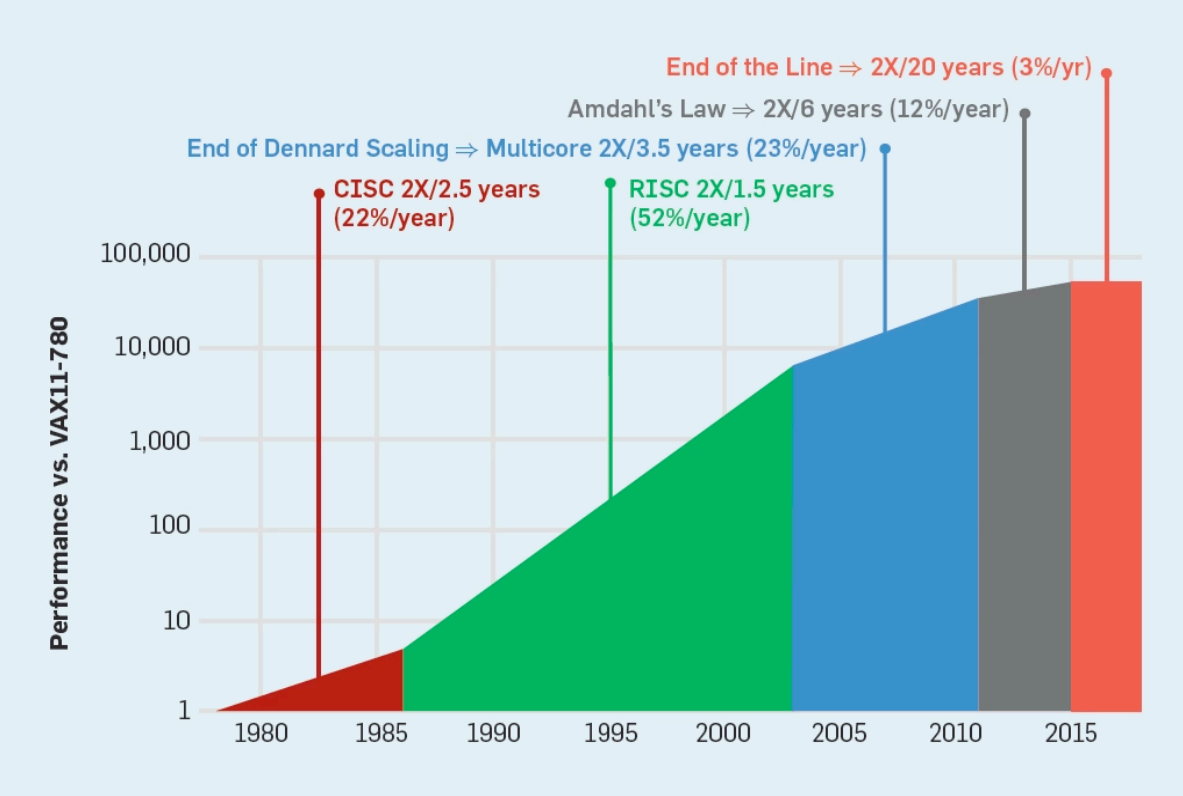
A Scalable Methodology for Agile Chip Development with Open-Source Hardware Components

Maico Cassel dos Santos[1,*], Tianyu Jia[2,*], Martin Cochet[3], Karthik Swaminathan[3], Joseph Zuckerman[1], Paolo Mantovani[1], Davide Giri[1], Jeff Jun Zhang[2], Erik Jens Loscalzo[1], Gabriele Tombesi[1], Kevin Tien[3], Nandhini Chandramoorthy[3], John-David Wellman[3], David Brooks[2], Gu-Yeon Wei[2], Kenneth Shepard[1], Luca Carloni[1], Pradip Bose[3]

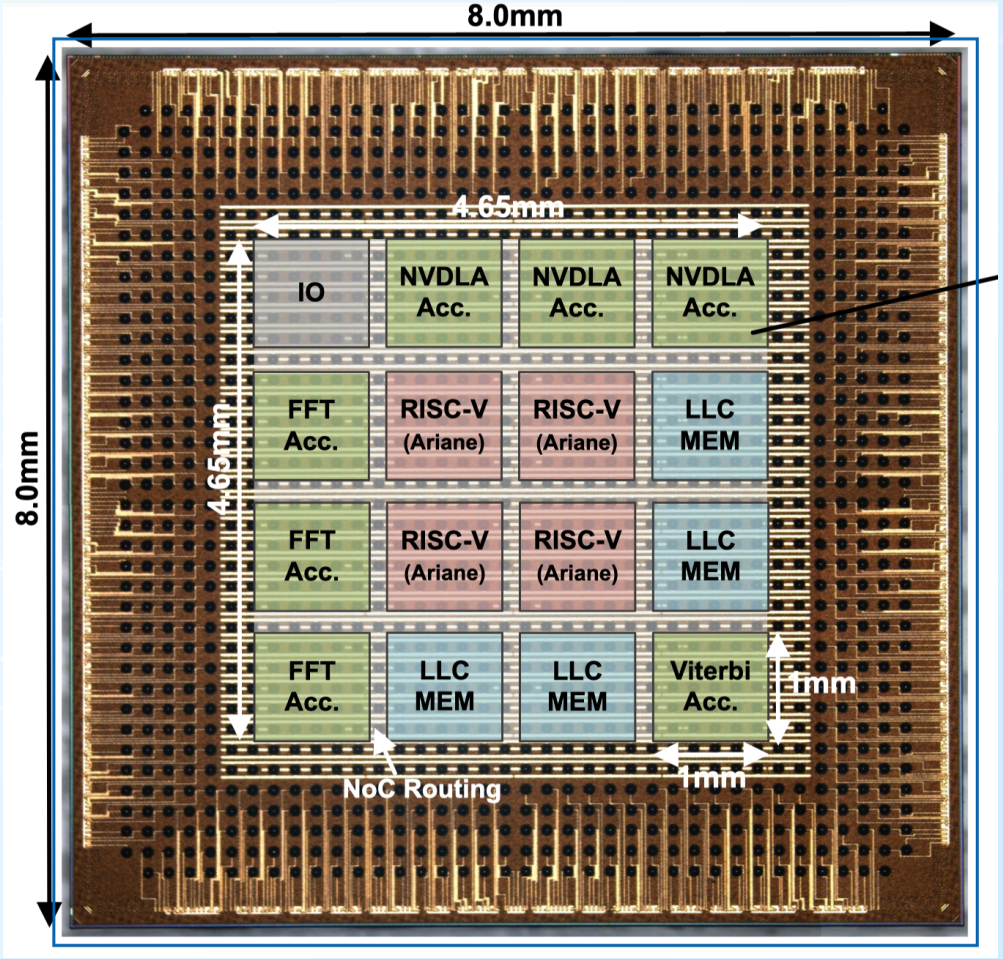
* This authors have equal contributions

[1]Columbia University [2]Harvard University [3]IBM Research

Technology limitations and heterogeneity

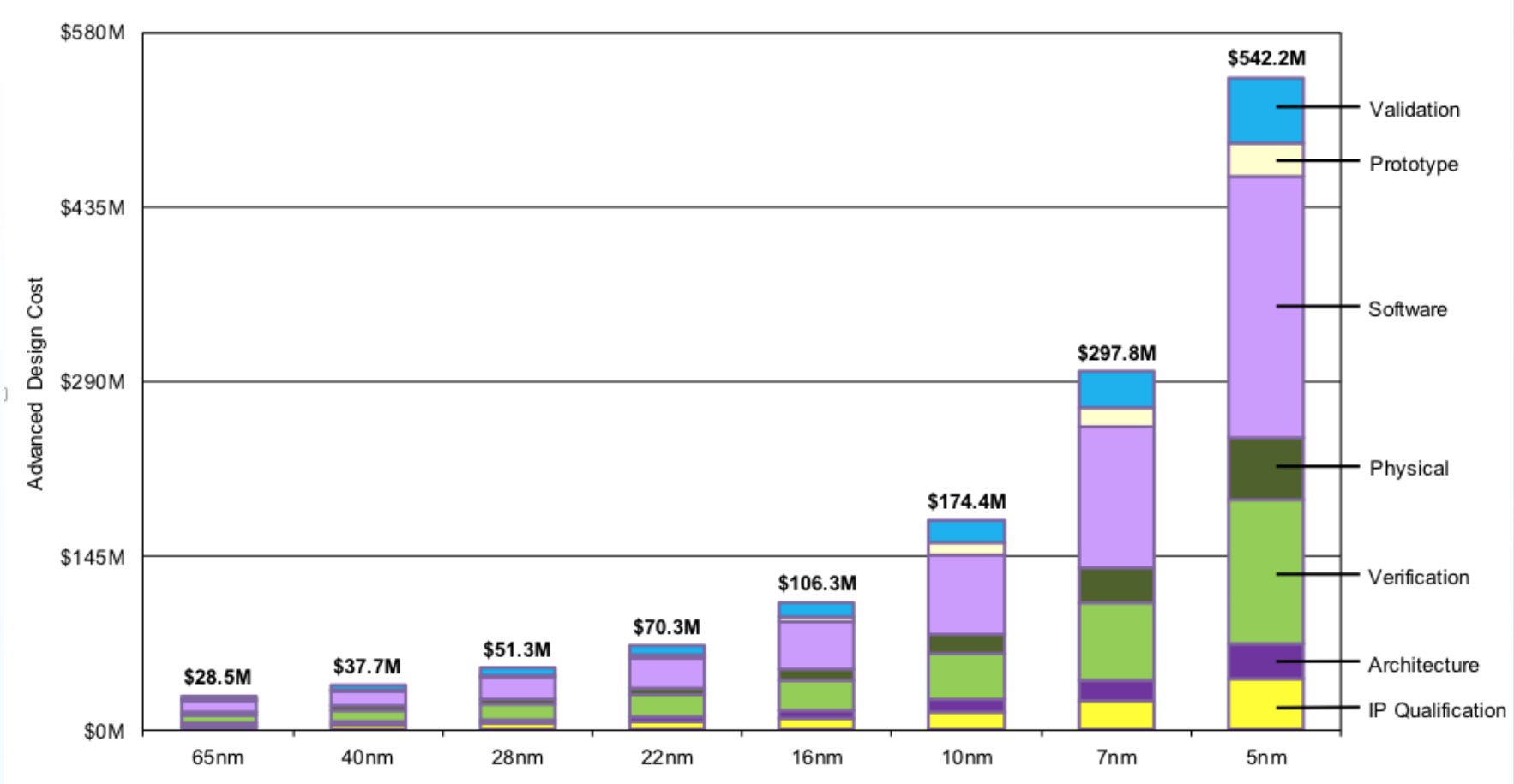


John L. Hennessy, David A. Patterson, Communications of the ACM, February 2019



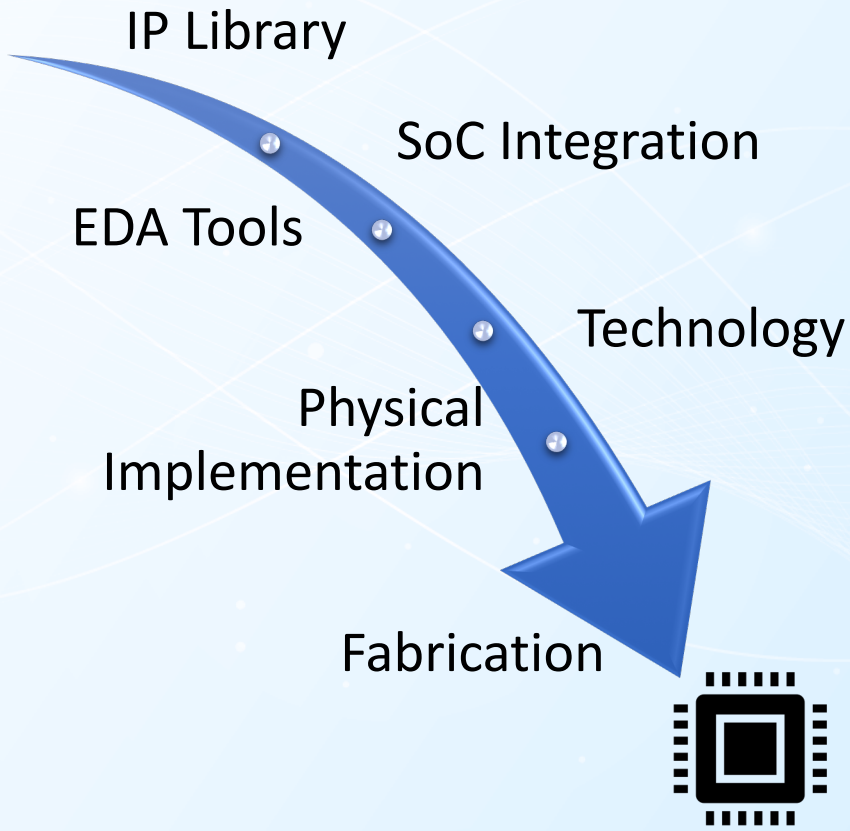
The EPOCHS-0 Heterogeneous Multi-core SoC

The need for design-reuse



IBS 2018

A light at the end of the tunnel?



Physical design flow key properties:

Flexibility

Smooth integration of OSH

Adaptable to different technologies and EDA tools

Robustness

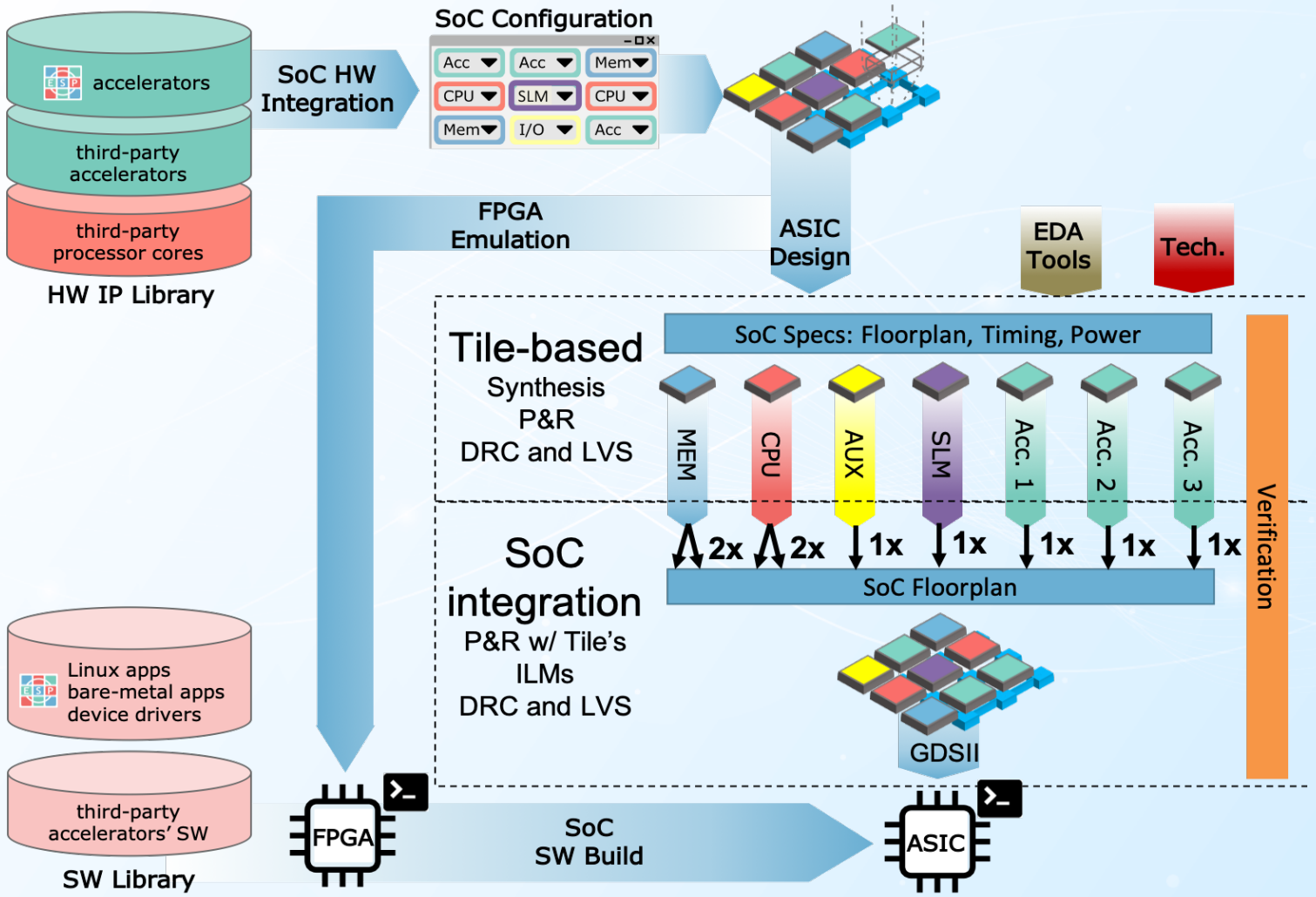
Achieves QoR metrics

Verifies correctness in all implementation stages

Scalability

Handles growth in size and complexity with a sublinear growth in computation infrastructure, engineering effort and design time

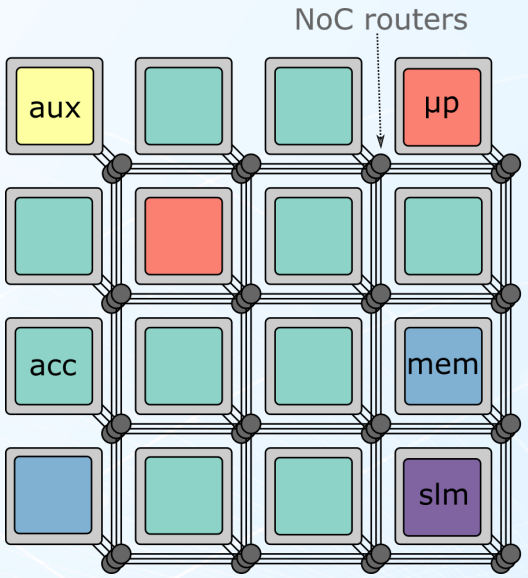
Our methodology for agile chip development



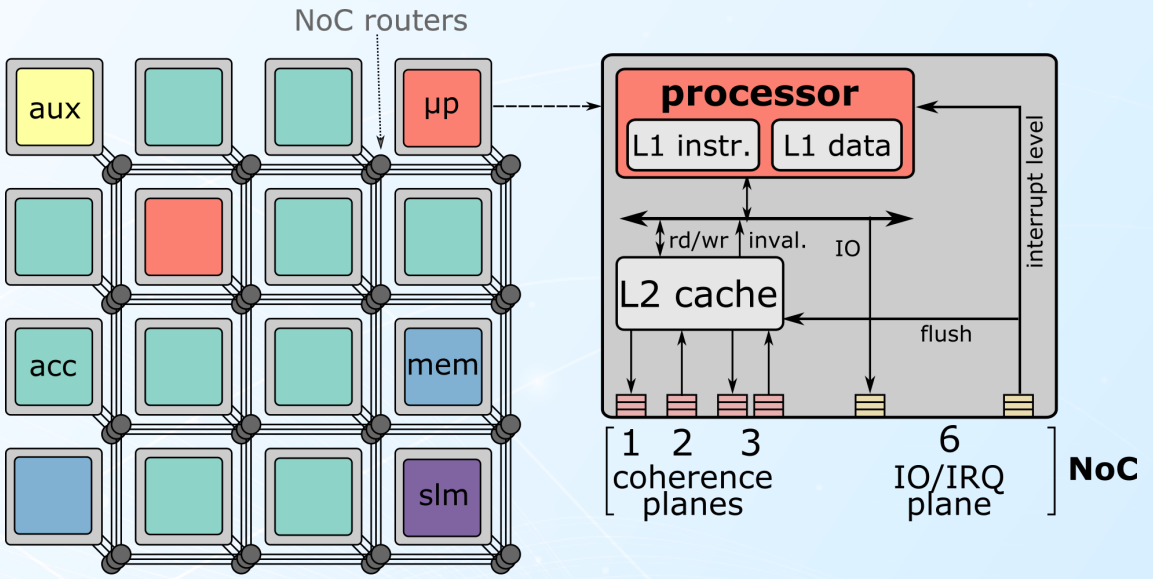
Outline

- ESP: background and enhancements for ASIC design
- Tile-based physical design flow
- SoC physical integration
- Verification and testing
- Experience with two chip designs
- Conclusion

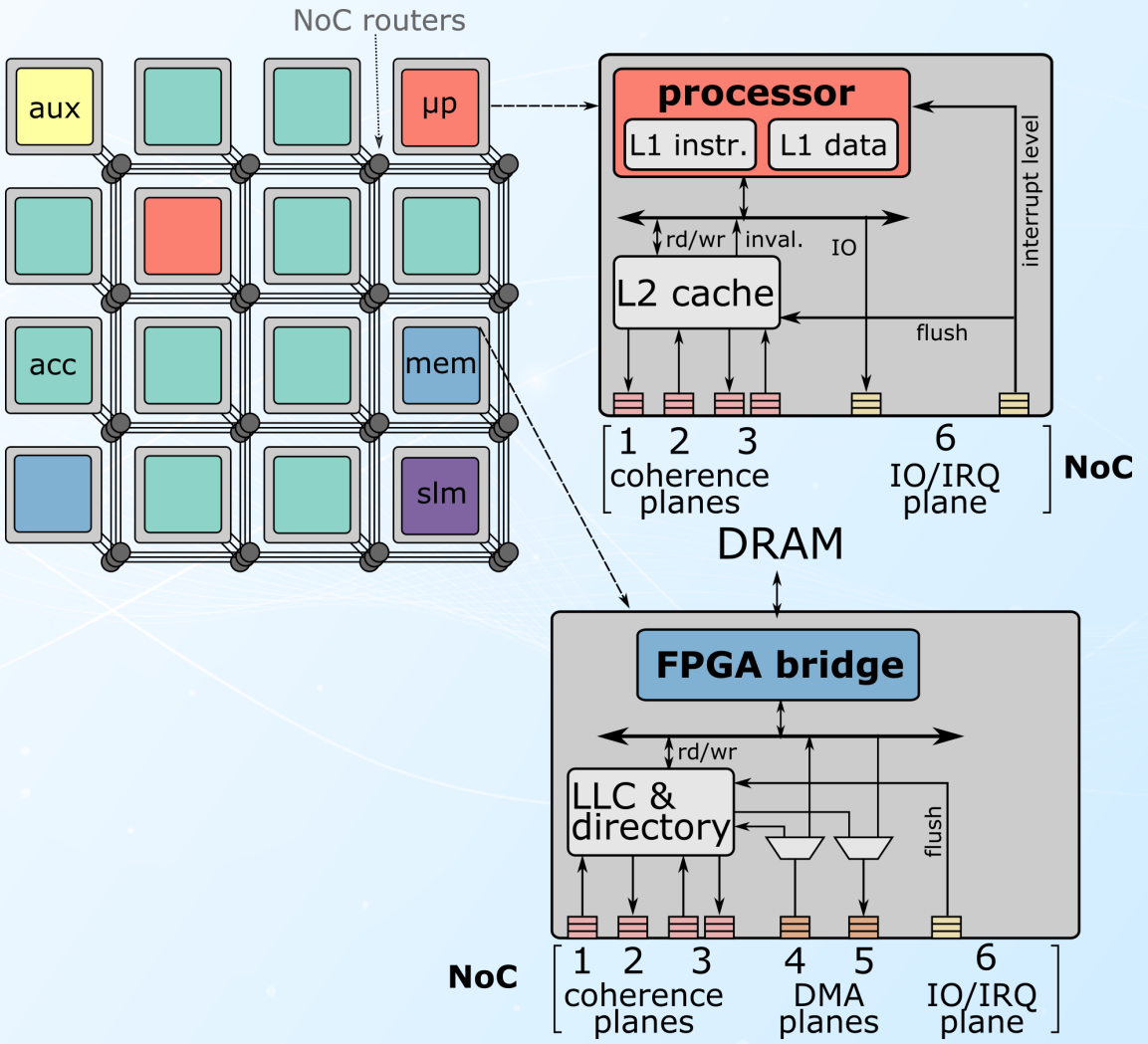
ESP background



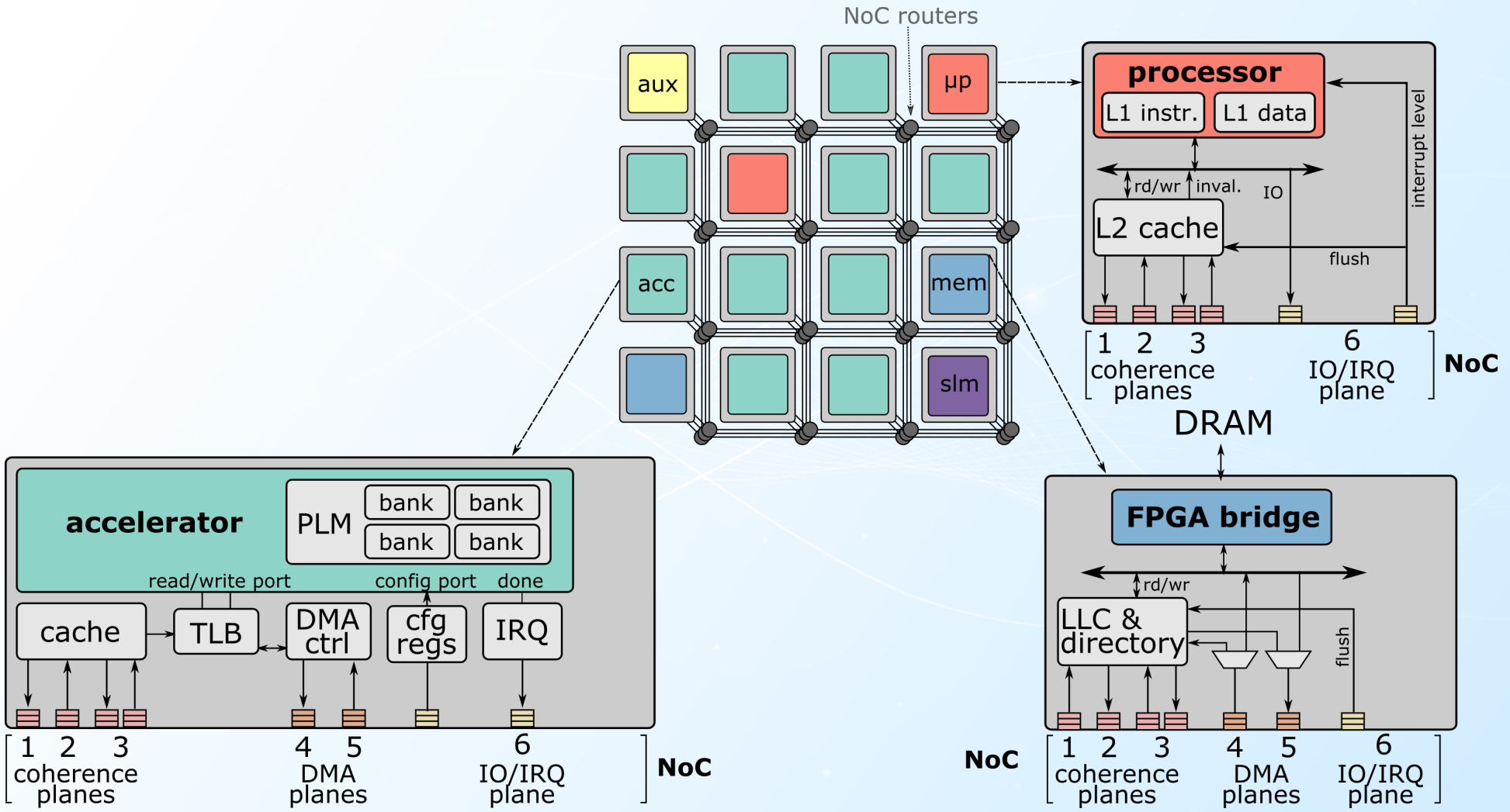
ESP background



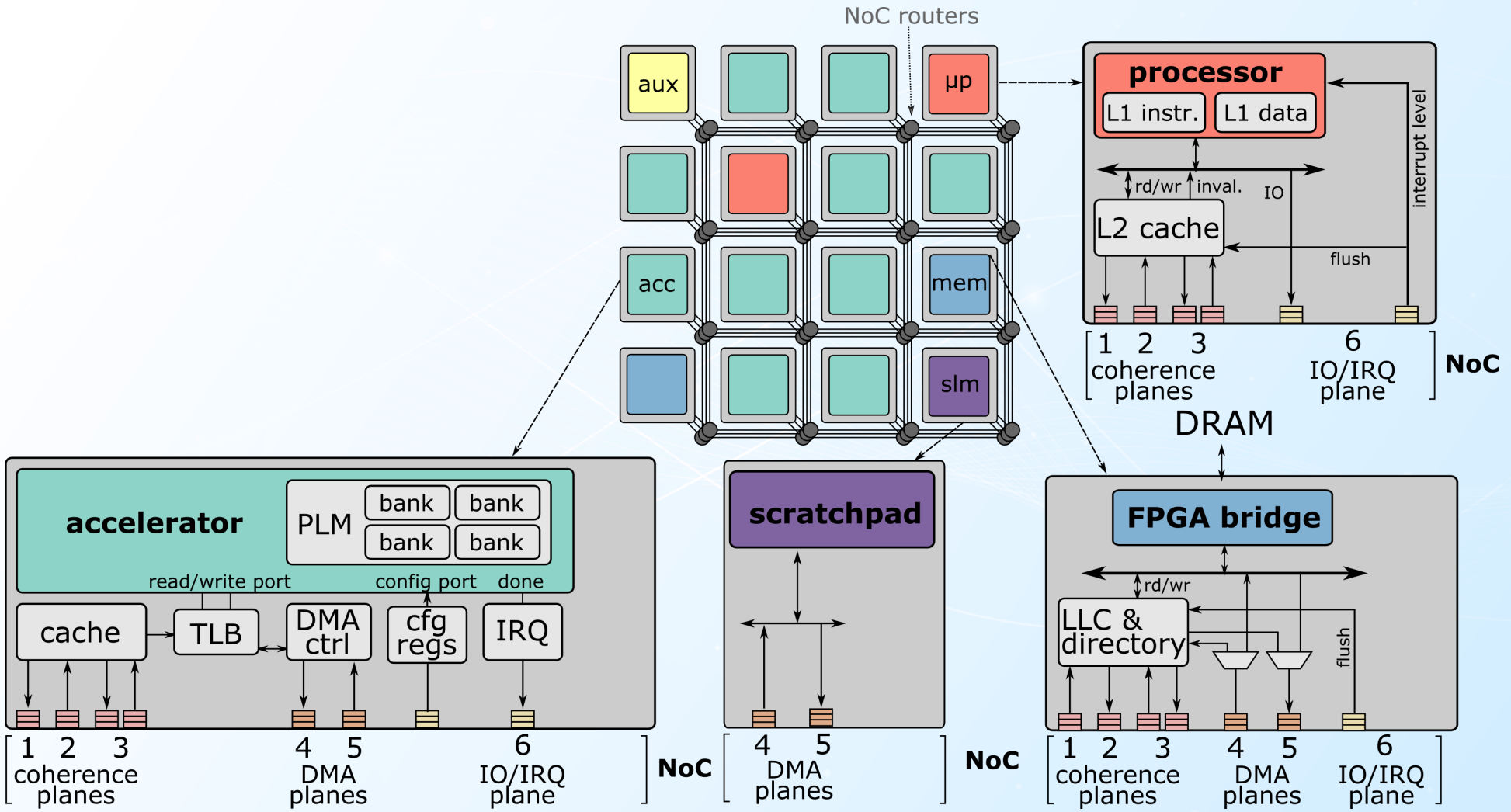
ESP background



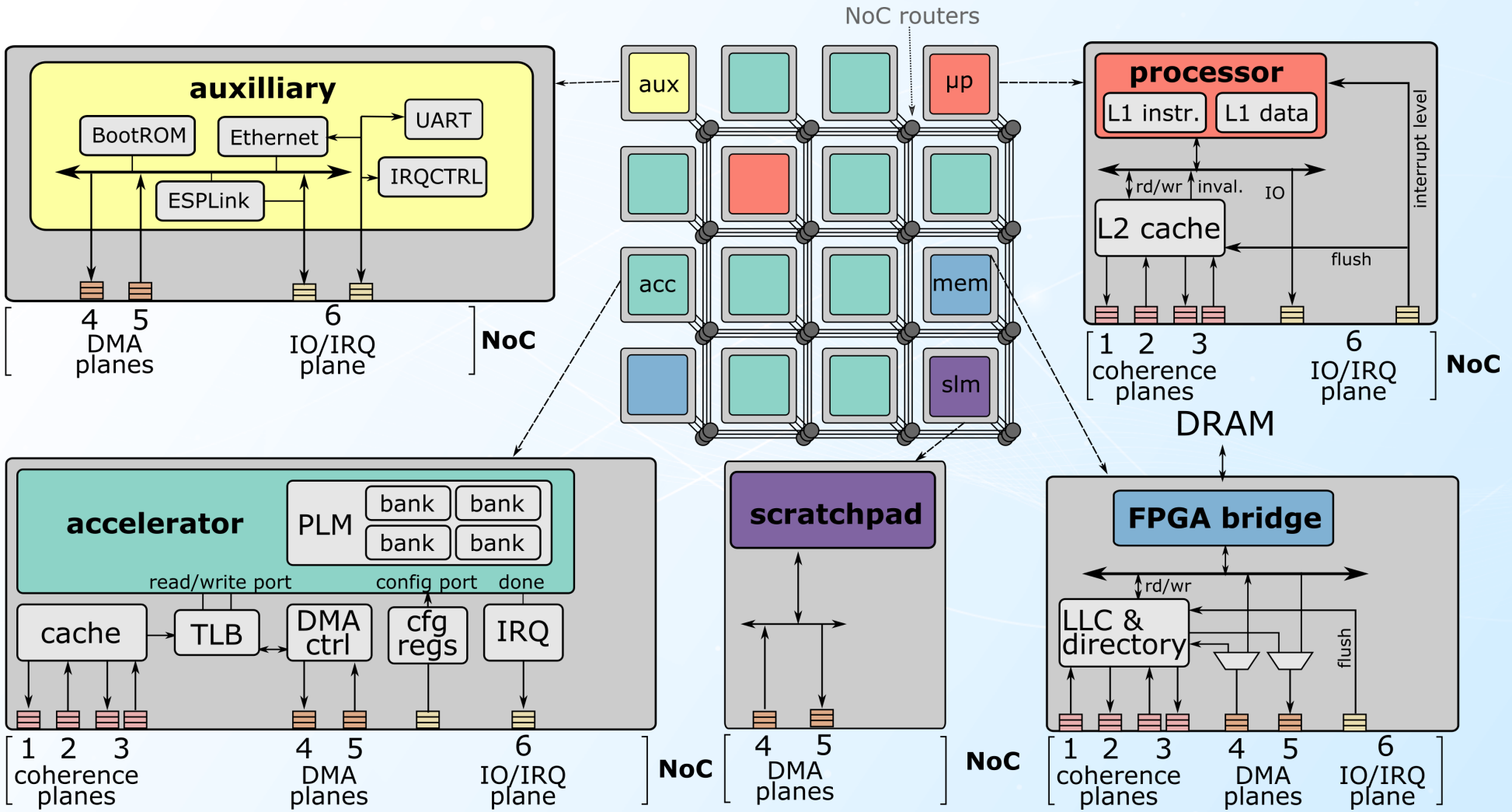
ESP background



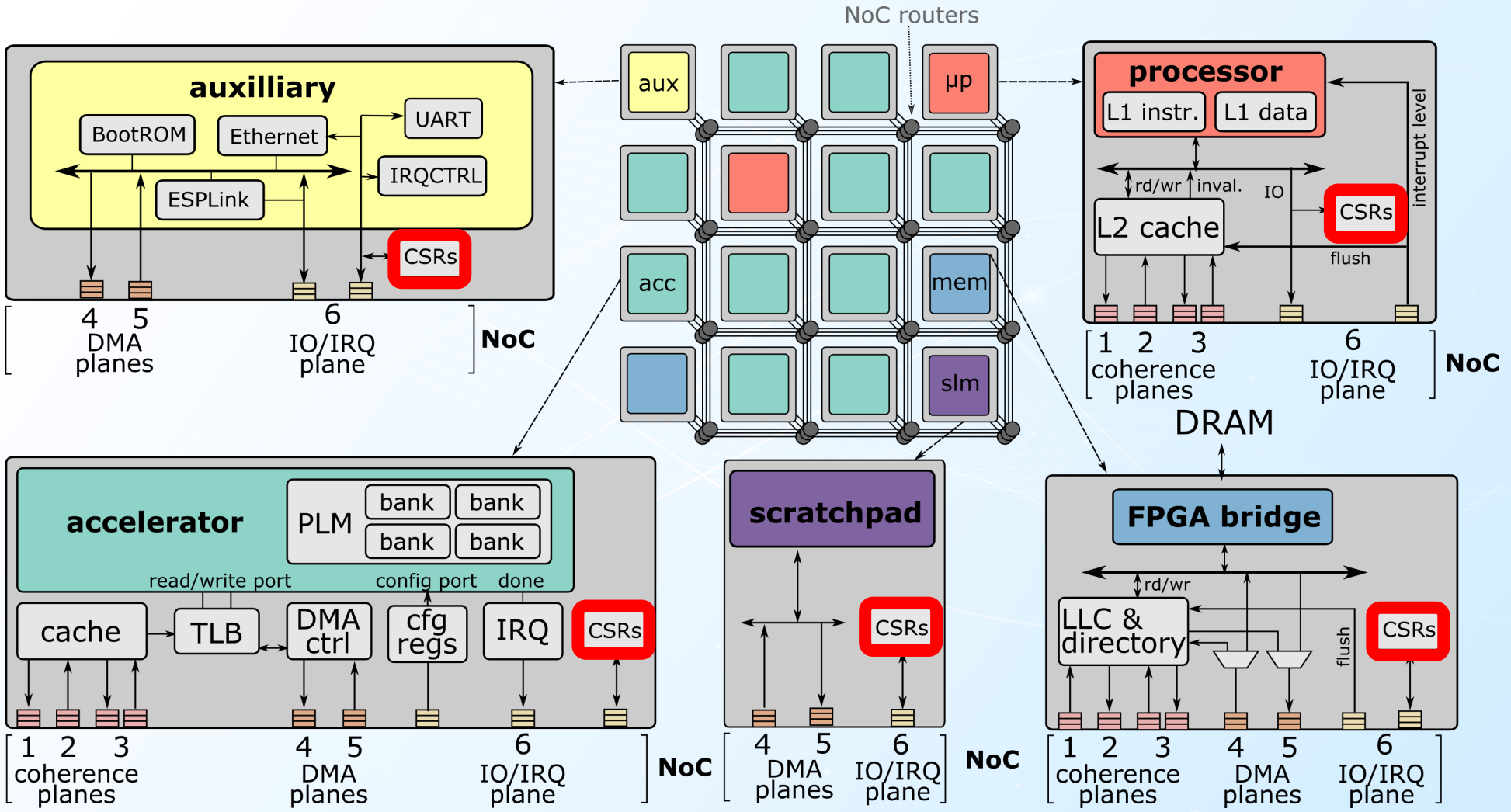
ESP background



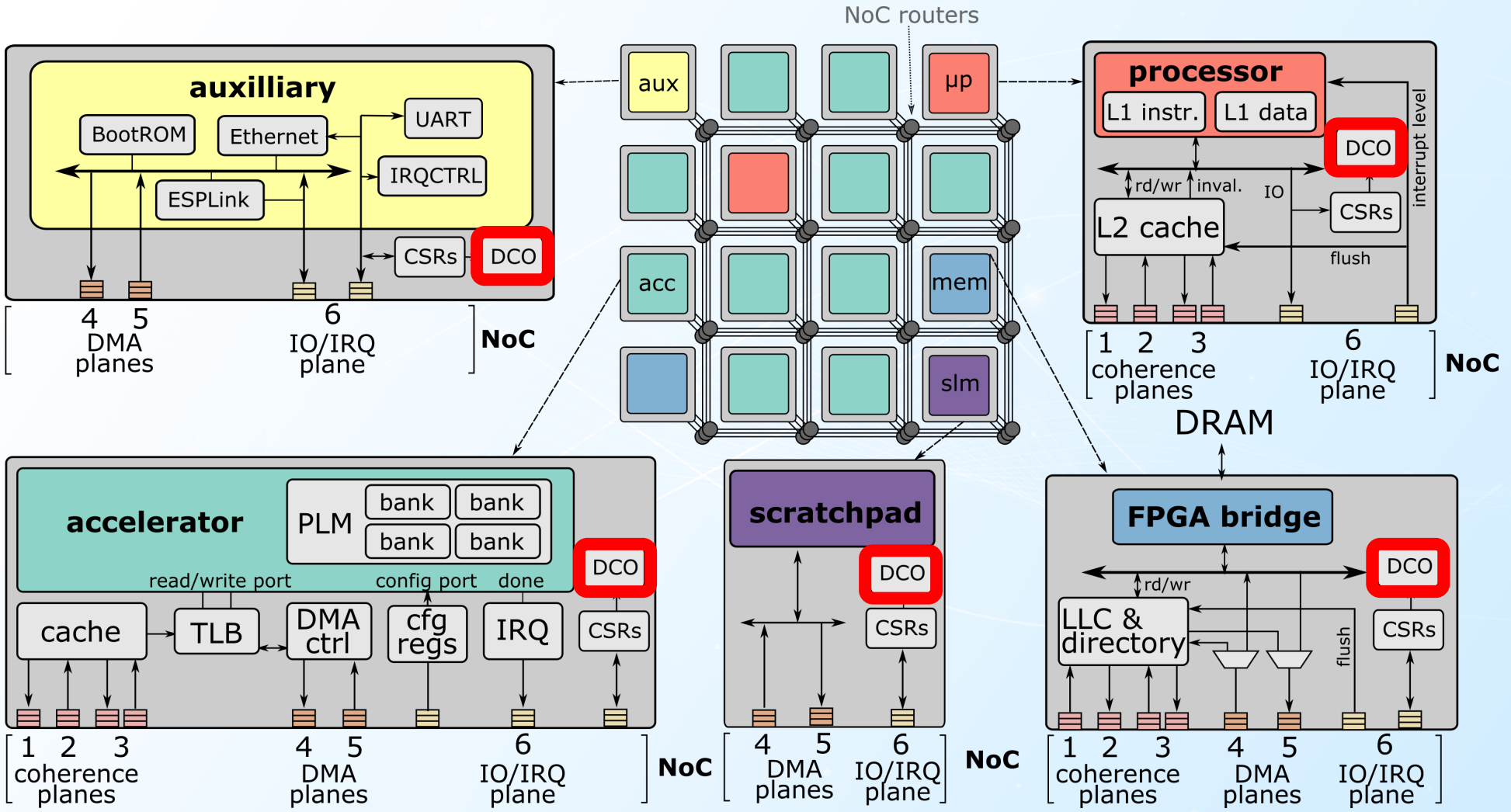
ESP background



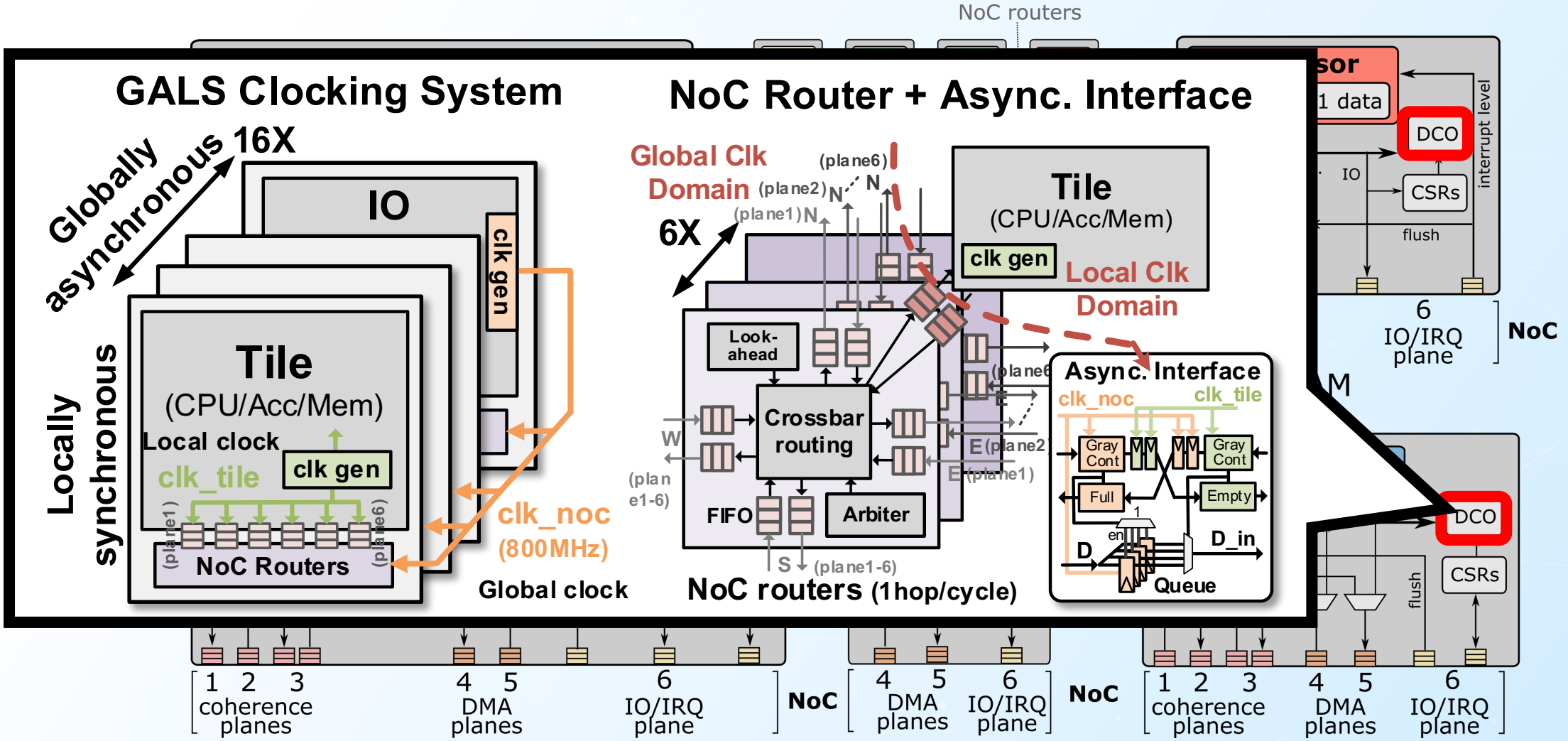
ESP enhancements for ASIC design



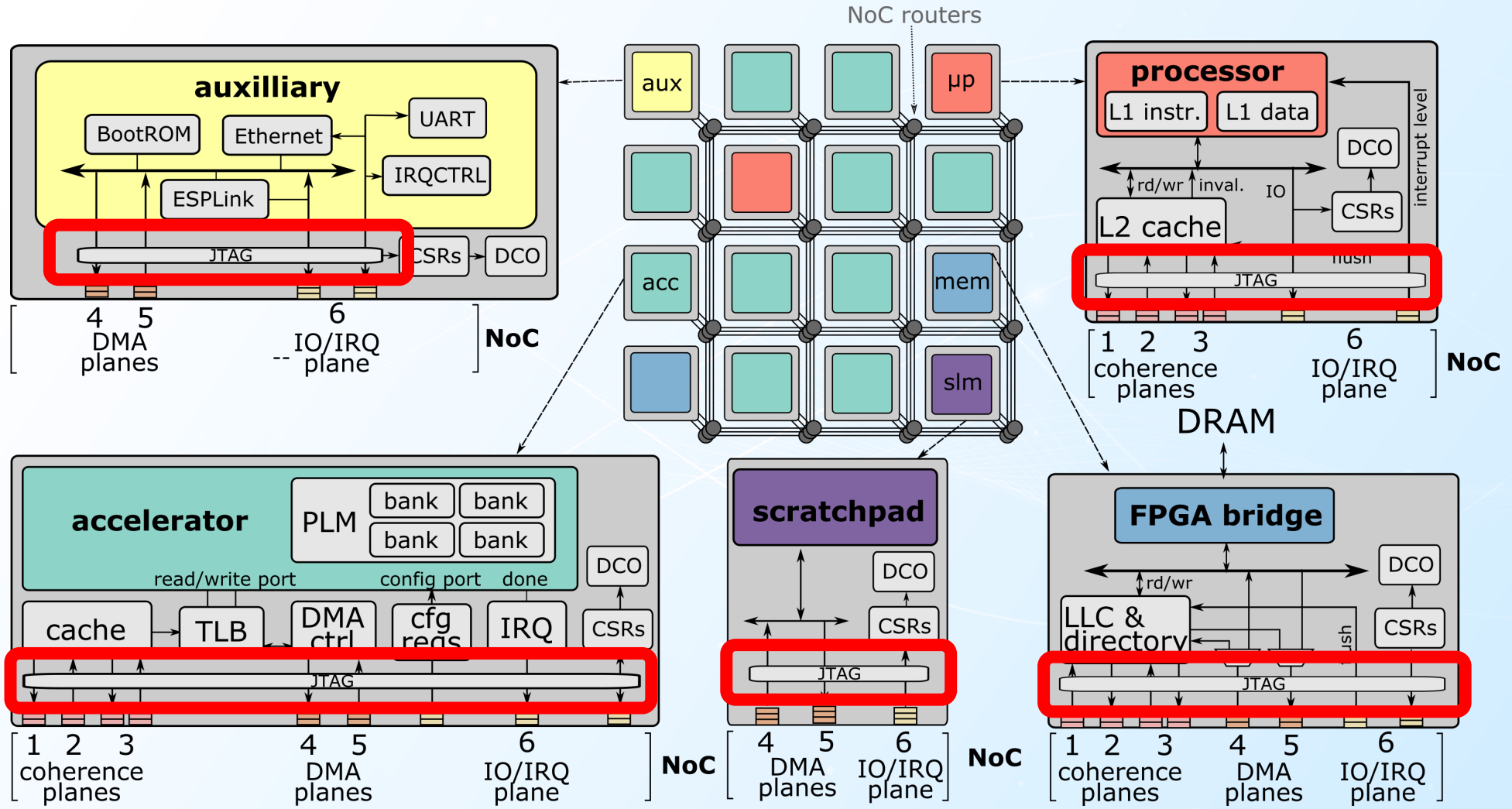
ESP enhancements for ASIC design



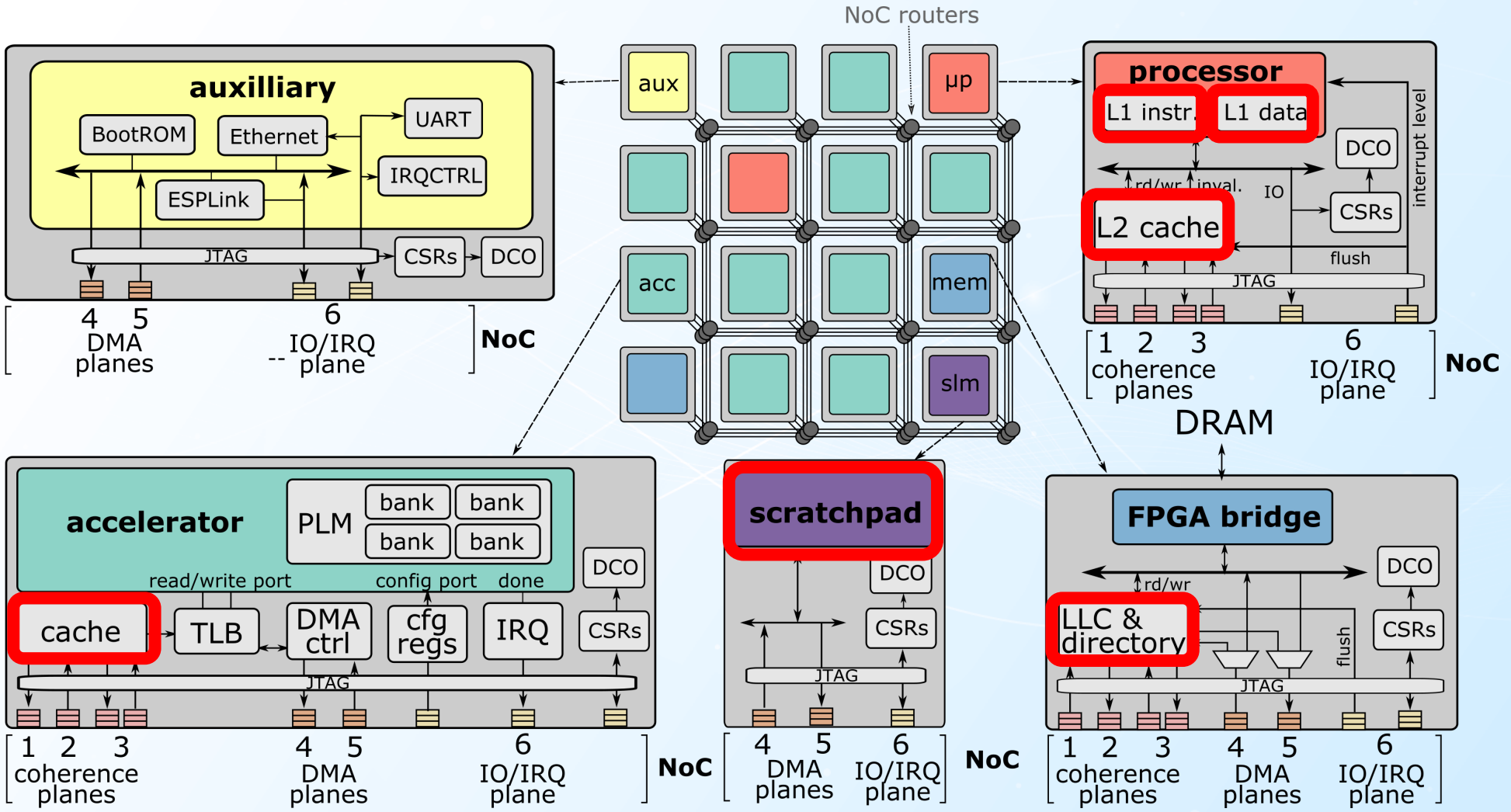
ESP enhancements for ASIC design



ESP enhancements for ASIC design



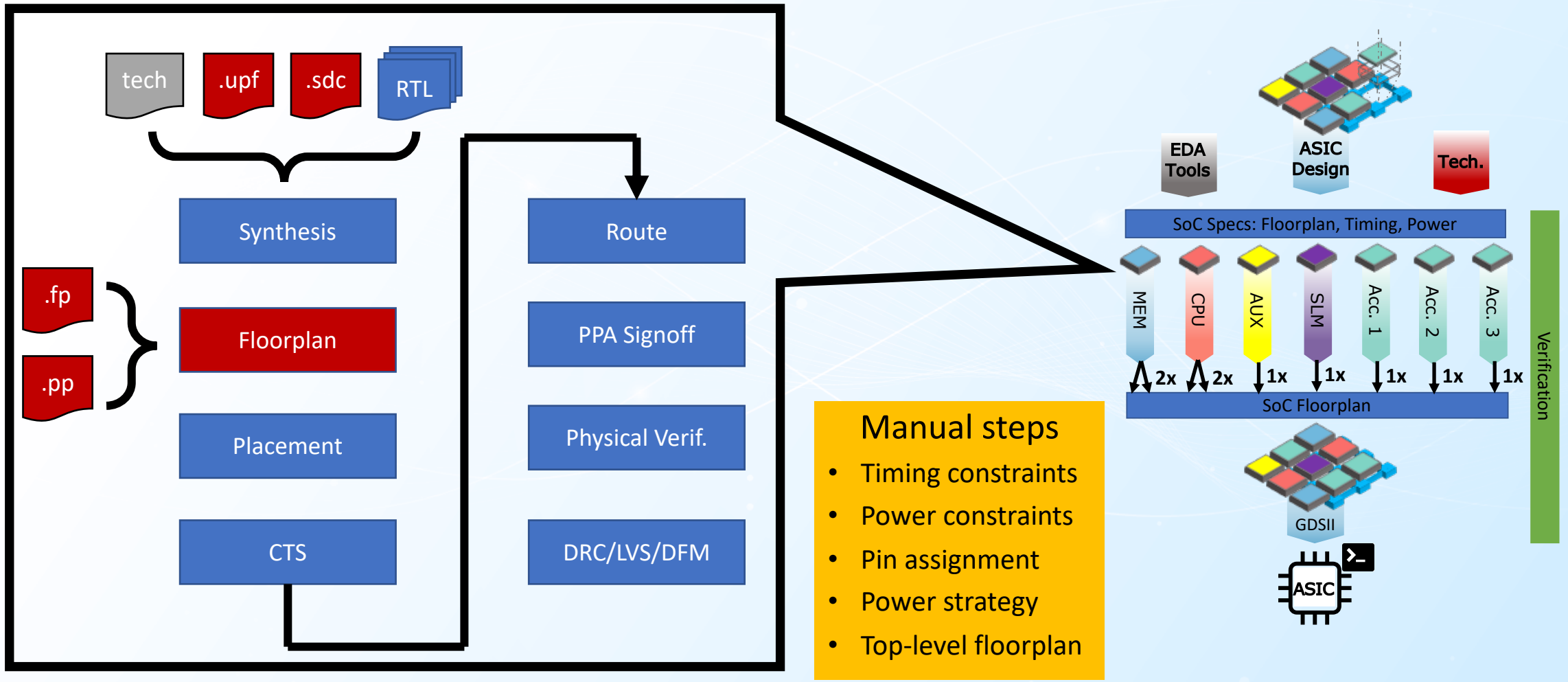
ESP enhancements for ASIC design



Outline

- ESP: background and enhancements for ASIC design
- Tile-based physical design flow
- SoC physical integration
- Verification and testing
- Experience with two chip designs
- Conclusion

Tile-based physical design flow

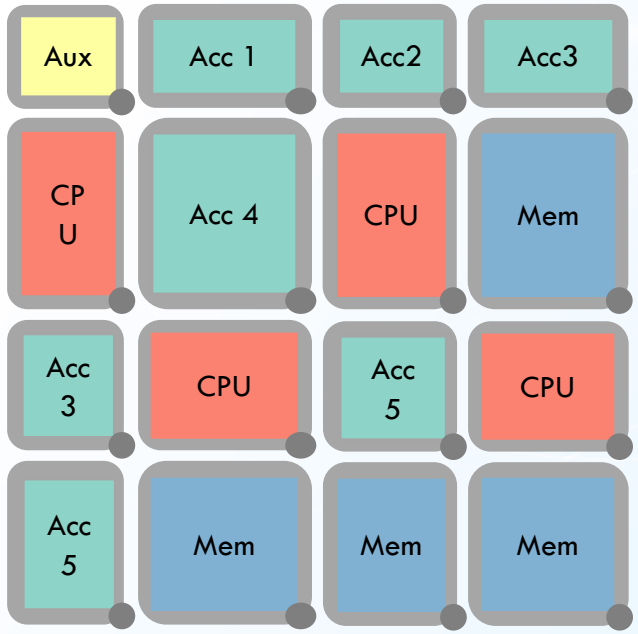


Manual steps

- Timing constraints
- Power constraints
- Pin assignment
- Power strategy
- Top-level floorplan

Tile-based physical design flow

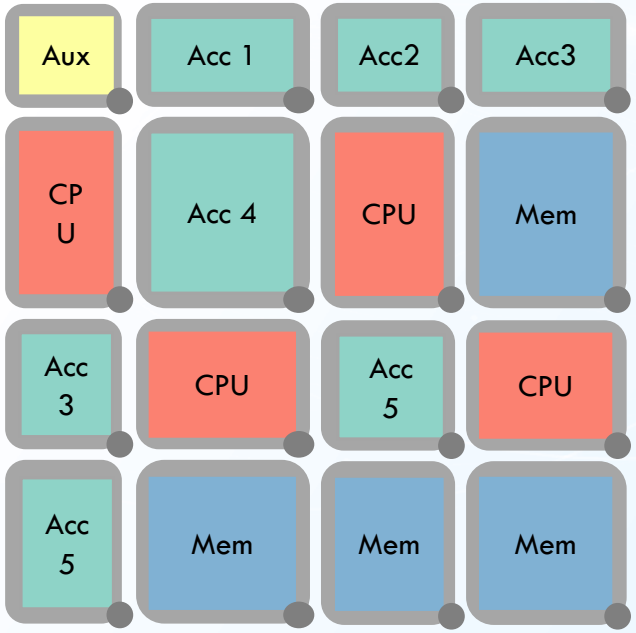
- Top-level floorplan



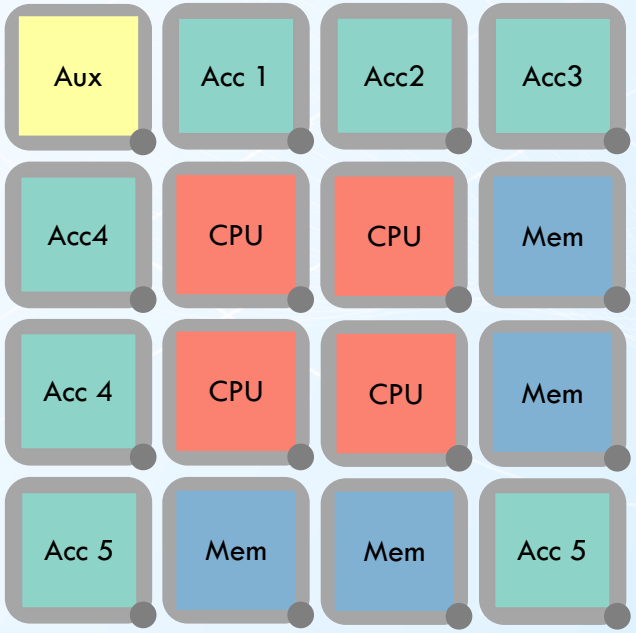
Custom shapes

Tile-based physical design flow

- Top-level floorplan



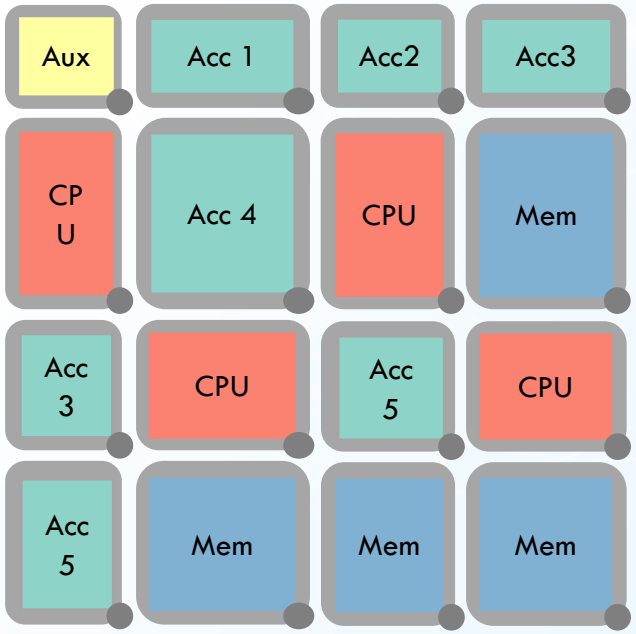
Custom shapes



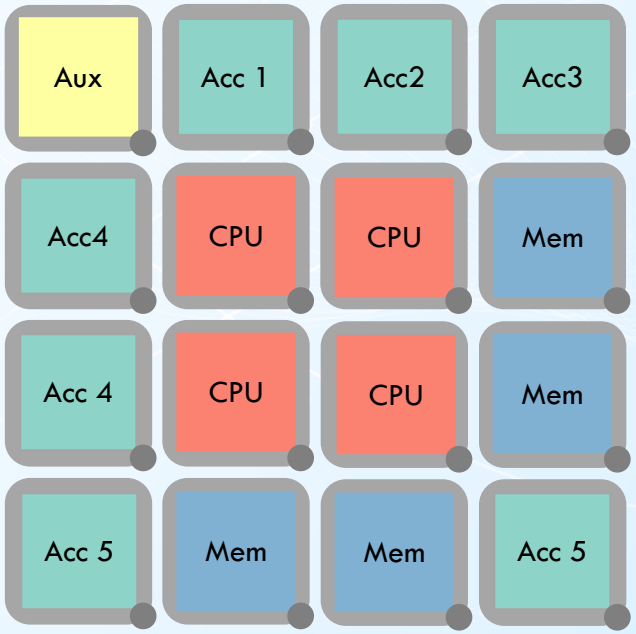
Fixed shapes

Tile-based physical design flow

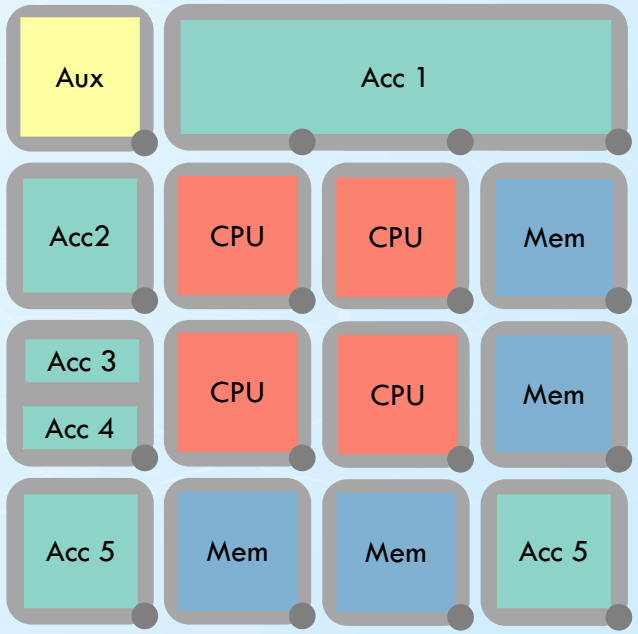
- Top-level floorplan



Custom shapes



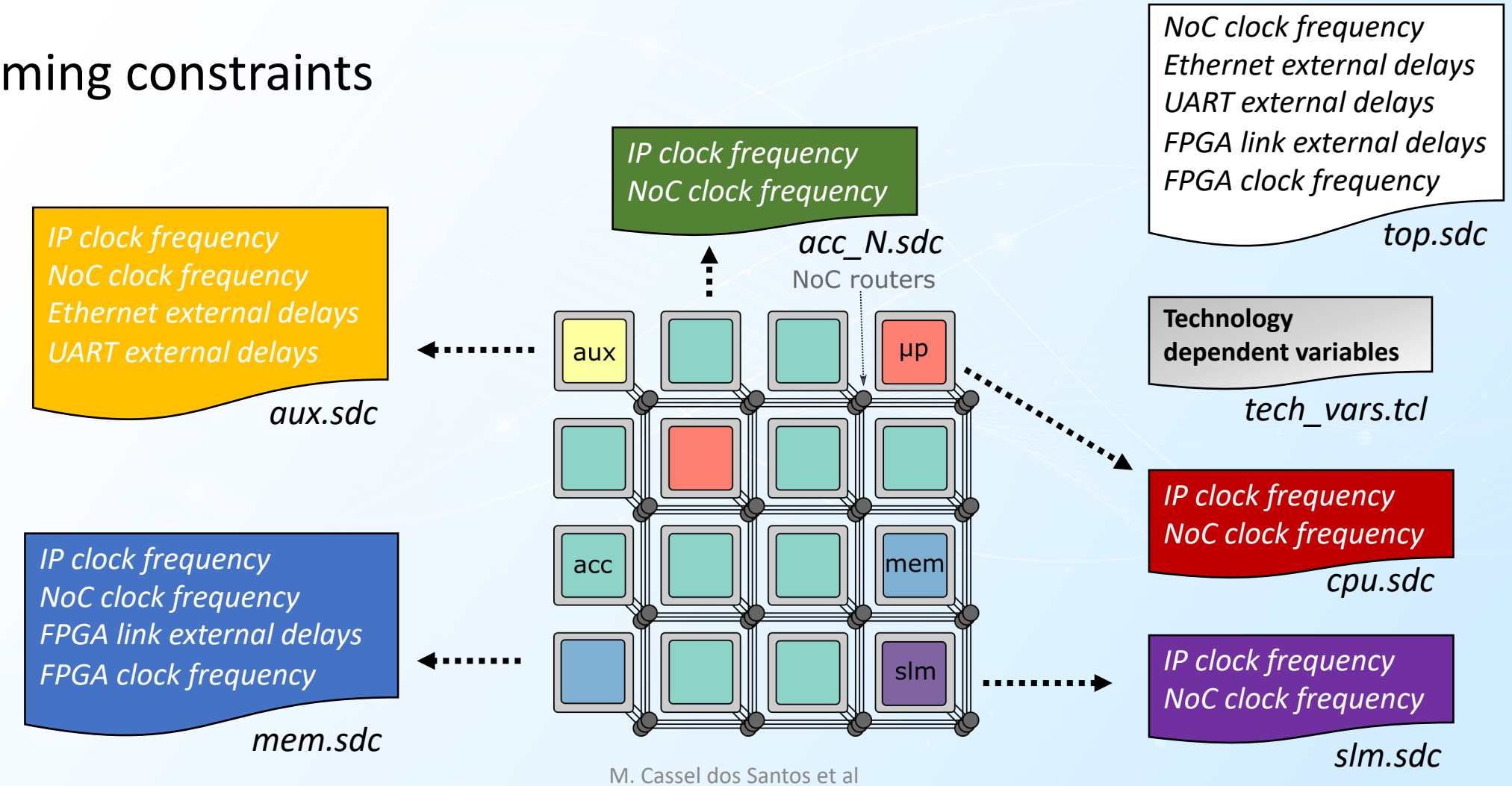
Fixed shapes



Regular shapes

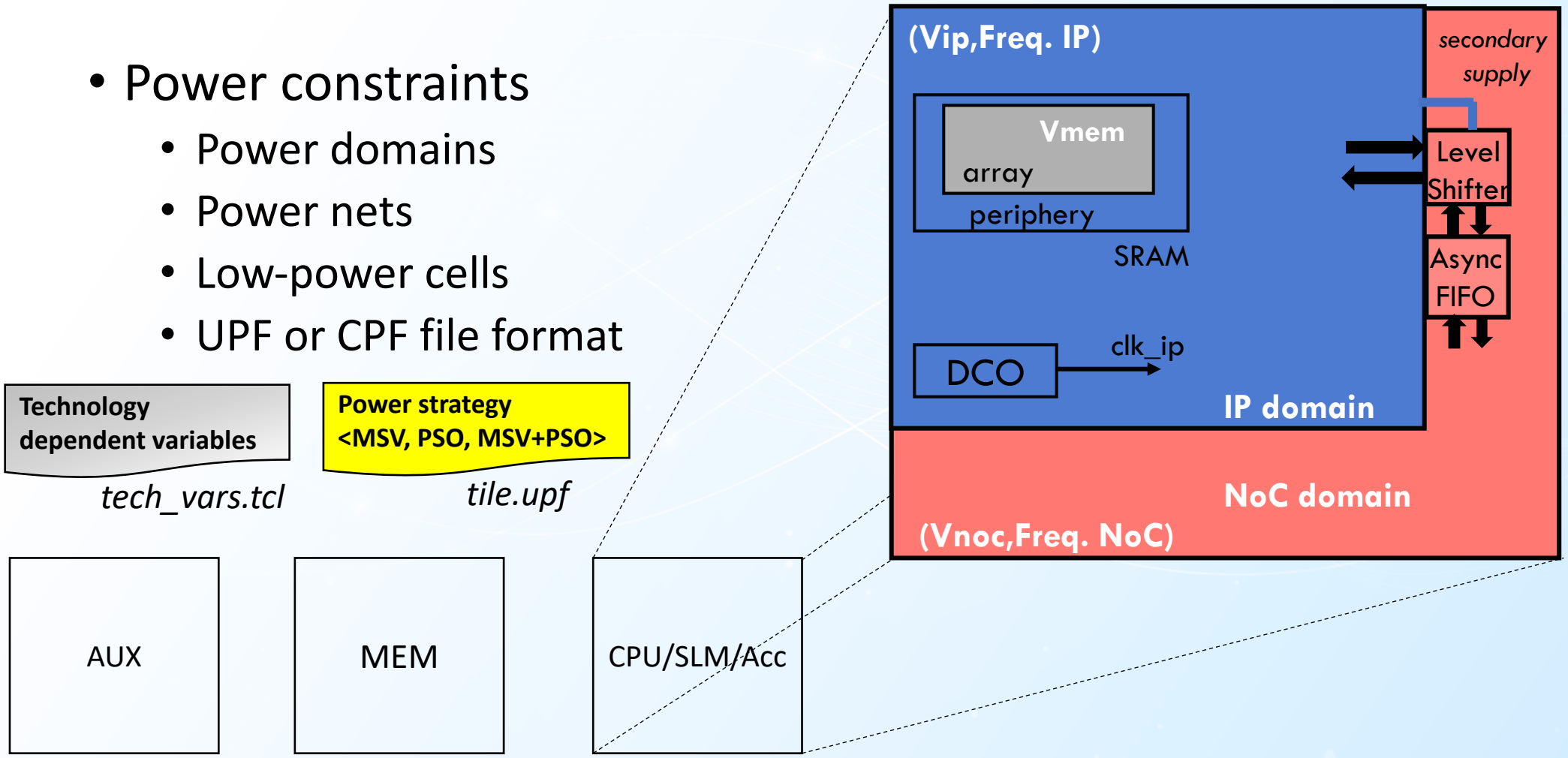
Tile-based physical design flow

- Timing constraints



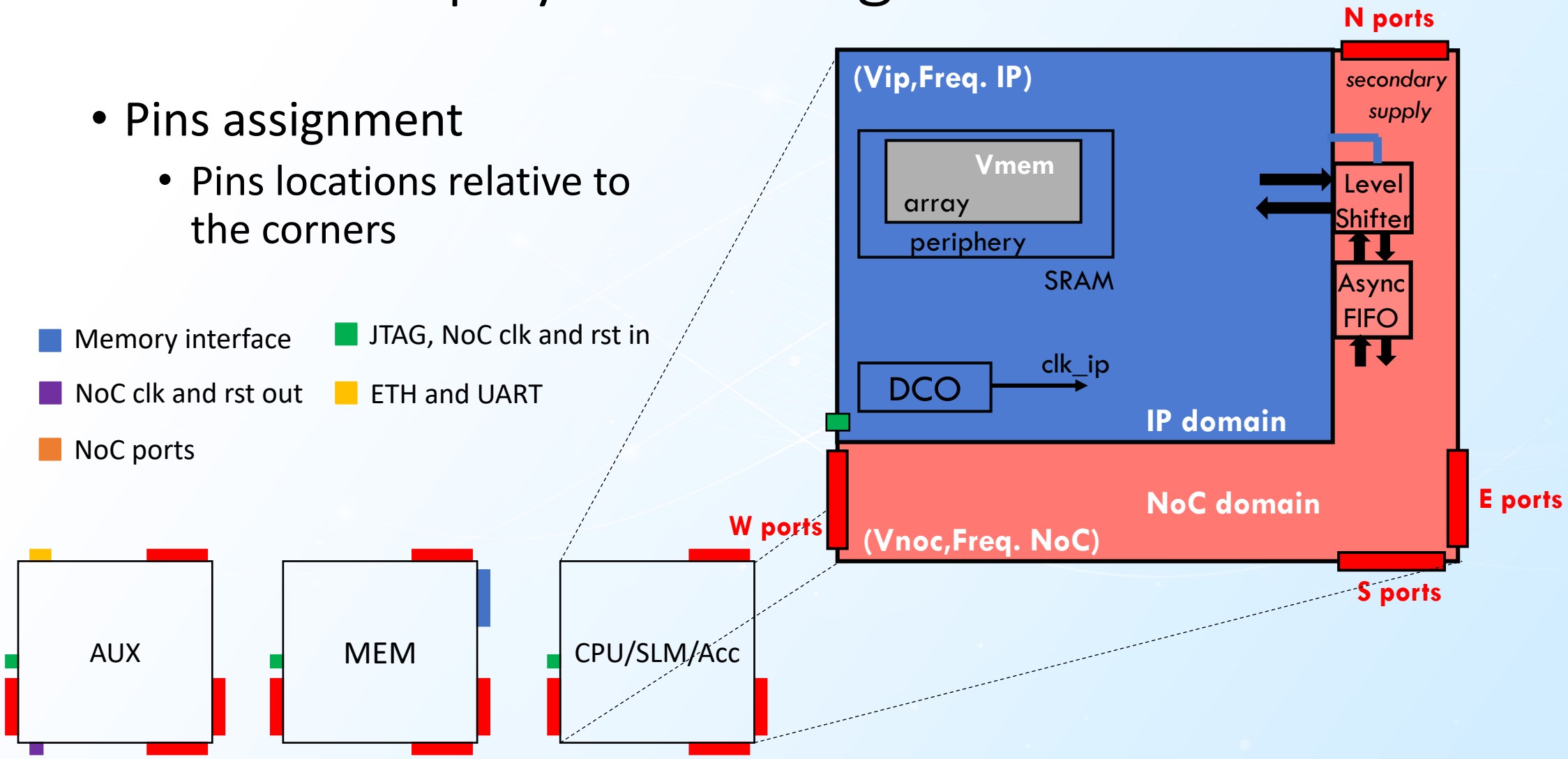
Tile-based physical design flow

- Power constraints
  - Power domains
  - Power nets
  - Low-power cells
  - UPF or CPF file format



Tile-based physical design flow

- Pins assignment
  - Pins locations relative to the corners



Tile-based physical design flow

- Power Strategy

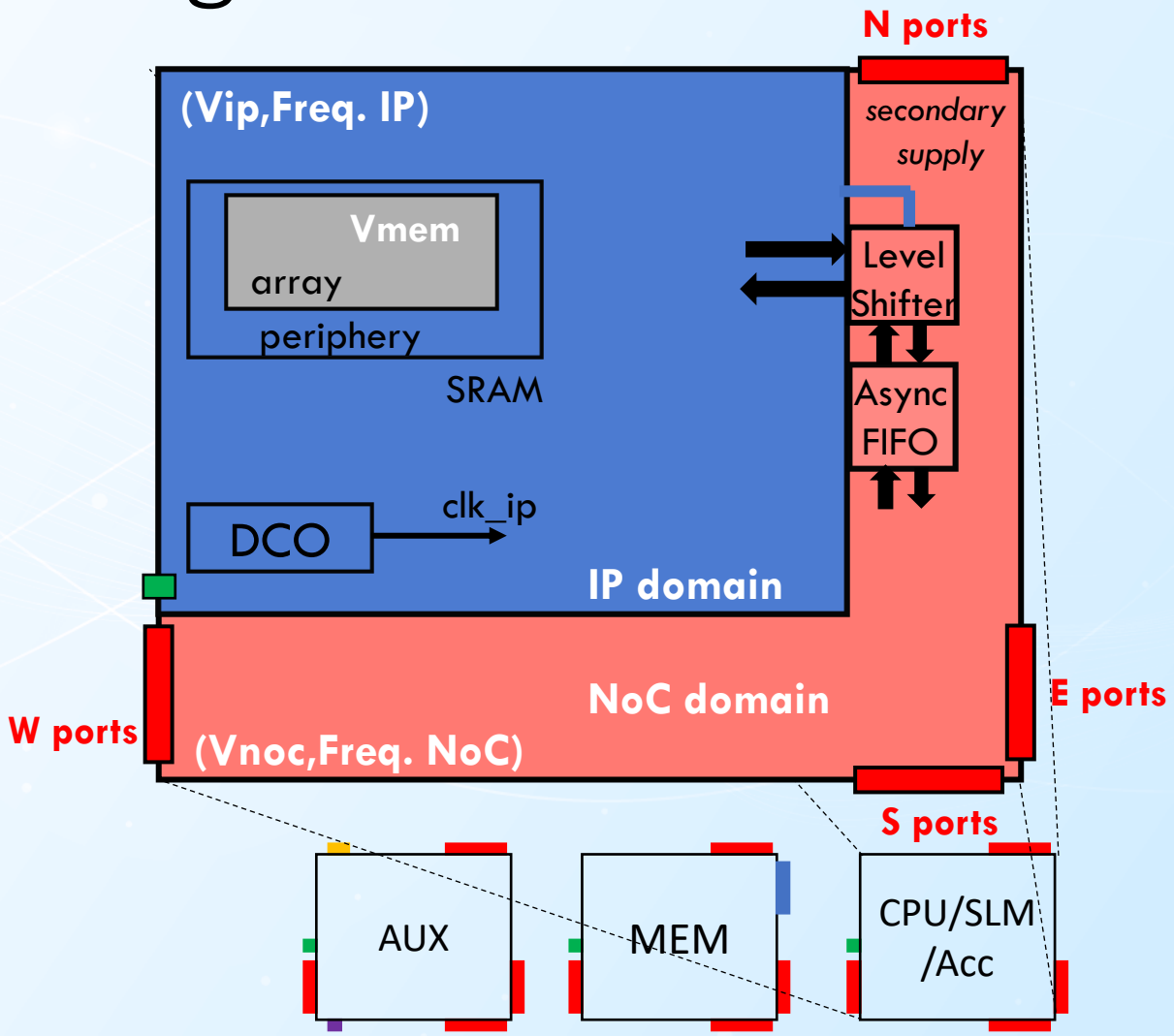
For a 13-layer process

Layers	IP	NoC
M1-4	Vip/Vss	Vnoc/Vss
M5-9	Vip/Vmem/ Vss	Vnoc/Vip/ Vss
M10-11	Vip/Vmem/Vnoc/Vss	

M12-M13 reserved for top level

For a 6-layer process

Layers	IP	NoC
M1	Vip/Vss	Vnoc/Vss
M2-4	Vip/Vss	Vnoc/Vss
M5-6	Vip/Vnoc/Vss	



Tile-based physical design flow

- Power Strategy

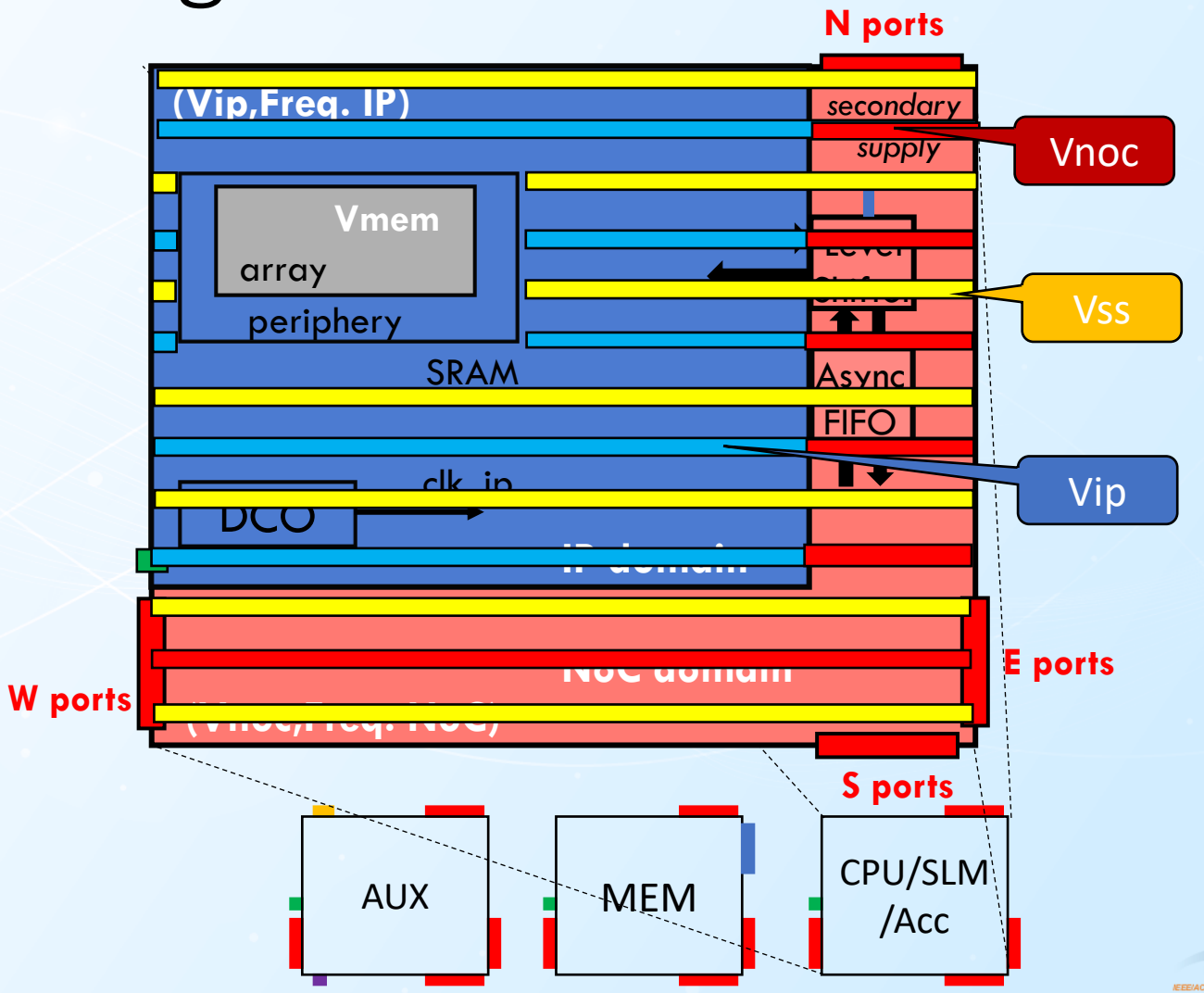
For a 13-layer process

Layers	IP	NoC
M1-4	Vip/Vss	Vnoc/Vss
M5-9	Vip/Vmem/ Vss	Vnoc/Vip/ Vss
M10-11	Vip/Vmem/Vnoc/Vss	

M12-M13 reserved for top level

For a 6-layer process

Layers	IP	NoC
M1	Vip/Vss	Vnoc/Vss
M2-4	Vip/Vss	Vnoc/Vss
M5-6	Vip/Vnoc/Vss	



Tile-based physical design flow

- Power Strategy

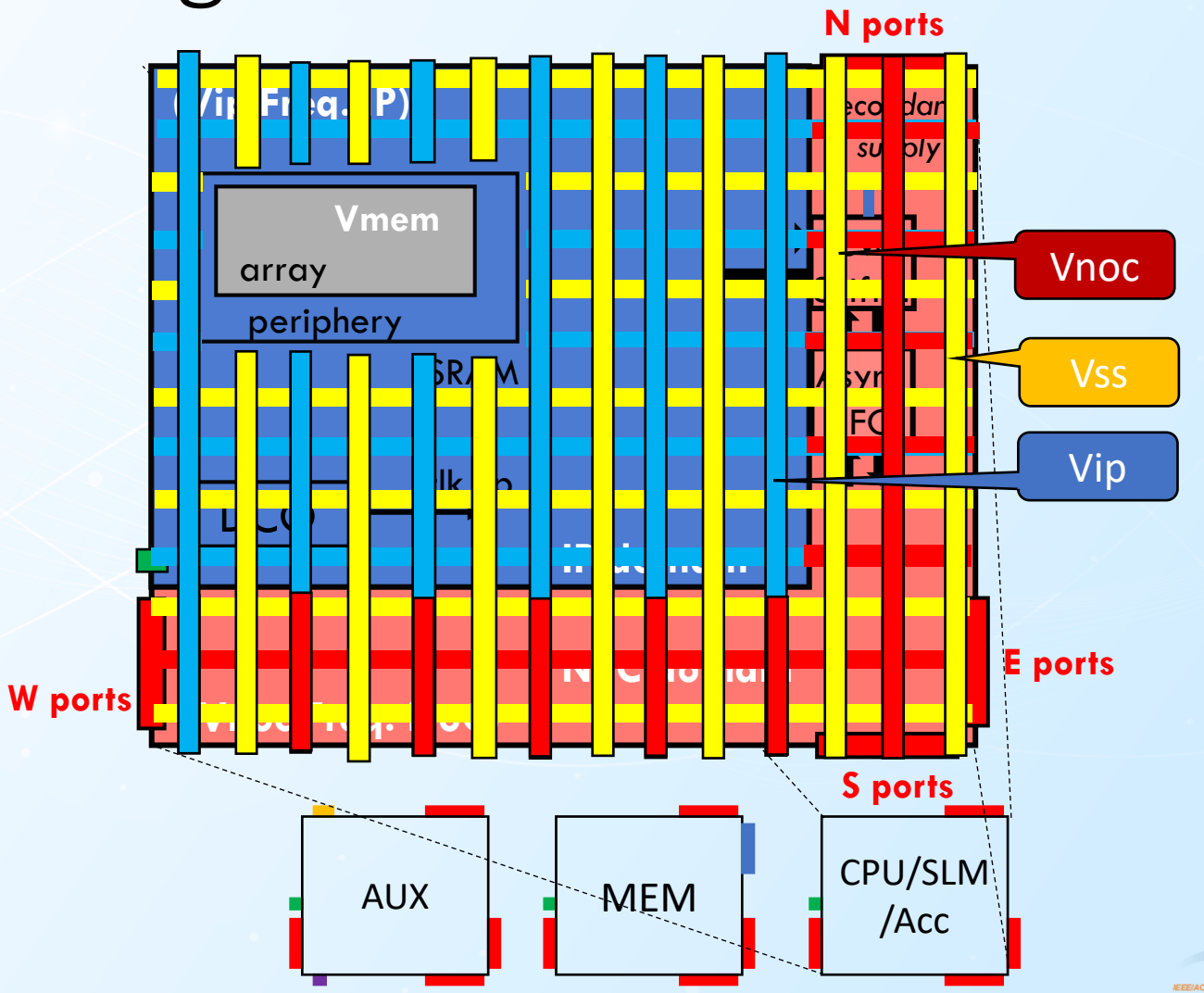
For a 13-layer process

Layers	IP	NoC
M1-4	Vip/Vss	Vnoc/Vss
M5-9	Vip/Vmem/ Vss	Vnoc/Vip/ Vss
M10-11	Vip/Vmem/Vnoc/Vss	

M12-M13 reserved for top level

For a 6-layer process

Layers	IP	NoC
M1	Vip/Vss	Vnoc/Vss
M2-4	Vip/Vss	Vnoc/Vss
M5-6	Vip/Vnoc/Vss	



Tile-based physical design flow

- Power Strategy

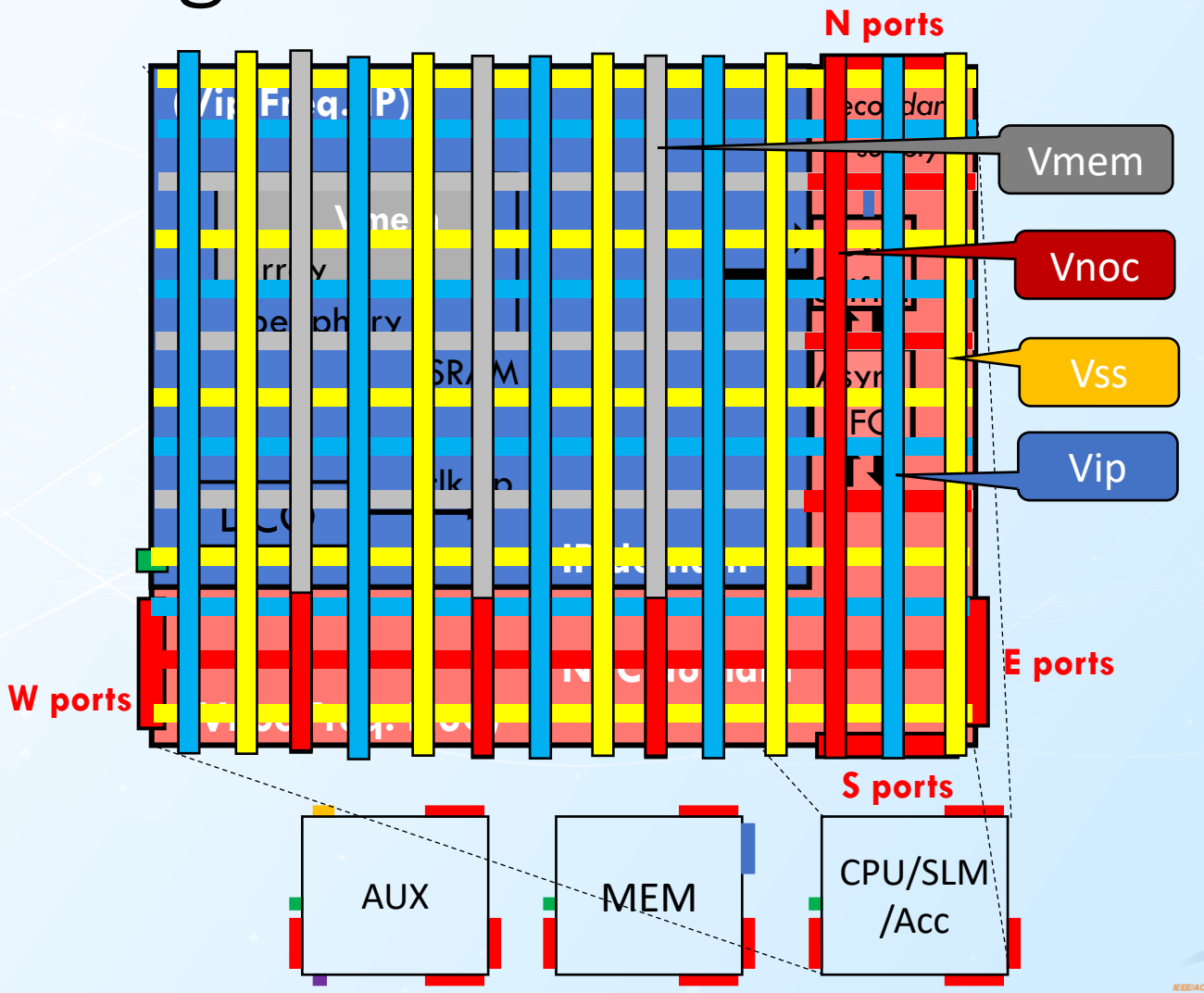
For a 13-layer process

Layers	IP	NoC
M1-4	Vip/Vss	Vnoc/Vss
M5-9	Vip/Vmem/ Vss	Vnoc/Vip/ Vss
M10-11	Vip/Vmem/Vnoc/Vss	

M12-M13 reserved for top level

For a 6-layer process

Layers	IP	NoC
M1	Vip/Vss	Vnoc/Vss
M2-4	Vip/Vss	Vnoc/Vss
M5-6	Vip/Vnoc/Vss	



Tile-based physical design flow

- Power Strategy

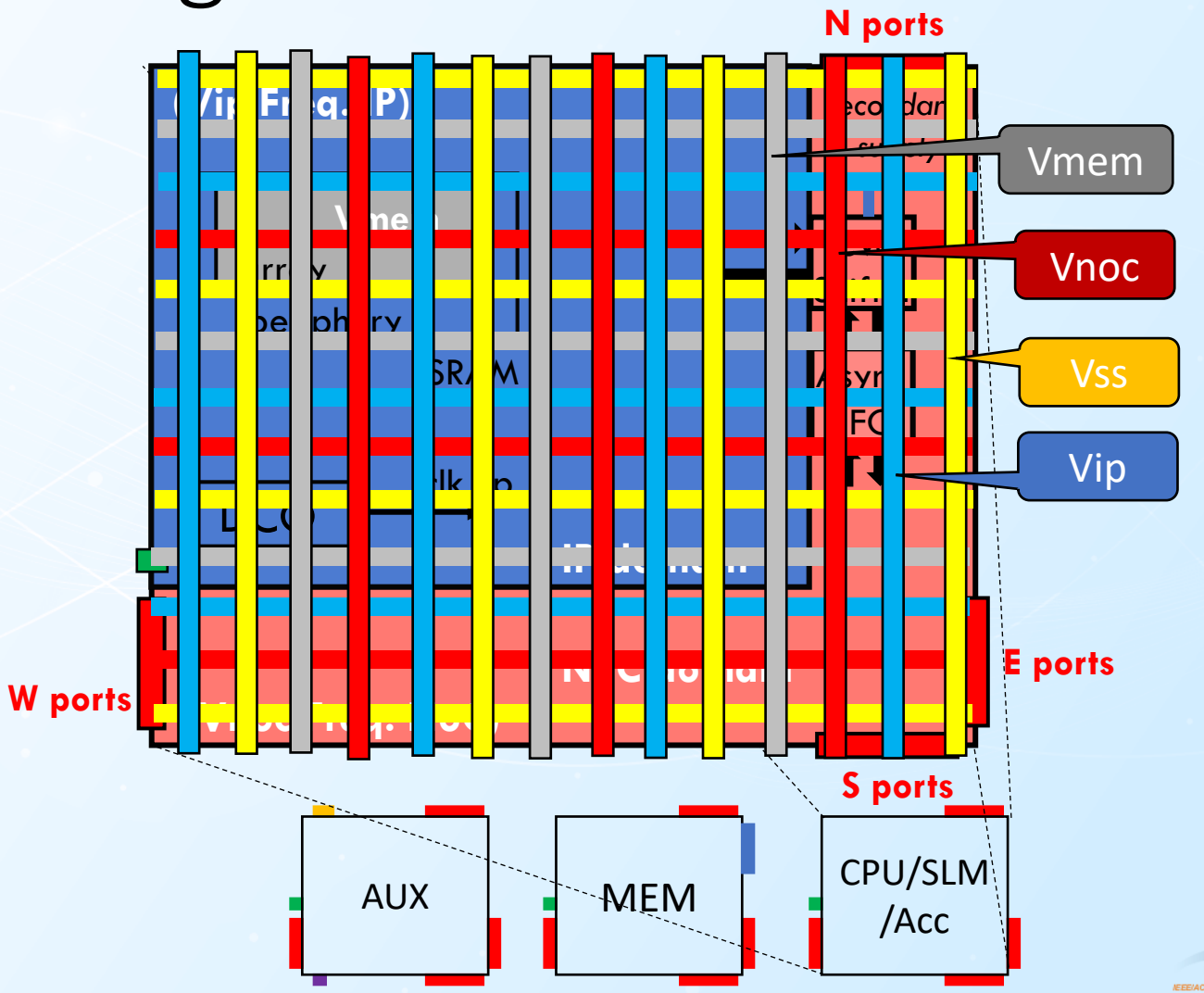
For a 13-layer process

Layers	IP	NoC
M1-4	Vip/Vss	Vnoc/Vss
M5-9	Vip/Vmem/ Vss	Vnoc/Vip/ Vss
M10-11	Vip/Vmem/Vnoc/Vss	

M12-M13 reserved for top level

For a 6-layer process

Layers	IP	NoC
M1	Vip/Vss	Vnoc/Vss
M2-4	Vip/Vss	Vnoc/Vss
M5-6	Vip/Vnoc/Vss	

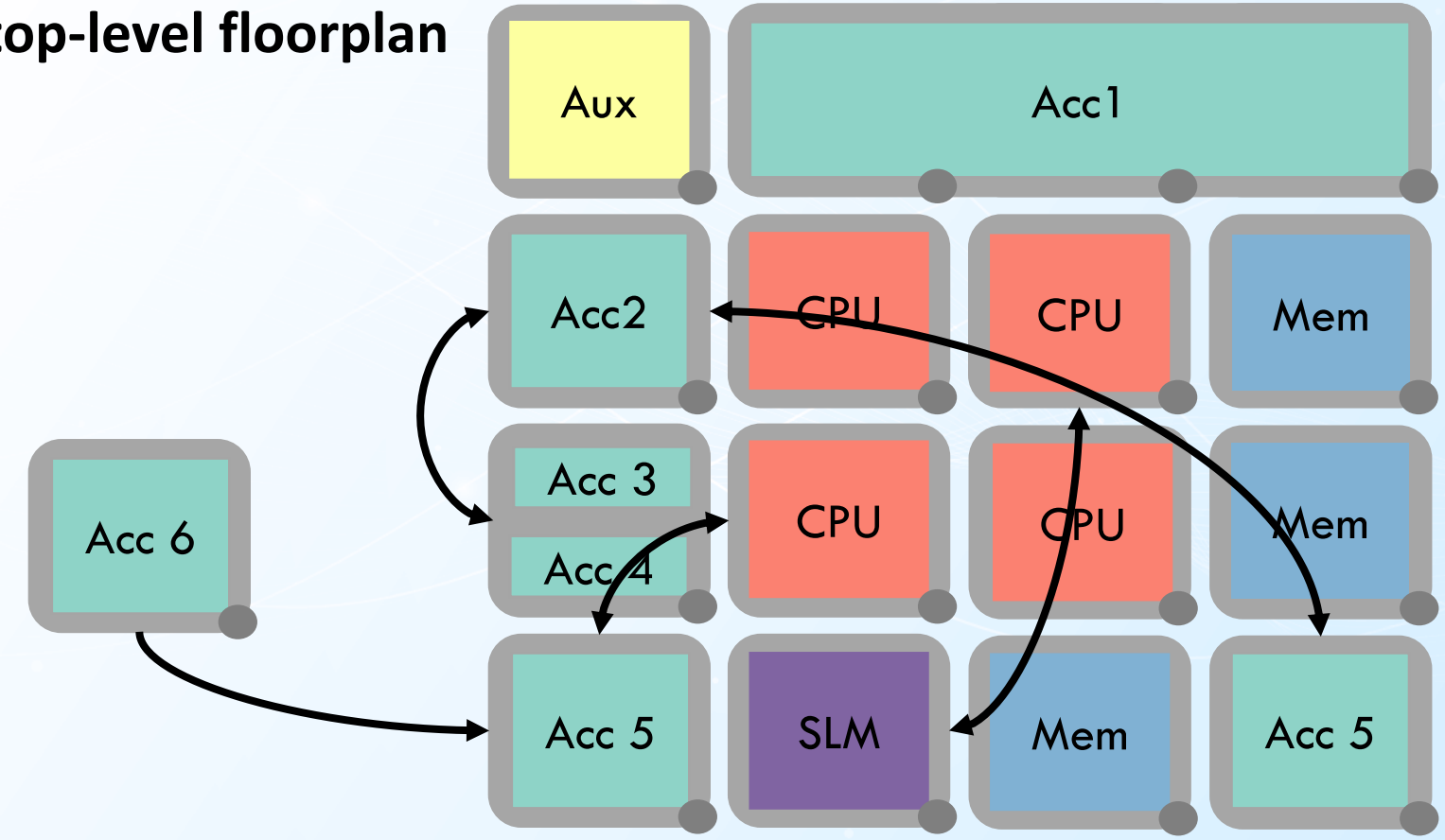


Outline

- ESP: background and enhancements for ASIC design
- Tile-based physical design flow
- SoC physical integration
- Verification and testing
- Experience with two chip designs
- Conclusion

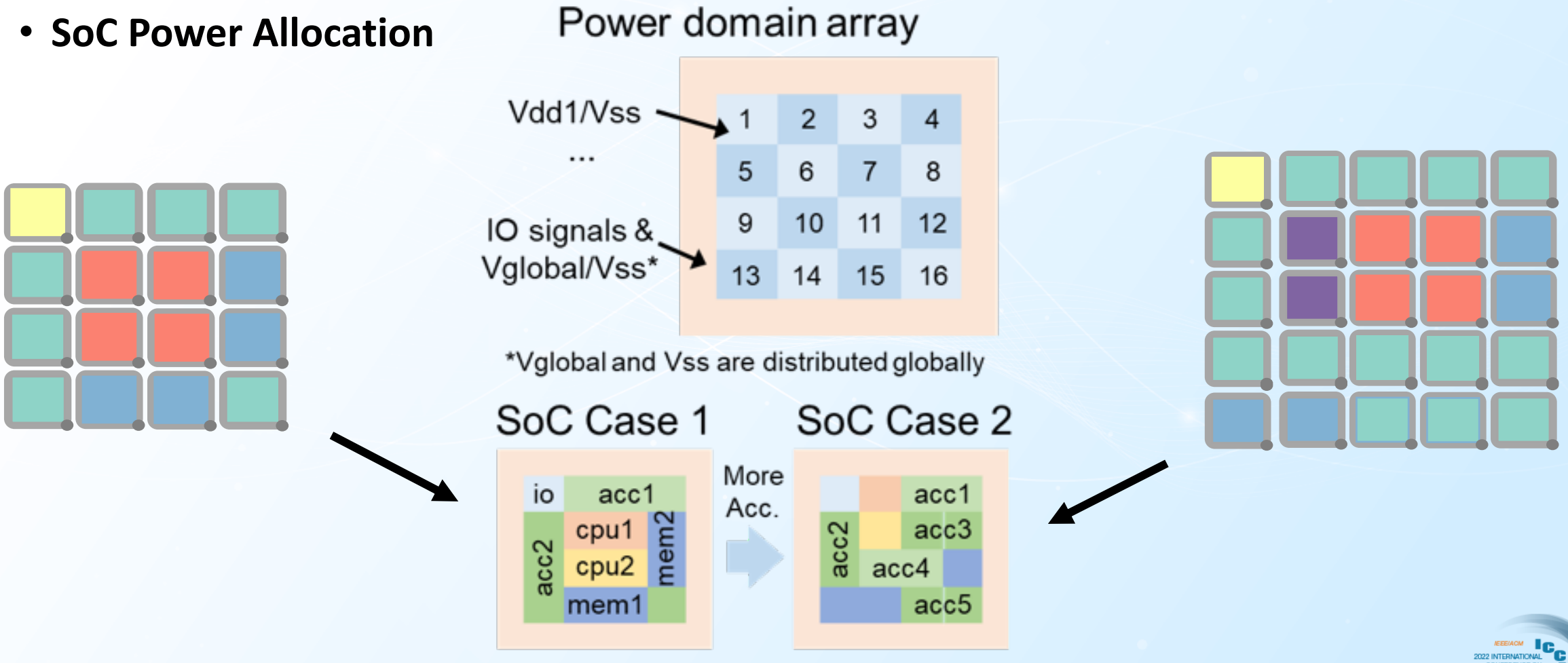
SoC physical integration

- **Flexible top-level floorplan**



SoC physical integration

- **SoC Power Allocation**



SoC physical integration

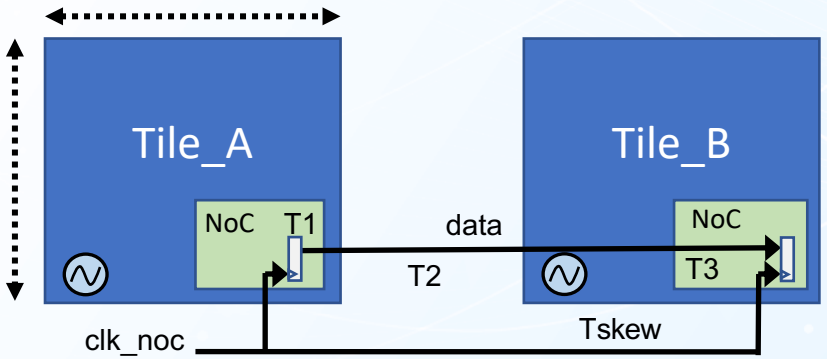
- **Timing Closure**

Setup timing:

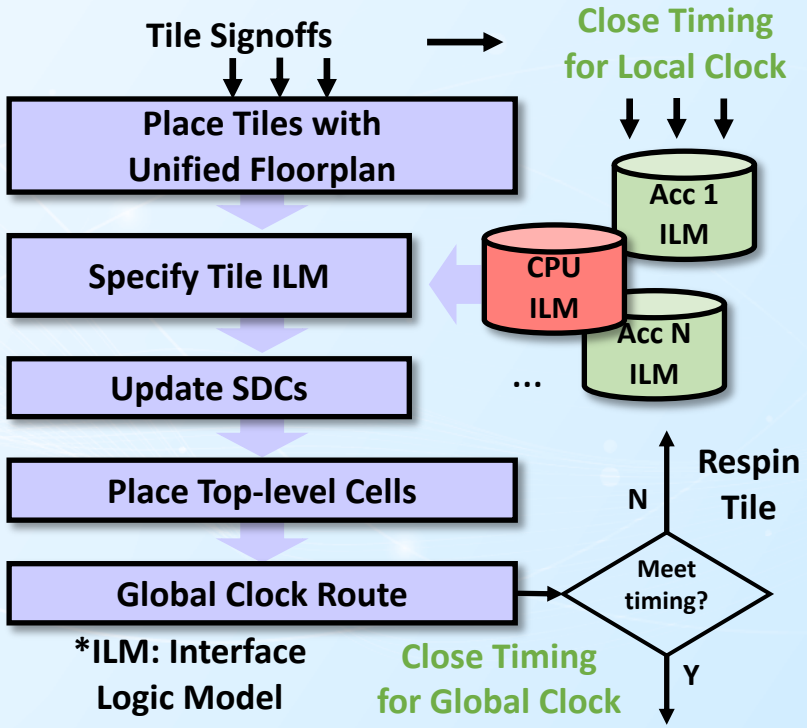
T1 + T2 + T3 < Tclk_noc – Tskew

Hold timing:

T1 + T2 + T3 > Tskew



Hierarchical ILM* - Based Timing Closure



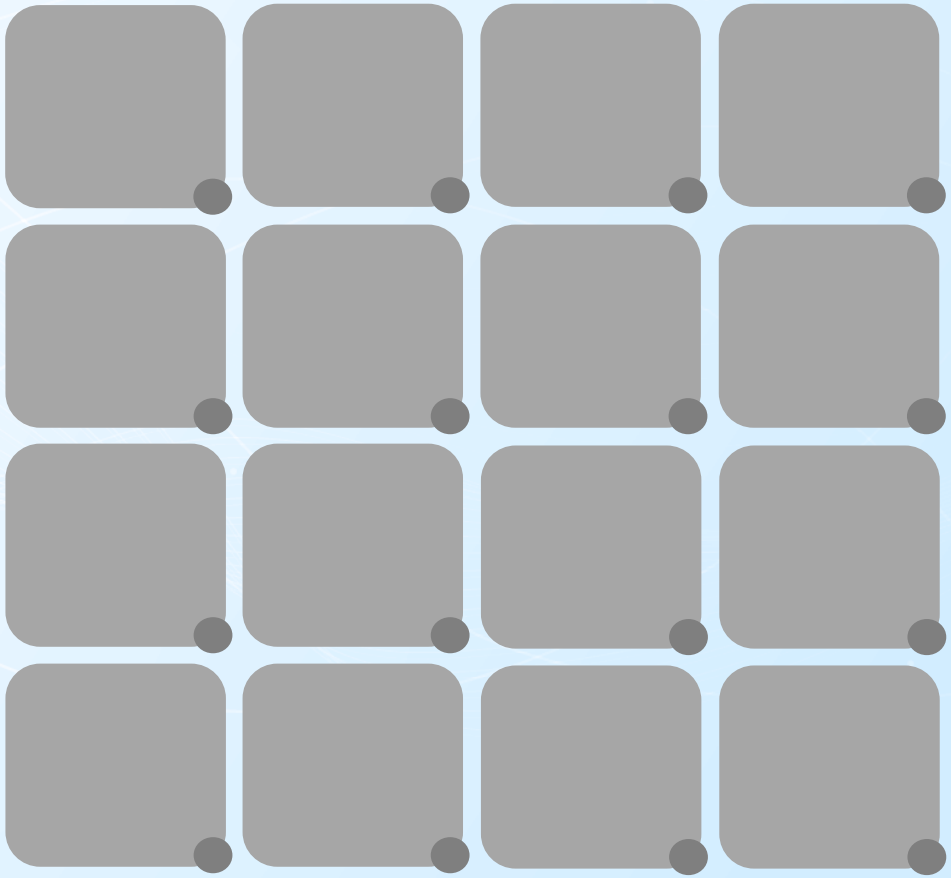
Only need to constrain neighboring tiles!

Outline

- ESP: background and enhancements for ASIC design
- Tile-based physical design flow
- SoC physical integration
- Verification and testing
- Experience with two chip designs
- Conclusion

Verification

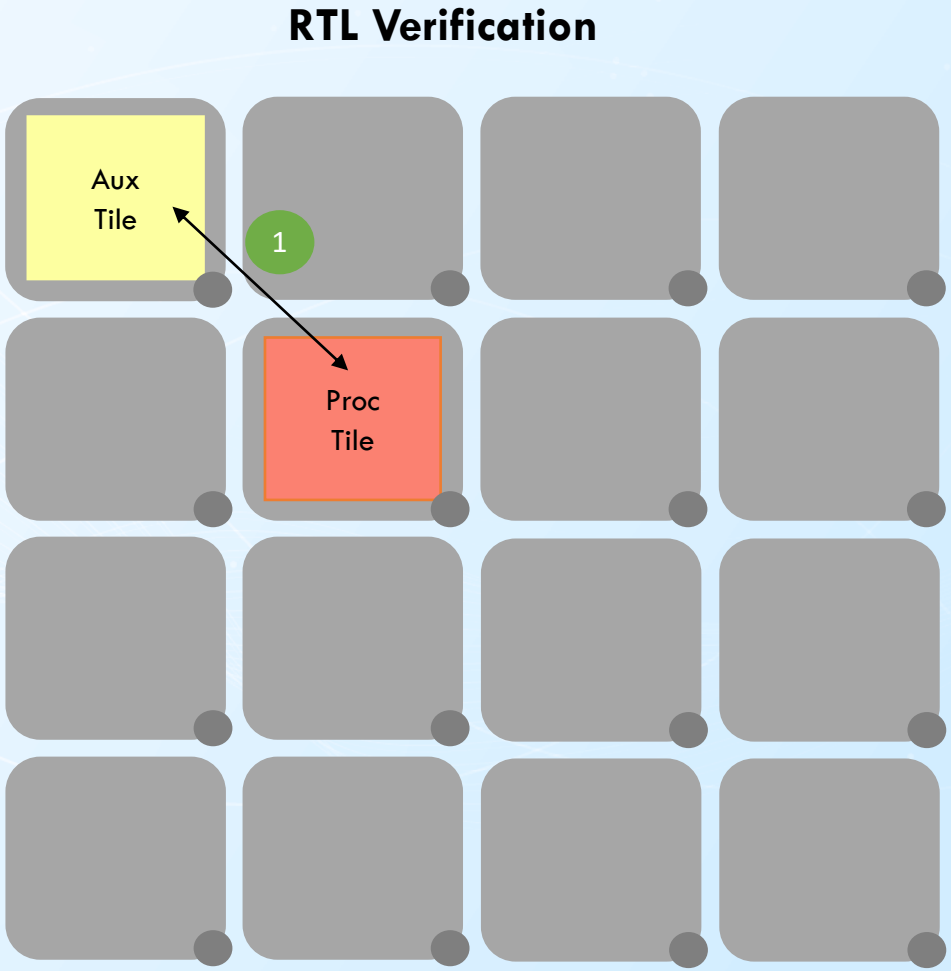
- Focus on system-level
  - Integration of new components
  - System functionality
- Assumptions
  - OSH components are thoroughly verified
  - ESP NoC, sockets, buses, and platform services are pre-verified
- Four verification approaches
  - RTL simulation using bare-metal applications
  - Netlist simulation with simplified bare-metal applications
  - Full system FPGA emulation for longer tests
  - Logic Equivalence Checking



Verification

Bare-metal

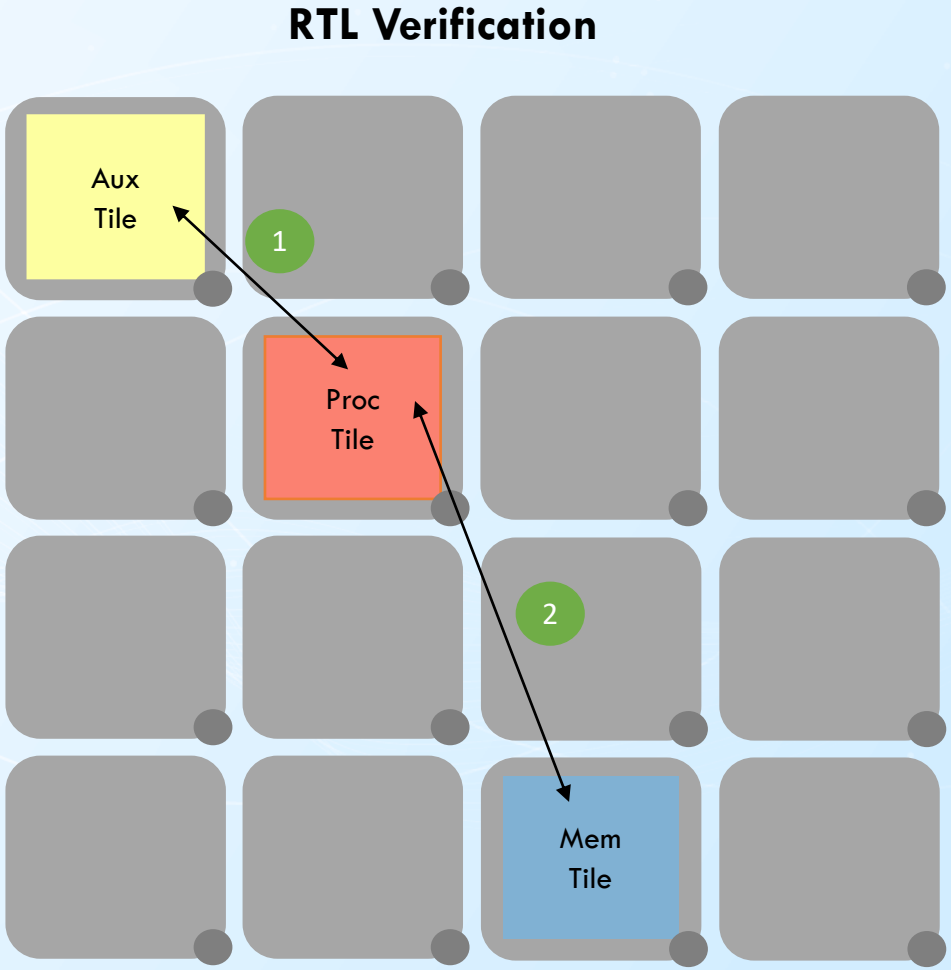
1. CPU executes bootloader from Aux tile



Verification

Bare-metal

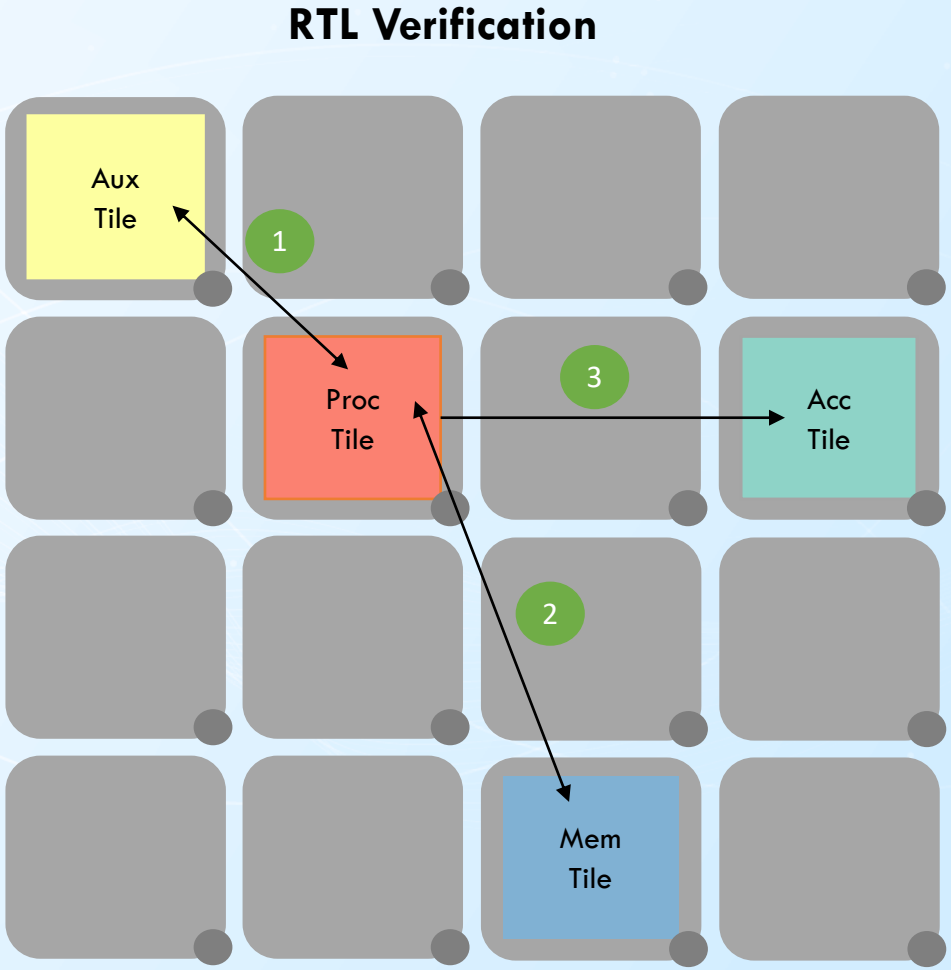
1. CPU executes bootloader from Aux tile
2. CPU executes program from main memory, sets up data for accelerator



Verification

Bare-metal

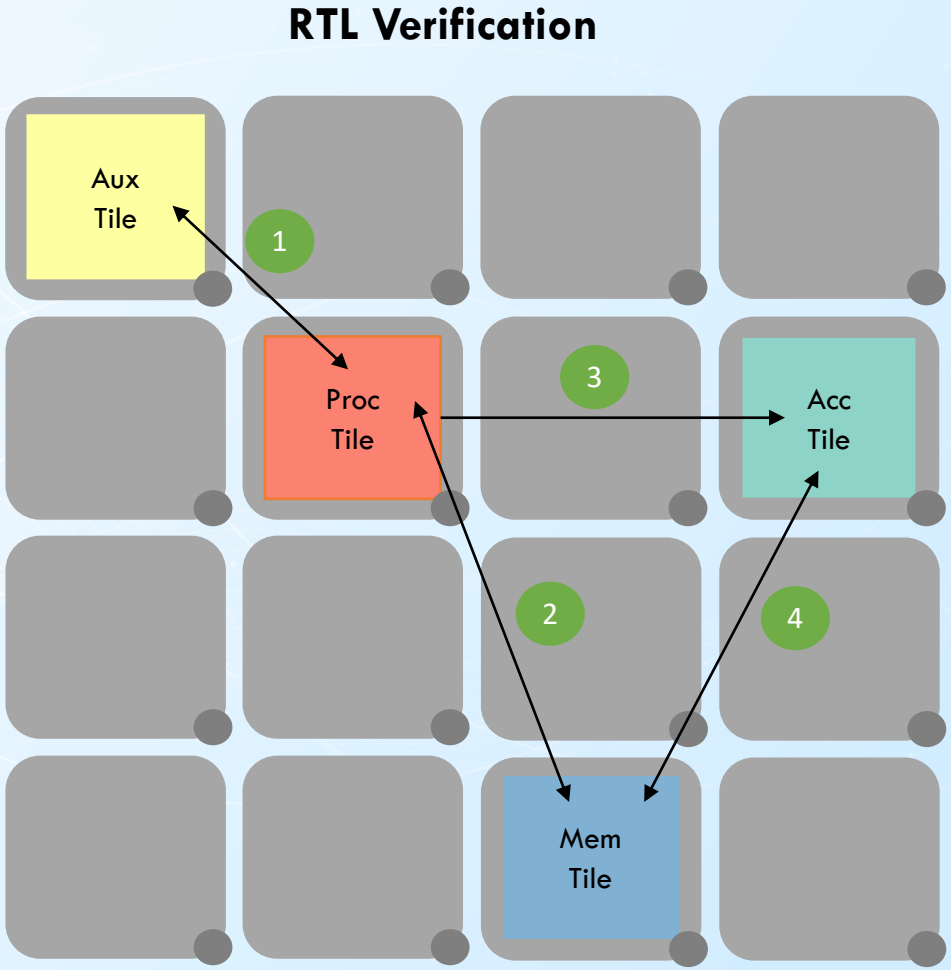
1. CPU executes bootloader from Aux tile
2. CPU executes program from main memory, sets up data for accelerator
3. CPU configures and starts accelerator



Verification

Bare-metal

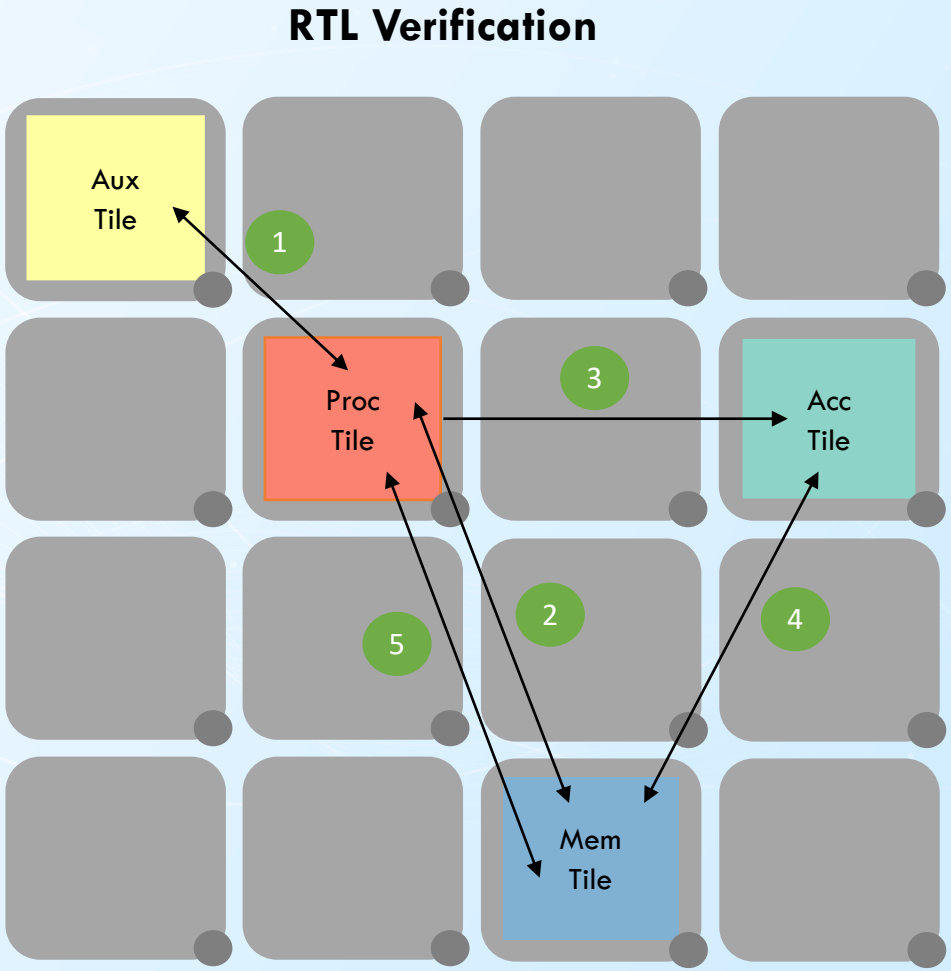
1. CPU executes bootloader from Aux tile
2. CPU executes program from main memory, sets up data for accelerator
3. CPU configures and starts accelerator
4. Accelerator performs DMA to main memory



Verification

Bare-metal

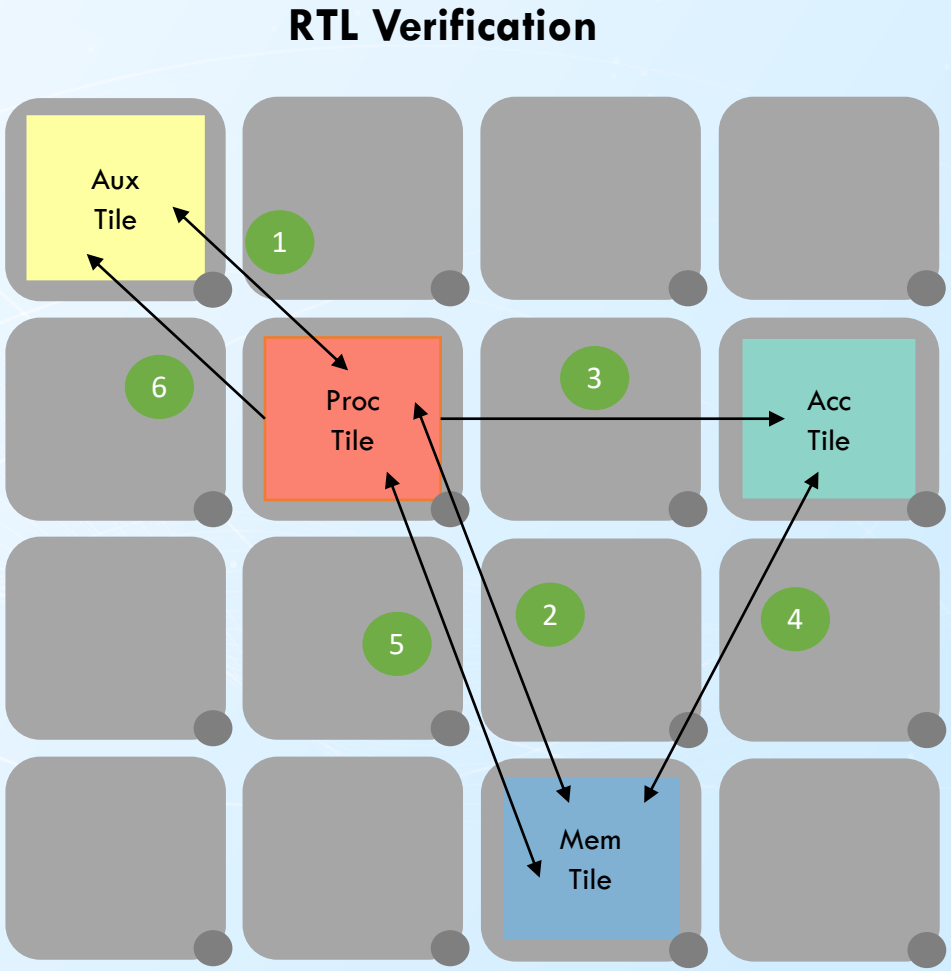
1. CPU executes bootloader from Aux tile
2. CPU executes program from main memory, sets up data for accelerator
3. CPU configures and starts accelerator
4. Accelerator performs DMA to main memory
5. CPU validates accelerator outputs



Verification

Bare-metal

1. CPU executes bootloader from Aux tile
2. CPU executes program from main memory, sets up data for accelerator
3. CPU configures and starts accelerator
4. Accelerator performs DMA to main memory
5. CPU validates accelerator outputs
6. CPU writes outputs through UART

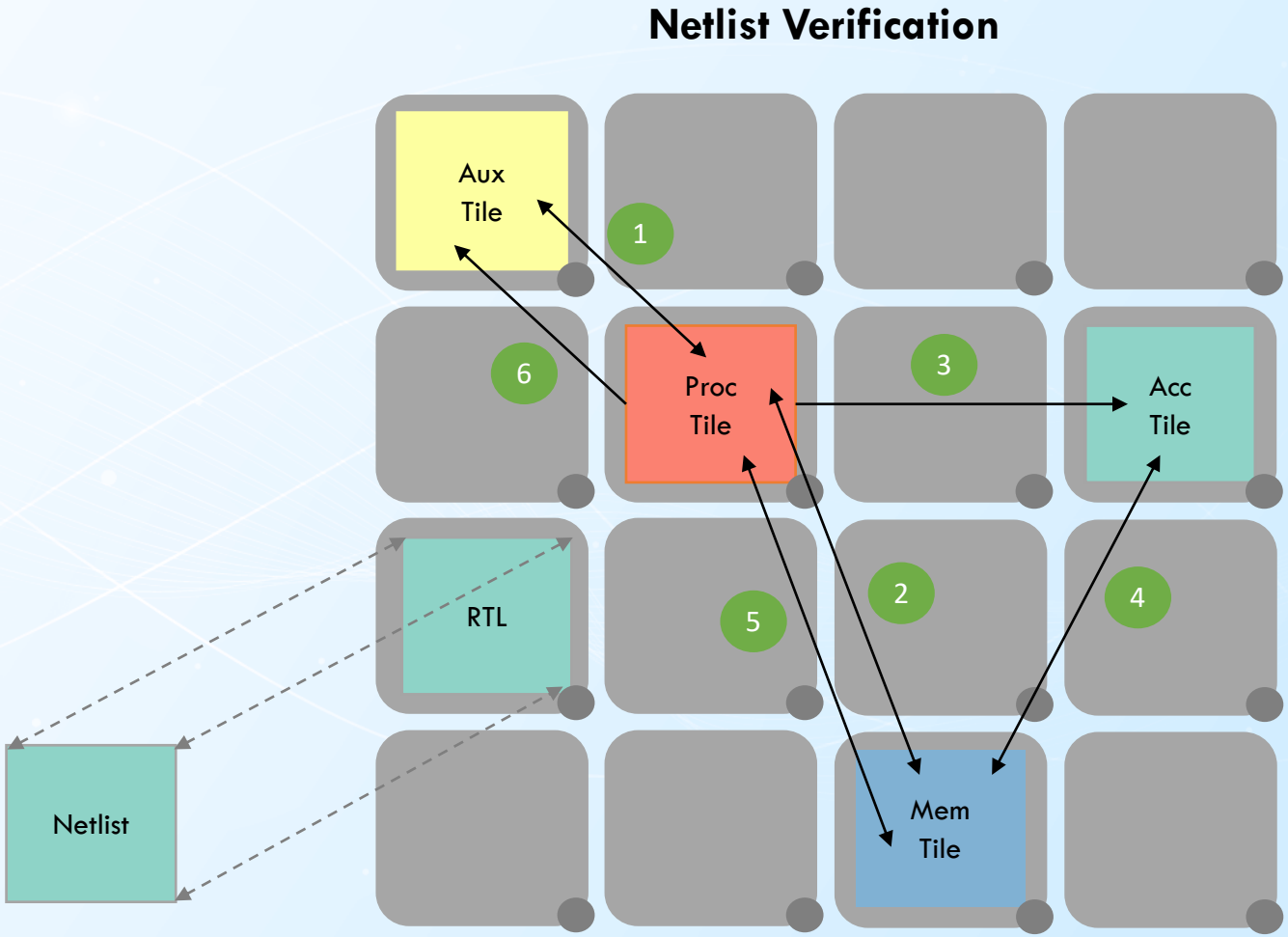


Verification

Simplified bare-metal

1. CPU executes bootloader from Aux tile
2. CPU executes program from main memory, sets up data for accelerator
3. CPU configures and starts accelerator
4. Accelerator performs DMA to main memory
5. CPU validates accelerator outputs
6. CPU writes outputs through UART

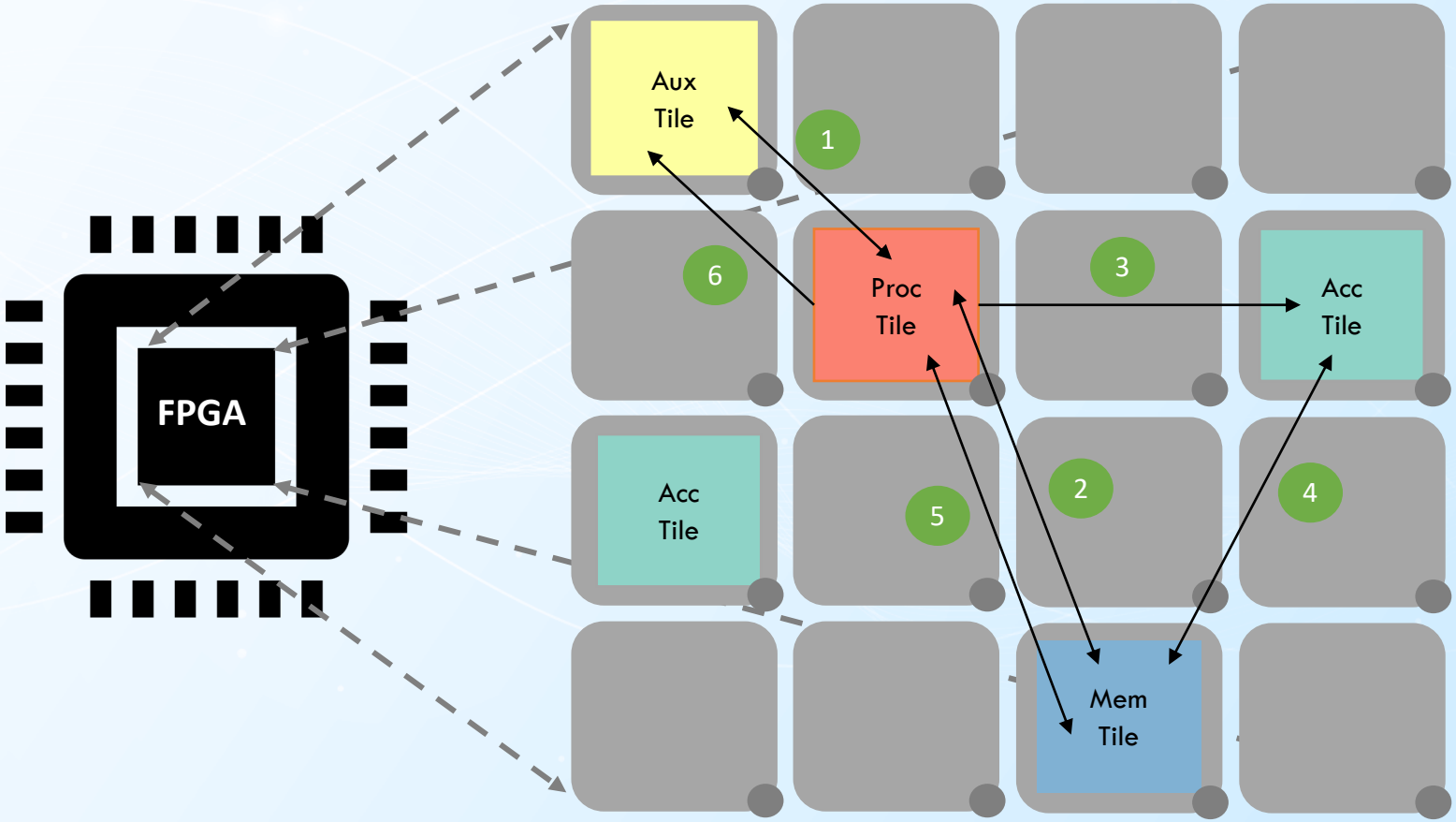
The tile under test uses the netlist
The rest of the system uses RTL



Verification

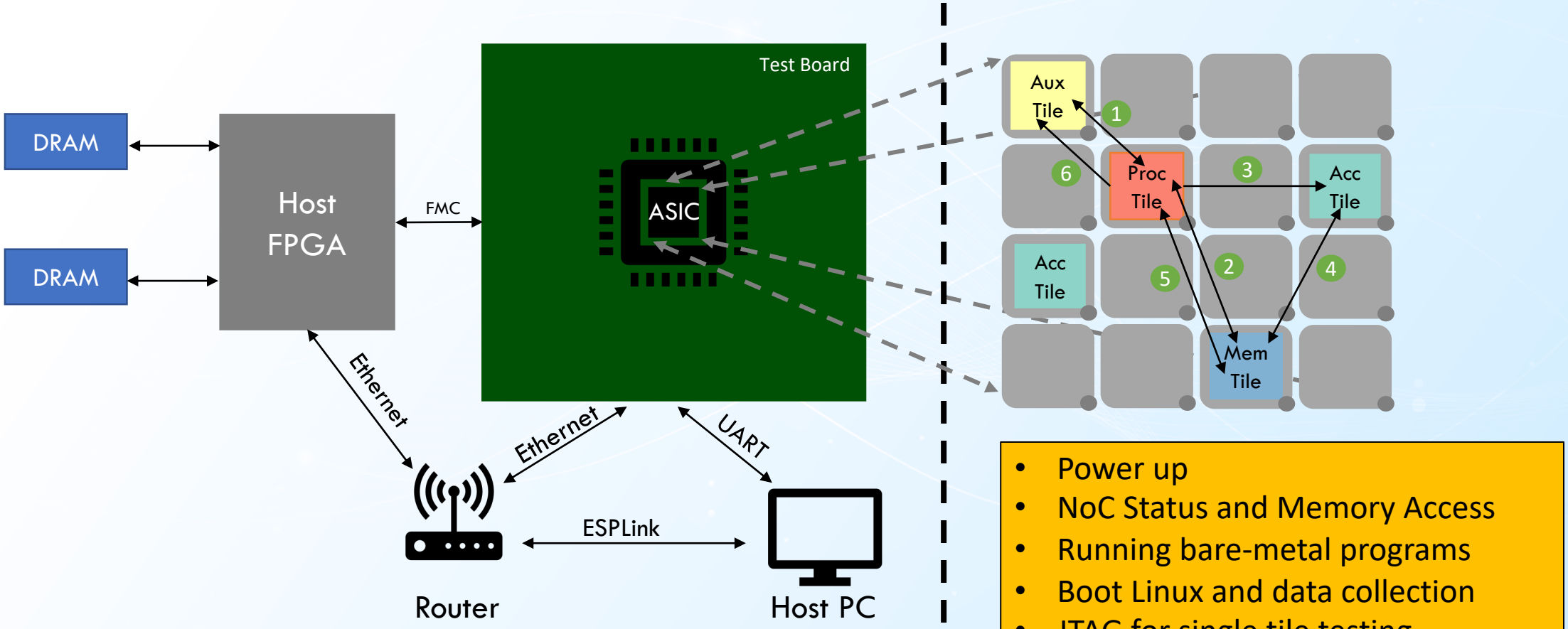
FPGA Emulation

- Boot Linux OS
- Run Linux applications
- Emulates full system application
- Performance estimates



For longer tests

Testing

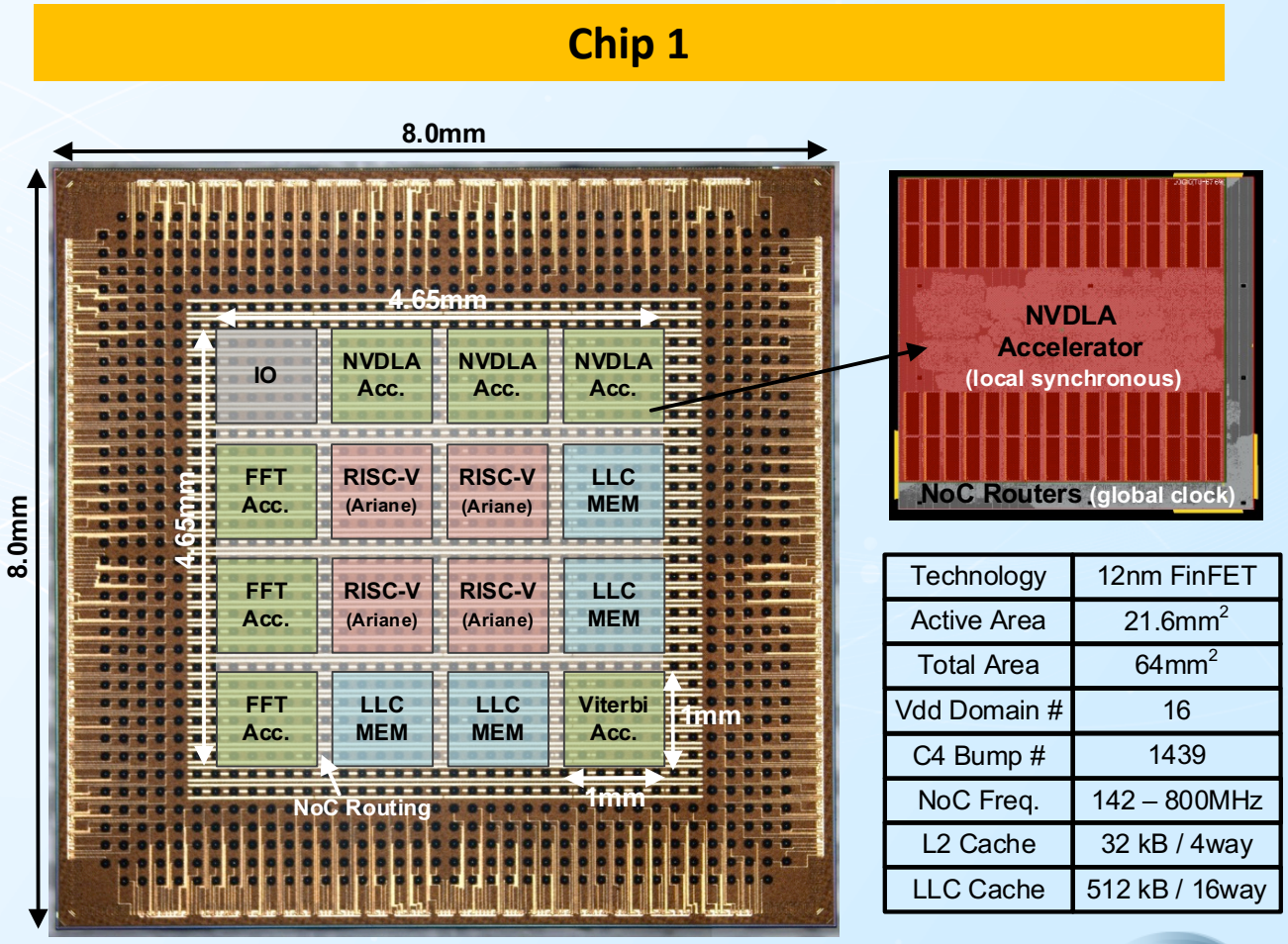


- Power up
- NoC Status and Memory Access
- Running bare-metal programs
- Boot Linux and data collection
- JTAG for single tile testing

Outline

- ESP: background and enhancements for ASIC design
- Tile-based physical design flow
- SoC physical integration
- Verification and testing
- Experience with two chip designs
- Conclusion

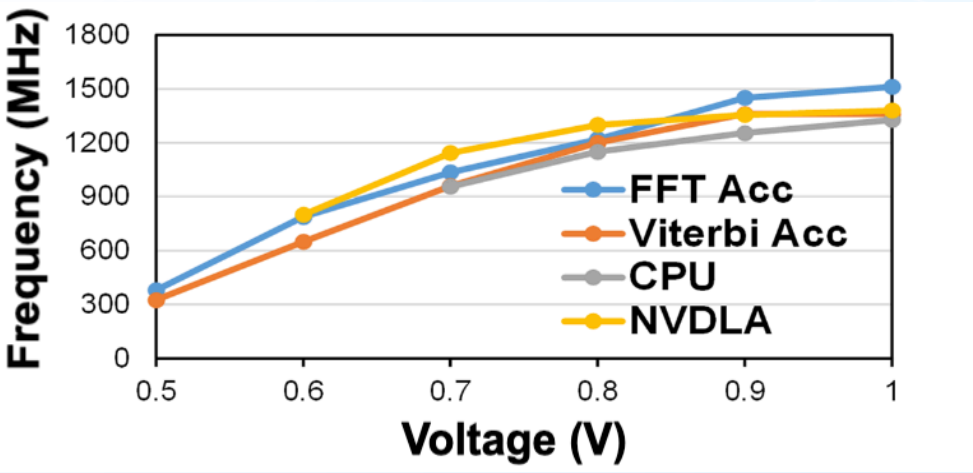
Experience with two chips designs



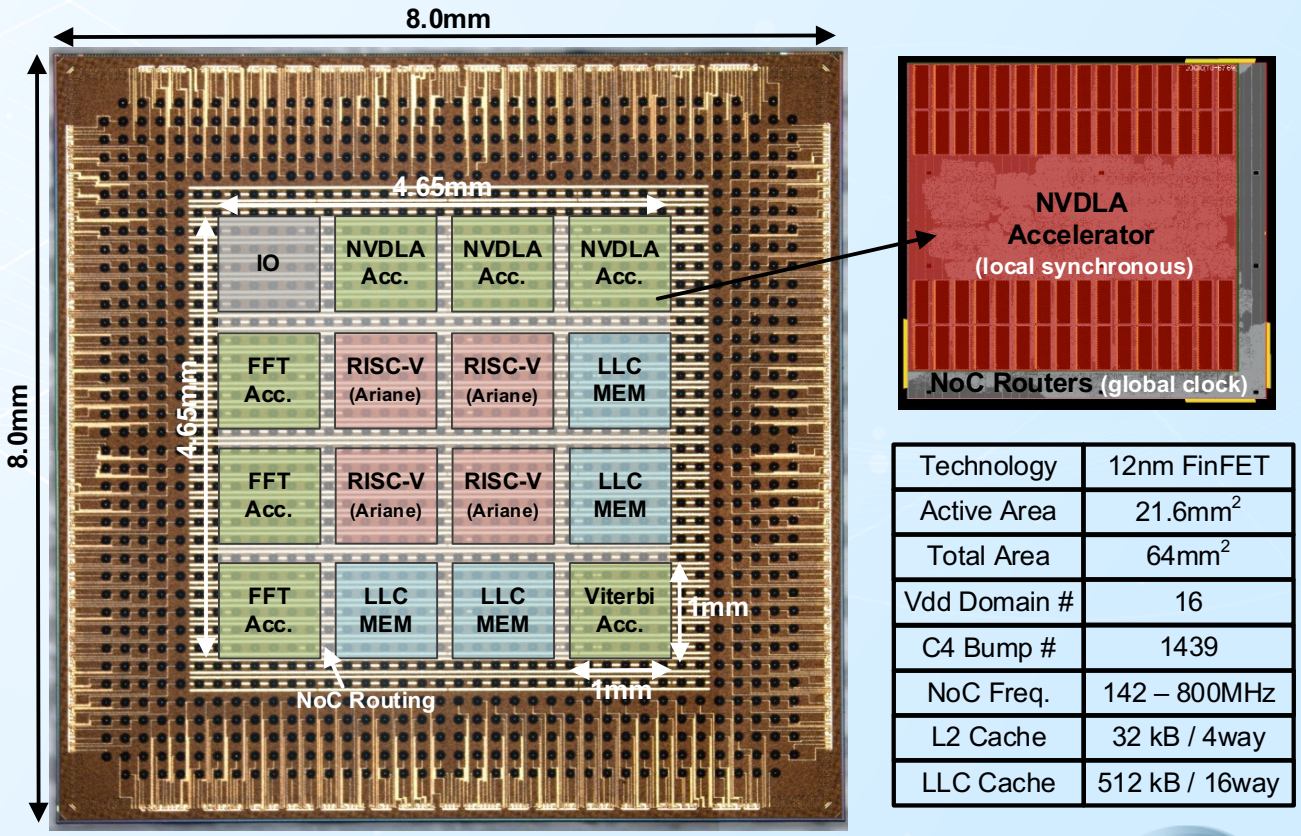
Technology	12nm FinFET
Active Area	21.6mm^2
Total Area	64mm^2
Vdd Domain #	16
C4 Bump #	1439
NoC Freq.	142 – 800MHz
L2 Cache	32 kB / 4way
LLC Cache	512 kB / 16way

T. Jia et al. (ESSCIRC 2022)

Experience with two chips designs



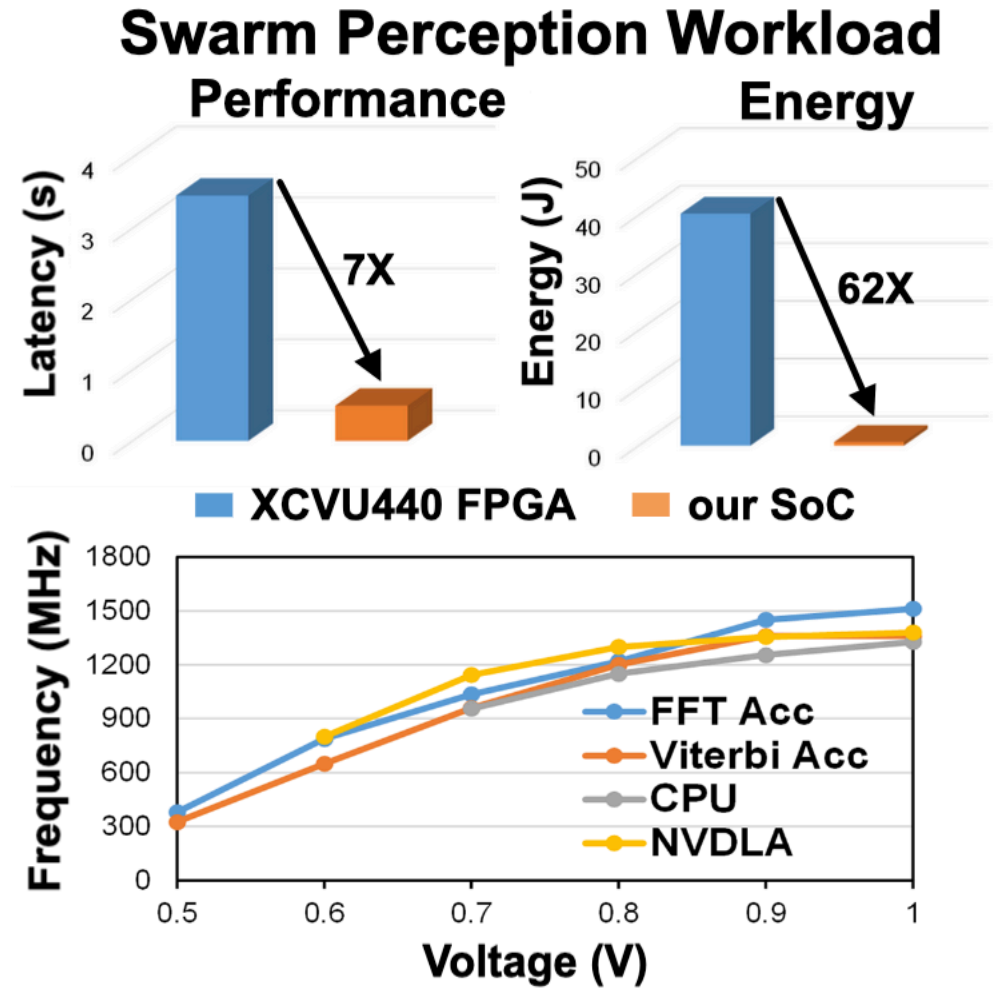
Chip 1

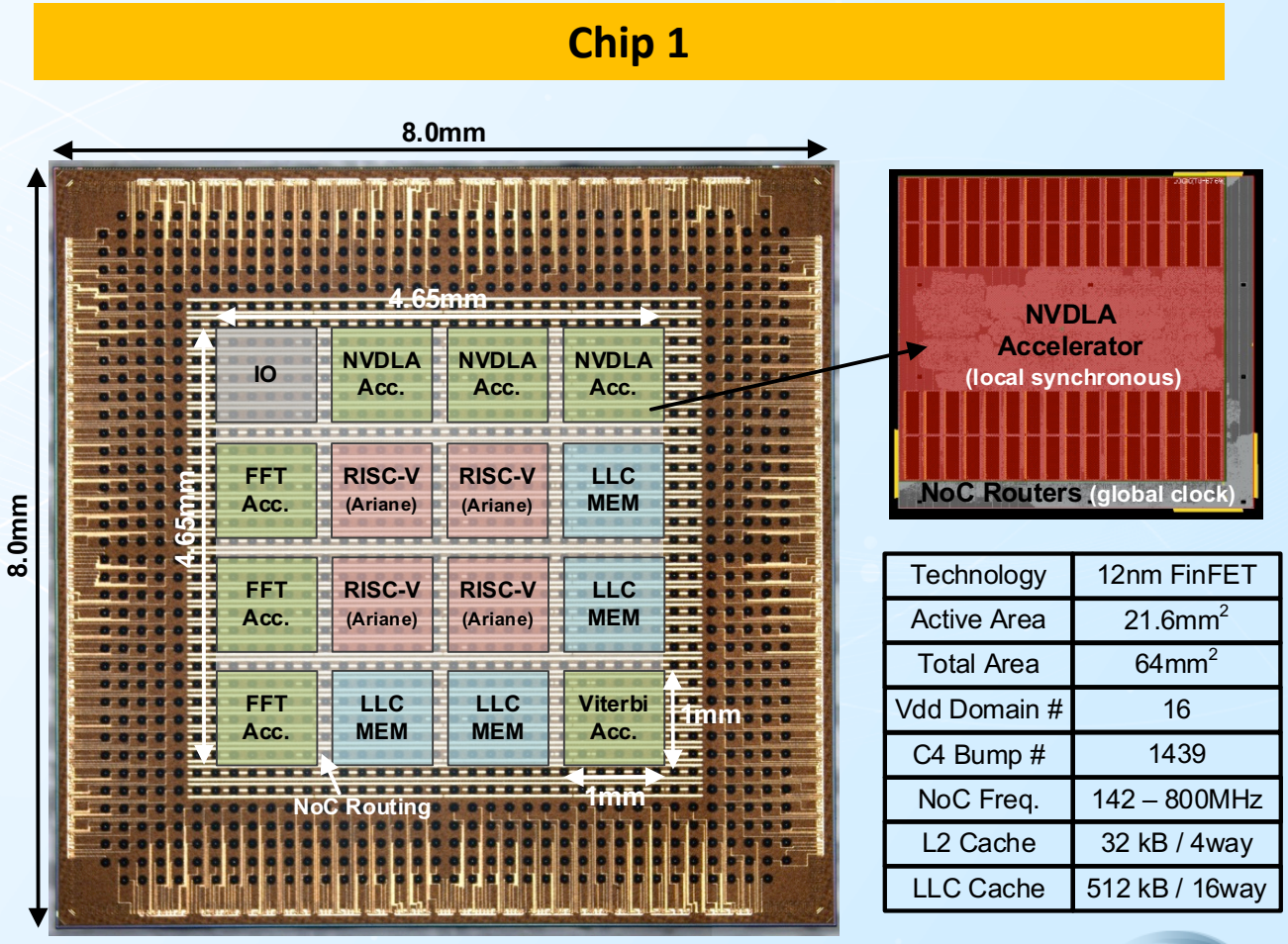


Technology	12nm FinFET
Active Area	21.6mm^2
Total Area	64mm^2
Vdd Domain #	16
C4 Bump #	1439
NoC Freq.	142 – 800MHz
L2 Cache	32 kB / 4way
LLC Cache	512 kB / 16way

T. Jia et al. (ESSCIRC 2022)

Experience with two chips designs

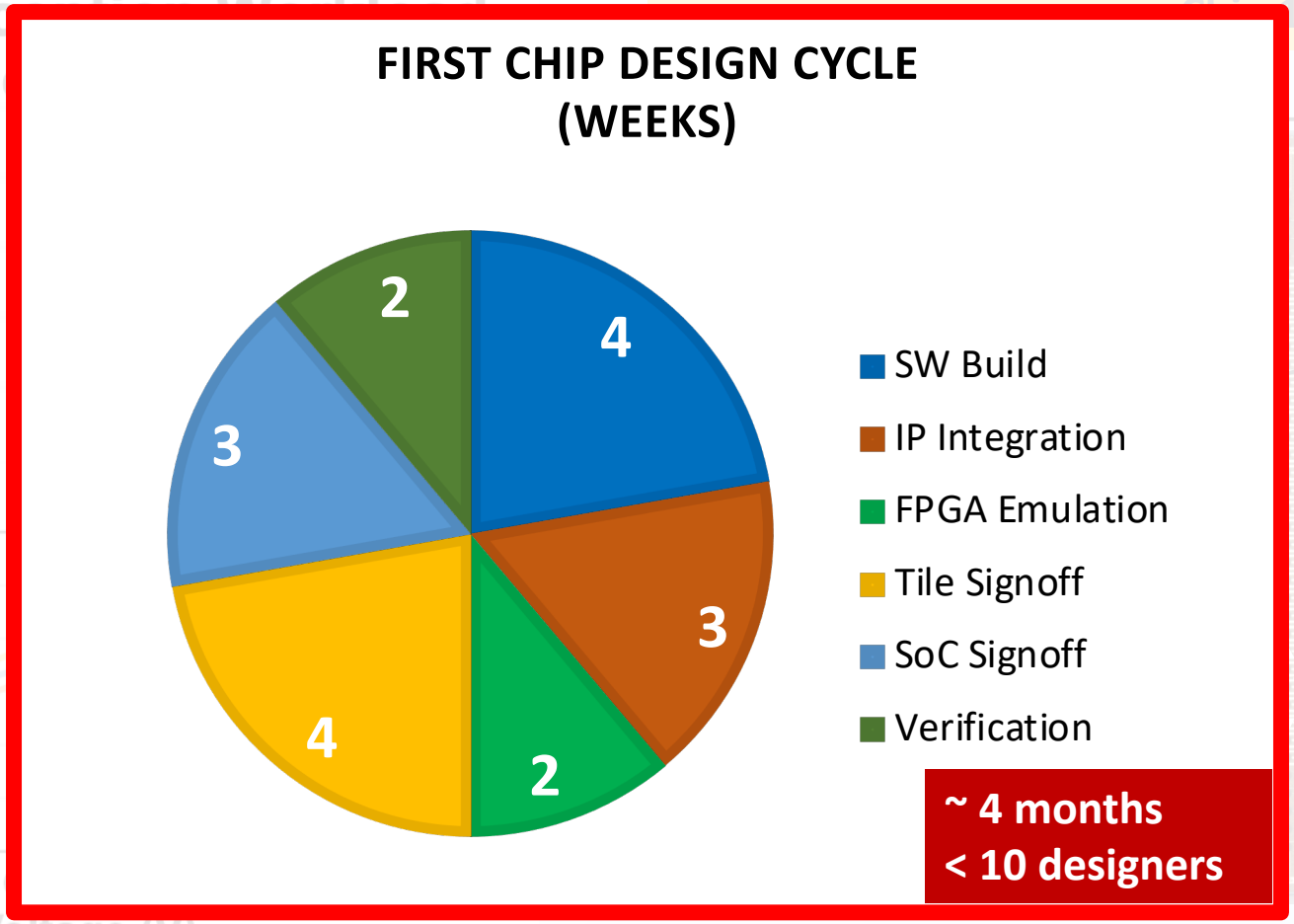




Technology	12nm FinFET
Active Area	21.6mm^2
Total Area	64mm^2
Vdd Domain #	16
C4 Bump #	1439
NoC Freq.	142 – 800MHz
L2 Cache	32 kB / 4way
LLC Cache	512 kB / 16way

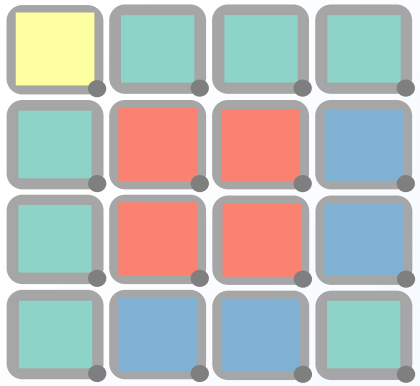
T. Jia et al. (ESSCIRC 2022)

Experience with two chips designs



Experience with two chips designs

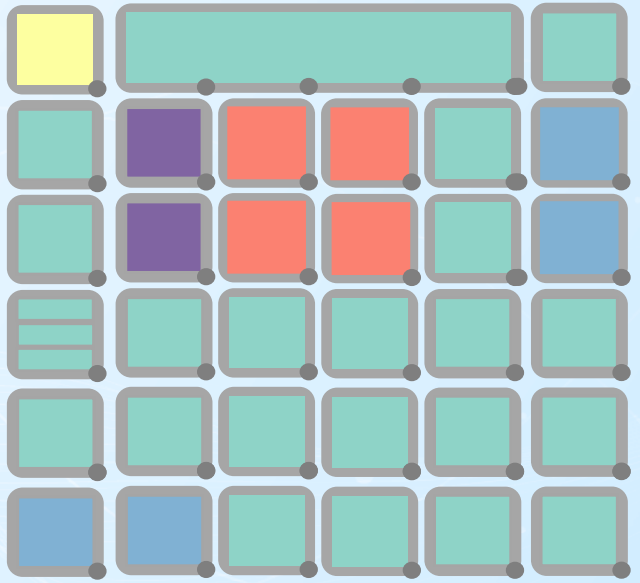
Chip 1



- 4x4 tiles
- 21.62 mm^2
- 17 clock domains
- 16 power domains
- Tile: 12 hours in 16-core 64GB RAM machine
- Top: 51 hours in 64-core 376 GB RAM machine

7 new accelerators tiles
2.25x more tiles
2.18x more clock domains
1.44x more power domains
2.96x more area
Same tile imp. running time
+29% top imp. running time

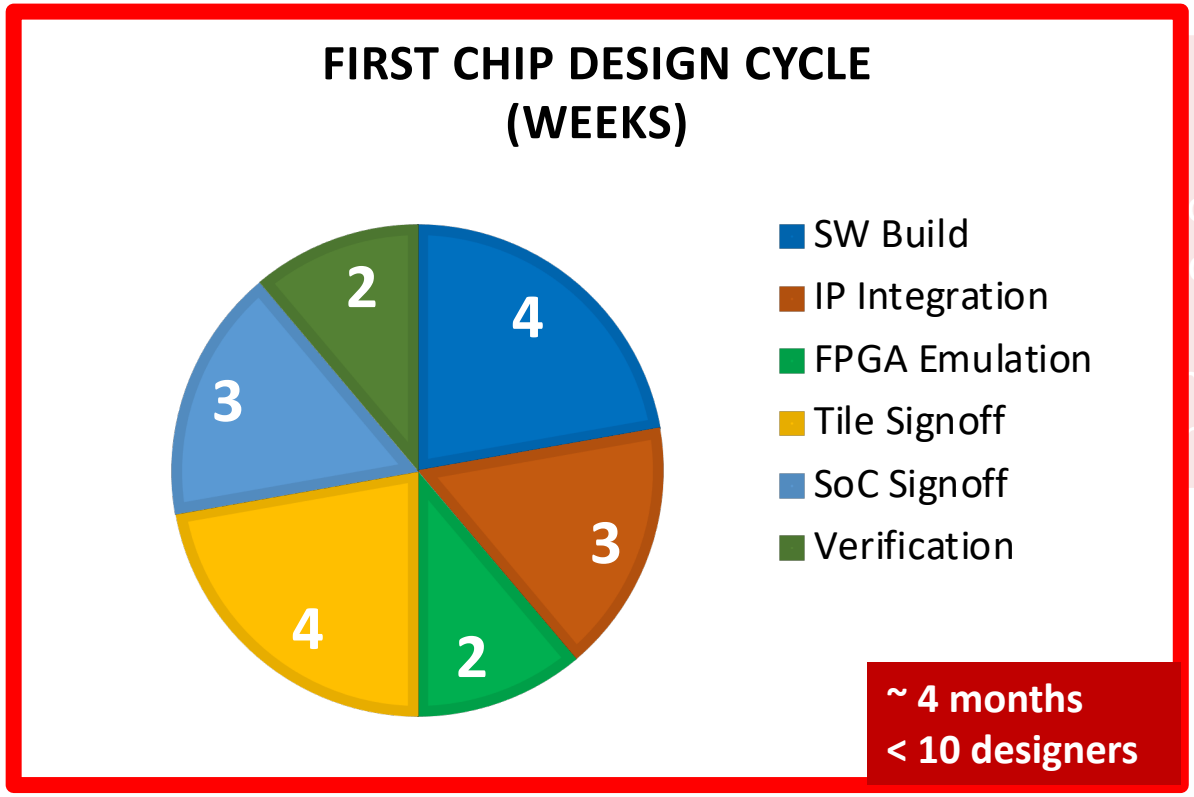
Chip 2



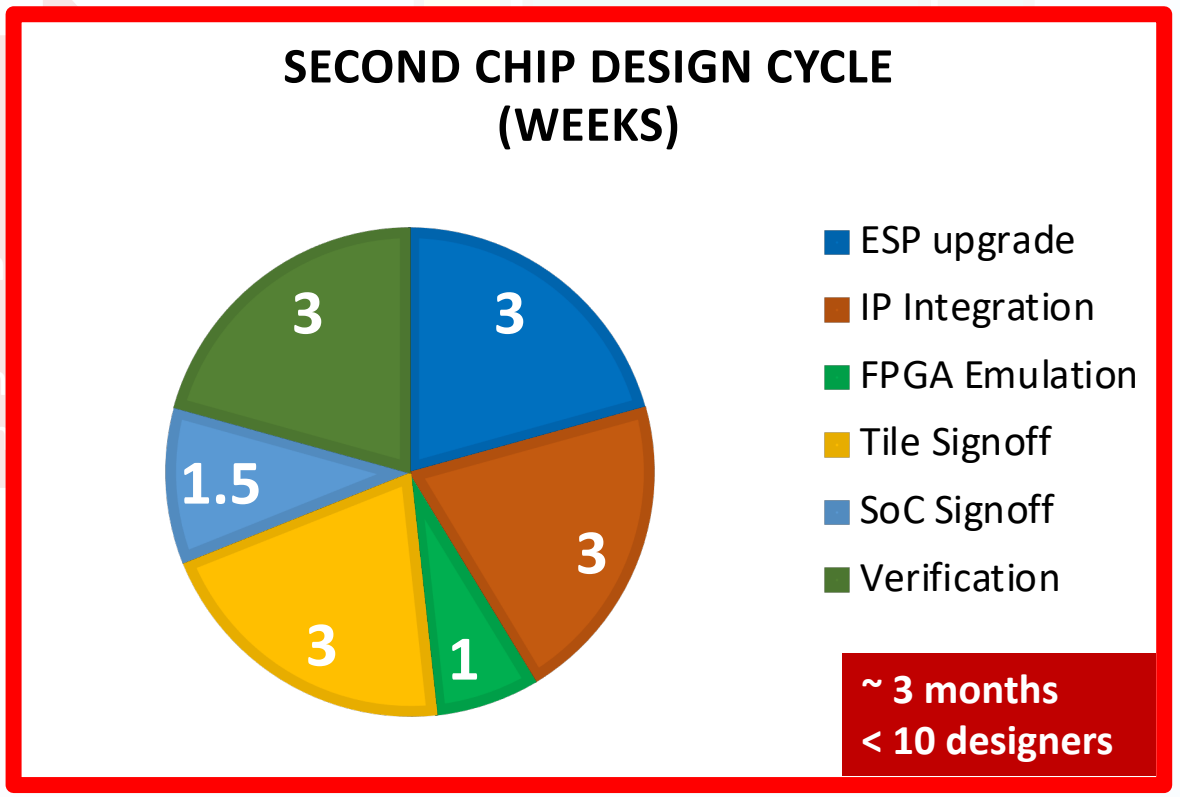
- 6x6 tiles
- 64 mm^2
- 37 clock domains
- 23 power domains
- Tile: 12 hours in 16-core 64GB RAM machine
- Top: 66 hours in 64-core 376 GB RAM machine

Experience with two chips designs

Chip 1



Chip 2



Conclusion

- Scalable, flexible and robust methodology for agile chip development with open-source hardware components
- Fully push-button OSH integration for ASIC prototyping
- Minimizes manual steps of physical design flow
- Smooth top-level integration
- Limits design computer resources as design scales
- Comprehensive verification in all design steps
- Rapid, user-friendly testing environment
- Increased productivity as demonstrated by two complex tape-outs

Thank you!

Maico Cassel dos Santos

mcassel@cs.columbia.edu