CICC

IEEE Custom Integrated Circuits Conference

Toward Open-Source Chiplets for HPC and AI: Occamy and Beyond

Paul Scheffler*, *Thomas Benz**, *Tim Fischer**,
*Lorenzo Leone**, *Sina Arjmandpour**, *Luca Benini**†

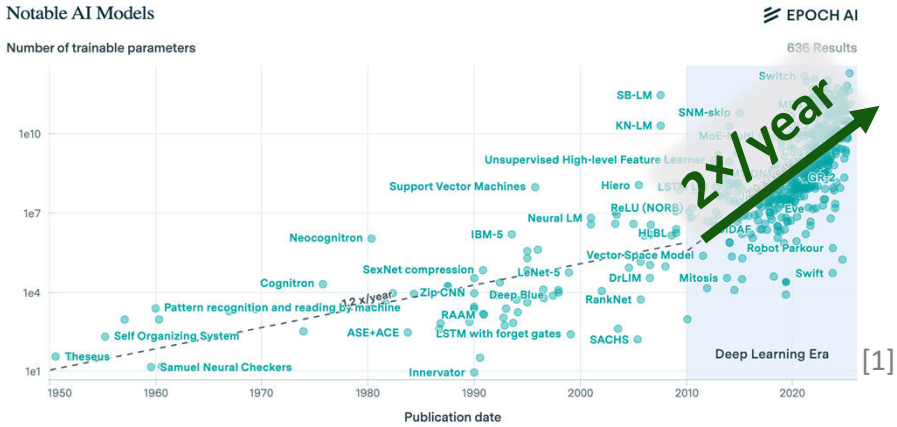
* *Integrated Systems Laboratory, ETH Zurich*
† *Department of Electrical, Electronic, and Information Engineering, University of Bologna*

22 April 2026

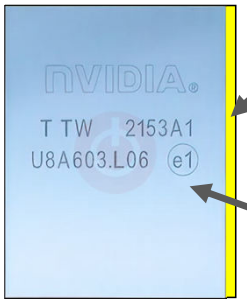
IEEE SOLID-STATE CIRCUITS SOCIETY™ | IEEE

The Monolithic Integration Bottleneck

- HPC and AI compute demand are **outgrowing** performance and BW
  - Gap widens with scaling slowdown
- Designers responded with **larger**, more **specialized** architectures
  - Rise of GPUs & ML accelerators
- ➢ Monolithic integration and single-die packaging become **bottlenecks**
  - Limited die area (reticle)
  - Decreasing yield (or increasing cost)
  - Limited inter-die connectivity



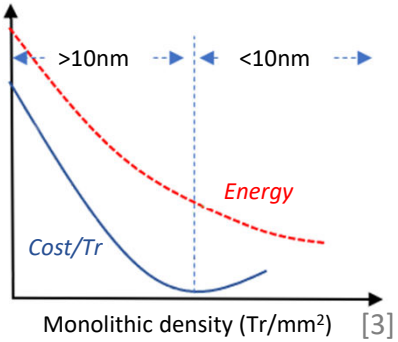
[1]



[2]

858 mm² reticle

814 mm² **(95%)**

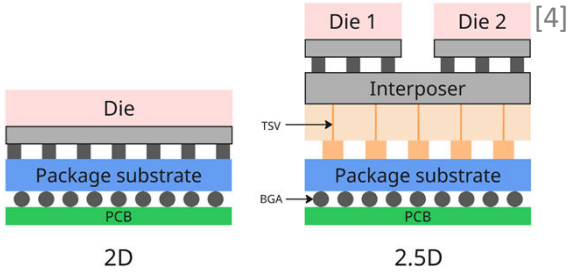


[3]

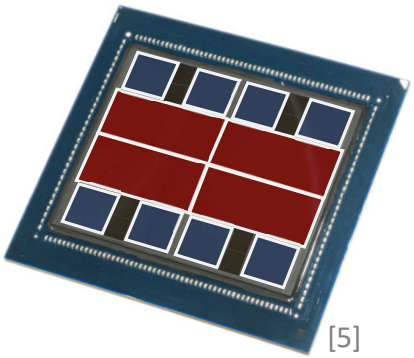
2.5D Integration and Composable Chiplets

- **2.5D integration** provides further scaling
  - Improved *yield*: multiple smaller dies
  - Improved *scale*: free from reticle limit
  - Improved *connectivity:* Si interposers
- **Standardized** chiplets would enable **composable** 2.5D architectures
- Multiple proposed ecosystems:
  - Chiplet System Architecture (*Arm*)
  - Open Chiplet Architecture (*TensTorrent*)
  - Composable Chiplets (*Zero ASIC*)

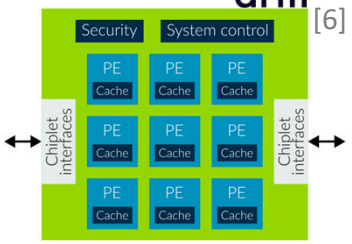
➢ How do we ensure **accessibility**?



[4]

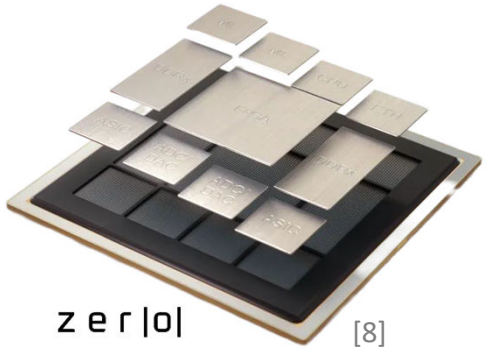

[5]

arm



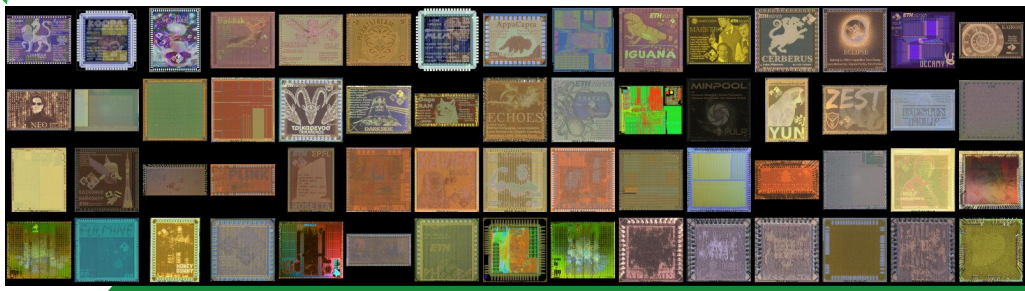
[6]

OCA
OPEN CHIPLET ARCHITECTURE [7]


zero [8]

Composability through Open Hardware

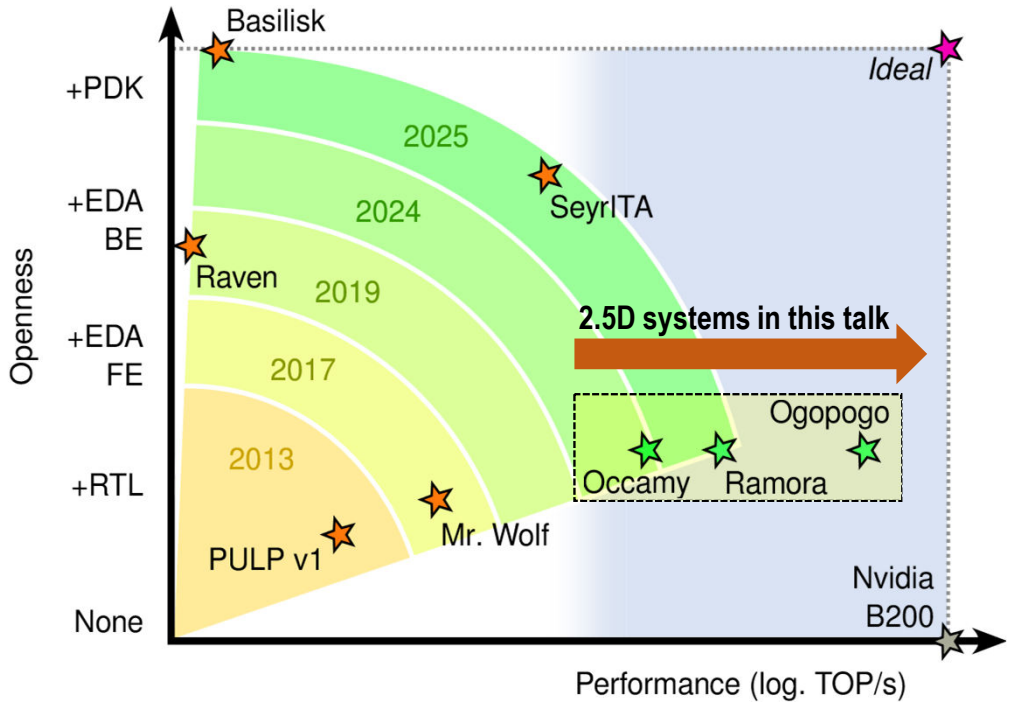
- **Open-source HW** is gaining traction
  - Open RTL designs, EDA, and PDKs
  - High design transparency at low cost
  - Low barriers to collaboration
- Ready for production
  - Large selection of free & open IPs
  - Numerous **silicon-proven** designs
- ➢ A great approach for **standardized** and **composable chiplets**!
  - No royalties, NDAs, or vendor lock-in
  - Free exchange of compatible designs



https://github.com/pulp-platform

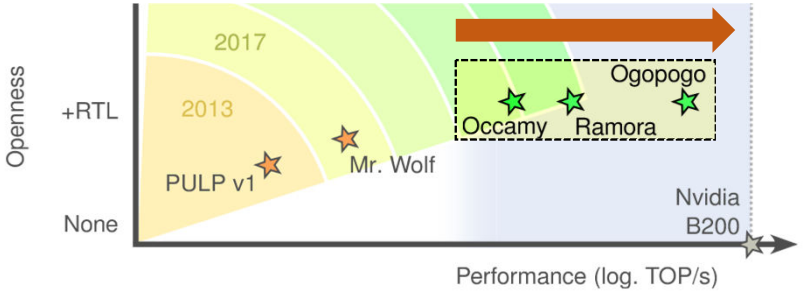
The Openness-Performance Tradeoff

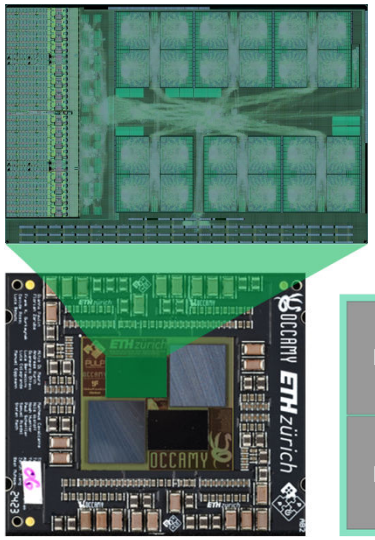
- Open silicon is **performance-limited**
  - Open-source designs, EDA, and PDKs **lag behind** proprietary SoA
  - No open design currently reaches proprietary SoA performance
- With increased **openness**, peak performance **decreases**
  - *SeyrITA* (open RTL + EDA) is notably ahead of *Basilisk* (also open PDK)
- ➢ We need **competitive open chiplets**
  - Must close *performance gap* to SoA

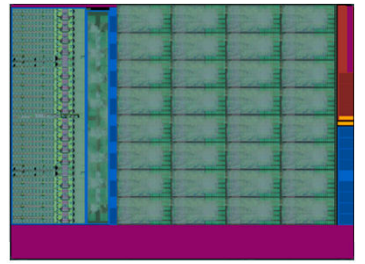


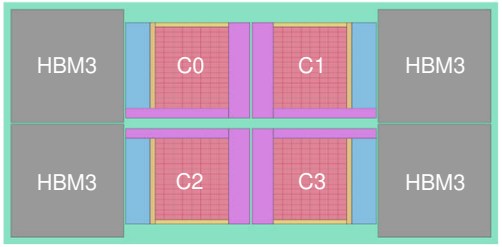
Contributions: A Roadmap for Open HPC/AI Chiplets

1. ***Occamy:*** The first open-RTL RISC-V 2.5D system demonstrated in silicon
   - Two **12nm chiplets** with 16 GiB **HBM2E** each
   - **432** cores, **876 DP-GFLOP/s**, stream ISA extensions
   - **89%** regular, **42–83%** irregular workload FP util.
2. ***Ramora:*** Improving Occamy with a NoC
   - **576** cores, **1.29 DP-TFLOP/s** on **same area**
   - **16×** faster **1.04 Tb/s** D2D interface
3. ***Ogopogo:*** A quad-chiplet 7nm concept
   - Four **7nm chiplets** with **HBM3** each
   - **4608** cores, **10.3 DP-TFLOP/s**, NoC xfer extensions
   - **19%** higher norm. compute density than B200
4. Next steps toward **end-to-end** open chiplets







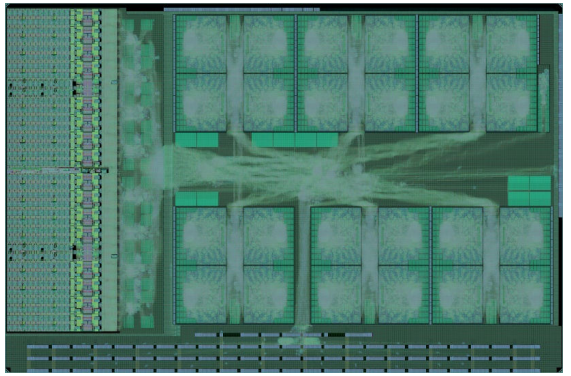


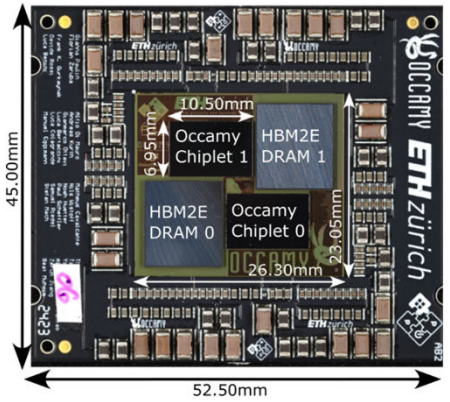
Outline

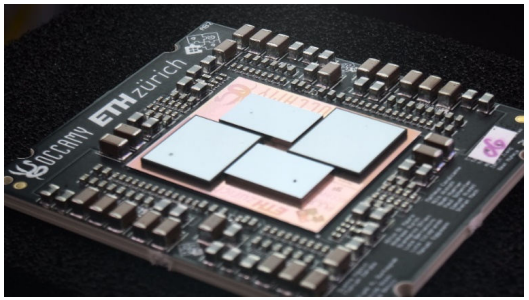
1. ***Introduction***
2. ***Occamy:*** A silicon-proven open 2.5D RISC-V system
3. ***Ramora:*** Improving Occamy with a NoC
4. ***Ogopogo:*** A quad-chiplet 7nm concept
5. ***Toward end-to-end open chiplets***
6. ***Conclusion***

Occamy: A Dual-Chiplet Silicon Demonstrator

- A 2.5D RISC-V-based manycore for efficient dense *and* sparse HPC and AI
  - Two **12nm chiplets** on **65nm interposer**
  - **432 RV32G cores** with multiple **ISA extensions** to maximize efficiency
  - **876 DP-GFLOP/s** peak, **32 GiB HBM2E**
- Achieves competitive *regular* (**89%**), leading *irregular* (**42–83%**) peak FP util.
  - Flexible **streaming units** for *sparse, dense,* and *mixed-regularity* HPC/AI workloads
- **Hierarchical** design approach







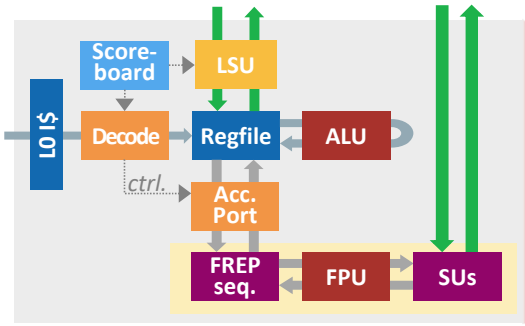
GlobalFoundries™

Micron Rambus

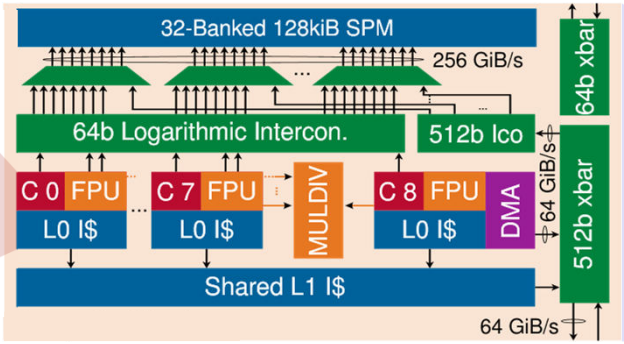
SYNOPSYS® AVERY design systems

Achieving Scale through Hierarchical Design

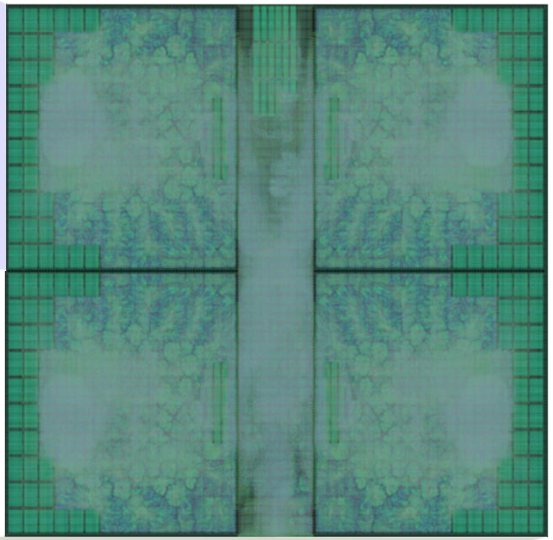
Worker Core



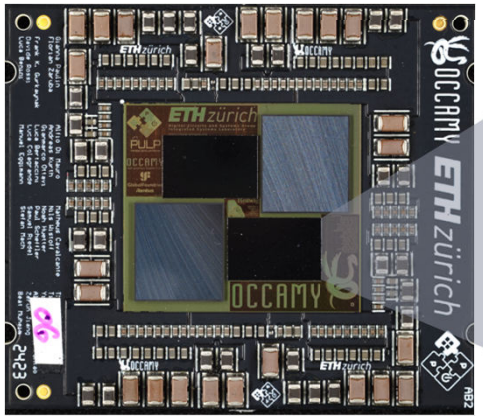
Compute Cluster



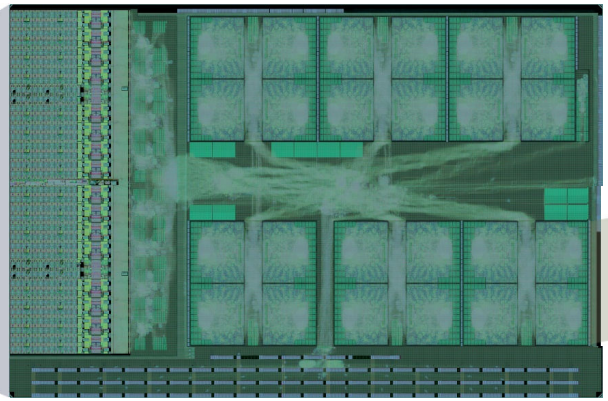
Occamy Group



Occamy System

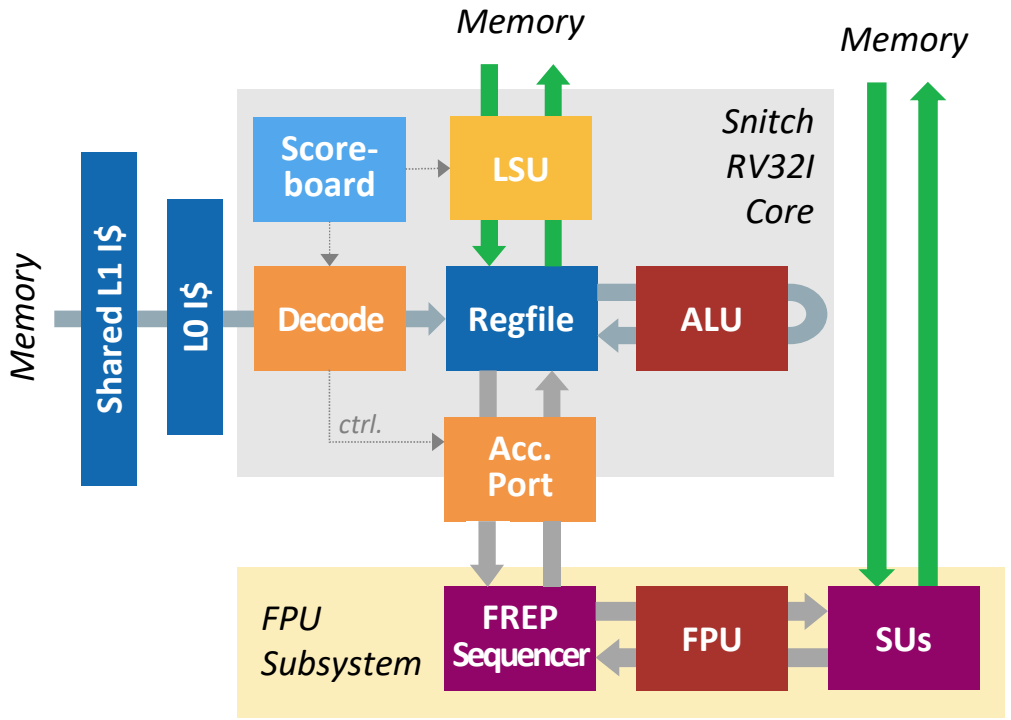


Occamy Chiplet



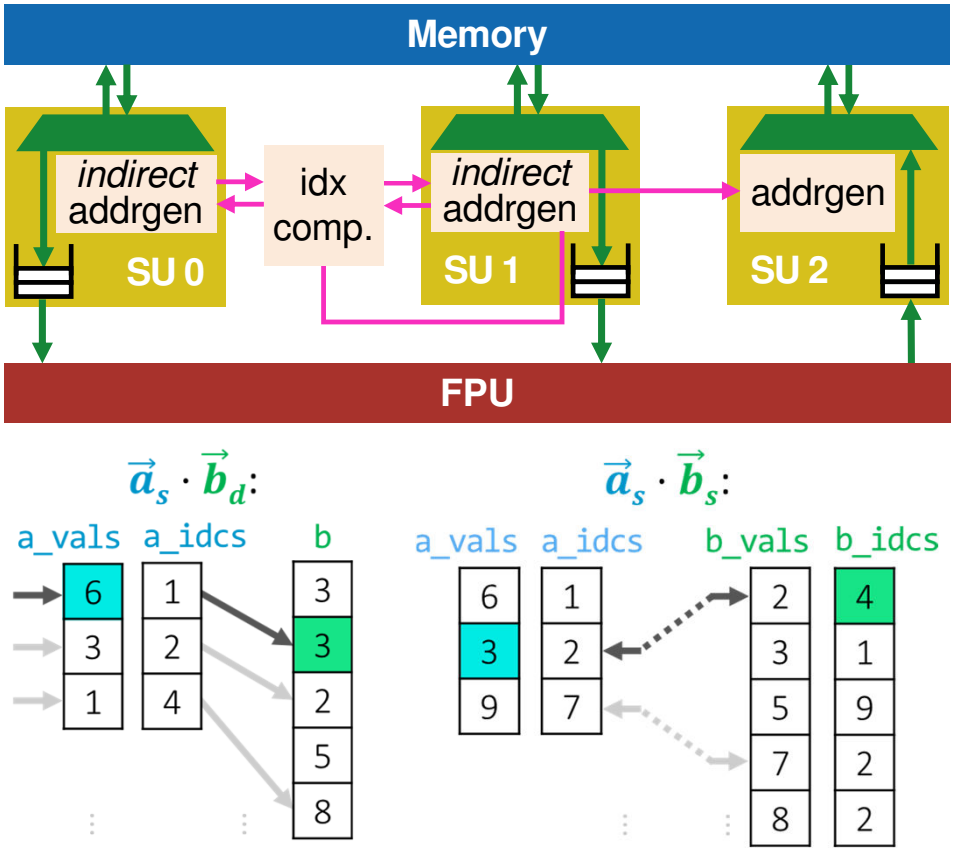
Worker Core: Tiny RV32I + Large FPU + ISA Extensions

- *Snitch*: **tiny**, **extensible** RV32I core
  - *Extensible* through accelerator port
  - *Latency-tolerant* through scoreboard
    → can issue ~10 nonblocking mem ops
- Paired with **FPU subsystem**
  - Pipelined, double-precision FPU
  - FP8–FP64 SIMD, FP8–FP16 widening DOTP
- **ISA extensions** for *near-ideal FPU util.*
  - ***FREP:*** dedicated HW loop for FPU
  - ***Streaming units (SUs):*** map *dense* or *sparse memory streams* to FP register accesses



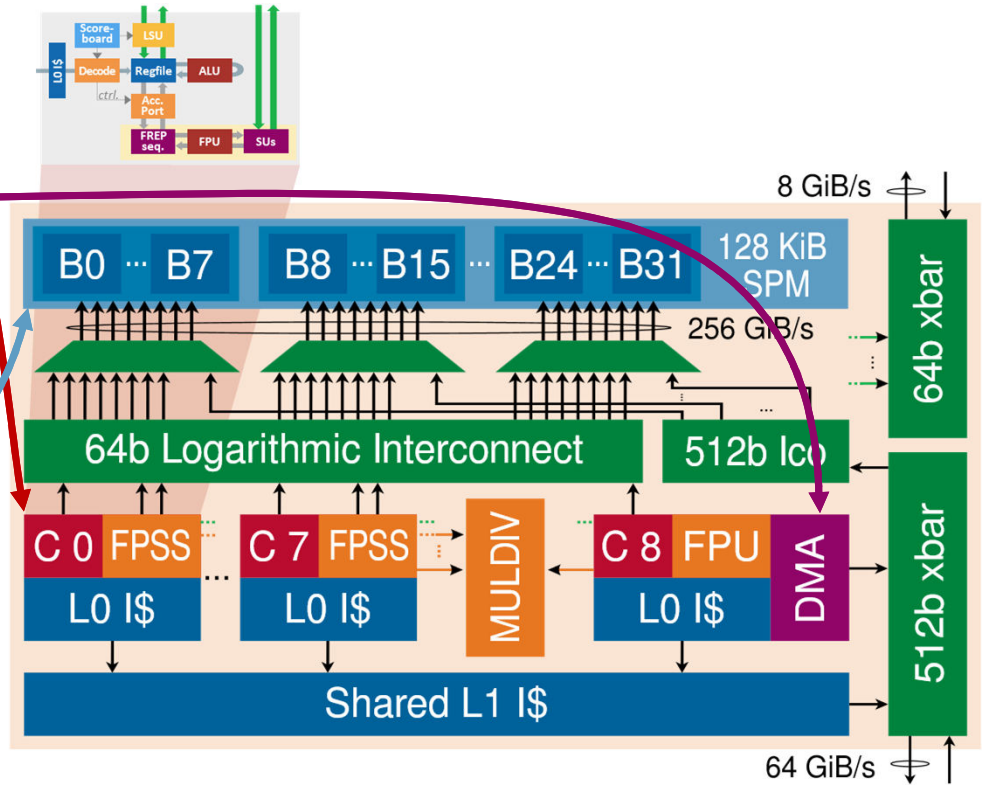
Streaming Units: The Key to High PE Efficiency

- Sparse SUs are **highly flexible**
  - Three **affine** SUs for dense workloads
  - Two **indirect** SUs for *scatter-gather:* sparse-dense LA, stencils, codebooks, …
  - Index comparator for **intersection** and **union***:* sparse-sparse LA
- With FREP: **continuous MAC issues** on both *dense* and *general sparse LA*
  - **High FPU utilization** across the board
  - Generalizes to other regular, irregular, and *mixed-regularity* workloads
- **Performance, efficiency, *and* flexibility**



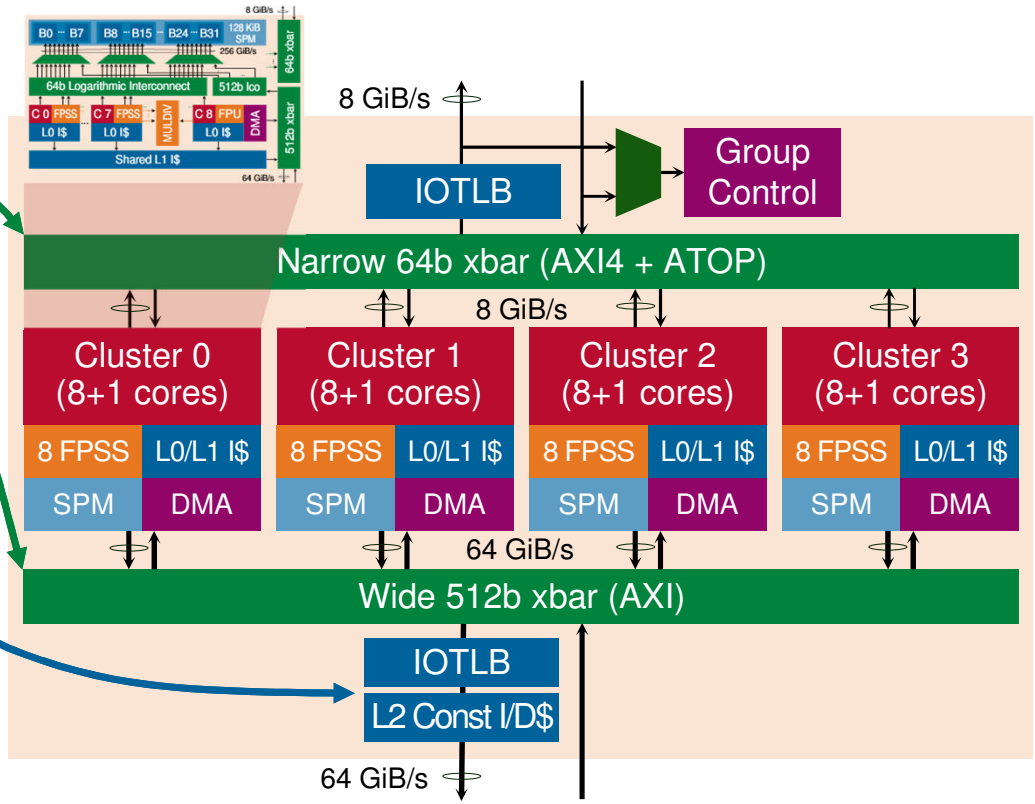
Compute Cluster: The Fundamental Processing Block

- **Eight** worker cores
  - Each: 64b SIMD FPU and 3 SUs and FREP
- *Ninth* core: **DMA control core**
  - 512b **DMA engine** to interconnect
  - HW support for async **≤2D transfers**
  - **Latency-tolerant** transfers (100s of cycles)
- 128 KiB, 32-bank **scratchpad (SPM)**
  - Low latency, high BW (24B/cycle/core)
  - Stores data chunks to be processed, **double-buffered** with DMA engine
- **Shared 8 KiB I-cache** and peripherals



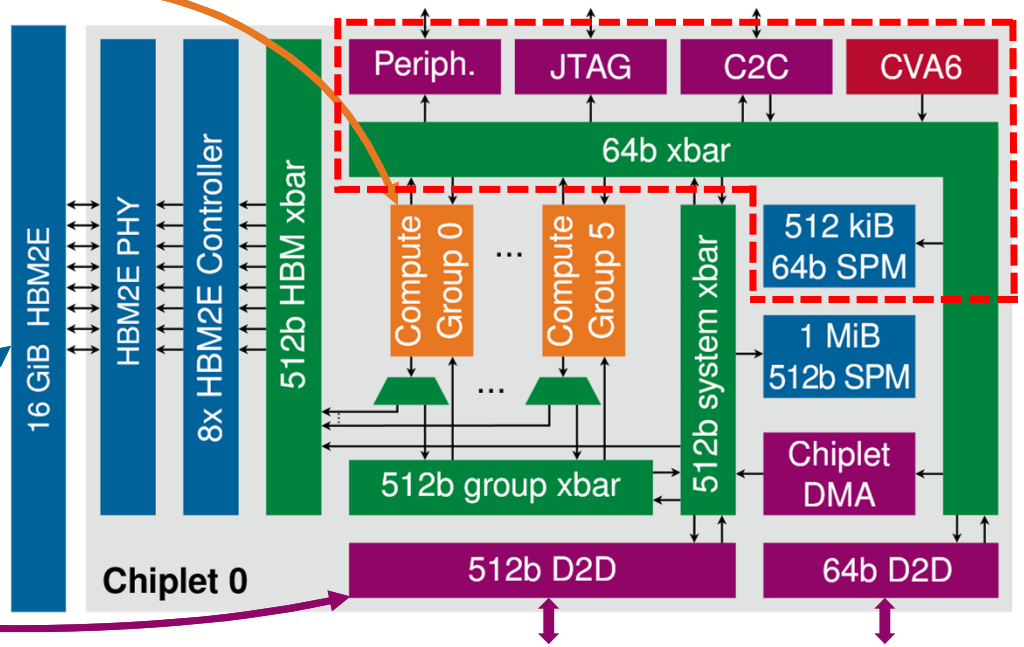
Four Clusters form a *Group*

- **Fully interconnected** on *two* crossbars*:*
  - ***64b***: messaging and control
  - ***512b***: bulk data and instructions
  - ➢ High-bandwidth local data exchange
- **Shared ports** (and BW) to top level
  - IOTLBs for remapping, access control
  - Configurable 32 KiB constant cache
- **Group controller** provides:
  - SW-controlled clock gating and reset
  - Interconnect isolation when gated



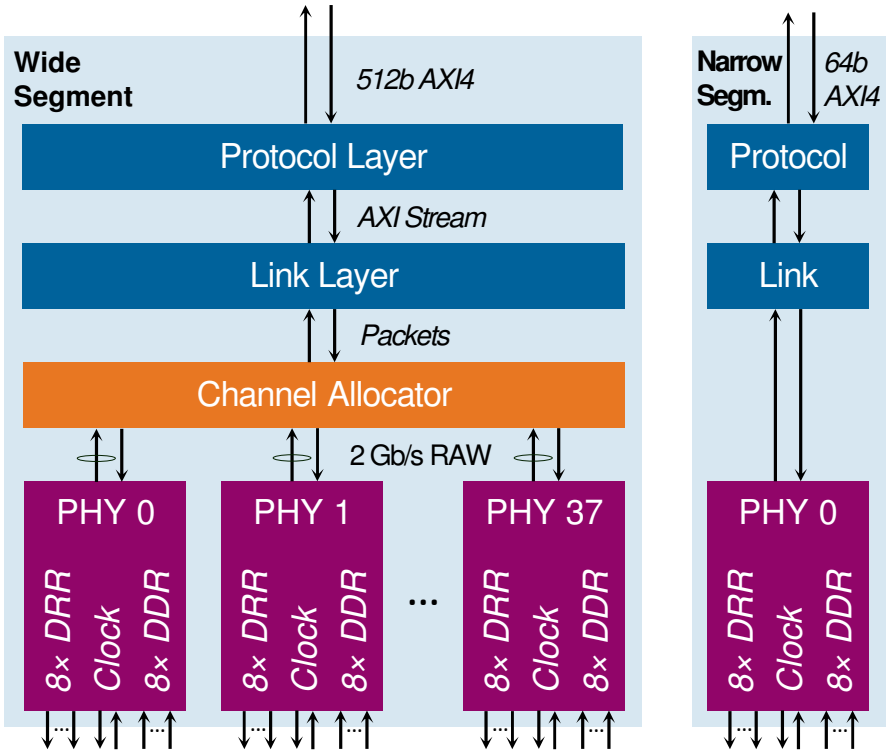
Occamy Chiplet: Six Groups + HBM2E + D2D Link

- **Groups** connect to **each other** and **HBM**
  - 512b data network of *three* xbars
  - Single 64b xbar for messaging
- Autonomous RV64 **host domain**
  - Linux-capable CVA6 manager core
  - Peripherals (SPI, I2C, UART, …) and SPM
- **16 GiB** of **410 GB/s HBM2E**
  - *Continuous* and *interleaved* access modes
- **64 Gb/s D2D interface**
  - Fully digital and fault-tolerant



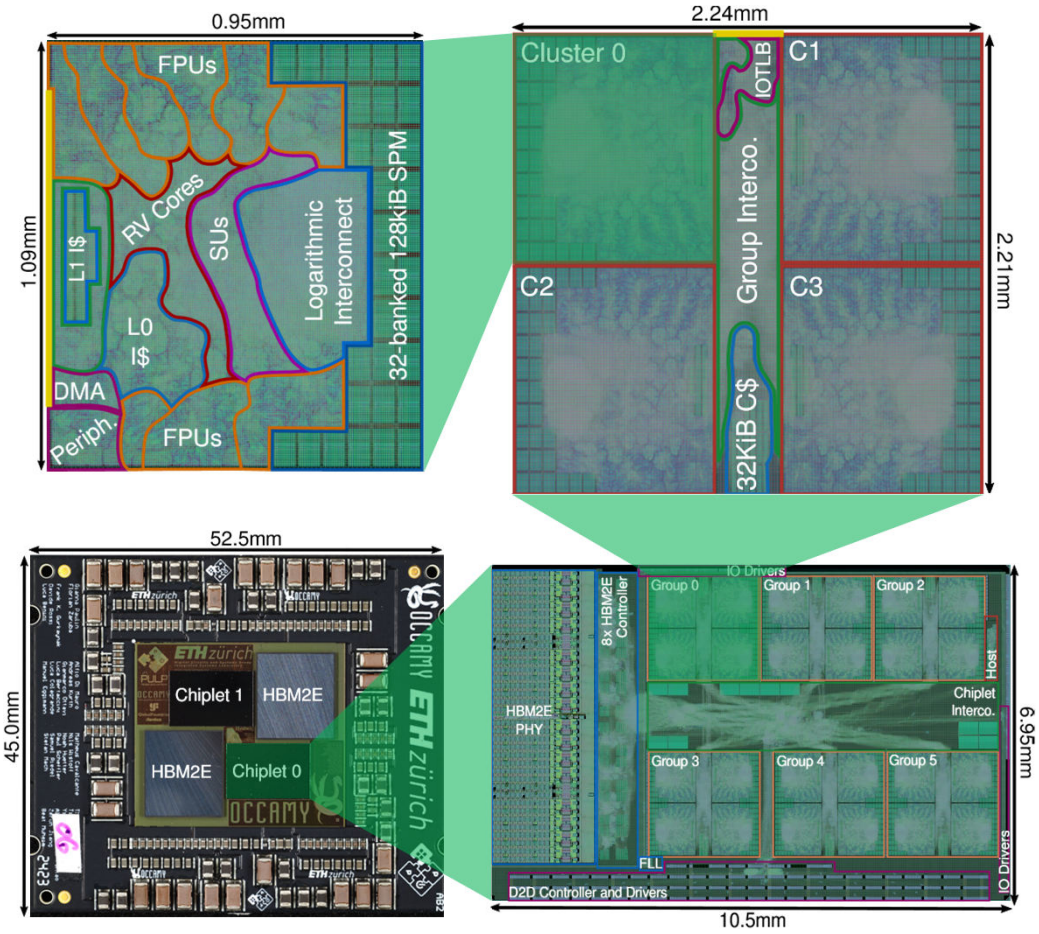
Die-to-Die Interface: Transparent D2D NoC Bridge

- **Transparently** bridges chiplet interconnects
  - **Direct access** to other chiplet's memory space
  - Transmits and injects any AXI4 traffic as-is
  - Manages latency and PHY-dependent slicing
- Channel allocator ensures **fault tolerance**
  - Detects and reconfigures to avoid faulty links
- **Narrow** (64b) and **wide** (512b) segments
  - Up to **1.33 Gb/s** and **64 Gb/s** duplex
  - Wide segment features **38 PHYs**
- **Ready for high-speed mixed-signal PHYs**



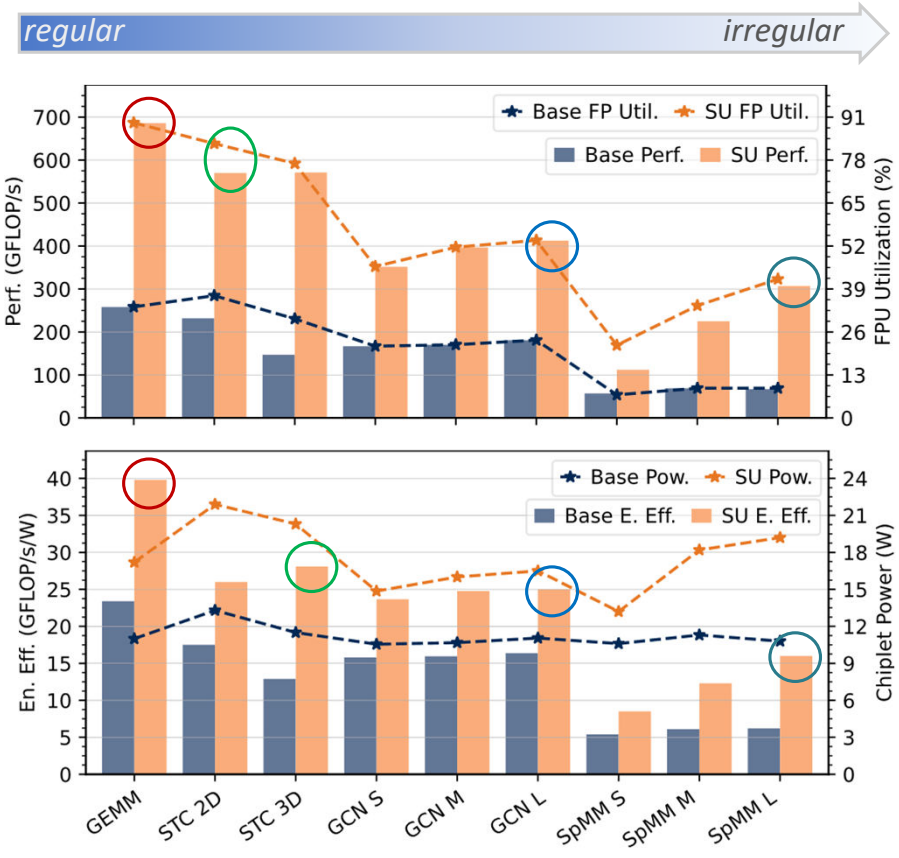
Hierarchical Implementation

- **73 mm² chiplets** fabricated in **GF 12LP+**
  - 1.14 GHz at typical conditions (0.8 V, 25 °C)
- **Hierarchical P&R** in Synopsys *Fusion*
  - *Cluster:* **44%** worker core, **17%** SPM
  - *Groups:* **83%** clusters
  - *Chiplet*: **39%** groups, **25%** HBM IF, **11%** D2D
- Mounted on 65 nm ***Hedwig*** **interposer**
  - BEOL only, connects chiplets & HBM stacks
- Placed on **carrier board**
  - **System module** fits std. LGA 2011 socket



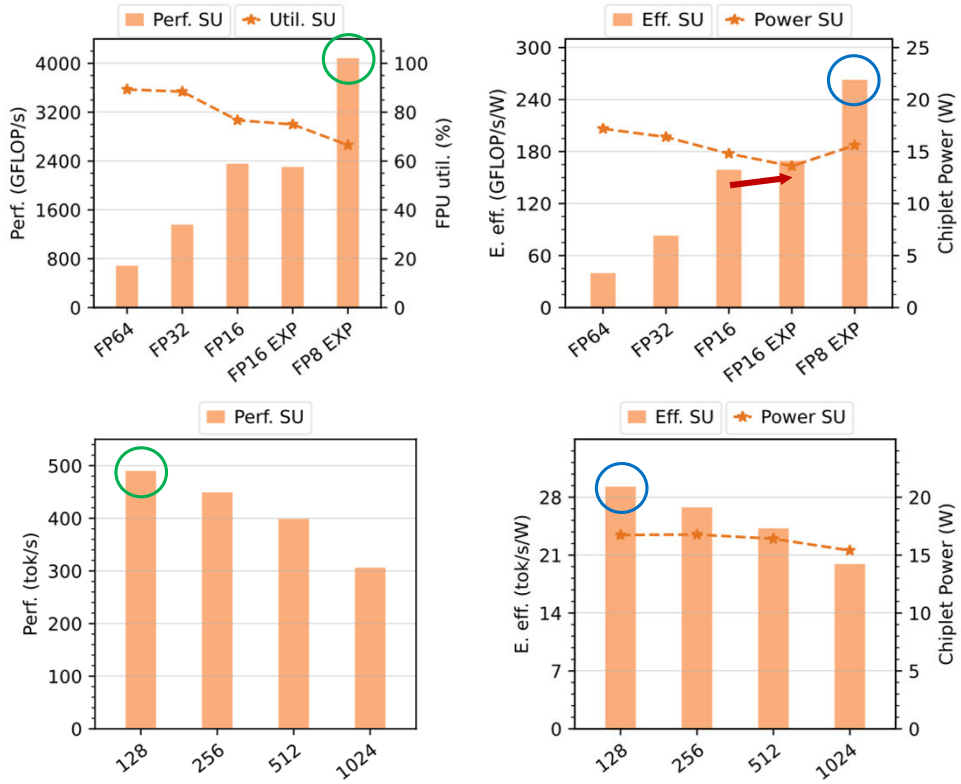
Performance: Dense and Sparse FP64 Workloads

- FP64 workloads @ 1 GHz, 0.8 V, 25 °C
  - RV32G baseline vs. code using SUs
  - *GCN*, *SpMM*: **real-world sparse matrices**
- *GEMM*: **2.7×** faster, **1.5×** less energy
  - **89%** FP util., **686 GFLOP/s**, **40 GFLOP/s/W**
- *Stencils*: up to **3.9×** faster, **2.2×** less energy
  - **83%** FP util., **571 GFLOP/s**, **28 GFLOP/s/W**
- *GCNs*: up to **2.2×** faster, **1.5×** less energy
  - **54%** FP util., **413 GFLOP/s**, **25 GFLOP/s/W**
- *SpMM*: up to **4.6×** faster, **2.6×** less energy
  - **42%** FP util., **307 GFLOP/s**, **16 GFLOP/s/W**



Performance: Mixed-Precision GEMM, LLMs, and D2D

- *GEMM:* near-linear precision scaling
  - *FP8-EXP:* **4.1 TFLOP/s** and **263 TFLOP/s/W**
  - FP16: expanding variant **6.5%** more efficient due to dedicated DOTP units
  - Slight FP. util losses to conversions, packing
- *GPT-J inference* in FP16 (prefill)
  - Up to **490 tok/s** and **29.3 tok/s/W**
  - Slow drop in efficiency as compute time shifts from GEMM to softmax
- *D2D link:* **1.6 pJ/b** at up to **96%** util.
  - **27** cycles latency for narrow core read
  - **61** cycles latency for wide DMA xfer



Occamy vs. SoA 2.5D CPUs and GPUs

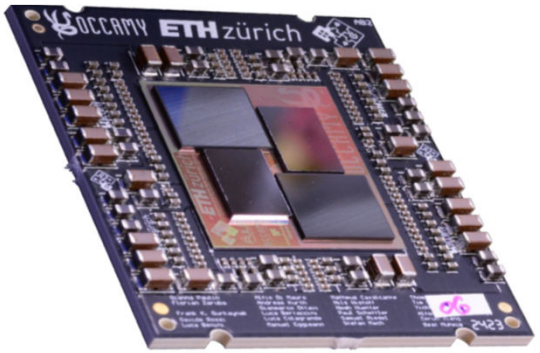
** Area normalized to GF12 LP+ node*
[a] At base (non-boost) clock
[b] Fully utilized tensor cores
[c] At below-peak clock (FP util. adjusted)
[d] Based on many-GPU supercomputers
[e] At 1 GHz clock
[f] Tensor cores unused (FP util. adjusted)

		Fujitsu A64FX [9]	AMD Rome [a] [10]	Nvidia A100 [11]	Nvidia H100 [12]	Occamy	
Peak Compute	FP Formats	FP64 … FP16	FP64 … FP32	FP64 … FP16	FP64 … FP8	FP64 … FP8	
	[TFLOP/s]	3.38 … 13.5	2.30 … 4.61	19.5 [b] … 312 [b]	**66.9** [b] … **1979** [b]	0.88 … 7.00	
	[GFLOP/s/mm²]*	13.9 … 55.7	5.21 … 10.4	30.8 [b] … 493 [b]	**57.2** [b] … **1691** [b]	17.0 … 136	*Above CPUs, but below GPUs*
Dense LA FP64	[TFLOP/s]	1.98 [c] [13]	1.60 [14]	18.5 [15]	**33.3** [d] [16,17]	0.686 [e]	
	FPU util.	72% [c] [13]	70% [14]	**95%** [15]	81% [d] [16]	89% [e]	*Competes w. GPUs*
	[GFLOP/s/W]	16.9 [c] [9]	-	41.4 [d] [18]	**65.4** [d] [18]	39.8 [e]	*approx. A100*
	[GFLOP/s/mm²]*	8.15 [c] [13]	3.62 [14]	**29.3** [15]	28.5 [d] [16]	13.3 [e]	
Stencils LA FP64	FPU util.	11% [c] [19]	37% [20]	49% [f] [21]	-	**83%** [e]	***1.7× SoA***
	[GFLOP/s/mm²]*	1.33 [c] [19]	1.93 [20]	9.58 [22]	-	**11.1** [e]	***1.2× SoA***
Sp-dense LA FP64	FPU util.	4.7% [c] [23]	8.1% [24]	2.9% [f] [25]	-	**42%** [e]	***5.2× SoA***
	[GFLOP/s/mm²]*	0.54 [c] [23]	0.42 [24]	0.45 [f] [25]	-	**5.95** [e]	***11× SoA***

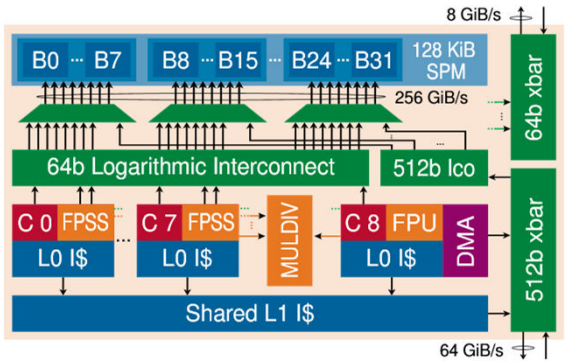
- **Competitive regular** peak FP util. (**89%**) and energy efficiency
- **Leading irregular** peak FP util. (**42–83%**) and area efficiency

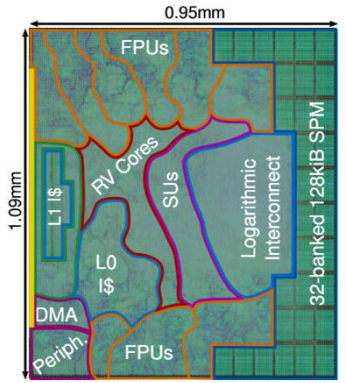
Occamy: An Open 2.5D System for HPC and AI

- The first open-RTL RISC-V 2.5D system demonstrated in silicon
  - Two **12nm chiplets** with **32 GiB HBM2E** each
  - **432 RV32G cores** with **876 DP-GFLOP/s** peak
- A highly efficient cluster architecture
  - **SPM** & **DMA engine** absorb HBM latencies
  - Flexible **sparse SUs** keep FPUs busy
- Efficient sparse *and* dense compute
  - Competitive regular (**89%**), leading irregular (**42–83%**) workload FP util.
  - Leading irregular area efficiency (**5.2×**, **11×**)



GlobalFoundries™
Micron Rambus
SYNOPSYS® AVERY design systems

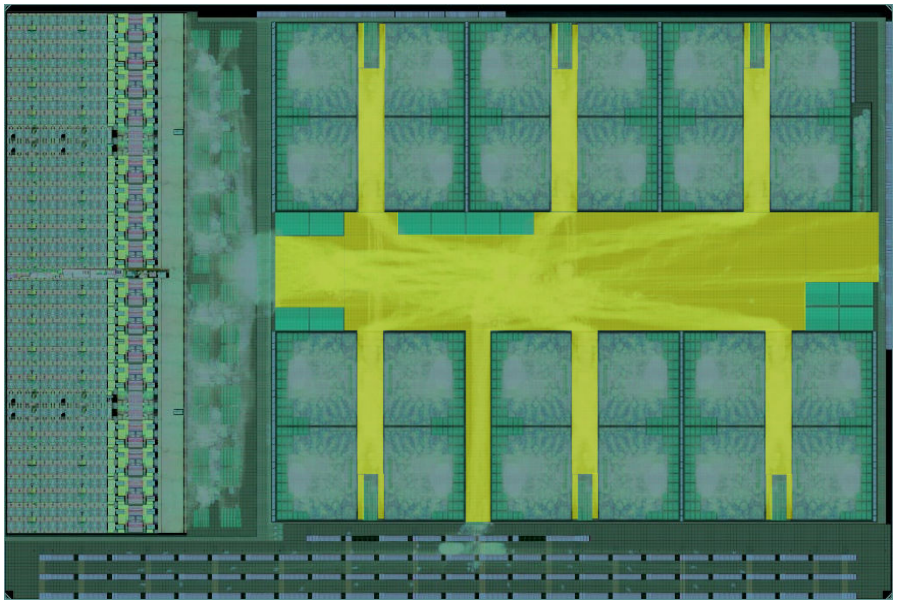




Open-source RTL: `github.com/pulp-platform/occamy`

There is One Problem to Address in Occamy…

- Occamy's xbar-based interconnect is **large** and **not scalable**
  - **31%** of compute domain area
  - Scales quadratically: $O(N_{mgrs} \times N_{endp})$
  - Centralized xbars create long routes and congestion → **limit QoR**
- Interconnect limits **off-chip BW**
  - 512b chiplet network required partitioning
  - ➢ Hard to accommodate **higher-BW D2D link**



How do we solve **interconnect scalability**?

There is One Problem to Address in Occamy…

- Occamy's xbar-based interconnect is **large** and **not scalable**
  - **31%** of compute domain area
  - Scales quadratically: $O(N_{mgrs} \times N_{...})$
  - Centralized xbars create long wires and congestion → limits frequency
- Interconnect limits off-chip BW
  - 512b chip-wide network required partitioning
  - Hard to accommodate higher-BW D2D link

We need a **Network-on-Chip:** scales **linearly** & enables **cluster tiling**

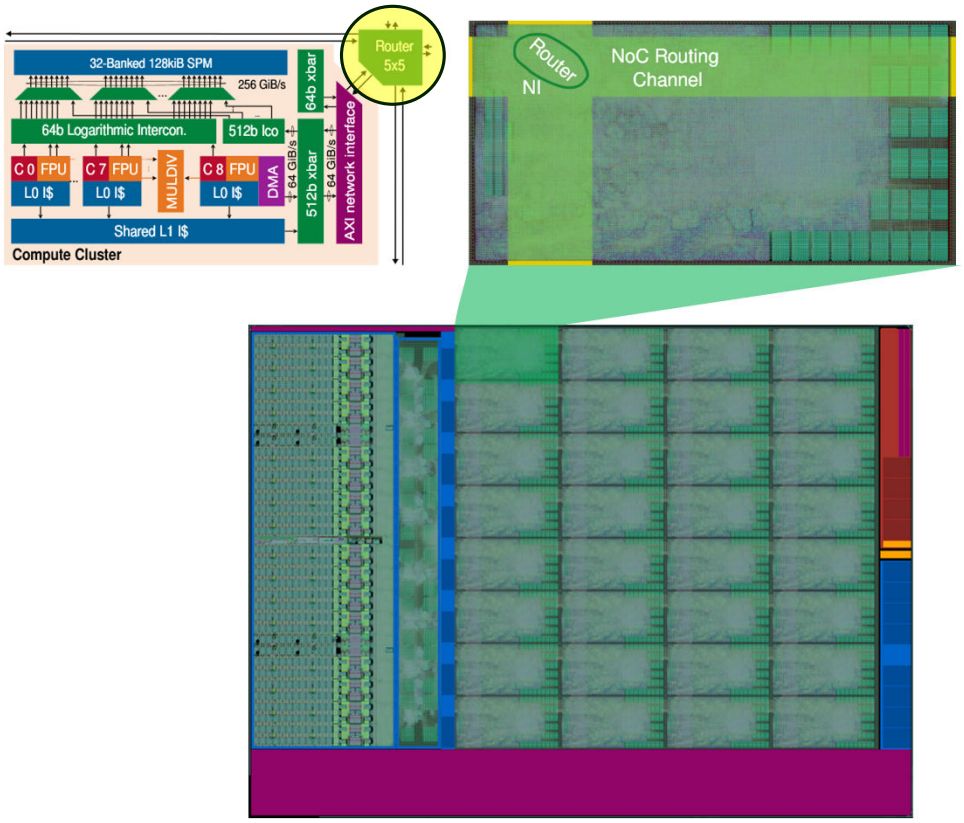
How do we solve **interconnect scalability**?

Outline

1. ***Introduction***
2. ***Occamy:*** A silicon-proven open 2.5D RISC-V system
3. ***Ramora:*** Improving Occamy with a NoC
4. ***Ogopogo:*** A quad-chiplet 7nm concept
5. *Toward end-to-end open chiplets*
6. *Conclusion*

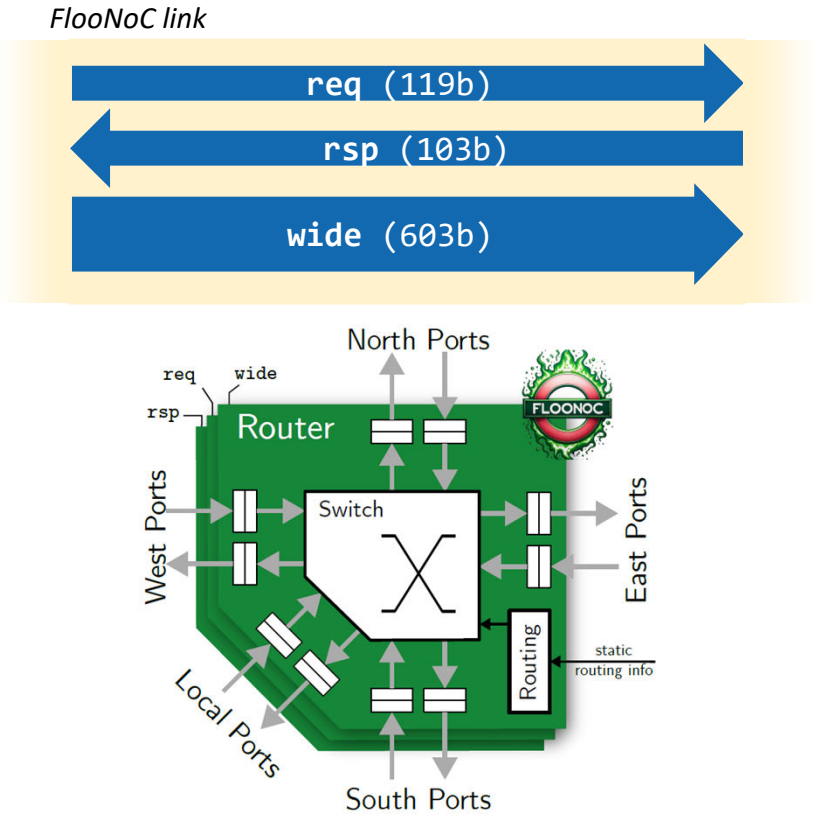
Ramora: Improving Occamy with a NoC

- Keep **cluster** architecture, but make it **tilable** in a 2D mesh NoC (*FlooNoC)*
  - **33%** more clusters on **same chiplet area**
  - **11%** faster clock → **1.29 DP-TFLOP/s**
  - Solves the scalability bottleneck
- Improves **HBM bandwidth utilization**
  - **11%** in low-, **22%** in high-traffic scenarios
- Enables a **16× faster D2D link** design
  - Same frontend, new LVDS PHYs
  - **1.04 Tb/s** duplex on wide segment



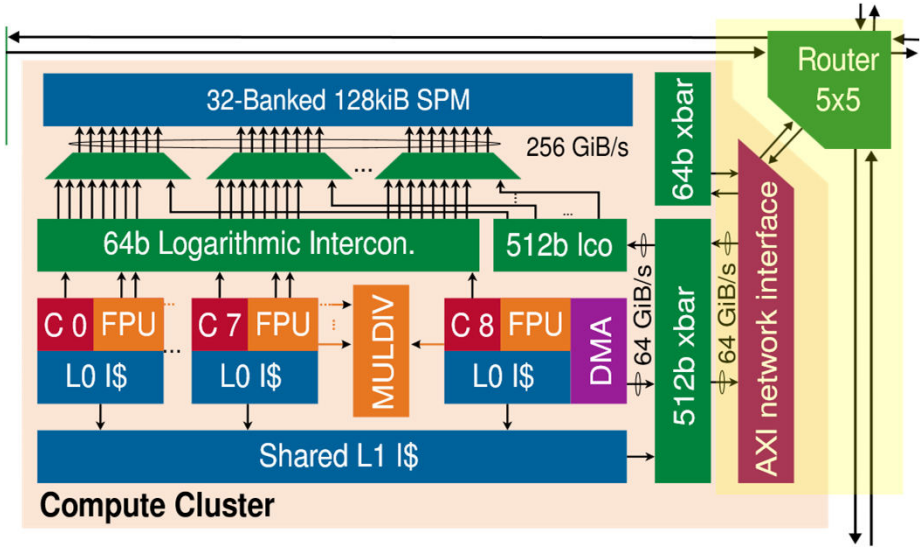
A Primer on FlooNoC

- Narrow (64b) *and* wide (512b) AXI4
  - Network interfaces (NIs) *with* and *without* RoB
- Three **physical channels** per link
  - *req:* requests (**AW**/**AR**) and *64b* write (**W**)
  - *rsp:* write responses (**B**) and *64b* read (**R**)
  - *wide: 512b* read (**R**) and write (**W**) data
  - **One link each way** between neighbors
- Routers keep channels **decoupled**
  - Ports are **fully connected** internally
  - **Static routing** rules: source-based, table-based, or dimension-ordered



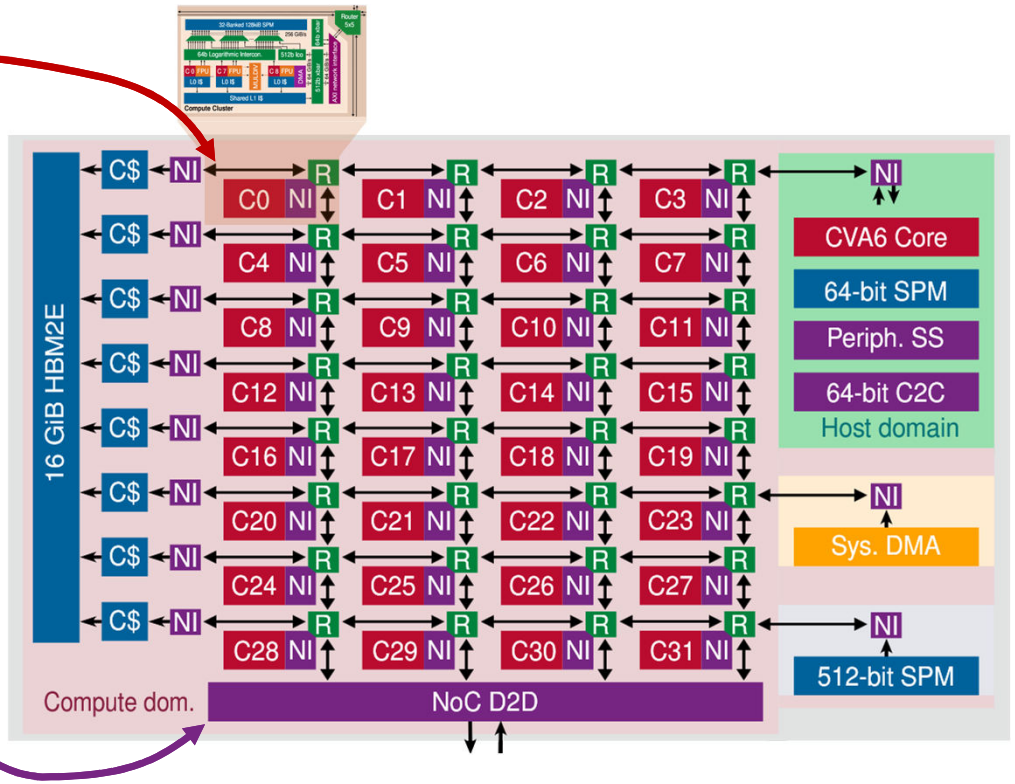
Turning the Cluster into a NoC Tile

- Add **NI** and **5×5 router** to cluster
  - Maintain proven cluster internals
- NI is **RoB-less** (lightweight)
  - Instead, extend DMA to support **out-of-order transfers**
  - Multiple backends simultaneously track different transaction parts
  - ➢ Maintains AXI intra-ID ordering without sacrificing performance
- Expose router ports at edges
  - Enables tiling through **abutment**



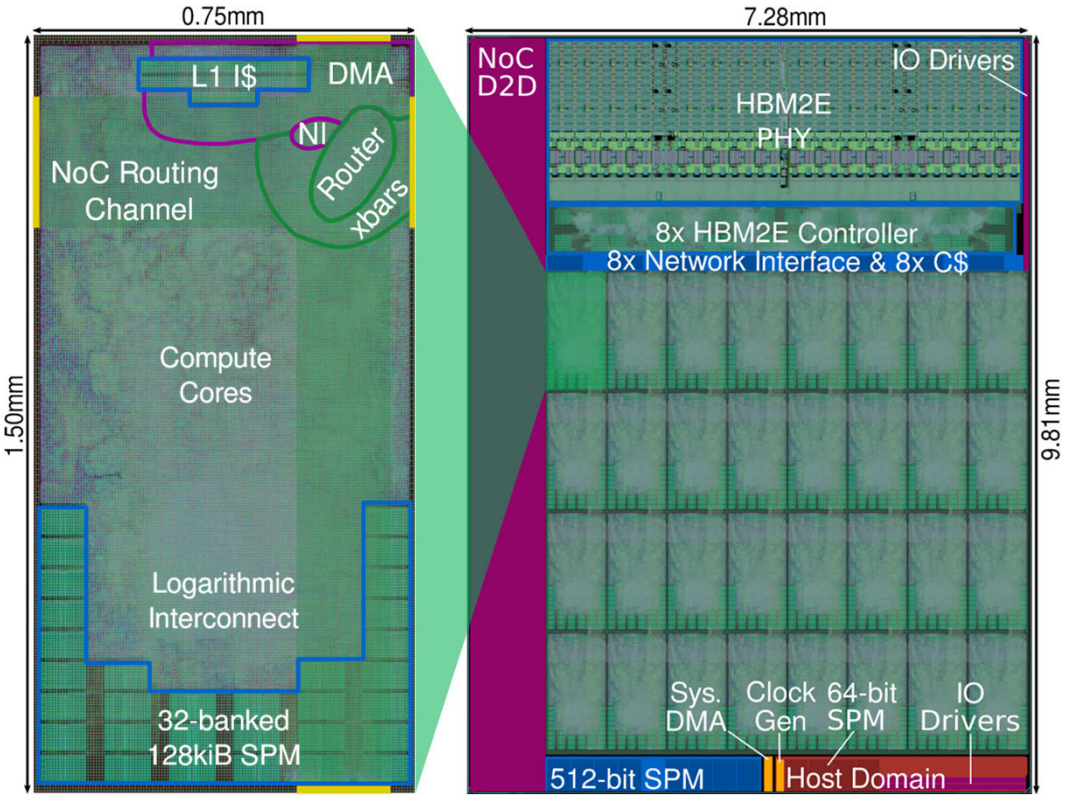
Ramora Chiplet: Less Hierarchy, More Compute

- **8×4 mesh** of clusters (**24 → 32**)
  - *Groups* become *rows*, each sharing BW of one HBM channel in-line
  - Constant caches attach directly to HBM
  - ➢ **Same or higher BW** with **less hierarchy**
- *Host*, *DMA*, *SPM* at right edge
- **16× faster D2D** with new LVDS PHY
  - **1.04 Tb/s** for *wide*, **137 Gb/s** for *narrow* xfers on **same macro area**
  - Direct flit transmission to other die → **virtual 16×4 mesh** across chiplets



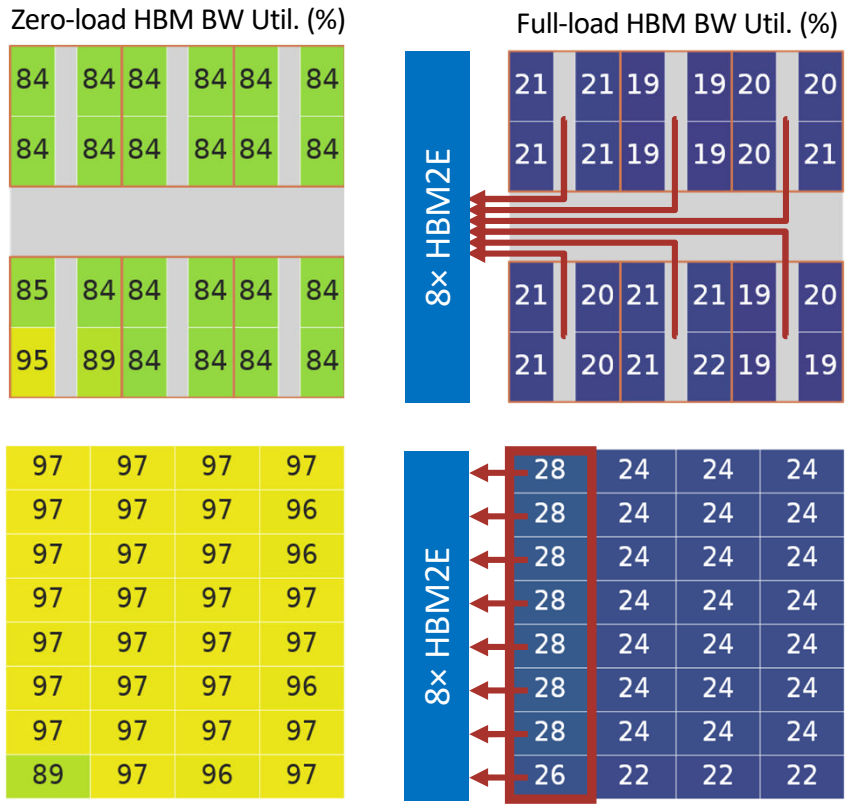
Ramora Implementation

- **71 mm^2** chiplet area in **GF 12LP+**
  - **11%** faster: 1.26 GHz at typical conditions
  - **2.1% smaller** than Occamy chiplet
  - **43%** higher compute density
- Still hierarchically implemented
  - *Cluster:* **8.6%** larger due to router, NI
  - *Comp. domain:* tiled clusters with small gaps for global routing (e.g. clock)
  - *Chiplet:* **55%** clusters, **25%** HBM, **13%** D2D
- NoC channels routed over existing HW
  - Reclaims **free top-metal resources** over "shallow" cluster logic (e.g. SPM, FPUs)



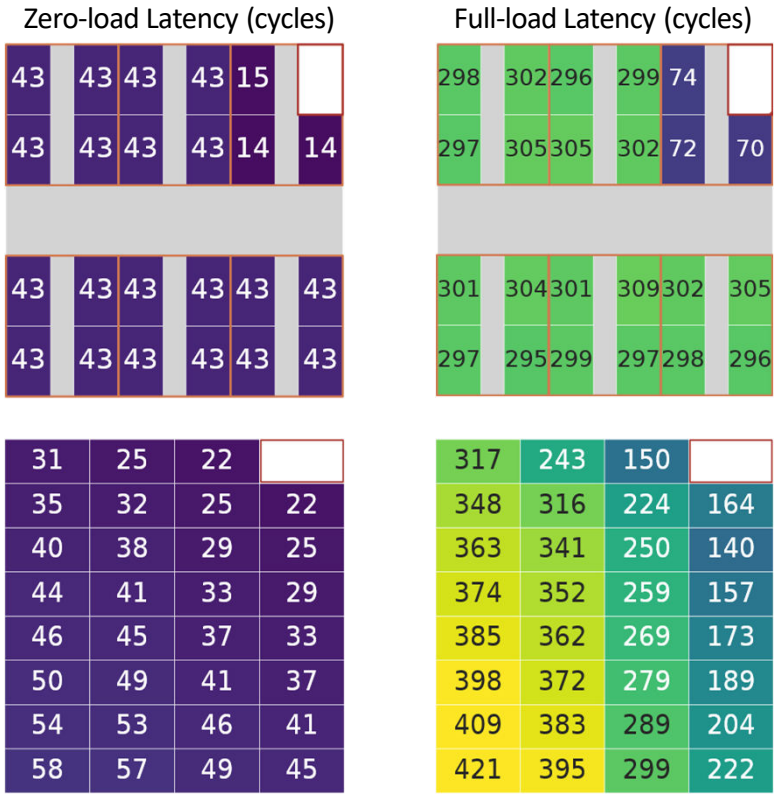
Evaluation: HBM Bandwidth Utilization

- We compare Ramora to Occamy
  - Repeated 4 KiB cluster DMA reads from HBM
  - *Zero* vs. *full* load: *one* or *all* clusters active
- *Zero load:* avg. **85%** to **97%** (**+10%**)
  - Fewer hops (routing steps) to HBM
- *Full load:* avg. **20%** to **25%** (**+22%**)
  - More requestor-side BW (6 vs. 8 links)
  - HBM no longer accessed through single xbar, but multiple routes → less contention
- NoC exhibits minor **NUMA effects**
  - Column closest to HBM gets more BW



Evaluation: NoC Latency and Energy Efficiency

- Narrow core reads from *one* cluster
  - *Zero load*: avg. **0.4%** *lower* latency
  - *High load:* avg. **8.4%** higher latency
- ➢ Attained scalability *and* higher BW *without* compromising latency
  - 2D mesh avoids extensive link pipelining needed in Occamy → similar latency
- Power simulation on 4 KiB DMA xfer
  - Cluster is otherwise *idle*
  - ➢ Only **15%** of cluster power consumed by components involved in xfer
  - ➢ Only **0.15 pJ/B/hop** spent in NoC



[a] 0.14 mm² dummy tile
[b] 0.034 mm² dummy tile
[c] simplex bandwidth
[d] only single tile

Ramora vs. SoA Mesh-NoC Systems

	Piton [26]	Celerity [27]	Ou et al. [28]	ESP [29]	Ramora
Technology Node	32 nm SOI	16 nm FinFET	14 nm	12 nm FinFET	12 nm FinFET
Frequency [GHz]	0.5	**1.4**	1.0	0.8	1.26
Mesh Dimensions	5 × 5	**8 × 62**	16 × 16	8 × 8	8 × 4
Phys. Chan. Data Widths	64b	32b	256b	64b	**512b + 2×64b**
NoC-to-Die Area Ratio	**2.9%**	7.8%	18% [a] or 35% [b]	–	3.5%
Tile-to-Tile BW [Gb/s] [c]	96	45	256	310	**806**
Aggregate BW [Tb/s]	4	**361**	– [d]	74	103
NoC Eff. [pJ/bit/hop]	0.45	–	–	2.0	**0.15**
NoC Power contrib.	–	–	–	23%	**5.7%**

Competitive speed & size

Wide links at low area cost

2.6× SoA

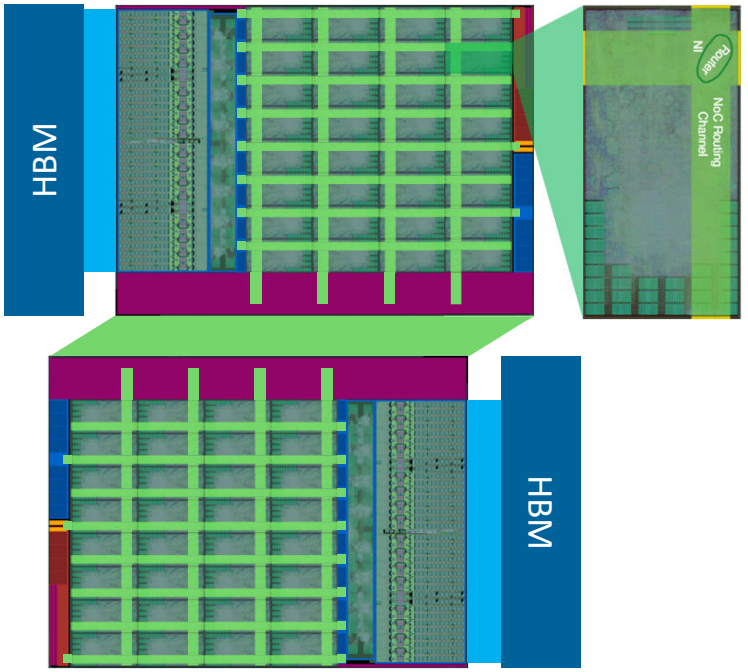
3.0× SoA

4.0× SoA

- **Wide**, **multi-channel** physical links at **low area overhead**
- **Leading** tile-to-tile bandwidth and NoC energy efficiency

Ramora: Making Occamy Scalable with a NoC

- **Slashed interconnect cost** with a mesh NoC
  - **33%** more clusters on **same chiplet area**
  - **11%** faster clock, **10–22%** higher HBM BW util.
  - Increased peak perf.: 876 to **1290 DP-GFLOP/s**
- Maintained Occamy's highly efficient and flexible clusters
  - Kept **SU acceleration** for dense *and* sparse workloads and **efficient DMA data movement**
- Enabled **efficient** scaling though tiling
  - **Wide** physical links at only **3.5%** overhead
  - NoC consumes **3×** less energy than SoA

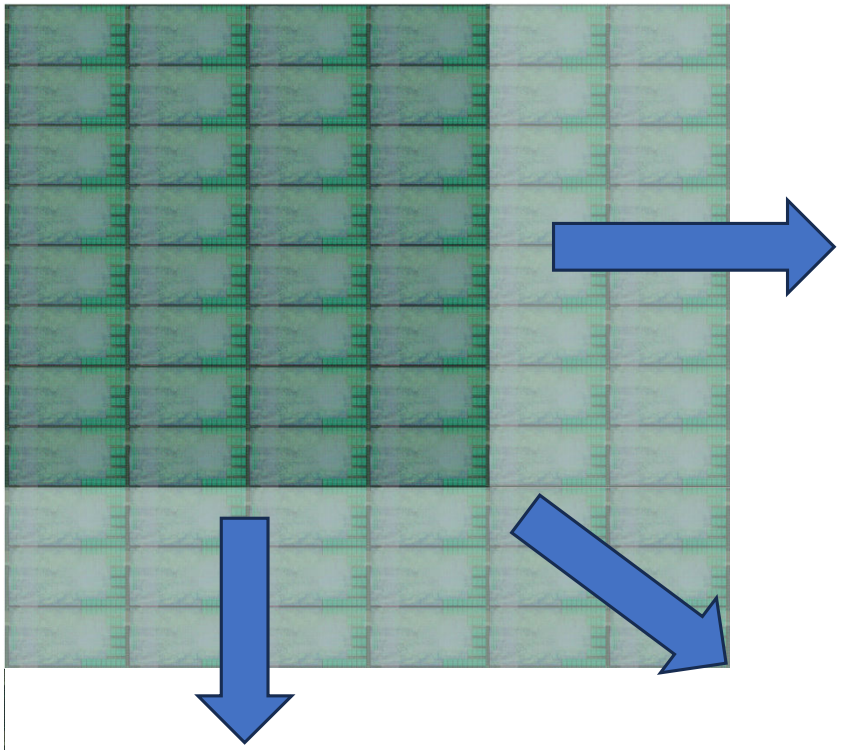


Ramora core: `github.com/pulp-platform/picobello`

FlooNoC: `github.com/pulp-platform/floonoc`

We Need To Go Bigger

- Ramora's *architecture* is scalable, but…
  - It matches Occamy in **chiplet area** and **count**, **technology node**, and **HBM**
  - ➢ Only **47%** higher peak performance
- ➢ Let's **close the gap** to commercial SoA!
  - **More chiplets** with **bigger meshes**
  - More advanced **node** and **HBM**
- Let's also **improve our NoC** further
  - Add NoC-level HW extensions to make data movement more **efficient**

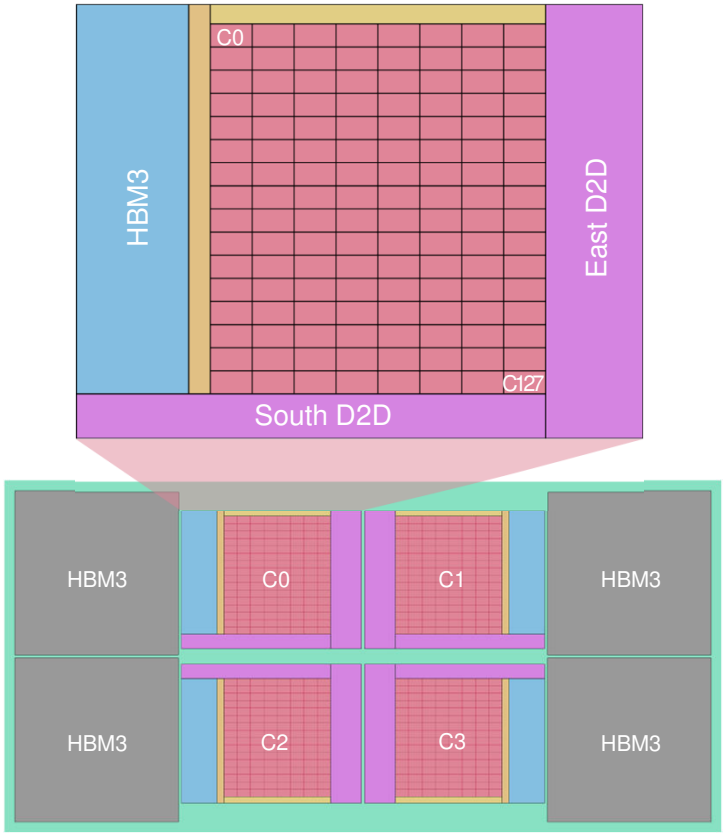


Outline

1. ***Introduction***
2. ***Occamy****:* A silicon-proven open 2.5D RISC-V system
3. ***Ramora:*** Improving Occamy with a NoC
4. ***Ogopogo:*** A quad-chiplet 7nm concept
5. *Toward end-to-end open chiplets*
6. *Conclusion*

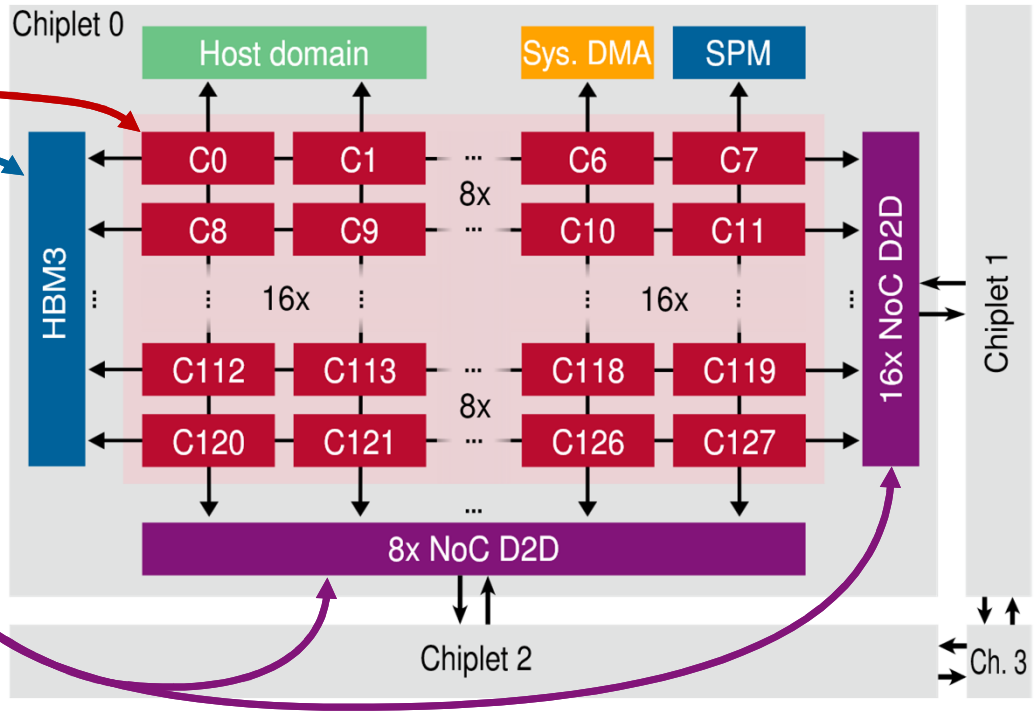
Ogopogo: A 7nm Quad-Chiplet Concept Architecture

- Expands *Ramora* to **8×** more clusters
  - **Four 16×8** chiplets in **TSMC 7nm FinFET**
  - Each with ***two*** **D2D links** and **HBM3**
  - ➢ **10.3 DP-TFLOP/s** peak performance
- Lightweight **NoC transport extensions**
  1. In-router handling of collectives
  2. In-stream vector operations
  3. Packed irregular streams
- ➢ **19%** higher node-normalized compute density than **Nvidia B200**
  - ➢ Perf. gap reduced to a matter of **die size**



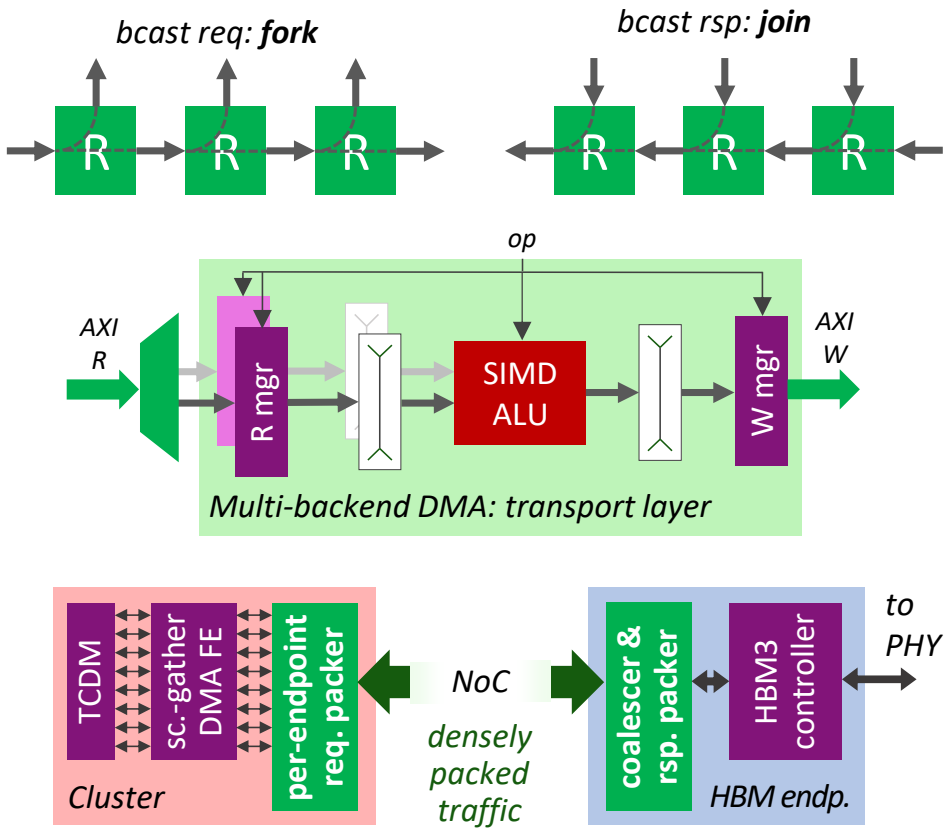
Ogopogo's Chiplets

- **Scale-up** of Ramora's chiplets
  - **128 clusters** in **4×** larger 16×8 mesh
  - **HBM3** with **2× BW** (6.4 Gb/s/pin)
- **Two D2D links** using LVDS PHYs
  - *South:* **1.02 Tb/s** for 8 NoC links
  - *East:* **2.05 Tb/s** for 16 NoC links
- *Host*, *DMA*, *SPM* at top edge
  - Host uses our new *Cheshire* platform: multi-core support, more peripherals



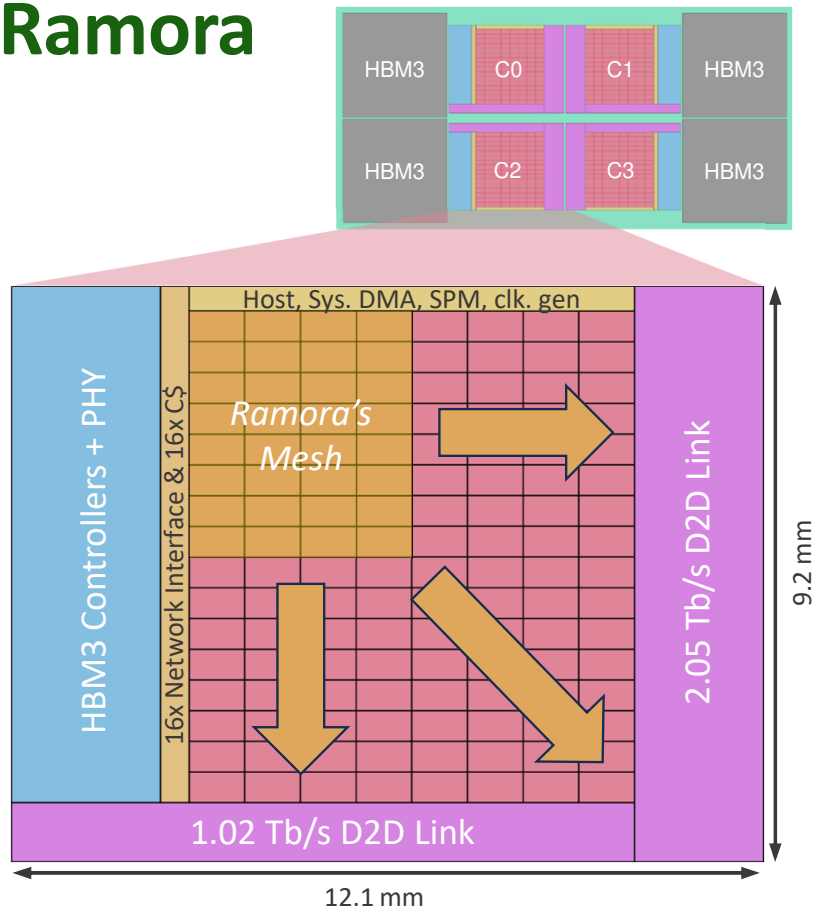
Lightweight Extensions for Efficient Data Movement

1. Handle collectives **inside routers**
   - Extend routers with *fork and join* primitives
   - Can perform **broadcast**, **multicast**, **sync**
   - ➢ **<10%** router area, no timing overhead
2. In-stream **vector operations**
   - Add streaming vector ALUs to cluster DMAs
   - Element-wise and reductive arith. / logic ops
   - ➢ Only **80 kGE** for **32×** / **12×** INT64 speedups
3. **Packed irregular streams**
   - Pack *strided*, *indirect* access streams tightly on NoC links, coalesce at HBM endpoints
   - ➢ **161 kGE** request packer improves random scatter-gather BW efficiency by **4.8×**



Evaluation: A Major Step Up from Ramora

- Estimated full chiplet from TSMC N7 cluster
  - **112 mm²** chiplet area, **1.26 GHz** typical clock
  - **41.1 DP-GFLOP/s/mm²** compute density
  - **64.9 DP-GFLOP/s/W** on FP64 GEMM
- ➢ An **order of magnitude** jump in performance
  - From 1.29 to **10.3 DP-GFLOP/s**
  - **2.8×** higher compute density
  - **1.5×** higher energy efficiency
- Data movement extensions only have **5.3%** area impact on compute domain



Evolution Of Our Designs & SoA Comparison

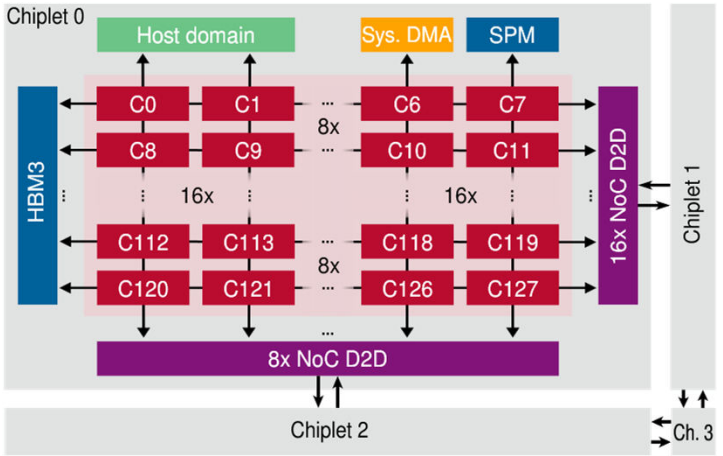
** Normalized to TSMC N7*
[a] Based on B200 render
[b] For FP64 GEMM
[c] Based on Occamy results and cluster, NoC power sim.
[d] For FP64 HPL, leverages FP emulation methods [31]

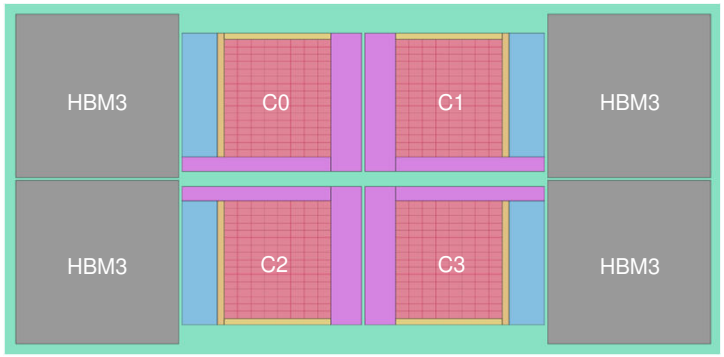
	Occamy	Ramora	Ogopogo	B200 [30]	
Technology Node	GF 12LP+	GF 12LP+	TSMC N7	**TSMC N4P**	
Die / Comp. Area [mm²]	146 / 83.7	143 / 86.4	448 / 251	**1600 / 737** [a]	*~4× less die area…*
HBMx Configuration	2 × 2E	2 × 2E	4 × 3	**8 × 3E**	
Peak Perf. [DP-TFLOP/s]	0.88	1.29	10.3	**40.0**	*…~4× smaller peak*
En. Eff. [DP-GFLOP/s/W]	39.8 [b]	42.2 [b,c]	64.9 [b,c]	**82.1** [d] [31]	*Only 21% lower*
Peak Comp. Density [DP-GFLOP/s/mm2]	10.5 23.1*	14.9 33.0*	41.1 **41.1***	**54.2** [a] 34.4*[a]	*Only 24% lower,* ***19% higher*** *normalized*
Total HBM BW [TB/s]	0.82	0.82	3.28	**8.00**	*Comparable total HBM & D2D BW*
Total D2D BW [Tb/s]	0.07	1.31	8.44	**14.4**	

- **19%** higher node-normalized compute density than B200
- Comparable **energy efficiency** and **HBM, D2D BW**
- Remaining performance gap is a matter of **absolute die size**

Ogopogo: SoA Performance Parity In Sight!

- Scaled Ramora to **4608 cores**
  - **Four TSMC N7** chiplets, **16×8** clusters each
  - **8×** peak performance: **10.3 DP-TFLOP/s**
  - **2.8×** compute density: **41.1 DP-GFLOP/s/mm²**
  - **1.5×** energy efficiency: **64.9 DP-GFLOP/s/W**
- Lightweight **NoC transport extensions**
  - Only **5%** compute-domain area impact
- **19%** higher node-normalized compute density than SoA **Nvidia B200** GPU
  - Comparable **energy eff.** and **HBM, D2D BW**
  - Remaining performance gap down to **die size**



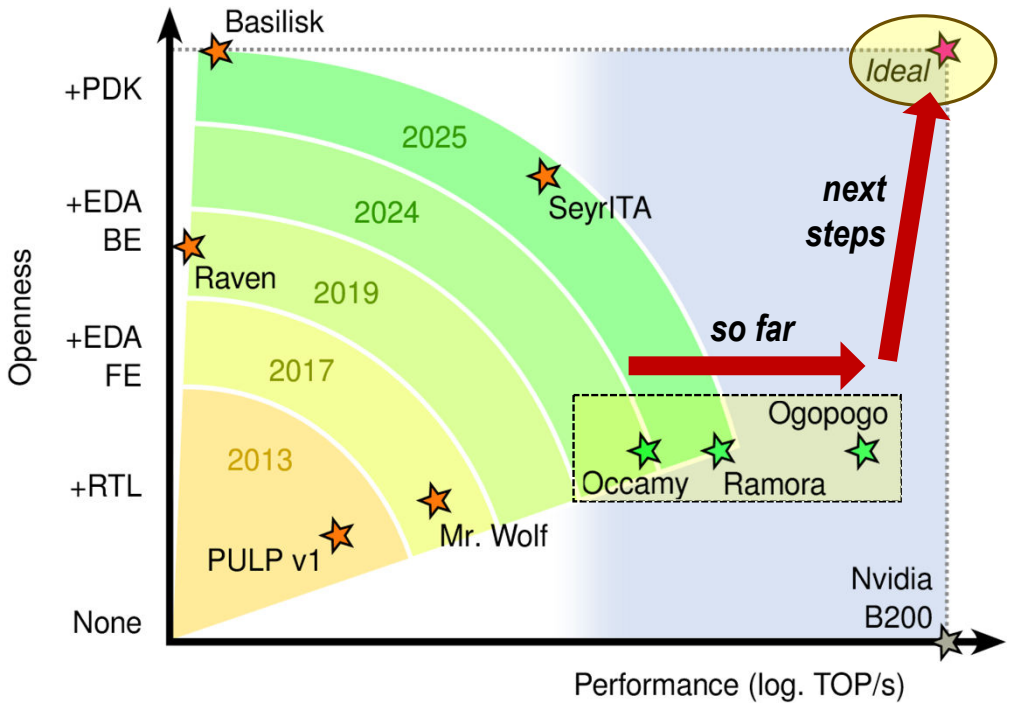


Outline

1. ***Introduction***
2. ***Occamy****:* A silicon-proven open 2.5D RISC-V system
3. ***Ramora:*** Improving Occamy with a NoC
4. ***Ogopogo:*** A quad-chiplet 7nm concept
5. ***Toward end-to-end open chiplets***
6. *Conclusion*

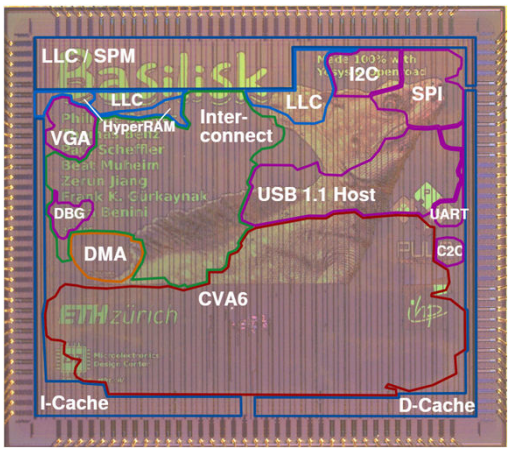
Next Step: Improving 2.5D System Openness

- We caught up on performance, *but*
  - We did not yet improve **openness**!
  - Still only open to logic-core RTL
- Improving openness has **benefits**
  - Design **transparency** (and thus **trust**)
  - Reduced **integration costs**
  - Lower barriers to **collaboration**
- ➢ Explore *opportunities* & *blockers* in
  - Open **simulation** and **EDA tools**
  - Open (manufacturable) **PDKs**
  - Open **off-die PHYs**



Improving Openness: Simulation and EDA

- Open *simulators* are **usable**, but **limited**
  - Our RTL already simulates in *Verilator*
  - Limitations in **gate-level simulation**, (S)Verilog **testbenches**, and **scalability**
    - All actively being improved in Verilator
- Open *EDA tools* are **far behind** commercial SoA, but **catching up**
  - Demonstrated our **host** subsystem in an **end-to-end open-source** 130nm SoC [32]
  - Currently implementing our **cluster** in 22nm node using *Yosys* and *OpenROAD* [33]
    - Full logic core may soon be implementable

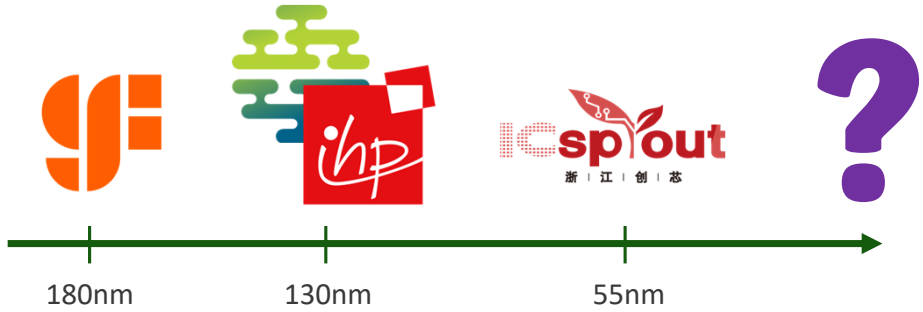


[32]

Improving Openness: PDKs and Off-Die PHYs

- Open PDKs exist **only for mature nodes**
  - GF 180nm, Skywater 130nm, IHP 130nm
  - *Recently announced:* ICSprout 55nm
  - ➢ We need **advanced open PDKs** for our ≤12nm compute chiplets!
- **Off-die PHYs** (e.g. DRAM, D2D links) are currently the **biggest hurdle**
  - Strongly coupled to PDK and EDA tools
  - Open mixed-signal design flows were demonstrated [34], but IP is rare

➢ *Primary challenges:* lack of **advanced PDKs** and difficulty of **open PHY design**



Efabless Recommended Open Source Analog Design Flow [34]

NGSPICE

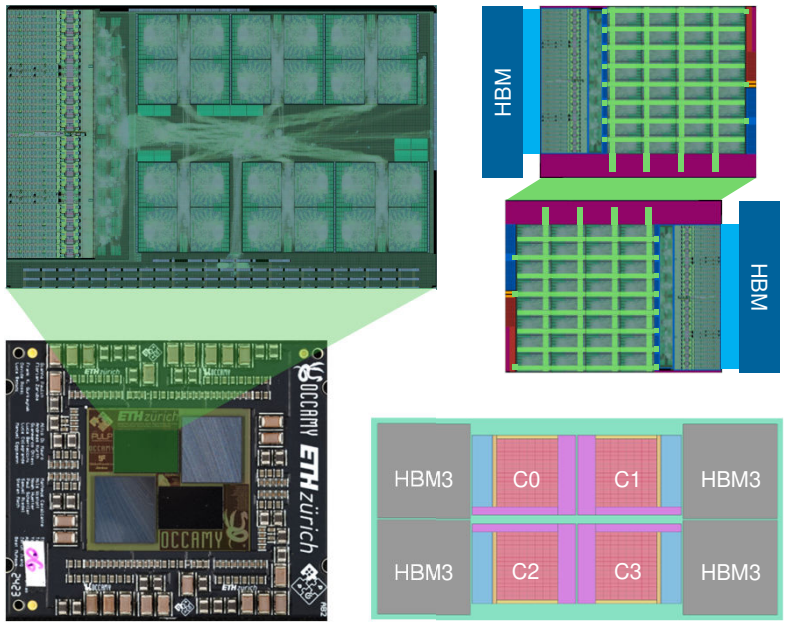
Magic VLSI Layout Tool
Current distribution version 8.3

Outline

1. ***Introduction***
2. ***Occamy****:* A silicon-proven open 2.5D RISC-V system
3. ***Ramora:*** Improving Occamy with a NoC
4. ***Ogopogo:*** A quad-chiplet 7nm concept
5. ***Toward end-to-end open chiplets***
6. *Conclusion*

Conclusion: A Roadmap Toward Open HPC/AI Chiplets

- ***Occamy*** is the first open-RTL RISC-V 2.5D system demonstrated in silicon
  - **Two 12nm** chiplets, **432** cores, **876 DP-GFLOP/s**
  - **89%** regular, **42–83%** irregular workload FP util.
- ***Ramora*** adds a scalable 2D mesh NoC
  - Same area, **576** cores, **1.29 DP-TFLOP/s**
  - **16×** faster **1.04 Tb/s** D2D interface
- ***Ogopogo*** scales to SoA compute density
  - **Four 7nm** chiplets, **4608** cores, **10.3 DP-TFLOP/s**
  - Remaining performance gap down to **die size**
- *Next:* increase **openness** beyond core RTL
  - *Bottlenecks:* **advanced PDKs** and **PHY design**



Occamy: `github.com/pulp-platform/occamy`

Ramora core: `github.com/pulp-platform/picobello`

FlooNoC: `github.com/pulp-platform/floonoc`

References

1. https://epoch.ai/data/ai-models
2. https://www.techpowerup.com/gpu-specs/h100-sxm5-96-gb.c3974
3. https://www.the-waves.org/2022/12/05/chiplet-technology-a-weak-reinvention-core/
4. https://commons.wikimedia.org/wiki/File:Integrated_Circuits_Structure.png
5. https://cdn.mos.cms.futurecdn.net/B8mAVs4Ei3jh5TGrmjJarE-970-80.jpg.webp
6. https://developer.arm.com/documentation/den0145/bet0/?lang=en
7. https://cdn.sanity.io/files/jpb4ed5r/production/bf1d903cec88d24280ede976e5f0e173ac31e95d.pdf
8. https://www.zeroasic.com/_astro/efabric-3d.KOEWxO_l_Z27XSrO.webp
9. https://ieeexplore.ieee.org/document/9731627
10. https://ieeexplore.ieee.org/document/9063103
11. https://ieeexplore.ieee.org/document/9365803
12. https://ieeexplore.ieee.org/document/10070122
13. https://dl.acm.org/doi/abs/10.1007/s42979-024-02958-3
14. https://ieeexplore.ieee.org/document/9556099
15. https://dl.acm.org/doi/10.1145/3524059.3532370
16. https://www.top500.org/lists/top500/2024/06/
17. https://blocksandfiles.com/2024/07/03/australia-dell-based-virga-ai-workload-cluster/
18. https://www.top500.org/lists/green500/2024/06/
19. https://journals.sagepub.com/doi/10.1177/10943420211065723
20. https://ieeexplore.ieee.org/document/9426456
21. https://dl.acm.org/doi/10.1145/3577193.3593716
22. https://dl.acm.org/doi/10.1145/3627535.3638476
23. https://ieeexplore.ieee.org/document/9307836
24. https://ieeexplore.ieee.org/document/9956740
25. https://ieeexplore.ieee.org/document/9460505
26. https://ieeexplore.ieee.org/document/8327053
27. https://ieeexplore.ieee.org/document/8903494
28. https://ieeexplore.ieee.org/document/9241710
29. https://ieeexplore.ieee.org/document/10454572
30. https://nvdam.widen.net/s/wwnsxrhm2w/blackwell-datasheet-3384703
31. https://ieeexplore.ieee.org/document/11196413
32. https://ieeexplore.ieee.org/document/11154384
33. https://github.com/open-source-eda-birds-of-a-feather/open-source-eda-birds-of-a-feather.github.io/blob/main/doc/slides_2025/BOF25_PULP_mbertuletti.pdf
34. http://www.opencircuitdesign.com/analog_flow/