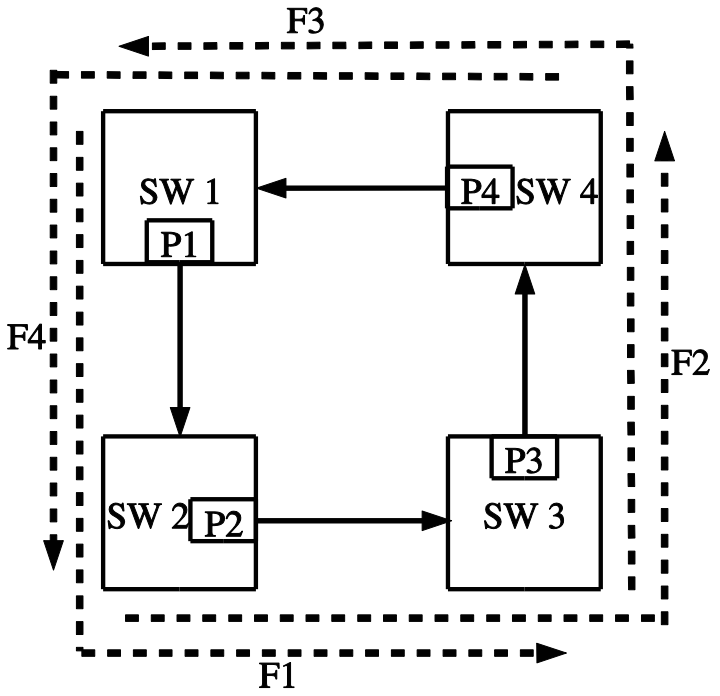
ROUTER ARCHITECTURE (XPIPES)



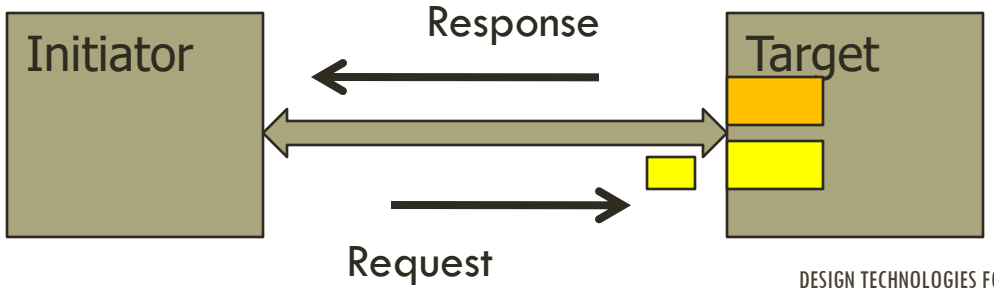
- Input and/or output buffering
- Wormhole switching
- Supports multiple flow control policies

DEADLOCKS

- Routing level



- Message level



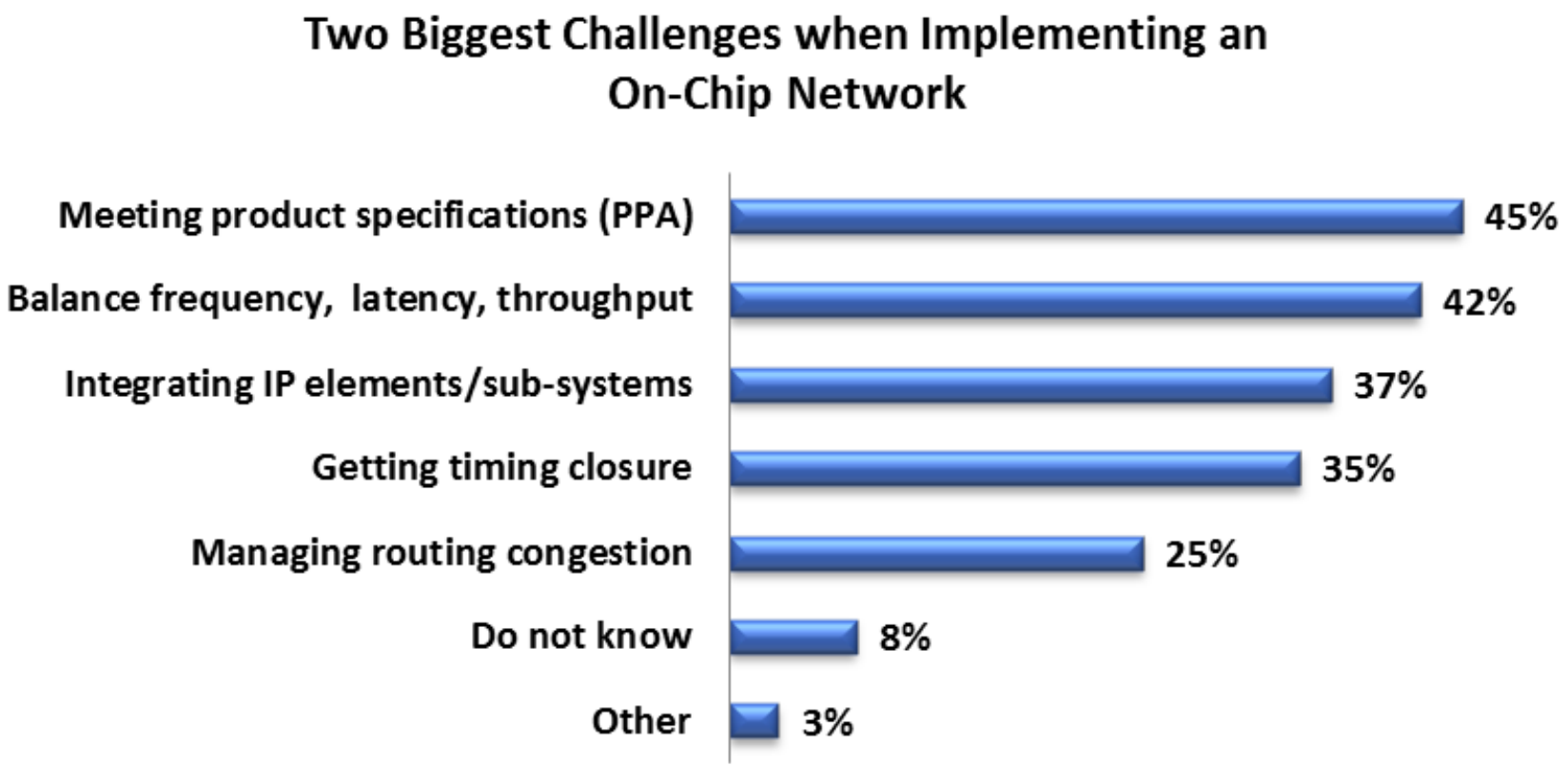
APPLICATION-SPECIFIC NOC DESIGN

Federico ANGIOLINI
LSI EPFL
2018

CMP VS. MPSOC INTERCONNECTS

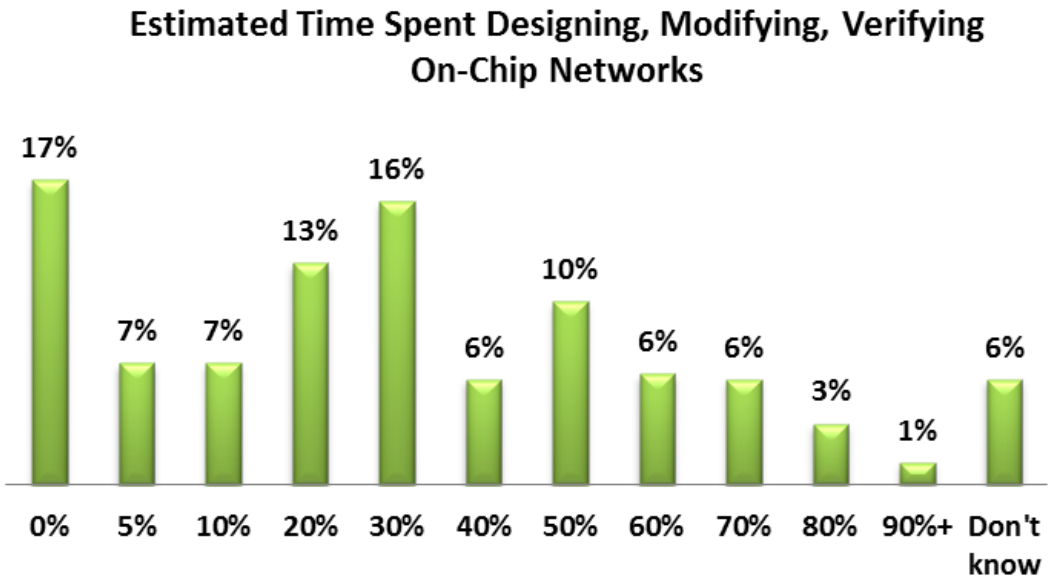
CMP	MPSoC
Regular interconnect (mesh, torus)	Custom topology
Unpredictable workload	Mostly predictable workloads
Synthetic workloads may be representative	All usage scenarios must be checked
Emphasis on routing, bisection bandwidth	Emphasis on latency, simulated performance
Homogeneous NoC architecture	Locally optimized architecture
Requires cache coherence	Requires core interoperability
Regular, short wiring	Irregular, long wiring

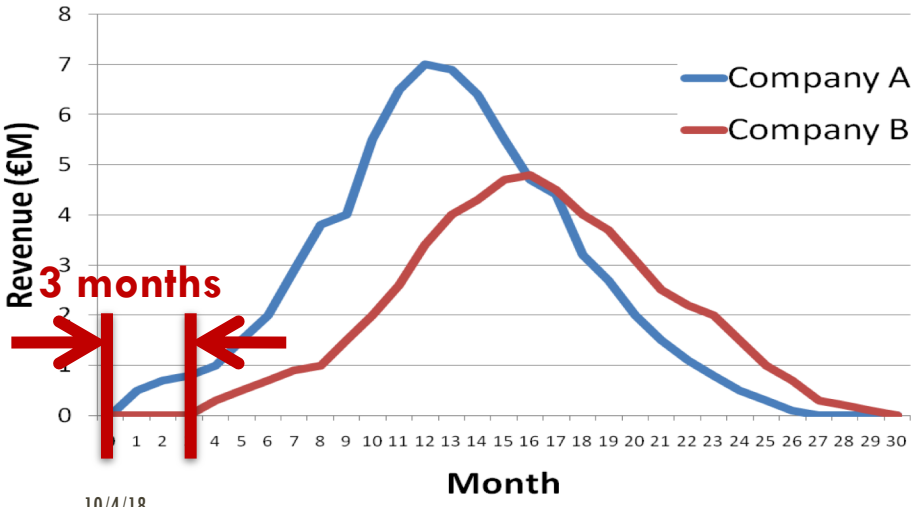
MANY CHALLENGES TO TACKLE



Sonics Whitepaper, 2012

...WHICH IS EXPENSIVE





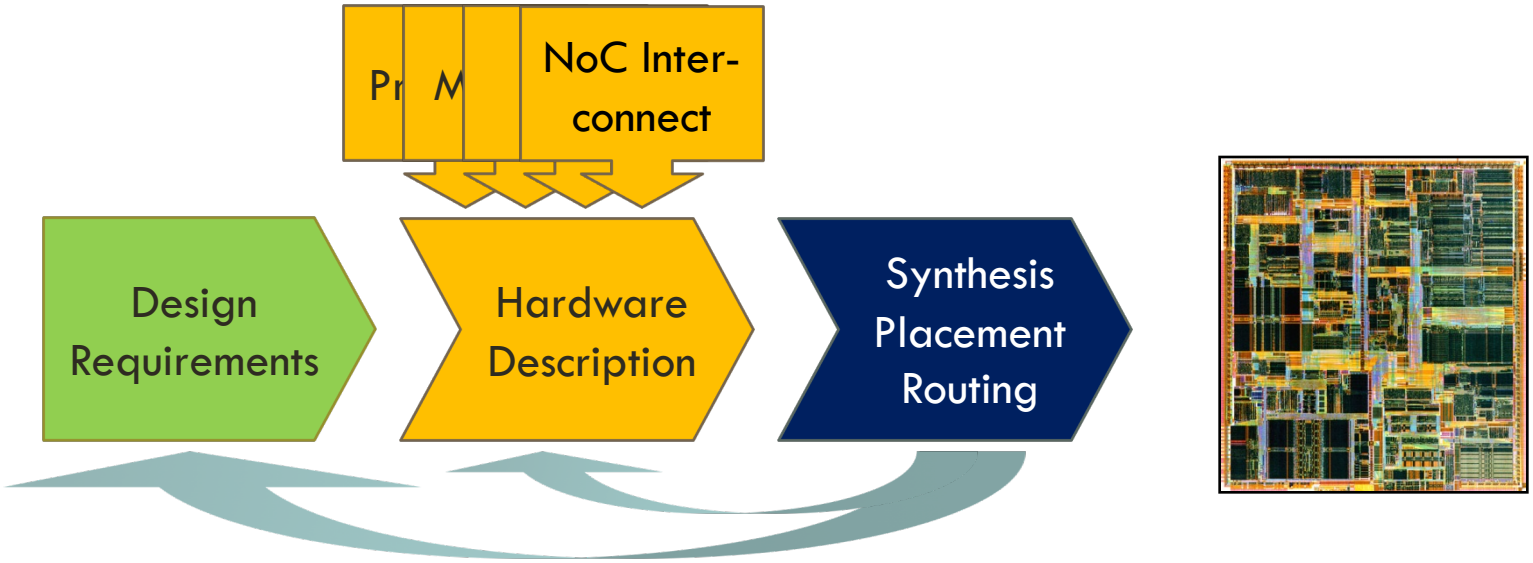
MPSoC architects spend ~28% of time defining the NoC

Sonics Whitepaper, 2012

Company A: € 76 M (profitable)

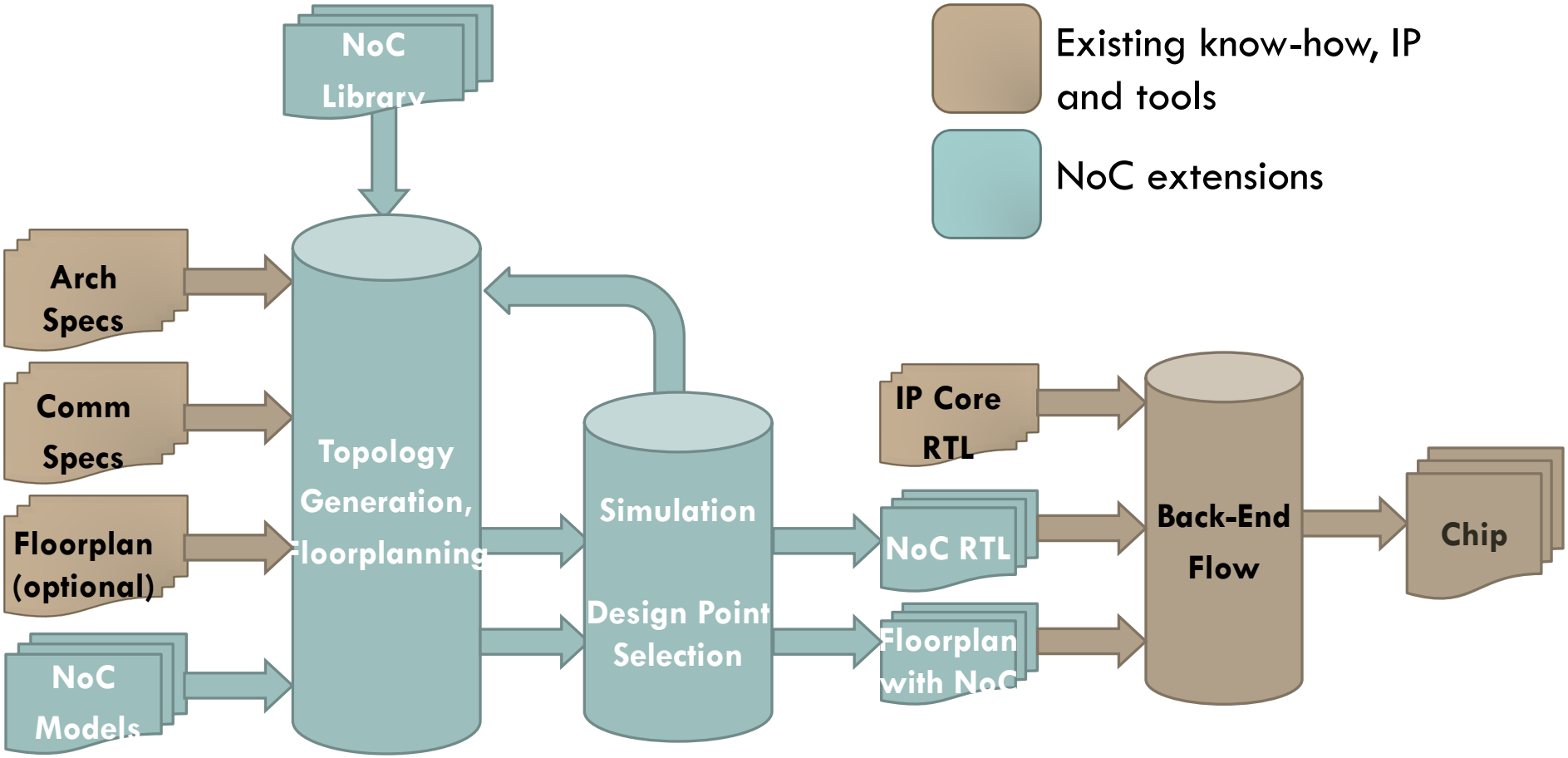
Company B: € 56 M (maybe bankrupt)

STANDARD DESIGN FLOW

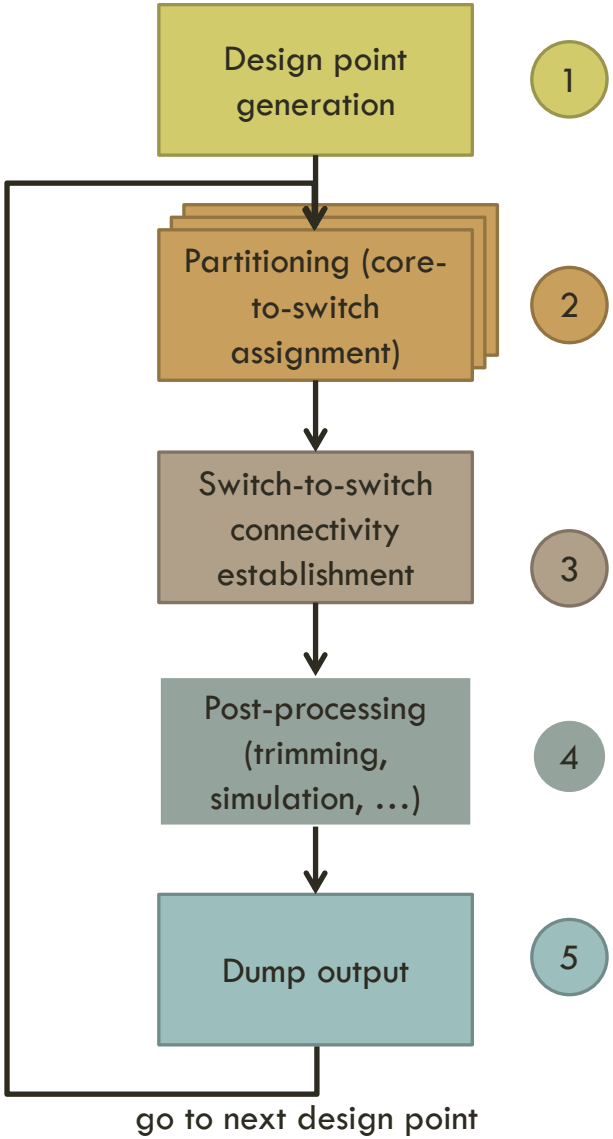


- Designer picks NoC with expertise + guesswork
- NoC is iterated on multiple times based on simulation and synthesis feedback
- Electronic Design Automation (EDA) Tools exist but most difficult work is manual

AN AUTOMATED DESIGN FLOW



REFERENCE ALGORITHM



Murali, DAC 2004
Murali, ASPDAC 2005
Bertozzi, IEEE TPDS 2005

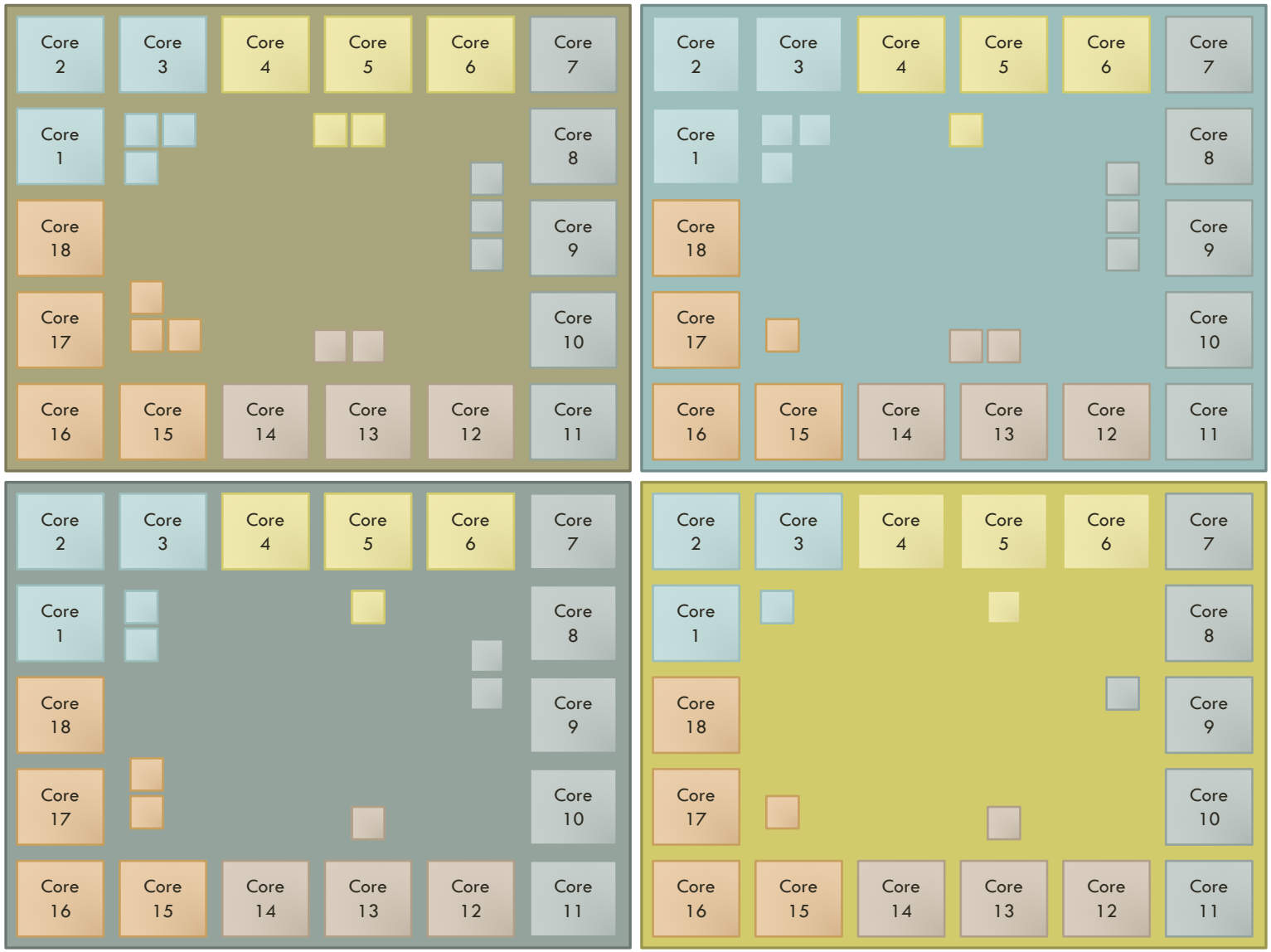
Other seminal works:
Pinto, ICCD 2003
Srinivasan, ICCD 2004

1 - DESIGN POINT GENERATION

- Identify NoC parameter permutations
  - NoC flit width
  - Switch count in each V/f island
  - Potentially any others: buffering, VCs…

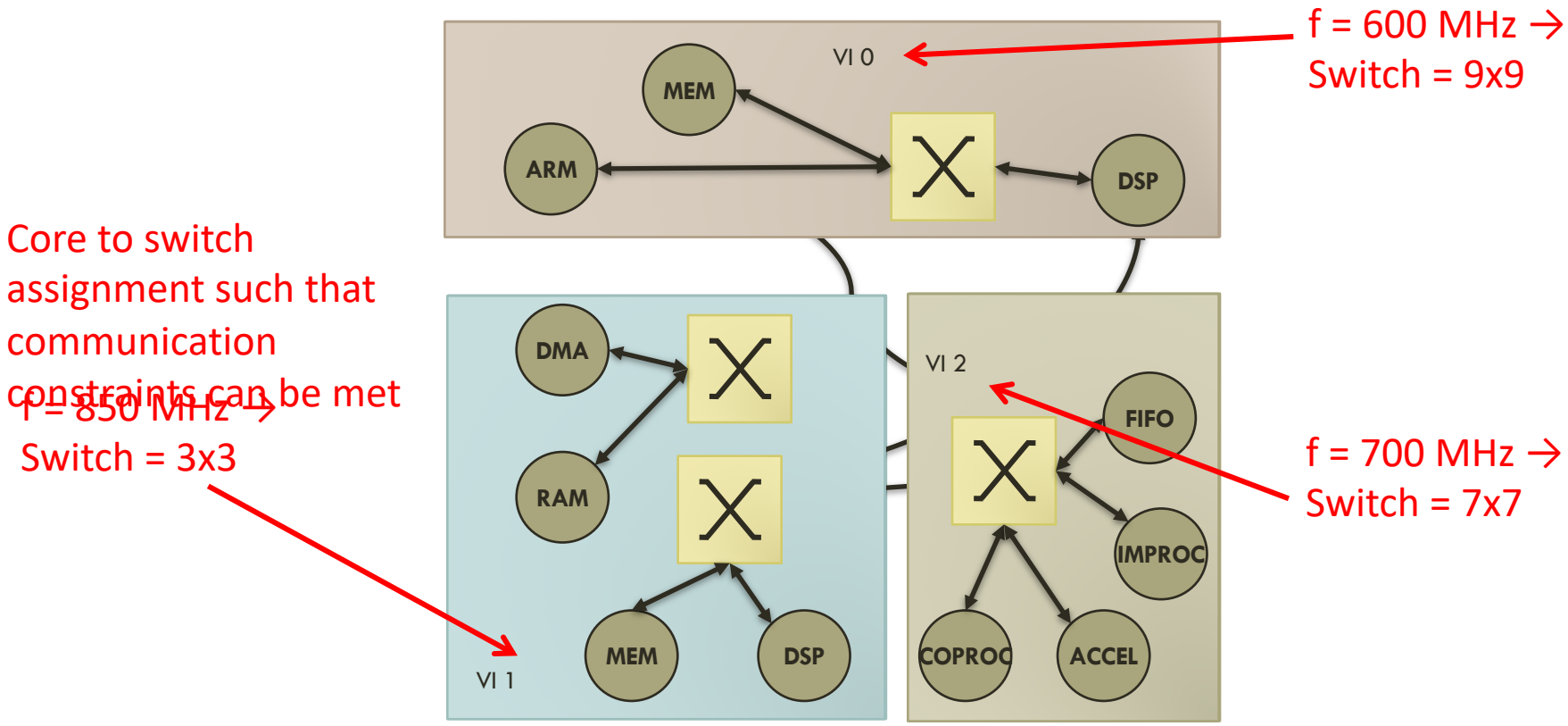
- Sweep grain impacts runtime, exploration thoroughness

EXAMPLE DESIGN POINTS

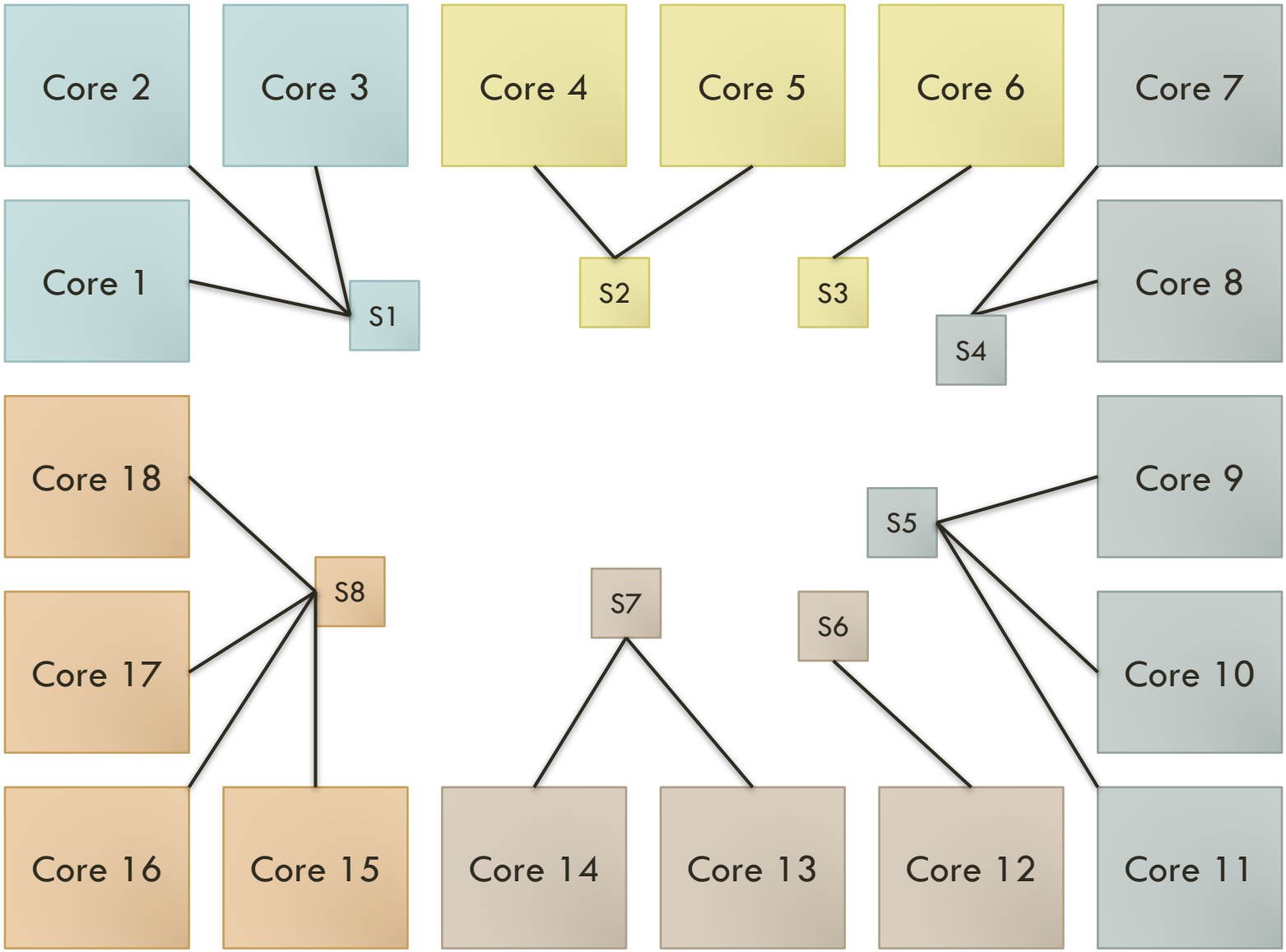


2 - DETERMINE SWITCHES IN V ISLANDS

- Operating frequency bounds maximum switch radix, thus max radix different in each VI



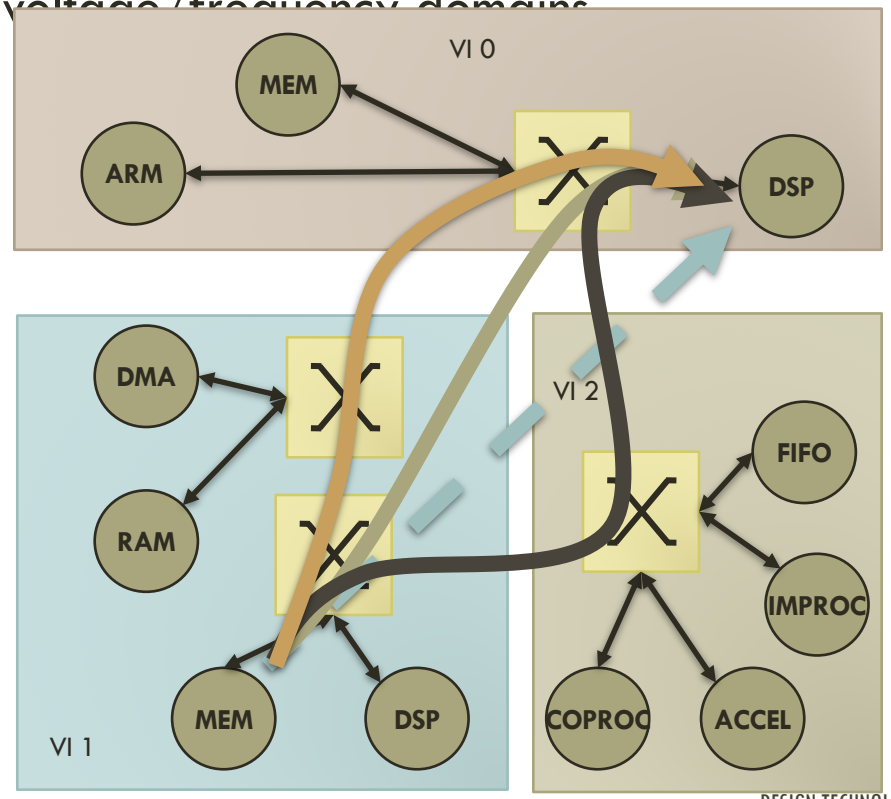
EXAMPLE TOPOLOGY AT END OF STEP 2



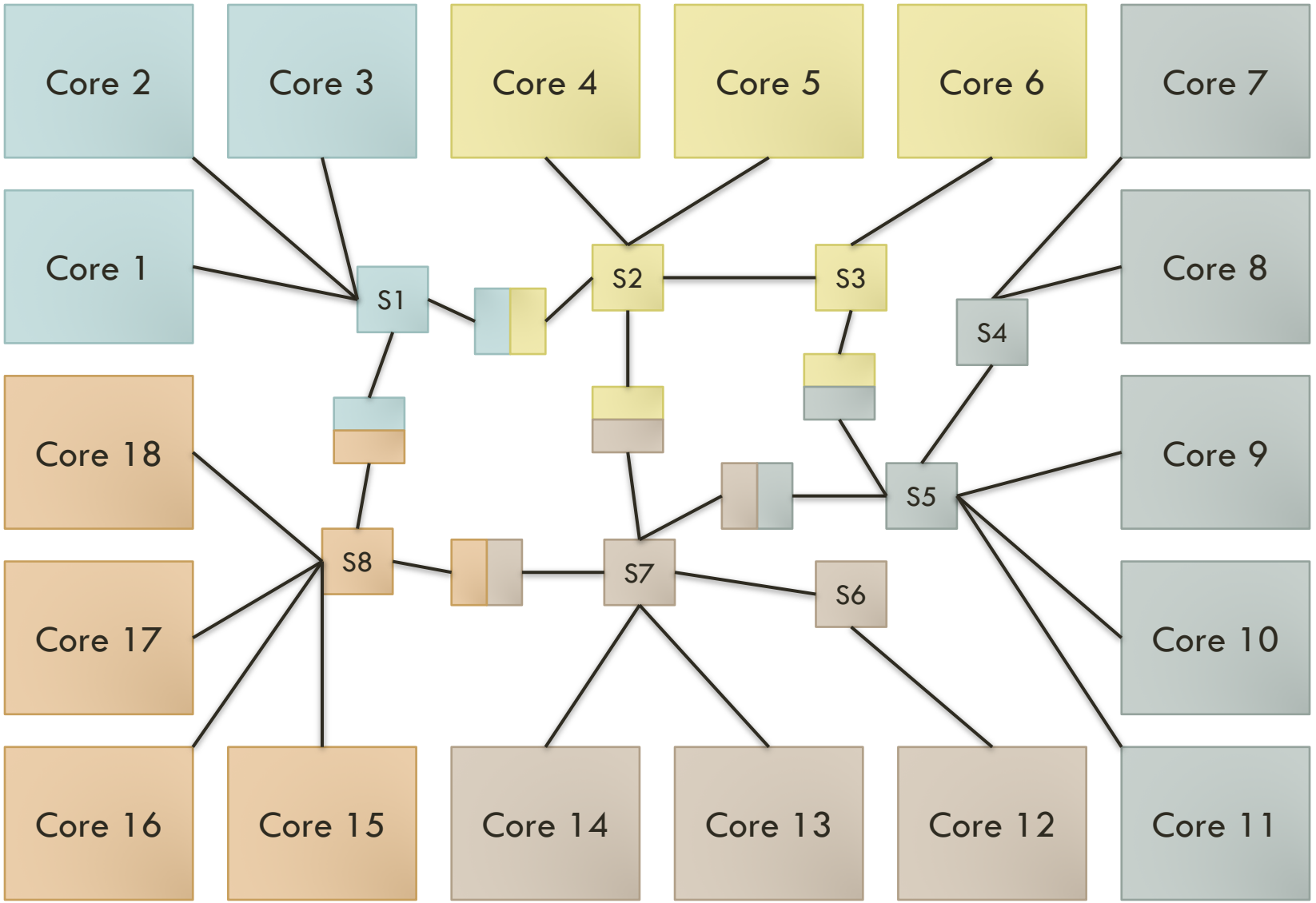
3 – FLOW PATH FINDING

- Establish physical links, consider
  - Cost function: latency, marginal power (to be minimized)
  - Constraints: max switch radix, minimum bandwidth
  - Deadlock freedom
  - Account for voltage/frequency domains

Many paths available !



EXAMPLE TOPOLOGY AT END OF STEP 3



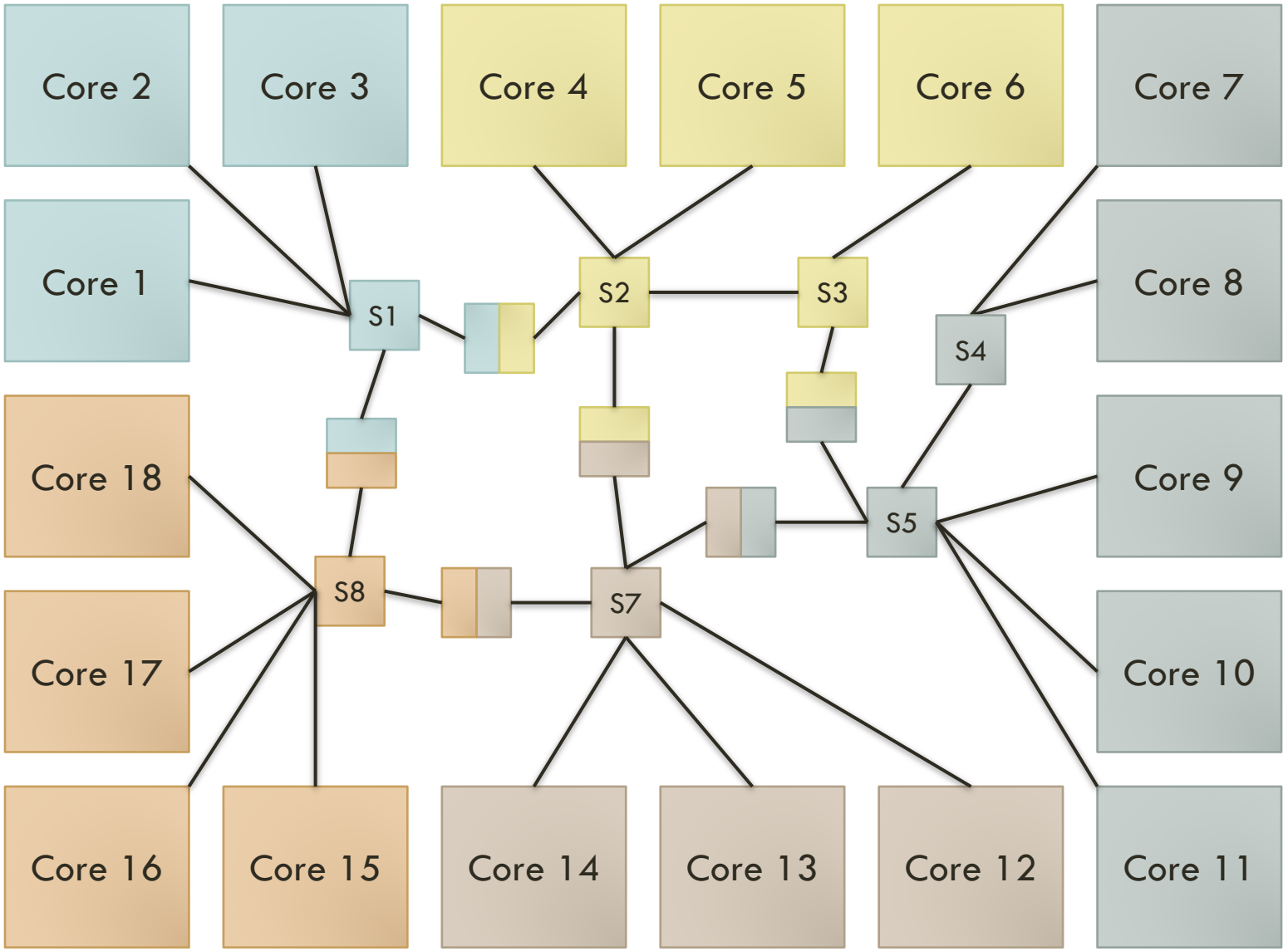
4, 5 – DESIGN POINT FINALIZATION

- Topology is now defined

- Optional optimizations and checks:
  - Trim 1x1 switches
  - Can trim flit width where excessive for requirements
  - Can refine buffer sizing to optimize performance/area/power
  - Can simulate for performance verification and iterate

- Dump topology to disk

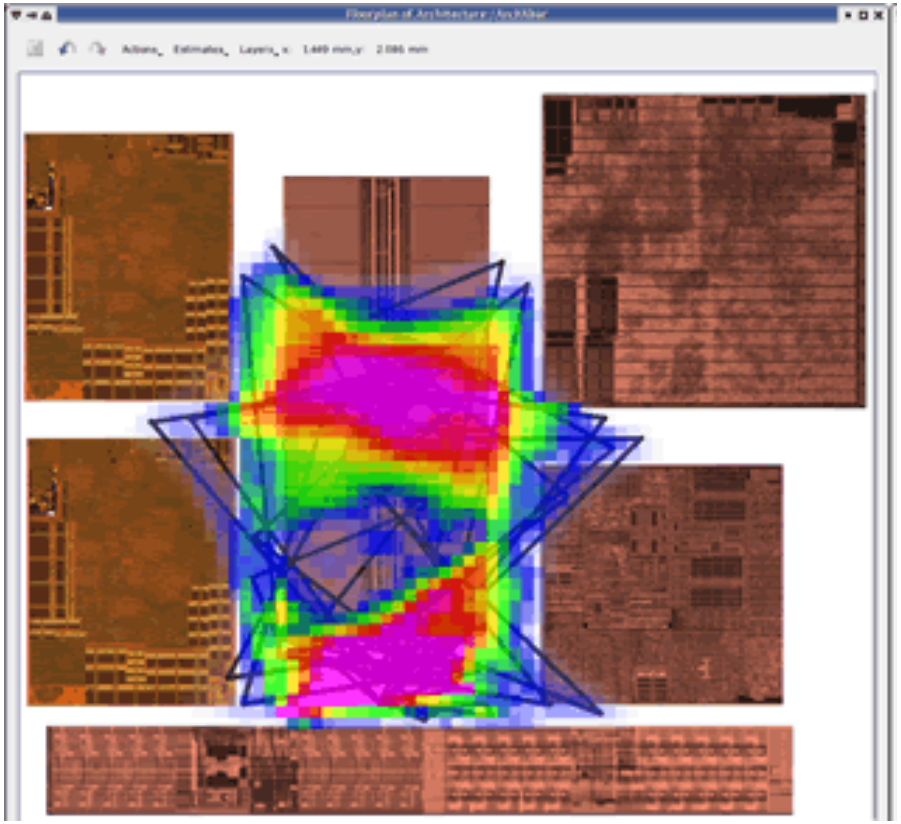
EXAMPLE TOPOLOGY AT END OF STEP 5



NOC FLOORPLANNING

Federico ANGIOLINI
LSI EPFL
2018

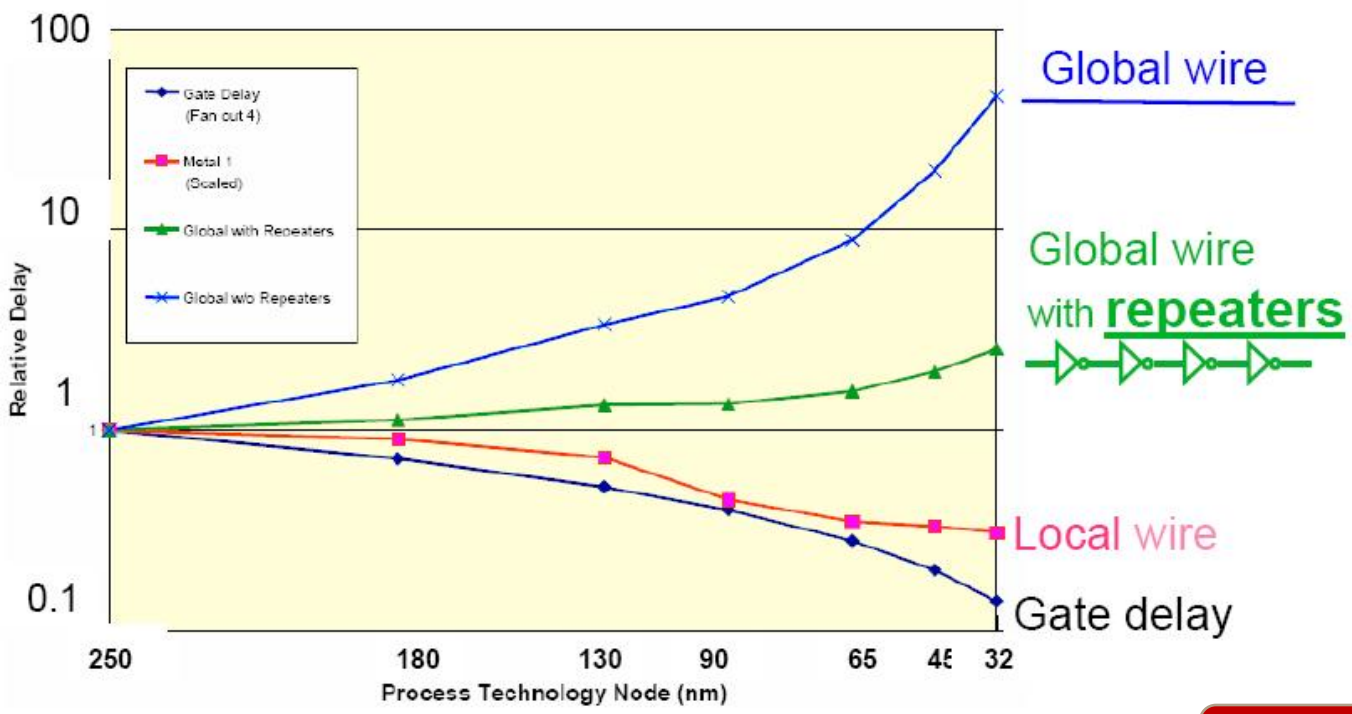
ROUTING CONGESTION IS AN ISSUE



Arteris Whitepaper

WIRING DELAY IS AN ISSUE

Gate delay gets better, wire delay gets worse



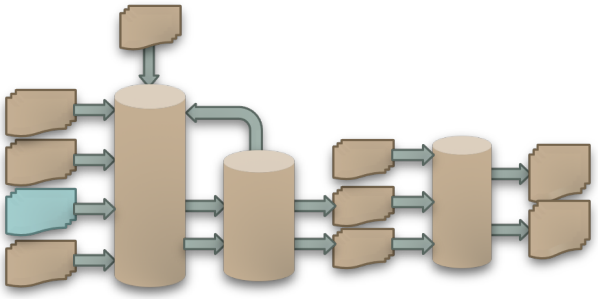
Delay for Metal 1 and Global Wiring versus Feature Size

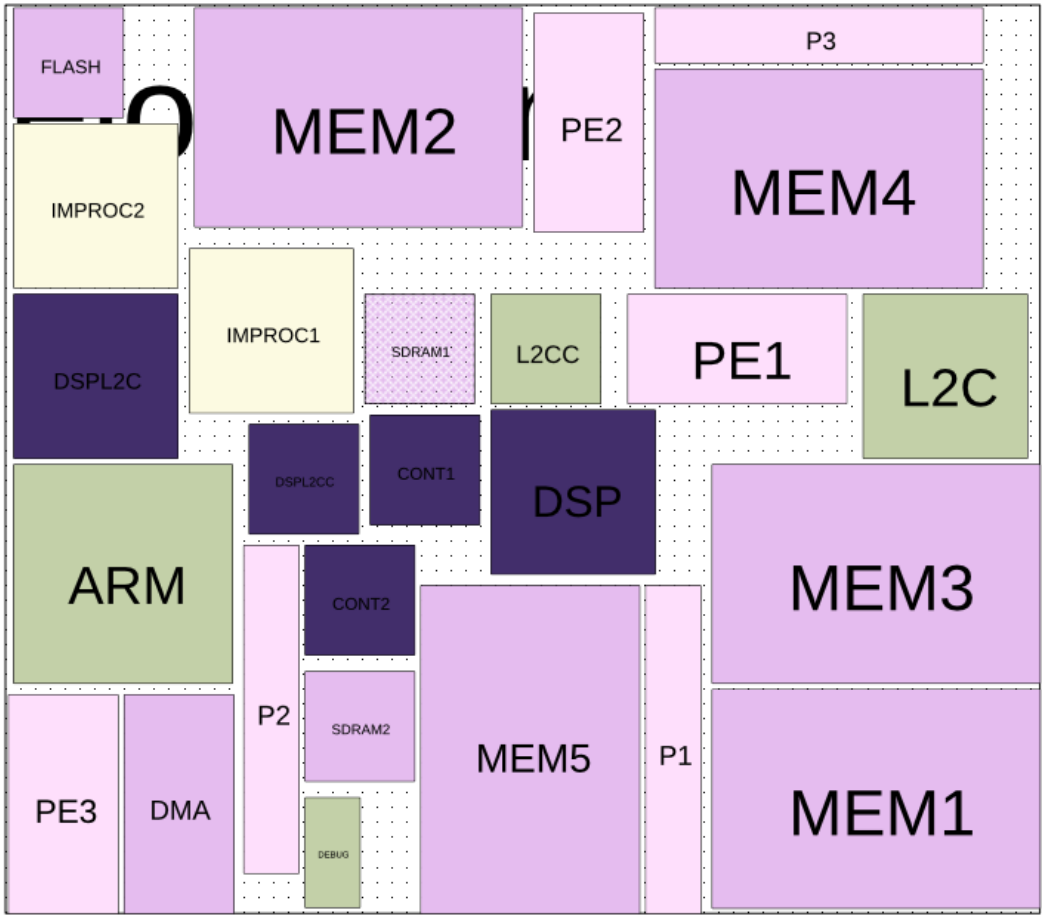
ITRS 2004

TRAVERSABLE WIRE LENGTH IS GOING DOWN

- In 40 nm, at same frequency, wires must be 20-25% shorter than in 65 nm

THEREFORE, FLOORPLAN INPUT IS ESSENTIAL



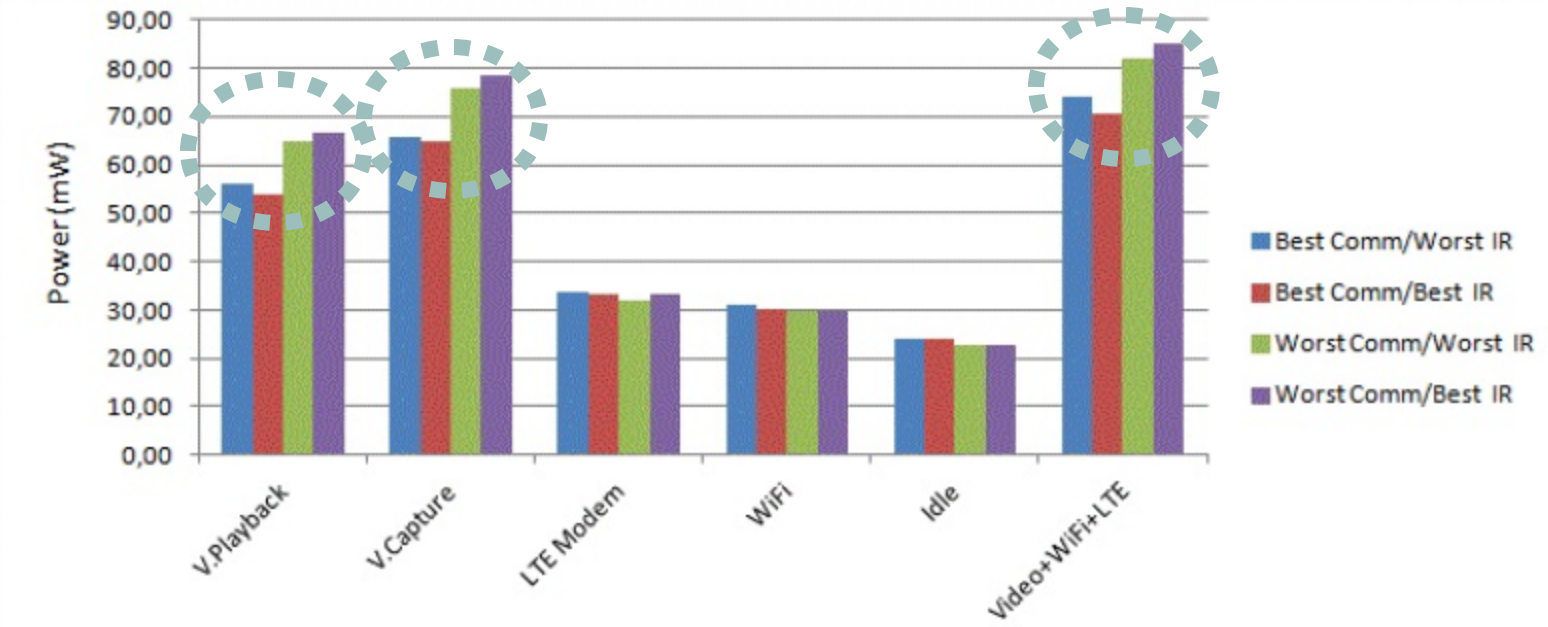


ACCOUNTING FOR FLOORPLAN

- Optional input
- Not to be reshuffled
  - Too many constraints (thermal, pins, analog macros…)
- Insert NoC in the floorplan
  - Figure out ideal NoC component positions
  - Identify links still too long, pipeline them
  - Estimate power according to library model
- Designer can always tweak to taste
- Output floorplan can be exported towards P&R tools

OPTIMIZED FLOORPLAN EFFECT

- Collaboration with Teklatech
- Different 40nm reference floorplans
- NoCs inserted by iNoCs with optimizations
- 15-20% savings in demanding use cases

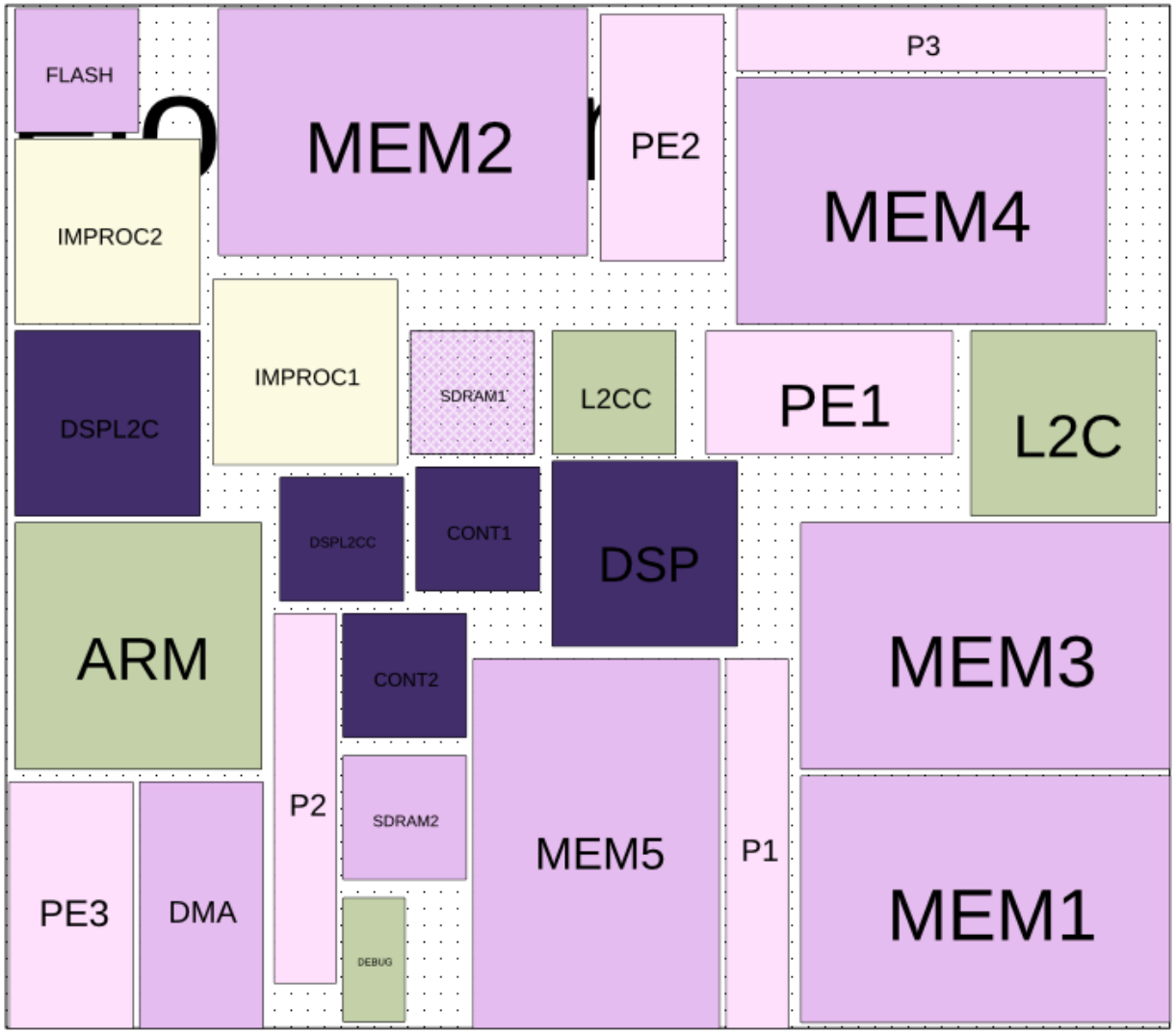


IMPACT ON SYNTHESIS ALGORITHM

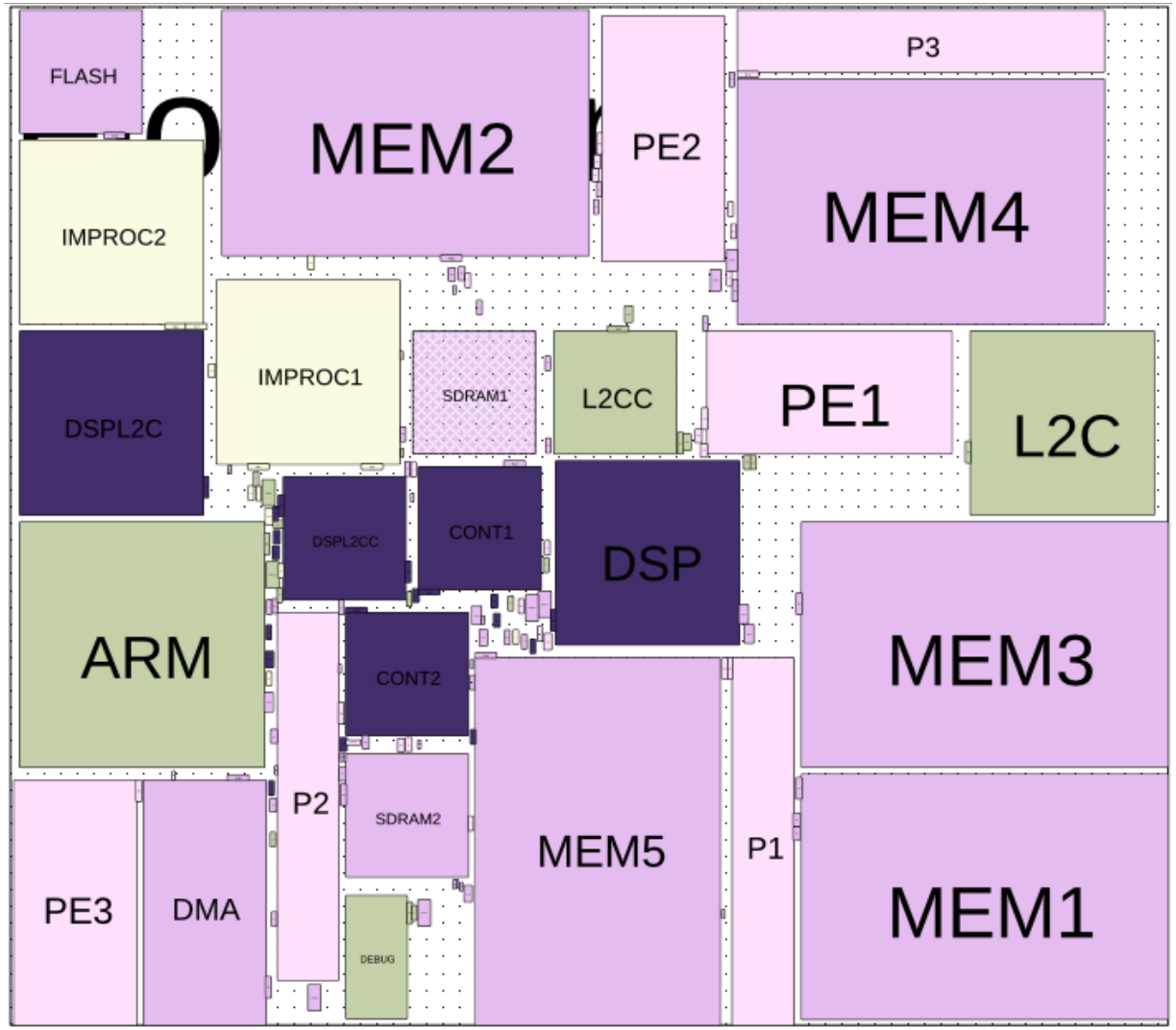
- Floorplan is updated simultaneously with topology synthesis (wires accounted in exploration)

- Two floorplanning steps:
  - **`doInitialFloorplanning()`**: places blocks in ideal positions. Can be iterated and refined during topology building.
  - **`doFloorplanning()`**: moves blocks from ideal positions to suitable locations (empty areas). Can be executed once as final pass.

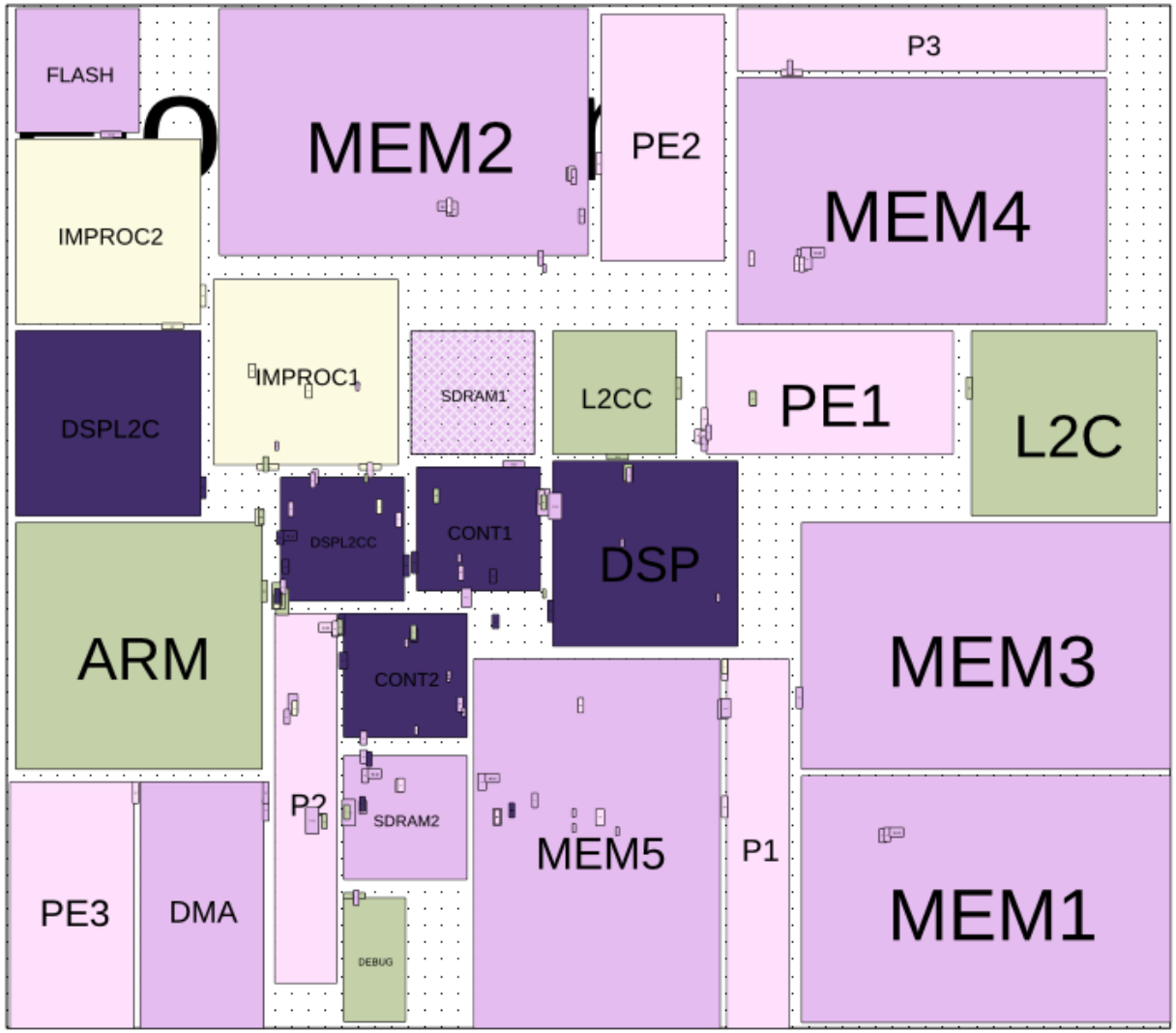
EXAMPLE: INPUT FLOORPLAN



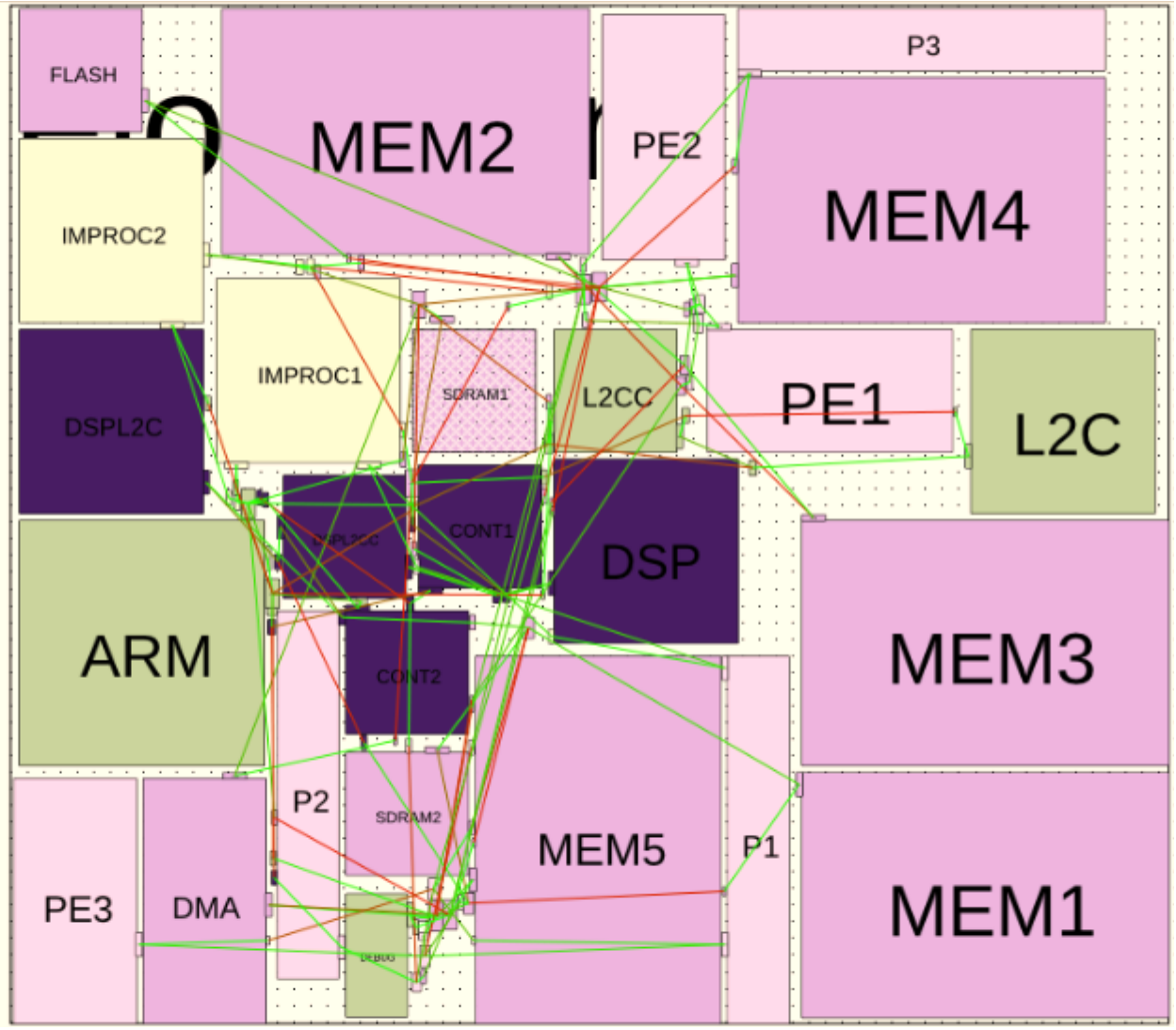
OUTPUT FLOORPLAN (NO OVERLAPS)



OUTPUT FLOORPLAN (OVERLAPS)

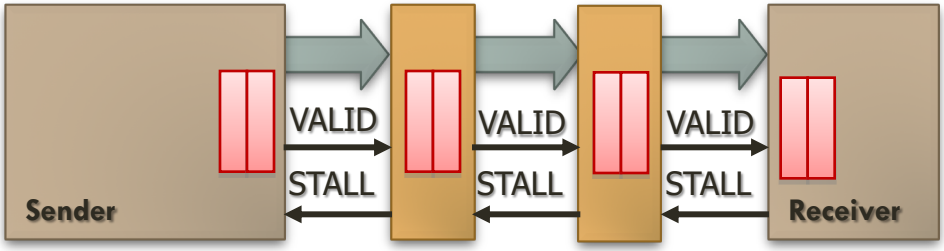


WIRE LENGTH ESTIMATION



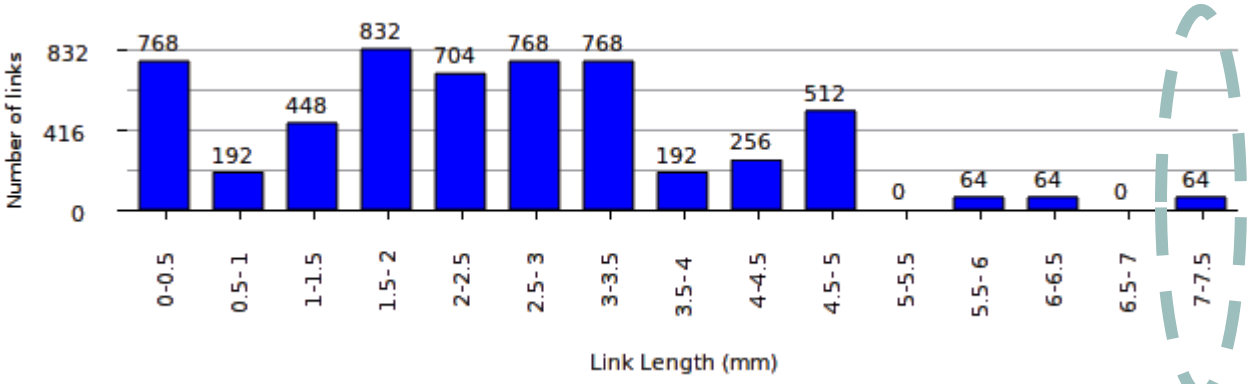
LINK PIPELINING

- Wire segmentation by topology design
  - Put more switches, or place them closer
- Wire segmentation by pipeline insertion
  - Flops/relay stations to break links

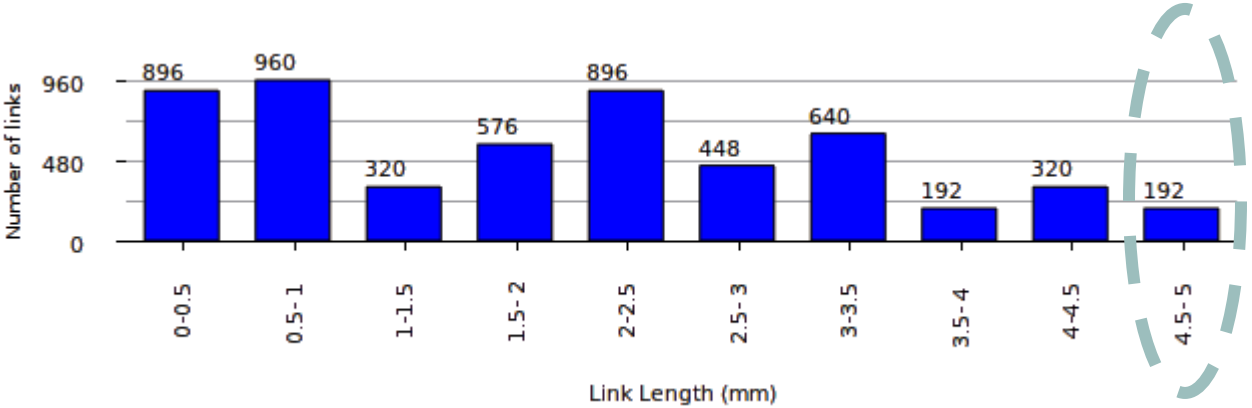


- Take advantage of pre-existing converters
  - If e.g. frequency converter is instantiated along link, space it evenly to help with segmenting

EXAMPLE: LINK LENGTH DISTRIBUTION



Original benchmark
max f = 400 MHz



Overclocked benchmark
max f = 800 MHz

CONCLUSIONS

Federico ANGIOLINI
LSI EPFL
2018

CONCLUSIONS

- IP-based design becoming prevalent
  - CMPs, MPSoCs
- System interconnect is evolving towards NoCs
  - Scalable performance, better physical design properties
- EDA tooling needed to help with NoC design
  - Devise best interconnect from high-level specifications
  - Accounting for floorplan is important