ETH zürich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

CachePool: Many-core cluster of customizable, lightweight scalar-vector PEs for irregular L2 data-plane workloads

Integrated Systems Laboratory (ETH Zürich)

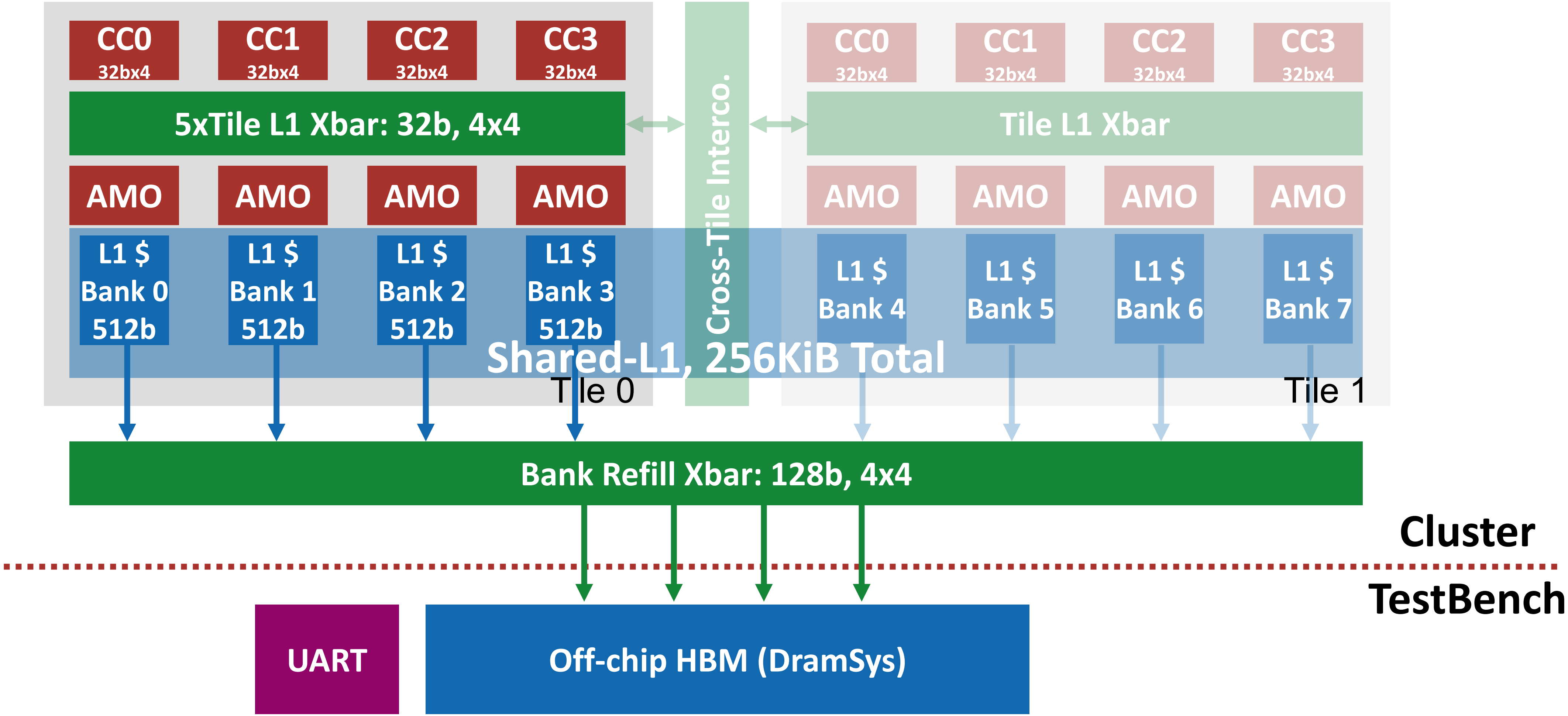
Zexin Fu, Diyou Shen zexifu, dishen@iis.ee.ethz.ch

Alessandro Vanelli-Coralli avanelli@iis.ee.ethz.ch
Luca Benini lbenini@iis.ee.ethz.ch

PULP Platform
Open Source Hardware, the way it should be!

@pulp_platform
pulp-platform.org
youtube.com/pulp_platform

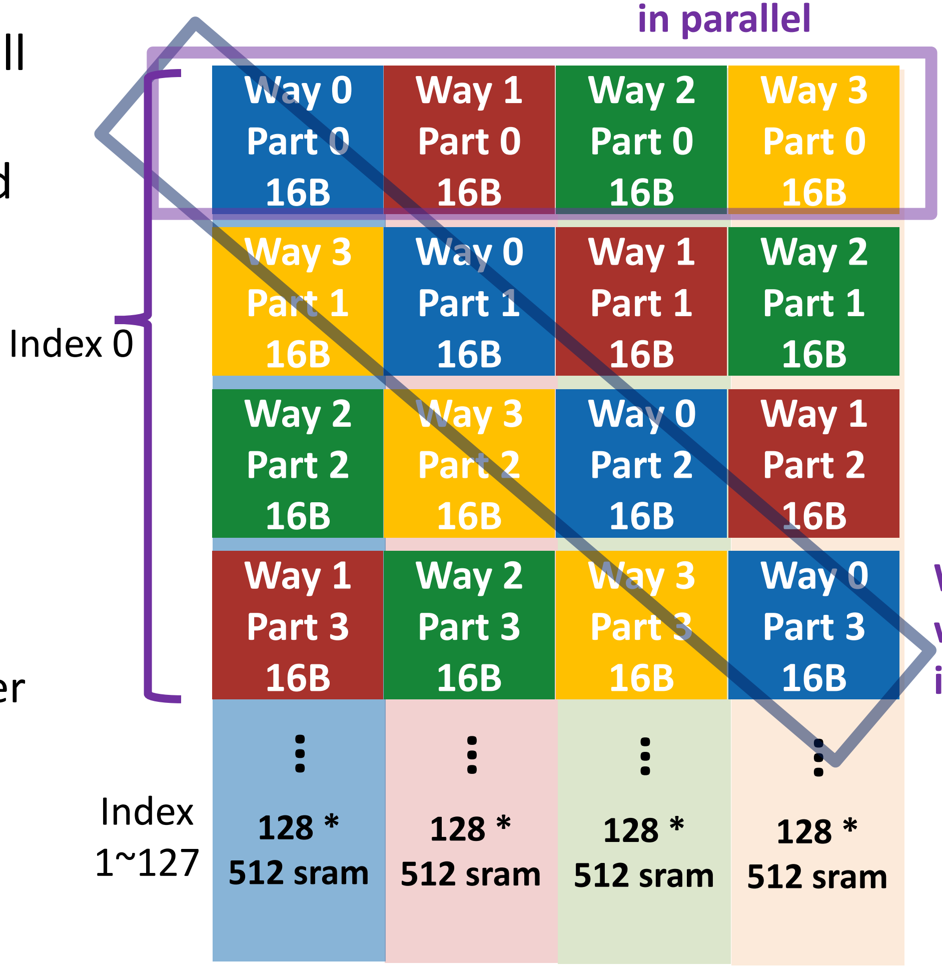
Hardware Development



Task 1: Cache Optimization

- **Folded Data SRAM**
  - **Idea**: Store the same slice (e.g., first 16 B) of all ways in one shared bank. For 4-way, can get banks that are 4× deeper, 4× narrower instead of per-way wide (64B) macros.
  - **Performance**
    - 1 cycle refill: Write full line in 1 cycle across multiple banks.
    - 1 cycle read: Read 1 slice from all ways.
  - **Advantages**
    - **Better Area Efficiency**: Uses narrower and deeper SRAMs
    - **Better Energy Efficiency**: Only 1 slice per way accessed per lookup, not full line.

Read request -> read 1/4 of a line in parallel

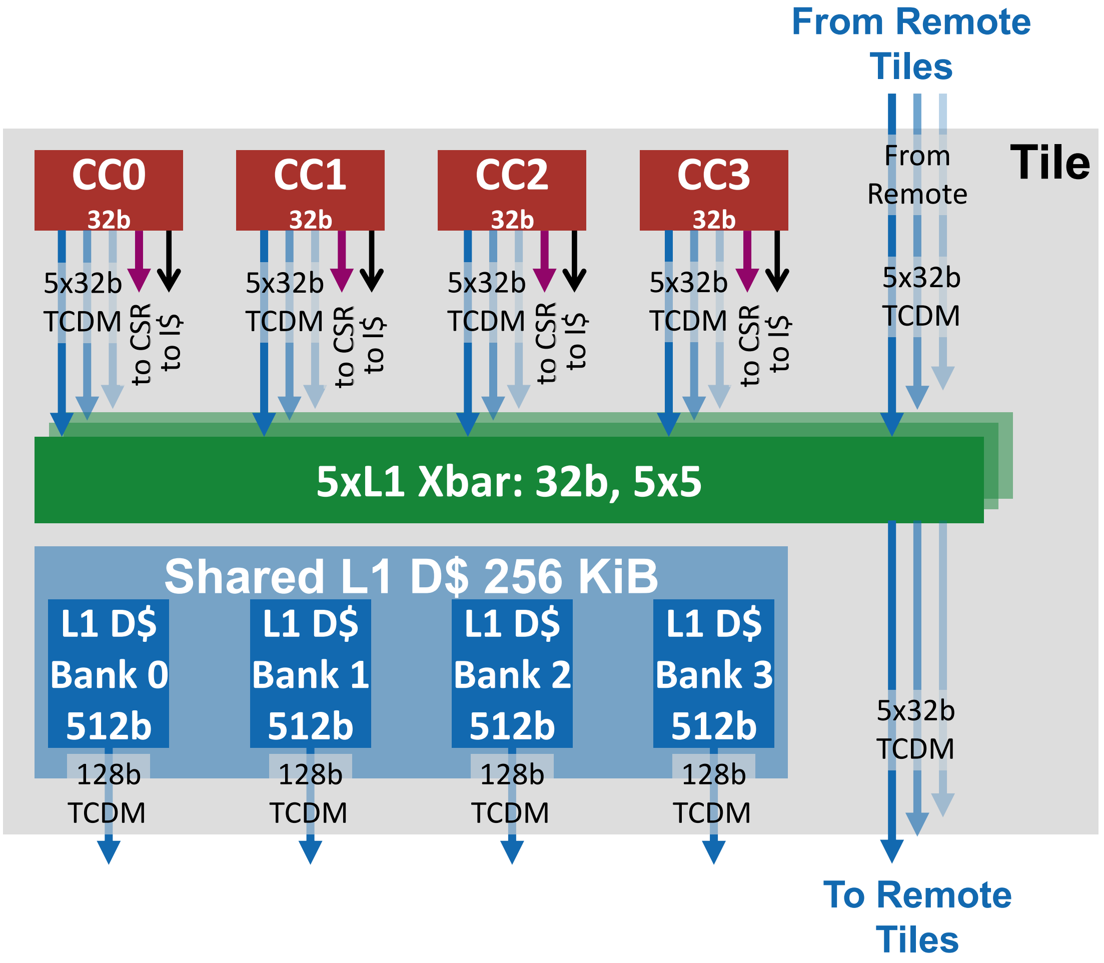


Index 0	Way 0 Part 0 16B	Way 1 Part 0 16B	Way 2 Part 0 16B	Way 3 Part 0 16B
	Way 3 Part 1 16B	Way 0 Part 1 16B	Way 1 Part 1 16B	Way 2 Part 1 16B
	Way 2 Part 2 16B	Way 3 Part 2 16B	Way 0 Part 2 16B	Way 1 Part 2 16B
	Way 1 Part 3 16B	Way 2 Part 3 16B	Way 3 Part 3 16B	Way 0 Part 3 16B
Index 1~127	⋮	⋮	⋮	⋮
	128 * 512 sram	128 * 512 sram	128 * 512 sram	128 * 512 sram

Write request -> write a full line in 1 cycle

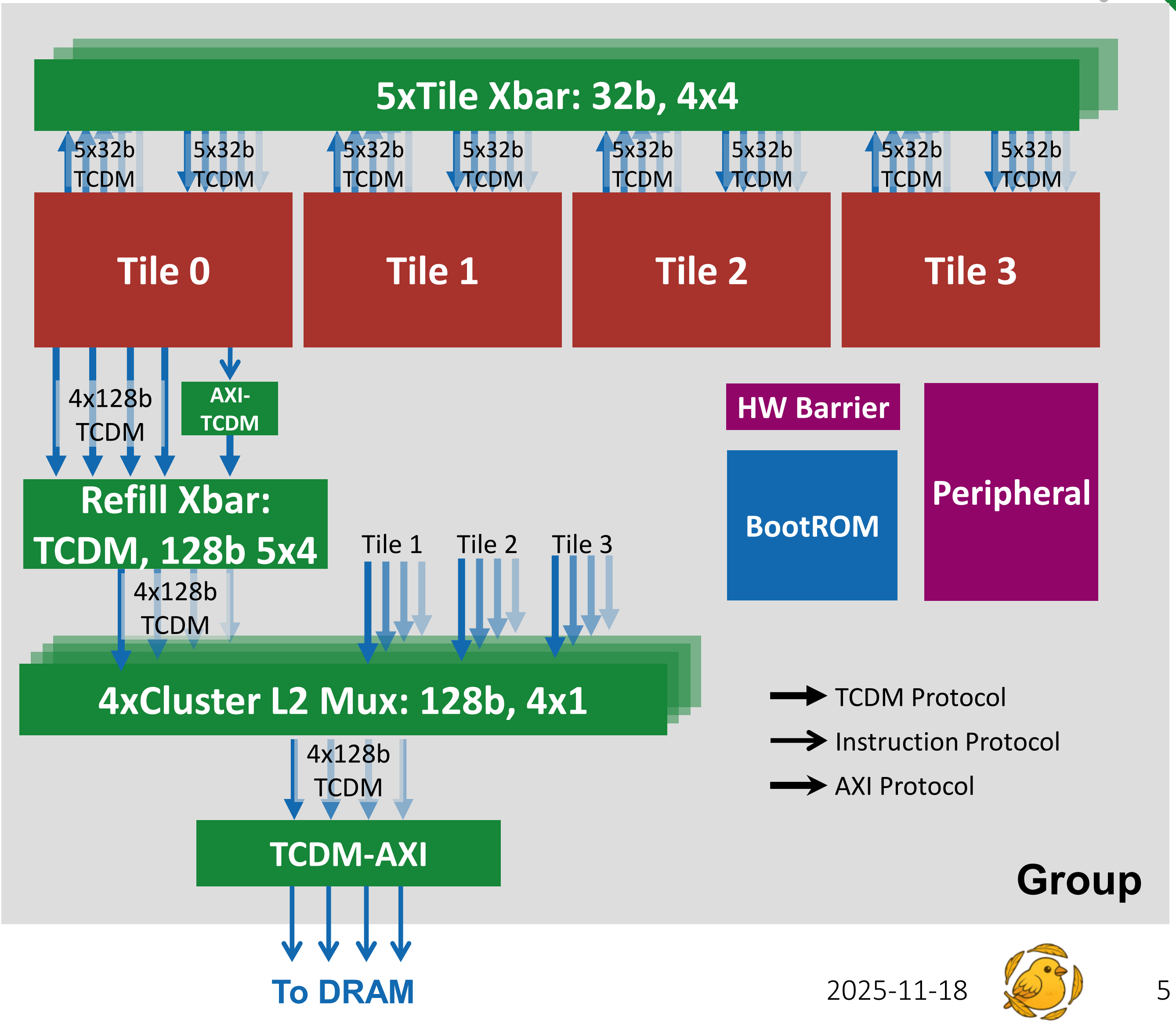
Task 2: Multi-Tile Scaling (Single Group)

- **Scale to "Group" level**
  - Current Plan: Pack 4 Tiles as a Group
  - Connected via xbar
  - Add partition support
  - **GVSoC** modeling will be in parallel developed and calibrated with RTL



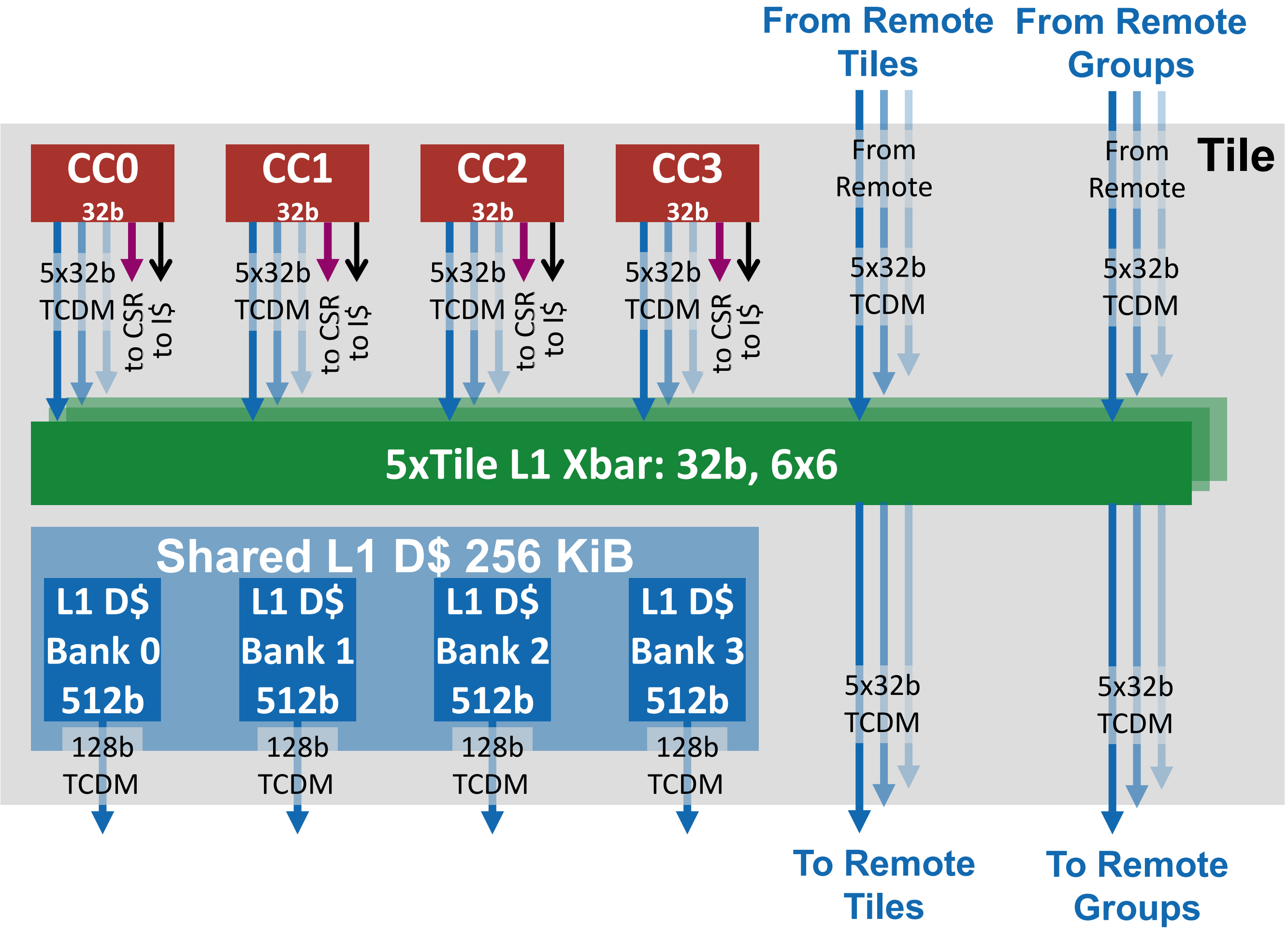
Task 2: Multi-Tile Scaling (Single Group)

- **Scale to “Group” level**
  - Current Plan: Pack 4 Tiles as a Group
  - Connected via xbar
  - Add partition support
  - **GVSoC** modeling will be in parallel developed and calibrated with RTL



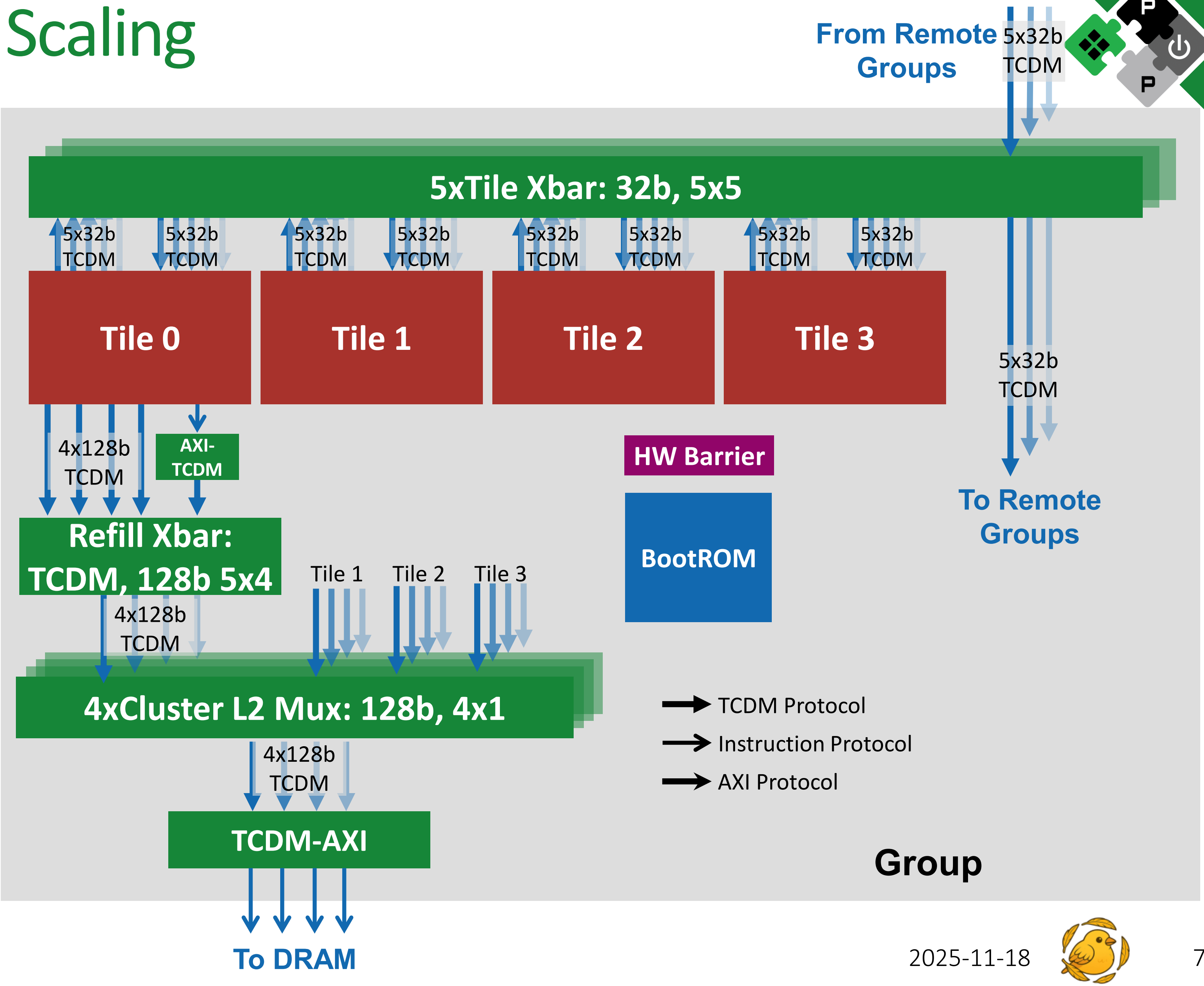
Task 3: Multi-Group Scaling

- **Scale to “Multi-Group” level**
  - On tile side: Add ports to remote groups on the tile xbar



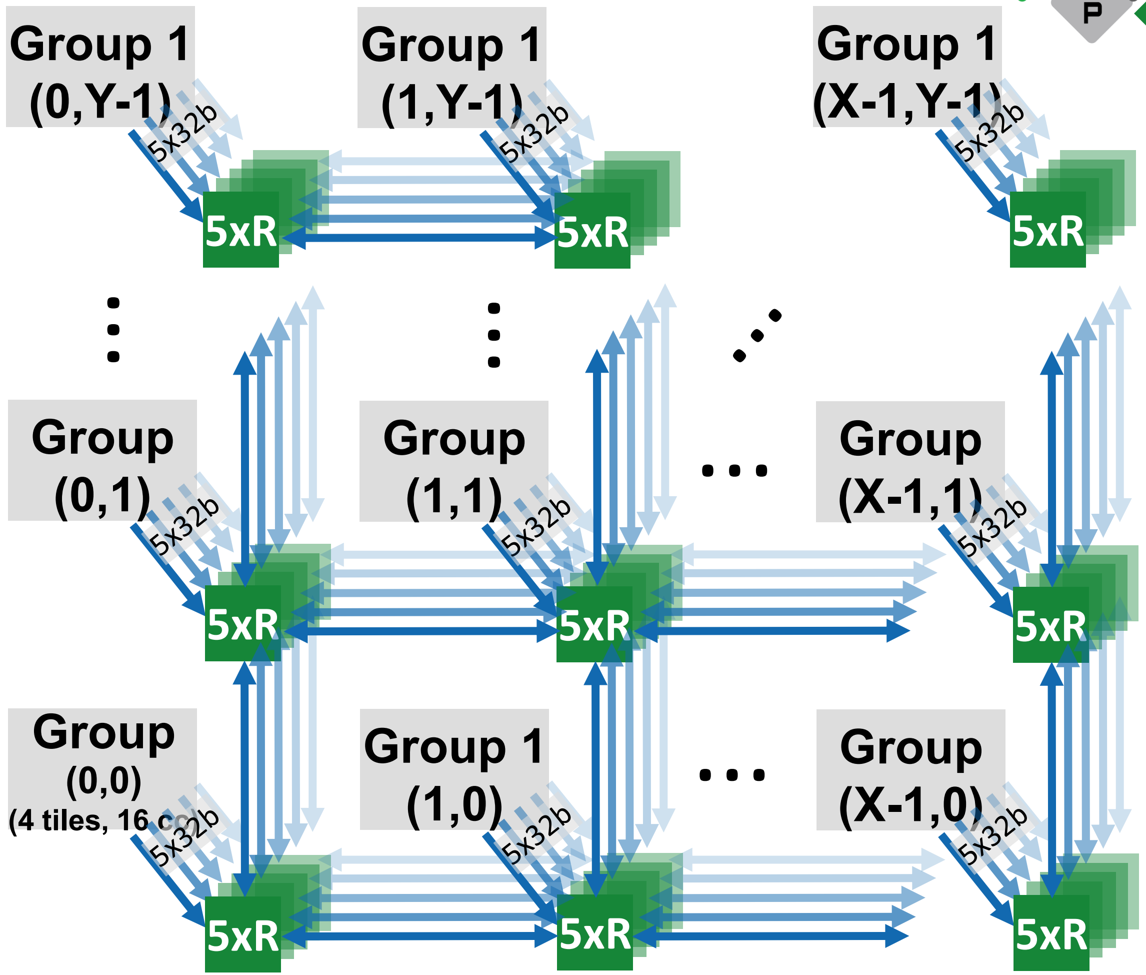
Task 3: Multi-Group Scaling

- **Scale to "Multi-Group" level**
  - On group side: Add ports to remote groups on the group xbar



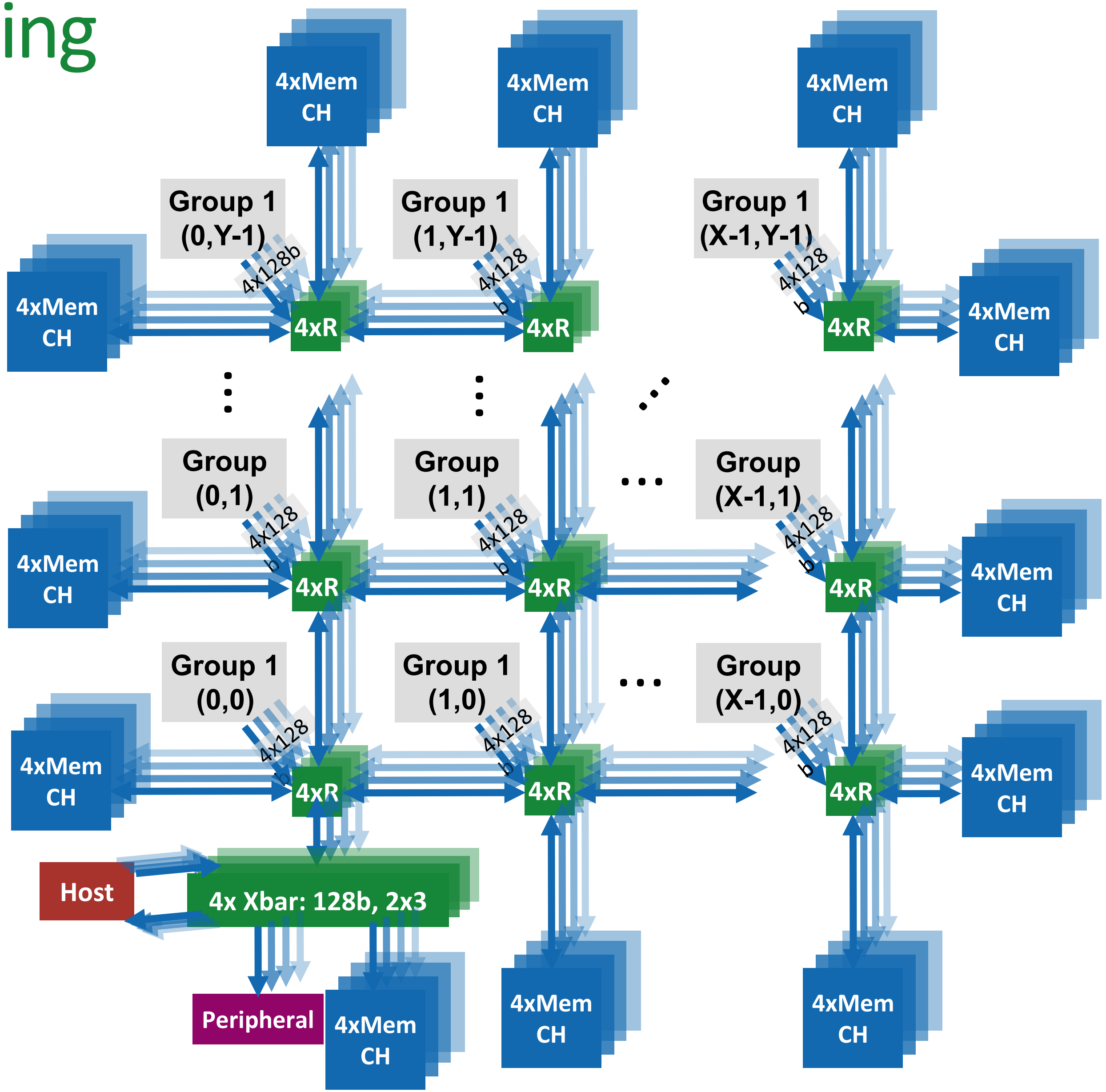
Task 3: Multi-Group Scaling

- **L1 cache NoC**
  - Connect Cores to L1$ banks
  - Mesh-based NoC, w/ package switching routers
  - Use multiple word-width (32b) sub-NoC
  - The number of sub-NoC to be explored (BW vs. physical feasibility)
- **Co-design with GVSoC**
  - GVSoC model will be developed in parallel with previous tasks
  - Early architecture verification
  - Reduce simulation time



Task 3: Multi-Group Scaling

- **Refill NoC (L2 NoC)**
  - Connect L1$ banks to memory, and cores to peripheral
  - Mesh-based NoC, w/ package switching routers
  - Use wide NoC (128b), the number of NoC to be explored (BW vs. physical feasibility)



Task 4: Design Space Exploration

- **Use the same structure as in Task 3**
  - Increasing Group counts => try to scale beyond 500 cores
  - Heterogenous tile design
    - Increase number of Snitch cores inside tile
  - Use RTL + GVSoC performance model to explore the sweetpoint configuration

Task 5: Software Development

- **In parallel with all tasks**
- **RLC Kernel**
  - Test and evaluate all user cases
  - Optimize algorithm
- **Vector Kernel**
  - Finish and benchmark the remaining kernels
  - Combine into a kernel chain

Timeline

		2025		2026										
		11	12	1	2	3	4	5	6	7	8	9	10	11
Task 1	Cache Controller Optimization [RTL]	🟥	🟥	🟥	🟥									
Task 2	Multi-Tile Scaling (Group) [RTL]	🟧	🟧	🟧	🟧	🟧	🟧							
Task 3	4/16-Group Scaling [RTL & GVSoC]				🟩	🟩	🟩	🟩	🟩	🟩				
Task 4	Design Space Exploration [RTL & GVSoC]							🟦	🟦	🟦	🟦	🟦		
Task 5	Software Development	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪		
	Report											🟩	🟩	

Thank you!

Q&A