A 12.4 TOPS/W @ 136GOPS AI-IoT System-on-Chip with 16 RISC-V, 2-to-8b Precision-Scalable DNN Acceleration and 30%-Boost Adaptive Body Biasing

Francesco Conti[1], Davide Rossi[1], Gianna Paulin[2], Angelo Garofalo[1], Alfio Di Mauro[2], Georg Rutishauer[2], Gianmarco Ottavi[1], Manuel Eggimann[2], Hayate Okuhara[1], Vincent Huard[3], Olivier Montfort[3], Lionel Jure[3], Nils Exibard[3], Pascal Gouedo[3], Mathieu Louvat[3], Emmanuel Botte[3], Luca Benini[1,2]

[1]University of Bologna, [2]ETH Zürich, [3]Dolphin Design

ETH zürich

DOLPHIN DESIGN

WiPLASH

Introduction and Motivation

- **Emerging application areas for AI-enabled IoT**
  - **Personalized Healthcare**
  - **Augmented Reality**
  - **Nano-Robotics**
- **Challenges**
  - **High computational demand from DNNs, other algorithms**
  - **Diverse computational patterns and requirements**
- **Opportunities**
  - **Bit-precision tolerance**
  - **Accelerable workloads**

Introduction and Motivation

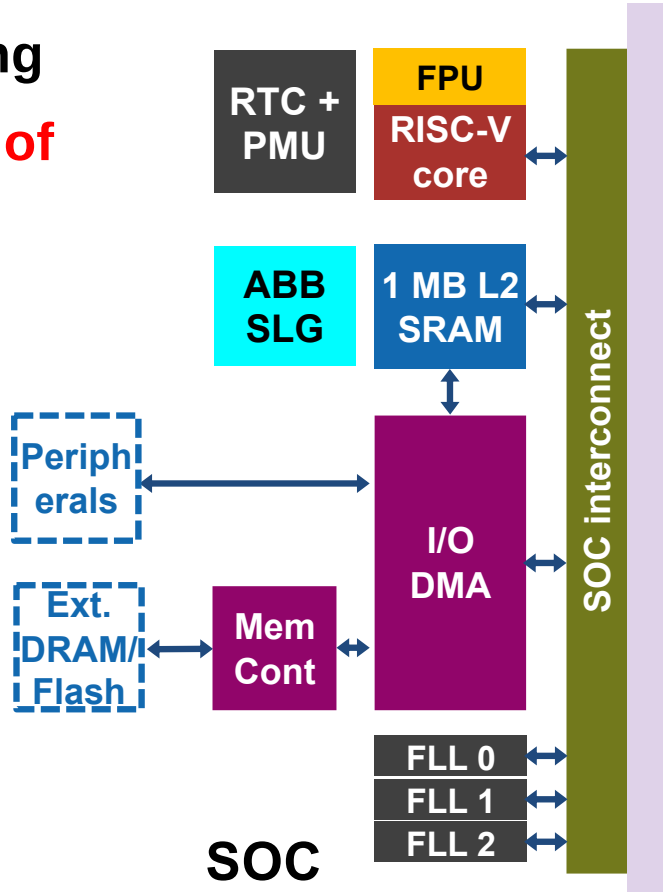
- **Emerging application areas for AI-enabled IoT**
  - Personalized Healthcare
  - Augmented Reality
  - Nano-Robotics
- **Challenges**
  - High computational demand from DNNs, other algorithms
  - Diverse computational patterns and requirements
- **Opportunities**
  - Bit-precision tolerance
  - Accelerable workloads

Introduction and Motivation

- **Emerging application areas for AI-enabled IoT**
  - Personalized Healthcare
  - Augmented Reality
  - Nano-Robotics
- **Challenges**
  - High computational demand from DNNs, other algorithms
  - Diverse computational patterns and requirements
- **Opportunities**
  - Bit-precision tolerance
  - Accelerable workloads

MARSELLUS: AI-IoT Heterogeneous SoC

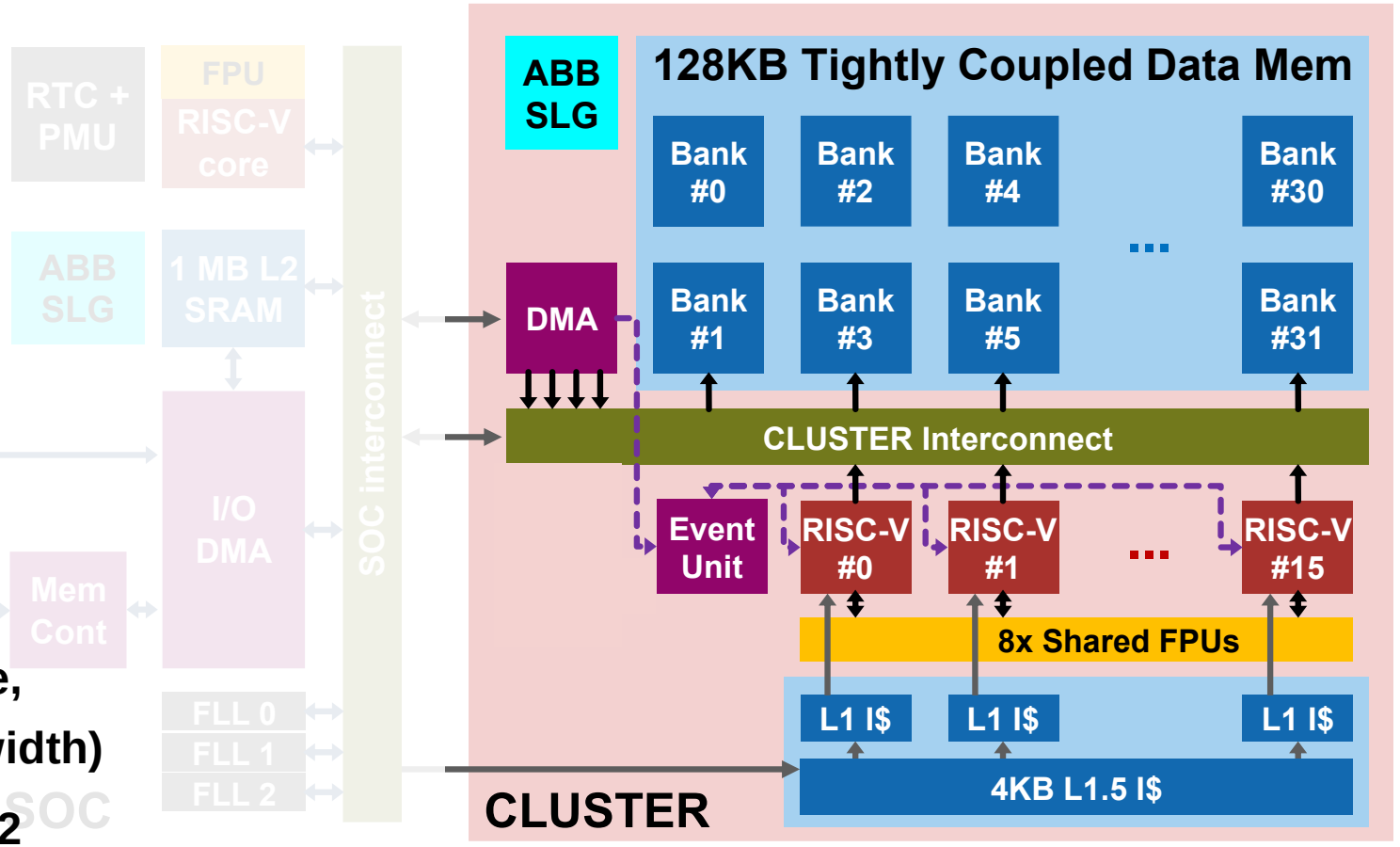
- Programmable RISC-V based System-on-Chip combining
  1. RISC-V MCU with 1MB of on-board SRAM
     - 32-bit *RV32IMCFXpulp*
     - 1MB L2 SRAM
     - Standard peripherals (SPI, I2C, UART, ...)
     - Off-chip *HyperRAM™ DRAM/FLASH
     - Autonomous I/O DMA
     - 3 Freq-Locked Loops



*https://www.cypress.com/products/hyperram-memory

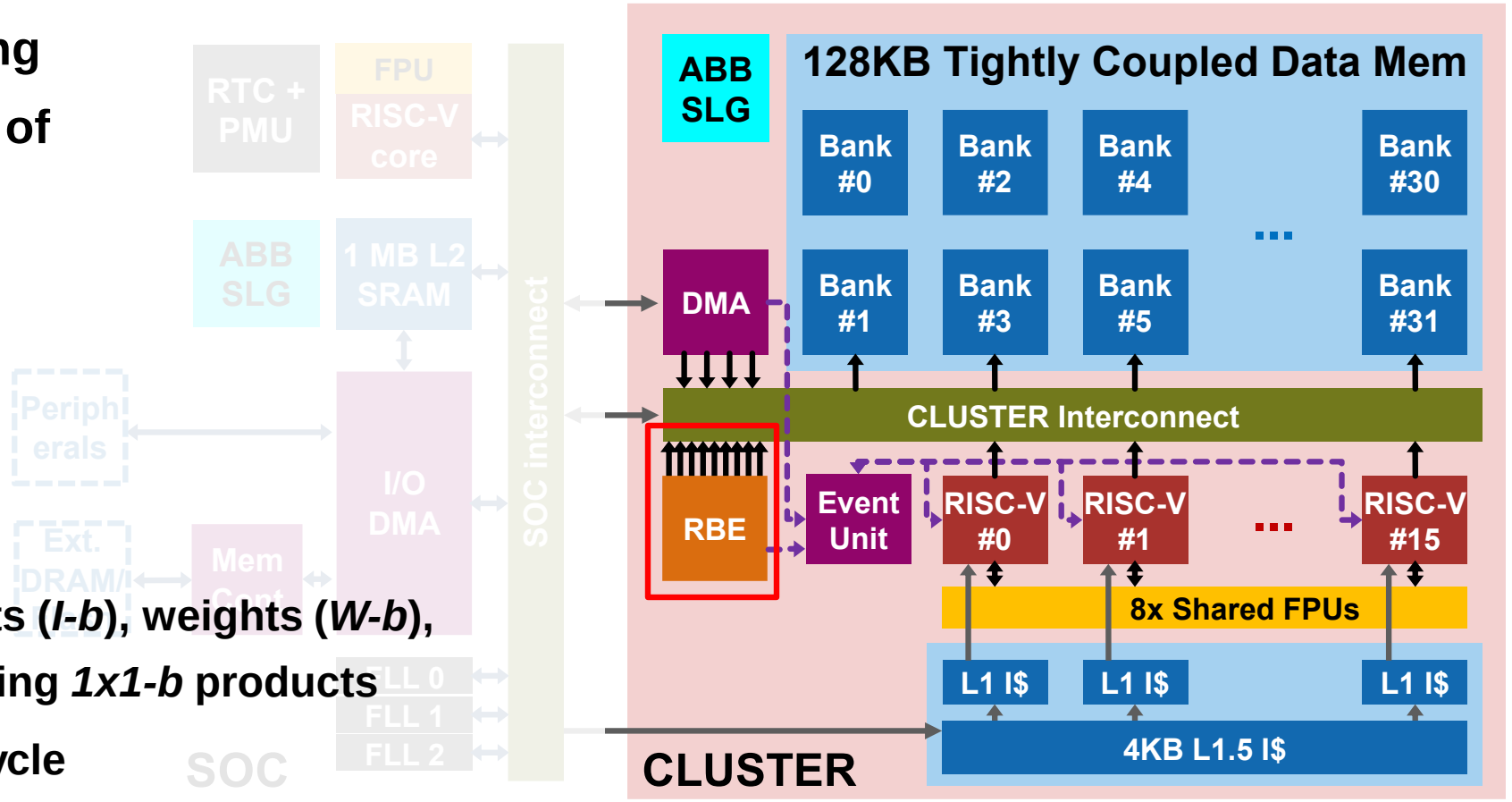
MARSELLUS: AI-IoT Heterogeneous SoC

- Programmable RISC-V based System-on-Chip combining
  1. RISC-V MCU with 1MB of on-board SRAM
  2. Cluster of 16 RISC-V 2-32b DSP cores
     - 16x *RV32IMCFXpulpNN*
     - 2b/4b + fused MAC&LOAD
     - 8x FP32/FP16/BF16 FPUs
     - 128 KB of L1 TCDM (0-wait-state, 400Gb/s @ 420MHz aggr. bandwidth)
     - DMA w/ 64b L2→L1 + 64b L1→L2



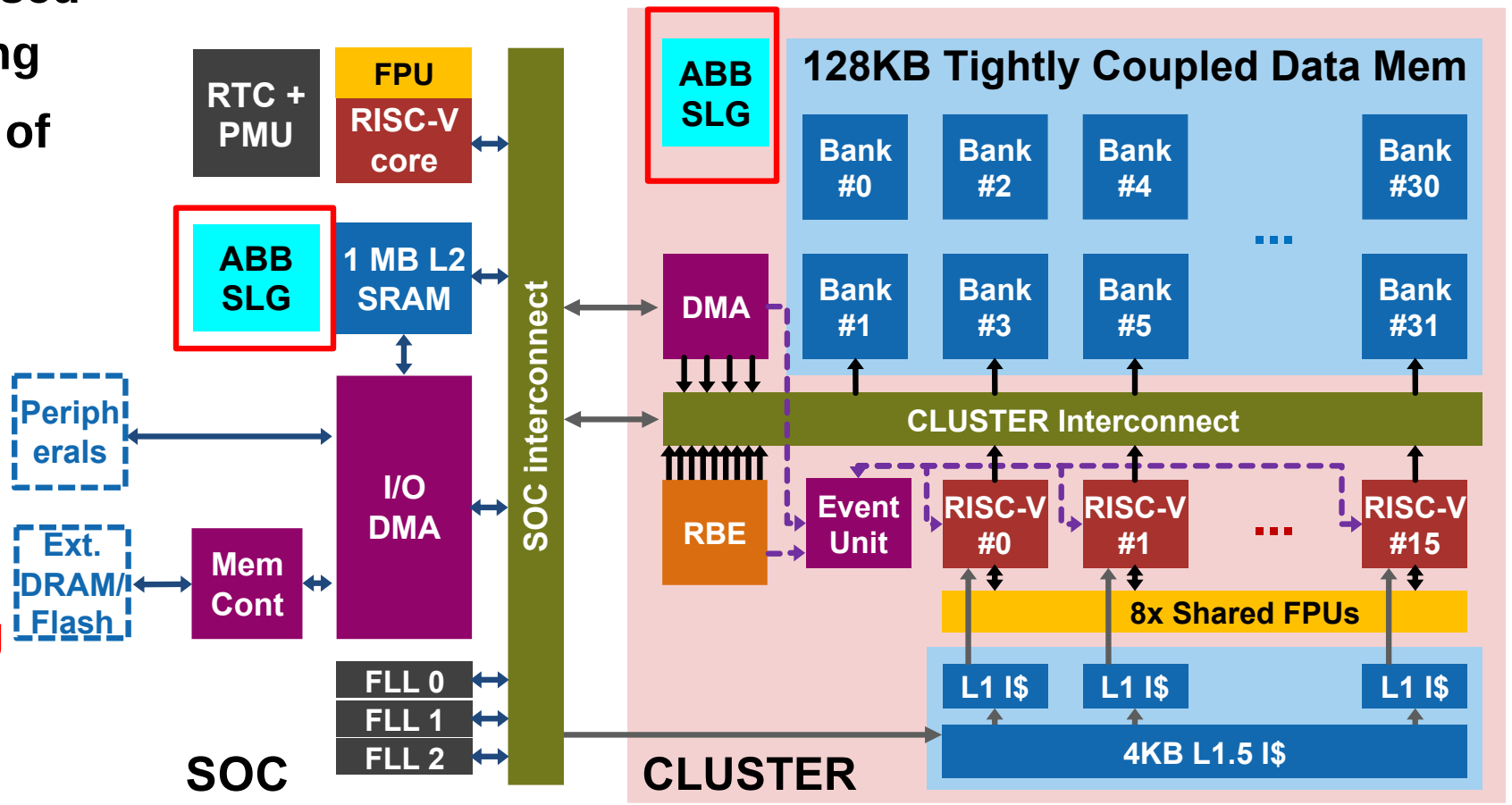
MARSELLUS: AI-IoT Heterogeneous SoC

- Programmable RISC-V based System-on-Chip combining
  1. RISC-V MCU with 1MB of on-board SRAM
  2. Cluster of 16 RISC-V 2-32b DSP cores
  3. 2-8b Reconfigurable Binary Engine for 3x3, 1x1 DNN kernels
     - asymmetric 2-8b inputs (*I-b*), weights (*W-b*), outputs (*O-b*) composing *1x1-b* products
     - up to 10^4 1x1b-MAC/cycle



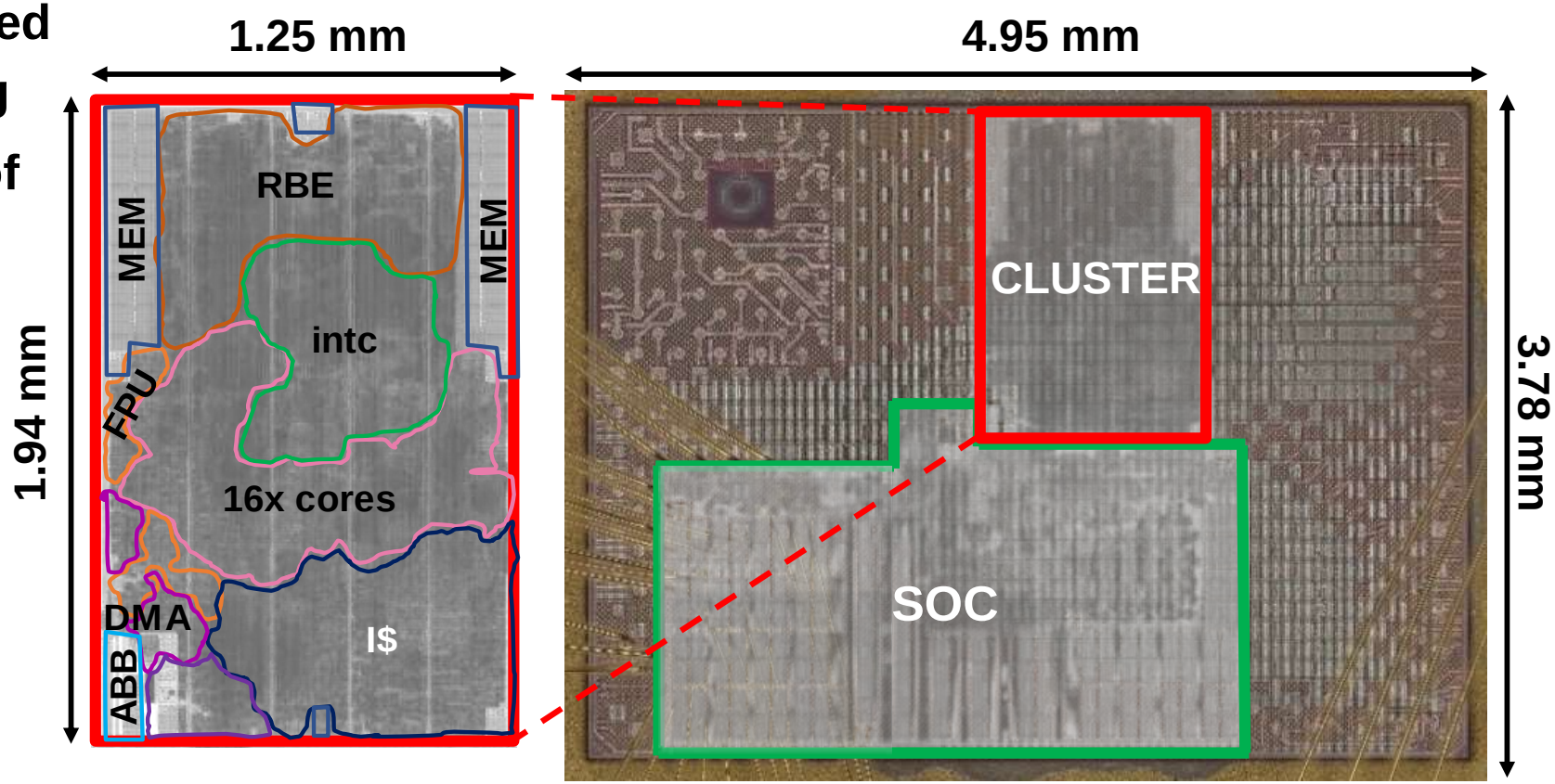
MARSELLUS: AI-IoT Heterogeneous SoC

- Programmable RISC-V based System-on-Chip combining
  1. RISC-V MCU with 1MB of on-board SRAM
  2. Cluster of 16 RISC-V 2-32b DSP cores
  3. 2-8b Reconfigurable Binary Engine for 3x3, 1x1 DNN kernels
  4. Adaptive Body Biasing with on-the-fly control
     - OCMs added to 1% most timing-critical endpoints
     - Adaptively boost freq or efficiency



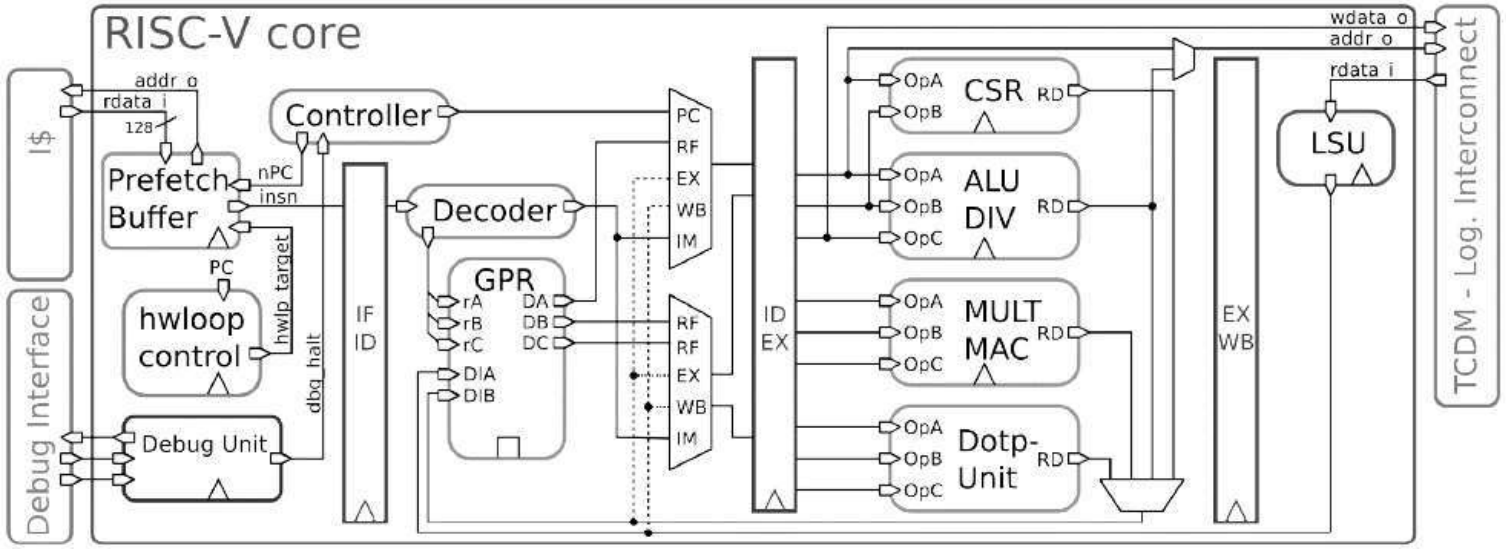
MARSELLUS: AI-IoT Heterogeneous SoC

- Programmable RISC-V based System-on-Chip combining
  1. RISC-V MCU with 1MB of on-board SRAM
  2. Cluster of 16 RISC-V 2-32b DSP cores
  3. 2-8b Reconfigurable Binary Engine for 3x3, 1x1 DNN kernels
  4. Adaptive Body Biasing with on-the-fly control

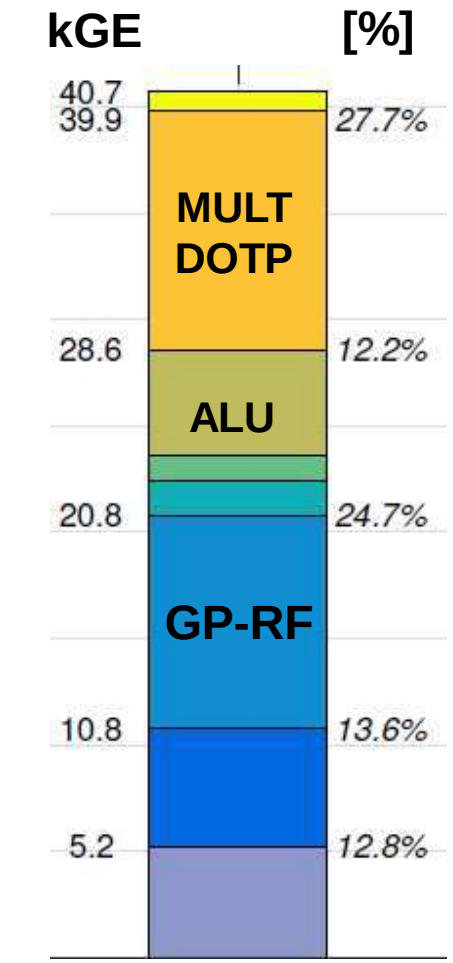


- Prototype implemented in GF 22FDX
  - ➢ flip-well LVT & SLVT cells, 2.43mm^2 for CLUSTER

Baseline RISC-V Core



40 kGE
70% RF+DP



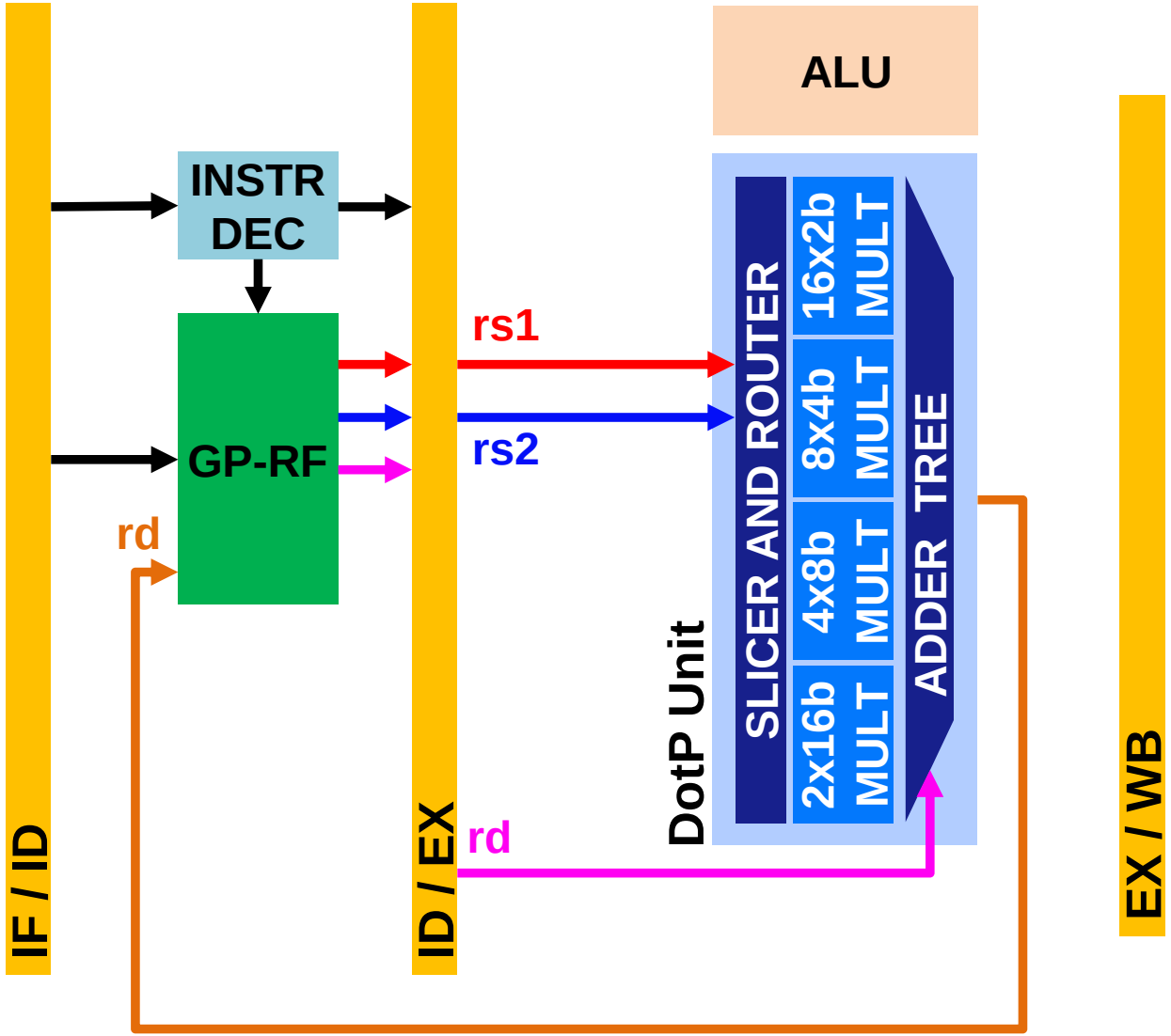
AREA BREAKDOWN

V1 ISA Baseline RISC-V RV32IMC

V2 ISA HW loops + post modified load/store + mac

V3 ISA SIMD 2/4 + dot product + shuffling +
bit manipulation unit + lightweight fixed point

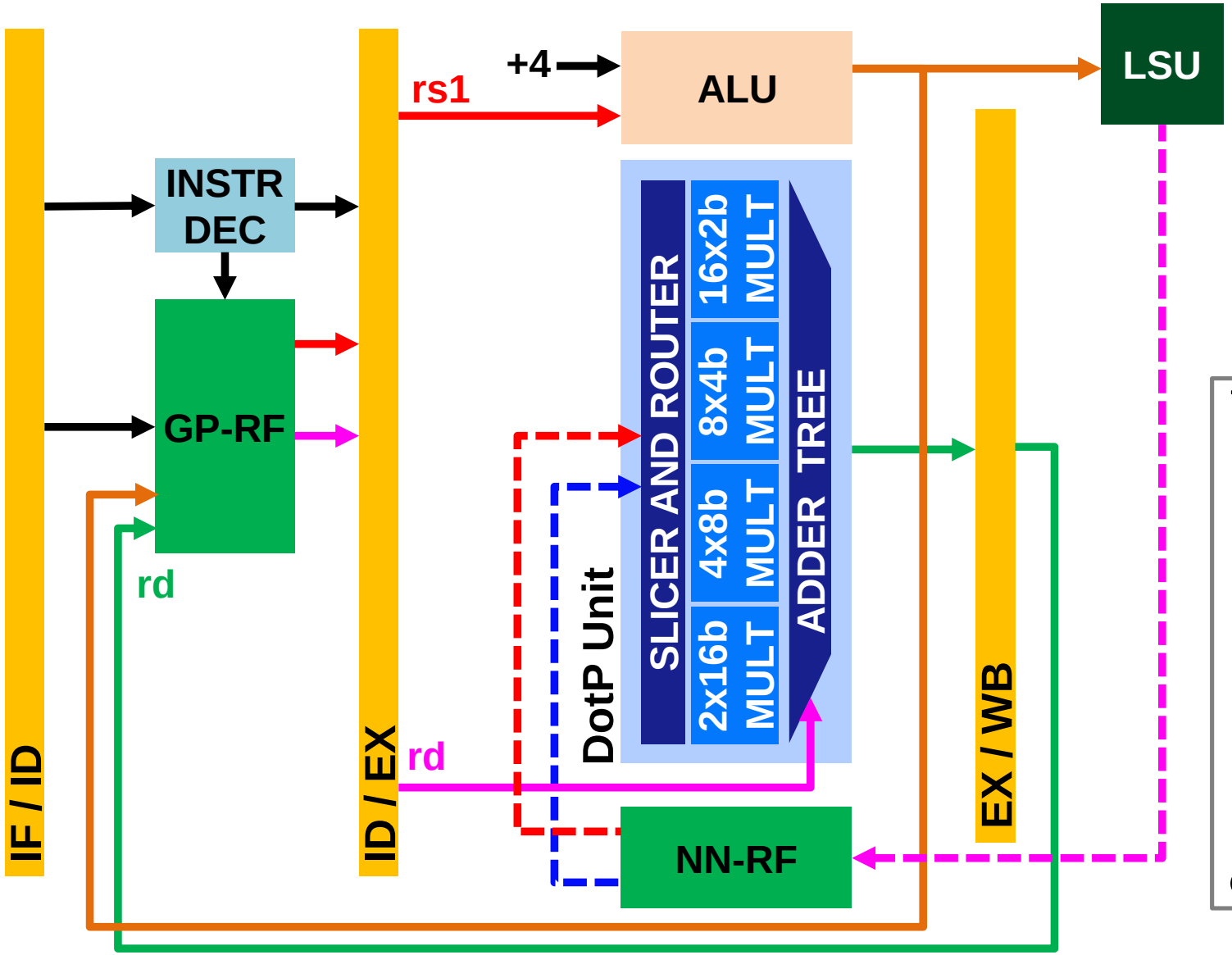
RISC-V MAC&LD core architecture



- *Xpulp* Dot-Product achieves up to 2.3 MAC/cycle/core @ 8-bit

```
lp.setup
  p.lw  w0, 4(w0!)
  p.lw  w1, 4(w1!)
  p.lw  w2, 4(w2!)
  p.lw  w3, 4(w3!)
  p.lw  x1, 4(x0!)
  p.lw  x2, 4(x1!)
  pv.sdotsp.b acc1, w0, x0
  pv.sdotsp.b acc2, w0, x1
  pv.sdotsp.b acc3, w1, x0
  pv.sdotsp.b acc4, w1, x1
  pv.sdotsp.b acc5, w2, x0
  pv.sdotsp.b acc6, w2, x1
  pv.sdotsp.b acc7, w3, x0
  pv.sdotsp.b acc8, w3, x1
end                 2.29 MAC/cycle/core
```

RISC-V MAC&LD core architecture

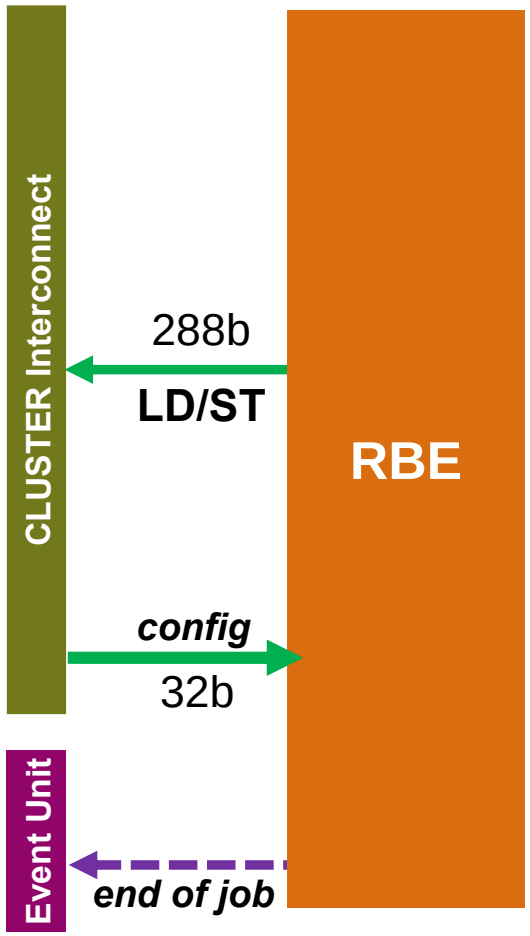


- *XpulpNN* adds new Dot-Product instructions that exploit a dedicated NN-RF
  1. to boost weight/input reuse
  2. to overlap loads and dot-product

```
lp.setup
  pv.nnsdotup.h zero, x1, 9
  pv.nnsdotsp.b acc1, w2, 0
  pv.nnsdotsp.b acc2, w4, 2
  pv.nnsdotsp.b acc3, w3, 4
  pv.nnsdotsp.b acc4, x1, 14
  pv.nnsdotsp.b acc5, w2, 17
  pv.nnsdotsp.b acc6, w4, 19
  pv.nnsdotsp.b acc7, w3, 21
  pv.nnsdotsp.b acc8, w1, 23
end
```

3.56 MAC/cycle (+55% boost)

Reconfigurable Binary Engine (RBE)



- Partially bit-serial dataflow
- "Bit" dimension is decomposed, in inputs, weights, outputs

$$y(k_{out}) = quant\left(\sum_{k_{in}}\left(W(k_{out}, k_{in}) \otimes x(k_{in})\right)\right)$$

32b accumulator

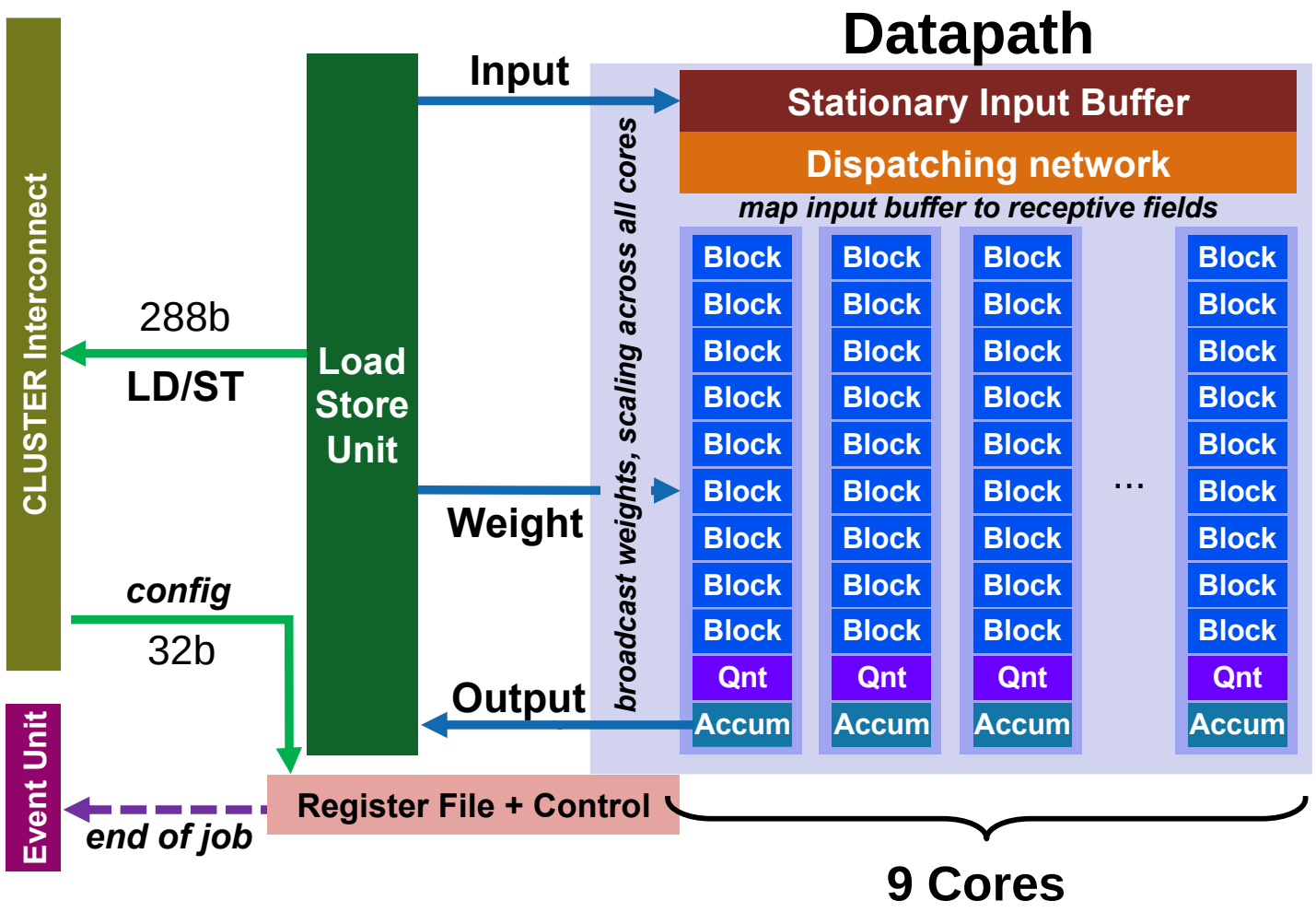
O-b output

W-b weights

I-b input

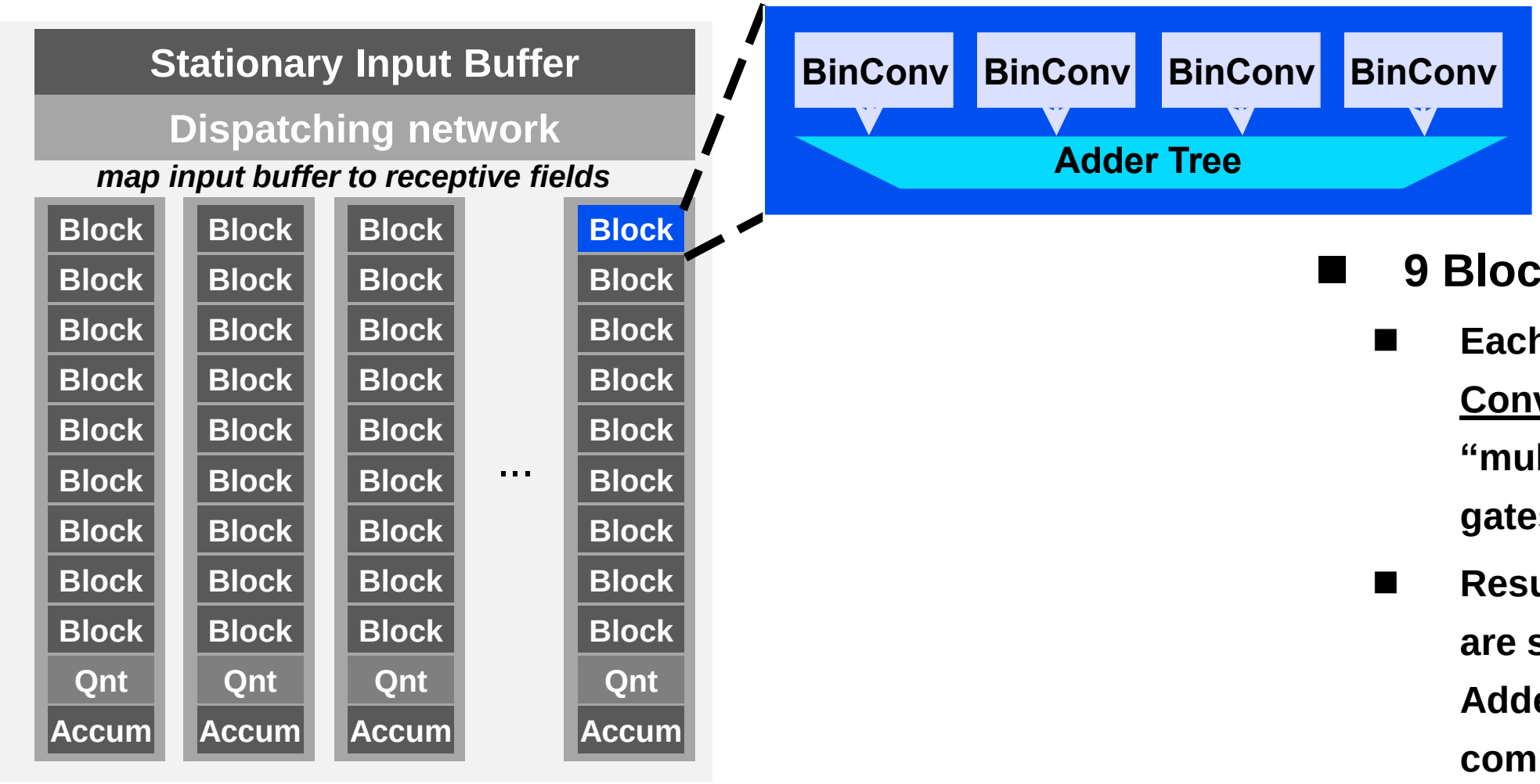
- DNN layer operating at *WxI*-b decomposed in *WxI* 1x1b MAC ops
  - RBE tensors use a special layout in memory to expose 1x1b-ops
- RBE is integrated in cluster with
  - 288b LD/ST initiator port toward CLUSTER interconnect
  - 32b config port + "end of job" event

Reconfigurable Binary Engine (RBE)



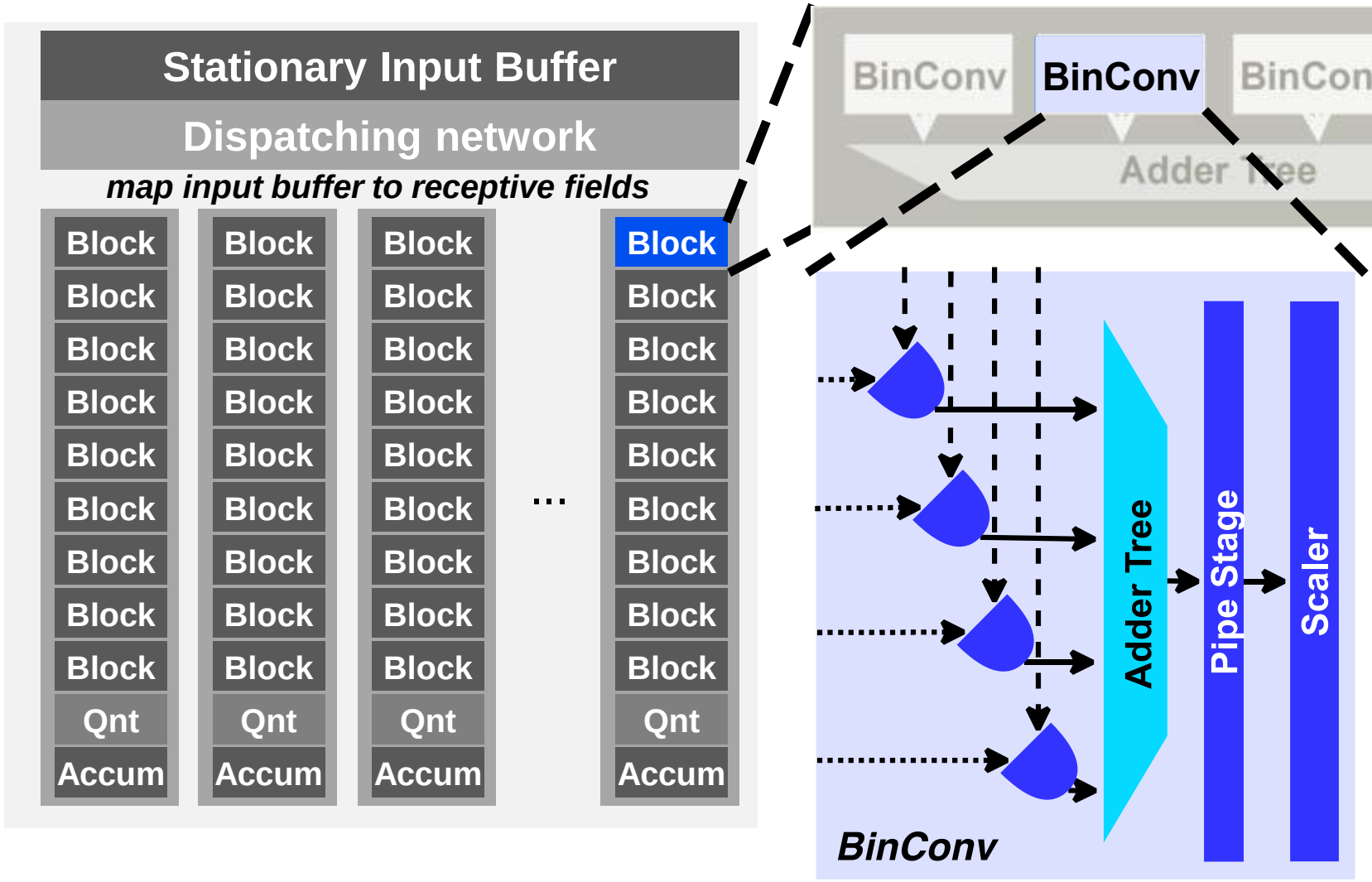
- RBE is 652 kGE divided in
  - Register File + Control (6%)
  - 288b Load/Store Unit (2%)
  - Datapath (92%)
- Datapath comprise 9 Cores
  - 1 Core = receptive field of one output in space across 32 Kout
  - Stationary input statically multi-casted to each Core receptive field
  - Output stored in 32x 32b accumulators per Core + Quantized
  - Weights renovated each cycle and broadcasted on all Cores

Reconfigurable Binary Engine (RBE)



- 9 Blocks per Core
  - Each with 4 Binary Convolvers with 32 1x1b "multipliers" (i.e., AND gates)
  - Results from all BinConvs are scaled, cumulated with Adder Trees to form a complete contribution

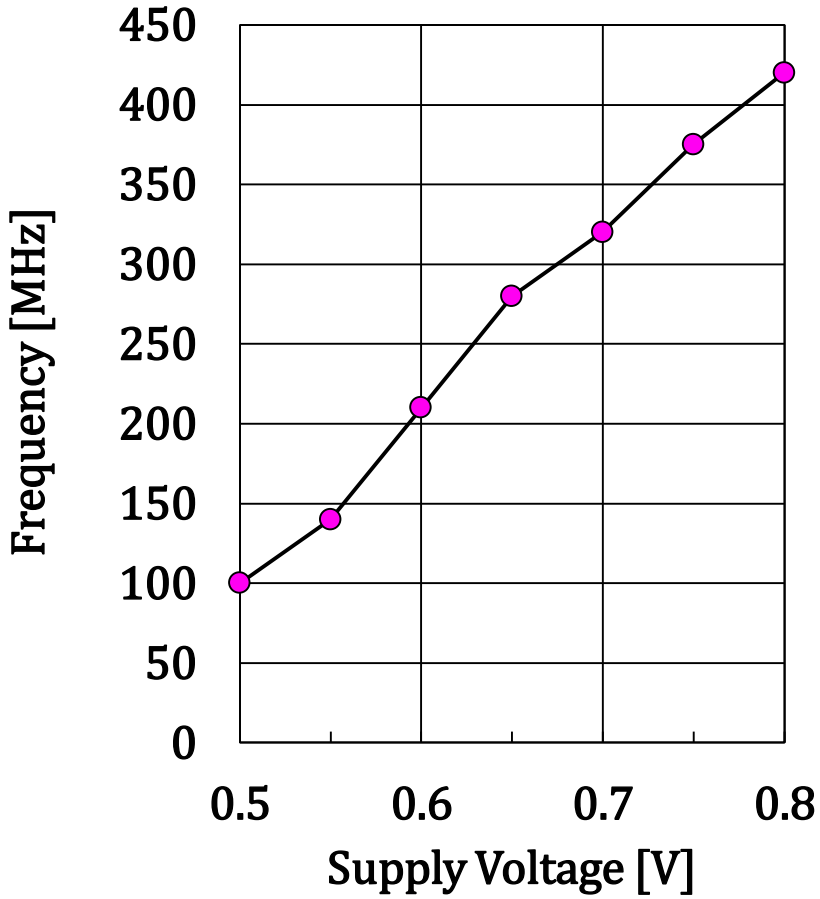
Reconfigurable Binary Engine (RBE)



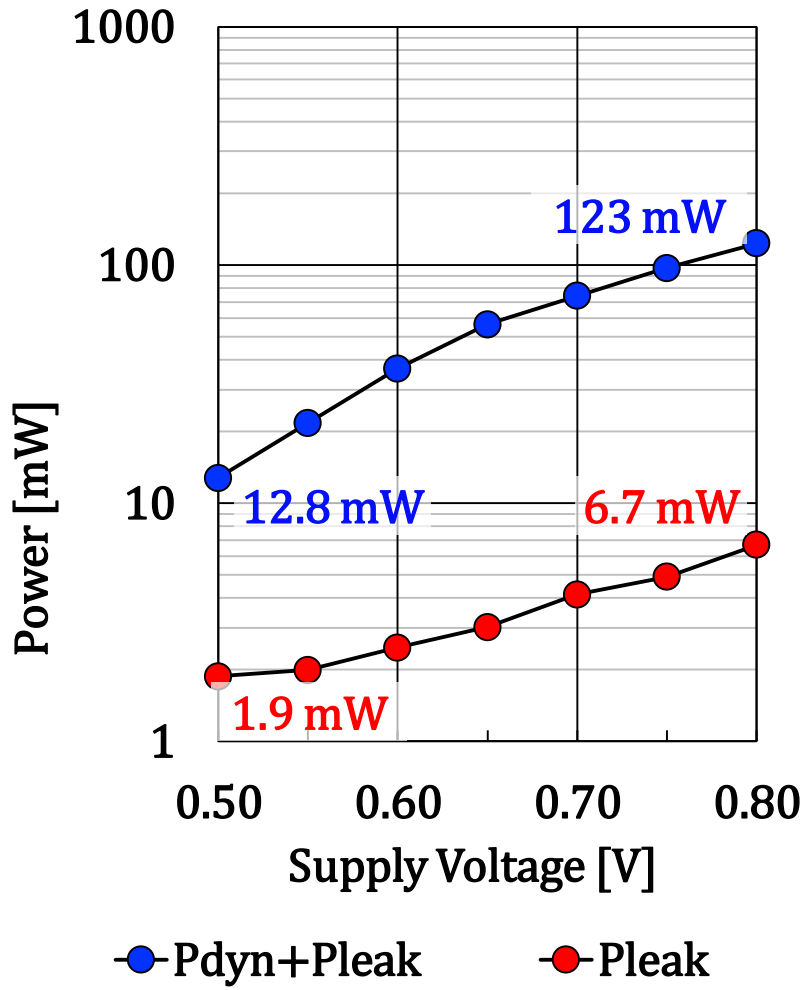
- 9 Blocks per Core
  - Each with 4 Binary Convolvers with 32 1x1b "multipliers" (i.e., AND gates)
  - Results from all BinConvs are scaled, cumulated with Adder Trees to form a complete contribution

Frequency & Power

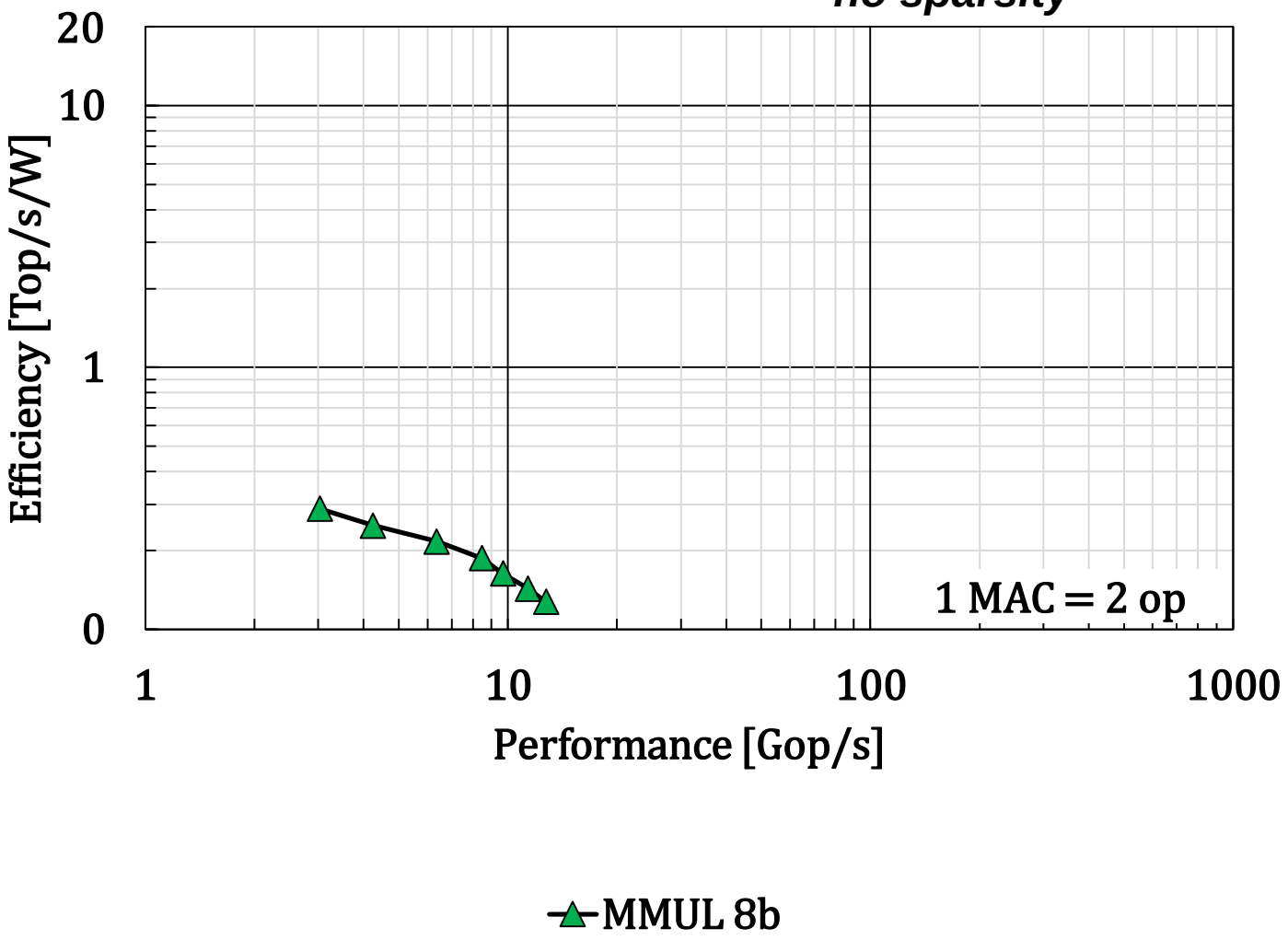
- *MARSELLUS* is operational within 0.5-0.8V voltage range
- Frequency range with no Vbb 100 MHz @ 0.5V → 420 MHz @ 0.8V
- CLUSTER power ranges between 12.8mW @ 100MHz, 0.5V → 123mW @ 420MHz, 0.8V



Workload: MatMul MAC&LD
no sparsity



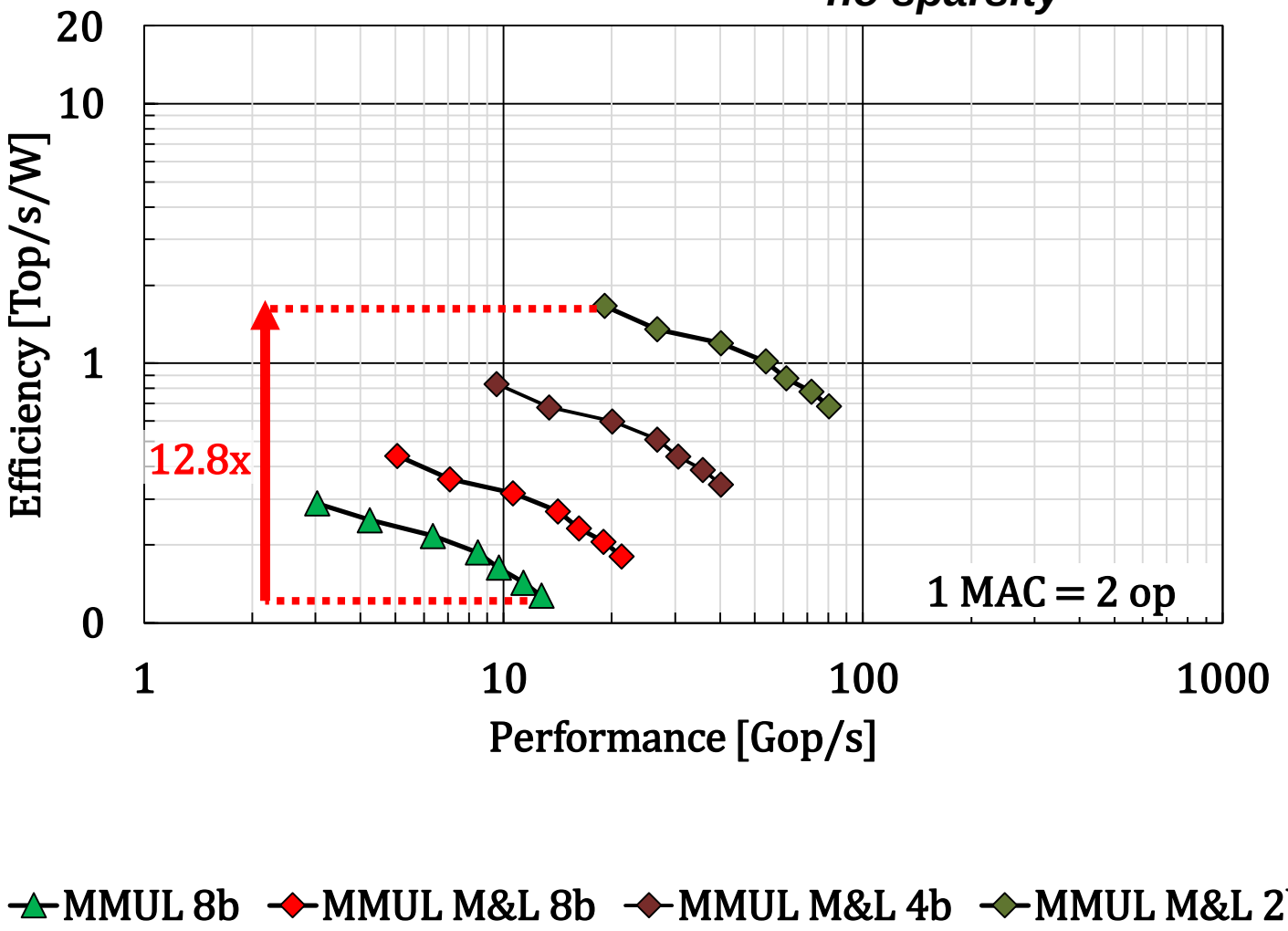
Efficiency Sweep



- **Baseline efficiency with 16 cores not using M&L is 130 Gop/s/W @ 0.8V → 290 Gop/s/W @ 0.5V**

Efficiency Sweep

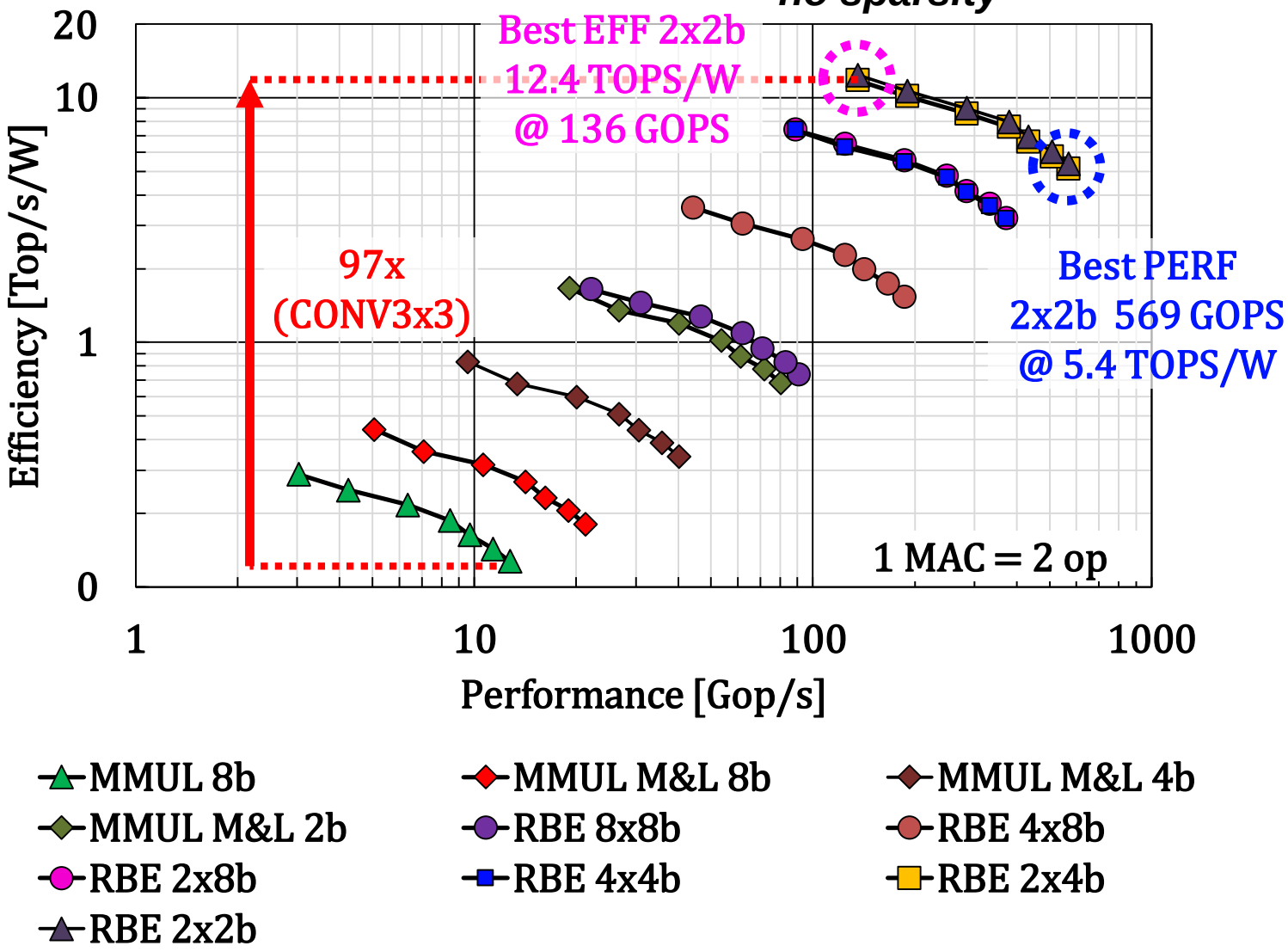
SW Workload: MatMul
no sparsity



- **Baseline efficiency with 16 cores not using M&L is 130 Gop/s/W @ 0.8V → 290 Gop/s/W @ 0.5V**
- **M&L and quantization down to 2b bring up to 12.8x improvement in efficiency**

Efficiency Sweep

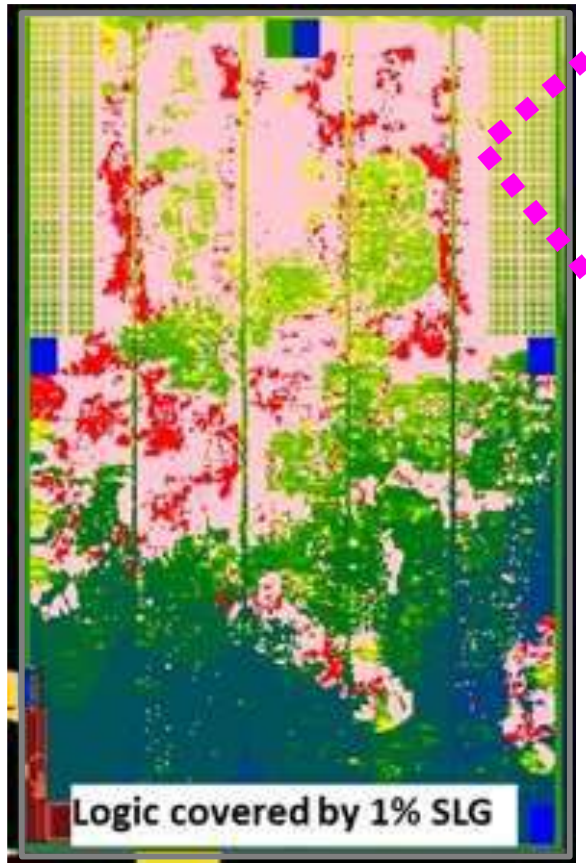
HW Workload: Conv3x3 32x16x3x3
no sparsity



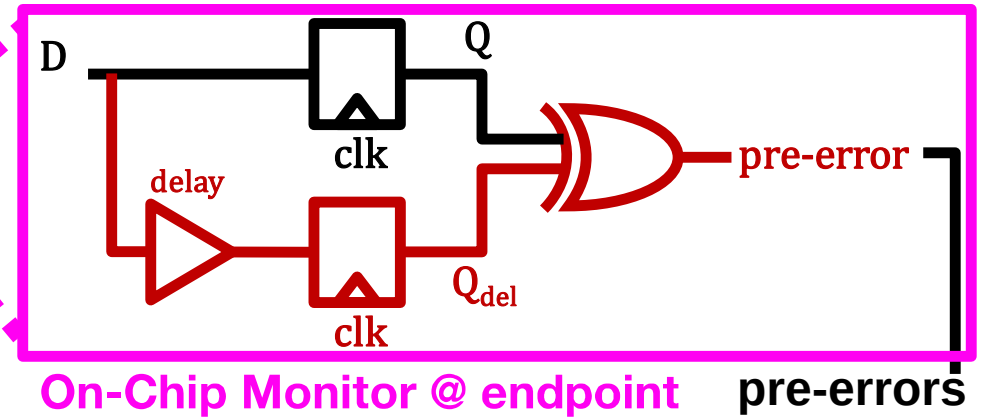
- Baseline efficiency with 16 cores not using M&L is 130 Gop/s/W @ 0.8V → 290 Gop/s/W @ 0.5V
- M&L and quantization down to 2b bring up to 12.8x improvement in efficiency
- Employing RBE on CONV3x3 kernels efficiency can be improved by a further 7.6x, up to 12.4Top/s/W

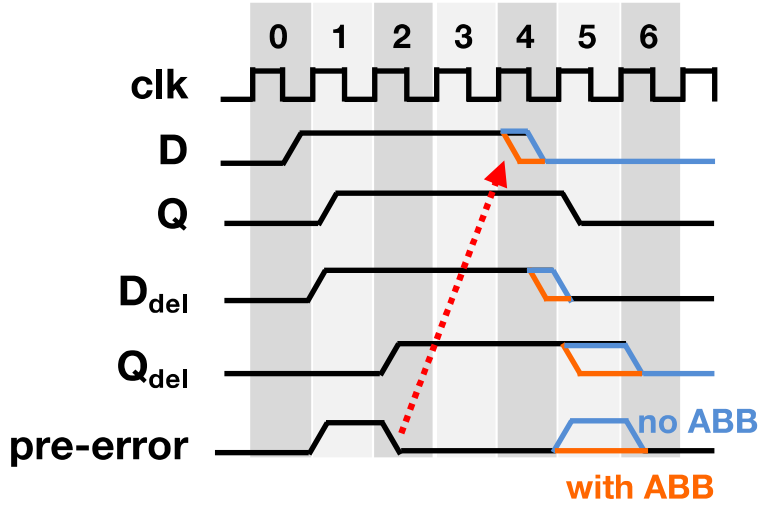
Adaptive Body Biasing

- **On-Chip Monitors detect pre-error conditions**
  - 1% most critical endpoints
  - shadow delayed FF compared with regular FF → difference = pre-error



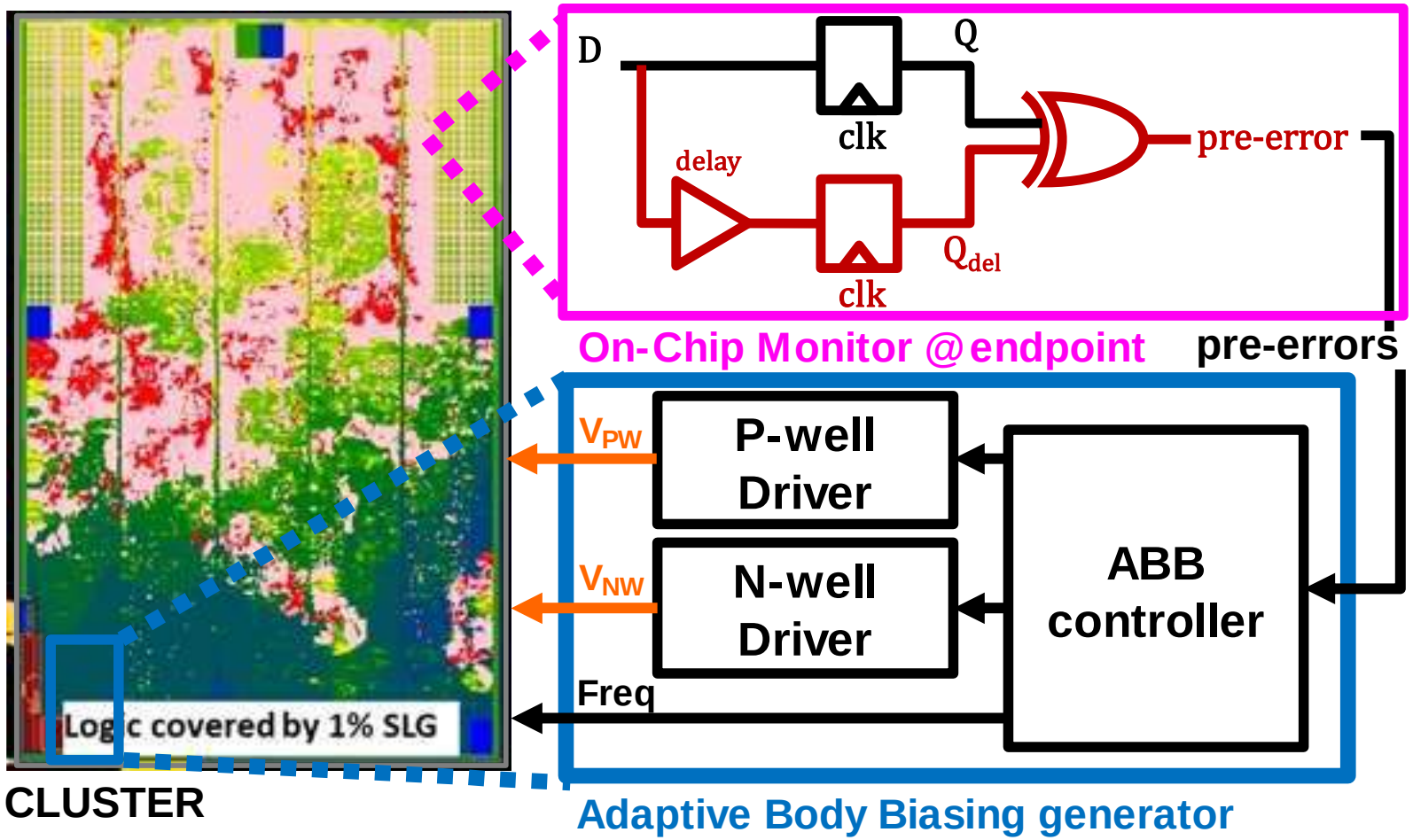
CLUSTER





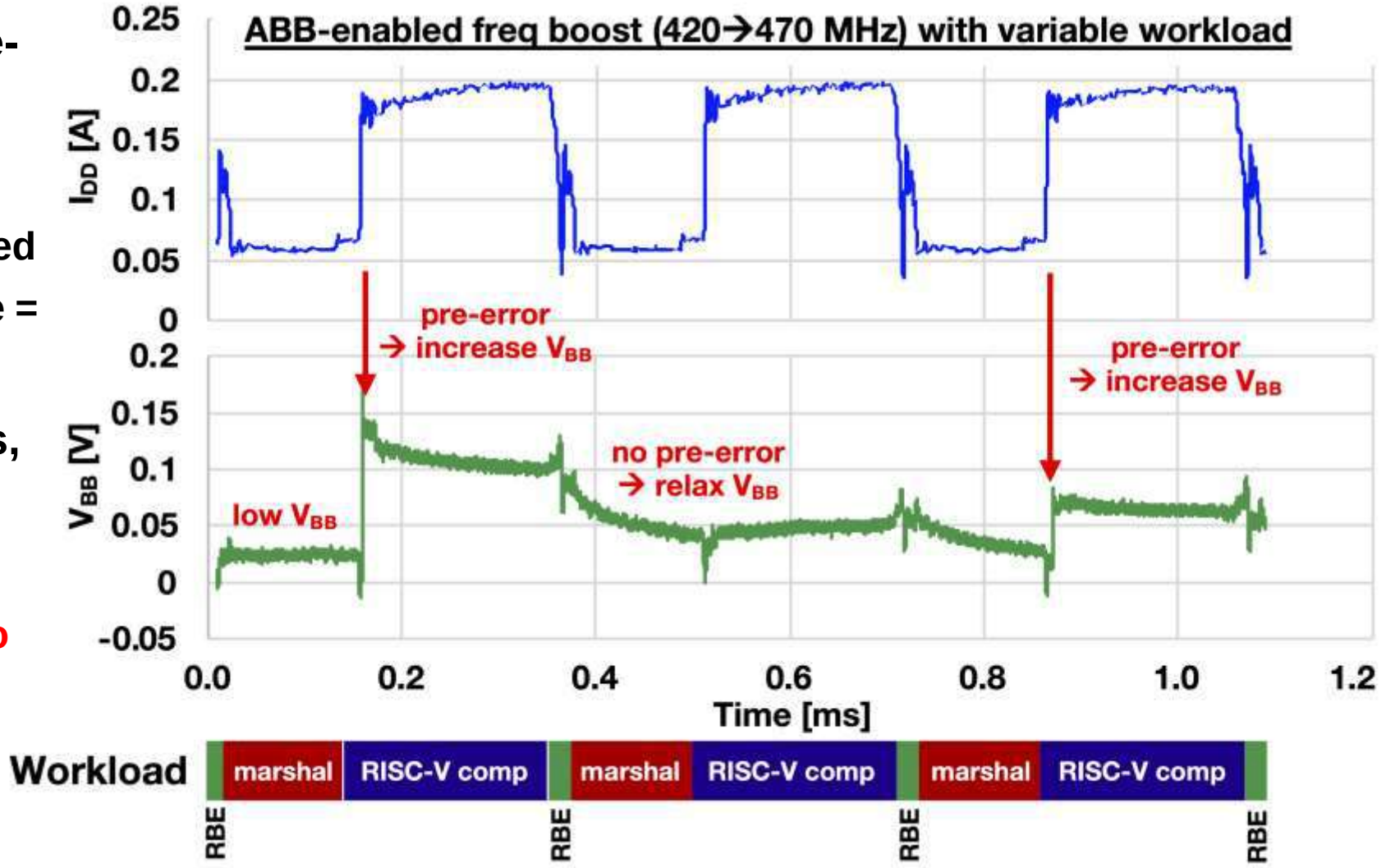
Adaptive Body Biasing

- On-Chip Monitors detect pre-error conditions
  - 1% most critical endpoints
  - shadow delayed FF compared with regular FF → difference = pre-error
- When OCM detect pre-errors, ABB generators adaptively tweak body biasing



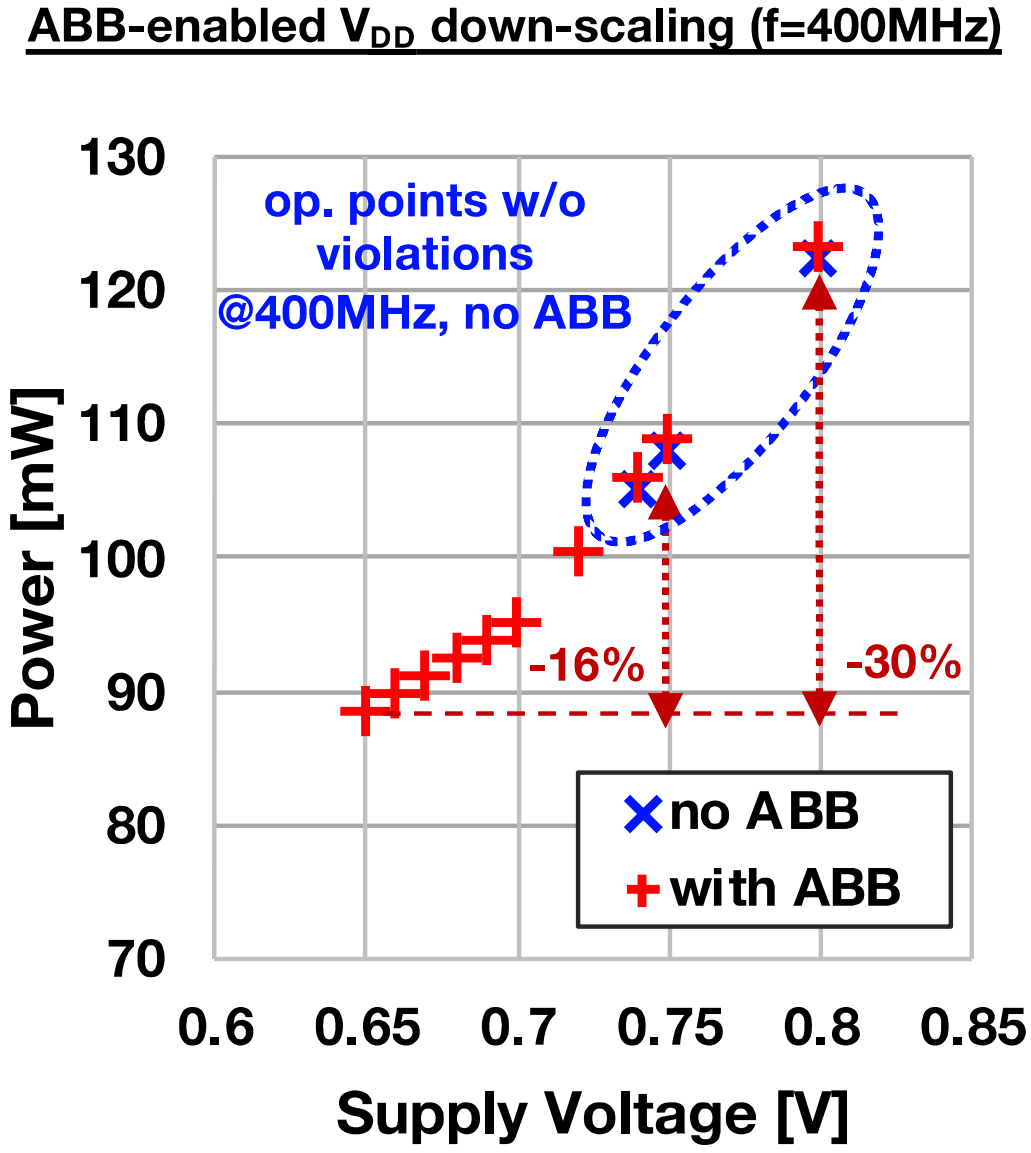
Adaptive Body Biasing

- **On-Chip Monitors** detect pre-error conditions
  - 1% most critical endpoints
  - shadow delayed FF compared with regular FF → difference = pre-error
- When OCM detect pre-errors, **ABB generators** adaptively tweak body biasing
  - Either boost frequency up to 12% (420 → 470MHz)

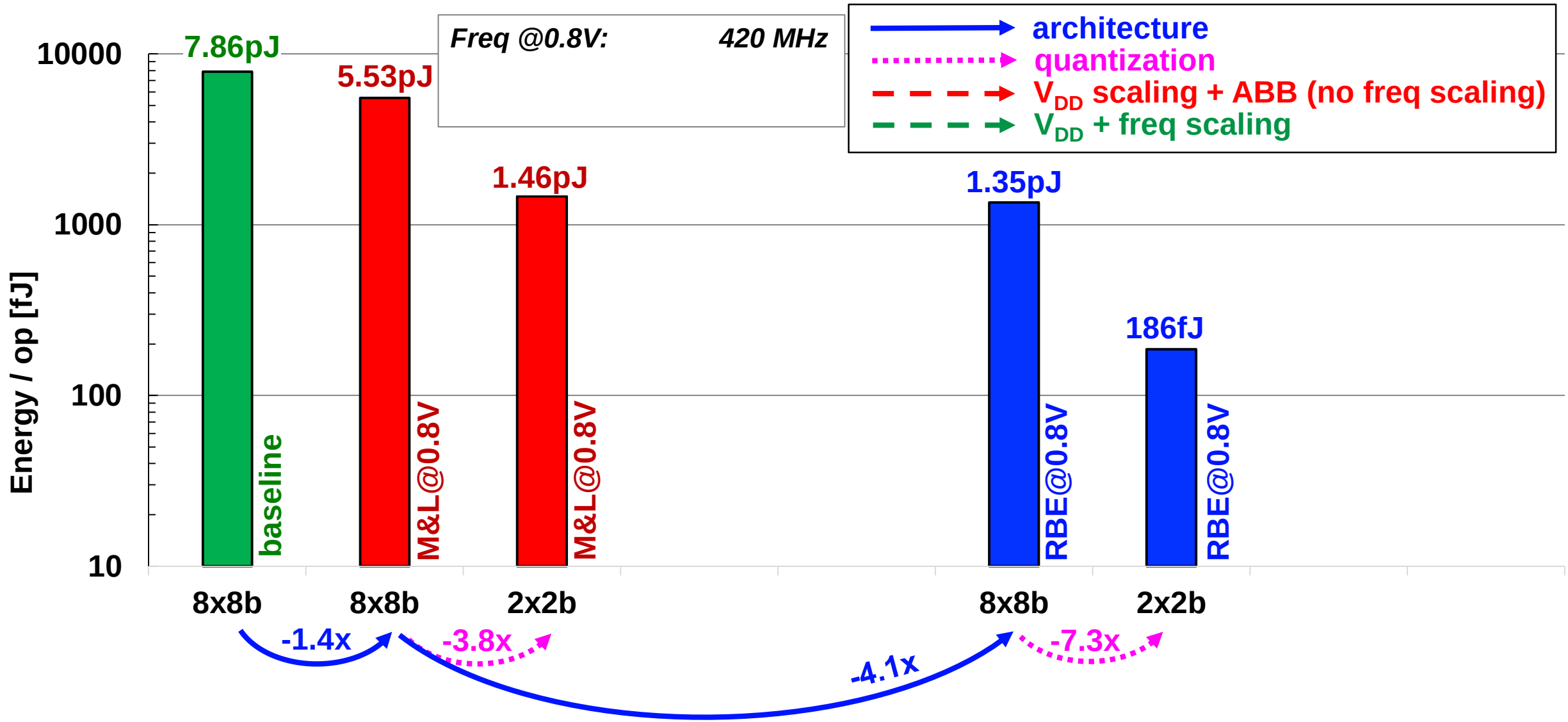


Adaptive Body Biasing

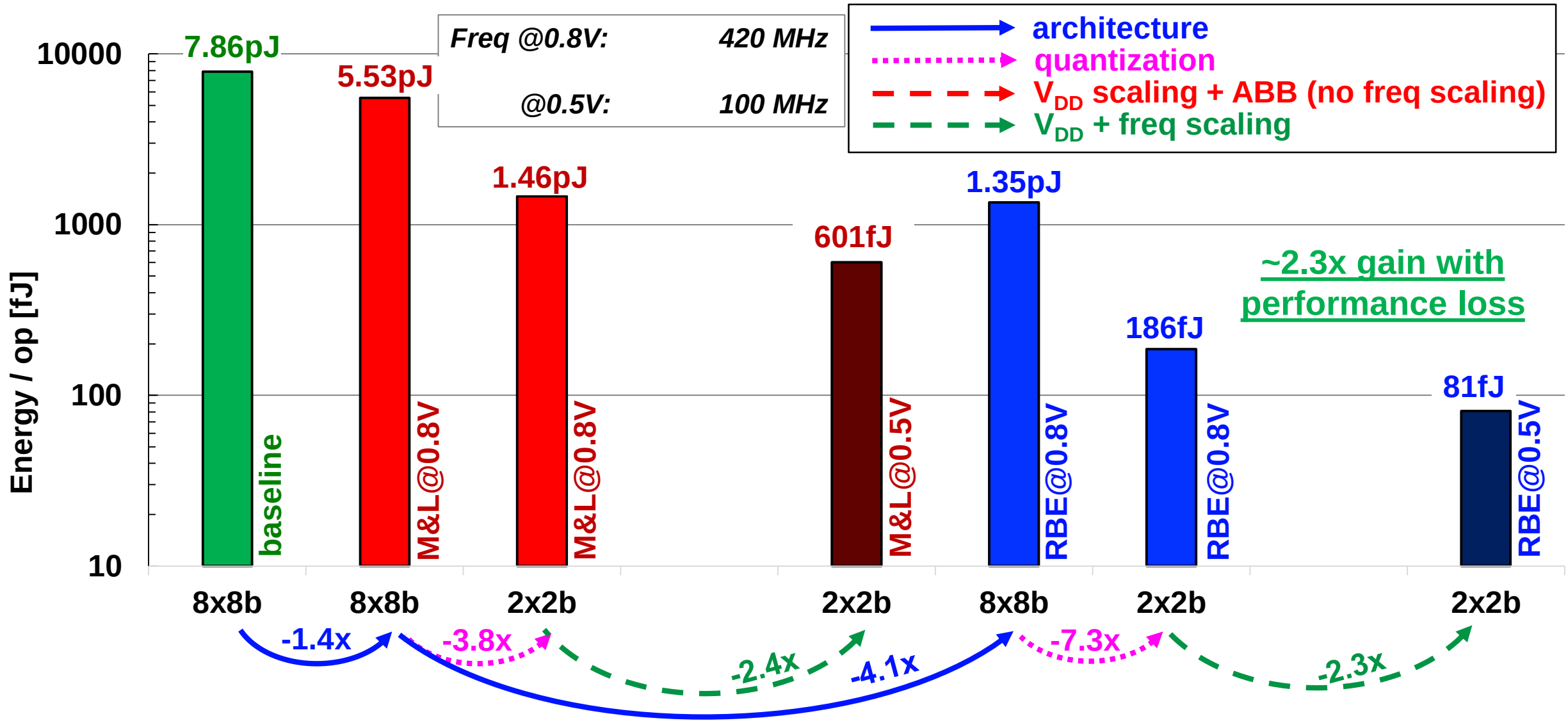
- On-Chip Monitors detect pre-error conditions
  - 1% most critical endpoints
  - shadow delayed FF compared with regular FF → difference = pre-error
- When OCM detect pre-errors, ABB generators adaptively tweak body biasing
  - Either boost frequency up to 12% (420 → 470MHz)
  - Or keep a fixed frequency target (e.g., 400 MHz) while dropping Vdd (0.8 → 0.65V) → +30% efficiency



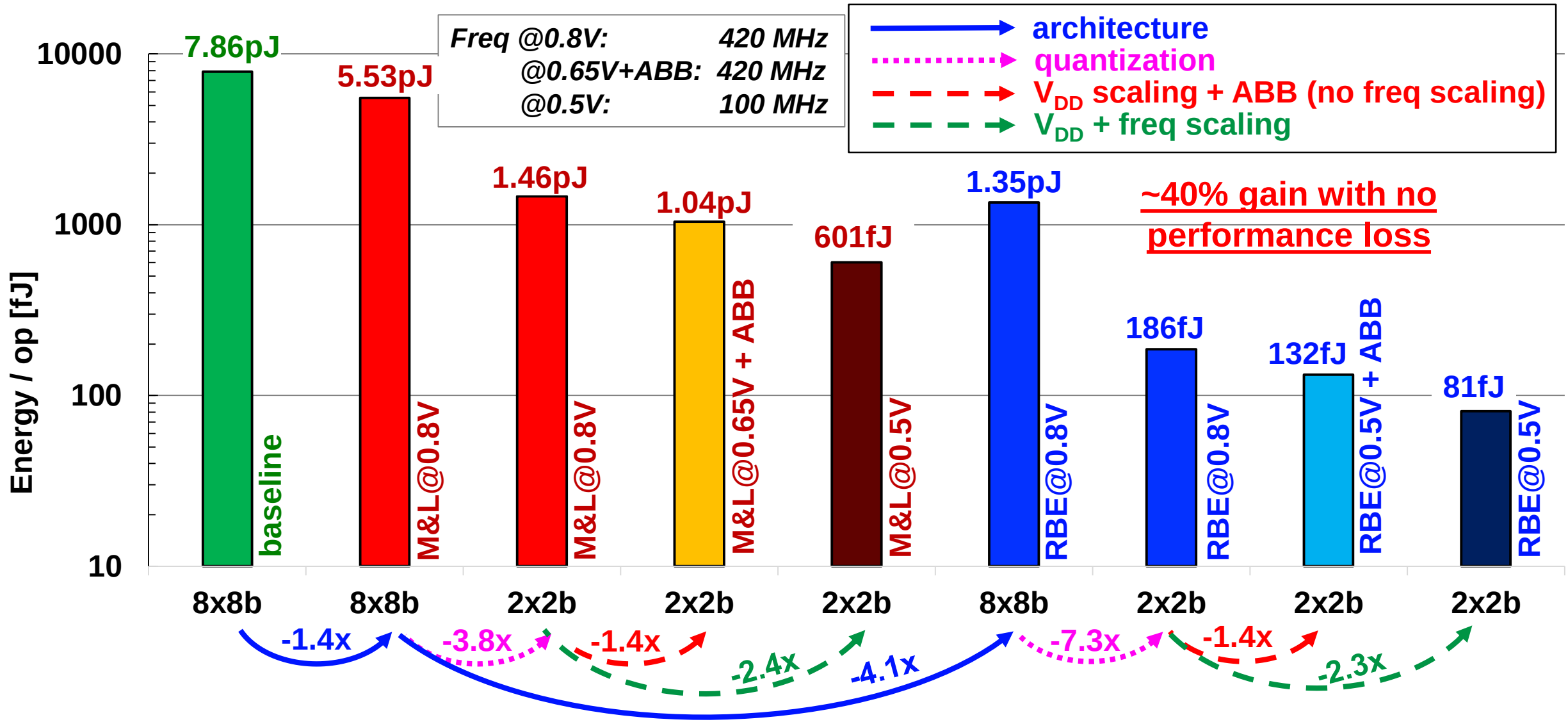
Efficiency Techniques in MARSELLUS



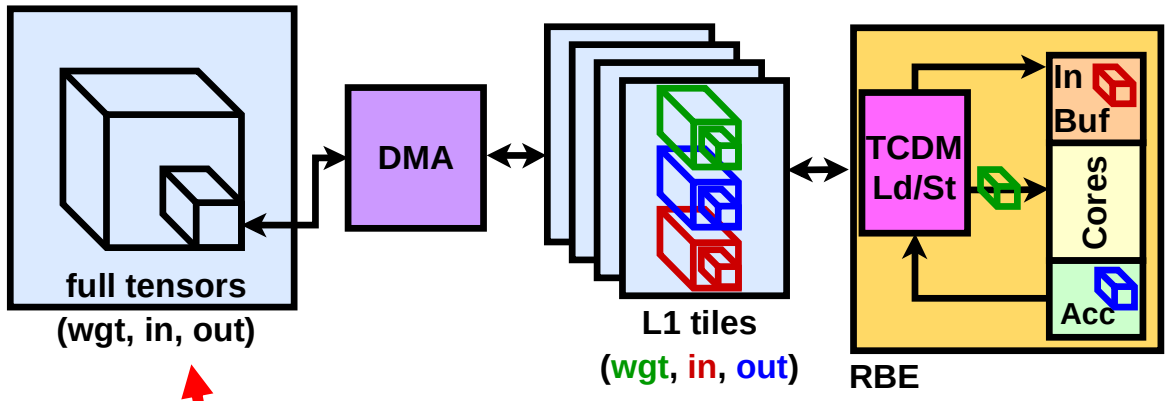
Efficiency Techniques in MARSELLUS



Efficiency Techniques in MARSELLUS



Full network: ResNet-20/CIFAR-10 on RBE

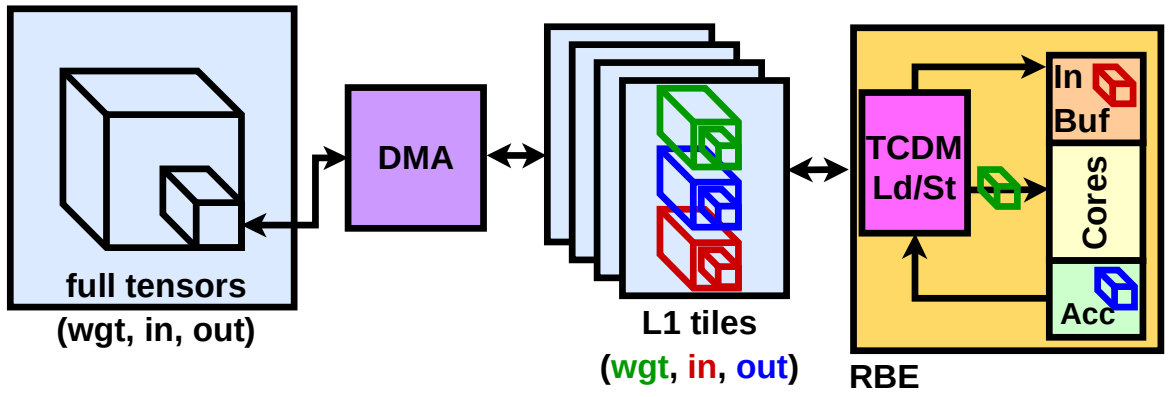


Full tensors resident on L2 (1 MB)

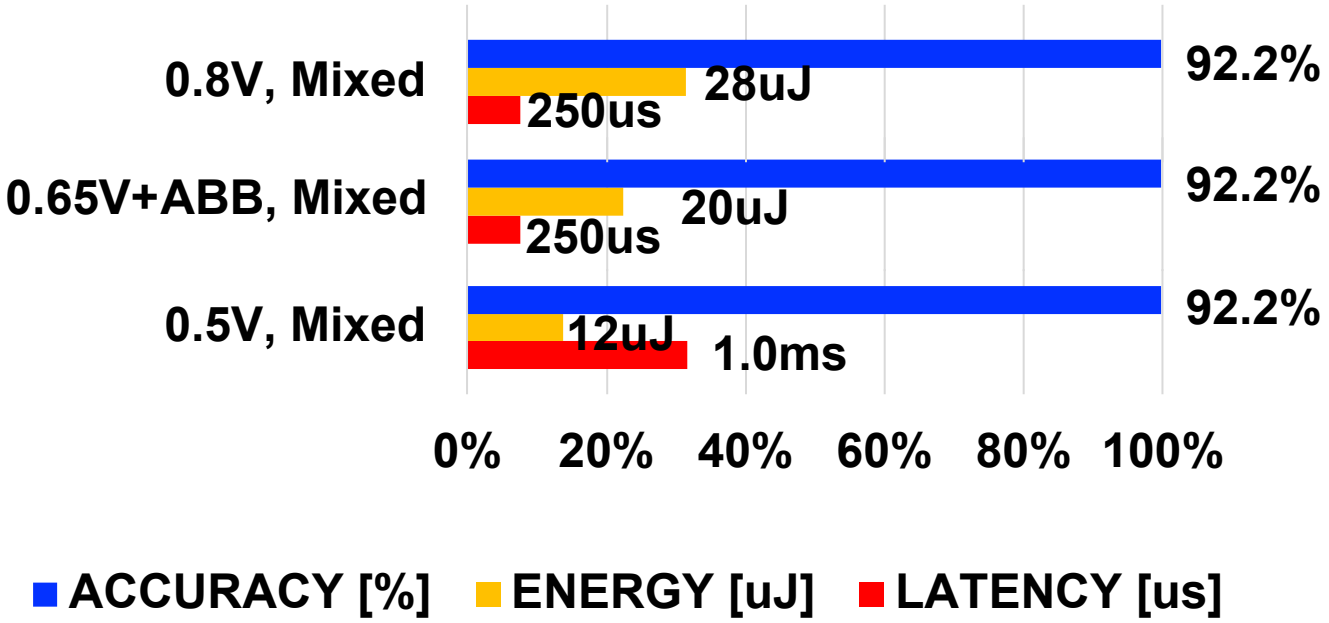
Tiling to fit into L1 TCDM (128 KB)

RBE accesses data in L1 TCDM and feeds internal buffers with smaller "subtiles" (working set)

Full network: ResNet-20/CIFAR-10 on RBE



- ResNet-20/CIFAR-10 quantized with HAWQ [5] @ mixed-precision (iso-accuracy)
  - Acc/Energy for CONV layers (no ADD)
  - In, Out, ADD quantized to 8-bits
- Can run at up to 4000fps @ 0.65V+ABB
  - 20 uJ/frame
- Or at 1000fps @ 0.5V
  - 12 uJ/frame



State-of-the-Art Comparison

	VEGA [1] ISSCC'21	SAMURAI [2] VLSIC'20	DIANA [3] ISSCC'22	QNAP [4] ISSCC'21	*MARSELLUS (this work)*
Technology	22nm FDX	28nm FD-SOI	22nm FDX + AIMC	28nm	22nm FDX
Die Area	10 mm^2	4.5 mm^2	10.24 mm^2	1.9 mm^2	18.7 mm^2 (cluster 1.9 mm^2)
Cores	10x RV32IMCFXpulp + HWCE	1x RV32IMCFXpulp + Digital Accel.	1x RV32CIMFXpulp + Digital Accel. + AIMC	Digital Accel.	1x RV32IMCFXpulp + 16x RV32IMCFXpulpnn + RBE
Max Frequency	450 MHz	350 MHz	320 MHz	470 MHz	420 MHz
Power range	1.7 uW - 49.4 mW	6.4 uW - 96 mW	10-129 mW (digital)	19.4-131 mW	12.8 mW - 123 mW
Best SW (INT) Perf	15.6 GOPS (8 RISC-V)	1.5 GOPS (1 RISC-V)	-	-	**90 GOPS (16 RISC-V M&L 2x2b, 0.8V+ABB)**
Best SW (INT) Eff	614 GOPS/W @ 7.6 GOPS (8 RISC-V)	230 GOPS/W @ 110 MOPS (1 RISC-V)	-	-	**1.66 TOPS/W @ 19 GOPS (16 RISC-V M&L 2x2b)**
Best HW-Accel Perf	32.2 GOPS (HWCE)	36 GOPS (Digital Accel.)	**180 GOPS (Digital), 29.5 TOPS (AIMC)**	140 GOPS (Digital Accel.)	**637 GOPS (RBE 2x2b, 0.8V+ABB)**
Best HW-Accel Eff	1.3 TOPS/W @ 15.6 GOPS (HWCE)	1.3 TOPS/W @ 2.8 GOPS (Digital Accel.)	**4.1 TOPS/W (Digital), 600 TOPS/W (AIMC)**	12.1 TOPS/W @ 140 GOPS (Digital Accel.)	**12.4 TOPS/W @ 136 GOPS (RBE 2x2b)**

State-of-the-Art Comparison

	VEGA [1] ISSCC'21	SAMURAI [2] VLSIC'20	DIANA [3] ISSCC'22	QNAP [4] ISSCC'21	*MARSELLUS (this work)*
Technology	22nm FDX	28nm FD-SOI	22nm FDX + AIMC	28nm	22nm FDX
Die Area	10 mm^2	4.5 mm^2	10.24 mm^2	1.9 mm^2	18.7 mm^2 (cluster 1.9 mm^2)
Cores	10x RV32IMCFXpulp + HWCE	1x RV32IMCFXpulp + Digital Accel.	1x RV32CIMFXpulp + Digital Accel. + AIMC	Digital Accel.	1x RV32IMCFXpulp + 16x RV32IMCFXpulpnn + RBE
Max Frequency	450 MHz	350 MHz	320 MHz	470 MHz	420 MHz
Power range	1.7 uW - 49.4 mW	6.4 uW - 96 mW	10-129 mW (digital)	19.4-131 mW	12.8 mW - 123 mW
Best SW (INT) Perf	15.6 GOPS (8 RISC-V)	1.5 GOPS (1 RISC-V)		-	**90 GOPS (16 RISC-V M&L 2x2b, 0.8V+ABB)**
Best SW (INT) Eff	614 GOPS/W @ 7.6 GOPS (8 RISC-V)	230 GOPS/W @ 110 MOPS (1 RISC-V)		-	**1.66 TOPS/W @ 19 GOPS (16 RISC-V M&L 2x2b)**
Best HW-Accel Perf	32.2 GOPS (HWCE)	36 GOPS (Digital Accel.)	**180 GOPS (Digital), 29.5 TOPS (AIMC)**	140 GOPS (Digital Accel.)	**637 GOPS (RBE 2x2b, 0.8V+ABB)**
Best HW-Accel Eff	1.3 TOPS/W @ 15.6 GOPS (HWCE)	1.3 TOPS/W @ 2.8 GOPS (Digital Accel.)	**4.1 TOPS/W (Digital), 600 TOPS/W (AIMC)**	12.1 TOPS/W @ 140 GOPS (Digital Accel.)	**12.4 TOPS/W @ 136 GOPS (RBE 2x2b)**

Machine Learning SW Perf & Efficiency boost of 5.8x & 2.7x compared to VEGA

State-of-the-Art Comparison

	VEGA [1] ISSCC'21	SAMURAI [2] VLSIC'20	DIANA [3] ISSCC'22	QNAP [4] ISSCC'21	*MARSELLUS (this work)*
Technology	22nm FDX	28nm FD-SOI	22nm FDX + AIMC	28nm	22nm FDX
Die Area	10 mm^2	4.5 mm^2	10.24 mm^2	1.9 mm^2	18.7 mm^2 (cluster 1.9 mm^2)
Cores	10x RV32IMCFXpulp + HWCE	1x RV32IMCFXpulp + Digital Accel.	1x RV32CIMFXpulp + Digital Accel. + AIMC	Digital Accel.	1x RV32IMCFXpulp + 16x RV32IMCFXpulpnn + RBE
Max Frequency	450 MHz	350 MHz	320 MHz	470 MHz	420 MHz
Power range	1.7 uW - 49.4 mW	6.4 uW - 96 mW	10-129 mW (digital)	19.4-131 mW	12.8 mW - 123 mW
Best SW (INT) Perf	15.6 GOPS (8 RISC-V)	1.5 GOPS (1 RISC-V)	-		**90 GOPS (16 RISC-V M&L 2x2b, 0.8V+ABB)**
Best SW (INT) Eff	614 GOPS/W @ 7.6 GOPS (8 RISC-V)	230 GOPS/W @ 110 MOPS (1 RISC-V)	-		**1.66 TOPS/W @ 19 GOPS (16 RISC-V M&L 2x2b)**
Best HW-Accel Perf	32.2 GOPS (HWCE)	36 GOPS (Digital Accel.)	**180 GOPS (Digital), 29.5 TOPS (AIMC)**	140 GOPS (Digital Accel.)	**637 GOPS (RBE 2x2b, 0.8V+ABB)**
Best HW-Accel Eff	1.3 TOPS/W @ 15.6 GOPS (HWCE)	1.3 TOPS/W @ 2.8 GOPS (Digital Accel.)	**4.1 TOPS/W (Digital), 600 TOPS/W (AIMC)**	12.1 TOPS/W @ 140 GOPS (Digital Accel.)	**12.4 TOPS/W @ 136 GOPS (RBE 2x2b)**

DNN Acceleration
Perf & Efficiency boost
3.5x & 3.0x
compared to DIANA (Digital)

State-of-the-Art Comparison (2)

	VEGA [1] ISSCC'21	SAMURAI [2] VLSIC'20	DIANA [3] ISSCC'22	QNAP [4] ISSCC'21	*MARSELLUS (this work)*
Best ResNet-20 Eff (CIFAR-10)	-	-	**14.4 TOPS/W (AIMC)**	-	**6.38 TOPS/W (RBE mixed)**
ResNet-20 Latency @ Best Eff	-	-	**1.26 ms**	-	**1.05 ms**
Best ResNet-18 Eff (ImageNet)	-	-	**19 TOPS/W**	12.62 TOPS/W (zero-skipping)	5.83 TOPS/W (RBE 4x4b)
ResNet-18 Latency @ Best Eff	-	-	**6.15ms**	24.8ms	48ms

State-of-the-Art Comparison (2)

	VEGA [1] ISSCC'21	SAMURAI [2] VLSIC'20	DIANA [3] ISSCC'22	QNAP [4] ISSCC'21	*MARSELLUS (this work)*
Best ResNet-20 Eff (CIFAR-10)	-	-	**14.4 TOPS/W (AIMC)**	-	**6.38 TOPS/W (RBE mixed)**
ResNet-20 Latency @ Best Eff	-	-	**1.26 ms**	-	**1.05 ms**
Best ResNet-18 Eff (ImageNet)	-	-	**19 TOPS/W**	12.62 TOPS/W (zero-skipping)	5.83 TOPS/W (RBE 4x4b)
ResNet-18 Latency @ Best Eff	-	-	**6.15ms**	24.8ms	48ms

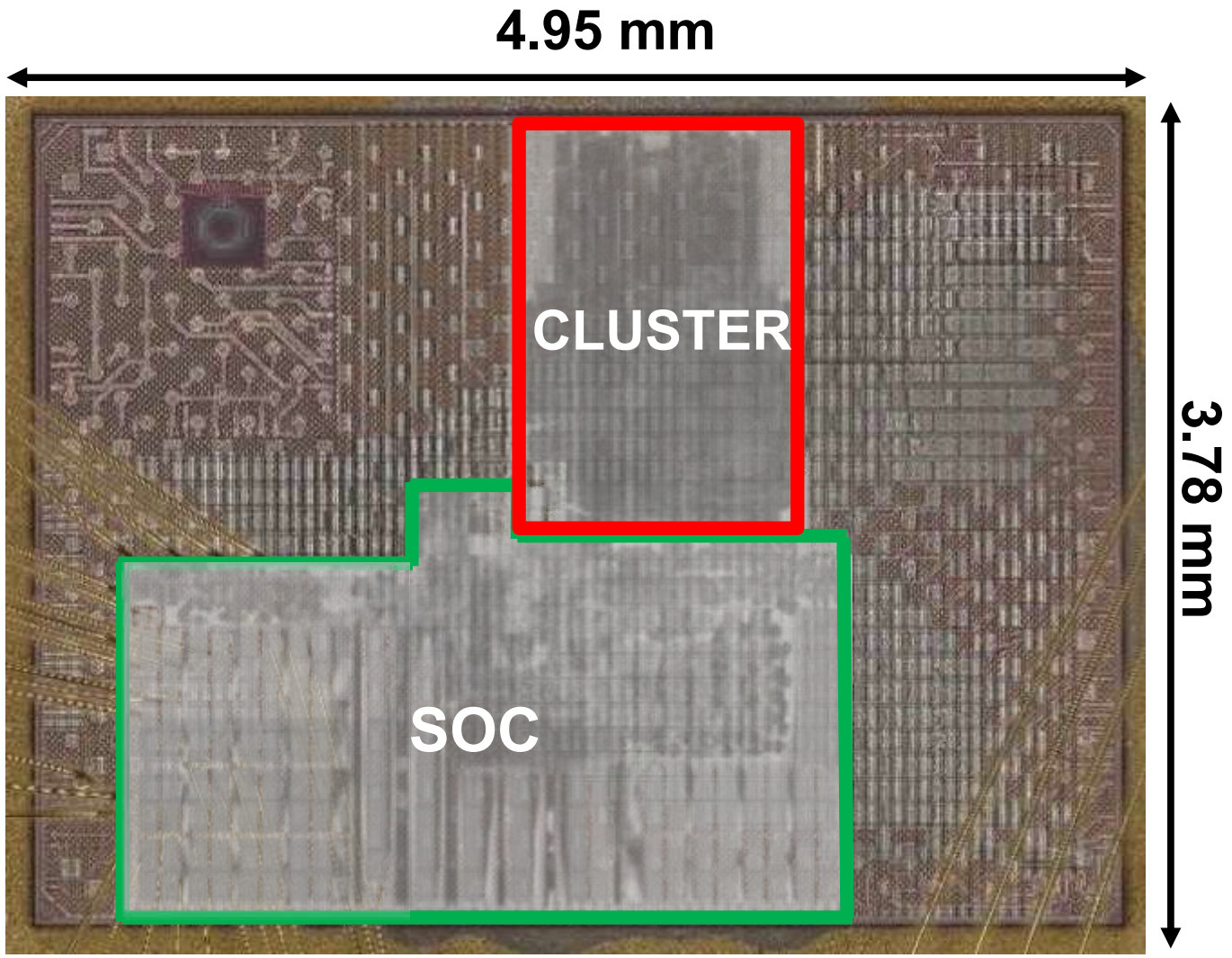
- **ResNet-20/CIFAR10: digital MARSELLUS vs mixed-signal AIMC-based accelerator DIANA**
  - **~40x lower performance and efficiency at peak, but just 2.3x worse efficiency at 20% better latency**
  - **Marginal energy advantage might be further neglected in full system**

State-of-the-Art Comparison (2)

	VEGA [1] ISSCC'21	SAMURAI [2] VLSIC'20	DIANA [3] ISSCC'22	QNAP [4] ISSCC'21	*MARSELLUS (this work)*
Best ResNet-20 Eff (CIFAR-10)	-	-	14.4 TOPS/W (AIMC)	-	6.38 TOPS/W (RBE mixed)
ResNet-20 Latency @ Best Eff	-	-	1.26 ms	-	1.05 ms
Best ResNet-18 Eff (ImageNet)	-	-	19 TOPS/W	12.62 TOPS/W (zero-skipping)	5.83 TOPS/W (RBE 4x4b)
ResNet-18 Latency @ Best Eff	-	-	6.15ms	24.8ms	48ms

- ResNet-20/CIFAR10: digital MARSELLUS vs mixed-signal AIMC-based accelerator DIANA
  - ~40x lower performance and efficiency at peak, but just 2.3x worse efficiency at 20% better latency
  - Marginal energy advantage might be further neglected in full system
- MARSELLUS shows competitive acceleration even compared with AIMC for TinyML and AI-IoT endpoint devices
  - Up to 5.83 TOPS/W on 4-bit ImageNet @ 20fps without AIMC, zero-skipping, sparsity

Conclusion



- **MARSELLUS combines**
  - state-of-the-art parallel and heterogeneous acceleration
  - aggressive performance and voltage scalability thanks to ABB
- **MARSELLUS achieves**
  - 1000x better performance and efficiency than commercial AI-IoT microcontrollers with similar power budget (~130mW)

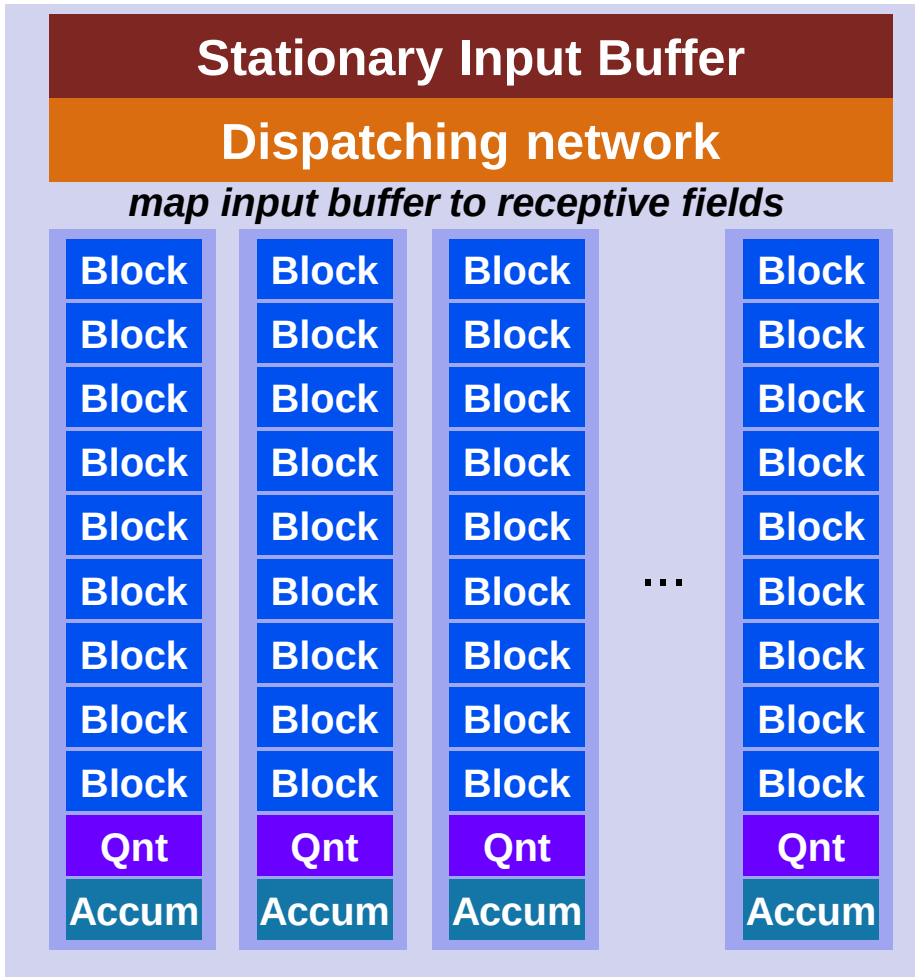
Thanks for the attention.

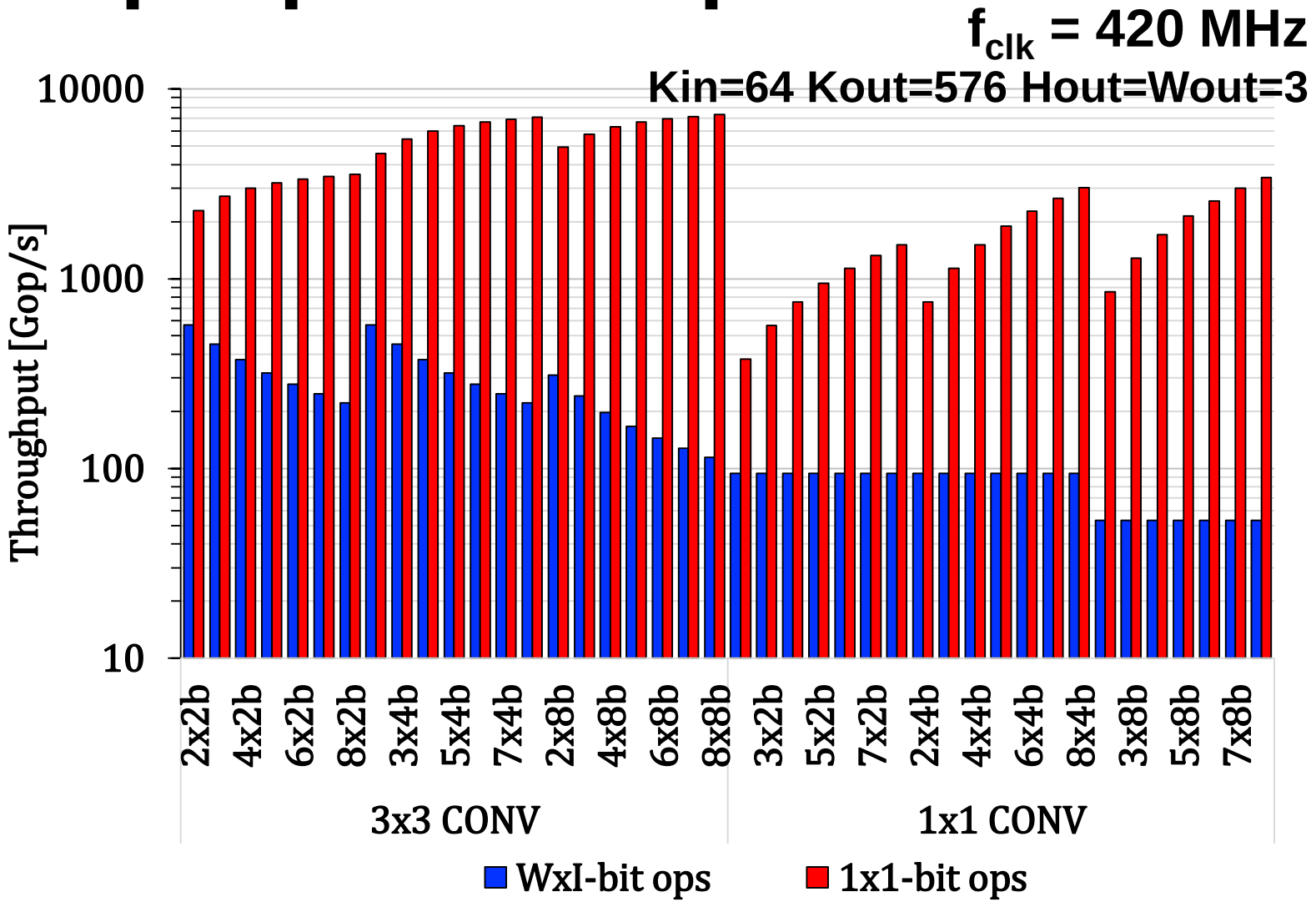
Questions?

References

- [1] D. Rossi et al., “A 1.3TOPS/W @ 32GOPS Fully Integrated 10-Core SoC for IoT End- Nodes with 1.7µW Cognitive Wake-Up From MRAM-Based State-Retentive Sleep Mode,” *ISSCC*, pp. 60-62, 2021.
- [2] I. Miro-Panades et al., “SamurAI: A 1.7MOPS-36GOPS Adaptive Versatile IoT Node with 15,000× Peak-to-Idle Power Reduction, 207ns Wake-Up Time and 1.3TOPS/W ML Efficiency,” *IEEE Symp. VLSI Circuits*, 2020.
- [3] K. Ueyoshi et al., “DIANA: An End-to-End Energy-Efficient Digital and ANAlog Hybrid Neural Network SoC,” *ISSCC*, pp. 256-257, 2022.
- [4] H. Mo et al., “A 28nm 12.1TOPS/W Dual-Mode CNN Processor Using Effective- Weight-Based Convolution and Error-Compensation-Based Prediction,” *ISSCC*, pp. 146-148, 2021.
- [5] Z. Dong et al., “HAWQ: Hessian AWare Quantization of Neural Networks With Mixed-Precision,” *ICCV*, pp. 293-302, 2019.

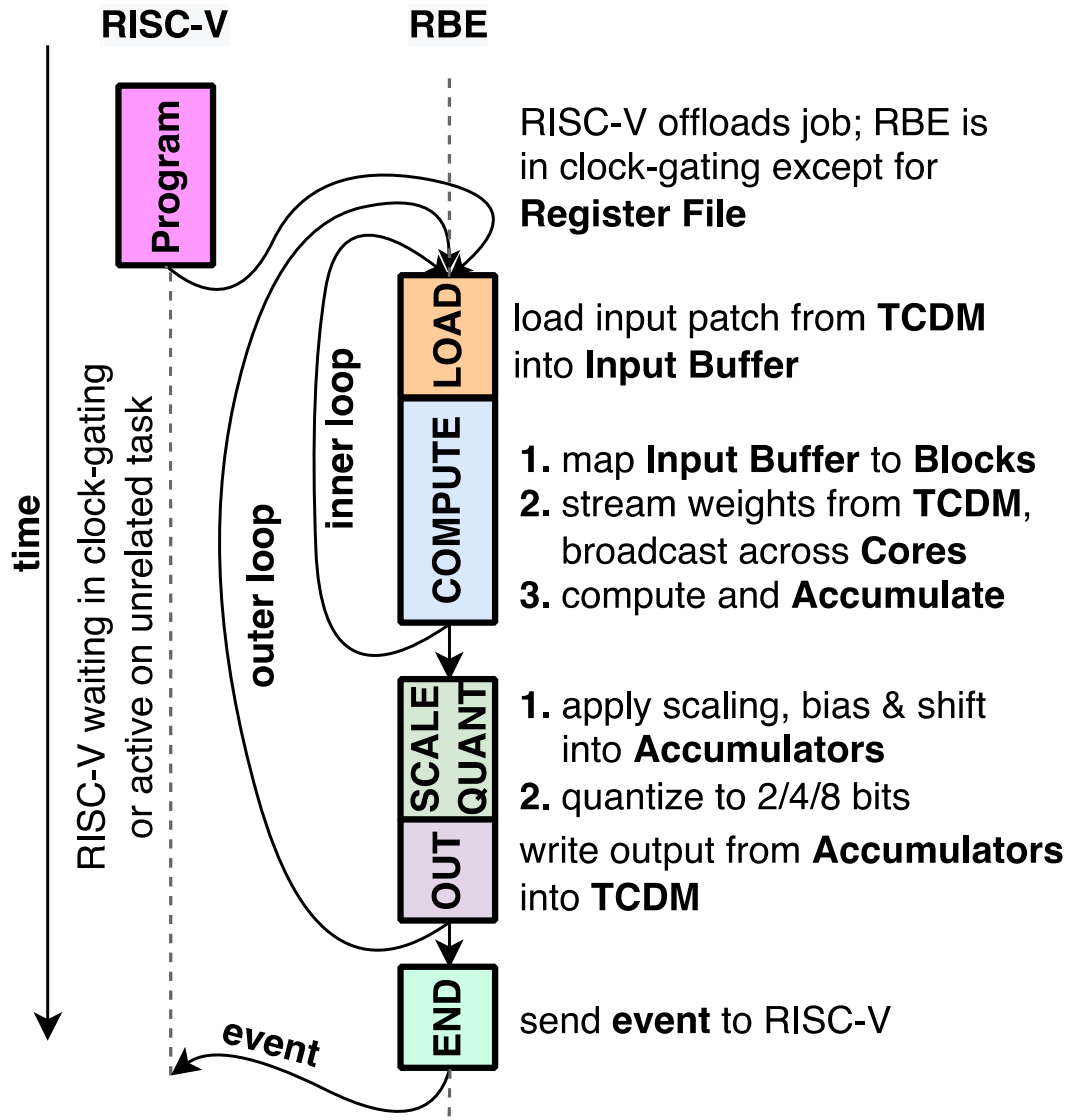
Appendix: RBE multiple-precision performance



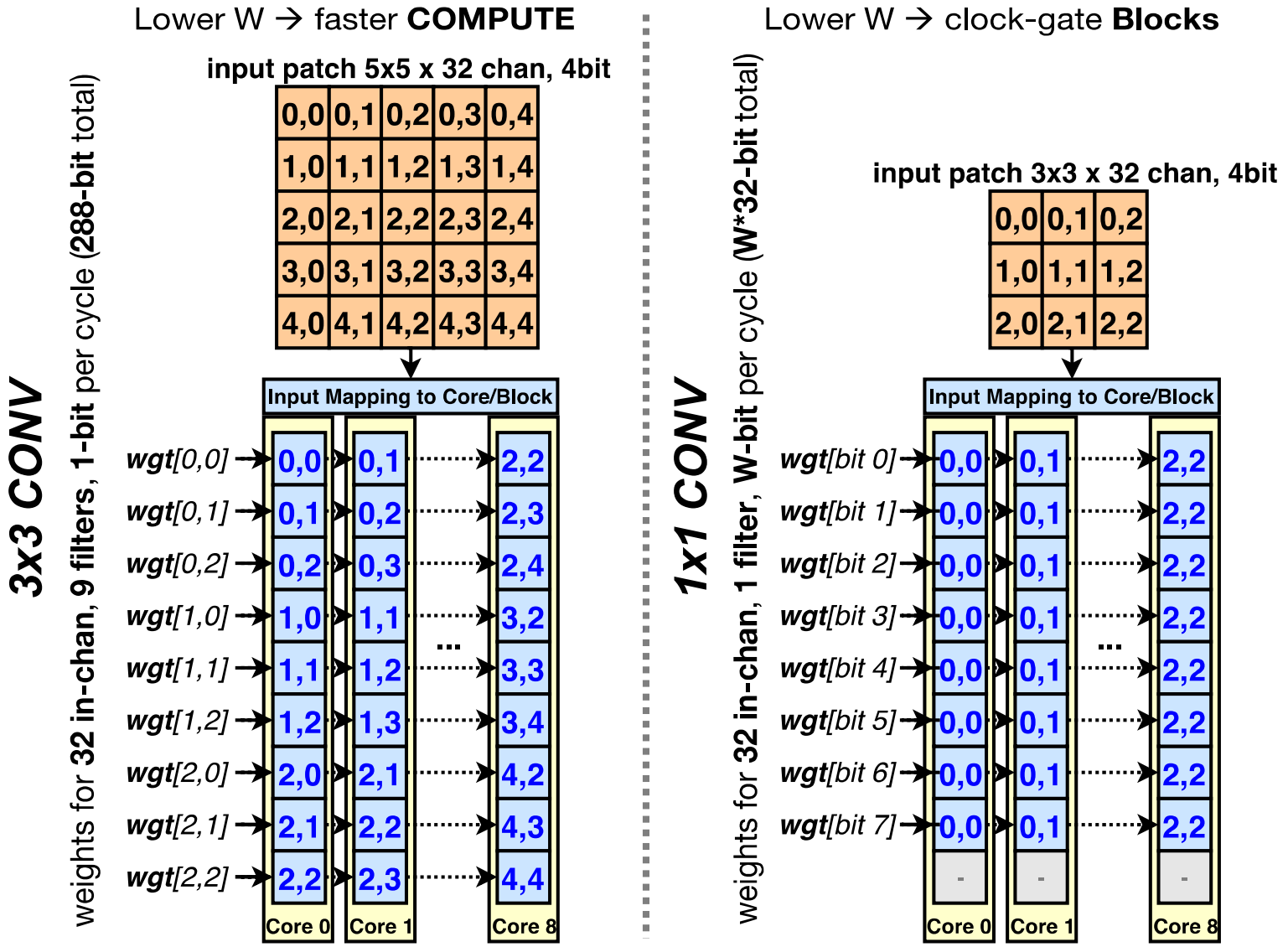


***all* WxI-b combinations are possible!**

Appendix: RBE Execution flow



Appendix: RBE CONV mapping



Appendix: RBE Dataflow loop nests

LOAD

COMPUTE

3x3 CONV

```
for(h_in=0 to 5) :
  for(bit_in=0 to I) :
    par_for(w_in=0 to 5)  : // due to Activation layout
    par_for(k_in=0 to 32) : // due to Activation layout
      load(in_buffer[h_in, bit_in, w_in, k_in])
      inp = map(in_buffer)
for(k_out=0 to 32) :
  for(bit_wgt=0 to W) :
    par_for(h_out=0 to 3)  : // over Cores
    par_for(w_out=0 to 3)  : // over Cores
    par_for(f=0 to 9)      : // over rows of Blocks
    par_for(bit_in=0 to I) : // over BinConvs
    par_for(k_in=0 to 32)  : // inside BinConv
      load(wgt[k_out, k_in_tiles, bit_wgt, f, k_in])
      acc[h_out, w_out, k_out] += dot(wgt, inp)
```

1x1 CONV

```
for(h_in=0 to 3) :
  for(bit_in=0 to I) :
    par_for(w_in=0 to 3)  : // due to Activation layout
    par_for(k_in=0 to 32) : // due to Activation layout
      load(in_buffer[h_in, bit_in, w_in, k_in])
      inp = map(in_buffer)
for(k_out=0 to 32) :
  par_for(h_out=0 to 3)   : // over Cores
  par_for(w_out=0 to 3)   : // over Cores
  par_for(bit_wgt=0 to W) : // over rows of Blocks
  par_for(bit_in=0 to I)  : // over BinConvs
  par_for(k_in=0 to 32)   : // inside BinConv
    load(wgt[k_out, k_in_tiles, bit_wgt, f, k_in])
    acc[h_out, w_out, k_out] += dot(wgt, inp)
```