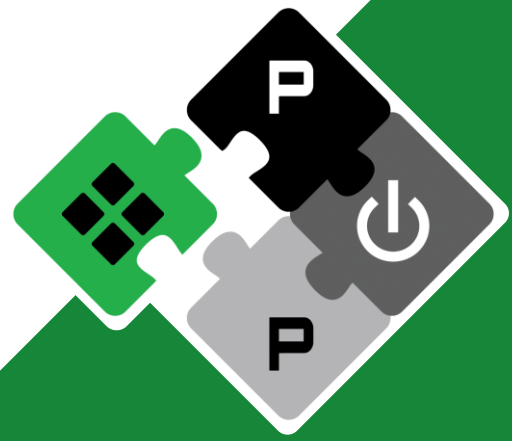
ETH zürich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

NextRAN-AI – 11/07/2025

Integrated Systems Laboratory (ETH Zürich)

Marco Bertuletti mbertuletti@iis.ee.ethz.ch
Yichao Zhang yiczhang@iis.ee.ethz.ch
Mahdi Abdollahpour mahdi.abdollahpour@unibo.it
Alessandro Vanelli-Coralli avanelli@iis.ee.ethz.ch
Luca Benini lbenini@iis.ee.ethz.ch

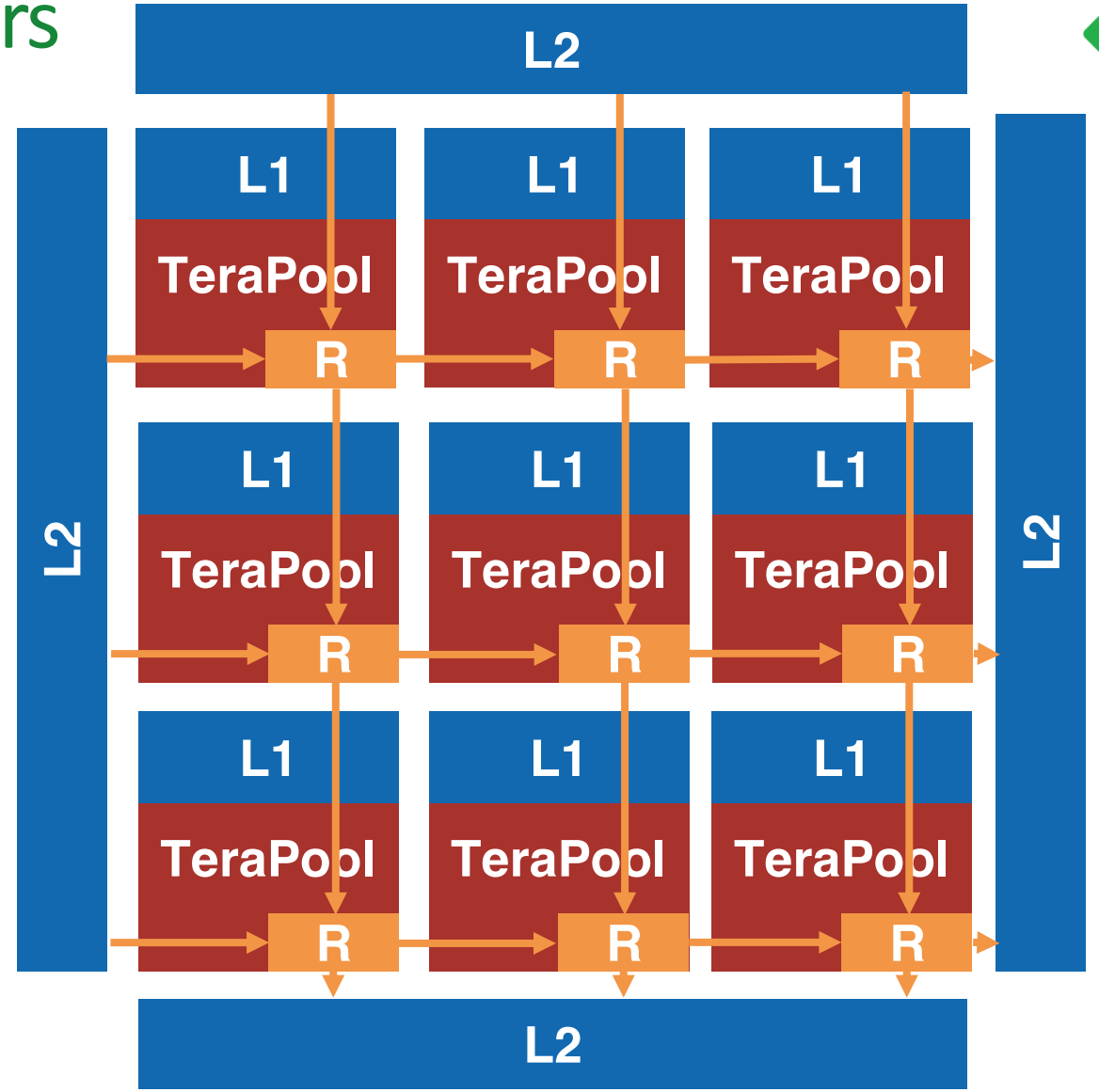


pulp-platform.org
@pulp_platform
company/pulp-platform
youtube.com/pulp_platform

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

Required System Parameters

- **System Design Parameters:**
  - Cluster peak-performance
  - Cluster L1 memory size
  - Cluster-NoC BW
- **Cluster interconnect design**



Evaluate Number of Clusters from FLOPs

- **14 models running in parallel, each with 4.2B compute workload**
- **TeraPool has 1024 cores with SIMD capabilities**
- **Replace Tiles with RedMulE-Tiles, containing one RedMulE and a core**
- **HP:** f16 floating-point, full utilization, and no overhead of transfers
- **HP:** the Physical Design in 7nm will run at 1GHz (920MHz in 12nm)

Configuration	NumCores	NumTiles	NumRMTiles	FLOPS/RedMulE
0RM	1024	128	0	64
16RM, 8x8	1024	128	16	64
32RM, 8x8	1024	128	32	64
64RM, 8x8	1024	128	64	64
16RM, 16x16	1024	128	16	256

$$N_{clusters} = 14 \times \frac{FLOPS}{FLOPS_{cluster}}$$

Evaluate WC Memory Footprint

- **HP: Model executed on a single cluster (worst-case for memory footprint)**
- **HP: Output buffer for layer-by-layer execution is used as input of next layer**
- **Maximum memory footprint evalued on longest skip-connection**
  - Evaluate input-size, weight-size, output-size for each layer, evaluate skip-connection size
  - Evaluate the maximum footprint for the longest skip-connection

$$\sum_{l=0}^{L} W_l + \max_{l \in [0,L]} (I_l + O_l) + Sk$$

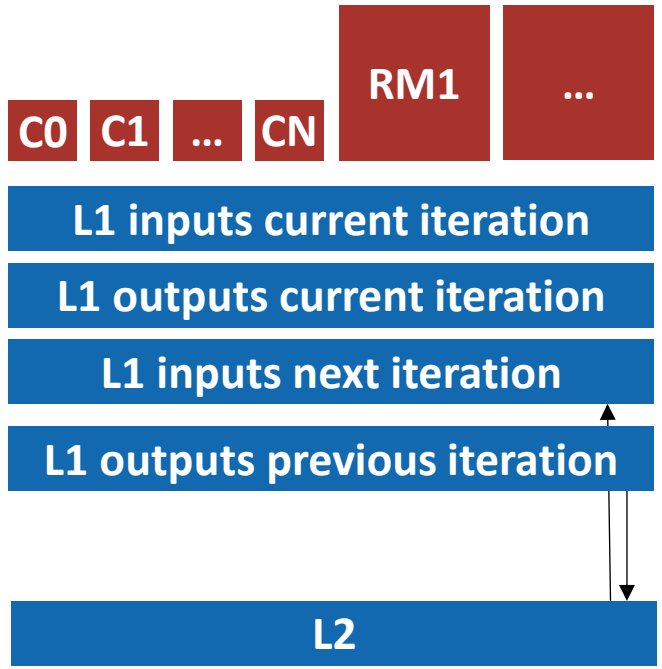
Skip-connection size

Sum of weight-sizes for all layers

Maximum of layer input-size & output-size sum

Evaluate WC Cluster-NoC BW

- **Cluster-NoC BW evaluated in WC to balance cluster peak-performance**
  - Assume no data-reuse (single-operator executed)
  - Assume operator with minimum data-reuse
  - Assume double-buffering
- **Three WC scenarios:**
  - Case 1: only cores are active
  - Case 2: only RedMulEs are active
  - Case 3: cores and RedMulEs are active
    (e.g. MatMul/Conv1D + Activation)



Worst case would be the **tensor addition**, but there is no case where we do it standalone, so we take the smallest **matmul**

Evaluate WC Cluster-NoC BW : Case1/2

- **Only RedMulEs/cores are working:**
  - There is data-reuse: each element of matrices A and B is used twice
  - Minumum data reuse for minum size of the attention matrix (32)

We assume 16b data

$$T_{transfer} = \frac{2 \times (MN + NP + MP)}{BW}$$

$$T_{compute} = \frac{MNP}{FLOPS_{RM/cores}}$$

$$BW = \frac{FLOPS_{RM/cores} \times 3N^2}{N^3} = \frac{6 \times FLOPS_{RM/cores}}{N}$$

We assume squared matrices

Evaluate WC Cluster-NoC BW : Case3

- **Cores and RedMulEs are both working:**
  - RedMulE executes a GEMM, cores execute Activations
  - There is data dependency: the output of the GEMM goes to Activation input
  - Operations are pipelined in L1

$$T_{transf.} = \frac{6 \times N^2}{BW}$$

Activation

Matrix multiplication

$$T_{compute} = \max\left(\frac{N^3}{FLOPS_{RM}}, \frac{N}{FLOPS_{cores}}\right)$$

$$BW = \frac{6 \times N^2}{\max\left(\frac{N^3}{FLOPS_{RM}}, \frac{N}{FLOPS_{cores}}\right)}$$

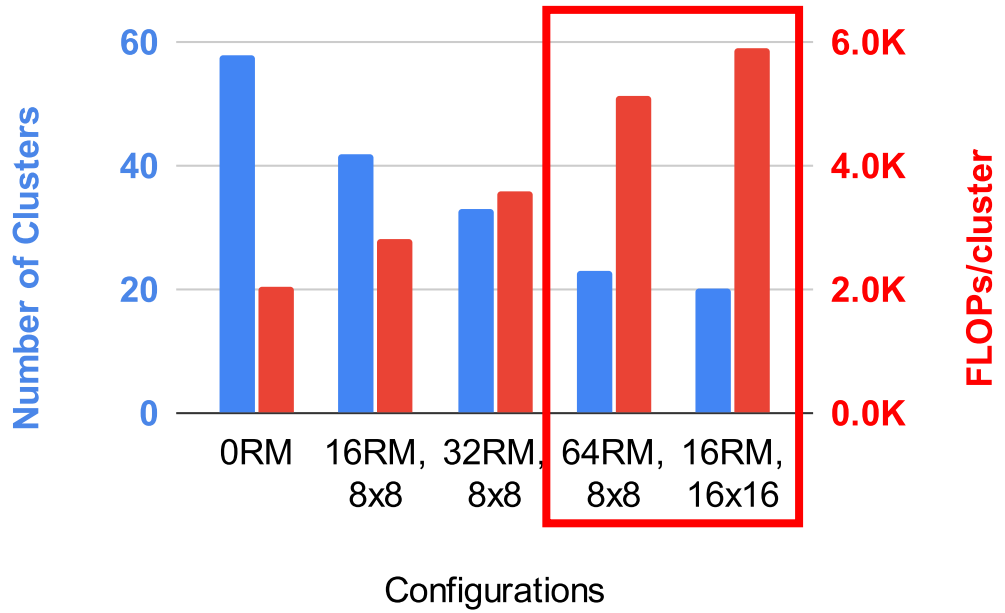
Conv1D can be run as MatMul

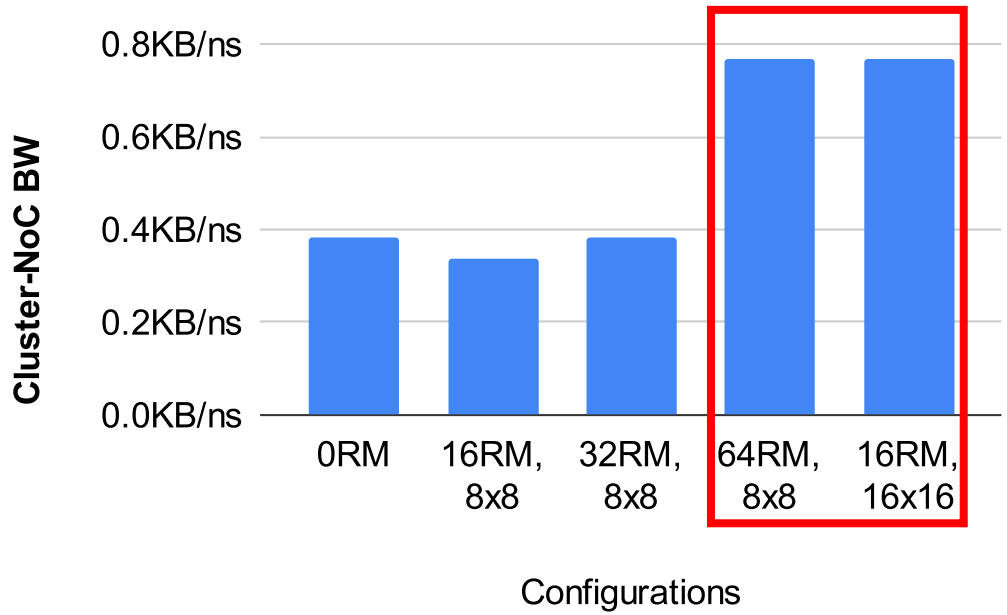
- Inner loop can be moved to outmost position:
  - Same degree data reuse (all tensors in L1)
  - Accumulation over the filter dimension
- Conv1D + activation is equivalent to MatMul + activation

```
/* Conv1D inner loops */

for (uint32_t i = 0; i < Cout; i++) {
 for (uint32_t j = 0; j < W; j++) {
  for (uint32_t k = 0; k < Cin; k++) {
   for (uint32_t f = 0; f < Wf; f++) {
    Y[i][j] += F[i][k][f] * X[k][j - Wf/2 + f];
   }
  }
 }
}

for (uint32_t f = 0; f < Wf; f++) {
 for (uint32_t i = 0; i < Cout; i++) {
  for (uint32_t j = 0; j < W; j++) {
   for (uint32_t k = 0; k < Cin; k++) {
    Y[i][j] += F[i][k][f] * X[k][j - Wf/2 + f];
   }
  }
 }
}
```

Required System Parameters

Cluster Memory	> 3.6 MB





- Only 2 configurations have >4kFLOPs/cycle and reasonable Number of Clusters
- But the 64RM 8x8 configuration injects more BW in the cluster interconnect

TeraPool + RedMulE = TensorPool configurations

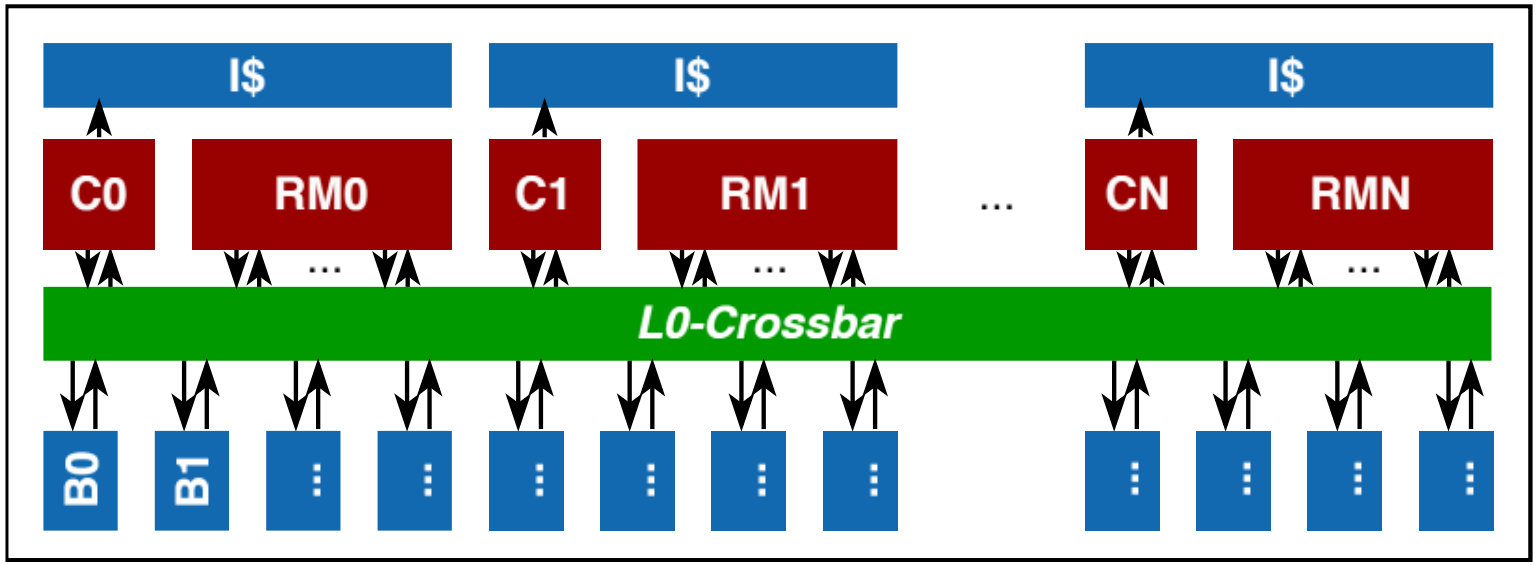
Connecting multiple RedMulE Tiles and Snitch Tiles to build a 4kFLOPs/cycle cluster with shared memory:

- How much area/power can we invest in flexibility (Snitch Tiles)?
- How do we build the Tiles to memory interconnects to achieve good BW to RedMulE?

We build the cluster interconnect under these <u>Hypotheses</u>:

- We select the 16x16 RedMulE as the most cost-effective (FLOPs/BW = 512b)
- We ensure that the RedMulE ports can have full BW to a subset of banks
- We constrain the maximum interconnect complexity to 32x32
- If a hierarchical level > Tile is added we use a factor 4x to esure floorplan central simmetry

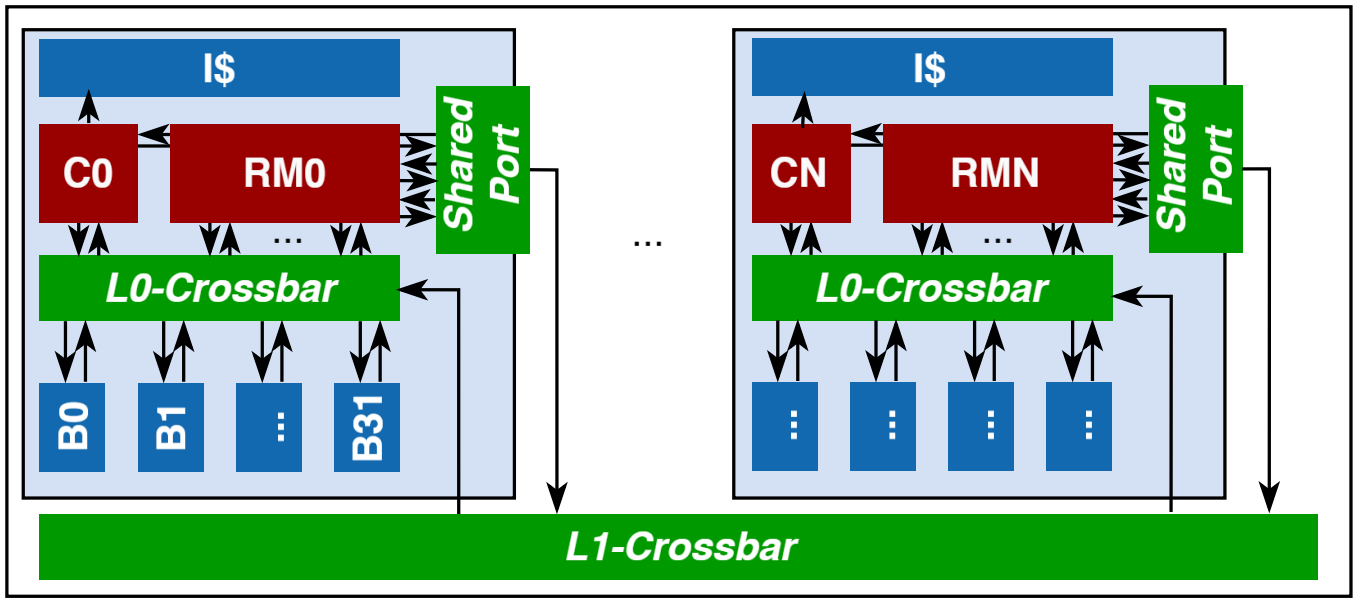
Flat Tile with only RedMulEs & NO Snitches



- One hierarchy of crossbars
- RedMulE has 32x32b ports = assume 32 banks per RedMulE → 512 banks of shared memory
- Not feasible 512x512 connection

FLOPS	NCores	NRM	NBank /Tile	NRM Tiles	LV1RM Ports	LV2RM Ports	NTiles	NG	NSGs	LEVEL 0		LEVEL 1		LEVEL 2	
										INxOUT	N	INxOUT	N	INxOUT	N
4096	0	16	512	1	0	0	0	0	0	512x512	1	-	-	-	-

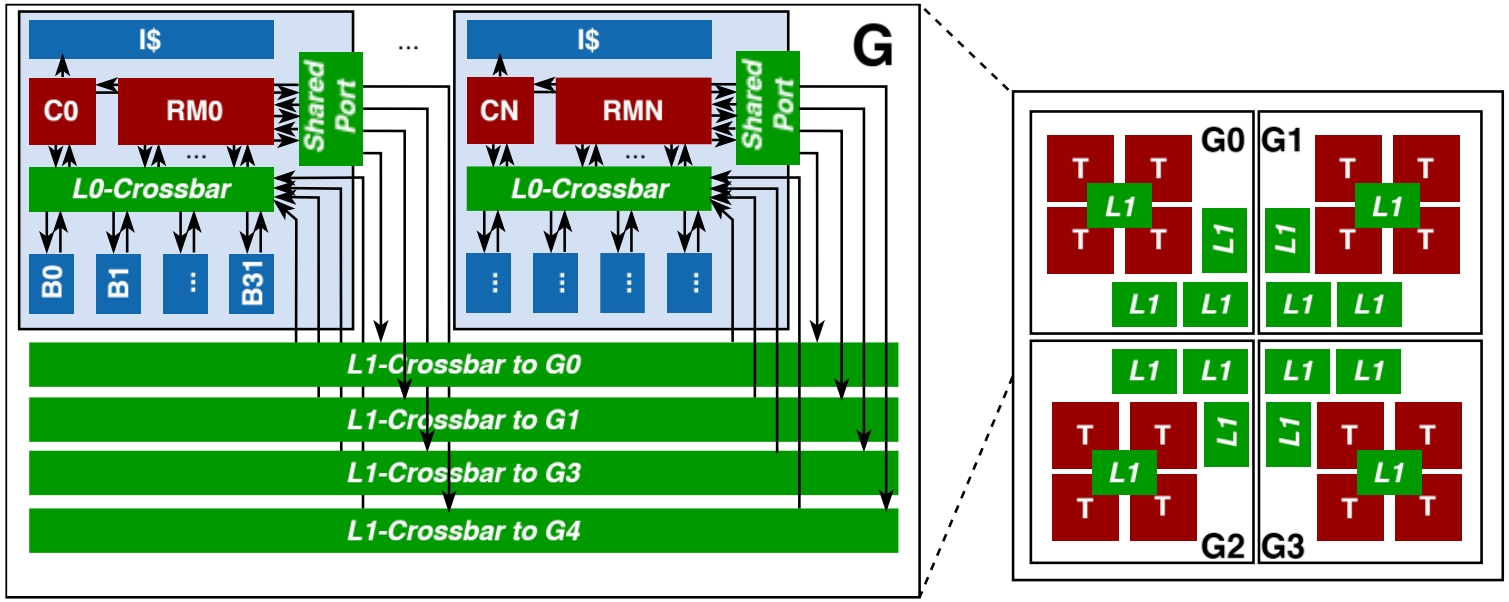
Flat Group with 8/16 RedMulE Tiles & NO Snitch Tiles



- A Tile with 2 RedMulEs is not possible → 64x64 L0
- We can add a Tile port → 32x32 xbars in the Group level
- But still poor RedMulE BW

FLOPS	NCores	NRM	NBank /Tile	NRM Tiles	LV1RM Ports	LV2RM Ports	NTiles	NG	NSGs	LEVEL 0		LEVEL 1		LEVEL 2	
										INxOUT	N	INxOUT	N	INxOUT	N
~~4096~~	~~0~~	~~16~~	~~512~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~512x512~~	~~1~~	-	-	-	-
4096	0	16	32	8	1	0	0	1	0	64x64	8	8x8	1	-	-
4096	0	16	32	16	1	0	0	1	0	32x32	16	16x16	1	-	-
4096	0	16	32	16	2	0	0	1	0	32x32	16	32x32	1	-	-

4 Group with 4 RedMulE Tiles & NO Snitch Tiles

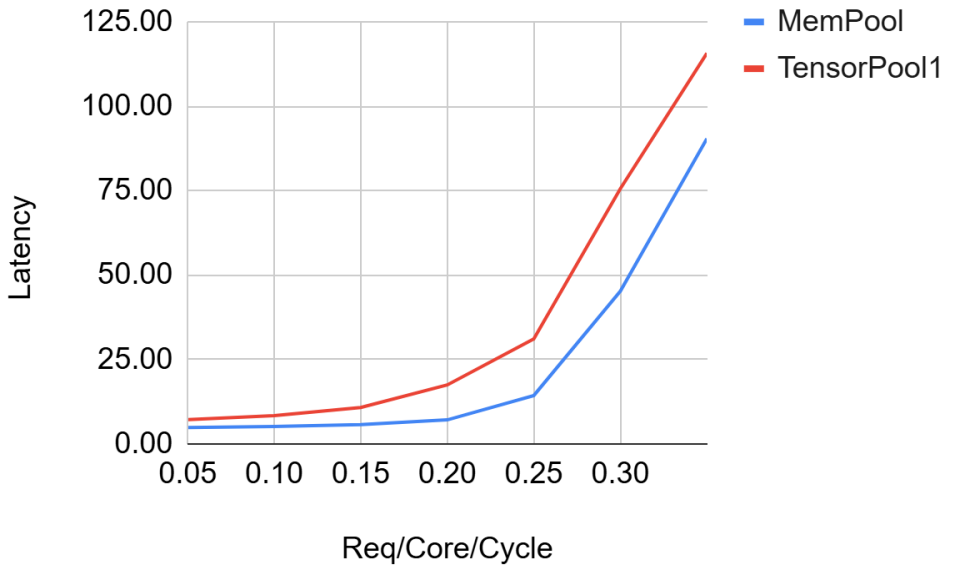


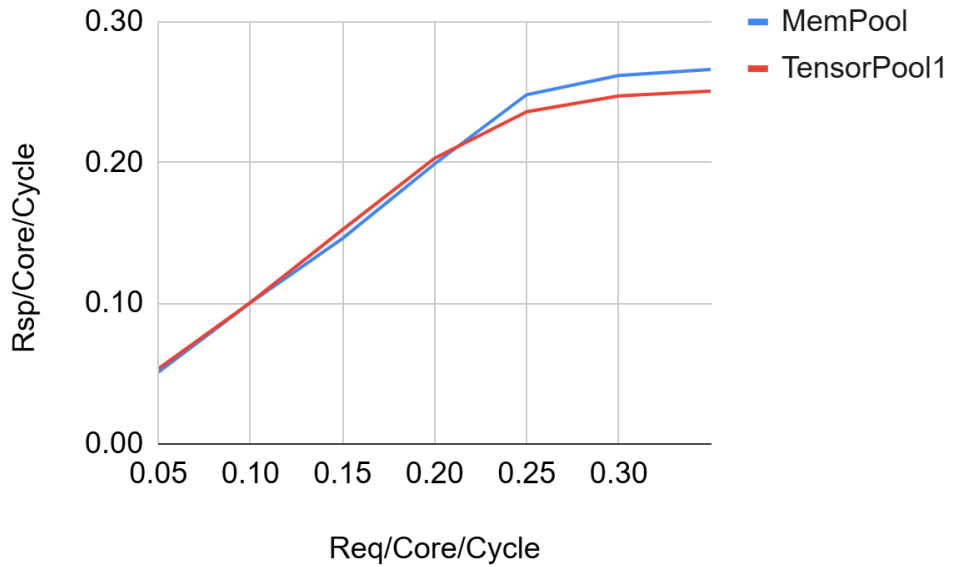
Best option is highlighted and increases the BW of RedMulE

FLOPS	NCores	NRM	NBank /Tile	NRM Tiles	LV1RM Ports	LV2RM Ports	NTiles	NG	NSGs	LEVEL 0		LEVEL 1		LEVEL 2	
										INxOUT	N	INxOUT	N	INxOUT	N
~~4096~~	~~0~~	~~16~~	~~512~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~512x512~~	~~1~~	-	-	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~8~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0~~	~~64x64~~	~~8~~	~~8x8~~	~~1~~	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~16~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0~~	~~32x32~~	~~16~~	~~16x16~~	~~1~~	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~16~~	~~2~~	~~0~~	~~0~~	~~1~~	~~0~~	~~32x32~~	~~16~~	~~32x32~~	~~1~~	-	-
4096	0	16	32	16	1	0	0	4	0	32x32	16	4x4	16	-	-
4096	0	16	32	16	8	0	0	4	0	32x32	16	32x32	16	-	-

WIP semi-analytical model of the interconnect

- TensorPool-1 has maximum throughput for random requests = 0.25
- For continuous operation of the accelerator (weights are loaded every 4 cycles i.e. Req/Port/Cycle = 0.25), we must tolerate 10 cycles average latency





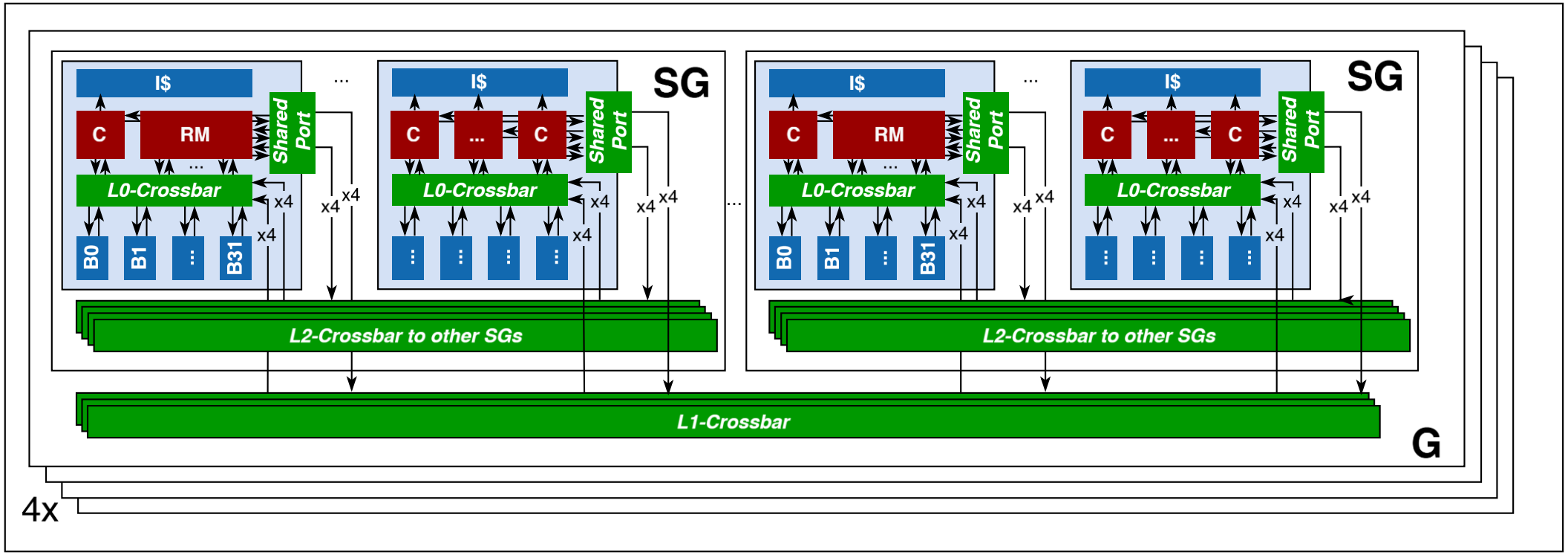
Add Snitch Tiles

If you add Snitch Tiles in the Group:

- Reduce the number of RedMulE ports per Tile in the Group interconnect to still get 32x32
- Complicated floorplan (16 Snitch Tiles + 4 RedMulE Tiles each Group)
- Solution = add a new hierarchy level

FLOPS	NCores	NRM	NBank /Tile	NRM Tiles	LV1RM Ports	LV2RM Ports	NTiles	NG	NSGs	LEVEL 0		LEVEL 1		LEVEL 2	
										INxOUT	N	INxOUT	N	INxOUT	N
~~4096~~	~~0~~	~~16~~	~~512~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~512x512~~	~~1~~	-	-	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~8~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0~~	~~64x64~~	~~8~~	~~8x8~~	~~1~~	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~16~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0~~	~~32x32~~	~~16~~	~~16x16~~	~~1~~	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~16~~	~~2~~	~~0~~	~~0~~	~~1~~	~~0~~	~~32x32~~	~~16~~	~~32x32~~	~~1~~	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~16~~	~~1~~	~~0~~	~~0~~	~~4~~	~~0~~	~~32x32~~	~~16~~	~~4x4~~	~~16~~	-	-
4096	0	16	32	16	8	0	0	4	0	32x32	16	32x32	16	-	-
5120	512	16	32	16	4	0	48	4	0	32x32	80	32x32	16	-	-

4SG/G, 4G, RedMulE Tiles + Snitch Tiles in each SG



- The Tiles in SG are connected with local xbars
- The other SGs in a G are connected with 3 remote-SG xbars
- The Tiles in a G are connected to Tiles in other Gs with 3 remote-G xbars

TeraPool + RedMulE = **TensorPool** configurations

FLOPS	NCores	NRM	NBank /Tile	NRM Tiles	LV1RM Ports	LV2RM Ports	NTiles	NG	NSGs	LEVEL 0		LEVEL 1		LEVEL 2	
										INxOUT	N	INxOUT	N	INxOUT	N
~~4096~~	~~0~~	~~16~~	~~512~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~512x512~~	~~1~~	-	-	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~8~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0~~	~~64x64~~	~~8~~	~~8x8~~	~~1~~	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~16~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0~~	~~32x32~~	~~16~~	~~16x16~~	~~1~~	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~16~~	~~2~~	~~0~~	~~0~~	~~1~~	~~0~~	~~32x32~~	~~16~~	~~32x32~~	~~1~~	-	-
~~4096~~	~~0~~	~~16~~	~~32~~	~~16~~	~~1~~	~~0~~	~~0~~	~~4~~	~~0~~	~~32x32~~	~~16~~	~~4x4~~	~~16~~	-	-
4096	0	16	32	16	8	0	0	4	0	32x32	16	32x32	16	-	-
5120	512	16	32	16	4	0	64	4	0	32x32	80	32x32	16	-	-
5120	512	16	32	16	4	8	48	4	16	32x32	64	28x28	12	11x11	64

- Simpler floorplan = only 1 RedMulE Tile and 4 Snitch Tiles per SG
- More RedMulE BW to local SubGroups (16 ports)*

* We are working on a semi-analytical model to see if the BW is good enough for continuous operation of the accelerator.

TensorPool area estimation

	Assumption: Cluster Physical Die Area in TSMC N7			
		TeraPool	TensorPool-1	TensorPool-2
SubGroup	NRMTiles/SG	0	0	1
	NSitchTiles/SG	8	0	4
	Nxbars/SG	4	0	4
	AreaSG [mm2]	1.31	0.00	0.842
Group	NRMTiles/G	0	4	0
	NSitchTiles/G	0	0	0
	NSG/G	4	0	4
	Nxbars/G	3	4	3
	AreaG [mm2]	9.03	1.66	4.78
Cluster	NG/C	4	4	4
	AreaC [mm2]	54.15	9.94	28.73
	Delta Area	1.00	0.18	0.53
System 32 clusters	Area [mm2]	1,732.94	317.97	919.32

- RedMulE area in 7nm and TeraPool area in 12nm + technology scaling
- Correction factor applied to the Snitch Tile & RedMulE Tile to take into account placement and routing of the interconnects

TensorPool energy-efficiency estimation

- Technology scaling from power measuremens
  (MatMul) in GF22 (RedMulE) and GF12 (TeraPool)
- Weighted average on the number of RedMulEs and Snitch cores
- Configurations with more cores have less efficiency
  (higher power consumption)

		TeraPool	TensorPool-1	TensorPool-2
Energy Efficiency	**GFLOPs/W**	420.24	4998.04	4605.65
	GFLOPs/W/mm2	7.76	92.29	85.05

Comparison with SoA architectures

	Technology	TFLOPs	Area [mm2]	GFLOPs/W	GFLOPs/W/mm2
Synopsys ARC NPX6 [1]	5nm	250.00	-	30,000.00	-
Jack Unit [2]	65nm	0.28	0.01	170.00	14,453.32
Ampere A100 [3]	7nm	390.00	826.00	1,300.00	1.57
TensorPool	7nm	163.84	919.32	4,605.66	5.01

[1] https://www.synopsys.com/dw/ipdir.php?ds=arc-npx6
[2] https://arxiv.org/pdf/2507.04772
[3] https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf

- Besides being general-purpose (512 Snitch individually-programmable cores), TensorPool is only 6.5x less energy-efficient than one of the most powerful NPUs on the market.
- From preliminary estimation a TensorPool chip can be 3.54x more energy-efficient and 3.18x more energy and area efficient than GP-GPU.

But we still have to build it: roadmap

- **TensorPool:**
  - Optimize RedMulE to L1 connection
  - TeraPool physical design in 7nm
  - TeraPool + RedMulE (TensorPool) physical design in 7nm
  - PPA on model microkernels and operators (combined RedMulE&Cores)
- **System Performance:**
  - Simulation speed is impaired by large design size → from RTL simulation to higher abstraction level, e.g. GVSoC
  - TensorPool GVSoC model developement
  - Data-Movement and end-end performance

TensorPool-1/2 configuration

TensorPool-1

System parameters:

- L1 Memory/Cluster: 4MiB
- NumClusters: 32
- AXI-NoC bandwidth: 0.8KiB/cycle
- Peak-Performance: 4.9 KFLOPs/cycle

Interconnect:

- NumRedMulEs: 16
- NumSubGroups: 16
- NumGroups: 4
- NumTiles (RedMulE/Snitch): 64
- NumBanksPerTile: 32

TensorPool-2

System parameters:

- L1 Memory/Cluster: 4MiB
- NumClusters: 32
- AXI-NoC bandwidth: 0.8KiB/cycle
- Peak-Performance: 4.1 KFLOPs/cycle

Interconnect:

- NumRedMulEs: 16
- NumSubGroups: 0
- NumGroups: 4
- NumTiles (RedMulE): 16
- NumBanksPerTile: 32