ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

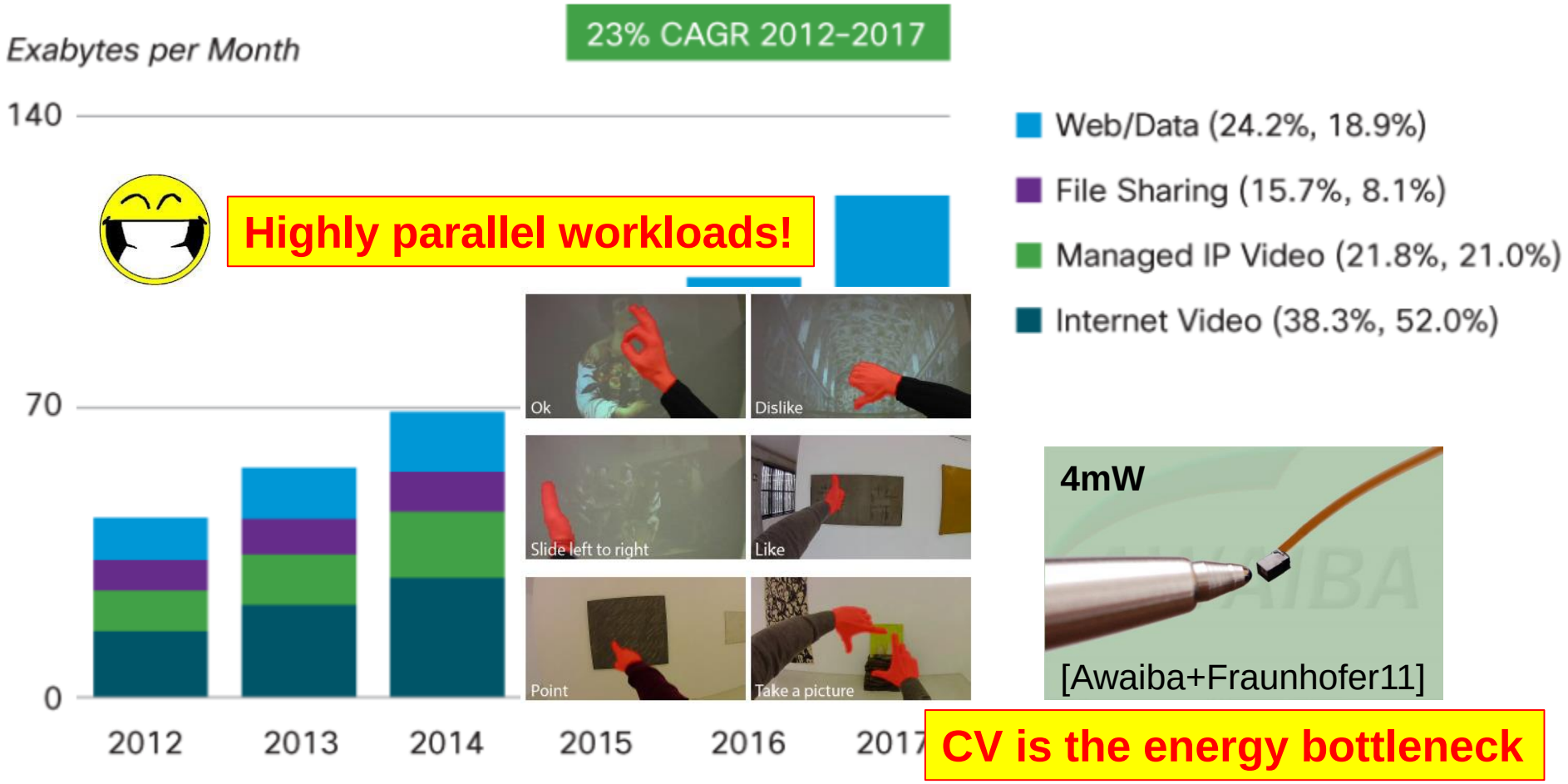
Parallel digital signal processing in a mW power envelope: how and why?

Davide Rossi

DEI-UNIBO

erc
European Research Council
Multithermand AdG
Multiscale Thermal Management of Computing Systems

IOT or Data Deluge?

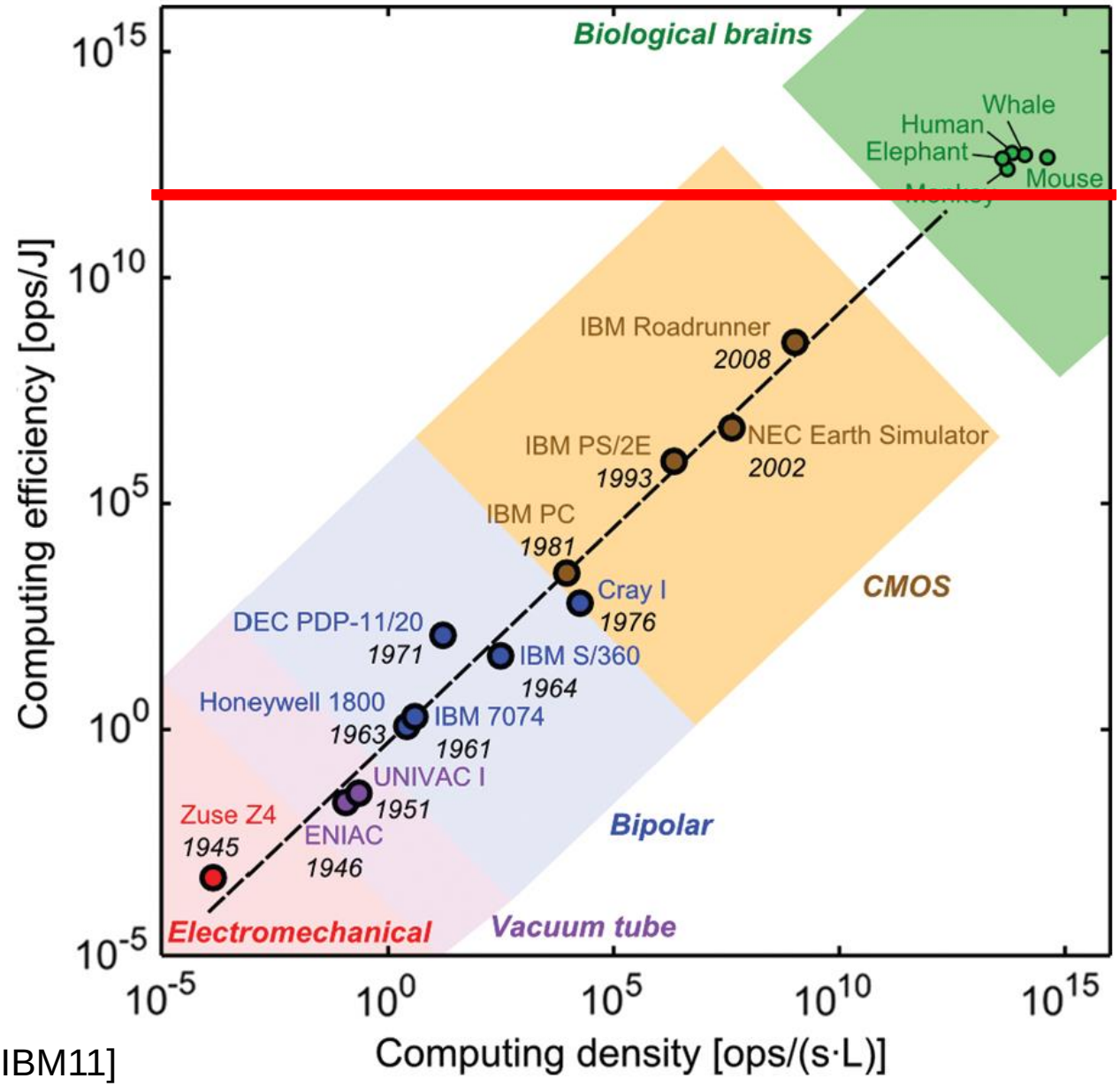


Source: Cisco VNI, 2013

The percentages within parenthesis next to the legend denote the relative traffic shares in 2012 and 2017.

In-situ stream processing & fusion + visual intelligence is a must!
- In a few mW power envelope!!

How efficient?



10^{12}ops/J

↓

1pJ/op

↓

1GOPS/mW

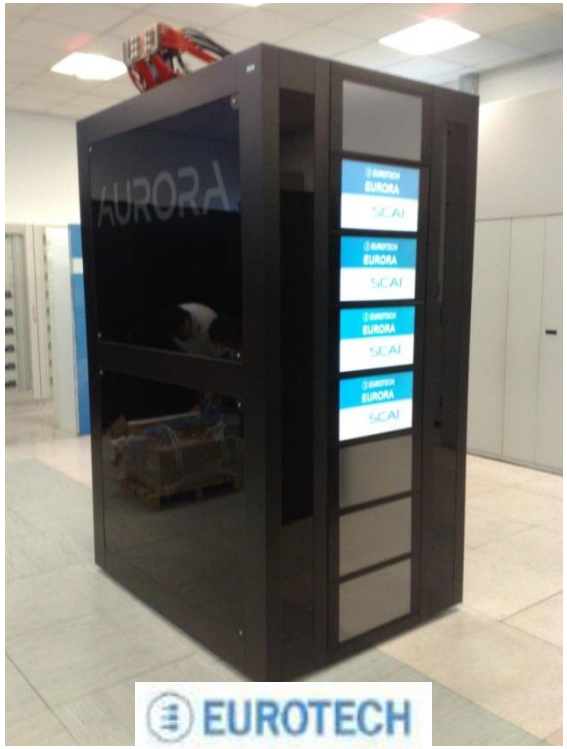
[RuchIBM11]

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The challenge of Energy Proportionality

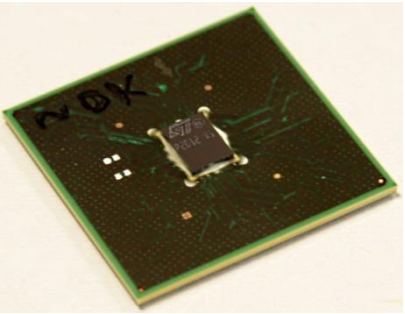
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

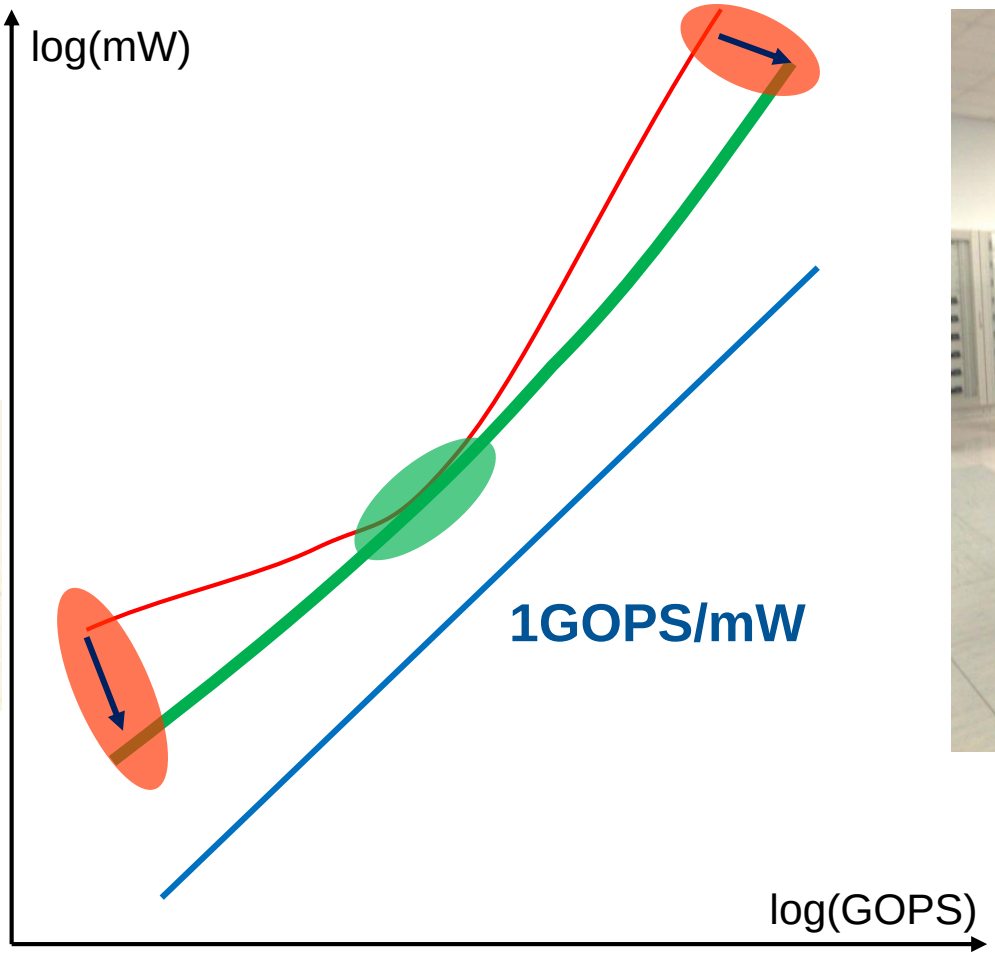
0,003GOPS/mW – 30KW



0,04GOPS/mW

life.augmented





From KOPS (10^3) to EOPS (10^{18})!

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

A Very Short Review on CMOS power
and power minimization

Summary of Power Dissipation Sources

$$P \sim \alpha \cdot (C_L + C_{CS}) \cdot V_{swing} \cdot V_{DD} \cdot f + (I_{DC} + I_{Leak}) \cdot V_{DD}$$

- α – switching activity
- C_L – load capacitance
- C_{CS} – short-circuit capacitance
- V_{swing} – voltage swing
- f – frequency
- I_{DC} – static current
- I_{leak} – leakage current

$$P = \frac{energy}{operation} \times rate + static\ \ power$$

The Traditional Design Philosophy

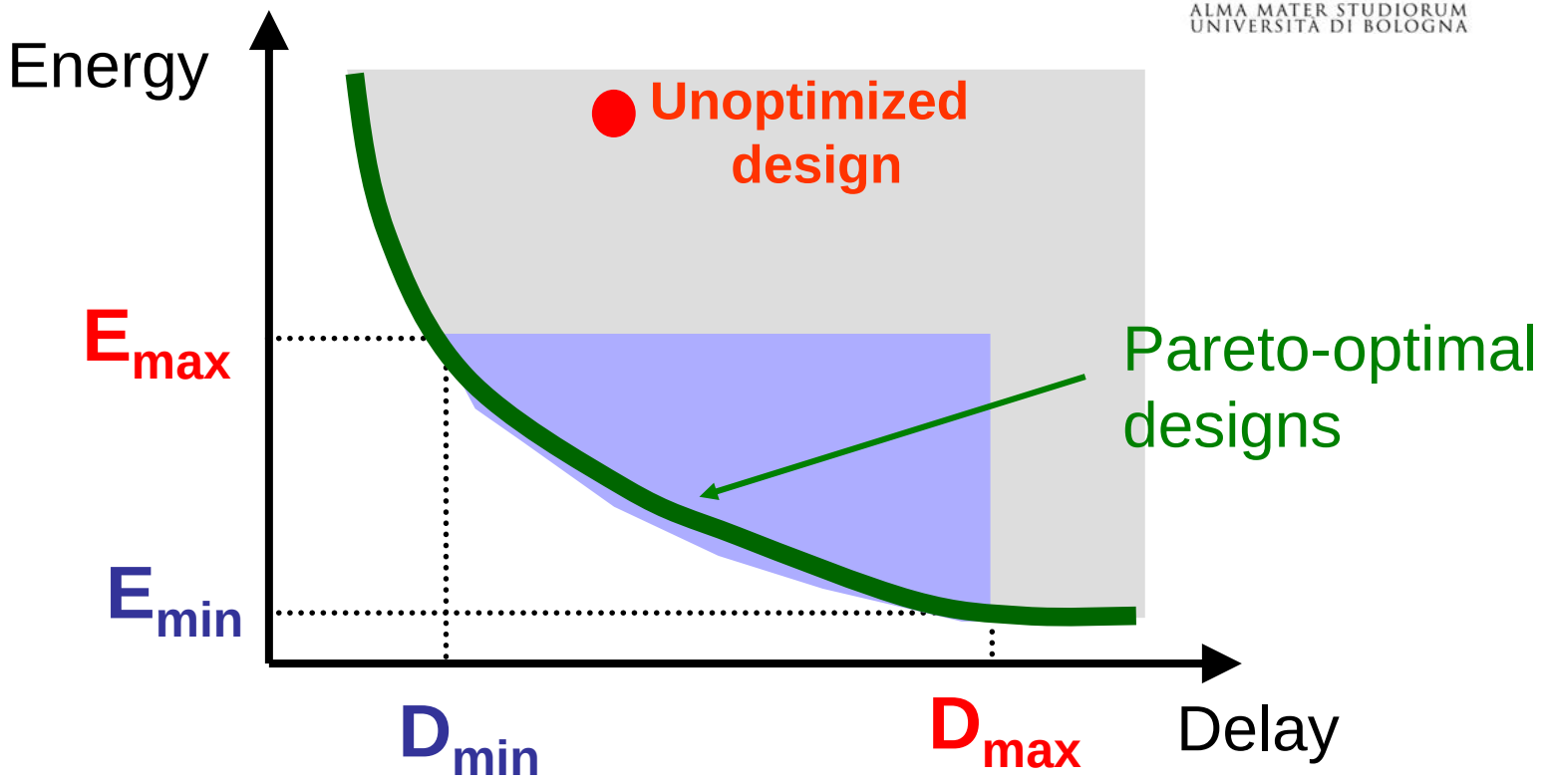
- Maximum performance is primary goal
  - Minimum delay at circuit level
- Architecture implements the required function with target throughput, latency
- Performance achieved through optimum sizing, logic mapping, architectural transformations.
- Supplies, thresholds set to achieve maximum performance, subject to reliability constraints

The New Design Philosophy

- Maximum performance (in terms of propagation delay) is too power-hungry, and/or not even practically achievable
- Many (if not most) applications either can tolerate larger latency, or can live with lower than maximum clock-speeds
- Excess performance (as offered by technology) to be used for energy/power reduction

Trading off speed for power

Exploring the Energy-Delay Space



In energy-constrained world, design is trade-off process

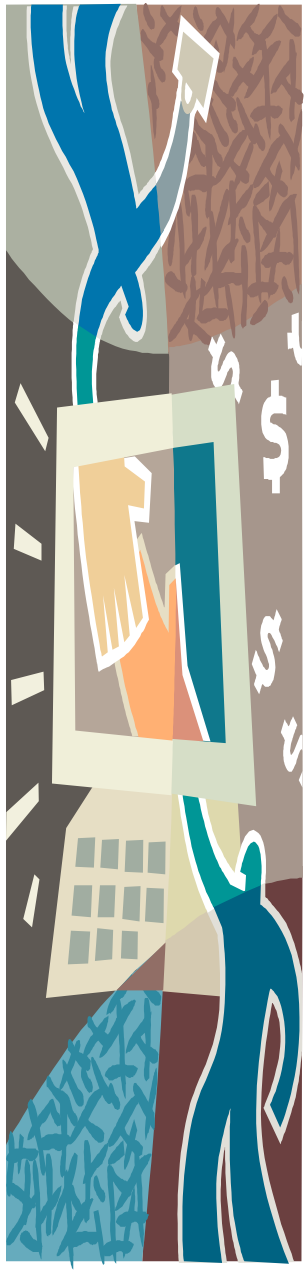
- Minimize energy for a given performance requirement
- Maximize performance for given energy budget

[Ref: D. Markovic, JSSC'04]

Reducing power @ all design levels

- Algoritmic level
- Compiler level
- Architecture level
- Organization level
- Circuit level
- Silicon level

- Important concepts:
  - Lower Vdd and freq. (even if errors occur) / dynamically adapt Vdd and freq.
  - Reduce circuit
  - Exploit locality
  - Reduce switching activity, glitches, etc.



Algoritmic level

- The best indicator for energy is …..
  …. the number of cycles

- Try alternative algorithms with lower complexity
  - E.g. quick-sort, O(n log n) $\Leftrightarrow$ bubble-sort, O (n^2)
  - … but be aware of the 'constant' : O(n log n) $\Rightarrow$ c*(n log n)

- Heuristic approach
  - Go for a good solution, not the best !!

Biggest gains at this level !!

Compiler level

- Source-to-Source transformations
  - loop trafo's to improve locality
- Strength reduction
  - E.g. replace Const * A with Add's and Shift's
  - Replace Floating point with Fixed point
- Reduce register pressure / number of accesses to register file
  - Use software bypassing
- Scenarios: current workloads are highly dynamic
  - Determine and predict execution modes
  - Group execution modes into scenarios
  - Perform special optimizations per scenario
    - DFVS: Dynamic Voltage and Frequency Scaling
    - More advanced loop optimizations
- Reorder instructions to reduce bit-transistions

Architecture level

- Going parallel
- Going heterogeneous
  - tune your architecture, exploit SFUs (special function units)
  - trade-off between flexibility / programmability / genericity and efficiency
- Add local memories
  - prefer scratchpad i.s.o. cache
- Cluster FUs and register files (see next slide)
- Reduce bit-width
  - sub-word parallelism (SIMD)

Organization (micro-arch.) level

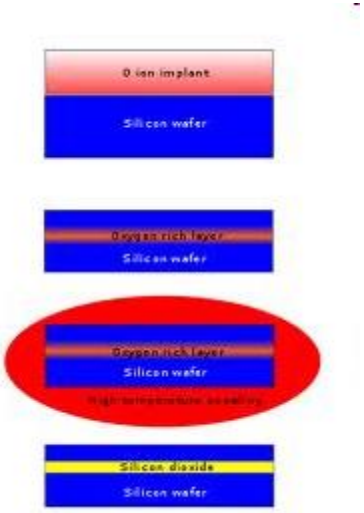
- Enabling Vdd reduction
  - Pipelining
    - cheap way of parallelism
  - Enabling lower freq. $\Rightarrow$ lower V_{dd}
  - Note 1: don't pipeline if you don't need the performance
  - Note 2: don't exaggerate (like the 31-stage Pentium 4)
- Reduce register traffic
  - avoid unnecessary reads and write
  - make bypass registers visible

Circuit level

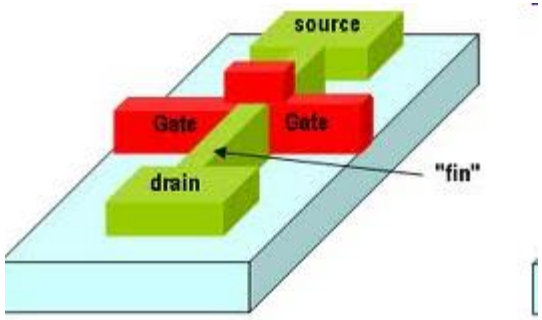
- Clock gating
- Power gating
- Multiple Vdd modes
- Reduce glitches: balancing digital path's
- Exploit Zeros
- Special SRAM cells
  - normal SRAM can not scale below Vdd = 0.7 - 0.8 Volt
- Razor method; replay
- Allow errors and add redundancy to architectural invisible structures
  - branch predictor
  - caches
- .. and many more ..

Silicon level

- Higher V_t (V_threshold)
- Back Biasing control
  - see thesis Maurice Meijer (2011)
- SOI (Silicon on Insulator)
  - silicon junction is above an electr. insulator (silicon dioxide)
  - lowers parasitic device capacitance



- Better transistors: Finfet
  - multi-gate
  - reduce leakage (off-state curent)



- .. and many more

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

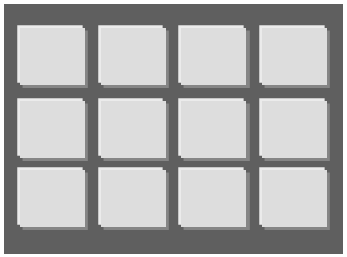
Two Ideas to Remember

…with their caveats

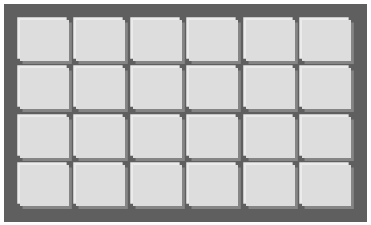
Go Simple+Parallel

13mm, **100W**, 48MB Cache, 4B Transistors, in 22nm

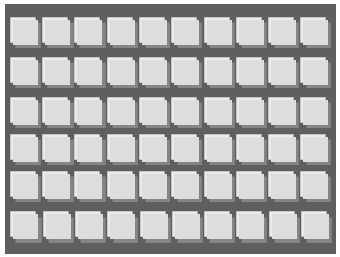
12 Cores



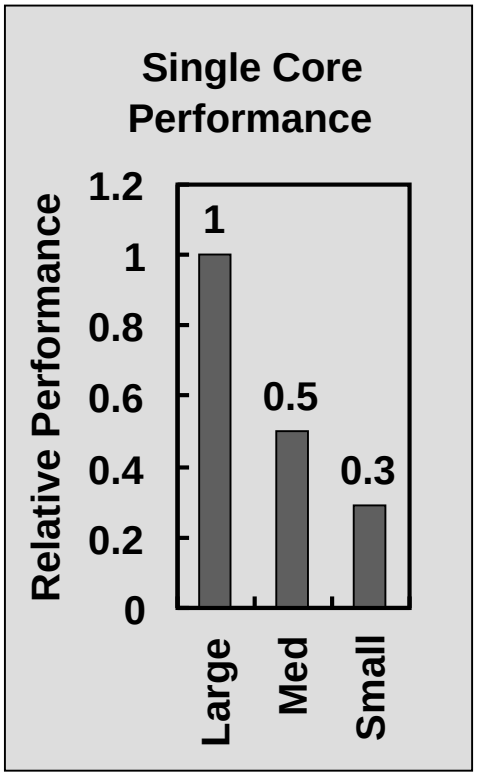
48 Cores

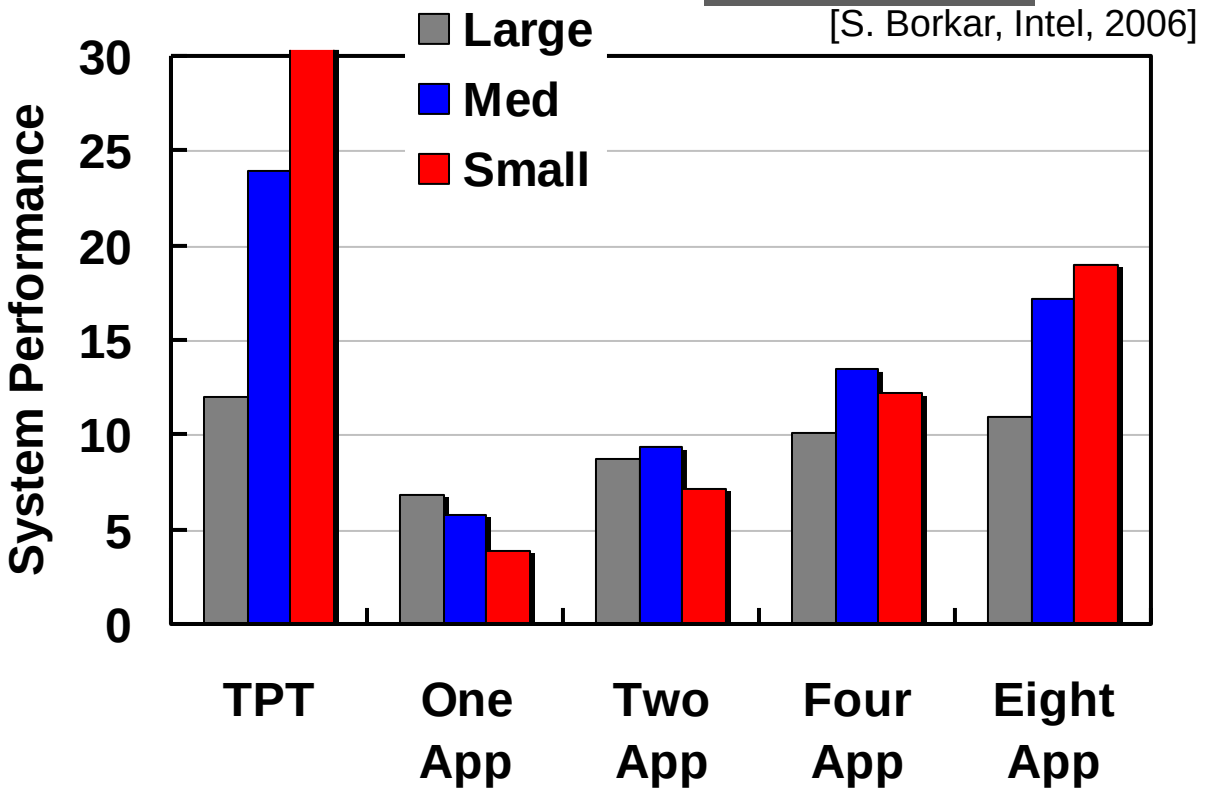


144 Cores



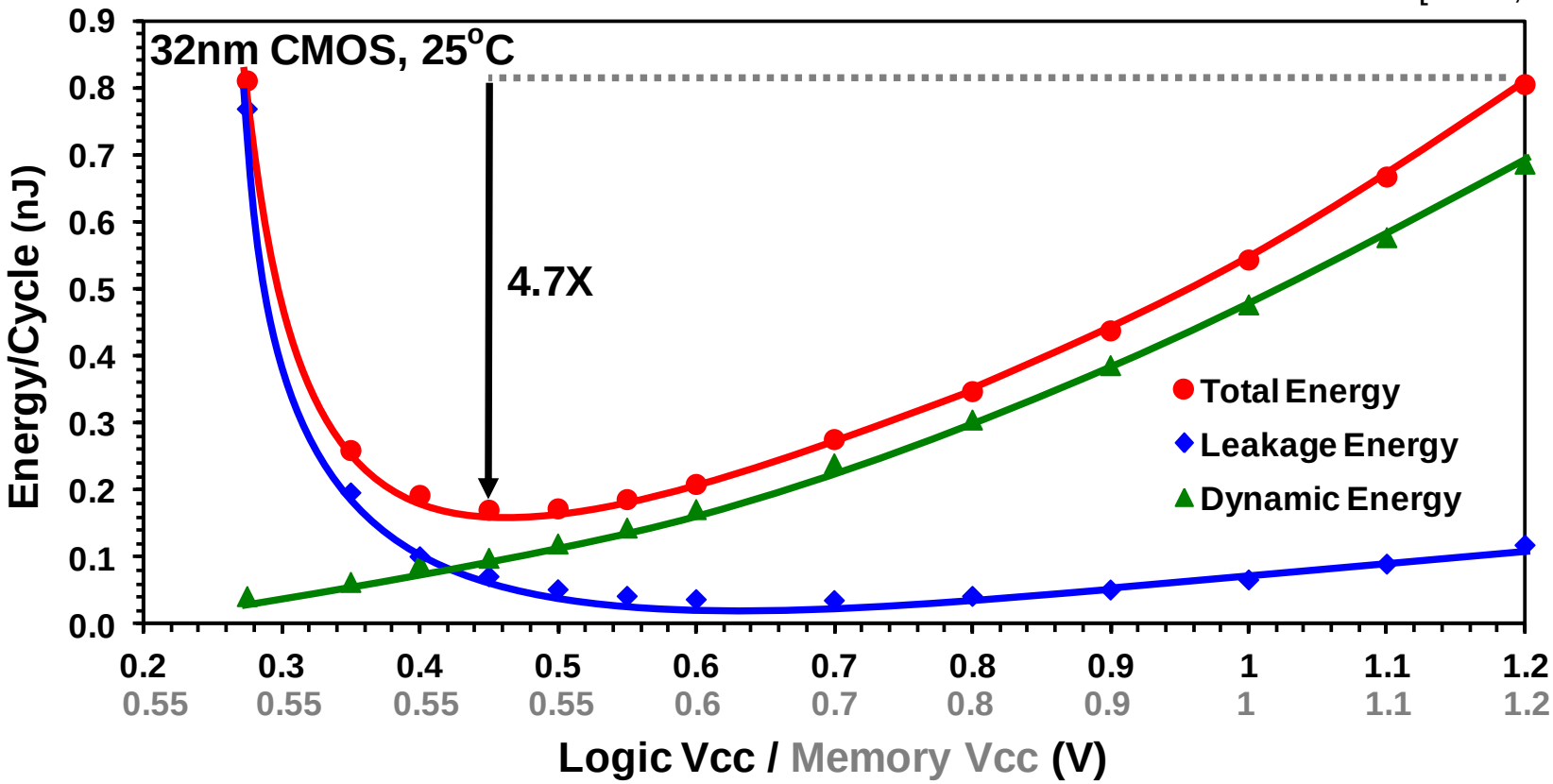
[S. Borkar, Intel, 2006]





Lower Voltage Supply

[V. De, Intel, 2013]



…but be careful with Leakage and its variability!

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

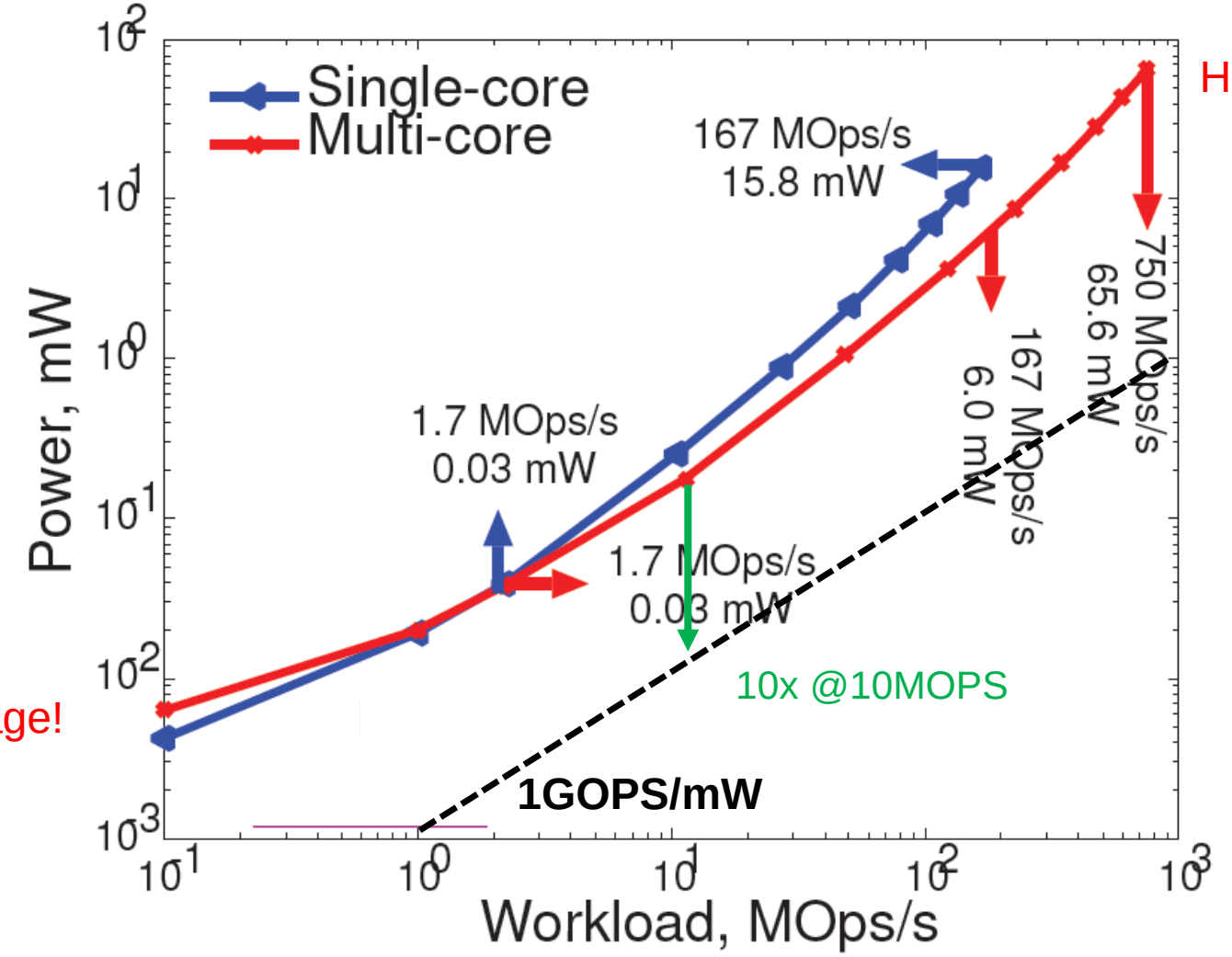
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Introducing PULP

NTC Multicore?

2010 - With EPFL (Atienza, Burg)

90nm LP-CMOS

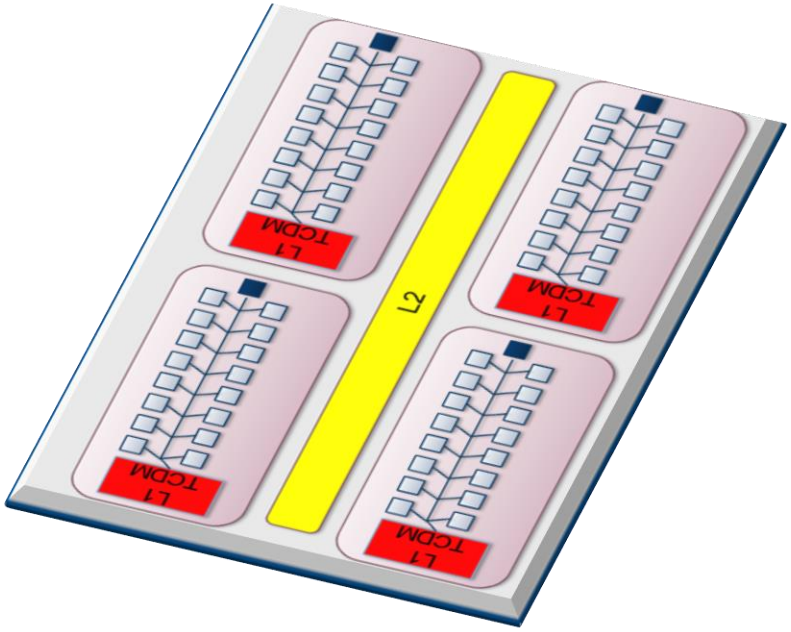


High Vdd

Leakage!

A pJ/OP Parallel ULP Computing Platform

- **pJ/OP** is traditionally* the target of ASIC + uCntr
  - ➡ Scalable: [KOPS,TOPS], 32bit architecture
  - ➡ Flexible: OpenMP, OpenVX
  - ➡ Open: Software & HW

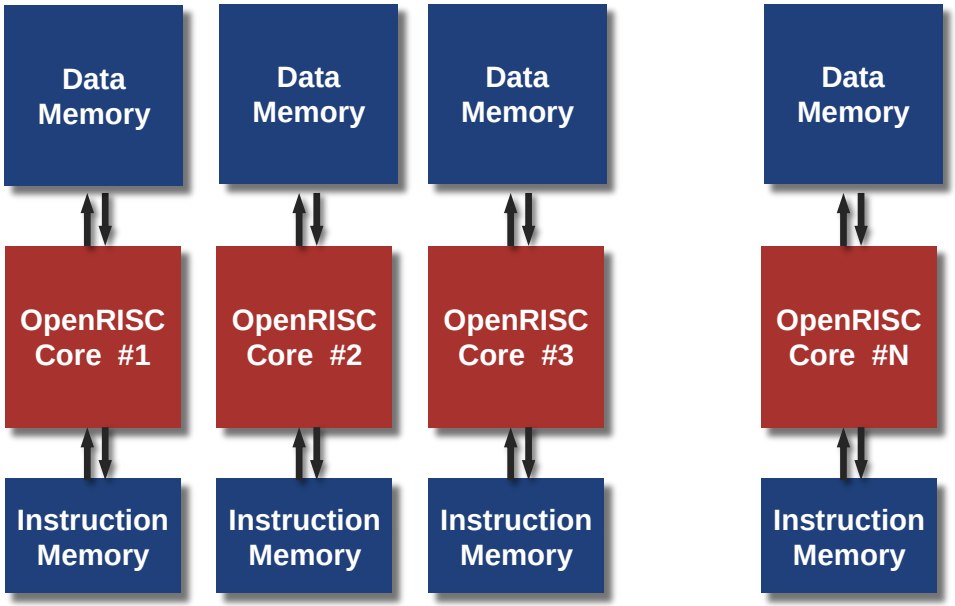


*1.57TOPS/W: Kim et al., "A 1.22TOPS and 1.52mW/MHz Augmented Reality Multi-core Processor with Neural Network NoC for HMD Applications", ISSCC 2014

Making PULP

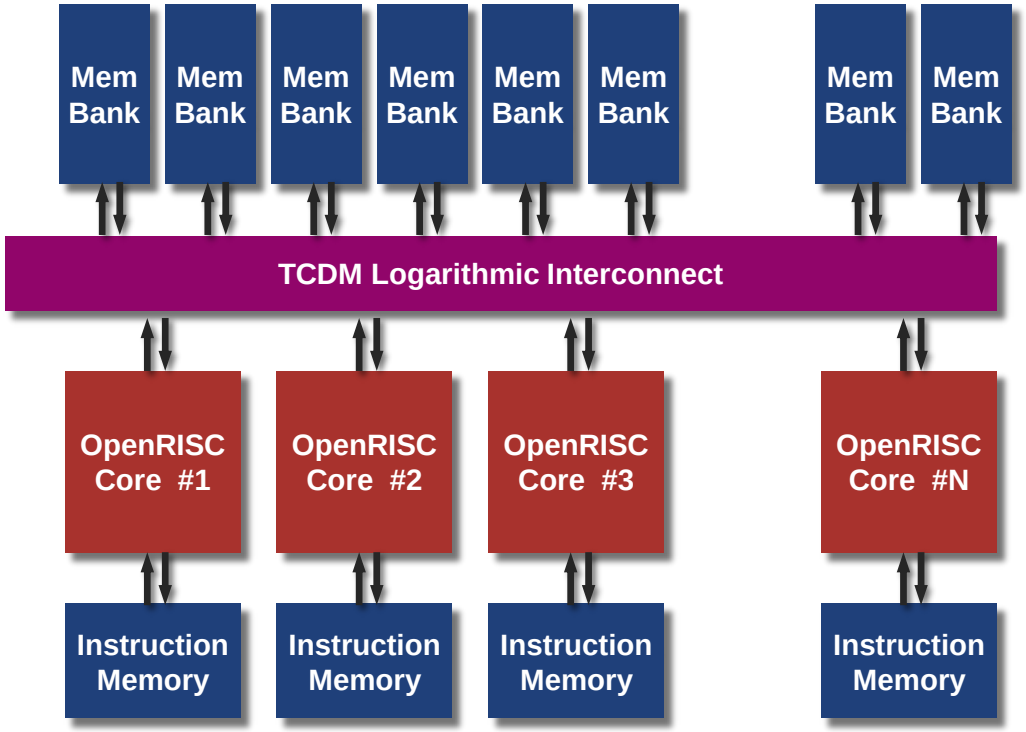
Start with an simple+efficient processor (~1IPC)

Making PULP



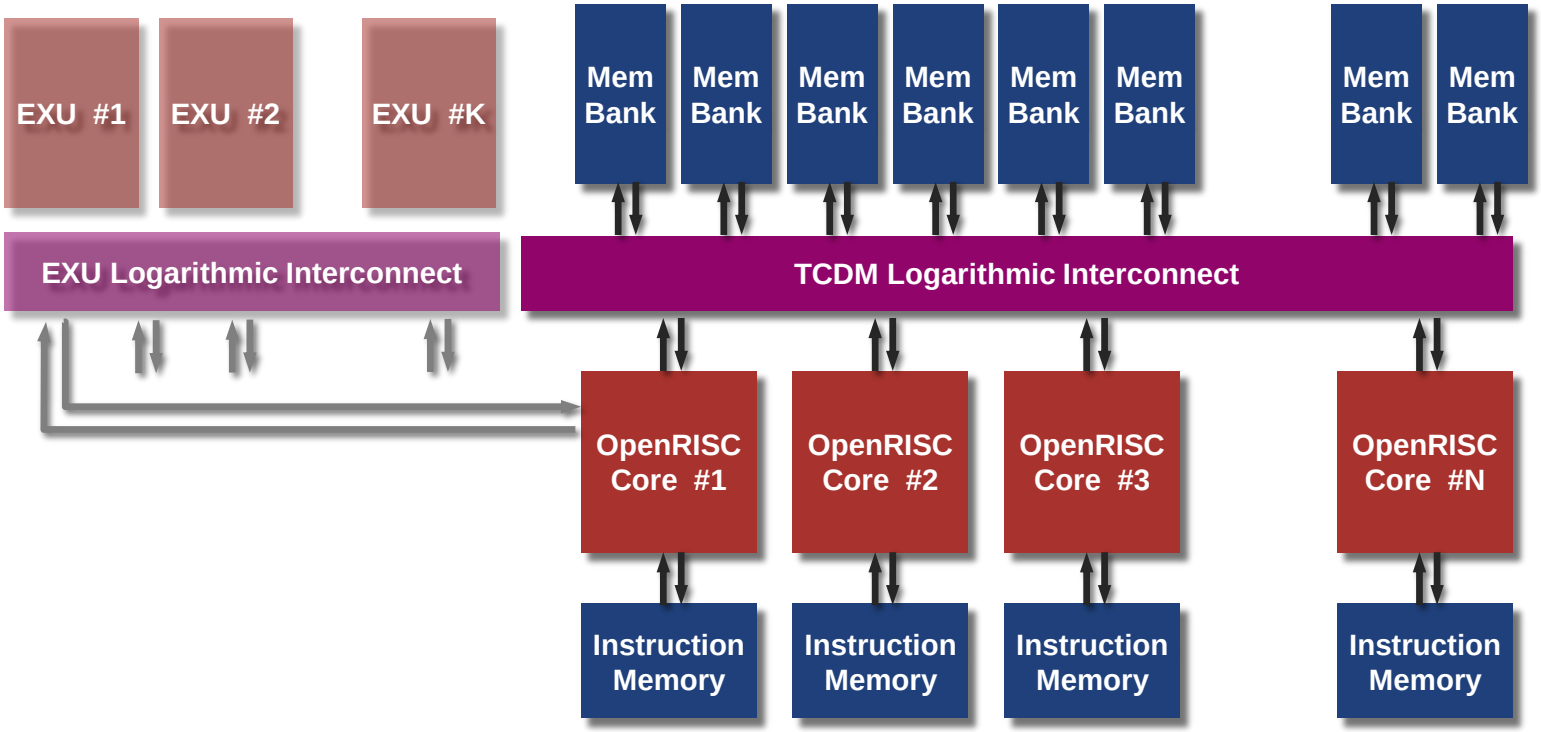
Parallel processors for performance @ NT

Making PULP



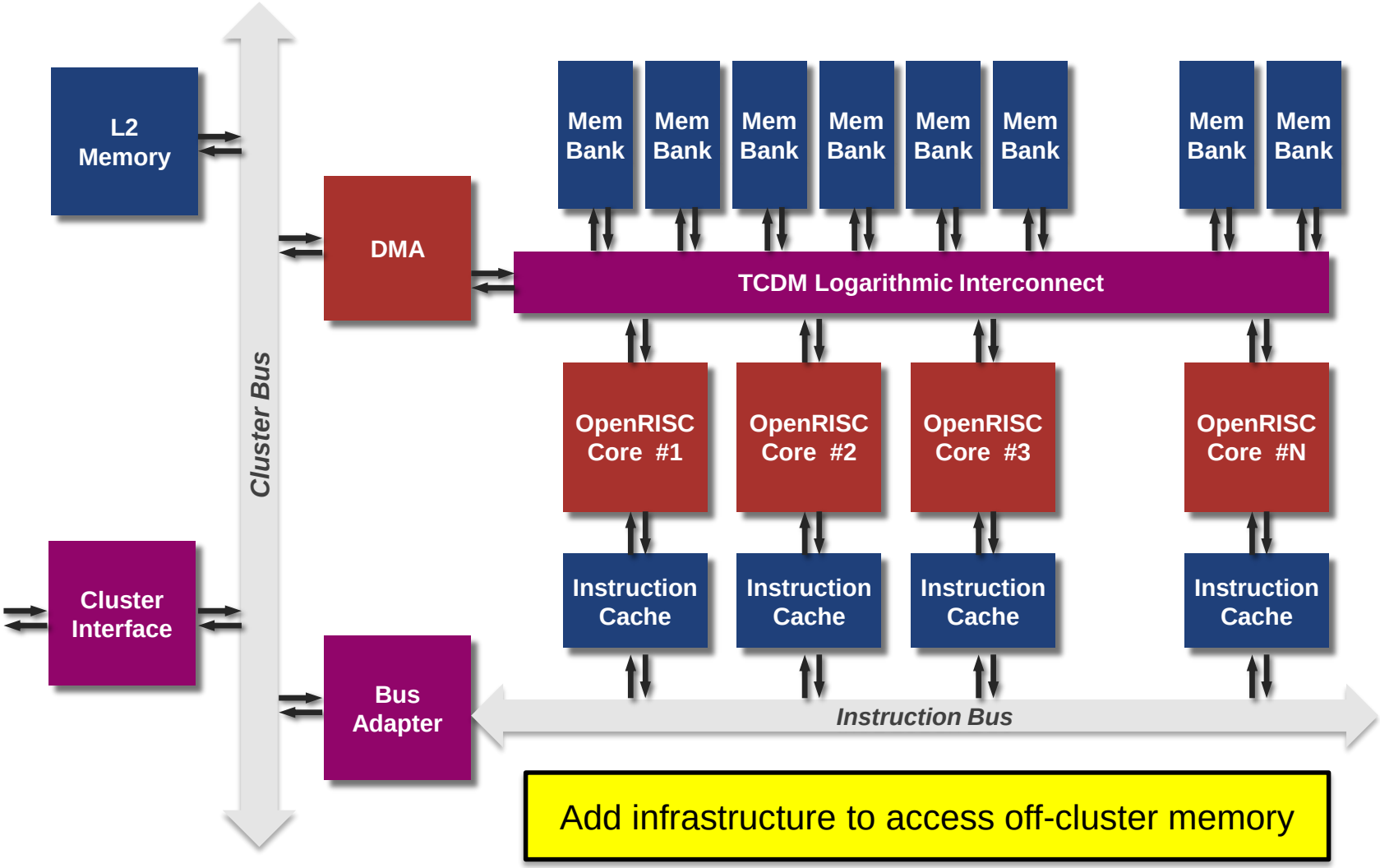
Parallel access to shared memory→Flexibility

Making PULP

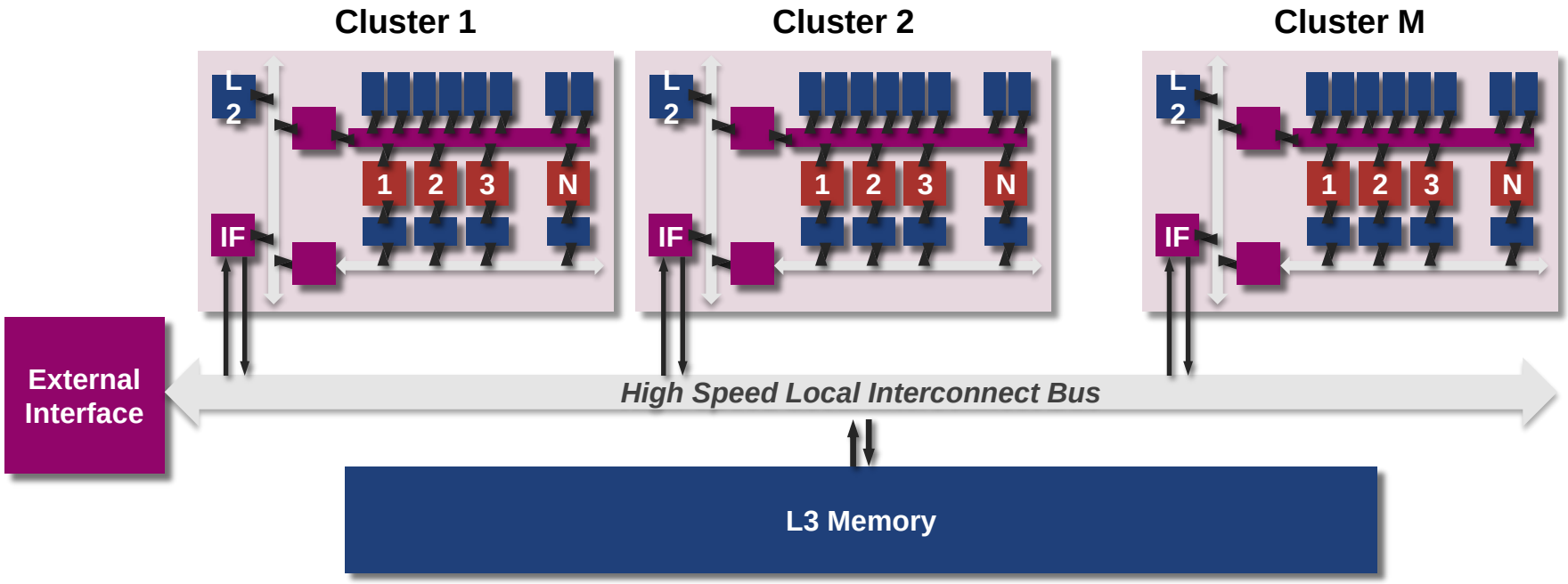


Optional Instruction Extension→Acceleration

Making PULP

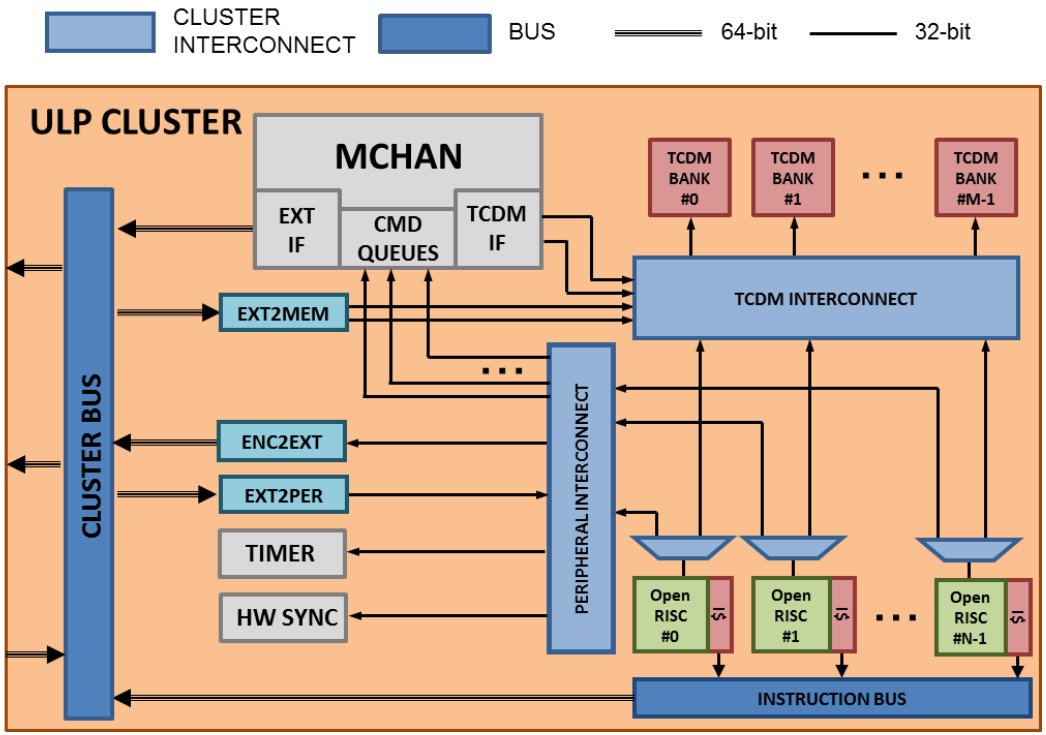

L2 Memory
DMA
Cluster Bus
Mem Bank
Mem Bank
Mem Bank
Mem Bank
Mem Bank
Mem Bank
Mem Bank
Mem Bank
TCDM Logarithmic Interconnect
OpenRISC Core #1
OpenRISC Core #2
OpenRISC Core #3
OpenRISC Core #N
Instruction Cache
Instruction Cache
Instruction Cache
Instruction Cache
Cluster Interface
Bus Adapter
Instruction Bus
Add infrastructure to access off-cluster memory

Making PULP



Multiple clusters (f,Vdd,Vbb) form a PULP system

PULP Cluster

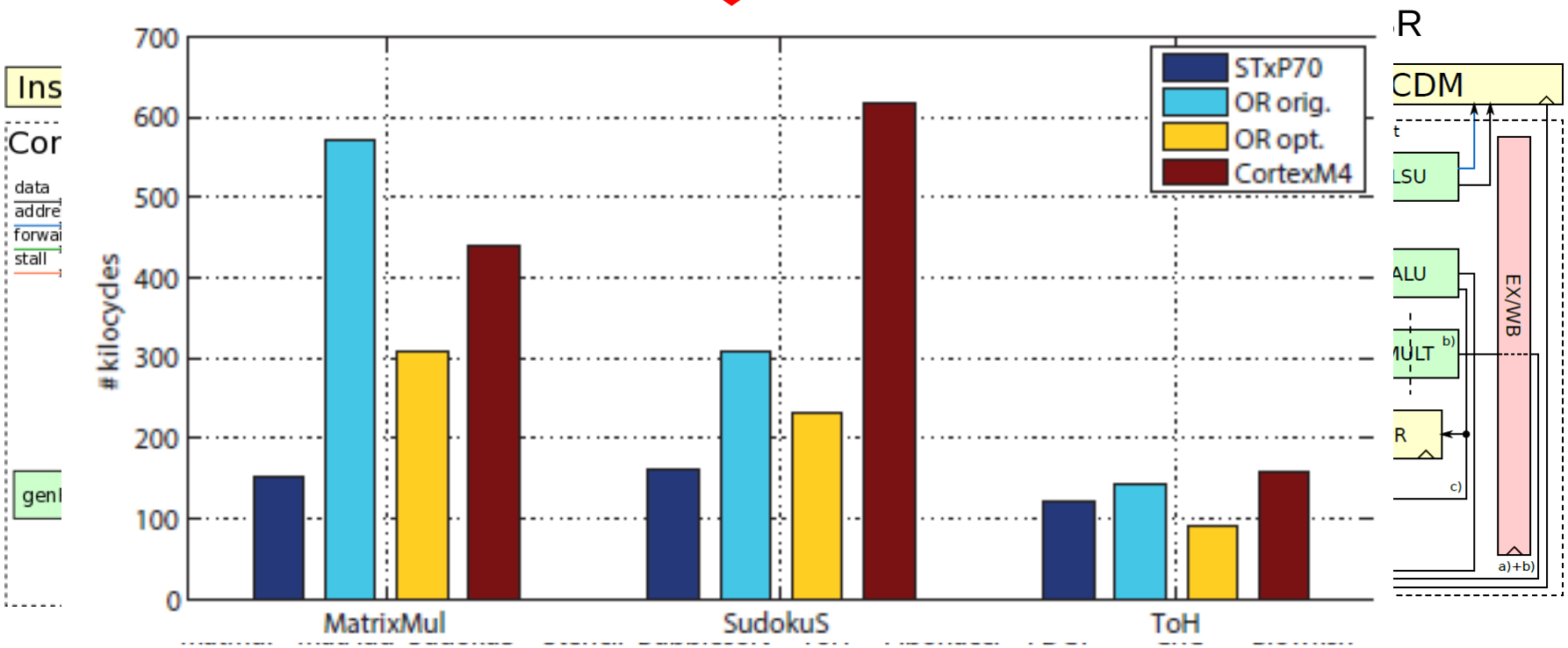


Design choices

- **I$** → high code locality & simple architecture
- **No D$** → low locality & high complexity: $Bpmm^2_{D\$}/Bpmm^2_{DTCDM}<0,4$
- **Sharing L1** → less copies, easy work-balancing, low T_{clk} overhead in NT
- **Multibank** → smaller energy per access, "almost" multiported

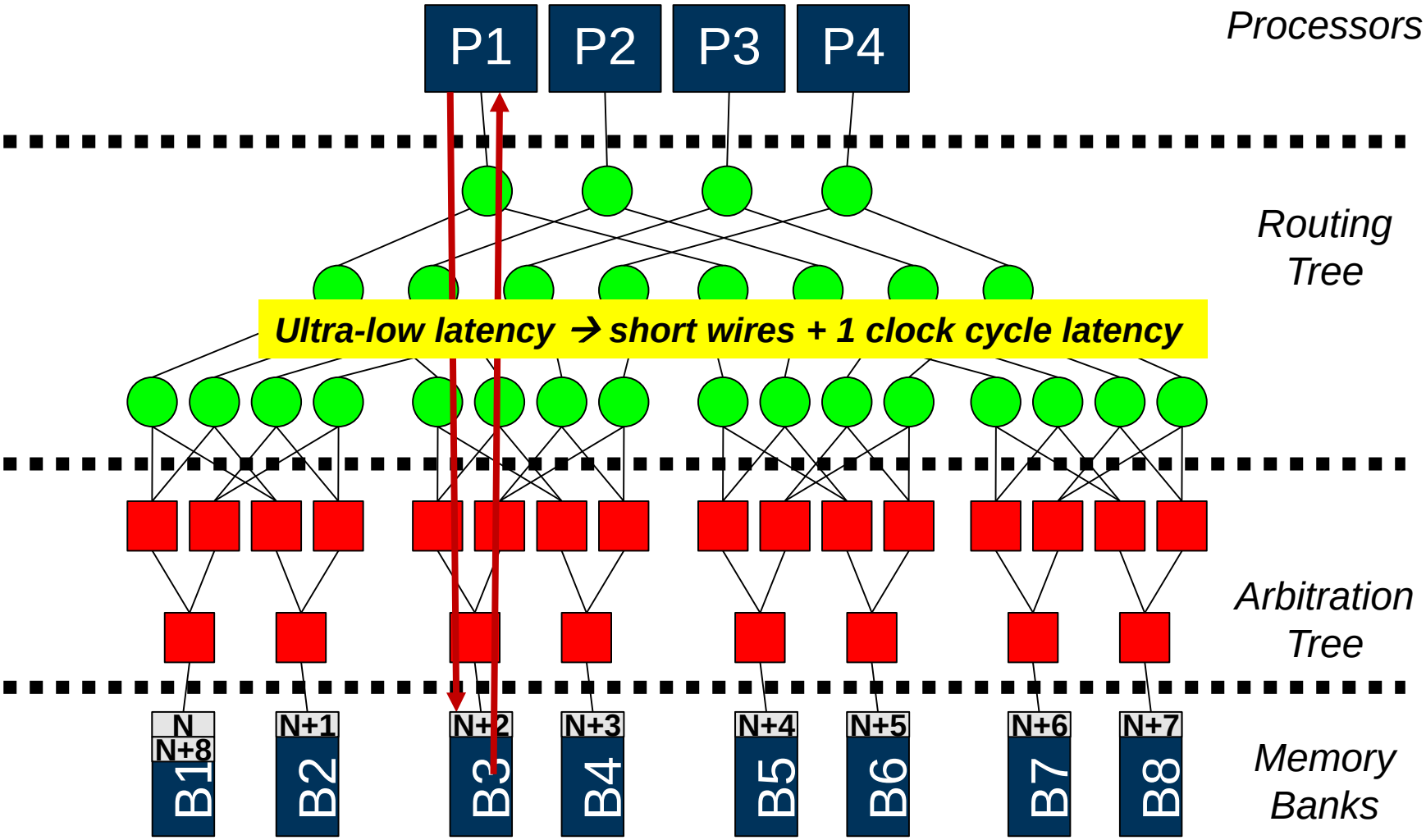
OpenRisc Optimization

- Superpipelining harmful for energy efficiency
  - Focused speed optimization on the critical path dominated by MEM
- Low pipeline depth → high IPC with simple microarchitecture



50% less energy per operation on average, 5% more area

Logarithmic Interconnect



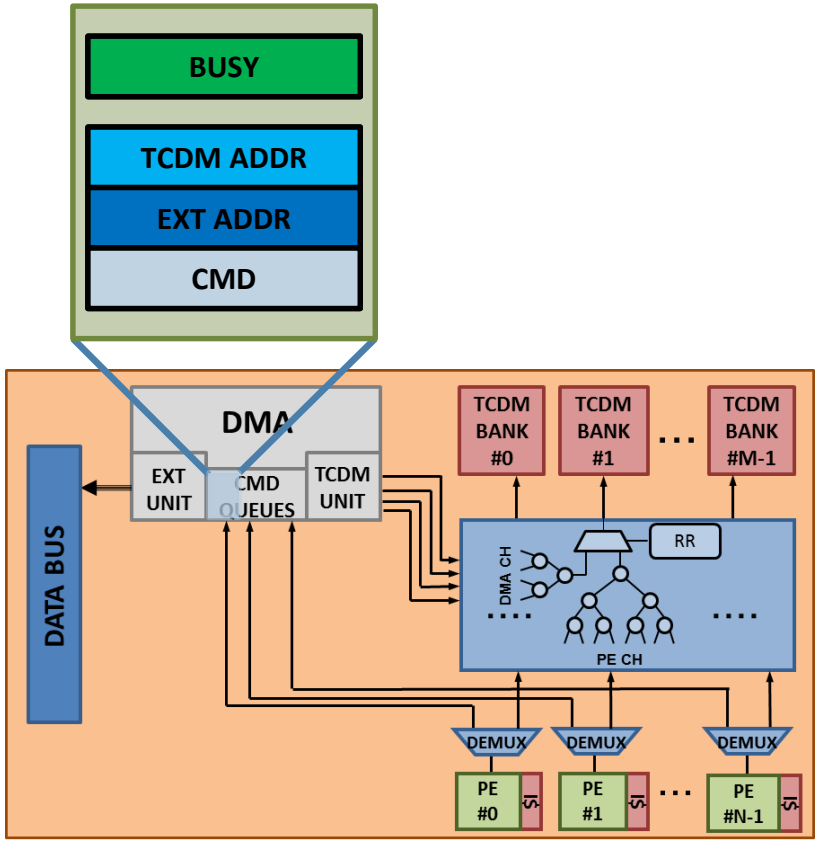
World-level bank interleaving «emulates» multiported mem

Low latency programming Interface

- Each command queue is dedicated to a core of the cluster: arbitration is made in hardware
  → No need to reserve (lock/unlock) the programming channel

- CORES program DMA through register mapped on the DEMUX
  → *The registers belongs to aliases, no need for the processors to calculate (per-core) offsets*

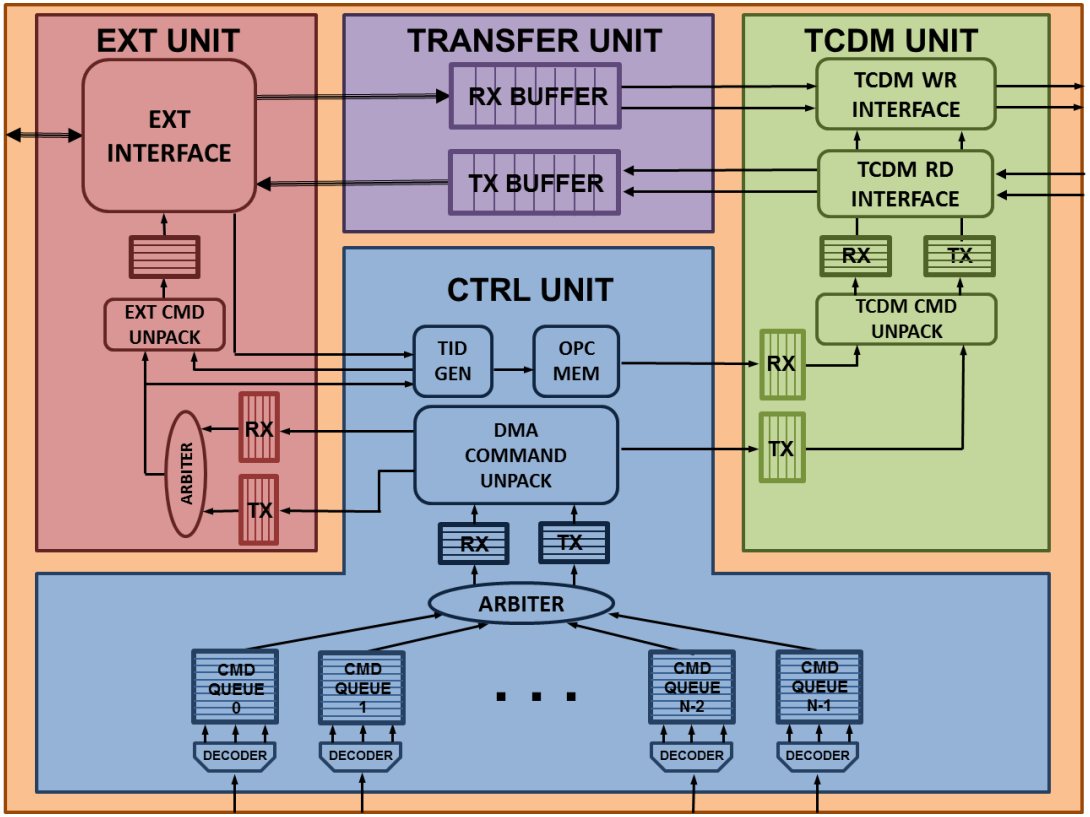
- A programming sequence requires
  1. *check a free command queue*
  2. *write address of buffer in TCDM*
  3. *write address of buffer in L2 memory*
  4. *Trigger data transfer*
  5. *Synchronization*



Programming Latency: ~10 CLOCK CYCLES!!!

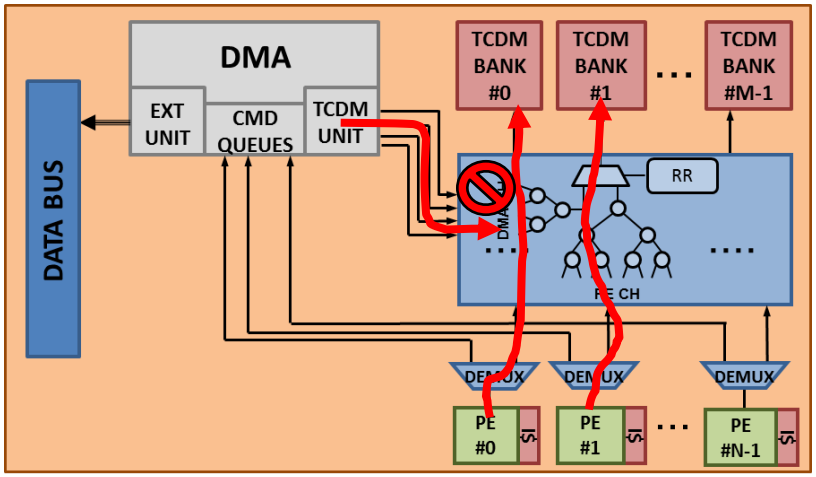
DMA Architecture Overview

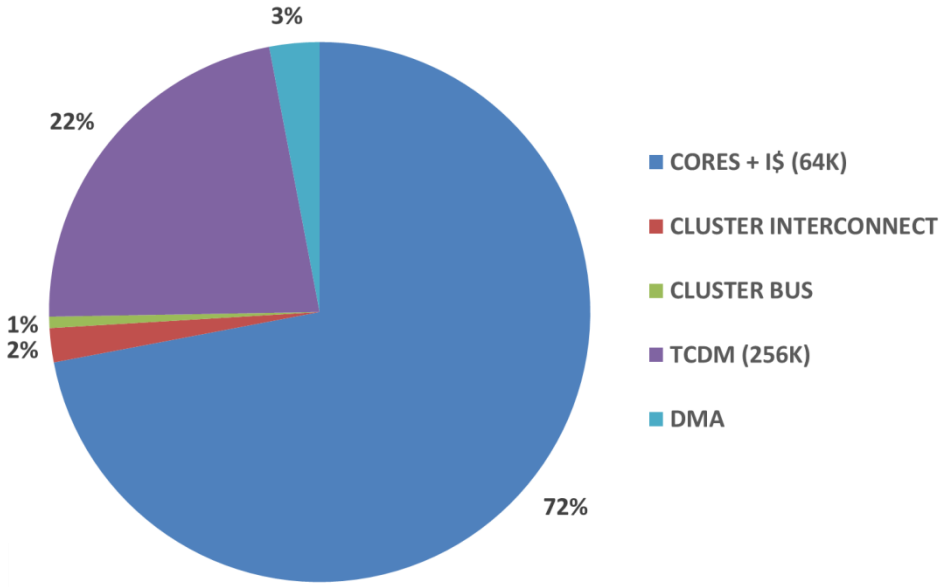
- **CTRL UNIT:**
  - Arbitration – forwarding and synchonization of incoming requests
- **TRANSFER UNIT:**
  - FIFO Buffers tor TX and RX channnels
- **TCDM UNIT:**
  - Bridge to TCDM protocol – 4 channels (2 RD and 2 WR) 32 bit each
- **EXT UNIT:**
  - Bridge to AXI, 64 bit

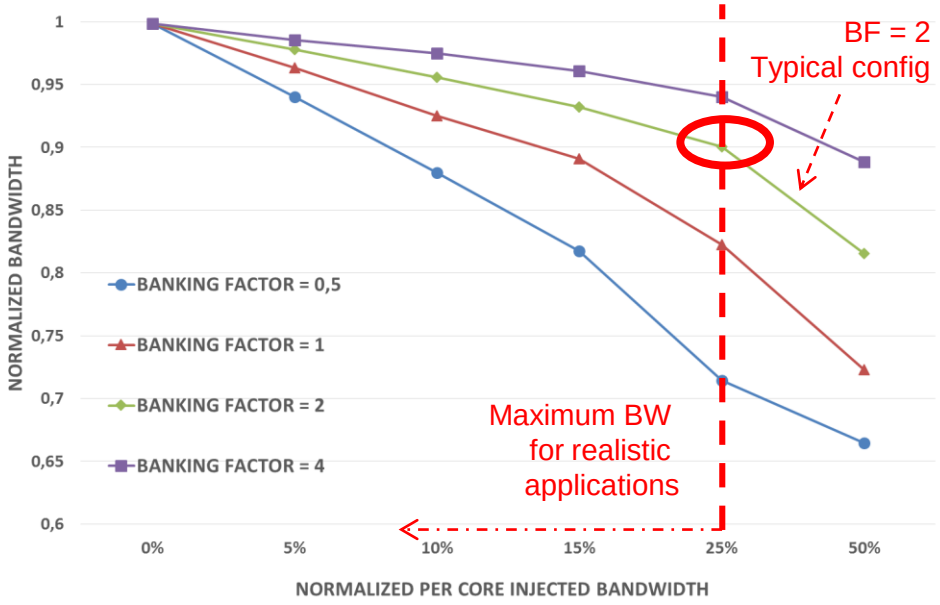


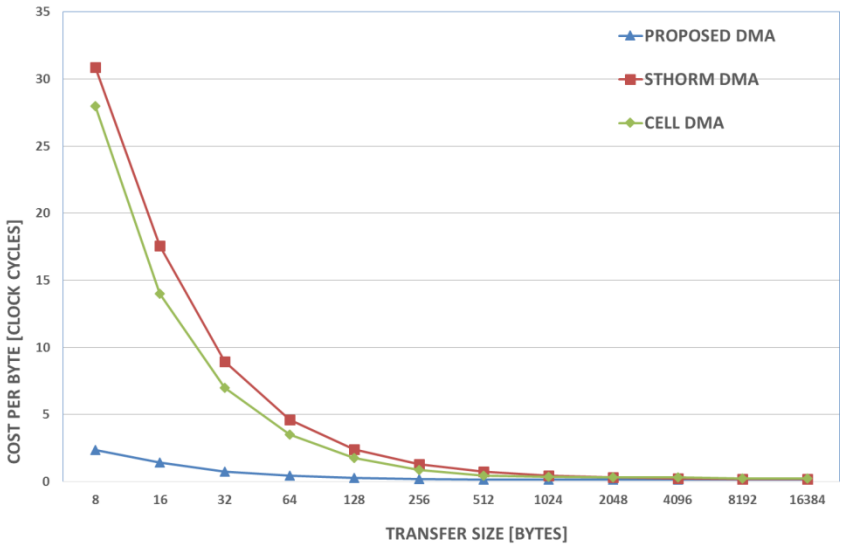
Key idea: only channel packets buffered internally – no DMA transfers!

Cluster DMA

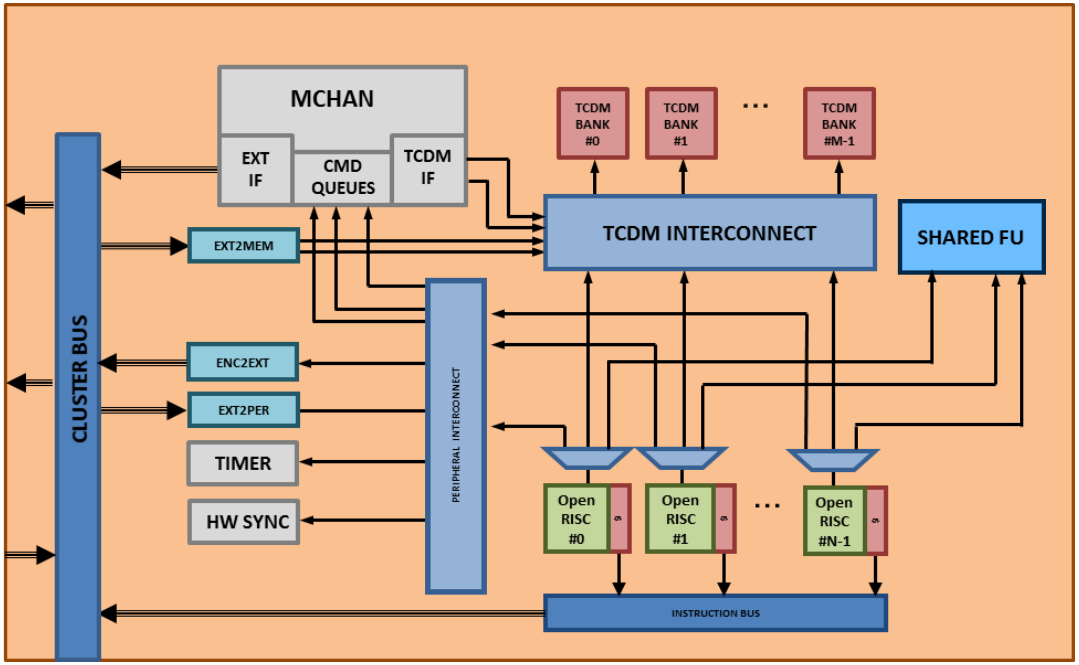

DMA
EXT UNIT
CMD QUEUES
TCDM UNIT
DATA BUS
TCDM BANK #0
TCDM BANK #1
TCDM BANK #M-1
RR
DMA
PE CH
DEMUX
DEMUX
DEMUX
PE #0
PE #1
PE #N-1
I$
I$
I$

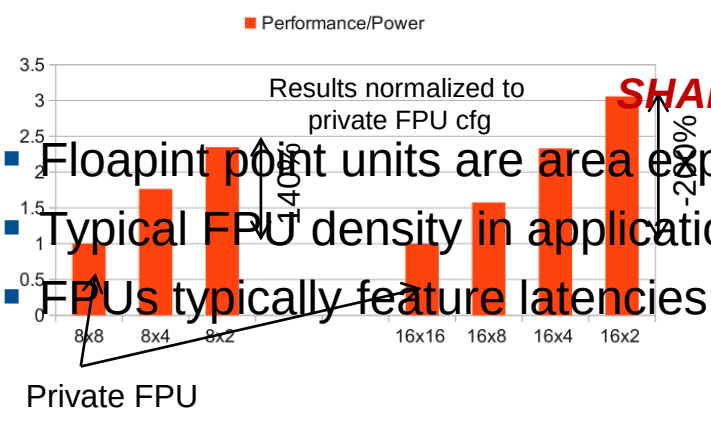

3%
22%
1%
2%
72%
CORES + I$ (64K)
CLUSTER INTERCONNECT
CLUSTER BUS
TCDM (256K)
DMA

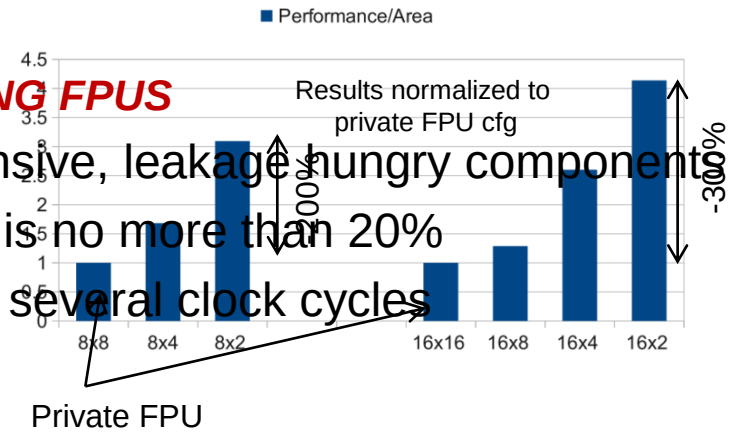

BF = 2
Typical config
Maximum BW for realistic applications
BANKING FACTOR = 0,5
BANKING FACTOR = 1
BANKING FACTOR = 2
BANKING FACTOR = 4
NORMALIZED BANDWIDTH
NORMALIZED PER CORE INJECTED BANDWIDTH
1
0,95
0,9
0,85
0,8
0,75
0,7
0,65
0,6
0%
5%
10%
15%
25%
50%


PROPOSED DMA
STHORM DMA
CELL DMA
COST PER BYTE [CLOCK CYCLES]
TRANSFER SIZE [BYTES]
35
30
25
20
15
10
5
0
8
16
32
64
128
256
512
1024
2048
4096
8192
16384

Sharing fucntional units (instruction set extensions)







SHARING FPUS

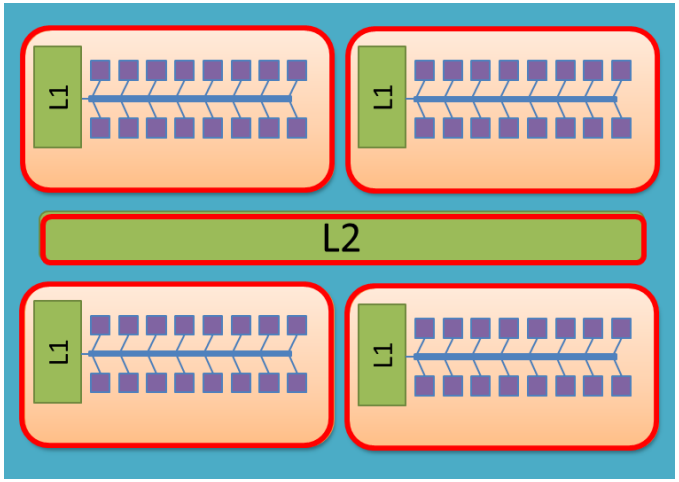
- Floapint point units are area expensive, leakage hungry components
- Typical FPU density in application is no more than 20%
- FPUs typically feature latencies of several clock cycles

Scaling up

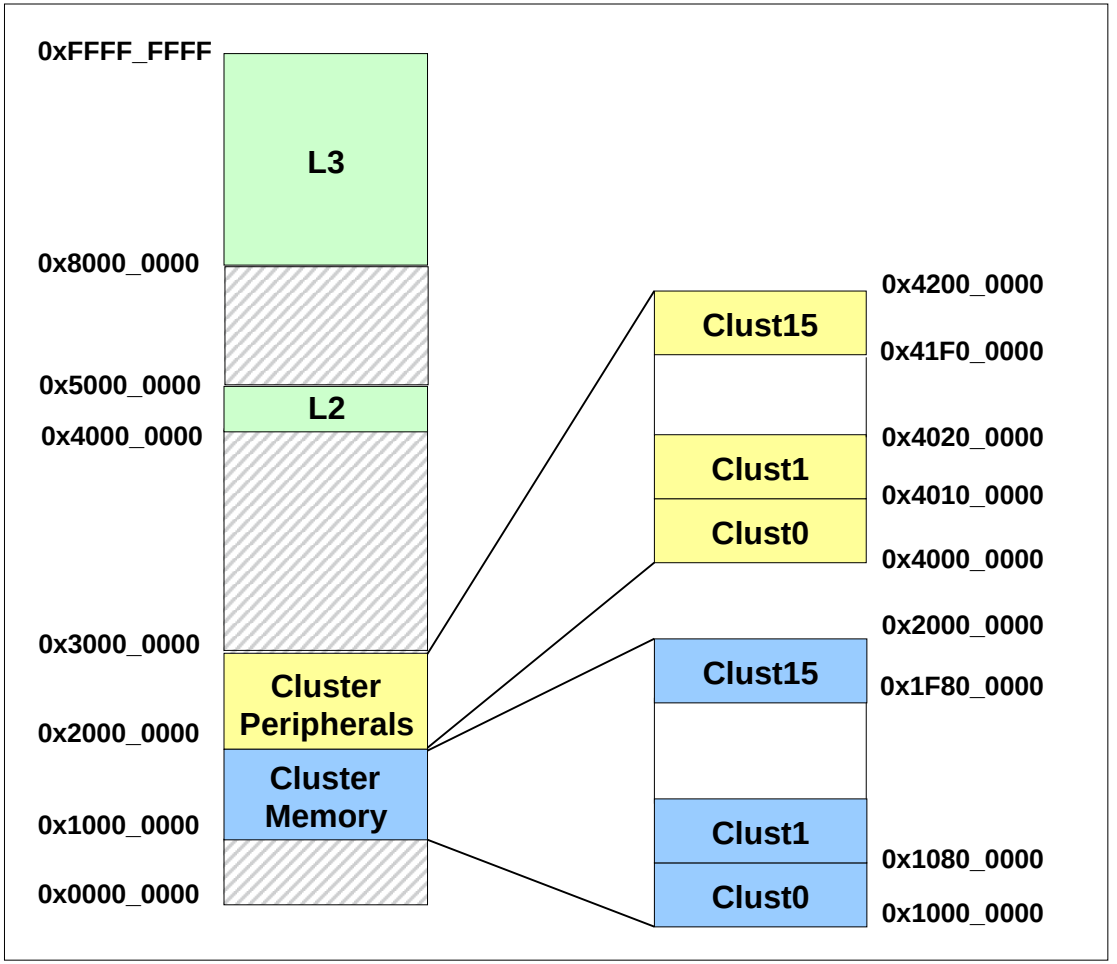
- 32 bit architecture

⇒ 4 GB of memory

- Clusters in Vdd, CLK domains
- L2 (2D & 3D) ready
- Host VM IF for Heterogeneous Computing



Memory Map:



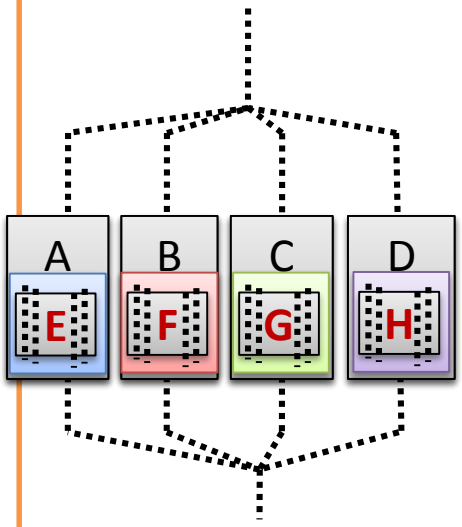
Programmability: OpenMP


```
while(1)
{
    #pragma omp parallel num_threads(4)
    {
        #pragma sections
        {
            #pragma section
            {
A.              #pragma omp parallel num_threads(16)
                ColorScaleConv();
            }
B
            }
            #pragma section
            {
C.              #pragma omp parallel num_threads(16)
                cvMoments();
            }
            #pragma section
            {
D.              #pragma omp parallel num_threads(16)
                cvAdd();
            }
        }
    }
}
```

A powerful abstraction for specifying structured parallelism

```
void ColorScaleConv()
{
   #pragma omp for
    for(i = 0; i < FRAME_SIZE; i++)
    {
        [ALGORITH]
    }
}
```

And very suitable for NUMA (cluster-based) systems

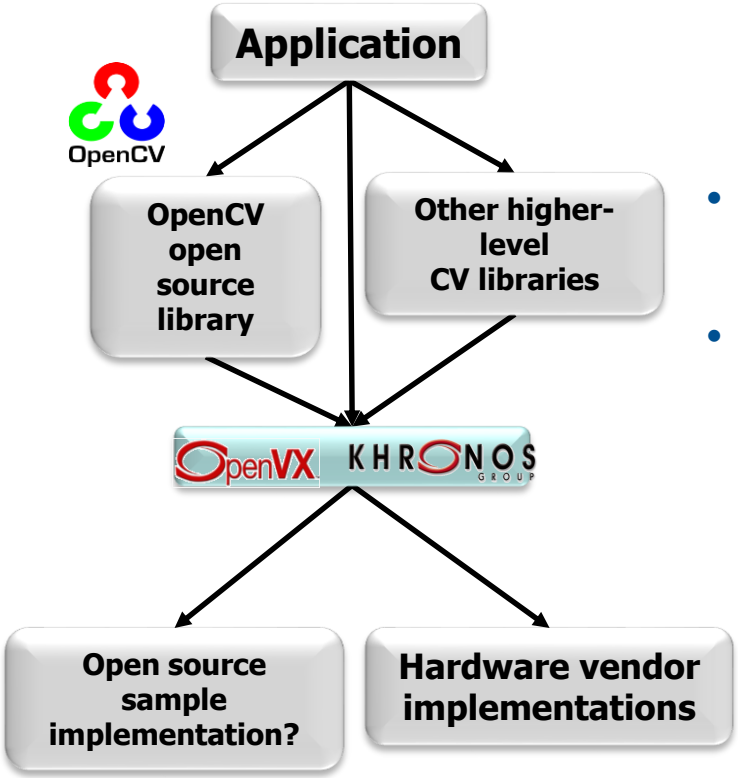


POLITECNICO MILANO

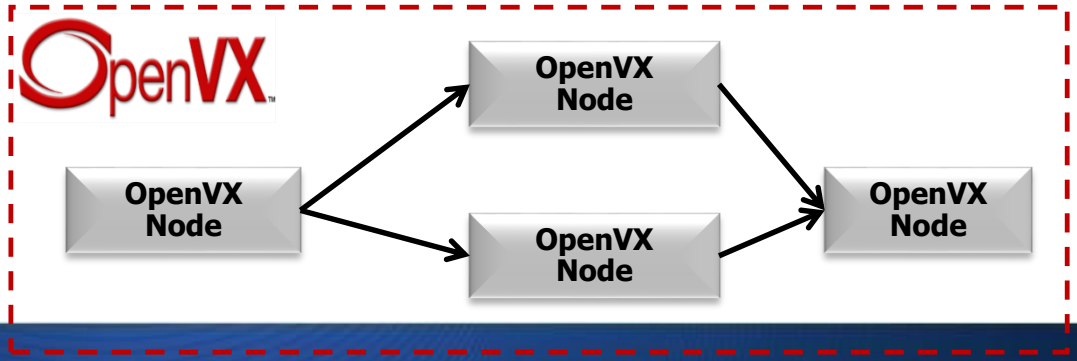
LLVM + OpenCL

+OpenVX →domain specific language

Programming: OpenVX

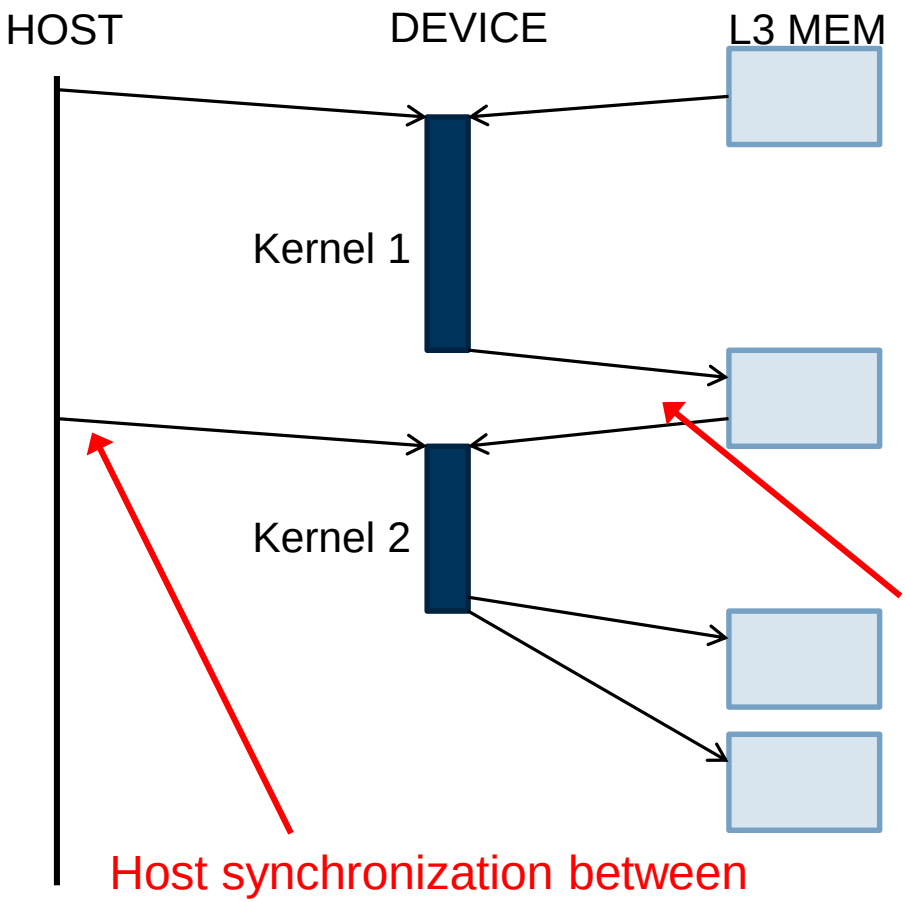


- C-based standard API for vision kernel
  - Defines a set of **standard kernels**
  - Enables hardware vendors to ***implement accelerated imaging and vision algorithms***
- Focus on enabling real-time vision
  - On mobile and embedded systems
- **Graph execution model**
  - Each node can be implemented in software or accelerated hardware
  - **Data transfer between nodes may be optimized**



Vanilla OVX Runtime

Leverages OpenCL runtime with OVX nodes treated as OCL kernels

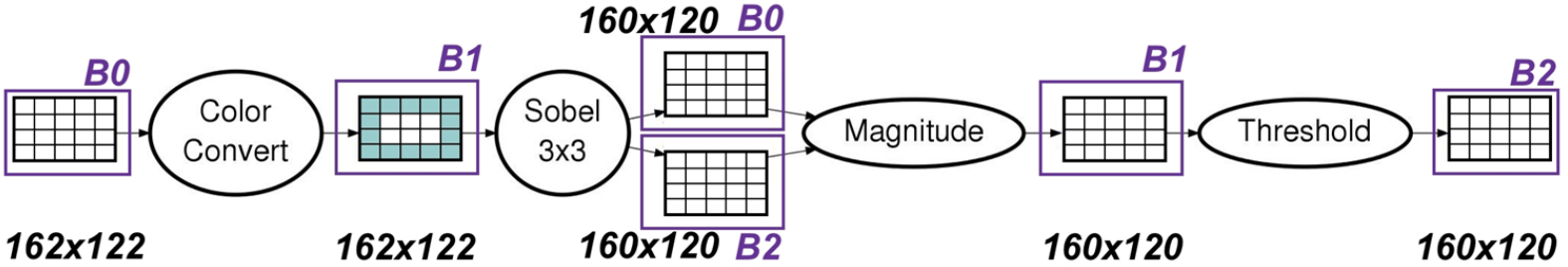


Intermediate results stored in L3 mem → each kernel boundary implies two accesses to all image data (write + read)

Host synchronization between successive kernels is imposed

Localized Execution

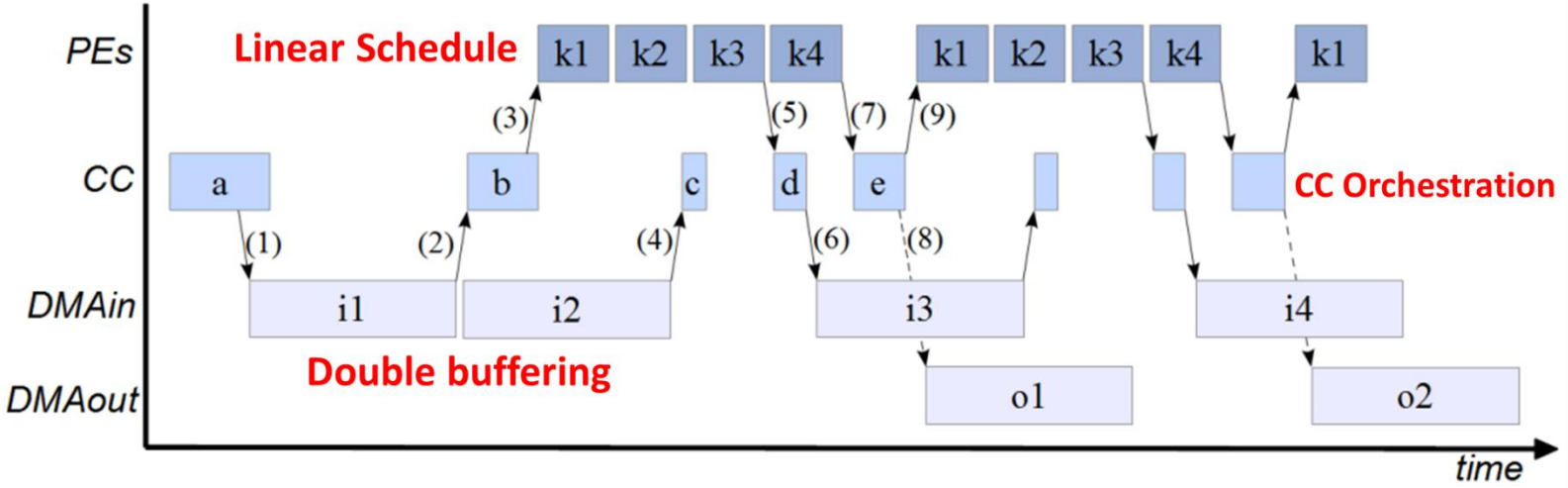
- Smaller image sub-regions (**tiles**) totally reside in TCDM
- All kernels are computed at tile level, no more at image level → *intermediate results are also allocated in TCDM*



- ✓ TCDM is partitioned into 3 buffers (B0, B1, B2)
- ✓ Output tile size is 160x120
- ✓ Sobel3x3 requires image overlapping (1 pixel for each direction), and so the tile size is 162x122
- ✓ ColorConvert does not require tile overlapping, but must provide a 162x122 result tile for the next stage → **tile size propagation**

Localized Execution

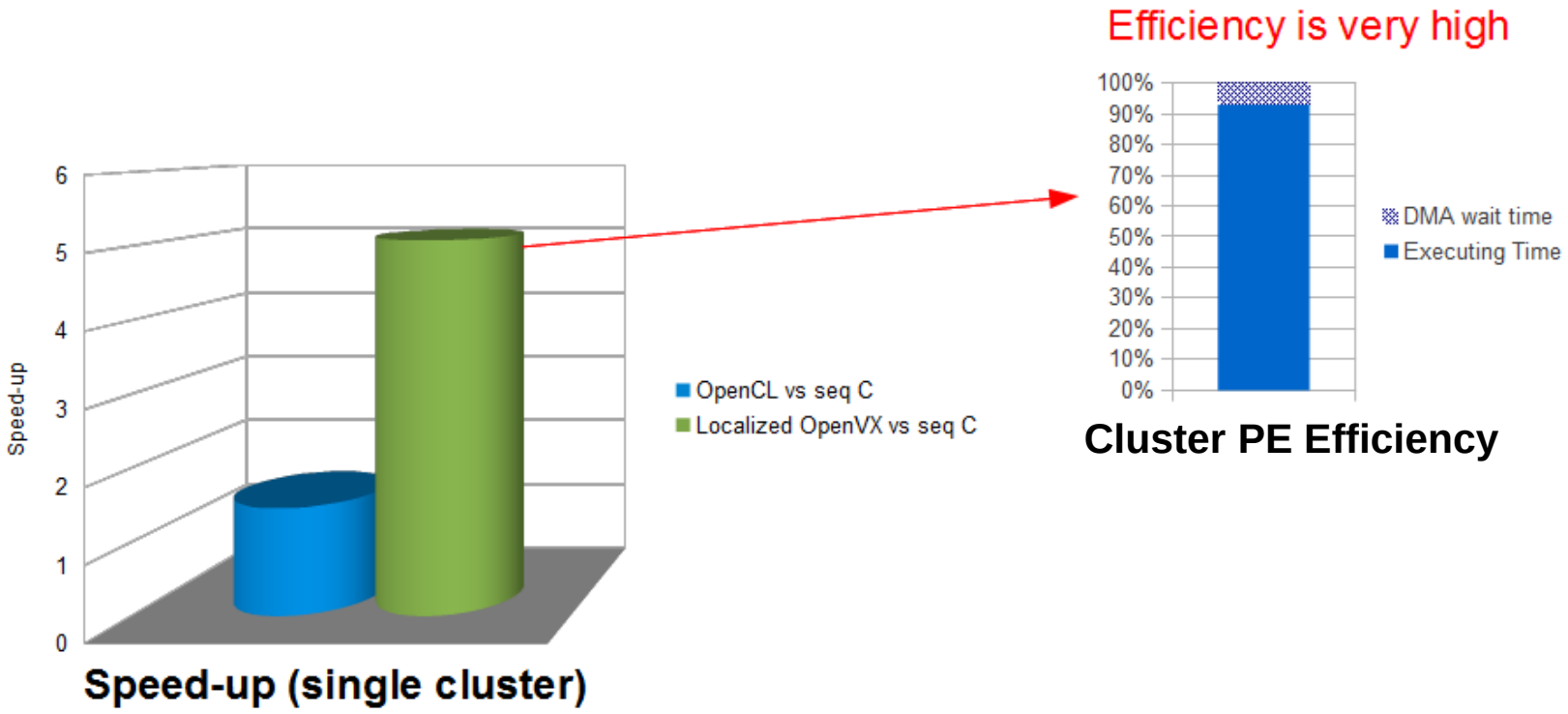
- Smaller image sub-regions (**tiles**) totally reside in TCDM
- All kernels are computed at tile level, no more at image level → *intermediate results are also allocated in TCDM*



- ✓ TCDM is partitioned into 3 buffers (B0, B1, B2)
- ✓ Output tile size is 160x120
- ✓ Sobel3x3 requires image overlapping (1 pixel for each direction), and so the tile size is 162x122
- ✓ ColorConvert does not require tile overlapping, but must provide a 162x122 result tile for the next stage → **tile size propagation**

Localized Execution Results

- Framework prototype
  - OpenVX support
  - A limited subset of kernel has been implemented
  - Polynomial time (i.e. fast) heuristics suitable for ***just-in-time execution***

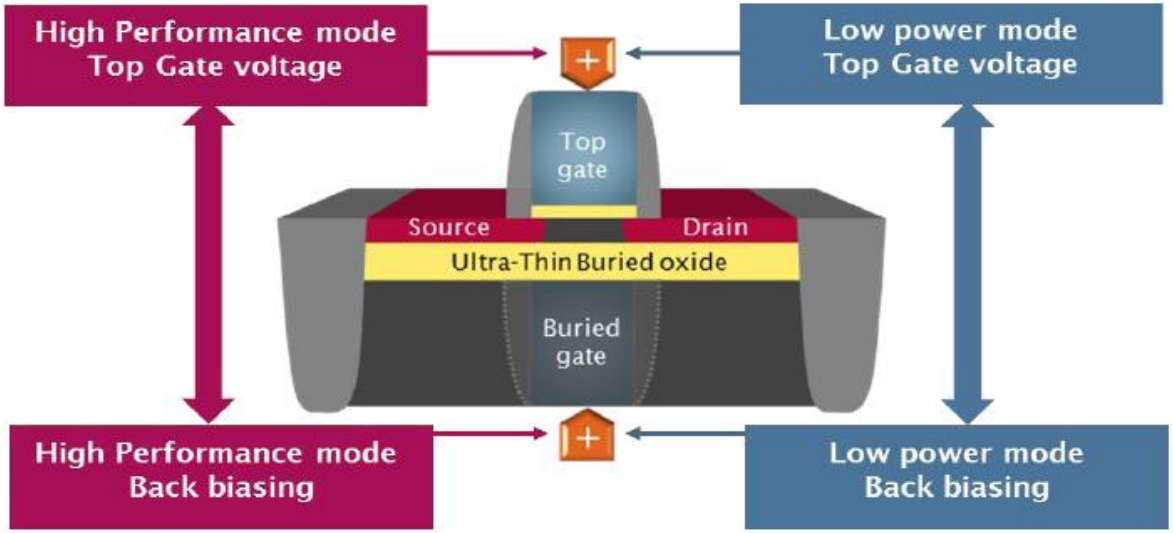


ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

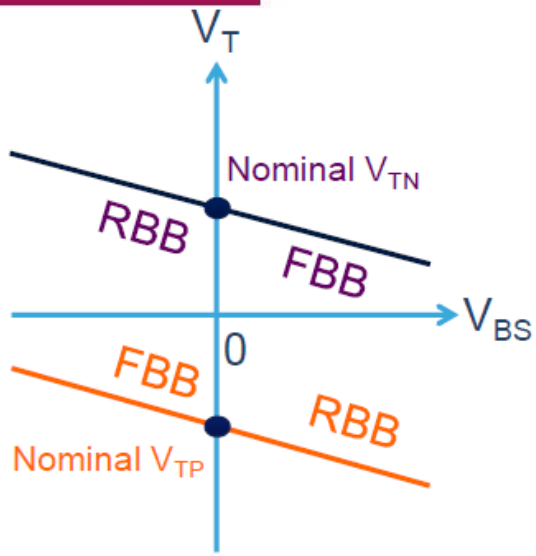
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

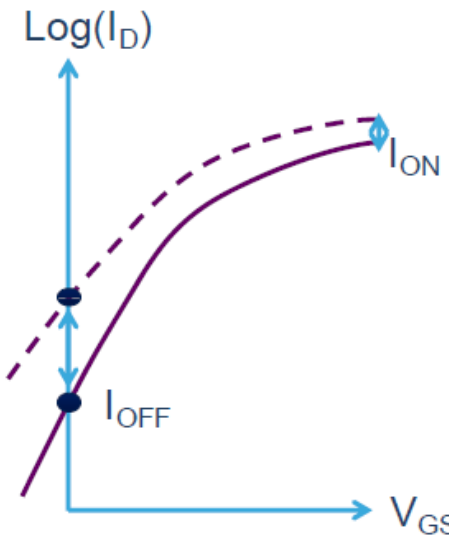
And what about Technology?

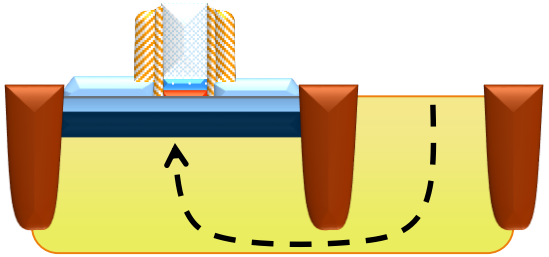
Near threshold FDSOI technology



life.augmented

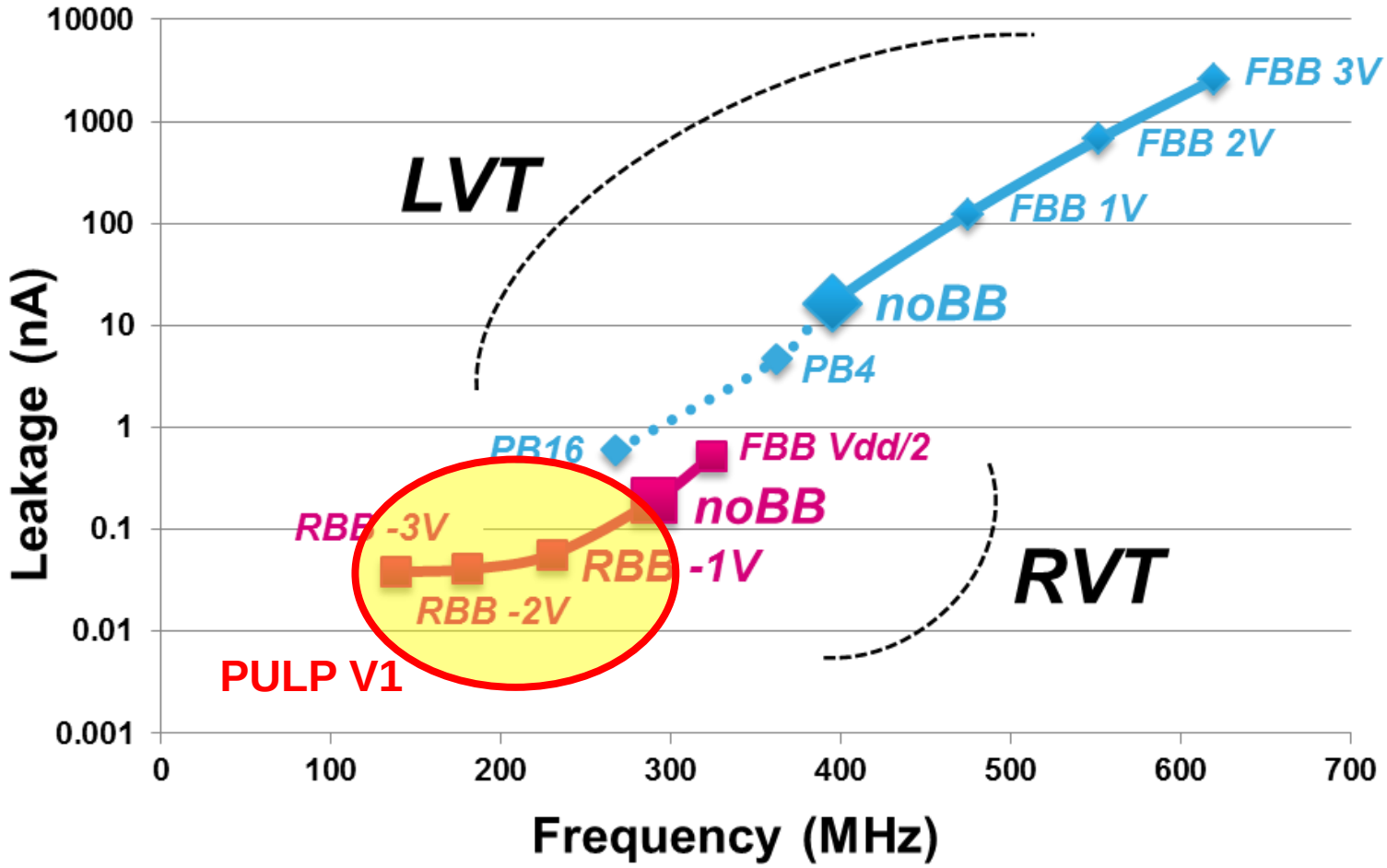




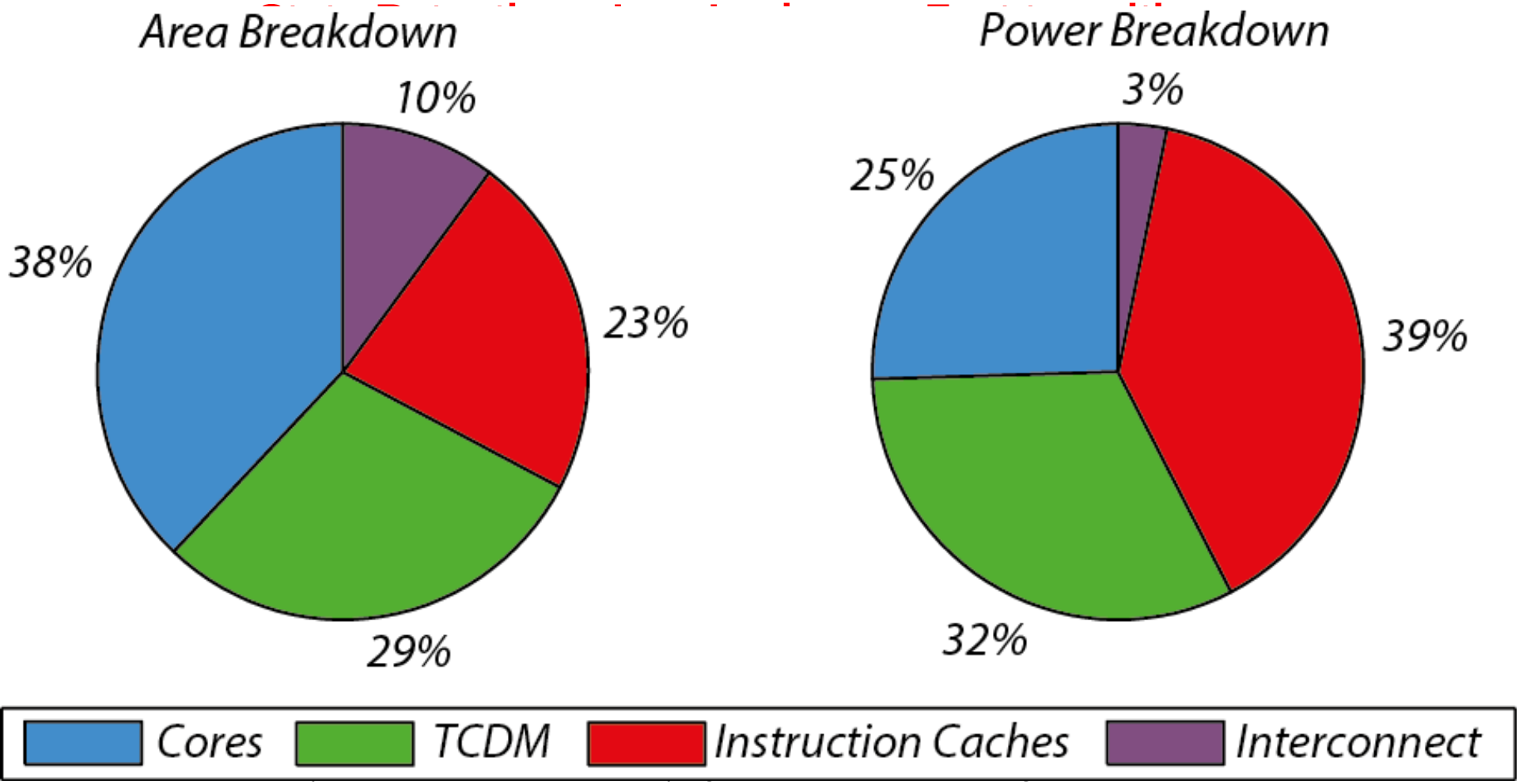


Body bias: Highly effective knob for leakage control!

Near threshold FDSOI technology



PULP V1: Doing nothing well (with RBB)



More than 50% of power into memories… this is the next focus area!

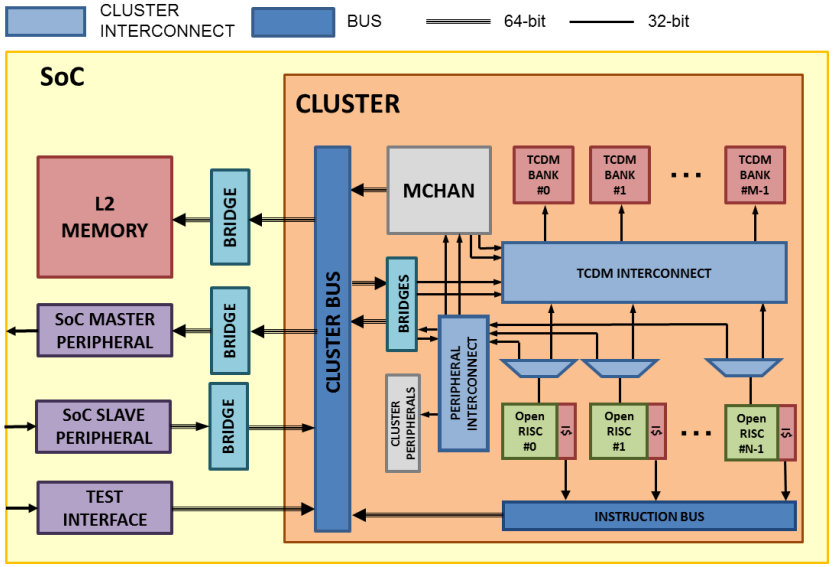
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

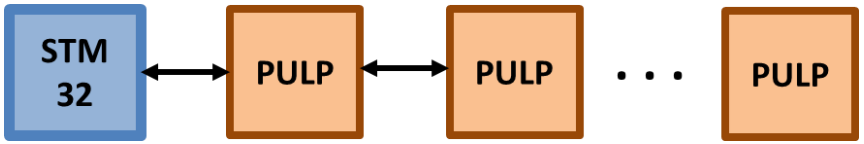
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Introducing PULP V2

Board/Application-ready chip

- Implementation of a master and a slave peripheral
  - Standard peripheral (e.g. SPI)
  - Integration with FPGAs or standard low power external memories
- PULP as multi-core accelerator for micro-controller host
  - STM32 host core
  - PULP muti-core accelerator
- Daisy chain of PULP chips
  - Pipeline of parallel processing units
  - Each core perfroms a stage of computation and forword temporary data to another stage

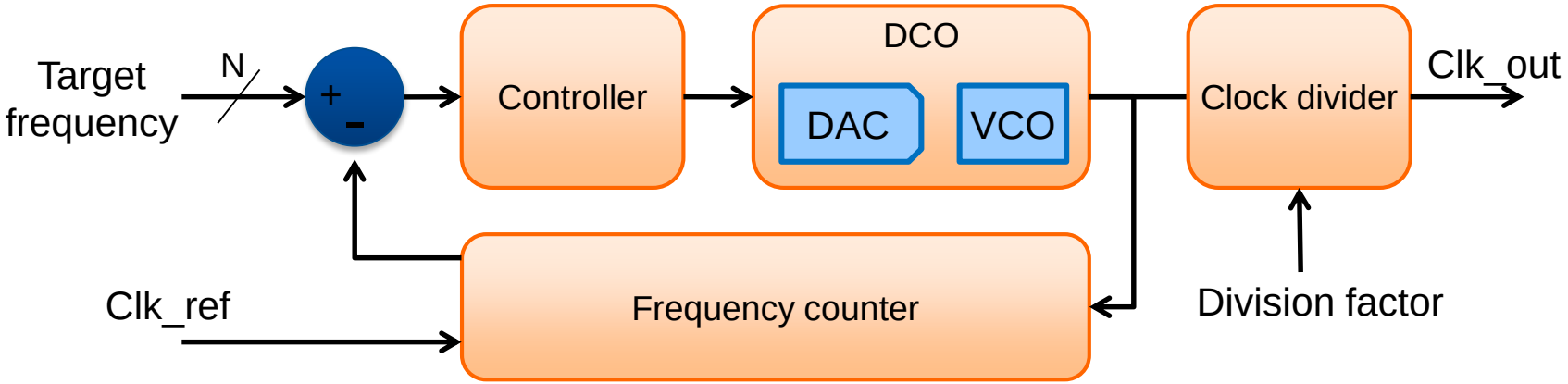




Standalone mode is also supported!

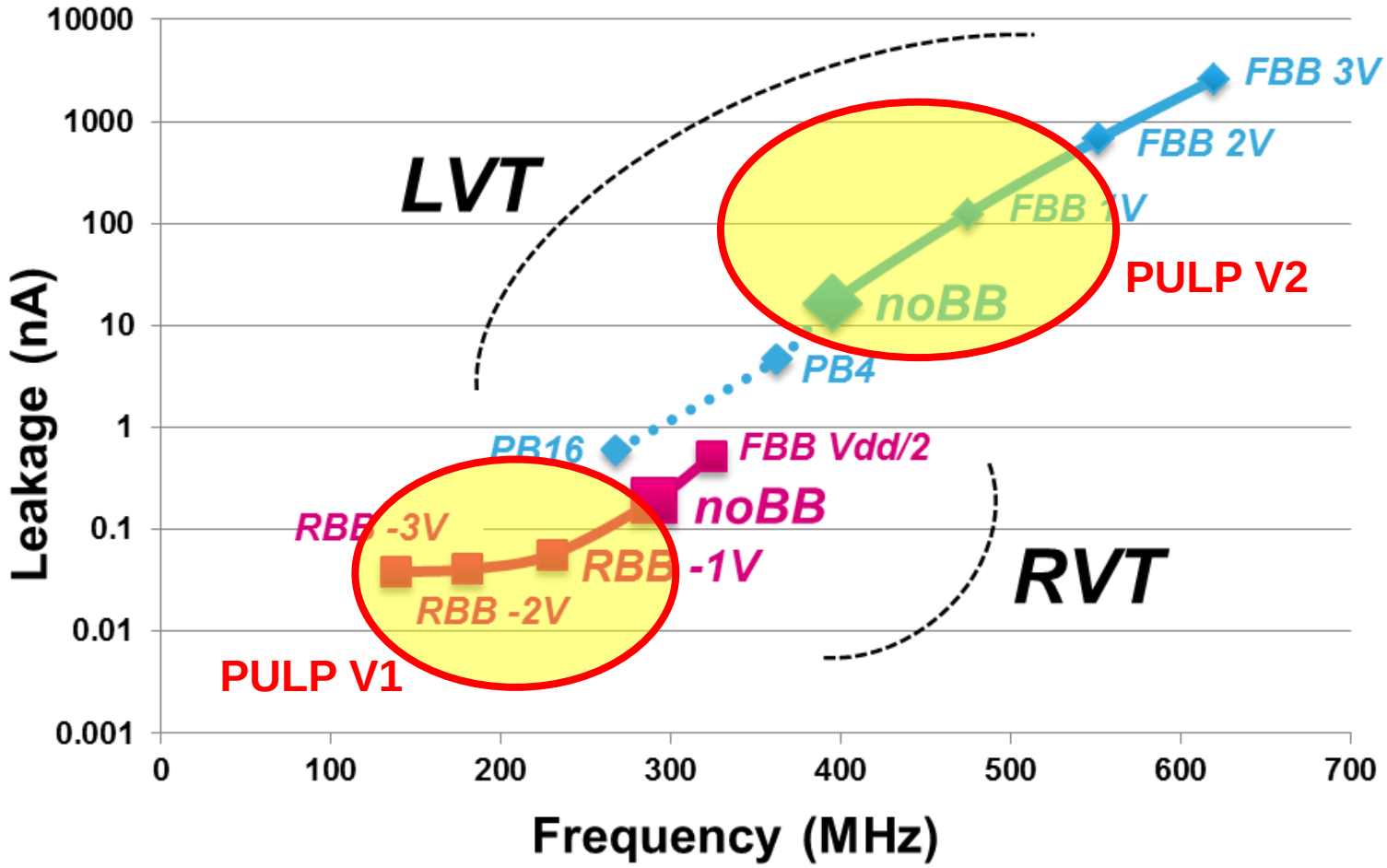
Clock generator: FLL

- All-digital clock generation based on a Frequency Lock Loop (FLL)
- From 2GHz down to 15KHz (through clock division)
- Frequency step 10MHz (at lowest division factor)
- Small area 3300µm² (50 times smaller than classic PLL)
- Suited to fine-grain GALS architectures
- Frequency reprograming in less than 180ns
- No frequency overshoot
- 15ps jitter



S. Lesecq, D. Puschini, E. Beigne, P. Vivet, and Y. Akgul, "Low-Cost and Robust Control of a DFLL for Multi-Processor System-on-Chip," IFAC Proceedings Volumes, vol. 18, pp. 1940–1945, 2011

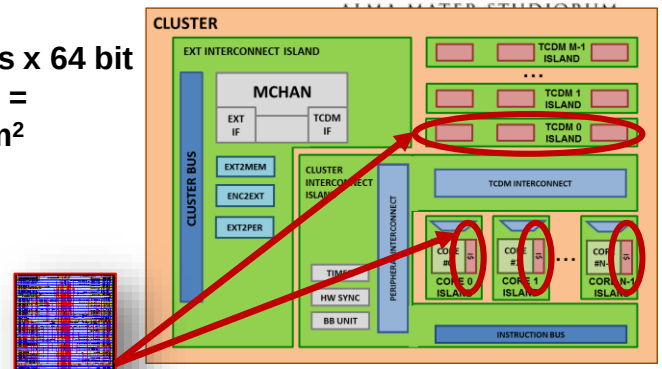
Near threshold FDSOI technology



Low-leakage vs. Low voltage (0.3V) → **reactive** or proactive?

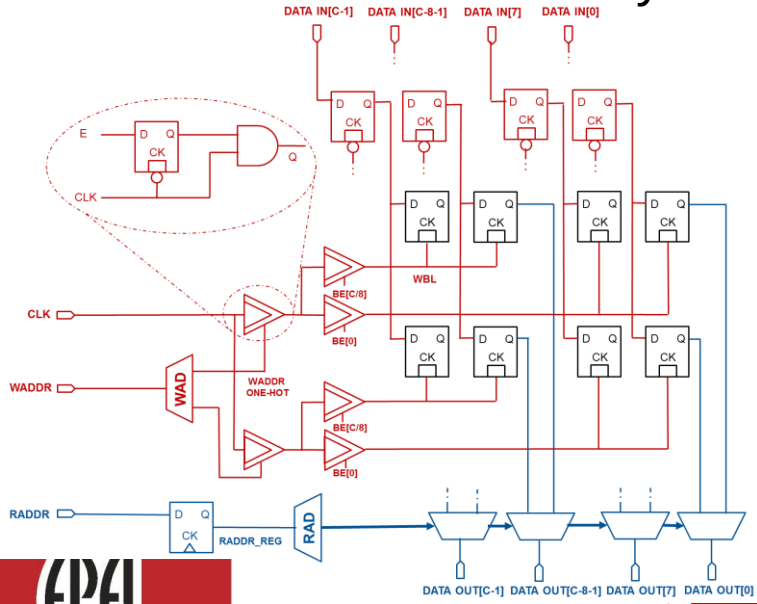
ULP Latch-based SCM

64 words x 64 bit
162 x 85 =
13.7k um²



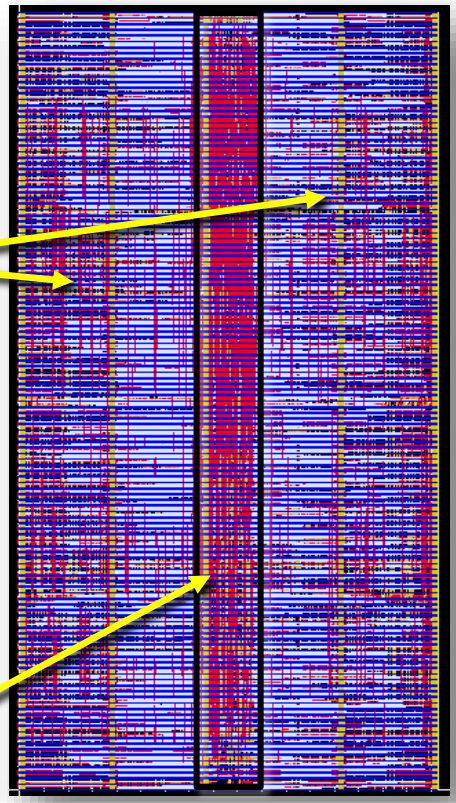
- Based exclusively on standard cells
- Voltage range identical to the core
  - Only static logic
- Layout based on guided P&R
  - Close to 100% density

SCMs



Macro Size:
128 x 32 (4 kb)
86µm x 160µm

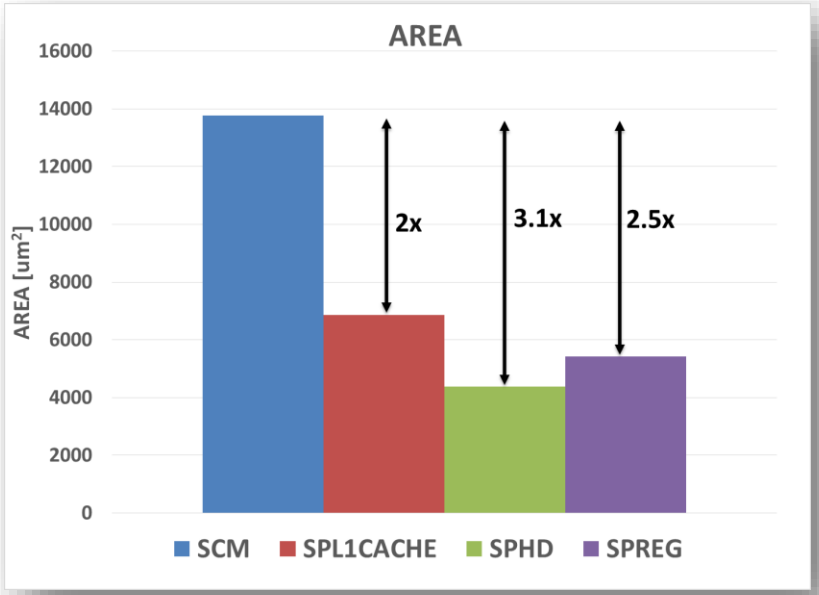
Input/Output Delay:
0.3ns/0.7ns @ss0.9V125°
2ns/3.3ns @tt0.3V25° (FBB)

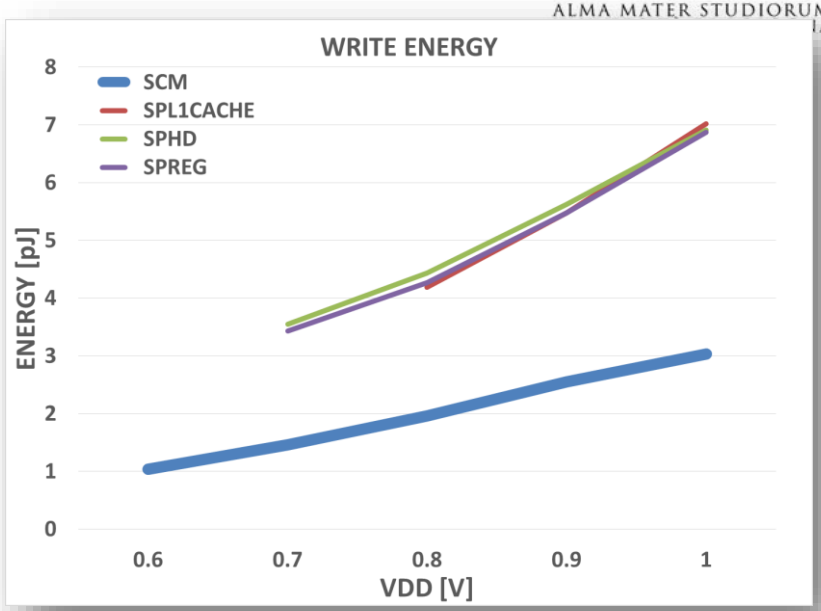


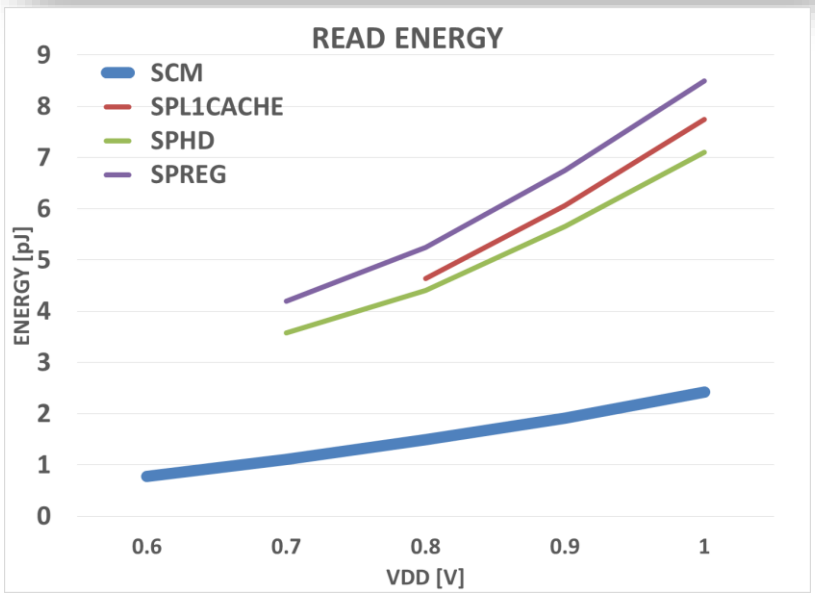
Storage Array + output mux

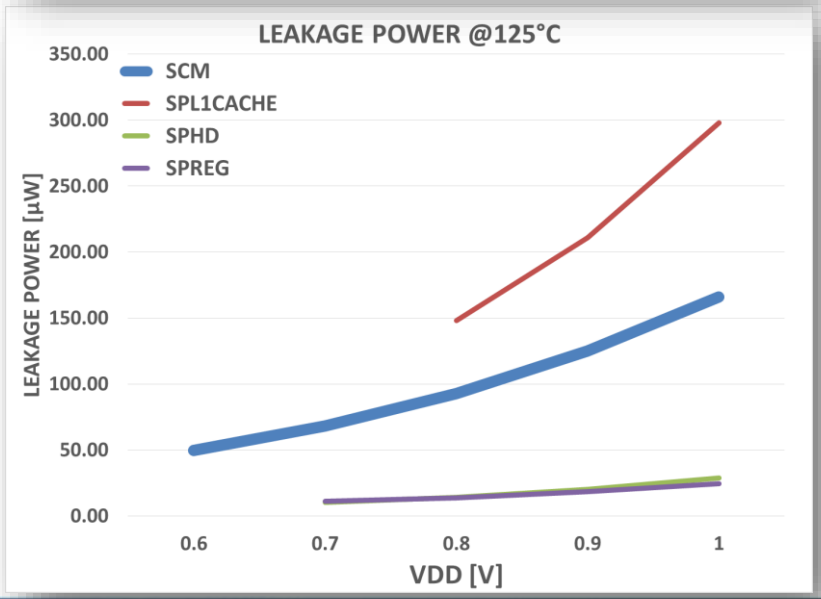
Decoders

Comparison to 64x64 SRAM

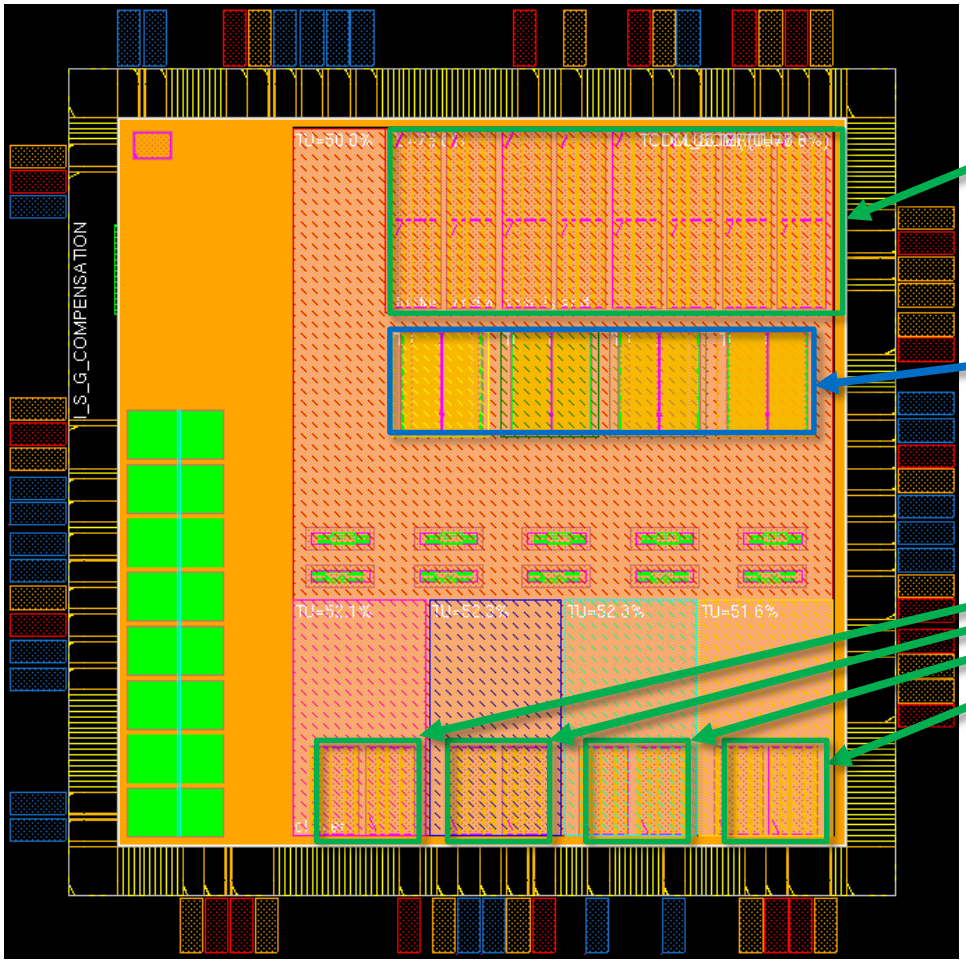








SCM Integration into PULP2



16 banks of
128x32 bit SCM
(8 kbyte)

8 banks of
512x32 bit SRAM
(16 kbyte)

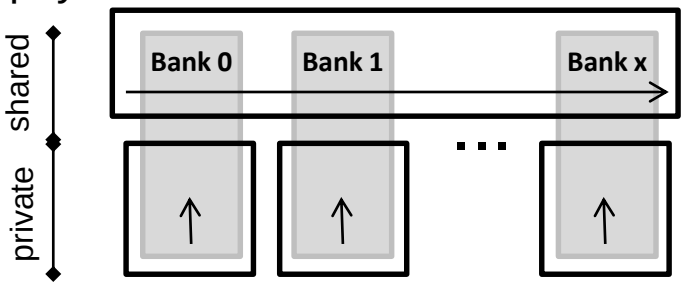
2 banks of
128x32 bit SCM
(2 kbyte)
Instruction
cache per core

PULP V2 is taping out this Week, PULP V3 is on the drawing board…

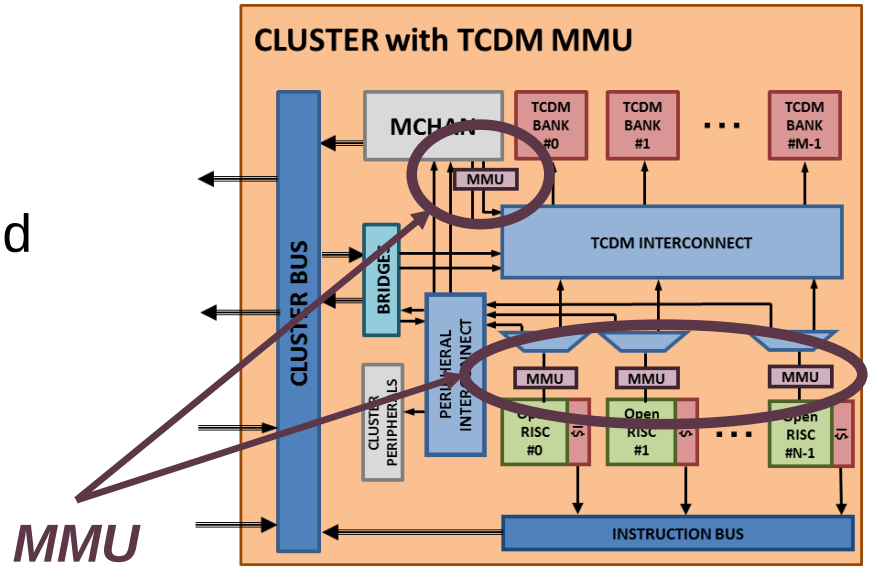
Reconfigurable address mapping

- Support for different address mappings:
  - Interleaved: horizontal shutdown and reduces conflicts in shared segments
  - Non-interleaved mapping: private memory avoids conflicts

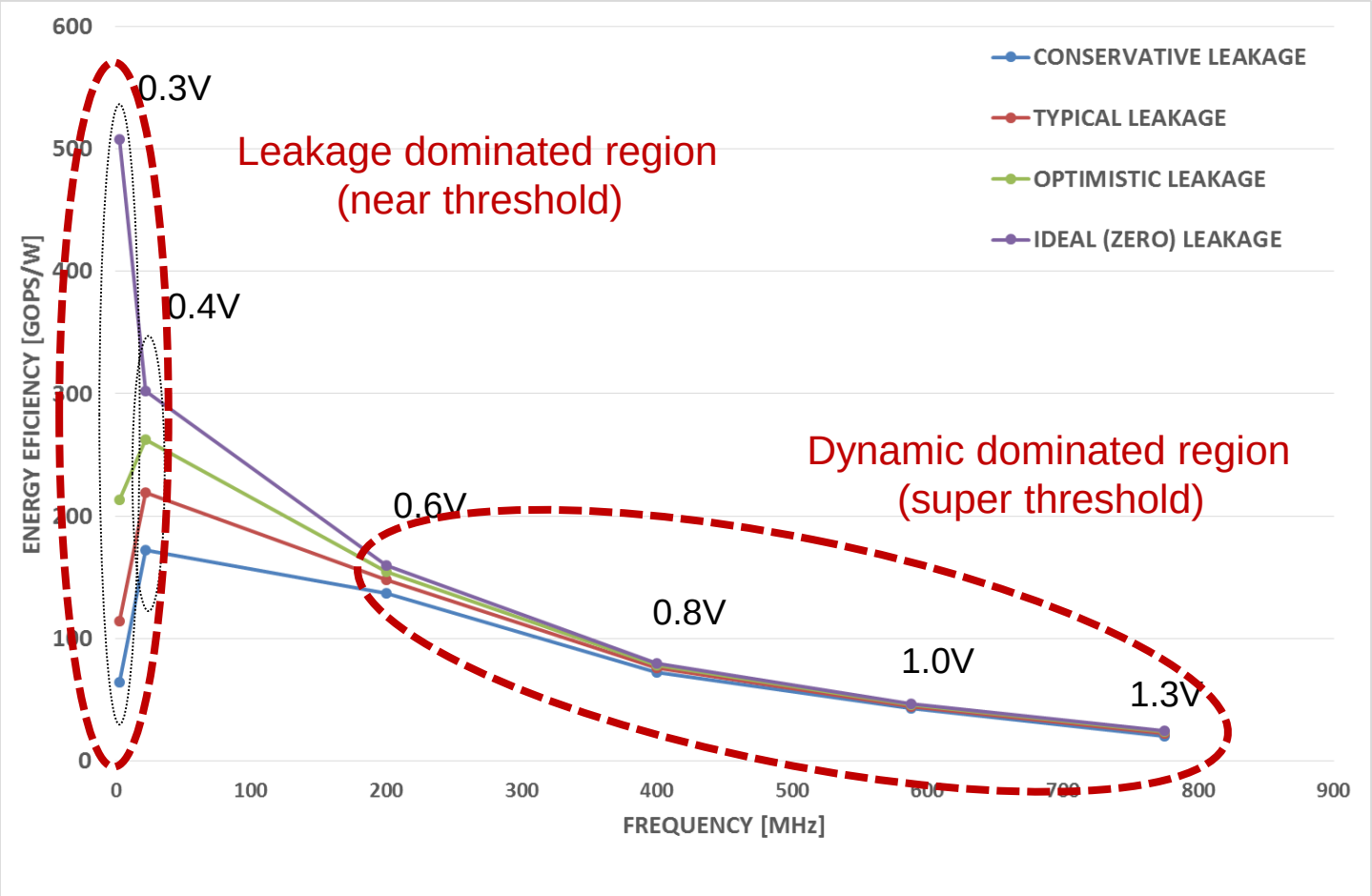
Mapping between logical-physical addresses



- Basic MMU
  - Coexistence of both shared and private memory segments with different address mappings
  - Adapt address mapping is adapted to accommodate partial memory shut-down

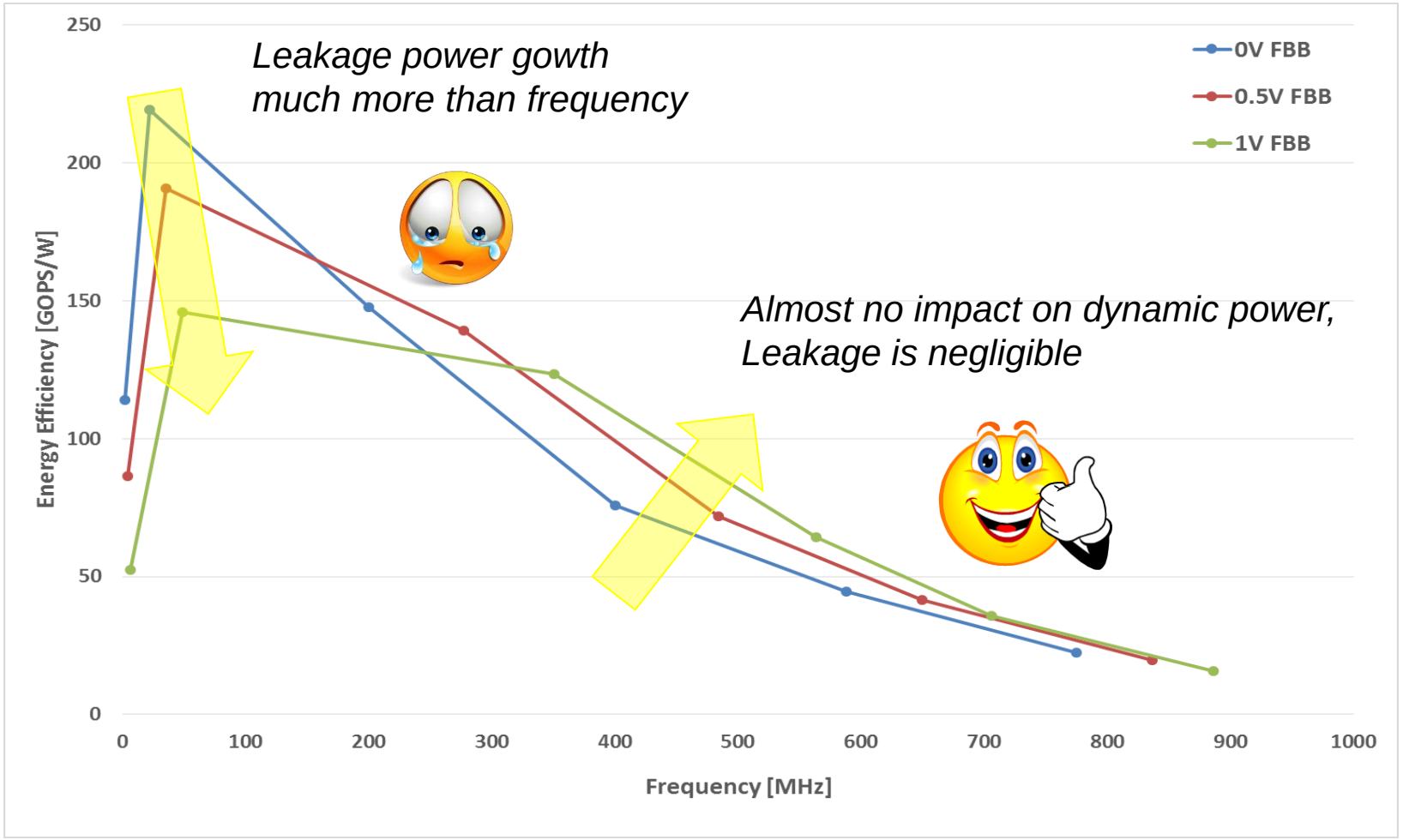


PULP2 Energy Efficiency (Estimated)



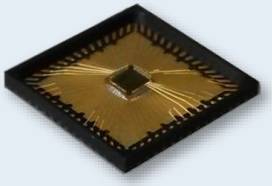
UP TO 220 GOPS/W IN A TYPICAL LEAKAGE SCENARIO!!!

Introducing FBB...

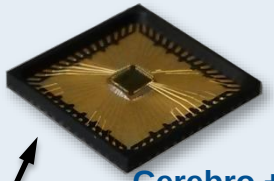


UP TO 1.5X ENERGY EFFICIENCY FOR HIGH WORKLOADS!!!

PULP Family Development



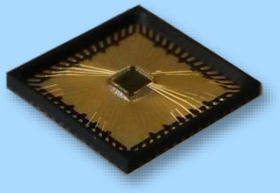
Cerebro v4 130nm
SAR ADC



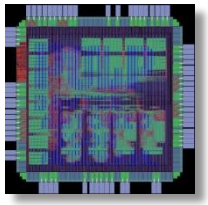
Cerebro + Pulp
130nm
ADC + Multicore cluster

Or10n 180nm
Processor

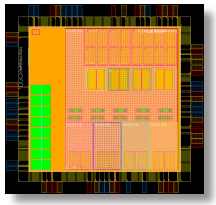
Sir10us 180nm
Processor+ Accelerator



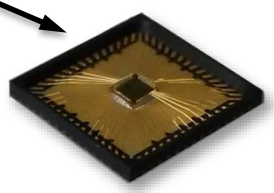
TBD 65nm
Processor+FPU



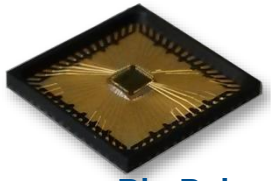
Pulp v1 28nm FDSOI
Body biasing options
0.1 Gops/mW
1.0 Gops @ 250MHz, 0.8V



Pulp v2 28nm FDSOI
Standalone system
0.2 Gops/mW
0.02 Gops @ 22MHz, 0.4V
3.4 Gops @ 850MHz, 1.3V



Pulp v3 28nm FDSOI
High-speed I/O
RISC, DMA v2



Big-Pulp 180nm
Multi-cluster
(IcySoC)

Jan 2014 | Jun 2014 | Jan 2015

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

An ULP Computing Ecosystem

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

HARDWARE IPs

- PROCESSOR
- INTERCONNECT (LOCA
- MEMORY HIERARCHY
  MEMO
- HARDWA
- …

SOFTWARE

- PILER/TOOLCHAIN
- GRAMMING MODELS
- IME

Building an open-source ecosystem for exploring (with silicon!) next-generation parallel computing platforms

SILICON

- OPTIMIZATION FLO
- IMPLEMENTATION F
- VERIFICATION FLOW
- FULL CUSTOM IPS
- …

POLITECNICO MILANO

EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

BUT ...

- SUPPORT FOR DEBUG
- SUPPORT FOR PROFILING
- DESIGN FOR TESTABILITY
- …..

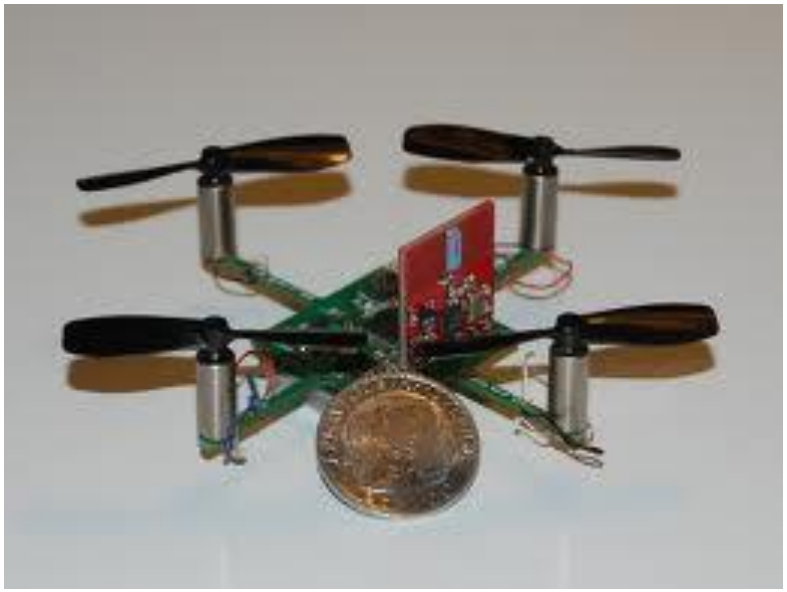
FROM RESEARCH TO INDUSTRY cea tech

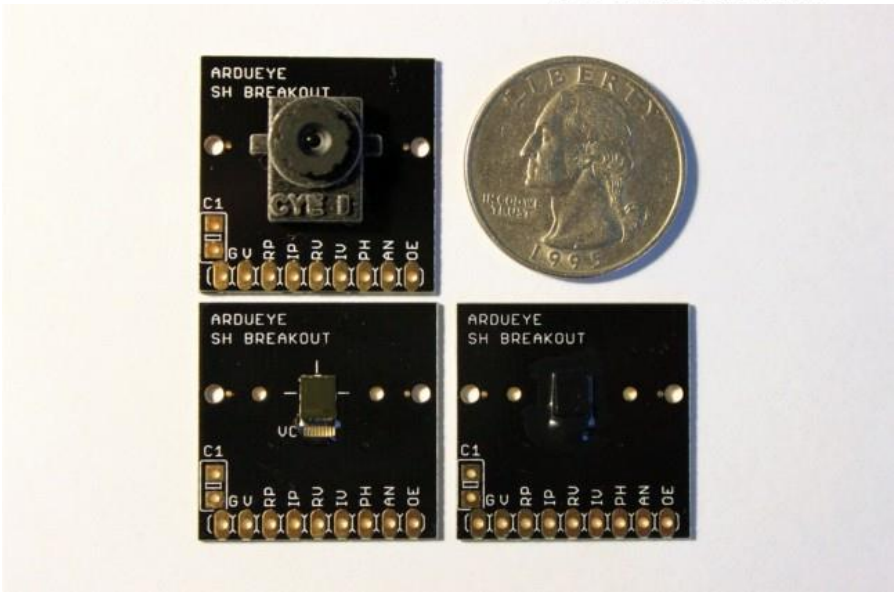
VALIDATION

- L PLATFORM
- EMULATION PLATFORM (FPGA)
- BENCHMARKS
- REGRESSION TESTS
- …

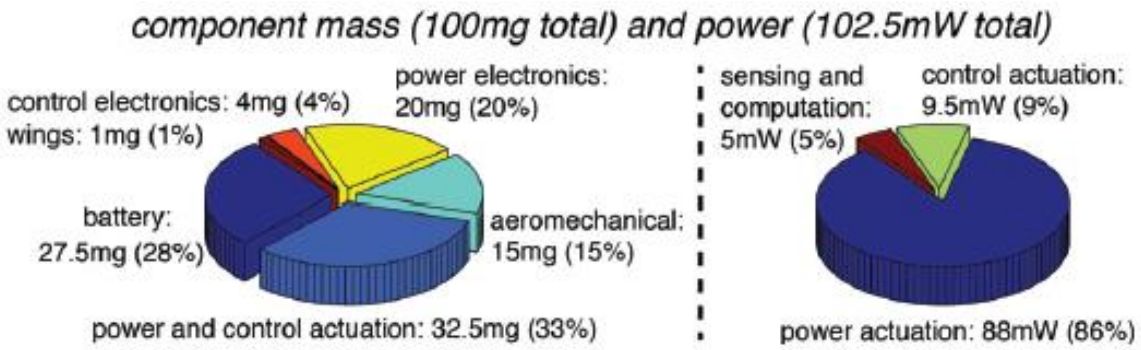
life.augmented

To do what?





112x112pixel (300uW)



[Wood13]

@60fps 0.76MPx/s → with 1KOPS/pixel we need 0.75Gops!

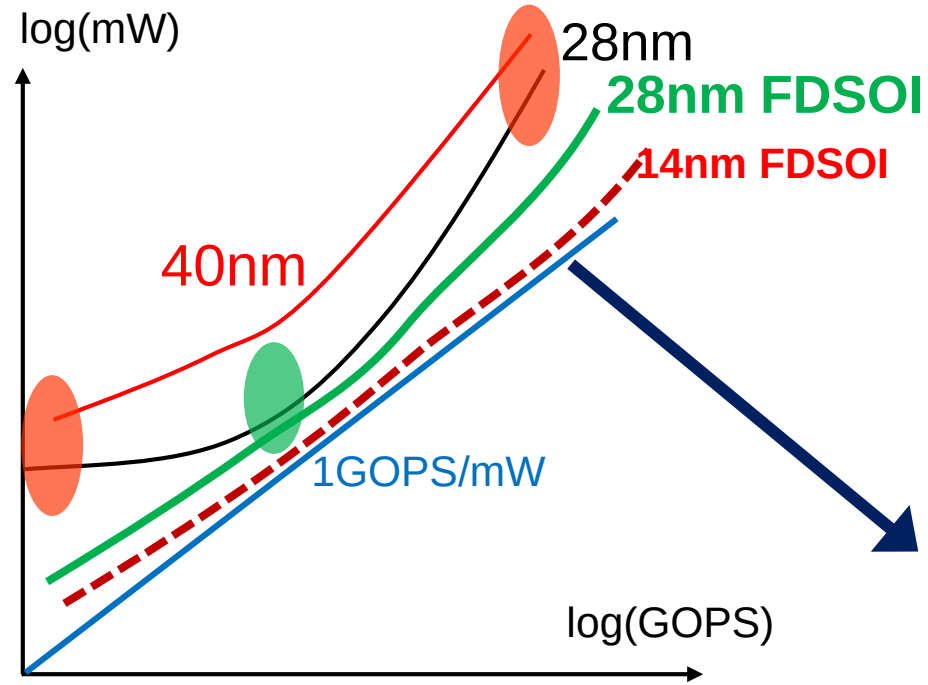
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

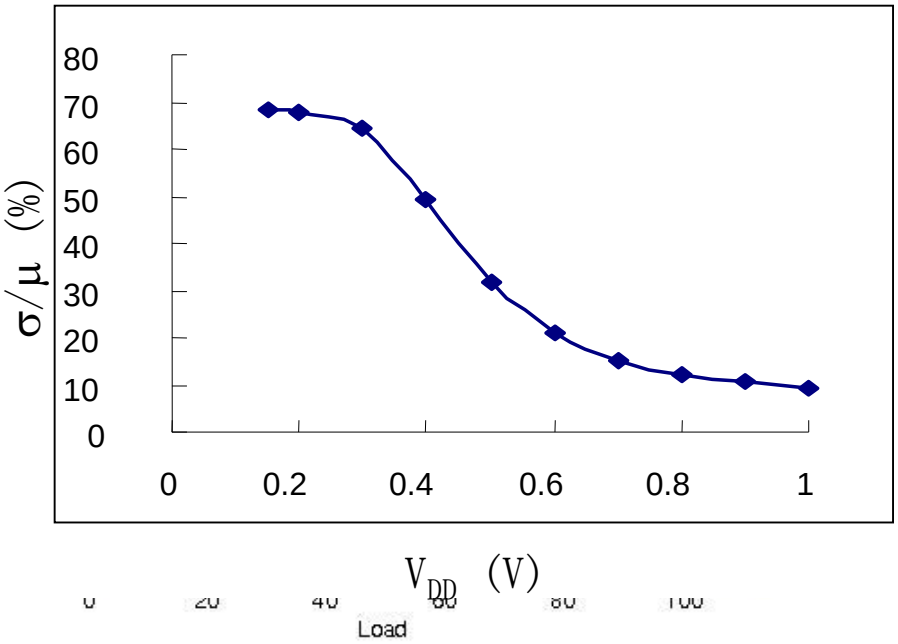
The Grand Challenge: Energy proportionality

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

3,200MOPS/W – 30KW

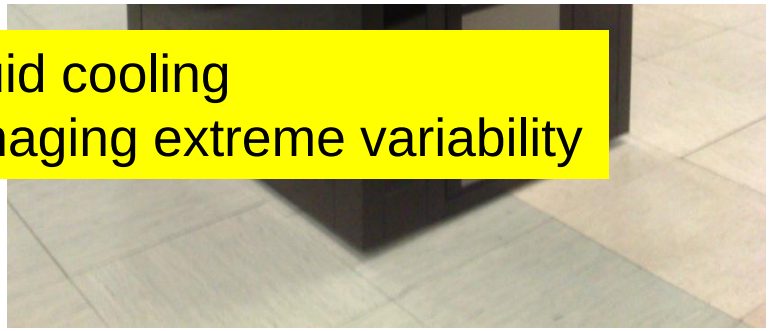
PULP1-2 32b, Kops-Tops @ pJ/W





Goal: reduce «bending up» of energy curve at low & **high** perf!

+ Liquid cooling
+ Managing extreme variability



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Thank you!

erc
European Research Council

Multithermand AdG
Multiscale Thermal Management of Computing Systems