Is an AI Accelerator All You Need? Overcoming Amdahl's Law With Tightly-Coupled Specialized Accelerators

Angelo Garofalo
ETH Zürich, Switzerland & University of Bologna, Italy

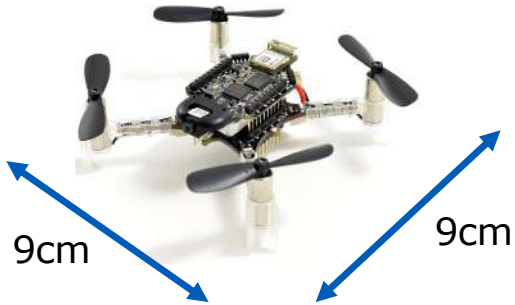
ISSCC 2023, FORUM 6
"The Future of Heterogeneous Multi-Core Architectures for AI and Other Specialized Processing"

Outline

- The quest for energy-efficient AI at the Edge of the IoT
  - The performance-power proportionality goal
  - Dealing with the end of the Dennard's scaling
- Low-power heterogeneous computing: mitigating deep acceleration effects
  - Multi-core heterogeneous systems
  - Tightly-coupled digital acceleration
  - Heterogeneous analog in-memory computing (AIMC) systems
- Insights for scaled-up heterogeneous powerful architectures
  - AIMC many-core architecture
  - Asymmetric chiplet-based systems
- Conclusions & Outlook

The Quest for Advanced AI Computing at the Edge

Object detection

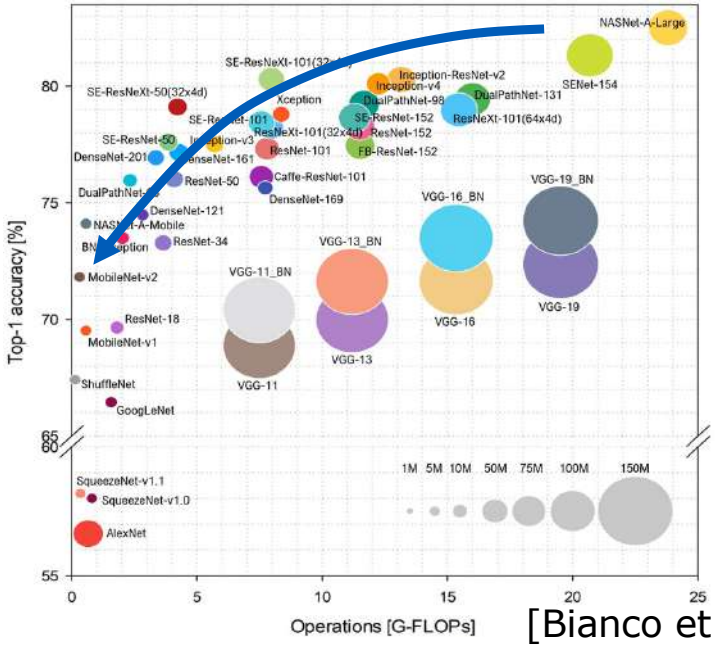


Obstacle avoidance & Navigation

Environment exploration

- Computing requirements for Edge AI
  - AI capabilities under tight power (passive cooling, few hundreds mW of budget)
  - Real-time constraints (Latency as important as Throughput)
  - Tight area budget (on-chip memory limited to ~1MB)

Reduced Precision DL Models



[Bianco et al. IEEE Access, 2018]

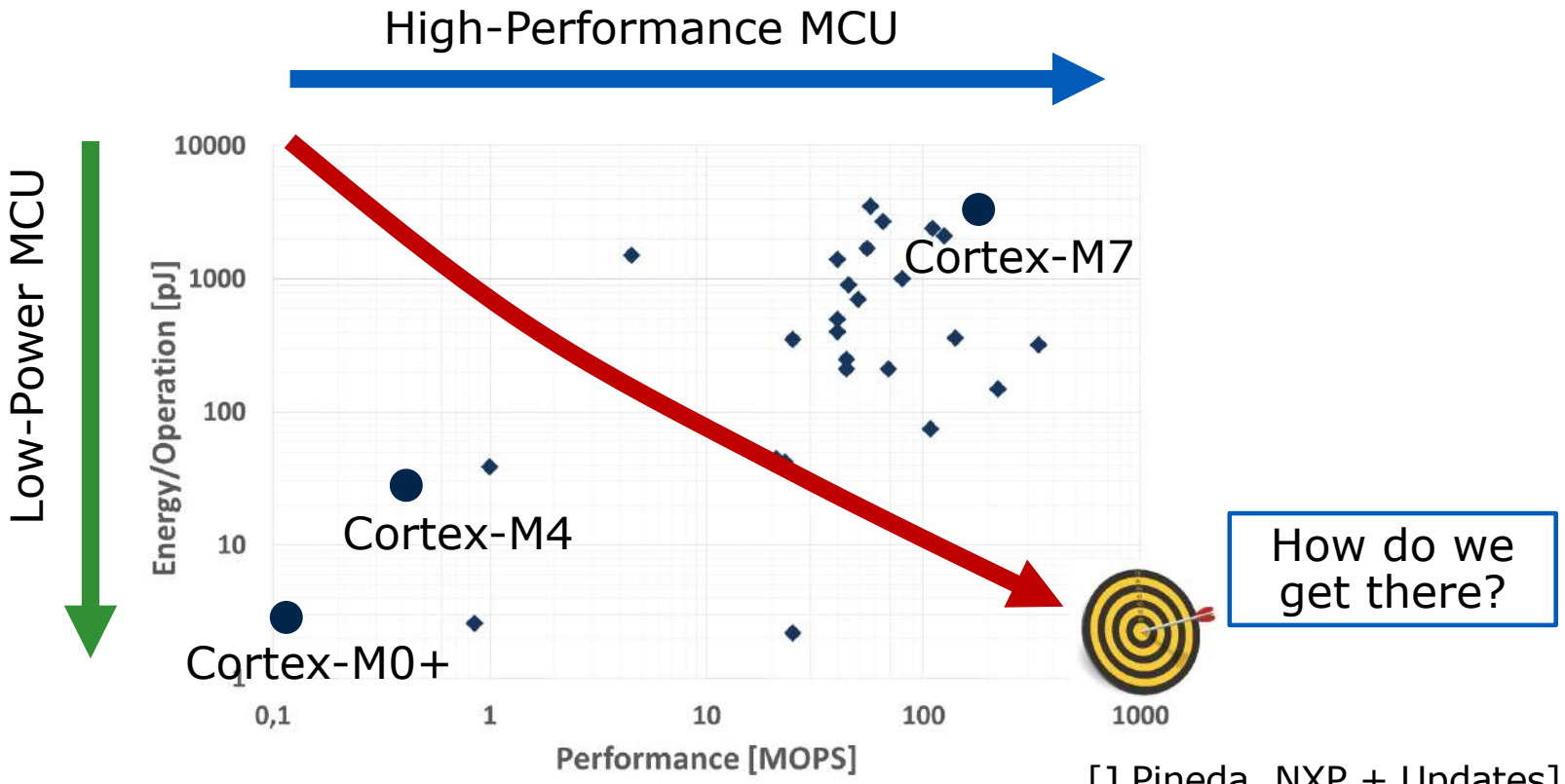
- **Quantized MobileNet-v1 model**

Quantization Method	Top1 Accuracy	Weight Memory Footprint
Full-Precision	70.9%	16.27 MB
INT-8	70.1% **0.8%**	4.06 MB **4x**
INT-4	66.46% **4.4%**	2.35 MB **7x**
Mixed-Precision	68% **2.9%**	2.09 MB **8x**

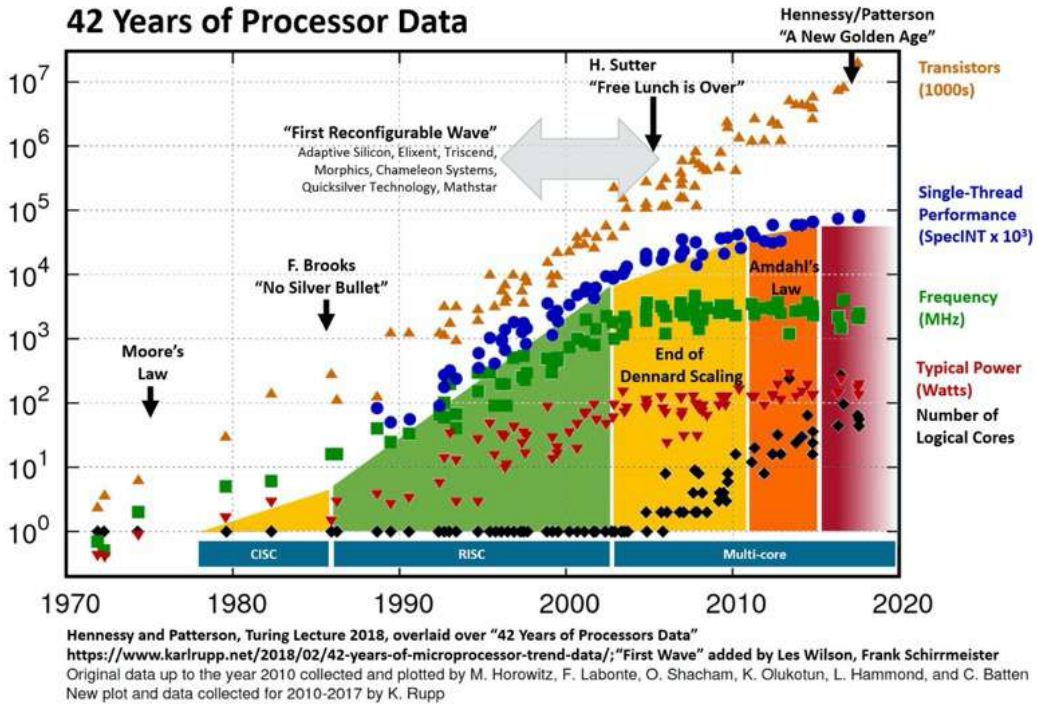
[Rusci et al., PML&S (2020)]

- MVM workload dominates (massive parallelism)
- Strong resiliency to Quantization (low-precision integer arithmetic is ok)
- Mixed-precision quantized neural networks (QNNs) are the natural target for Edge AI

The Performance-Power Proportionality Goal



Technology Scaling is Not Enough

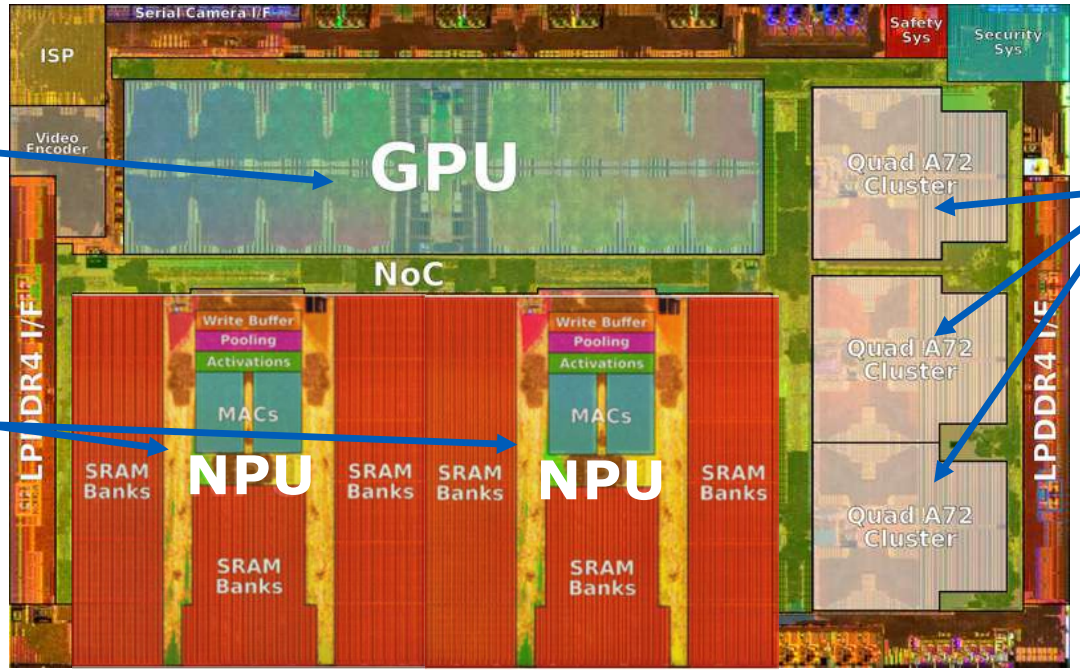


- Increase number of processing cores (..but up to how many cores?)
- Hardware specialization (..but what about flexibility?)

HP Systems: Tesla Full Self Driving SoC Example

GPGPU for post-processing (single & double FP) 600FLOPS@1GHz;

NPU for NN workloads 36.8TOPS/NN @2GHz; (INT8 precision)



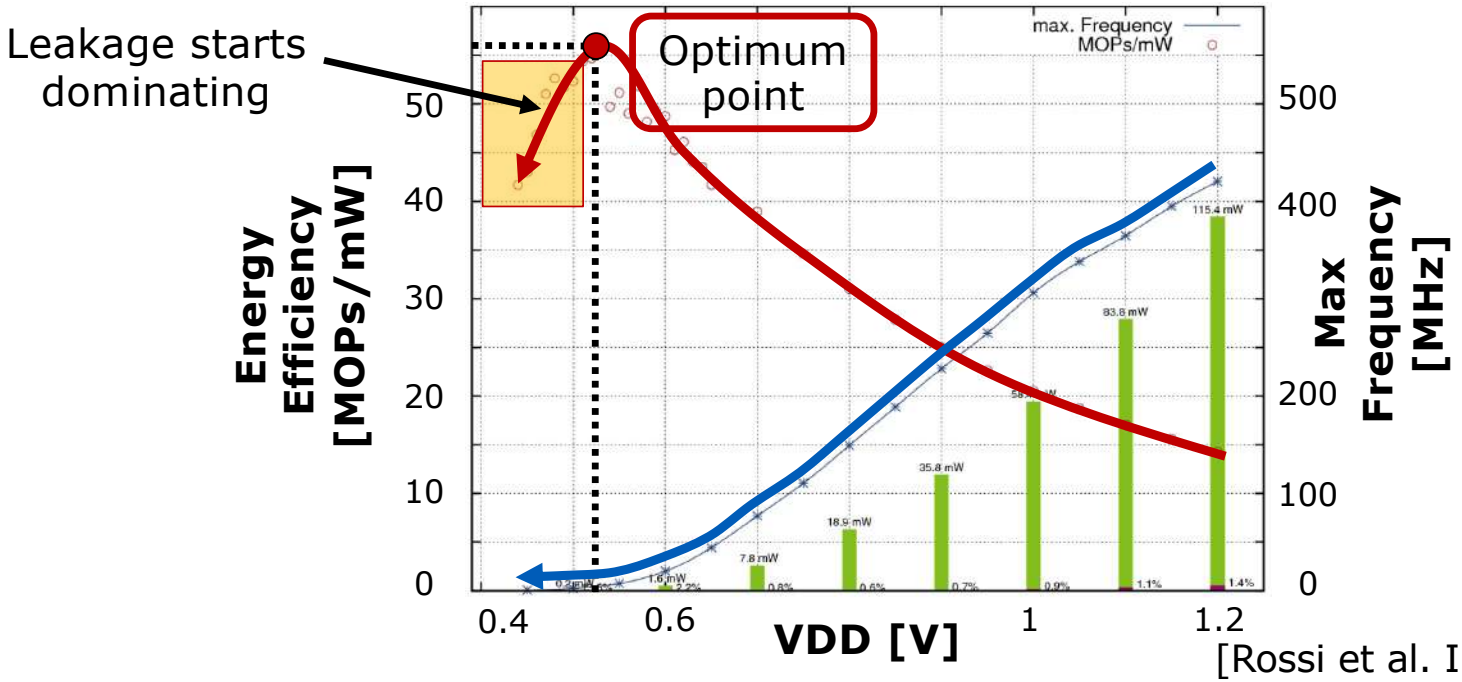
Three clusters of Quad Arm Cortex-A72 for GP processing (2.2GHz)

- Full Self-Driving (FSD) SoC in 14nm FINFET technology
- 6 B Transistors within 265mm^2 die area, ~100W Power Envelope
- 68 GB/s on-chip BW

Outline

- The quest for energy-efficient AI at the Edge of the IoT
  - The performance-power proportionality goal
  - Dealing with the end of the Dennard's scaling
- Low-power heterogeneous computing: mitigating deep acceleration effects
  - Multi-core heterogeneous systems
  - Tightly-coupled digital acceleration
  - Heterogeneous analog in-memory computing (AIMC) systems
- Insights for scaled-up heterogeneous powerful architectures
  - AIMC many-core architecture
  - Asymmetric chiplet-based systems
- Conclusions & Outlook

Near-Threshold Parallel Computing



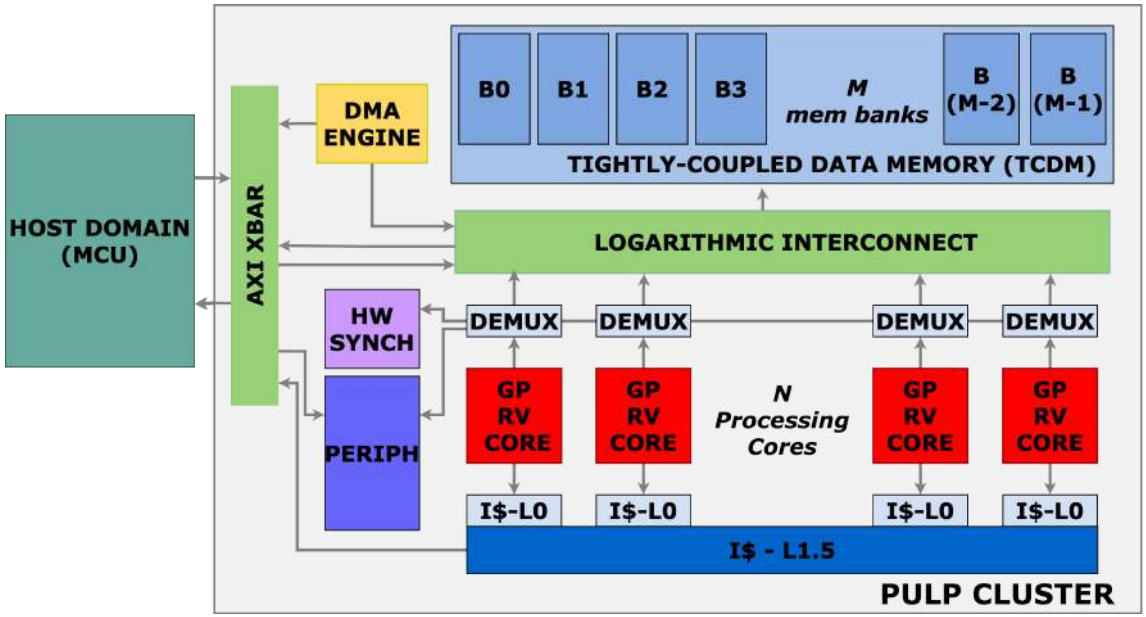
AI

Parallelism
Low-Precision

[Rossi et al. IEEE Micro, 2017]

- SIMD insns rather than super-scalar execution for high energy-efficiency
- More PEs at the optimum energy point working on parallel workloads
- .. To achieve high performance-power proportionality (1pJ/Op @ GOPS)

Parallel Ultra-Low Power (PULP) Platform

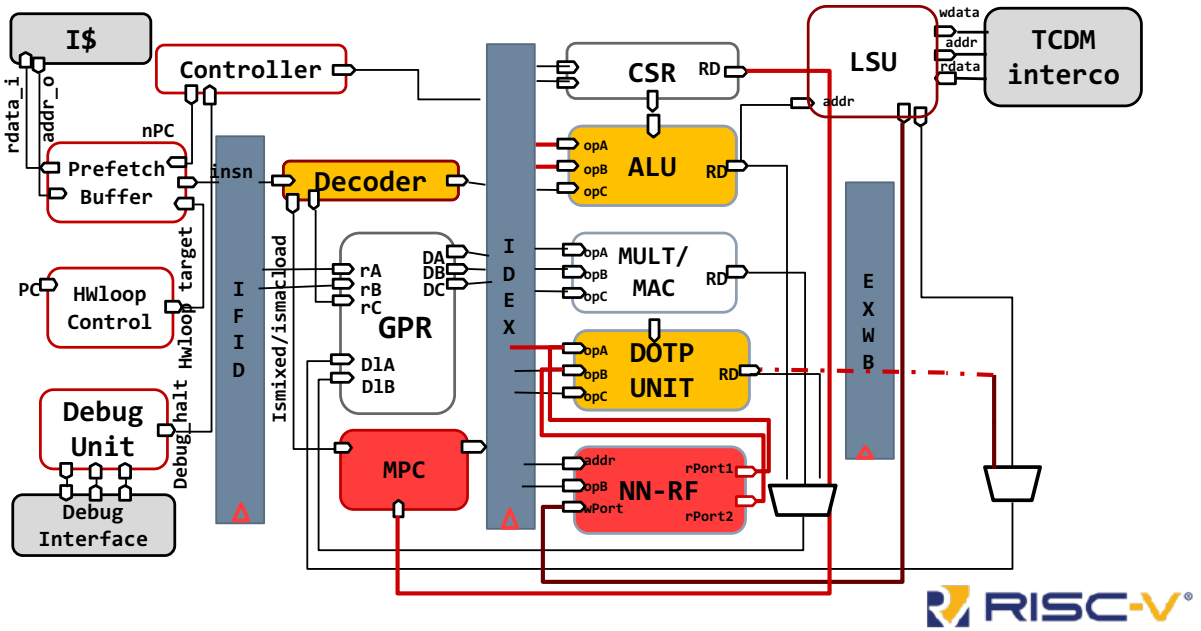


[Rossi et al. JSSC 2022]

- *N* Specialized RISC-V cores (domain-specific ISA instructions)
- Single-cycle latency Logarithmic Interconnect towards TCDM
- SW-managed (DMAs) tightly-coupled data mem (TCDM) rather than data caches
- Synchronization in HW for low-energy thread dispatching and PEs synch

Processor Specialization

4-stage, single in-order issue pipeline



- **[Baseline] RV32IMC:** not good for machine learning workload (~44kGE)
- **[DSP] Xpulp**: 16/8-bit SIMD, HW loops, post-modified LD/ST & bit-manip. insns (~55kGE)
- **[AI] XpulpNN**: 4-bit, 2-bit SIMD (and mixed-precisions) & fused MAC-Load ops (~65kGE)

DNN Acceleration Through Domain-Specific ISA

8-bit Convolution Kernel

RV32IMC

N

```
addi    a0,a0,1
addi    t1,t1,1
addi    t3,t3,1
addi    t4,t4,1
lbu     a7,-1(a0)
lbu     a6,-1(t4)
lbu     a5,-1(t3)
lbu     t5,-1(t1)
mul     s1,a7,a6
mul     a7,a7,a5
add     s0,s0,s1
mul     a6,a6,t5
add     t0,t0,a7
mul     a5,a5,t5
add     t2,t2,a6
add     t6,t6,a5
bne     s5,a0,1c000bc
```

RV32IMCXpulp [DSP]

N/4

```
lp.setup
p.lw  w1, 4(a0!)
p.lw  w2, 4(a1!)
p.lw   x1, 4(a2!)
p.lw   x2, 4(a3!)
pv.sdotsp.b   s1, w1, x1
pv.sdotsp.b   s2, w1, x2
pv.sdotsp.b   s3, w2, x1
pv.sdotsp.b   s4, w2, x2
End: …
```

HW Loop

LD/ST with post increment

8-bit SIMD sdotp

9x less insns than RV32IMC
22% area & **9%** power overhead

RV32IMCXpulpNN [AI]

N/4

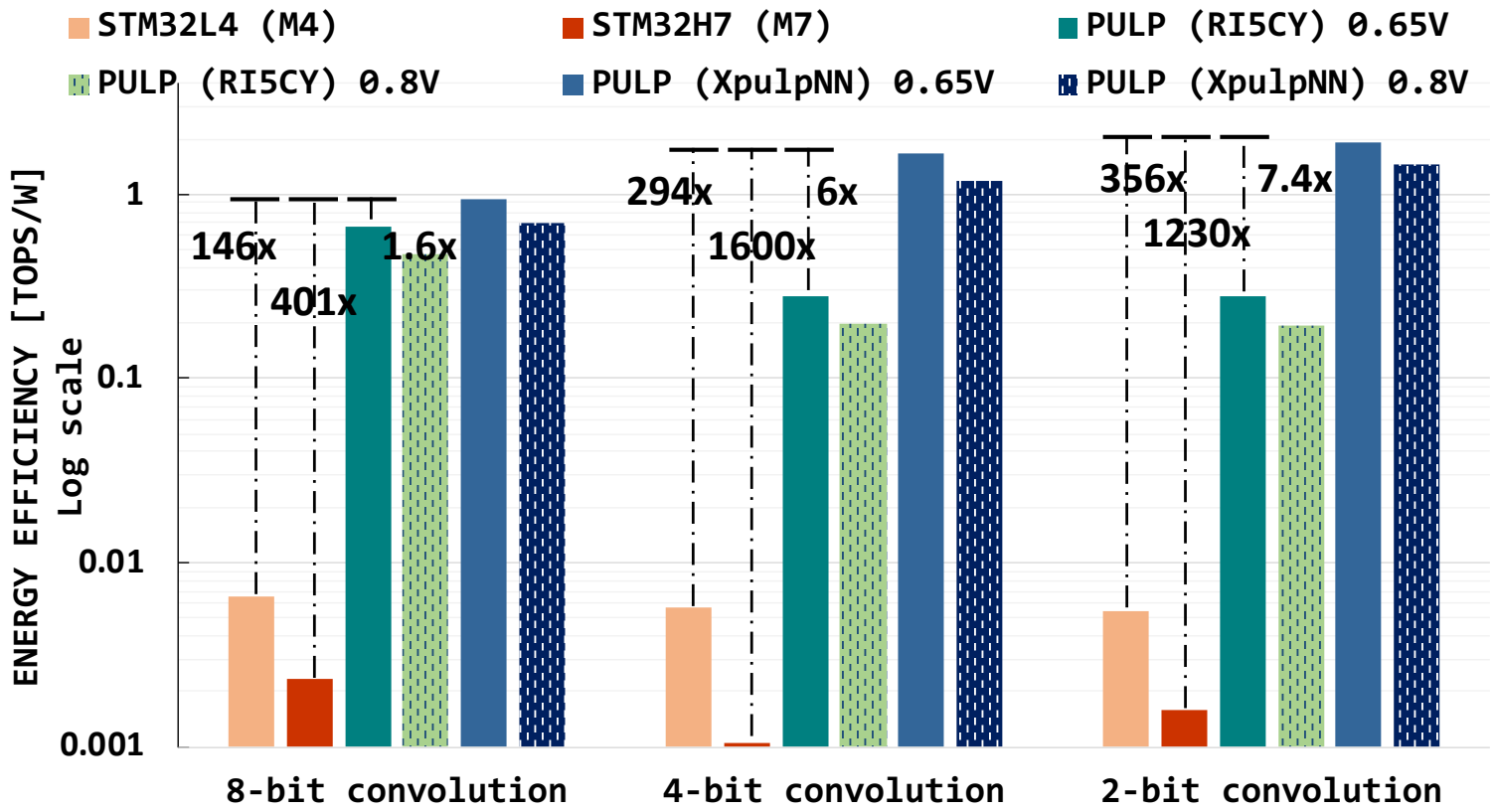
```
Init NN-RF (outside of the loop)
lp.setup
pv.nnsdotup.h  s0,ax1,9
pv.nnsdotsp.b  s1, aw2, 0
pv.nnsdotsp.b  s2, aw4, 2
pv.nnsdotsp.b  s3, aw3, 4
pv.nnsdotsp.b  s4, ax1, 14
end
```

8-bit SIMD Fused MAC-Load

Up to 1.7x less insns than Xpulp
Up to 94% Dotp-Unit utilization
18% & **2.5%** extra-area & power

- Power ⇧ (slightly); Latency ⇩ ⇩ ⇩(>10x) on QNN kernels ⇛ Energy ⇩ ⇩
- Parallelization of the workload (multi-core) for high throughput

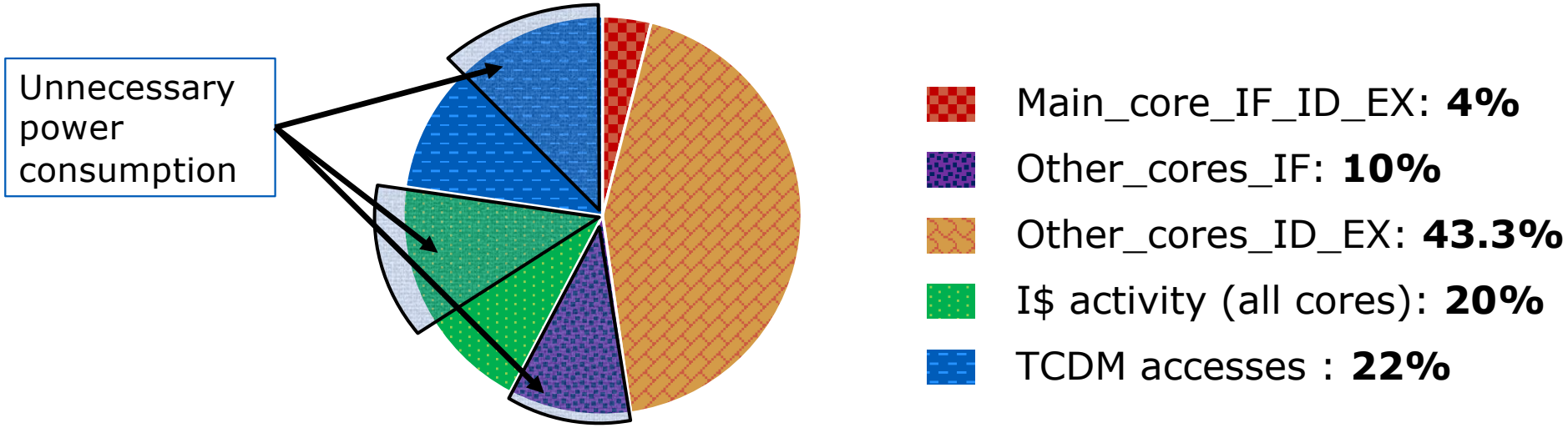
8 RVXpulpnn Cores Cluster (22nm)



[Garofalo et al. IEEE TETC, 2021]

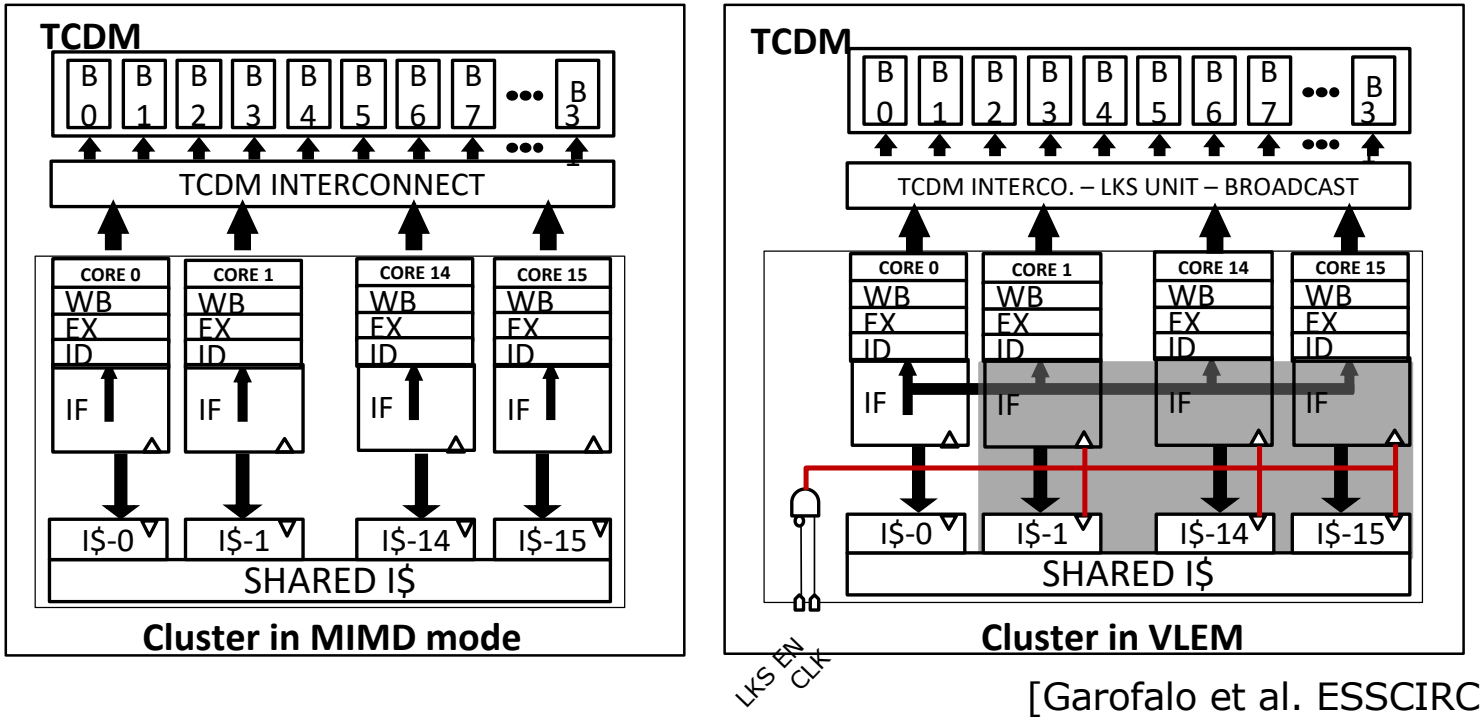
Multi-Core Heterogeneous Computing

Power analysis of a parallel 8-bit x 4-bit convolution



- Reduce unnecessary power consumption (not spent in computation)
- Exploit convolution's instruction and memory data access pattern regularity
- Increase energy efficiency at low extra-area cost
  - → reconfigurable MIMD/SIMD architecture

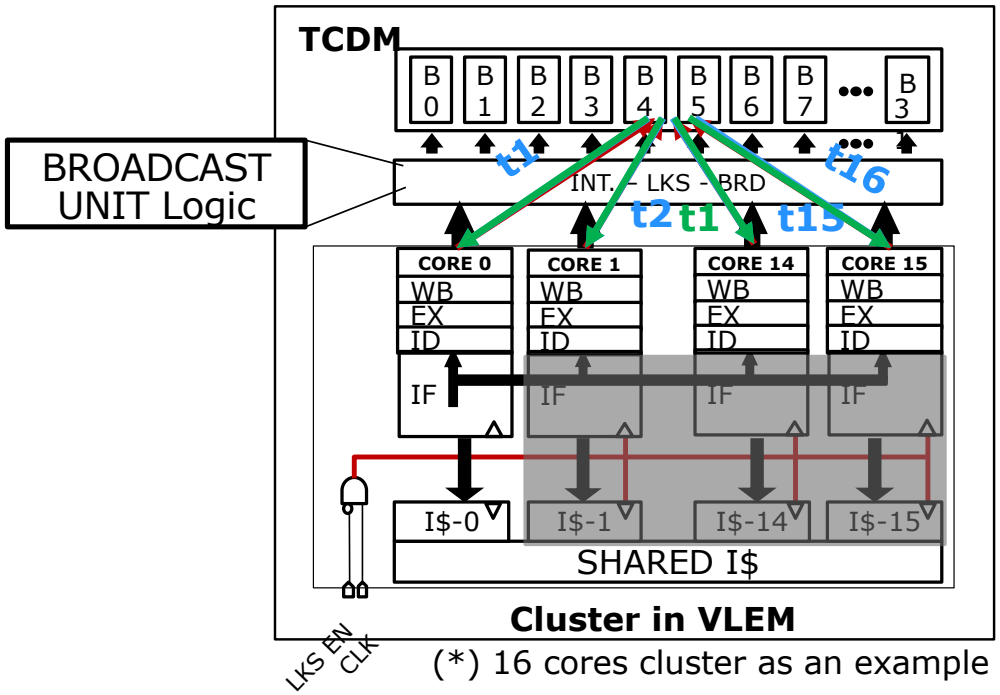
Multi-Core Heterogeneous Computing - 2



[Garofalo et al. ESSCIRC 2021]

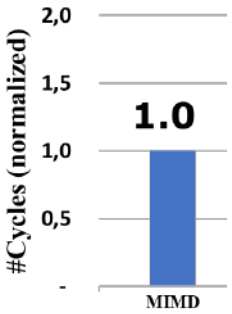
- Cores enter in SIMD (VLEM) mode when executing regular kernels (in two clock cycles)
- In SIMD, instruction flow orchestrated only by the MAIN core → Less energy
- Cores resume in MIMD mode on divergent branches (..or control tasks) → Flexibility

Multi-Core Heterogeneous Computing - 3



(*) 16 cores cluster as an example

Convolution exec. kernels

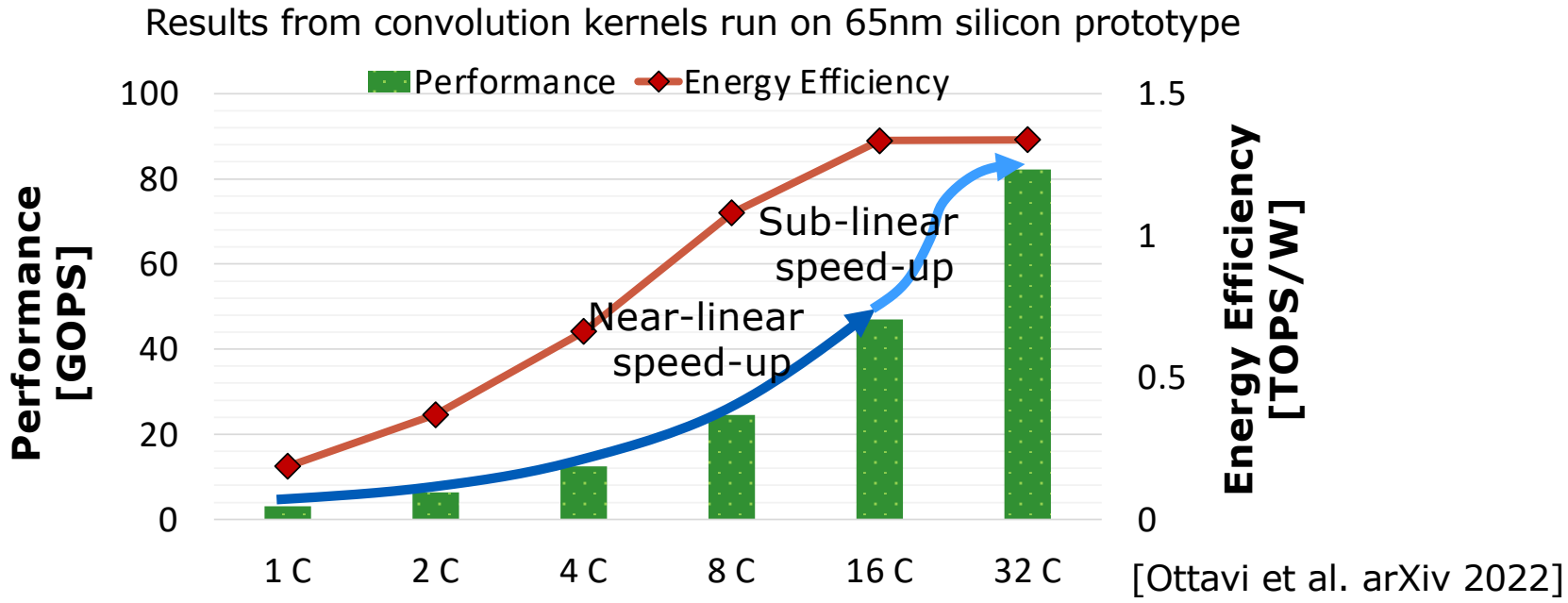


~44% energy saving w.r.t. MIMD mode

<10% area overhead w.r.t. MIMD only cluster

- **Overhead**: many clk cycles to unlock execution in case of concurrent accesses
  - Eliminate overhead to access at same address → **BROADCAST UNIT**
  - Misalign static data and stacks to avoid accesses to the same mem bank

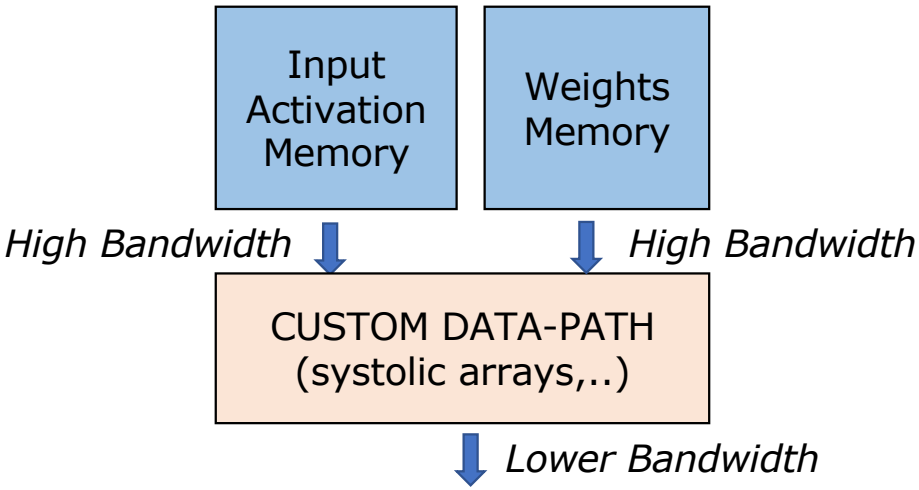
System Scalability

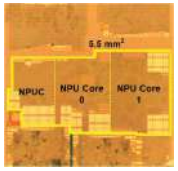


[Ottavi et al. arXiv 2022]

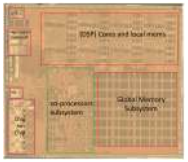
- Performance bottleneck at the core's boundaries (single 32-bit data port)
- Energy efficiency bounded by limited scalability of low-latency local interco
- Custom datapaths to improve throughput and efficiency..
  - ..but at which cost?

Specialized AI Accelerators - Digital

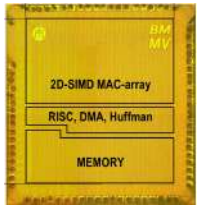




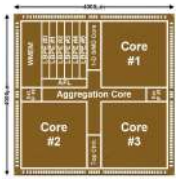
[Song et al., ISSCC19, 8nm]



[Desoli et al., ISSCC17, 28nm]



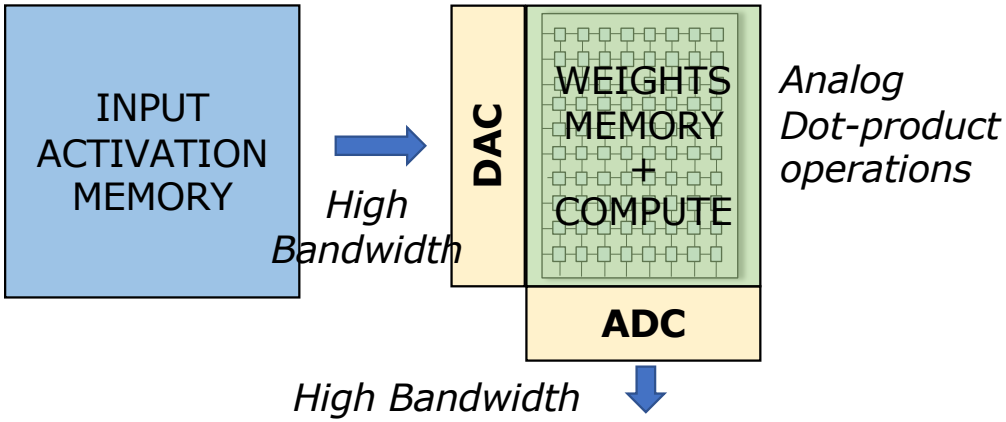
[Moons et al. ISSCC 2017]



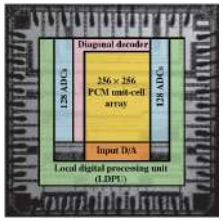
[Lee et al., ISSCC18, 65nm]

- Custom MAC units, data-paths, data precision (binary, ternary, 2-8 bits)
- Perf. **[0.1, 1] TOPS**, En. Eff. **[1, 100] TOPS/W**
- Hardware challenge moved at the boundaries, on data movements

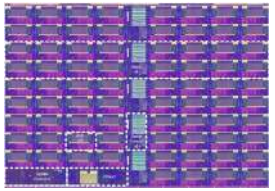
Specialized AI Accelerators - Analog



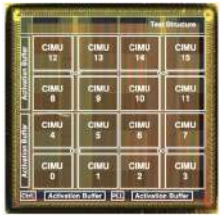
Not in scale



[Khaddam et al. JSSCC, 2022]



[Fick et al. ISSCC 2022]



[Jia et al. ISSCC 2021]

- Perf.: **[10, 100] TOPS**
- En. Eff. **[100, 1000] TOPS/W**

- Weights are stored where the computation happens (nvAIMC arrays)
- Analog MAC more efficient than digital MAC (10x better perf. and en. eff.)
- Energy efficiency vs. accuracy
- Compute density vs. re-programmability

Edge AI Processors

ISA-based multi-core acceleration

Custom digital data-paths

Analog In-Memory Computing cores

1-5pJ/OP

0.05-1 pJ/OP

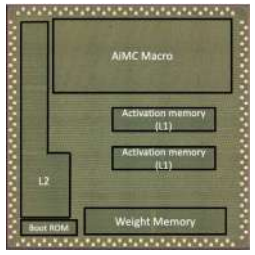
10-100 fJ/OP

SW programmable

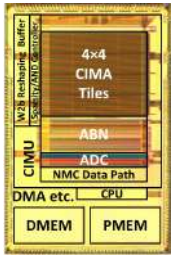
Re-configurable

Only MVMs

- Heterogeneous integration for flexible and efficient AI computation
  - But this introduces some challenges…



[Ueyoshi et al. ISSCC 2022]



[Jia et al. JSSC 2020]

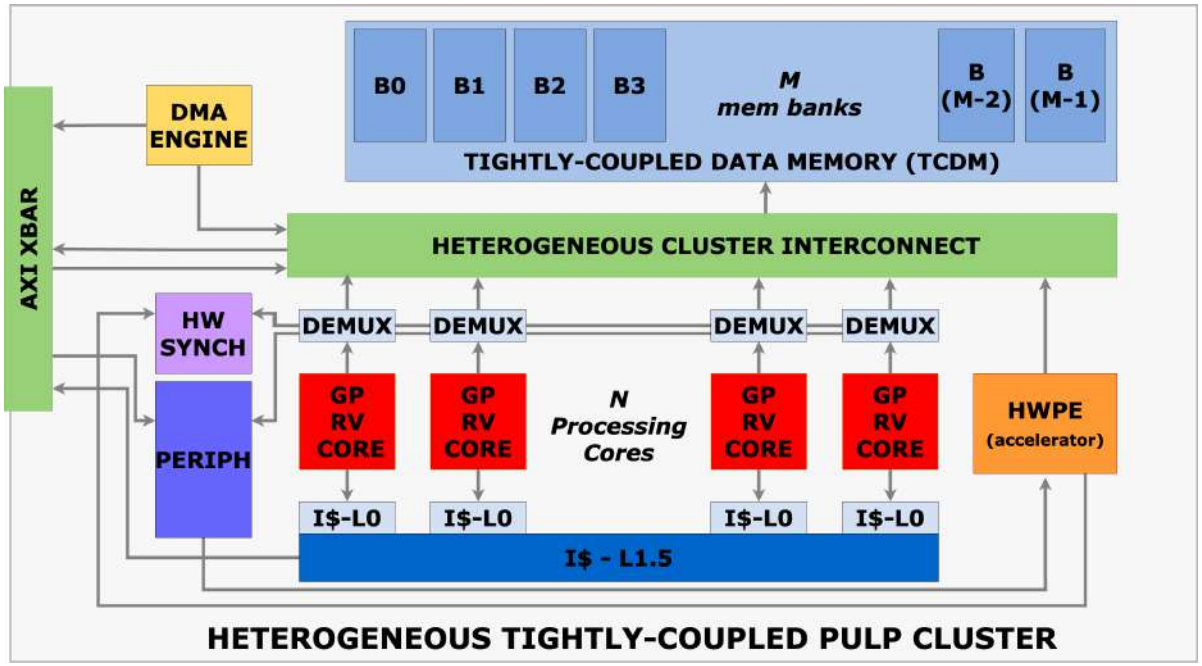
[Rossi et al. JSSC 2022]

Towards Heterogeneous Integration

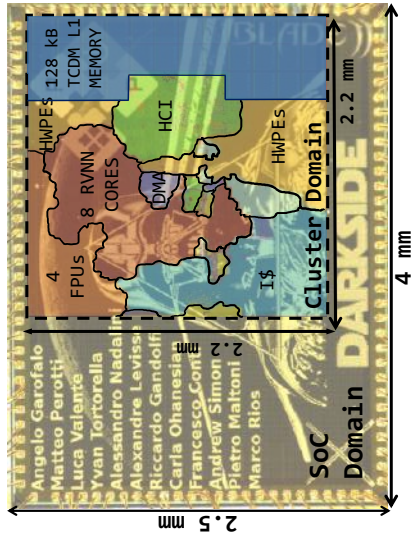
- Performance bottleneck keeps shifting toward non-accelerated kernels
- (as consequence of Amdahl's effect)
  - Solutions
  - Balance workload through **many acceleration sub-systems**
  - Residual computation perfomed in **SW**

- Hardware challenge becomes efficient data movements within the system
  - Solutions
  - High-bandwidth **low-latency** interconnect
  - Reduce data-exchange latency through **tightly-coupled integration** schemes

Tightly-Coupled Digital Acceleration



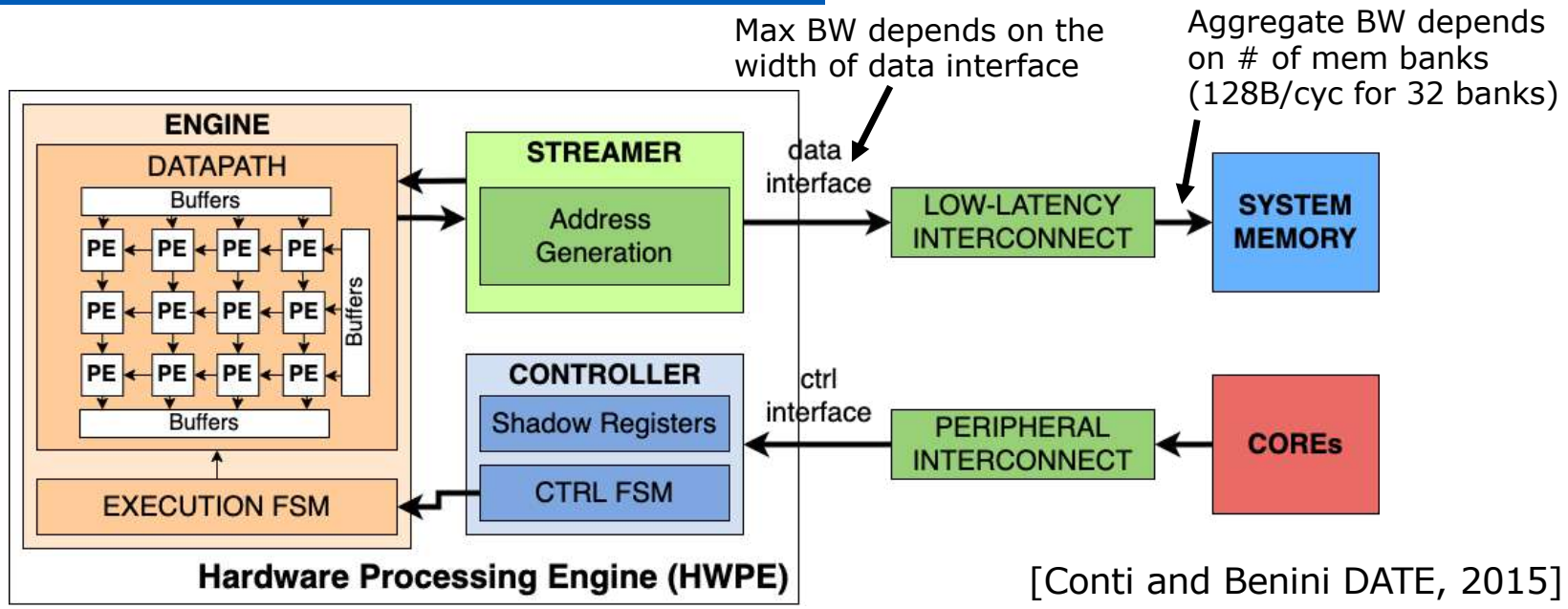
SoC layout, 65nm



[Garofalo et al. OJSSCS, 2022]

- Acceleration with flexibility: zero-copy HW/SW cooperation
- Many tightly-coupled accelerators for Amdahl's effects mitigation
- HCI: Low-Latency (one clk cyc), High-Bandwidth (76B/cyc)

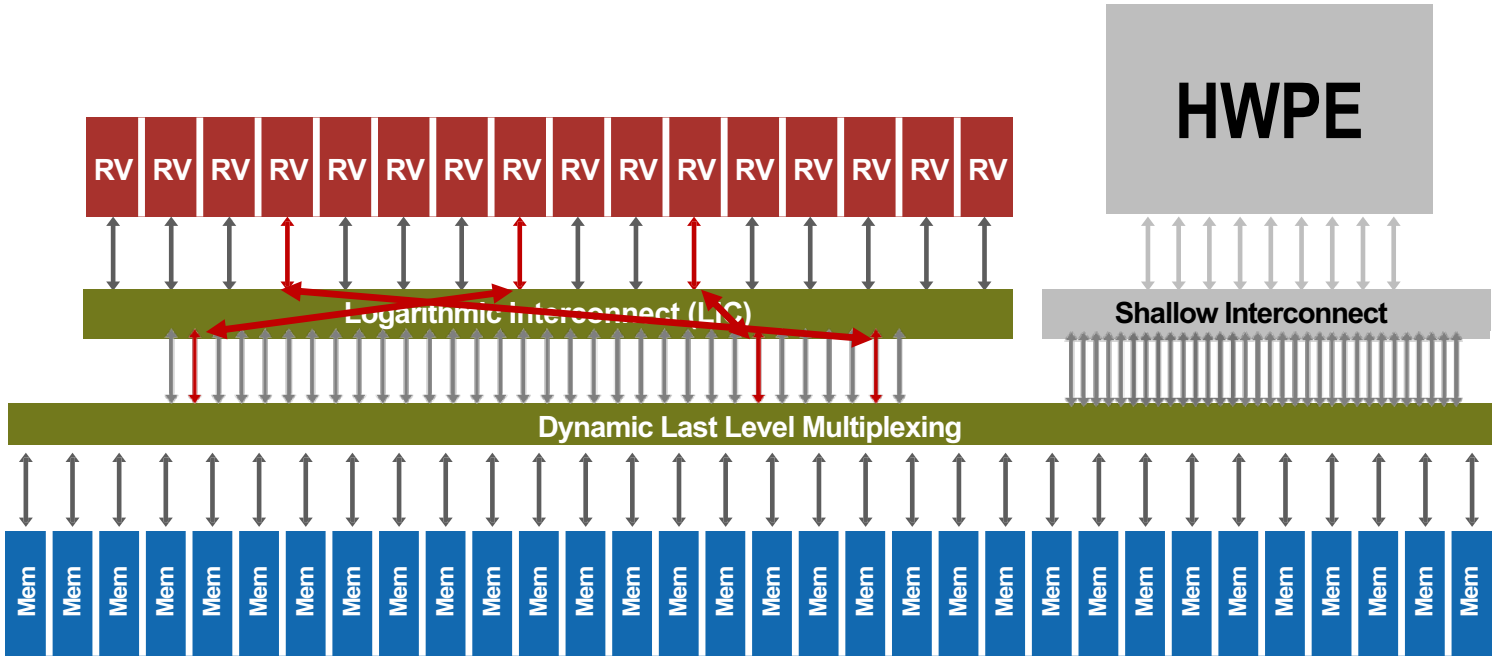
Hardware Processing Engines (HWPEs)



[Conti and Benini DATE, 2015]

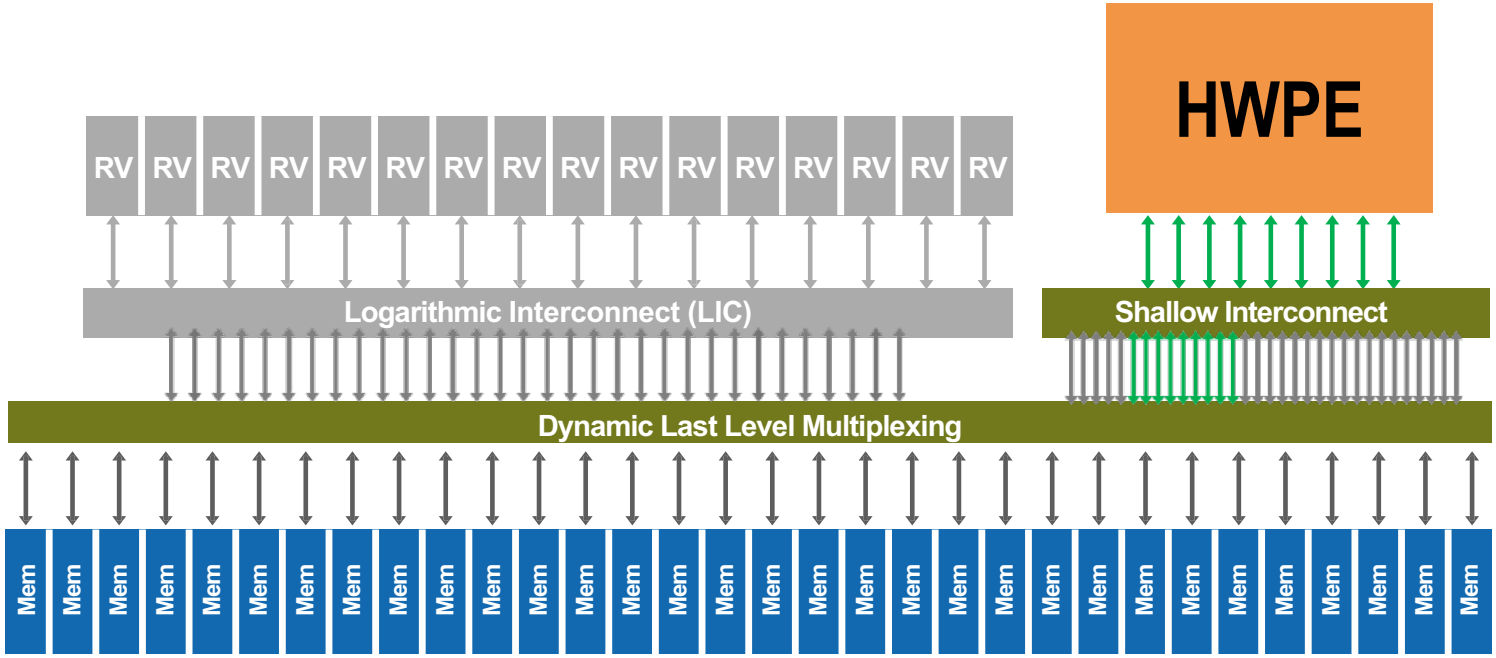
- HWPE efficiency:
  - Dedicated control (no I-fetch) with shadow registers (overlapped config-exec)
  - Specialized high-BW interconnect towards L1 buffer (on data interface)
  - Specialized compute datapath & minimal internal storage (e.g. line buffers)

Heterogeneous Cluster Interconnect



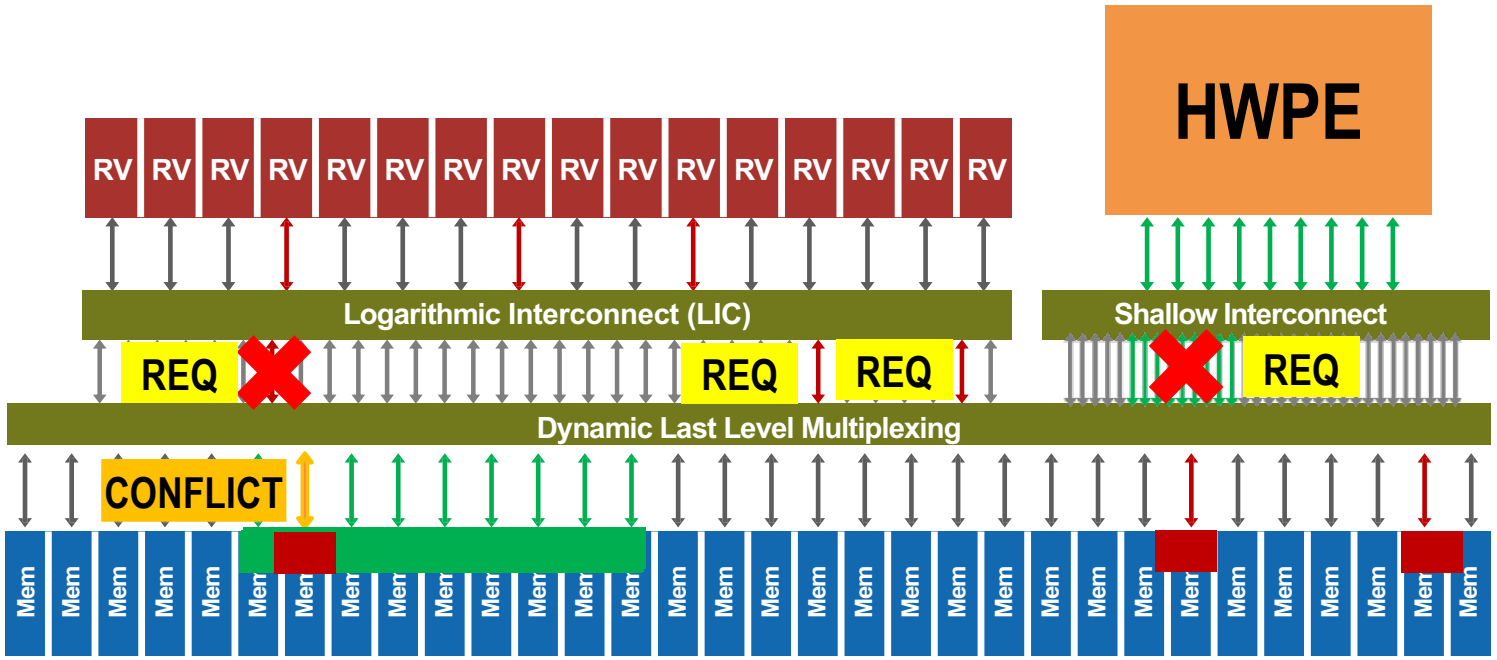
- Processors with their narrow memory ports are arbitrated in the LIC

Heterogeneous Cluster Interconnect - 2



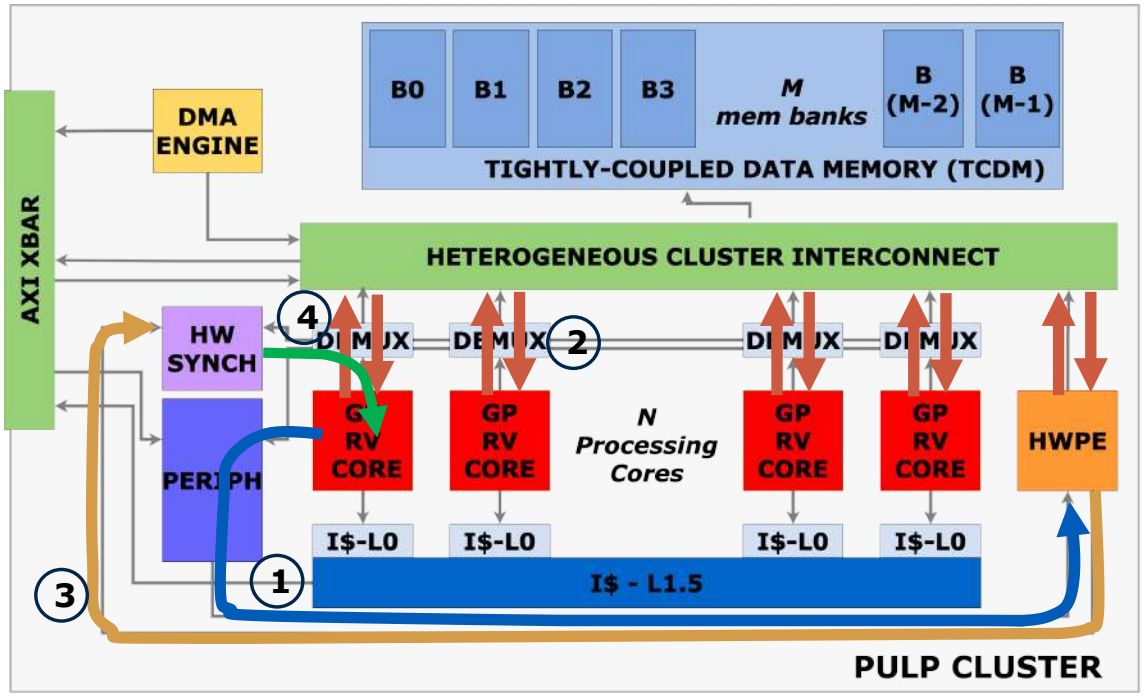
- HWPE exposes a unified high-bandwidth port (e.g. 256-bit) towards the shallow interconnect (simple ”wide” port reordering block without self-contention)

Heterogeneous Cluster Interconnect - 3



- Final level dispatches accesses to single 32-bit memory banks and manages conflicts by means of rotating configurable priority (e.g. max 3 stall cycles)
- If no conflicts, HWPEs and cores can access concurrently the memory.

HW Synchronizer

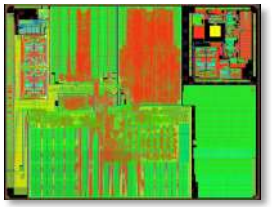


[Glaser et al. IEEE TPDS 2020]

- Fast and low-energy thread dispatching and synchronization
- Automatic clock gating of PEs waiting on a barrier
- Event-based computation (between accelerators/DMAs and GP cores)

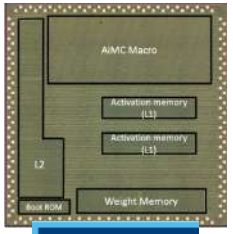
Tightly-Coupled Acceleration: Success Stories

Darkside SoC with many HWPEs: **transposer, systolic array, depth-wise engine**..
[Garofalo et al. ESSCIRC 2022]



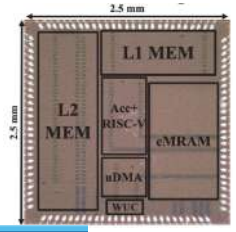
DOLPHIN INTEGRATION
The enabler of low-power Systems-on-Chip

Marsellus SoC with **RBE**
ISSCC 2023 (to appear)



KU LEUVEN

Diana SoC with **AIMC**
[Ueyoshi et al. ISSCC 2022]



KU LEUVEN MAGICS

TinyVers SoC with **HWPE**
[Jain et al. IEEE VLSI 2022]

GREENWAVES TECHNOLOGIES

GAP9 SoC with **NE16**
(commercial)

GAP9 with HWPE (NE16)

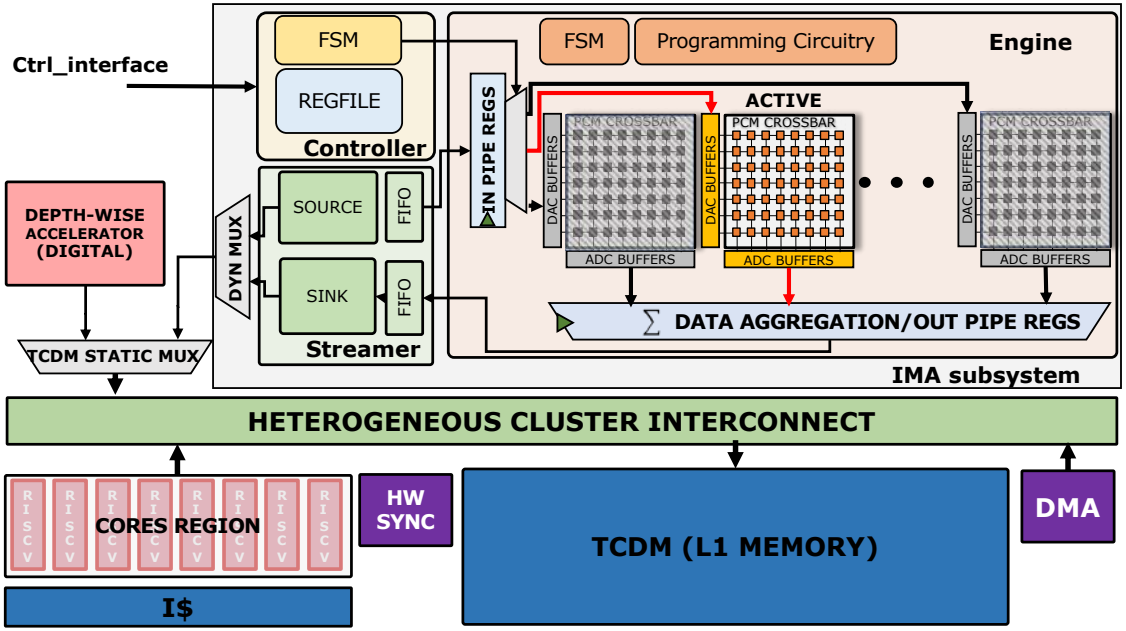
Submitter	Board Name	SoC Name	Processor(s) & Number	Accelerator(s) & Number	Software	Notes	Benchmark Results							
							Task	Visual Wake Words		Image Classification		Keyword Spotting		Anomaly Deteciton	
							Data	Visual Wake Words Dataset		CIFAR-10		Google Speech Commands		ToyADMOS (ToyCar)	
							Model	MobileNetV1 (0.25x)		ResNet-V1		DSCNN		FC AutoEncoder	
							Accuracy	80% (top 1)		85% (top 1)		90% (top 1)		0.85 (AUC)	
							Units	Latency in ms	Energy in uJ	Latency in ms	Energy in uJ	Latency in ms	Energy in uJ	Latency in ms	Energy in uJ
Greenwaves Technologies	GAP9 EVK	GAP9	RISC-V Core (1+9)	NE16 (1)	GreenWaves GAPFlow	GAP9 (370MHZ, 0.8Vcore)		1.13	58.4	0.62	40.4	0.48	26.7	0.18	7.29
Greenwaves Technologies	GAP9 EVK	GAP9	RISC-V Core (1+9)	NE16 (1)	GreenWaves GAPFlow	GAP9 (240MHZ, 0.65Vcore)		1.73	40.8	0.95	27.7	0.73	18.6	0.27	5.25
OctoML	NRF5340DK	nRF5340	Arm® Cortex®-M33		microTVM using CMSIS-NN backend	128MHz		232.0		316.1		76.1		6.27	
OctoML	NUCLEO-L4R5ZI	STM32L4R5ZIT6U	Arm® Cortex®-M4		microTVM using CMSIS-NN backend	120MHz, 1.8Vbat		301.2	15531.4	389.5	20236.3	99.8	5230.3	8.60	443.2
OctoML	NUCLEO-L4R5ZI	STM32L4R5ZIT6U	Arm® Cortex®-M4		microTVM using native codegen	120MHz, 1.8Vbat		336.5	17131.6	389.2	21342.8	144.0	7950.5	11.7	633.7
Plumerai	B_U585I_IOT02A	STM32U585	Arm® Cortex®-M33												
Plumerai	CY8CPROTO-062-4343w	PSoC 62 MCU	Arm® C M4												
Plumerai	DISCO-F746NG	STM32F746	Arm® C M7												
Plumerai	NUCLEO-L4R5ZI	STM32L4R5ZIT6U	Arm® C M4												
Silicon Labs	xG24-DK2601B	EFR32MG24	Arm® Cortex®-M33	Silicon Labs MVP(1)	TensorFlowLite for Microcontrollers, CMSIS-NN, Silicon Labs Gecko SDK			111.6	1139.2	120.9	1234.7	36.3	401.9	5.43	47.3
STMicroelectronics	NUCLEO-H7A3ZI-Q	STM32H7A3ZIT6Q	Arm® Cortex®-M7		X-CUBE-AI v7.3.0	280MHz, 3.3Vbat		50.7	7978.5	54.3	8707.3	16.8	2721.8	1.82	266.5
STMicroelectronics	NUCLEO-L4R5ZI	STM32L4R5ZIT6U	Arm® Cortex®-M4		X-CUBE-AI v7.3.0	120MHz, 1.8Vbat		230.5	10066.6	226.9	10681.6	75.1	3371.7	7.57	323.0
STMicroelectronics	NUCLEO-U575ZI-Q	STM32U575ZIT6Q	Arm® Cortex®-M33		X-CUBE-AI v7.3.0	160MHz, 1.8Vbat		133.4	3364.5	139.7	3642.0	44.2	1138.5	4.84	119.1
Syntiant	NDP9120-EVL	NDP120	M0 + HiFi	Syntiant Core 2 (98MH	Syntiant TDK	Syntiant Core 2 (98MHz, 1		4.10	97.2	5.12	139.4	1.48	43.8		
Syntiant	NDP9120-EVL	NDP120	M0 + HiFi	Syntiant Core 2 (30MH	Syntiant TDK	Syntiant Core 2 (30MHz, 0		12.7	71.7	16.0	101.8	4.37	31.5		
Qualcomm Innovation Center	Next Generation Snapdragon Mobile Platform HDK	Next Generation Snapdragon Mobile Platform	Qualcomm Kryo CPU(1)	Qualcomm Sensing Hub(1)	Qualcomm AI Stack									0.098	

Working at best E.E. operating point

MobileNetV1 inference in 1.73ms at 41uJ

- Best-in-class in **latency** and **energy efficiency** in MLPerf Tiny 1.0
  - ..by two-to-three orders of magnitude

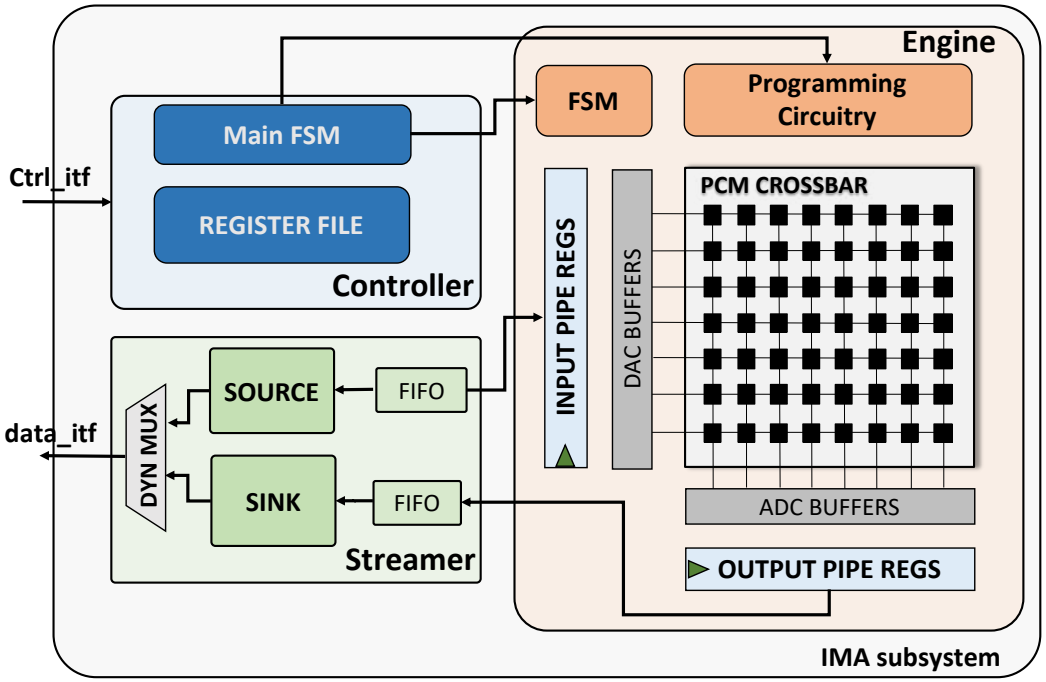
Heterogeneous In-Memory Computing Cluster



[Garofalo et al. IEEE JETCAS, 2022]

- 34 AIMC Xbars, 25mm^2 layout in 22nm FDX technology
- Enough storage capacity to host a full MobileNetV2 (~1MB) execution
- Sequential execution model tailored for the needs of an advanced IoT device

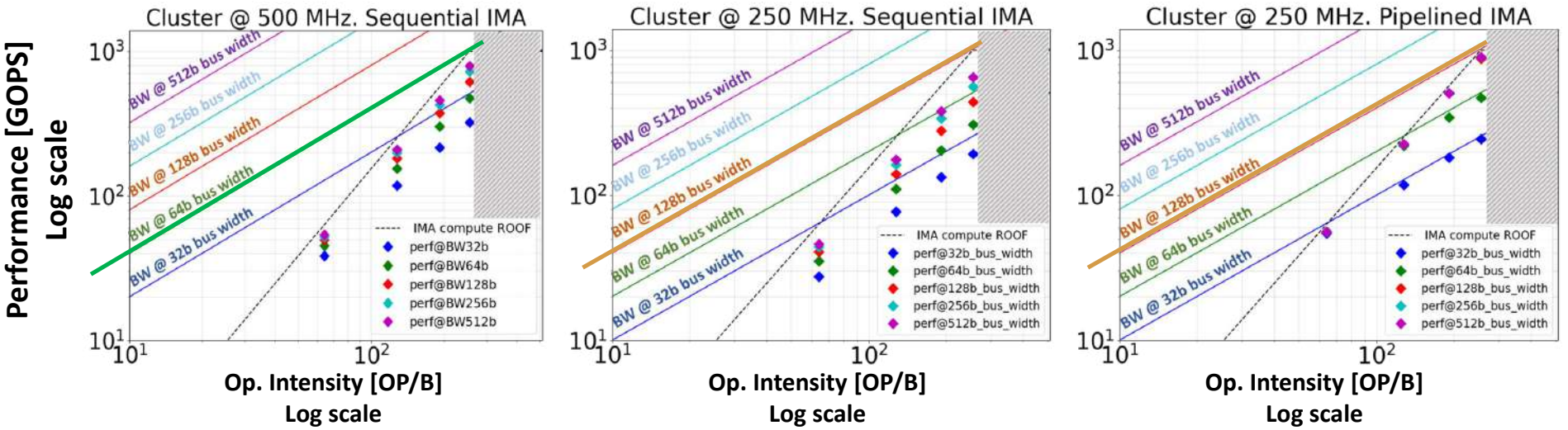
Heterogeneous In-Memory Computing Cluster



[Ottavi et al. AICAS, 2021]
[Khaddam et al. JSSCC, 2022]

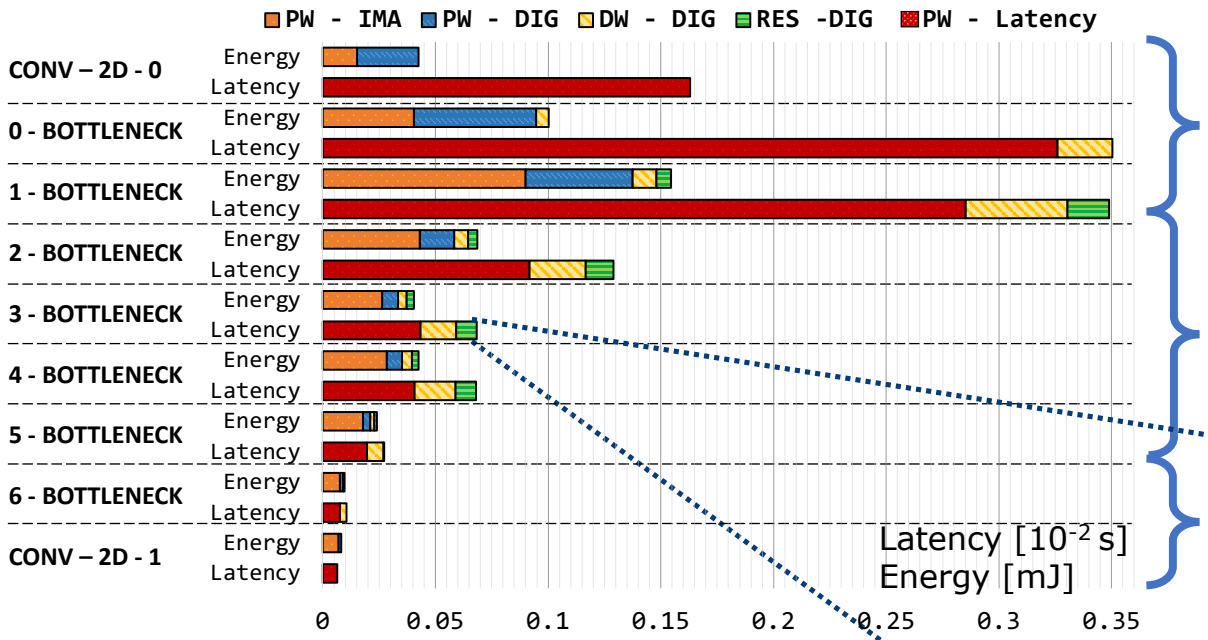
- **256x256** PCM-based IMC array
- Weights precision: **4-bit**
- Input/Output precision: **8-bit**
- MVM operations: 130K in 130ns
  - Peak performance **1TOPS**
- Peak Efficiency **11.7 TOPS/W**
- Integrated into HWPE
- Data_itf sized to sustain BW requirement of the IMC core

Roof-Line of an IMC Heterogeneous Cluster

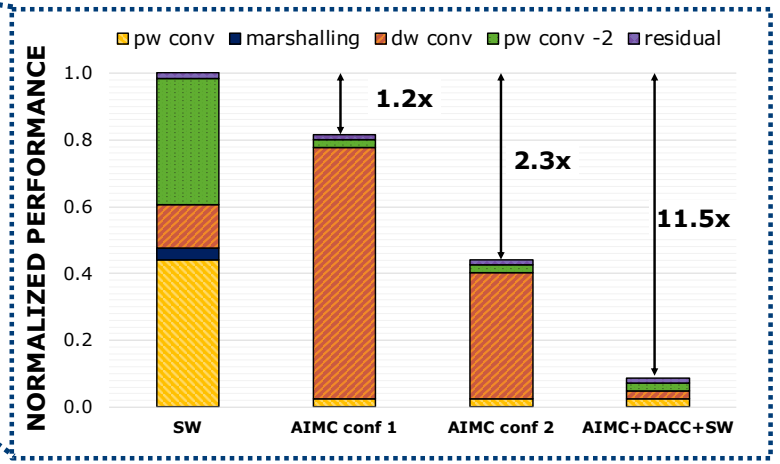


- Compute roof determined by MVM op. latency, XBAR size and array utilization
- BW determined by architecture and physical implementation of digital system
- Pipelined execution (overlap computation with streaming) to fully utilize BW

End-to-End MobileNetV2 Execution



- Energy dominated by data-transfers
- Significant non-MVM workloads executed on digital accelerators

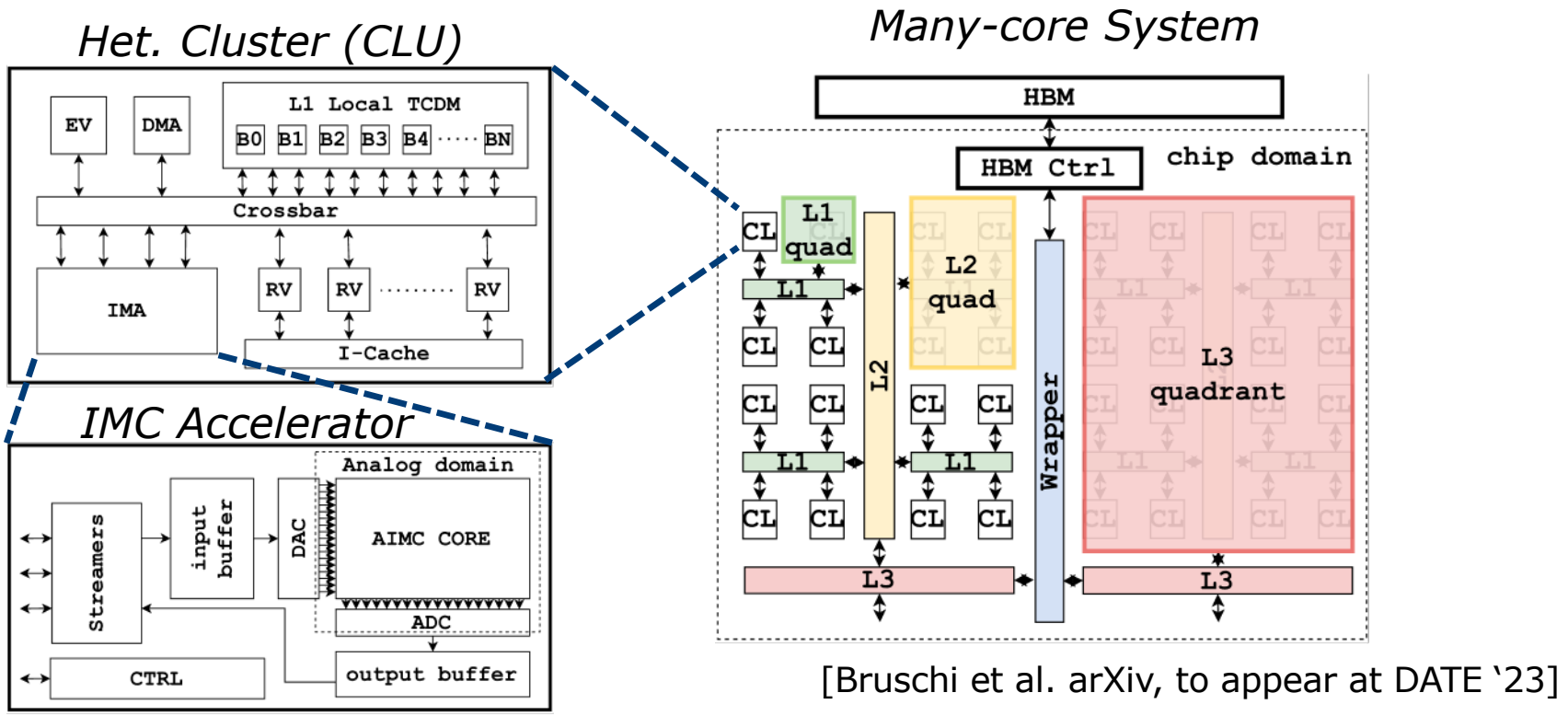


- Peak Performance bounded to ~1TOPS due to sequential execution model
  - How do we scale-up? Towards many-core IMC system..

Outline

- The quest for energy-efficient AI at the Edge of the IoT
  - The performance-power proportionality goal
  - Dealing with the end of the Dennard's scaling
- Low-power heterogeneous computing: mitigating deep acceleration effects
  - Multi-core heterogeneous systems
  - Tightly-coupled digital acceleration
  - Heterogeneous analog in-memory computing (AIMC) systems
- Insights for scaled-up heterogeneous powerful architectures
  - AIMC many-core architecture
  - Asymmetric chiplet-based systems
- Conclusions & Outlook

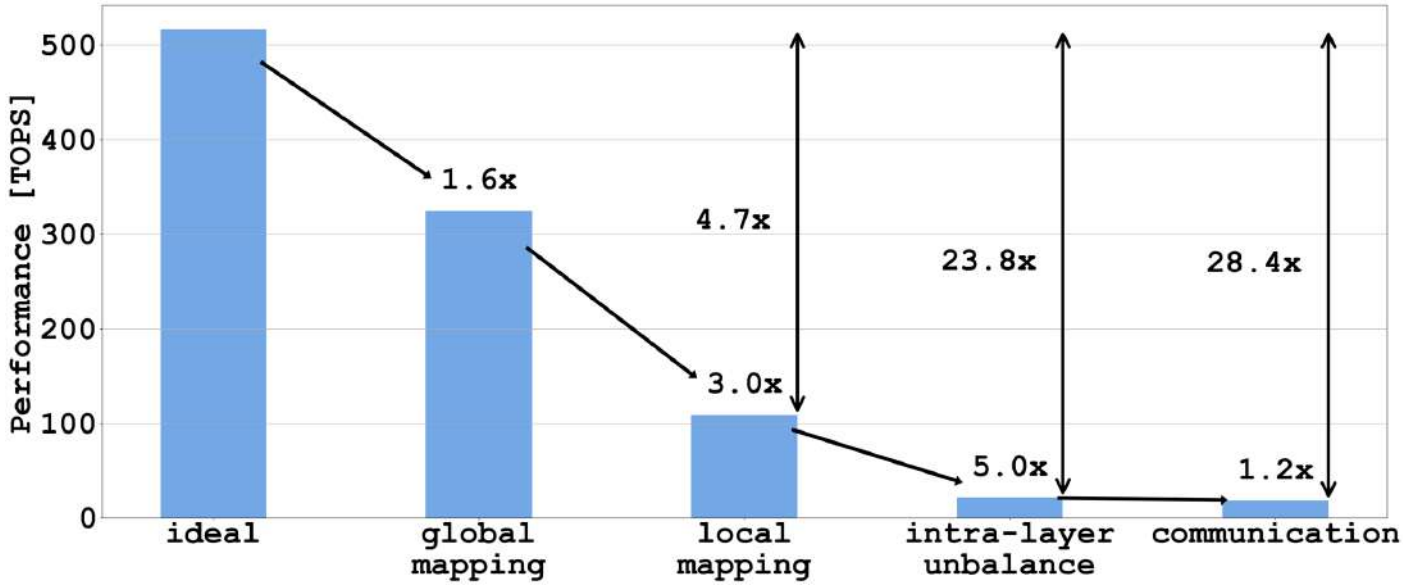
Explorative Scaled-up AIMC Many-Core System



[Bruschi et al. arXiv, to appear at DATE '23]

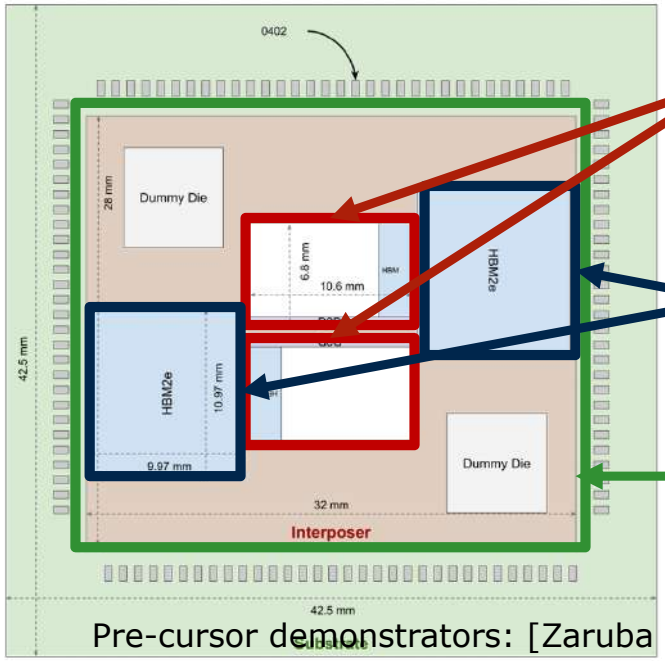
- 512 cluster, Low-Latency Hierarchical NoC;
- Not anymore in the IoT Domain (multi-core AIMC parallelism, batching exec.)

Exploring System-Level Inefficiencies (ResNet-18)



- HW-aware design space exploration framework to balance resources
  - More Heterogeneity → Analog Clusters, Purely Digital, Purely SW, Mem..
- Custom execution scheduling to increase HW resource utilization
- On-Chip communication seems not a big issue..
  - ..but strong assumptions made on chip area: 480mm^2 → **Chiplets!**

Leveraging Chiplet Technology: Occamy example

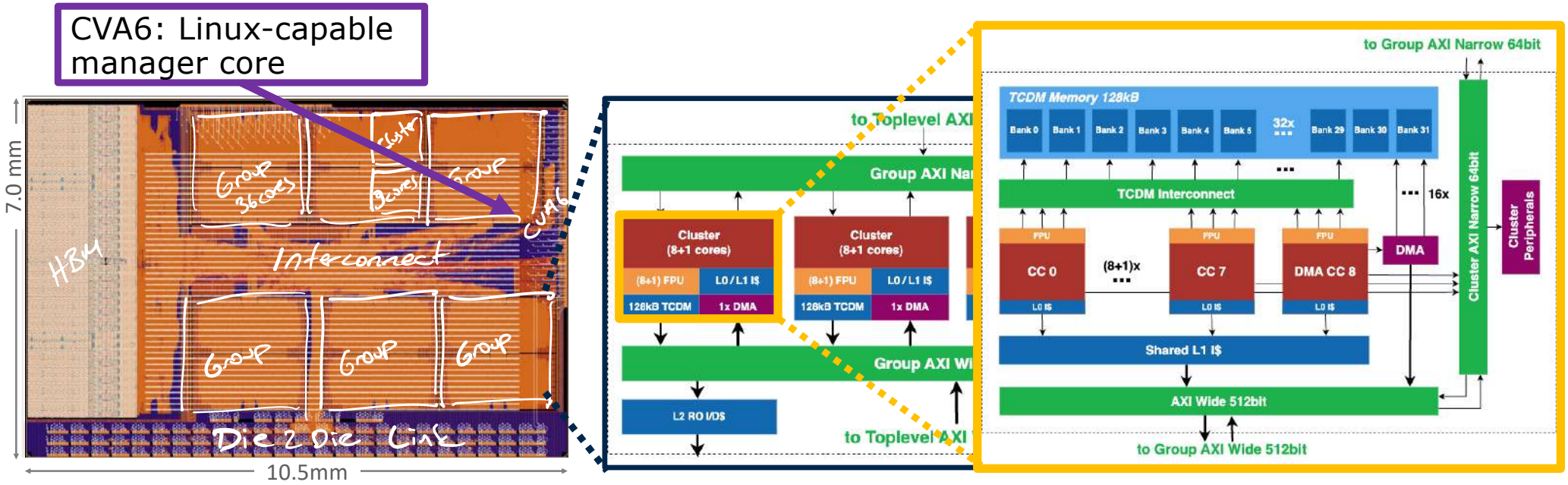


- **Dual chiplet system "Occamy"**
  - Tech: GF12LP+
  - Chiplet area 73mm^2
- **HBM2e**
  - 16 GB
- **Interposer "Hedwige"**
  - Tech: GF 65nm, passive

Pre-cursor demonstrators: [Zaruba et al., IEEE TC, 2020] [Zaruba et al., IEEE Micro, 2020]

- Mitigate slow-down of Moore's Law and end of Dennard's Scaling
- Enable heterogeneity at chiplet level (also different technology nodes)
- Unified chiplet interfaces required to ease integration & packaging

Occamy Chiplet



- 216 "snitch" cores/chiplet, **384** GDFLOPS/chiplet @ 1 GHz
- On-chip memory: 8-ch 16 GB HBM2e @ 512 GB/s, 2MB + 512kB per cluster SPM
- Hierarchical on-chip interconnect based on AXI4 protocol
- D2D link: 304 Gb/s @ 250 MHz (duplex)

Conclusions & Outlook

- Efficient data movements and mitigation of Amdahl's effects are key challenges to address to scale up AI efficiency on low-power but also high-performance systems

- Current trend towards multi-core heterogeneous AI systems
  - Many heterogeneous accelerators tightly-coupled to SW cores
  - Fast and efficient on-chip communication
  - Event-driven execution modes for on-demand resource scheduling

- Challenges and Outlook
  - System-level integration of multi-core AIMC processors
  - Heterogeneous hardware design space exploration
  - Leveraging chiplet technology to deploy massive heterogeneity in a single system
  - Optimized hardware-aware AI schedulers & compilers to improve data reuse (i.e. reduce data movements), HW utilization, overall system compute efficiency

References

1. Bianco, Simone, Remi Cadene, Luigi Celona, and Paolo Napoletano. "Benchmark analysis of representative deep neural network architectures." IEEE Access 6 (2018): 64270-64277.
2. Rusci, M., Capotondi, A., & Benini, L. (2020). Memory-driven mixed low precision quantization for enabling deep network inference on microcontrollers. *Proceedings of Machine Learning and Systems*, *2*, 326-335.
3. Ottavi, G., Garofalo, A., Tagliavini, G., Conti, F., Benini, L., & Rossi, D. (2020, July). A mixed-precision RISC-V processor for extreme-edge DNN inference. In 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI) (pp. 512-517). IEEE.
4. Garofalo, A., Ottavi, G., Di Mauro, A., Conti, F., Tagliavini, G., Benini, L., & Rossi, D. (2021, September). A 1.15 TOPS/W, 16-Cores Parallel Ultra-Low Power Cluster with 2b-to-32b Fully Flexible Bit-Precision and Vector Lockstep Execution Mode. In *ESSCIRC 2021-IEEE 47th European Solid State Circuits Conference (ESSCIRC)* (pp. 267-270). IEEE.
5. Garofalo, A., Ottavi, G., Conti, F., Karunaratne, G., Boybat, I., Benini, L., & Rossi, D. (2022). A Heterogeneous In-Memory Computing Cluster For Flexible End-to-End Inference of Real-World Deep Neural Networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*.
6. Bruschi, N., Tagliavini, G., Garofalo, A., Conti, F., Boybat, I., Benini, L., & Rossi, D. (2022). End-to-End DNN Inference on a Massively Parallel Analog In Memory Computing Architecture. *arXiv preprint arXiv:2211.12877*.
7. Garofalo, A., Tortorella, Y., Perotti, M., Valente, L., Nadalini, A., Benini, L., ... & Conti, F. (2022). Darkside: A Heterogeneous RISC-V Compute Cluster for Extreme-Edge On-Chip DNN Inference and Training. *IEEE Open Journal of the Solid-State Circuits Society*.
8. Gautschi, M., Schiavone, P. D., Traber, A., Loi, I., Pullini, A., Rossi, D., ... & Benini, L. (2017). Near-threshold RISC-V core with DSP extensions for scalable IoT endpoint devices. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *25*(10), 2700-2713.
9. Ottavi, G., Karunaratne, G., Conti, F., Boybat, I., Benini, L., & Rossi, D. (2021, June). End-to-end 100-TOPS/W Inference With Analog In-Memory Computing: Are We There Yet?. In *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (pp. 1-4). IEEE.

References - 2

10. Glaser, F., Tagliavini, G., Rossi, D., Haugou, G., Huang, Q., & Benini, L. (2020). Energy-efficient hardware-accelerated synchronization for shared-L1-memory multiprocessor clusters. IEEE Transactions on Parallel and Distributed Systems, 32(3), 633-648.
11. Zaruba, F., Schuiki, F., Hoefler, T., & Benini, L. (2020). Snitch: A tiny pseudo dual-issue processor for area and energy efficient execution of floating-point intensive workloads. IEEE Transactions on Computers, 70(11), 1845-1860.
12. Zaruba, F., Schuiki, F., & Benini, L. (2020). Manticore: A 4096-core RISC-V chiplet architecture for ultraefficient floating-point computing. IEEE Micro, 41(2), 36-42.
13. Rossi, D., Conti, F., Eggiman, M., Di Mauro, A., Tagliavini, G., Mach, S., ... & Benini, L. (2021). Vega: A Ten-Core SoC for IoT Endnodes With DNN Acceleration and Cognitive Wake-Up From MRAM-Based State-Retentive Sleep Mode. IEEE Journal of Solid-State Circuits, 57(1), 127-139.
14. Khaddam-Aljameh, R., Stanisavljevic, M., Mas, J. F., Karunaratne, G., Brändli, M., Liu, F., ... & Eleftheriou, E. (2022). HERMES-core—a 1.59-TOPS/mm 2 PCM on 14-nm CMOS in-memory compute core using 300-ps/LSB linearized CCO-based ADCs. IEEE Journal of Solid-State Circuits, 57(4), 1027-1038.
15. Fick, L., Skrzyniarz, S., Parikh, M., Henry, M. B., & Fick, D. (2022, February). Analog Matrix Processor for Edge AI Real-Time Video Analytics. In 2022 IEEE International Solid-State Circuits Conference (ISSCC) (Vol. 65, pp. 260-262). IEEE.
16. Jia, H., Ozatay, M., Tang, Y., Valavi, H., Pathak, R., Lee, J., & Verma, N. (2021, February). 15.1 a programmable neural-network inference accelerator based on scalable in-memory computing. In 2021 IEEE International Solid-State Circuits Conference (ISSCC) (Vol. 64, pp. 236-238). IEEE.
17. Ueyoshi, K., Papistas, I. A., Houshmand, P., Sarda, G. M., Jain, V., Shi, M., ... & Verhelst, M. (2022, February). DIANA: An End-to-End Energy-Efficient Digital and ANAlog Hybrid Neural Network SoC. In 2022 IEEE International Solid-State Circuits Conference (ISSCC) (Vol. 65, pp. 1-3). IEEE.
18. Jia, H., Valavi, H., Tang, Y., Zhang, J., & Verma, N. (2020). A programmable heterogeneous microprocessor based on bit-scalable in-memory computing. IEEE Journal of Solid-State Circuits, 55(9), 2609-2621.

References - 3

19. Rossi, D., Pullini, A., Loi, I., Gautschi, M., Gürkaynak, F. K., Teman, A., ... & Benini, L. (2017). Energy-efficient near-threshold parallel computing: The PULPv2 cluster. Ieee Micro, 37(5), 20-31.
20. Moons, B., Uytterhoeven, R., Dehaene, W., & Verhelst, M. (2017, February). 14.5 envision: A 0.26-to-10tops/w subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm fdsoi. In 2017 IEEE International Solid-State Circuits Conference (ISSCC) (pp. 246-247). IEEE.
21. Chen, Y. H., Krishna, T., Emer, J. S., & Sze, V. (2016). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. IEEE journal of solid-state circuits, 52(1), 127-138.
22. Ottavi, G., Garofalo, A., Tagliavini, G., Conti, F., Di Mauro, A., Benini, L., & Rossi, D. (2022). Dustin: A 16-Cores Parallel Ultra-Low-Power Cluster with 2b-to-32b Fully Flexible Bit-Precision and Vector Lockstep Execution Mode. arXiv preprint arXiv:2201.08656.
23. Bruschi, N., Tagliavini, G., Garofalo, A., Conti, F., Boybat, I., Benini, L., & Rossi, D. (2022). End-to-End DNN Inference on a Massively Parallel Analog In Memory Computing Architecture. arXiv preprint arXiv:2211.12877. [To appear at DATE 2023].
24. Burrello, A., Garofalo, A., Bruschi, N., Tagliavini, G., Rossi, D., & Conti, F. (2021). Dory: Automatic end-to-end deployment of real-world dnns on low-cost iot mcus. IEEE Transactions on Computers, 70(8), 1253-1268.
25. Di Mauro, A., Scherer, M., Rossi, D., & Benini, L. (2022). Kraken: A direct event/frame-based multi-sensor fusion soc for ultra-efficient visual processing in nano-uavs. arXiv preprint arXiv:2209.01065.
26. Jain, V., Giraldo, S., De Roose, J., Boons, B., Mei, L., & Verhelst, M. (2022, June). TinyVers: A 0.8-17 TOPS/W, 1.7 µW-20 mW, Tiny Versatile System-on-chip with State-Retentive eMRAM for Machine Learning Inference at the Extreme Edge. In 2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits) (pp. 20-21). IEEE.

Is an AI Accelerator All You Need? Overcoming Amdahl's Law With Tightly-Coupled Specialized Accelerators

BACK-UP SLIDES

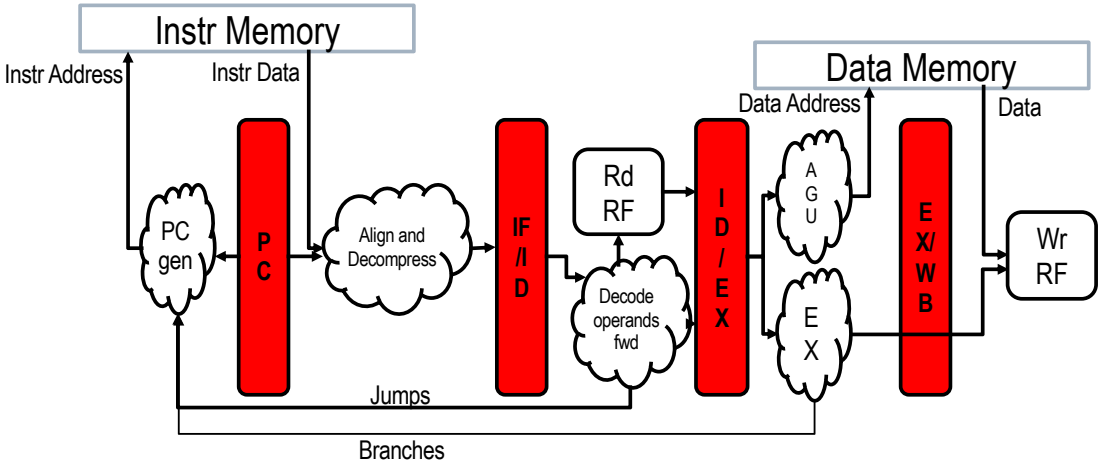
ISSCC 2023, FORUM 6

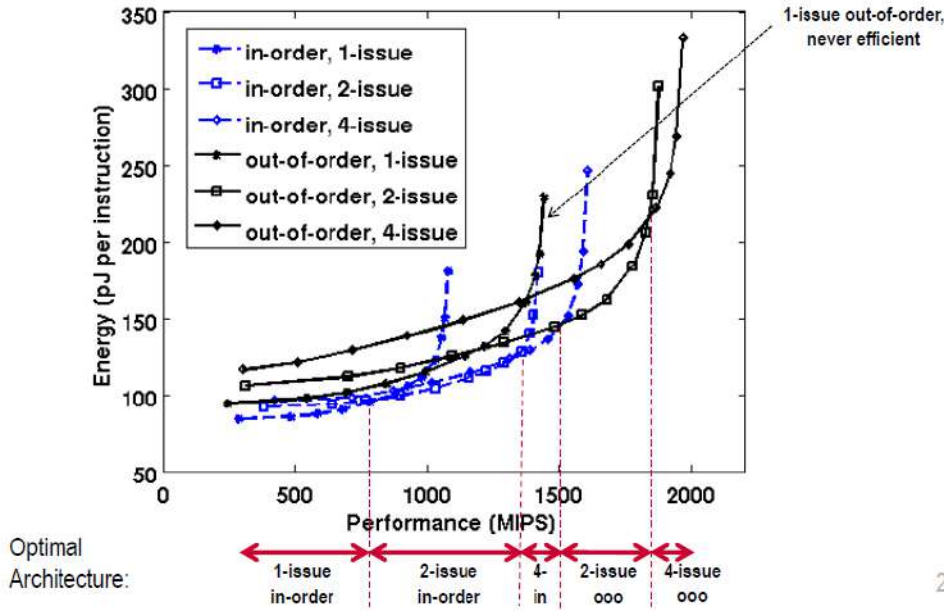
"The Future of Heterogeneous Multi-Core Architectures for AI and Other Specialized Processing"

From High-Performance to Low-Power

- Multi-Core Architectures
  - Speed-up general-purpose tasks
  - Domain-specific insns rather than super-scalar cores for higher en. efficiency
- Dedicated acceleration solutions to boost AI workloads
  - Processor specialization (low-bitwidth integer arithmetic)
  - Dedicated digital/analog data-paths
- Heterogeneous Integration to mitigate deep acceleration effects
  - CPUs + domain-specific GP cores + NN Engines
  - High-Bandwidth Low-Latency On-Chip Communication

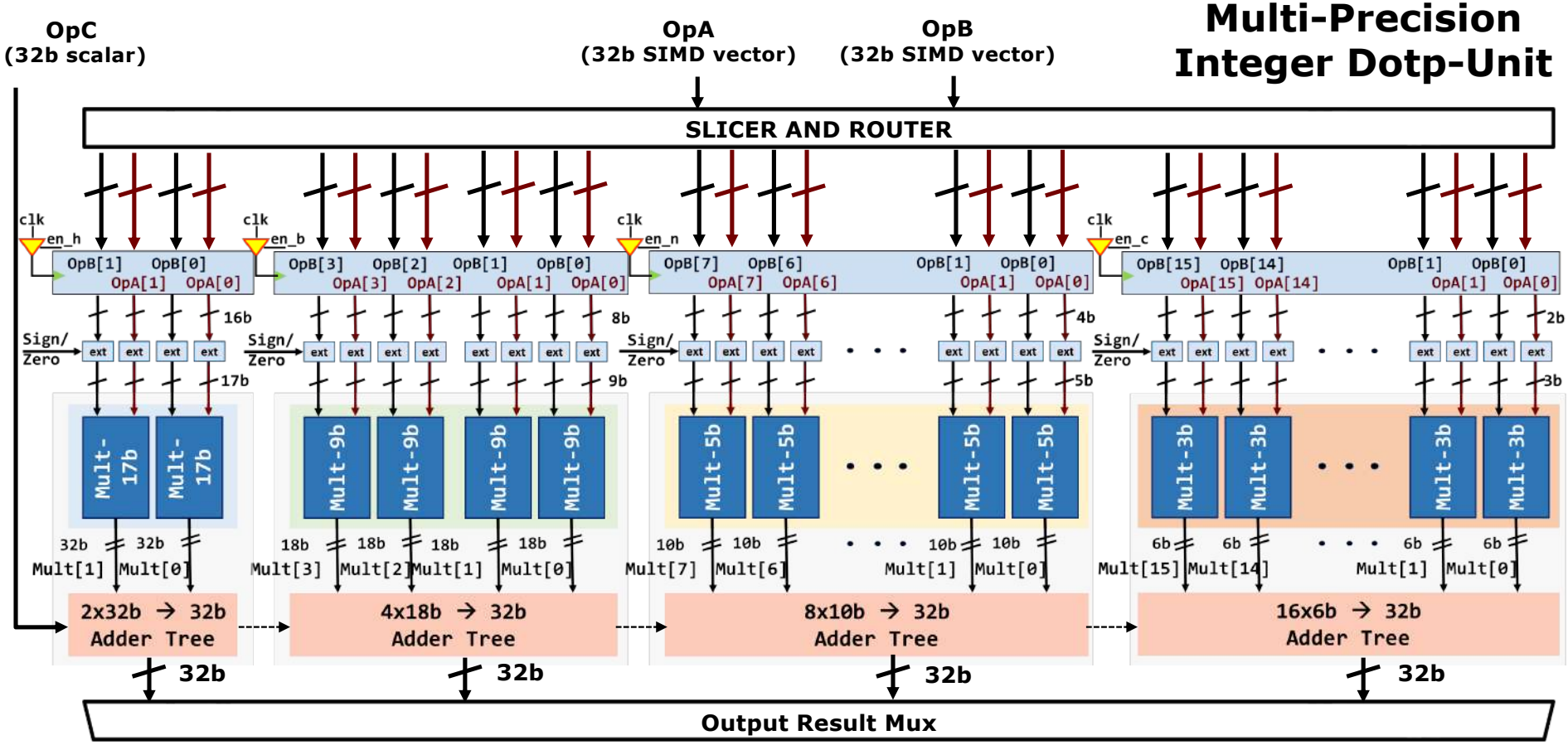
The Tunnel: High-Performance vs. High-Efficiency



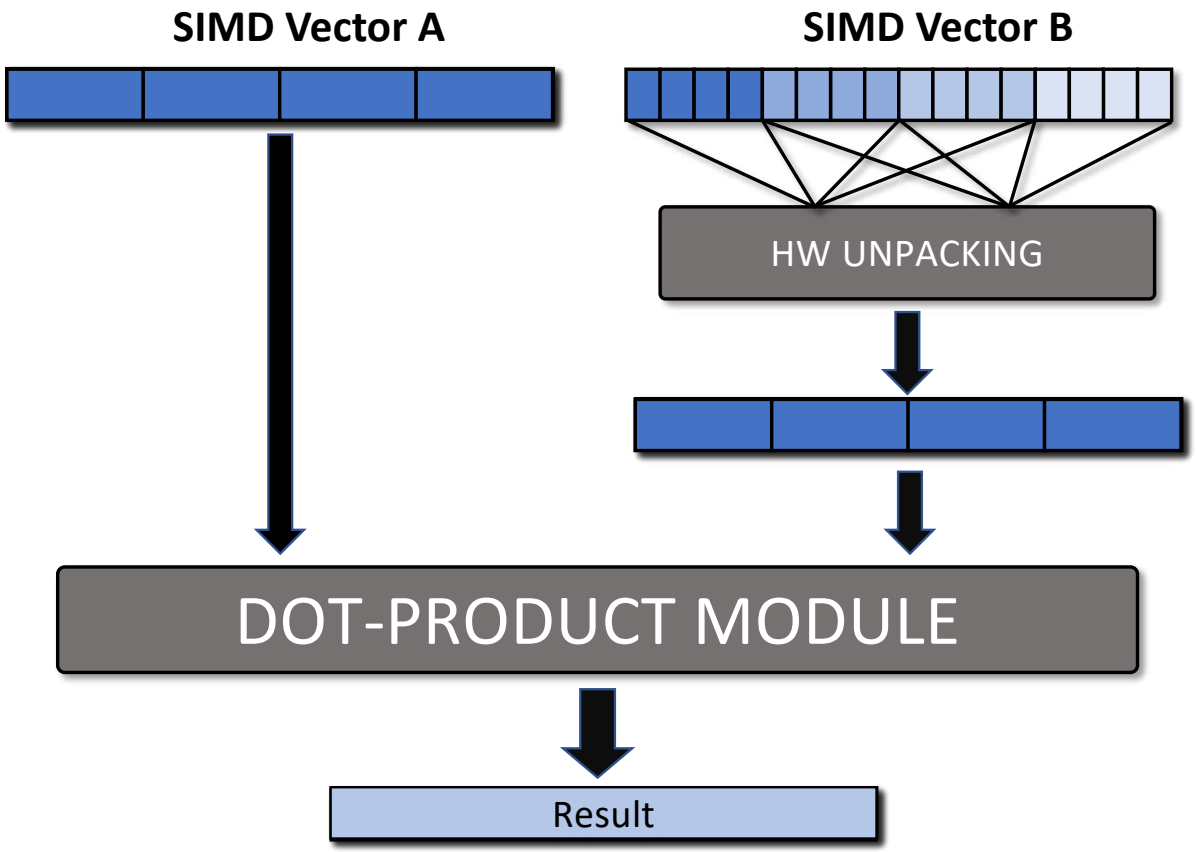


- **"Classical" core performance scaling trajectory** [AziziISCA10]
  - Faster CLK → deeper pipeline → IPC drops
  - Recover IPC → superscalar → ILP bottleneck (dependencies)
  - Mitigate ILP bottlenecks → OOO → huge power, area cost!

Extended Dot-Product Unit



Mixed-Precision Controller



- Mixed-Precision operations require a controller: selection of the correct sub-word of the lowest precision operand (Vector B) to be used in current SIMD op.
- The controller is programmed by control status registers (CSRs).

Dynamic Bit-Scalable Execution

Standard Instructions

pv.dotsp.h	pv.dotup.h	pv.dotusp.h	pv.sdotsp.h	pv.sdotup.h	pv.sdotusp.h
pv.dotsp.b	pv.dotup.b	pv.dotusp.b	pv.sdotsp.b	pv.sdotup.b	pv.sdotusp.b
pv.dotsp.n	pv.dotup.n	pv.dotusp.n	pv.sdotsp.n	pv.sdotup.n	pv.sdotusp.n
pv.dotsp.c	pv.dotup.c	pv.dotusp.c	pv.sdotsp.c	pv.sdotup.c	pv.sdotusp.c
pv.dotsp.m4x2	pv.dotup.m4x2	pv.dotusp.m4x2	pv.sdotsp.m4x2	pv.sdotup.m4x2	pv.sdotusp.m4x2
pv.dotsp.m8x2	pv.dotup.m8x2	pv.dotusp.m8x2	pv.sdotsp.m8x2	pv.sdotup.m8x2	pv.sdotusp.m8x2
pv.dotsp.m8x4	pv.dotup.m8x4	pv.dotusp.m8x4	pv.sdotsp.m8x4	pv.sdotup.m8x4	pv.sdotusp.m8x4
pv.dotsp.m16x8	pv.dotup.m16x8	pv.dotusp.m16x8	pv.sdotsp.m16x8	pv.sdotup.m16x8	pv.sdotusp.m16x8
pv.dotsp.m16x4	pv.dotup.m16x4	pv.dotusp.m16x4	pv.sdotsp.m16x4	pv.sdotup.m16x4	pv.sdotusp.m16x4
pv.dotsp.m16x2	pv.dotup.m16x2	pv.dotusp.m16x2	pv.sdotsp.m16x2	pv.sdotup.m16x2	pv.sdotusp.m16x2
pv.dotsp.sc.h	pv.dotup.sc.h	pv.dotusp.sc.h	pv.sdotsp.sc.h	pv.sdotup.sc.h	pv.sdotusp.sc.h
pv.dotsp.sc.b	pv.dotup.sc.b	pv.dotusp.sc.b	pv.sdotsp.sc.b	pv.sdotup.sc.b	pv.sdotusp.sc.b
pv.dotsp.sc.c	pv.dotup.sc.c	pv.dotusp.sc.c	pv.sdotsp.sc.c	pv.sdotup.sc.c	pv.sdotusp.sc.c
pv.dotsp.sc.n	pv.dotup.sc.n	pv.dotusp.sc.n	pv.sdotsp.sc.n	pv.sdotup.sc.n	pv.sdotusp.sc.n
pv.dotsp.sc.m4x2	pv.dotup.sc.m4x2	pv.dotusp.sc.m4x2	pv.sdotsp.sc.m4x2	pv.sdotup.sc.m4x2	pv.sdotusp.sc.m4x2
pv.dotsp.sc.m8x2	pv.dotup.sc.m8x2	pv.dotusp.sc.m8x2	pv.sdotsp.sc.m8x2	pv.sdotup.sc.m8x2	pv.sdotusp.sc.m8x2
pv.dotsp.sc.m8x4	pv.dotup.sc.m8x4	pv.dotusp.sc.m8x4	pv.sdotsp.sc.m8x4	pv.sdotup.sc.m8x4	pv.sdotusp.sc.m8x4
pv.dotsp.sc.m16x8	pv.dotup.sc.m16x8	pv.dotusp.sc.m16x8	pv.sdotsp.sc.m16x8	pv.sdotup.sc.m16x8	pv.sdotusp.sc.m16x8
pv.dotsp.sc.m16x4	pv.dotup.sc.m16x4	pv.dotusp.sc.m16x4	pv.sdotsp.sc.m16x4	pv.sdotup.sc.m16x4	pv.sdotusp.sc.m16x4
pv.dotsp.sc.m16x2	pv.dotup.sc.m16x2	pv.dotusp.sc.m16x2	pv.sdotsp.sc.m16x2	pv.sdotup.sc.m16x2	pv.sdotusp.sc.m16x2
pv.dotsp.sci.h	pv.dotup.sci.h	pv.dotusp.sci.h	pv.sdotsp.sci.h	pv.sdotup.sci.h	pv.sdotusp.sci.h
pv.dotsp.sci.b	pv.dotup.sci.b	pv.dotusp.sci.b	pv.sdotsp.sci.b	pv.sdotup.sci.b	pv.sdotusp.sci.b
pv.dotsp.sci.c	pv.dotup.sci.c	pv.dotusp.sci.c	pv.sdotsp.sci.c	pv.sdotup.sci.c	pv.sdotusp.sci.c
pv.dotsp.sci.n	pv.dotup.sci.n	pv.dotusp.sci.n	pv.sdotsp.sci.n	pv.sdotup.sci.n	pv.sdotusp.sci.n
pv.dotsp.sci.m4x2	pv.dotup.sci.m4x2	pv.dotusp.sci.m4x2	pv.sdotsp.sci.m4x2	pv.sdotup.sci.m4x2	pv.sdotusp.sci.m4x2
pv.dotsp.sci.m8x2	pv.dotup.sci.m8x2	pv.dotusp.sci.m8x2	pv.sdotsp.sci.m8x2	pv.sdotup.sci.m8x2	pv.sdotusp.sci.m8x2
pv.dotsp.sci.m8x4	pv.dotup.sci.m8x4	pv.dotusp.sci.m8x4	pv.sdotsp.sci.m8x4	pv.sdotup.sci.m8x4	pv.sdotusp.sci.m8x4
pv.dotsp.sci.m16x8	pv.dotup.sci.m16x8	pv.dotusp.sci.m16x8	pv.sdotsp.sci.m16x8	pv.sdotup.sci.m16x8	pv.sdotusp.sci.m16x8
pv.dotsp.sci.m16x4	pv.dotup.sci.m16x4	pv.dotusp.sci.m16x4	pv.sdotsp.sci.m16x4	pv.sdotup.sci.m16x4	pv.sdotusp.sci.m16x4
pv.dotsp.sci.m16x2	pv.dotup.sci.m16x2	pv.dotusp.sci.m16x2	pv.sdotsp.sci.m16x2	pv.sdotup.sci.m16x2	pv.sdotusp.sci.m16x2

Virtual Instructions

pv.dotsp.v	pv.sdotsp .v
pv.dotsp.sc .v	pv.sdotsp.sc .v
pv.dotsp.sci .v	pv.sdotsp.sci .v
pv.dotup .v	pv.sdotup .v
pv.dotup.sc .v	pv.sdotup.sc .v
pv.dotup.sci .v	pv.sdotup.sci .v
pv.dotusp .v	pv.sdotusp .v
pv.dotusp.sc .v	pv.sdotusp.sc .v
pv.dotusp.sci .v	pv.sdotusp.sci .v

- Virtual I
- No e
- Reus

ic Bit-
le Execution

10,4(x4!)
11,4(x5!)
p.v x20,x11,x10

ne
instruction
per type

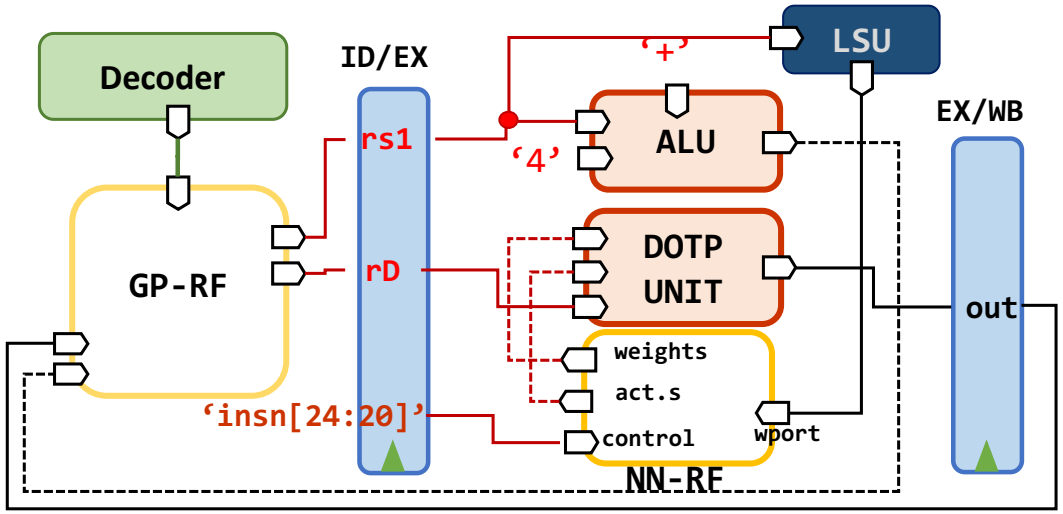
ode

```
int main()
{
....
SIMD_FMT(M8x4);
convolution(A, W, Res);
....
SIMD_FMT(M8x2);
convolution(A, W, Res);
....}
```

```
pv.packlo.b x15, x5, x6
pv.packhi.b x15, x7, x8
pv.sdotsp.b x20, x15, x10
```

instruction
per format
and type

Mac-Load Instruction



- *MatMul's operands* memory access patern extremely regular
  - And known before execution
- Fuse MAC with LOAD operations in one single-cycle insns
  - "Fused" data-path, contemporary cosume current operand & prefetch next-one
  - Reduce overhead of Loads into *MatMul* innermost loops

Mac-Load: *MatMul* Innermost Kernel

8-bit innermost loop of the 4x2 *Matmul kernel* of the PULP-NN library

```
      lp.setup      l1, l2,end
      p.lw          w1, 4(aw1!)
      p.lw          w2, 4(aw2!)
      p.lw          w3, 4(aw3!)
      p.lw          w4, 4(aw4!)
      p.lw          x1, 4(ax1!)
      p.lw          x2, 4(ax2!)
      pv.sdotusp.b  s1, x1, w1
      pv.sdotusp.b  s2, x1, w2
      pv.sdotusp.b  s3, x1, w3
      pv.sdotusp.b  s4, x1, w4
      pv.sdotusp.b  s5, x2, w1
      pv.sdotusp.b  s6, x2, w2
      pv.sdotusp.b  s7, x2, w3
end:  pv.sdotusp.b  s8, x2, w4
```

HW LOOP

LD/ST WITH POST INCREMENT

8-B SIMD MAC

8 SIMD MACs with 6 explicit LOADs

8-bit innermost loop of the 4x2 *MatMul kernel* of the extended PULP-NN library with nnsdotp

```
INIT
NN-RF pv.nnsdotusp.h  zero, aw1,16
      pv.nnsdotusp.h  zero, aw2,18
      pv.nnsdotusp.h  zero, aw3,20
      pv.nnsdotusp.h  zero, aw4,22
      pv.nnsdotusp.h  zero, ax1,8
      lp.setup        l1, l2, end
      pv.nnsdotup.h   zero,ax2,9
      pv.nnsdotusp.b  s1, aw2, 0
      pv.nnsdotusp.b  s2, aw4, 2
      pv.nnsdotusp.b  s3, aw3, 4
      pv.nnsdotusp.b  s4, ax1, 14
      pv.nnsdotusp.b  s5, aw2, 17
      pv.nnsdotusp.b  s6, aw4, 19
      pv.nnsdotusp.b  s7, aw3, 21
end:  pv.nnsdotusp.b  s8, aw1, 23
```

8 SIMD MACs with 1 explicit LOAD

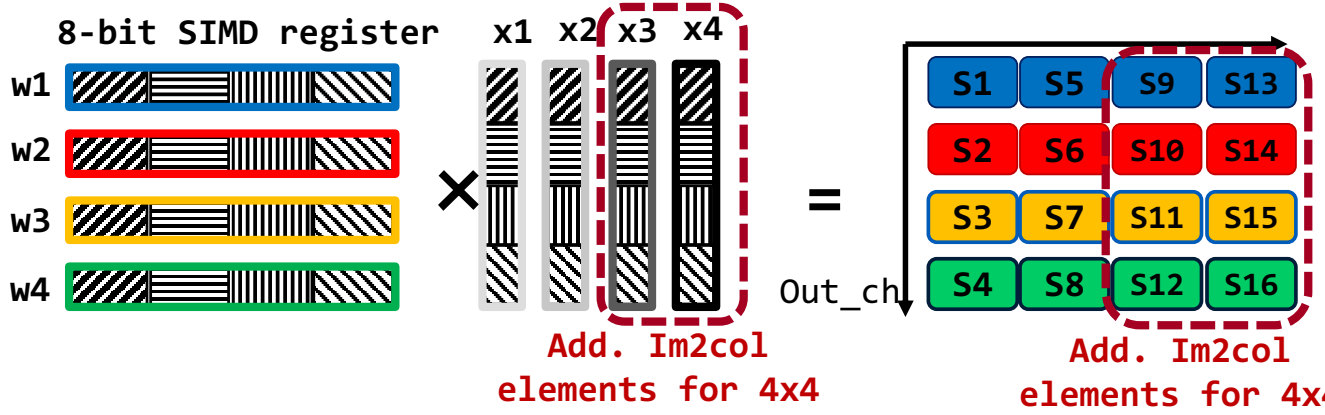
W_i X_i : weight/activation elements

AW_i AX_i : addresses for the MEM access

S_i : accumulators

L_i : loop setup

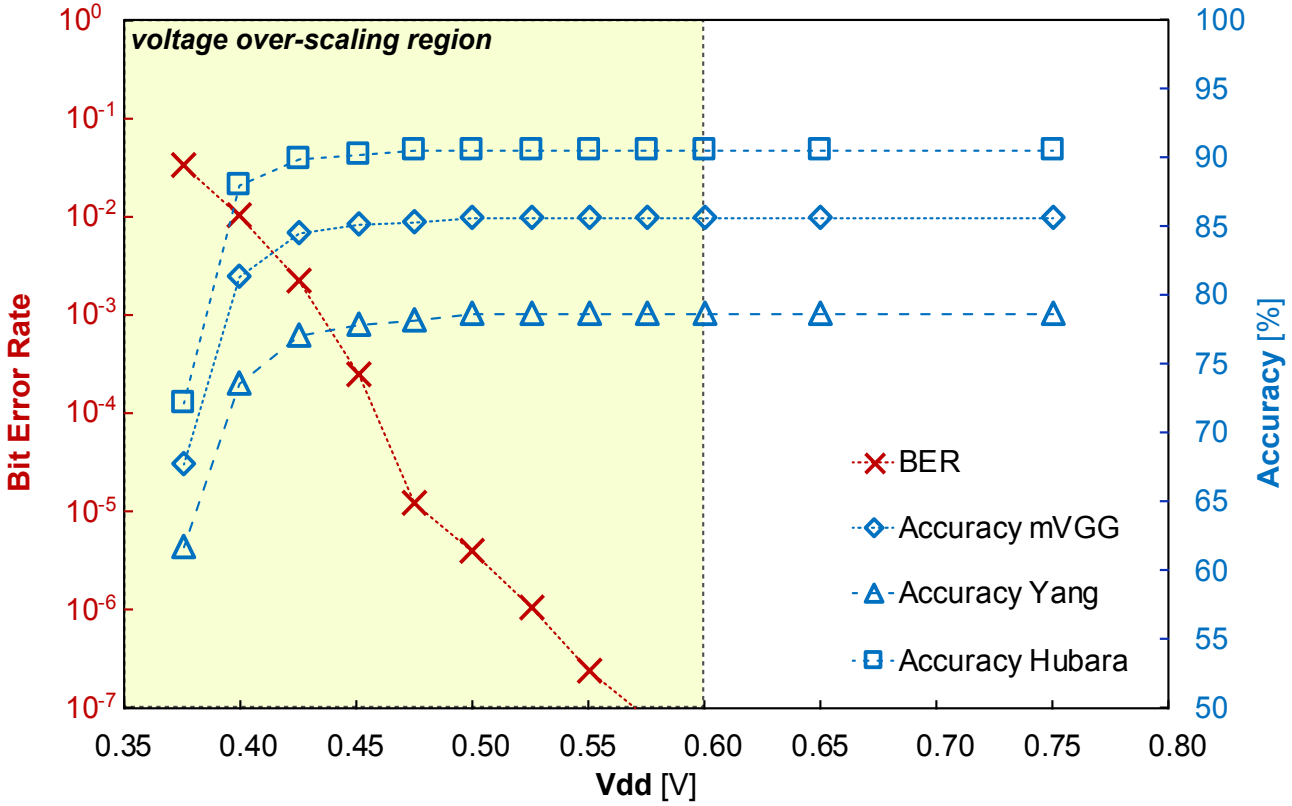
Mac-Load: *MatMul* Data-Reuse



Register File Resources required	4x2 kernel		4x4 kernel	
	SIMD	SIMD+nnsdotp	SIMD	SIMD+nnsdotp
Loop Setup	2	2	2	2
Addresses for MEM access	6	6	8	8
MAC operands	6	0 (op. from NN-RF)	8	0 (op. from NN-RF)
MAC accumulators	8	8	16	16
Total	22 regs	16 regs	34 regs	**26 regs**

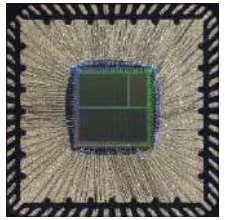
Up to 94% of SIMD Dotp Unit Utilization on MatMul kernels

Working with Unreliability: **Quentin** (**ULP**)

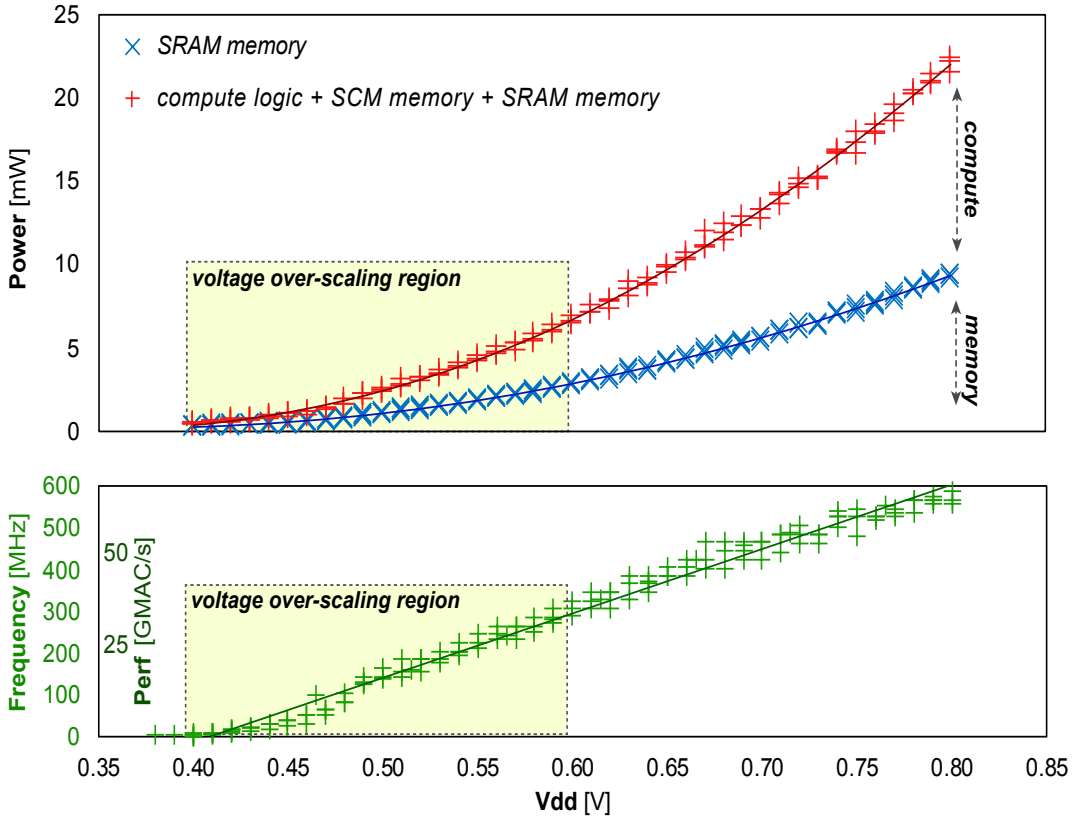


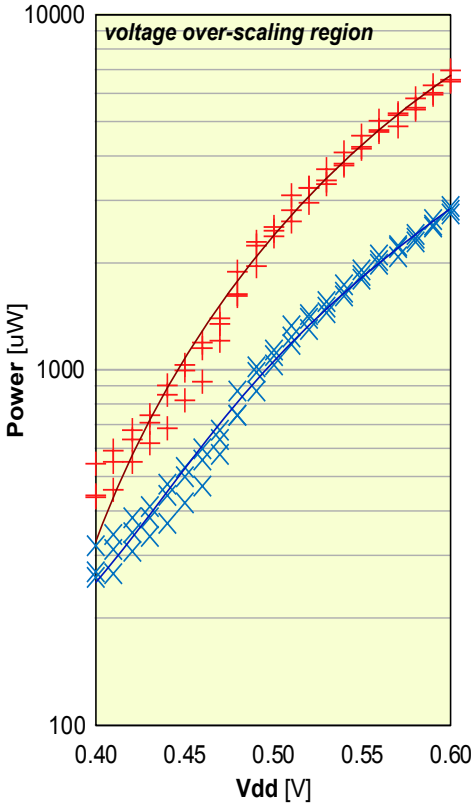
Di Mauro et al. [IEDM19]

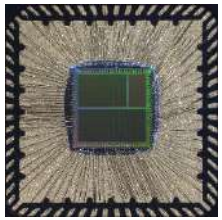
GF 22FDX SoC with embedded hardware accelerator for BNNs (**XNE**)



Working with Unreliability: **Quentin** (**ULP**)



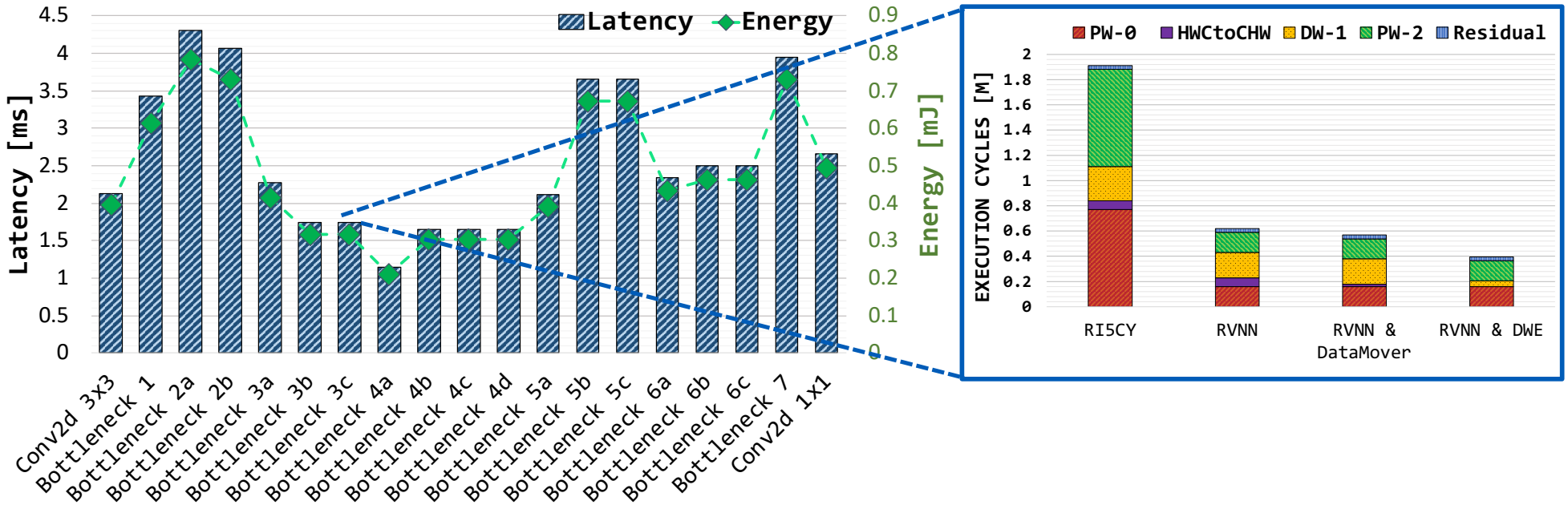




Perf Tuning:
22mW@140Gop/s
635µW@20Gop/s

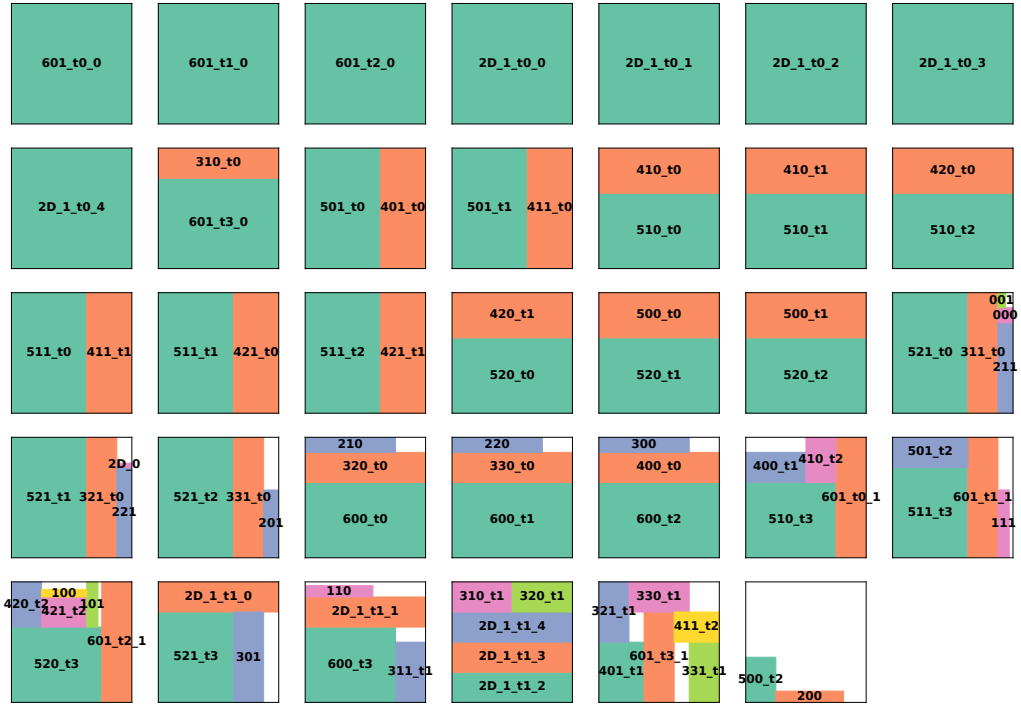
Di Mauro et al. [IEDM19]

End-to-End MobileNetV2 on Darkside



- Heterogeneous acceleration mitigates Amdahl's effects on *Bottlenecks*
- Still room to accelerate PW convolutions

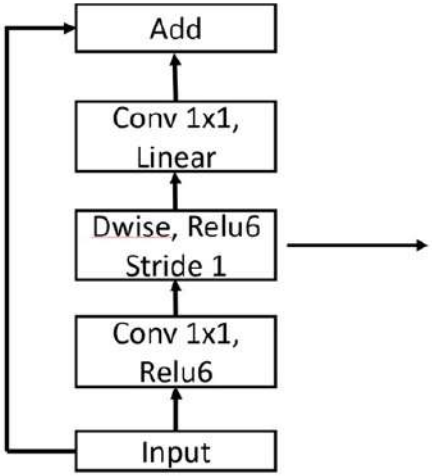
MobileNetV2 mapping on AIMC

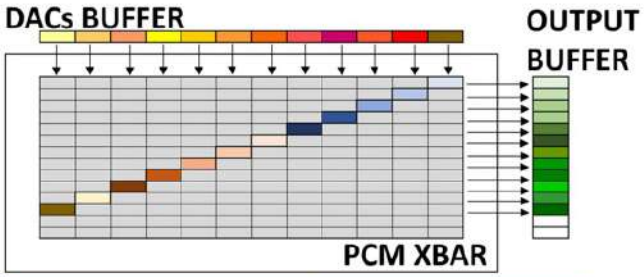


Only PointWise layers mapped on AIMC

- Tile&Pack algorithm to deploy MBNTv2 on the smallest number of AIMC cores
- High storage utilization, poor bit-cell compute utilization on small layers

Depth-Wise Execution on AIMC





Layer	Size	IN-CH	OUT-CH
Conv 1x1	28x28	32	192
Dwise	28x28	192	192
Conv 1x1	28x28	192	32

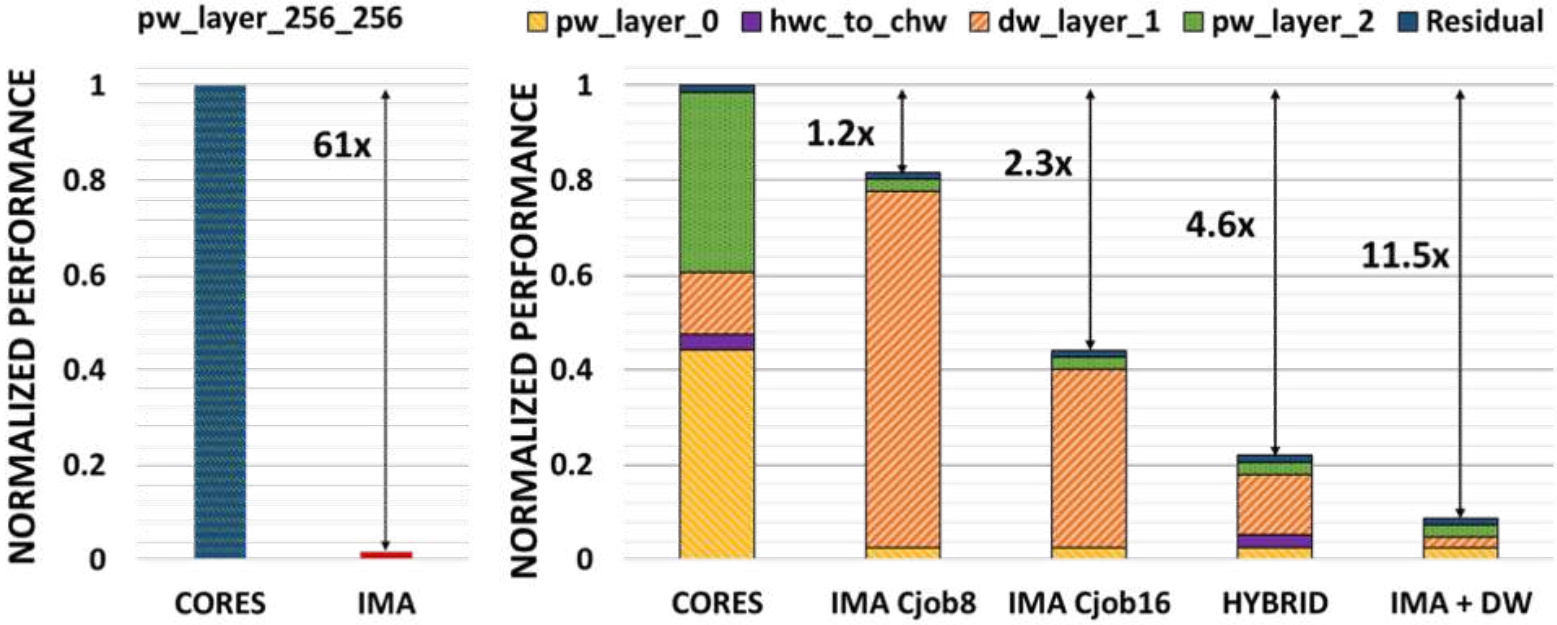
- Representative of modern CNN
- Bottlenecks are building block of MobileNet
- DW are used to reduce number of parameters in CNN for mobile applications
- DWs do not map efficiently on IMA

Execution Mappings

- Full SW solution (8-cores)
- All layers in IMA: 16 channels/job 54% IMA area overhead
- All layers in IMA: 8 channels/job 25% IMA area overhead
- Conv 1x1 on IMA while DW on 8 cores: 0% IMA area overhead
- Conv 1x1 on IMA, DW on Dig Acc., Final Add on SW Cores

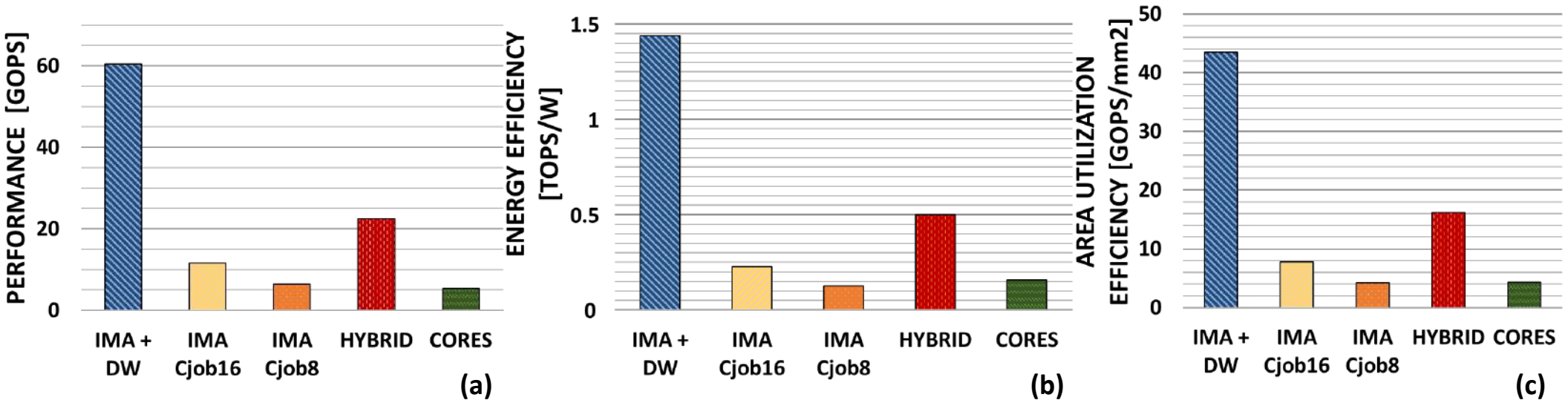
Mitigation of Amdahl's Law on Het. AIMC System

Execution of *Bottleneck* layer



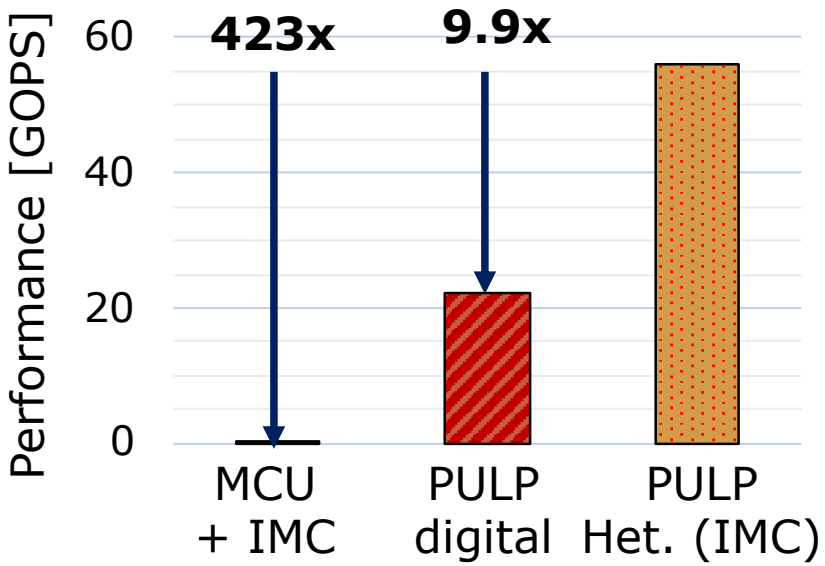
- AIMC cores inefficient to execute depth-wise convolutions
  - Poor utilization of the XBAR
- Better to split workload among diversified compute blocks

AIMC Performance, Energy and Area Efficiency



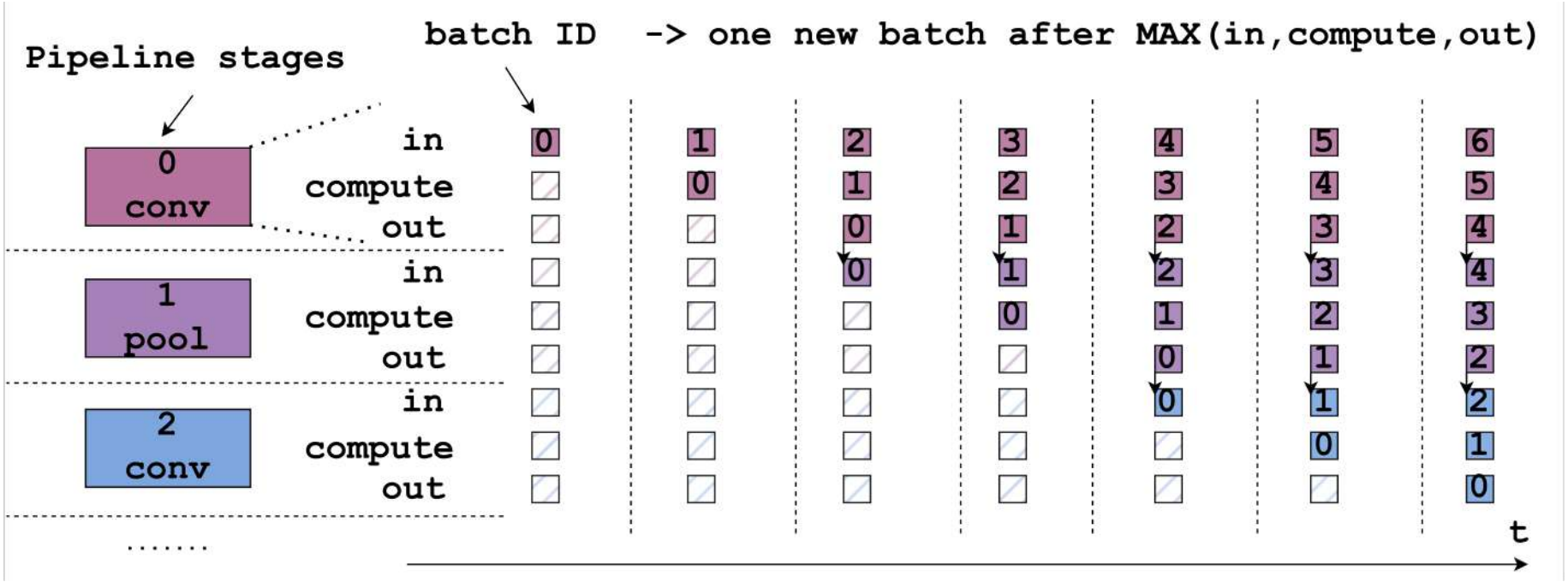
AIMC-Based System vs Purely Digital Systems

End-to-End execution of MobileNetV2



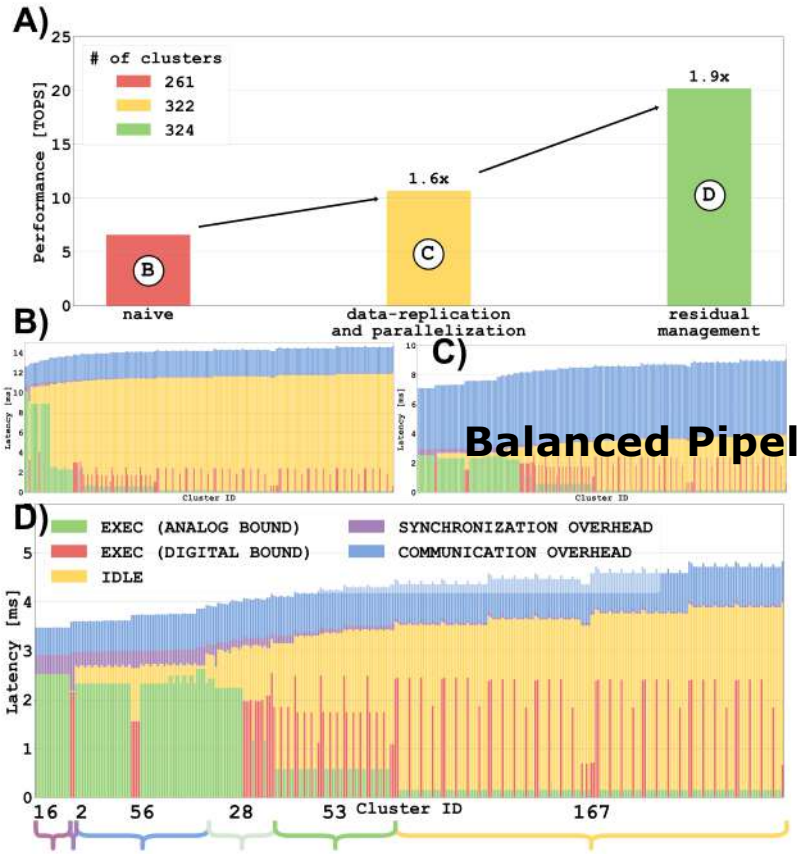
- Single-core MCU is inefficient to complement AIMC computing capabilities
- AIMC heterogeneous cluster for highest performance

AIMC Many-Core Computational Model



- Balancing the pipeline essential to achieve high Throughput
  - Data Replication, intra-layer parallelization
  - Non-trivial due to flow graph loops (e.g. *Residual layers)*

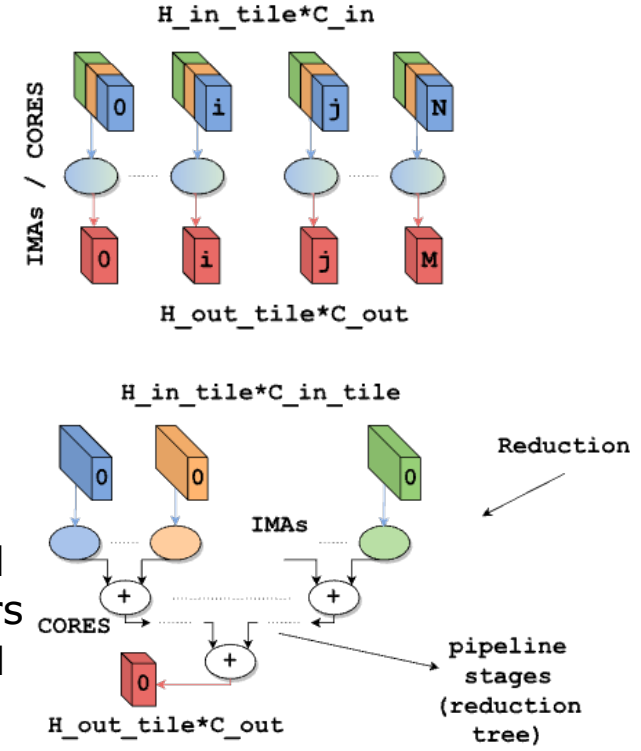
AIMC Many-Core System: Balancing The Pipeline



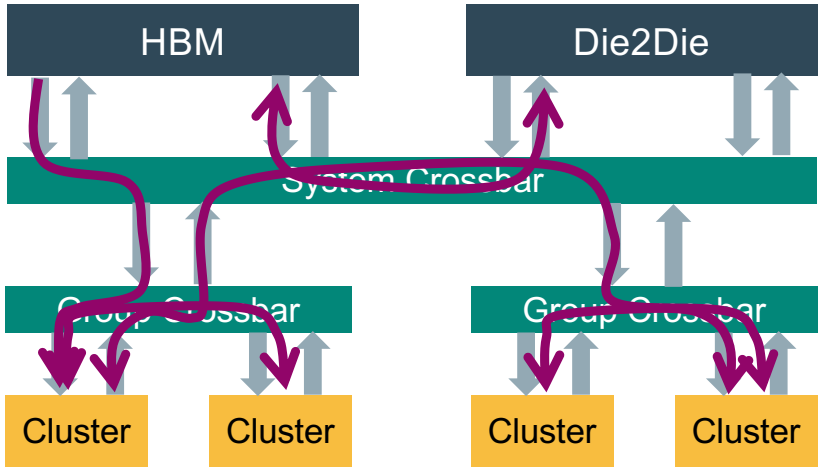
Balanced Pipeline

Residuals stored in unused clusters rather than HBM

Data-replication & Parallelization



Occamy NoC: Efficient and Flexible Data Movement



DORY: [Burrello et al. IEEE TC, 2021]

- Problem: HBM accesses are critical in terms of
  - Access energy
  - Congestion
  - High latency
- Solution: reuse data on lower levels of memory hierarchy
  - Between **clusters**
  - Across **groups**
- Smartly distribute workload
  - **Clusters:** DORY framework for tiling strategies
  - **Chiplets:** layer pipelining, ..