ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Sub-pJ-Operation Scalable Computing

Why, When, How?

Luca Benini

IIS-ETHZ & DEI-UNIBO

erc European Research Council

SEVENTH FRAMEWORK PROGRAMME

nano-tera.ch

http://www.pulp-platform.org

PULP Parallel Ultra Low Power

NXP

life.augmented

SAMSUNG

IoT: Near-Sensor Processing

	INPUT BANDWIDTH	COMPUTATIONAL DEMAND	OUTPUT BANDWIDTH	COMPRESSION FACTOR
Image				
Tracking: [*Lagroce2014]	80 Kbps	1.34 GOPS	0.16 Kbps	500x
Voice/Sound				
Speech: [*VoiceControl]	256 Kbps	100 MOPS	0.02 Kbps	12800x
Inertial				
Kalman: [*Nilsson2014]	2.4 Kbps	7.7 MOPS	0.02 Kbps	120x
Biometrics				
SVM: [*Benatti2014]	16 Kbps	150 MOPS	0.08 Kbps	200x

- ***Extremely compact output (single index, alarm, signature)***
- ***Computational power of ULP µControllers is not enough***
- ***Parallel worloads***

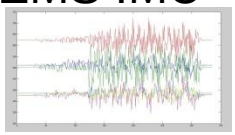
System View

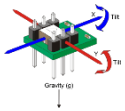
Sense

Analyze and Classify

Transmit

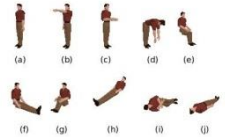
MEMS IMU



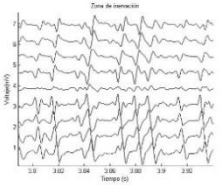


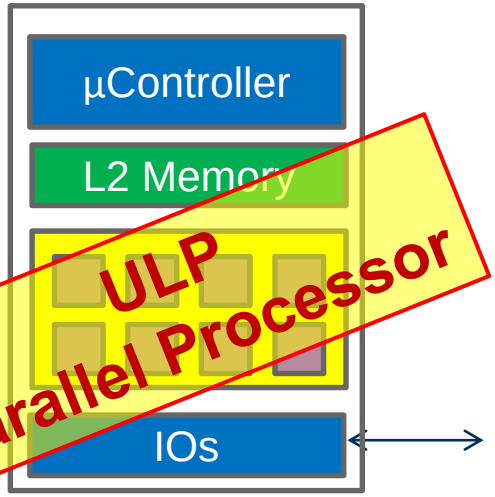
MEMS Microphone

ULP Imager



EMG/ECG/EIT





1 ÷ 2000 MOPS
1 ÷ 10 mW

Short range, BW

Low rate (periodic) data

SW update, commands

Long range, low BW

LoRa SEMTECH

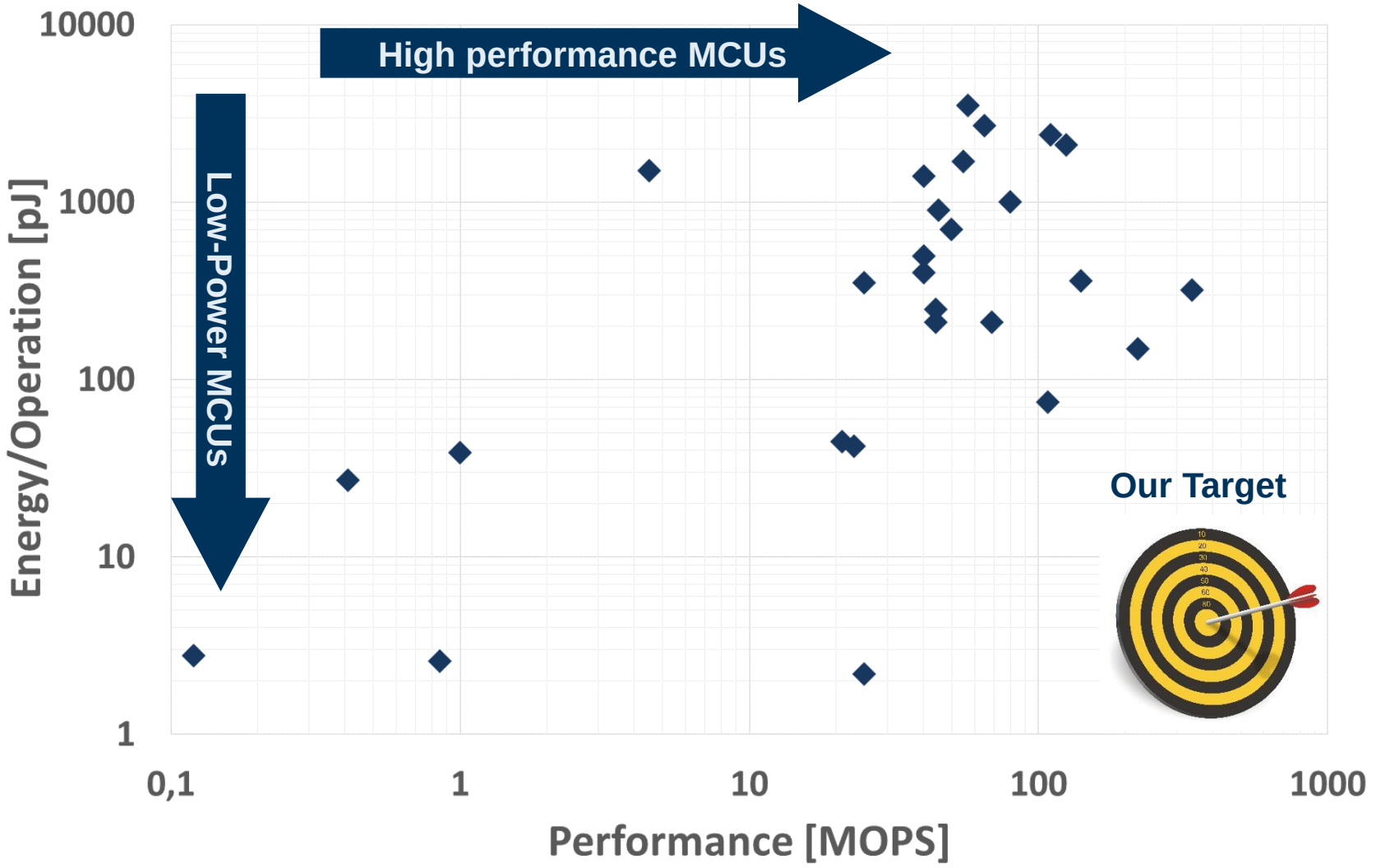
100 µW ÷ 2 mW

Battery + Harvesting powered
→ a few mW power envelope

Idle: ~1µW
Active: ~ 50mW

Microcontrollers Landscape

*not exhaustive



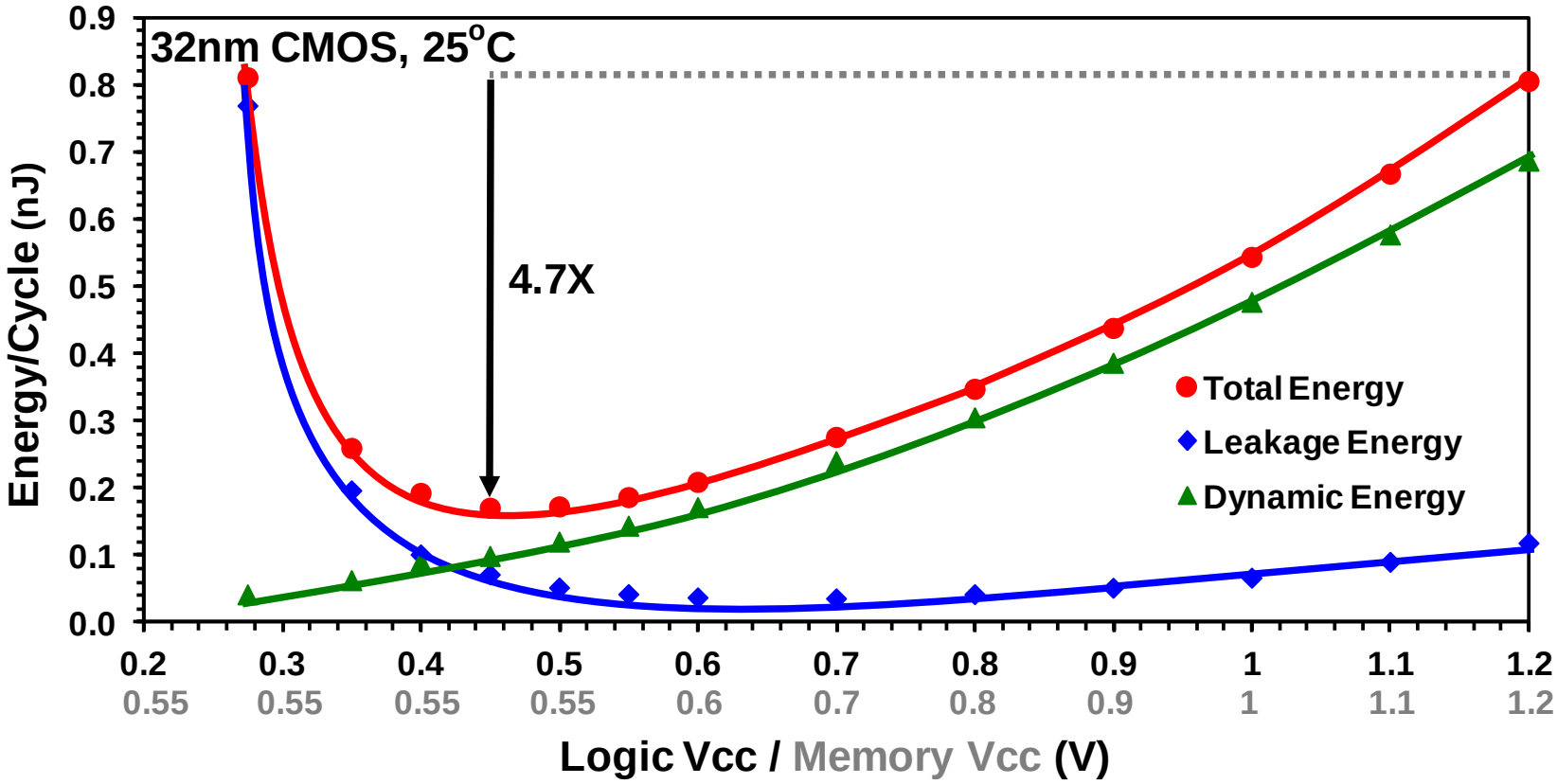
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Near-Threshold Multiprocessing

Minimum energy operation

Source: Vivek De, INTEL – Date 2013



Near-Threshold Computing (NTC):

1. **Don't waste energy pushing devices in strong inversion**
2. **Recover performance with parallel execution**

Near-Threshold Multiprocessing

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

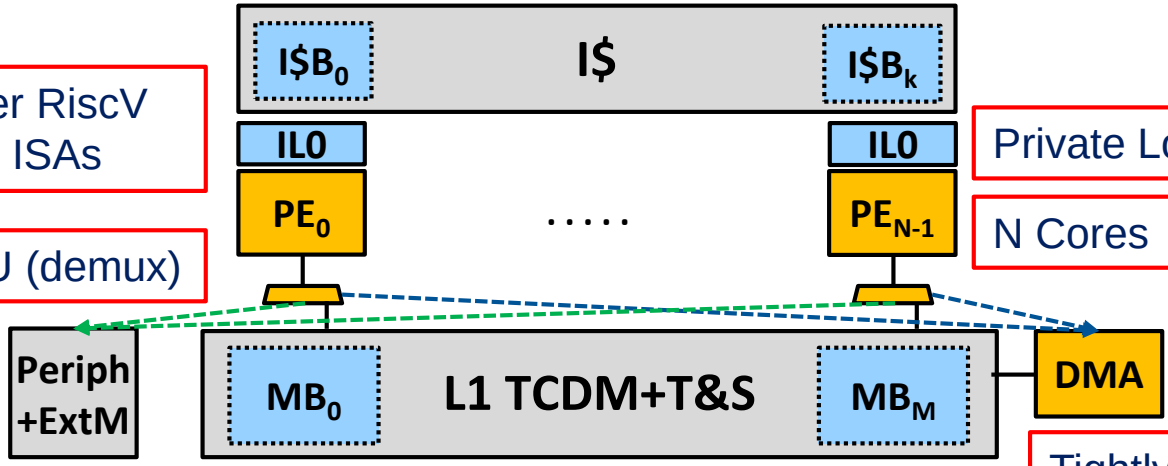
PULP Parallel Ultra Low Power

Ultra-low power scalable computing

Shared L1 I$ with Multi-instruction load

4-stage, in-order RiscV and OpenRISC ISAs

Micro-MMU (demux)



Private Loop/Prefetch Buffer

N Cores

Tightly Coupled DMA

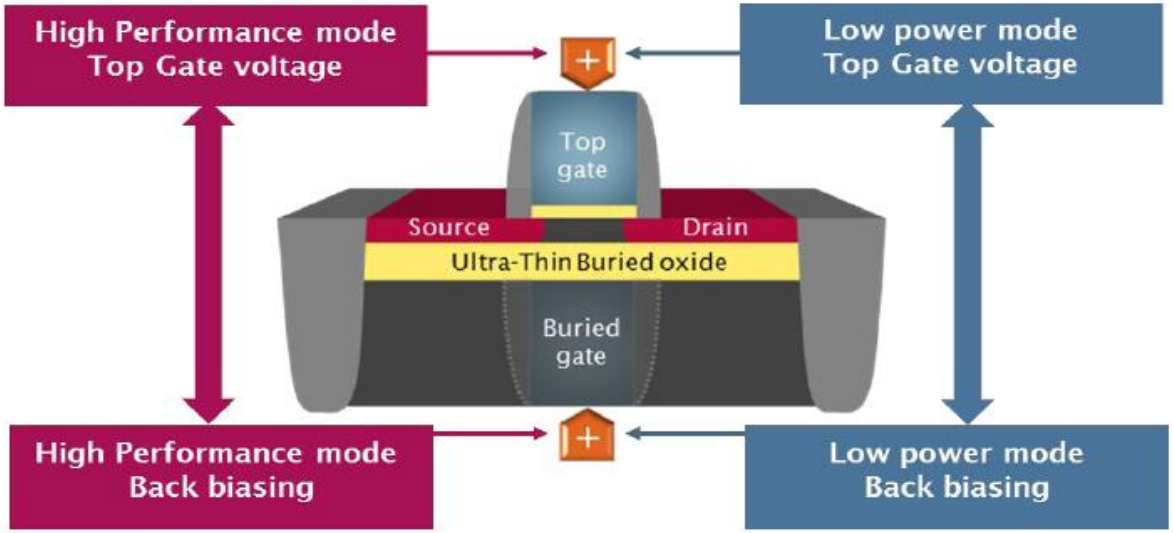
Shared L1 DataMem + Atomic Variables

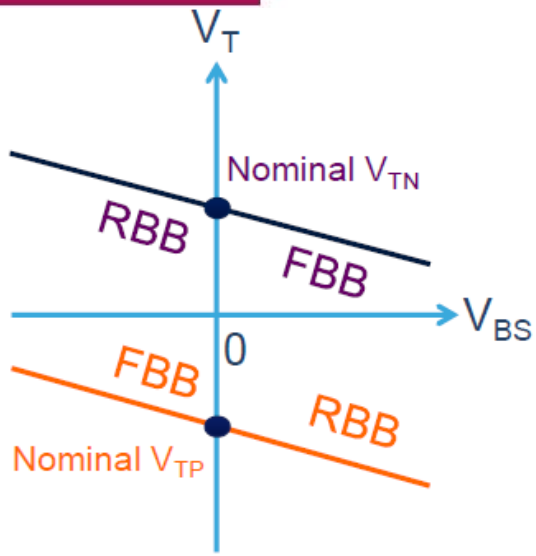
1.. 8 PE-per-cluster, 1…32 clusters

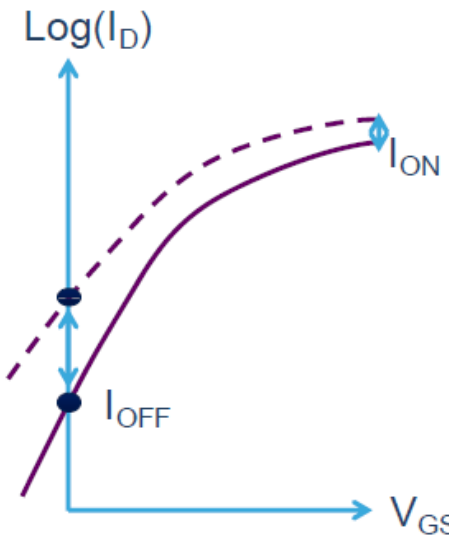
NT but parallel → Max. Energy efficiency when Active

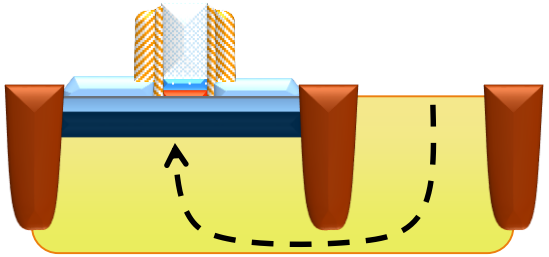
+ strong PM for (partial) idleness

Near threshold FDSOI technology





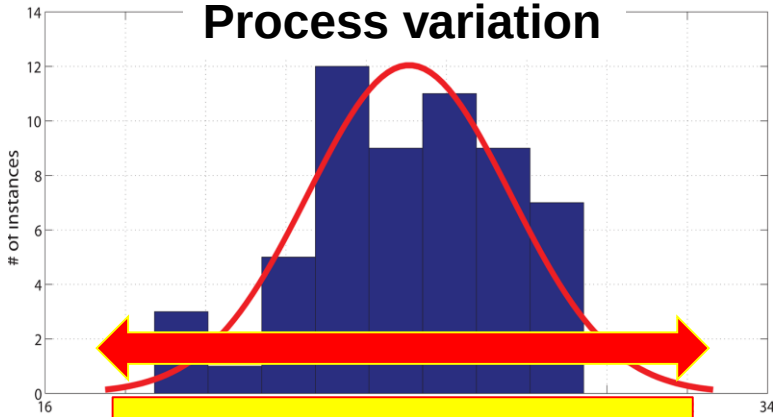




Body bias: Highly effective knob for power & variability management!

Body Biasing for Variability Management

Process variation

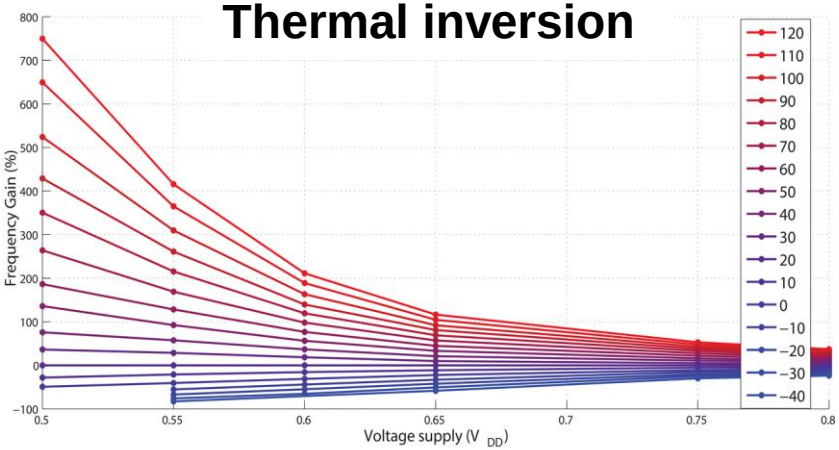


120° C

100x @0.5V

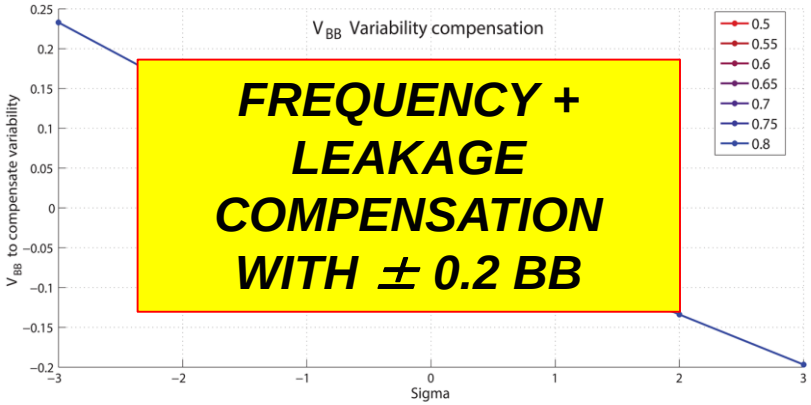
-40° C

Thermal inversion

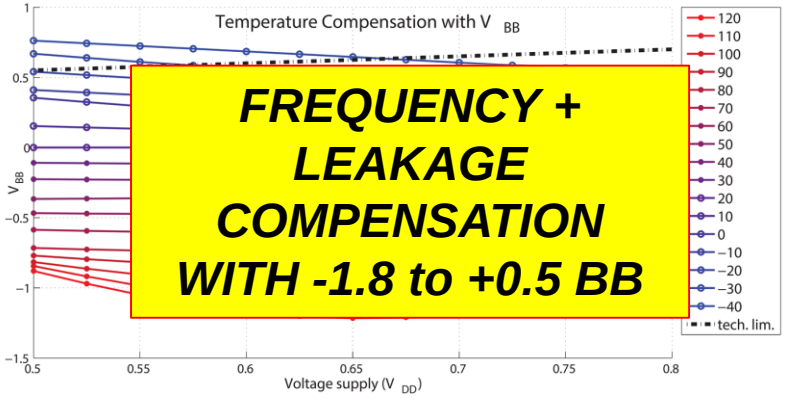


RVT transistors

FBB/RBB



FREQUENCY + LEAKAGE COMPENSATION WITH ± 0.2 BB

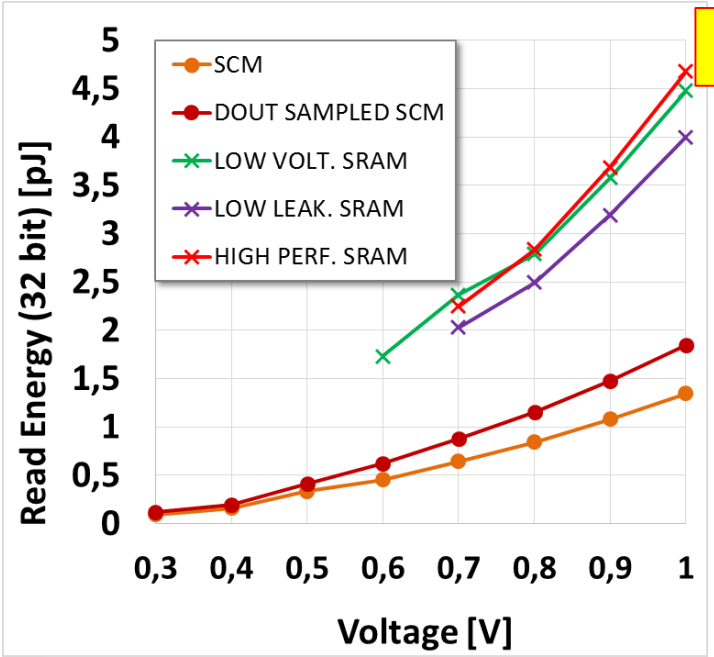


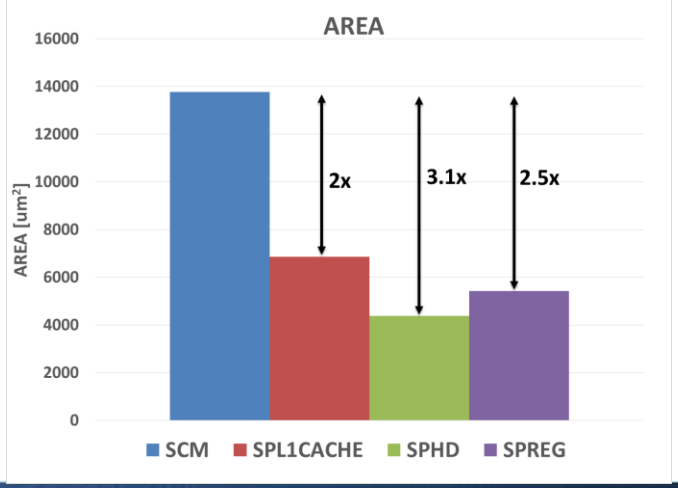
FREQUENCY + LEAKAGE COMPENSATION WITH -1.8 to +0.5 BB

ULP (NT) Bottleneck: Memory

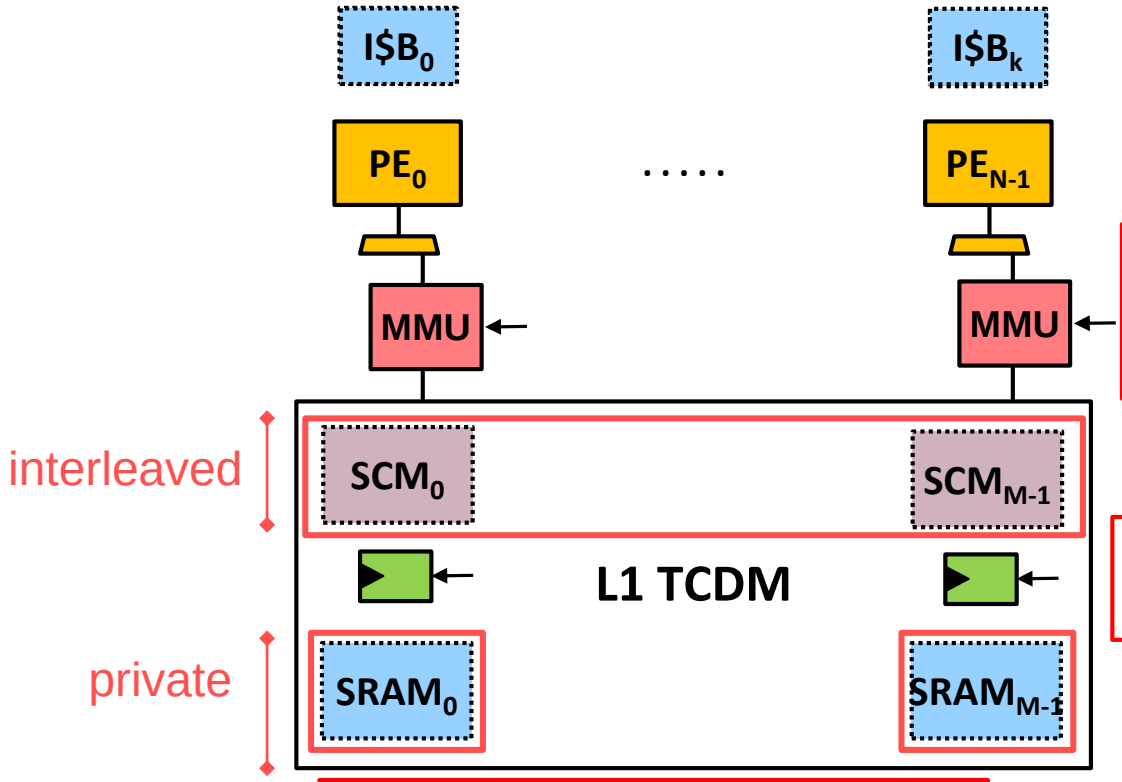
- "Standard" 6T SRAMs:
  - High VDDMIN
  - Bottleneck for energy efficiency
- Near-Threshold SRAMs (8T)
  - Lower VDDMIN
  - Area/timing overhead (25%-50%)
  - High active energy
  - Low technology portability
- Standard Cell Memories:
  - Wide supply voltage range
  - Lower read/write energy (2x - 4x)
  - Easy technology portability
  - Major area overhead (2x)

256x32 6T SRAMS vs. SCM





Heterogeneous + Reconfigurable Memory Architecture

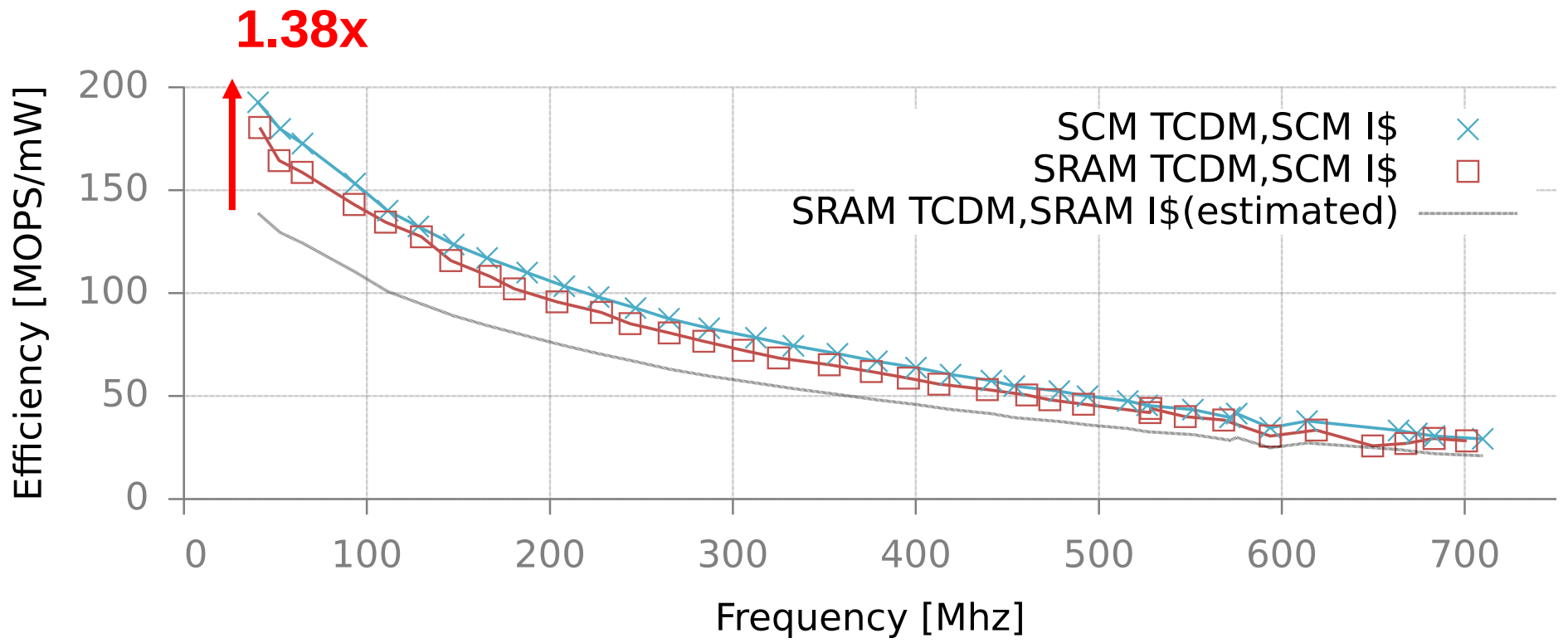


MMU (logical/physical add map):
Interleaved/private addresses
Shutdown of SRAM banks

Reconfigurable Pipeline Stages
for SRAMs degradation@low VDD

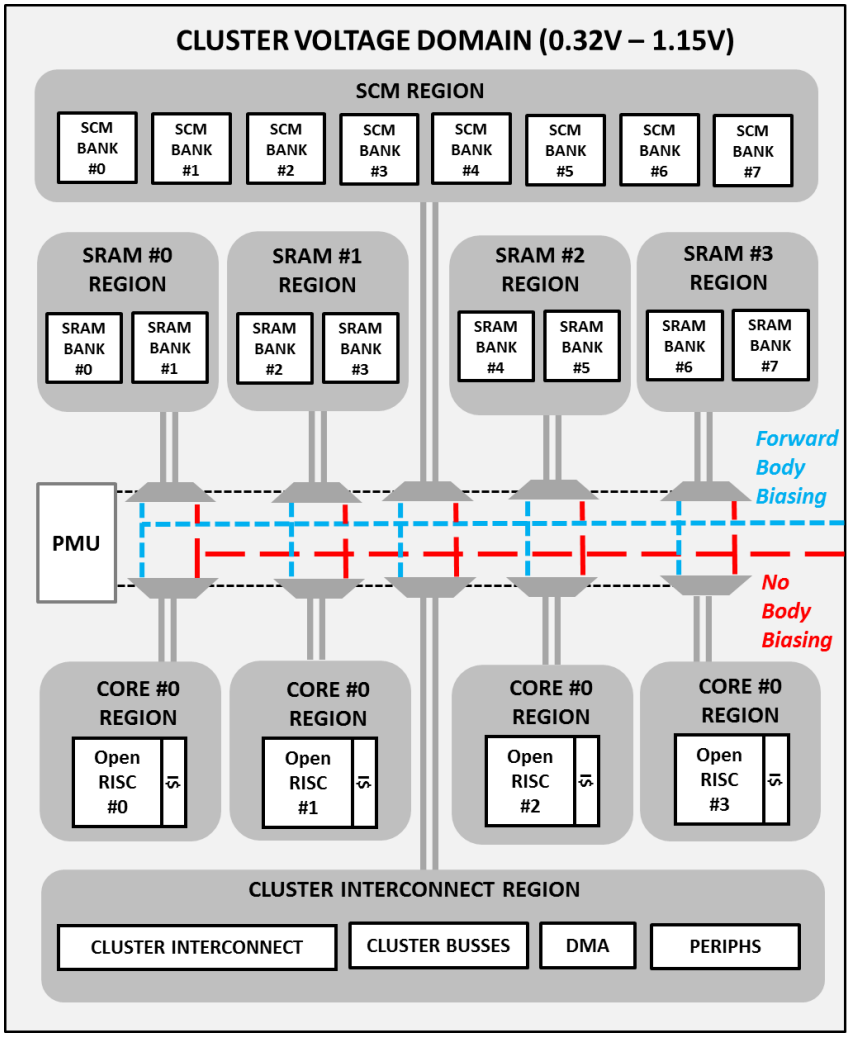
SCM on I$ and part of TCDM
to widen the operating range and
boost energy efficiency,
SRAM for density

Heterogeneous Memory Energy Efficiency Boost



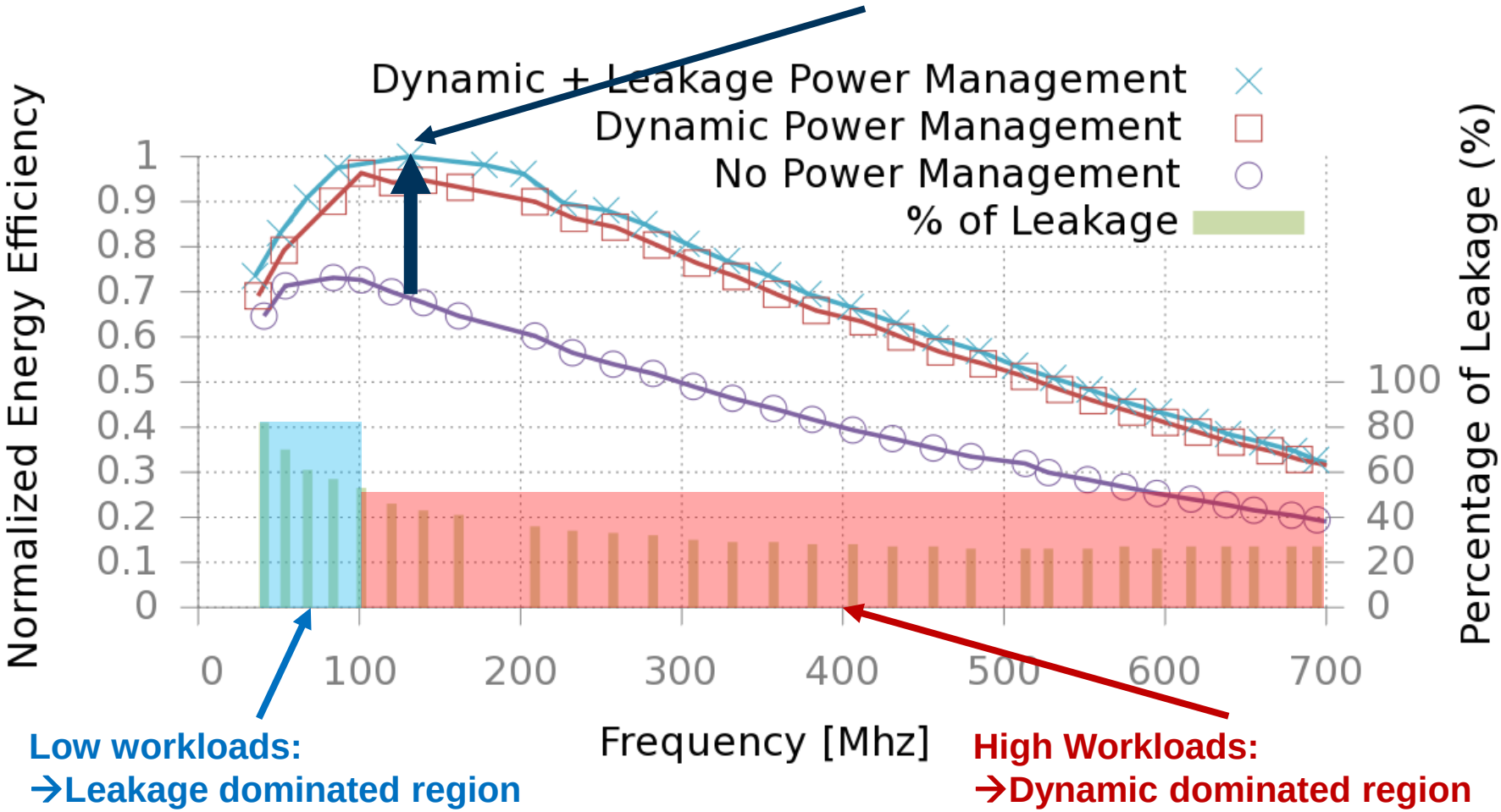
Selective FBB

- The cluster is partitioned in separate **clock gating and body bias regions**
- **Body bias multiplexers (BBMUXes)** control the well voltages of each region
- A **Power Management Unit (PMU)** automatically manages transitions between the operating modes
- Power modes of each region:
  - **Boost mode:** active + FBB
  - **Normal mode:** active + NO BB
  - **Idle mode:** clock gated + NO BB



Sequential processing with selective FBB

Energy efficiency improvement in best energy point: 1.6x

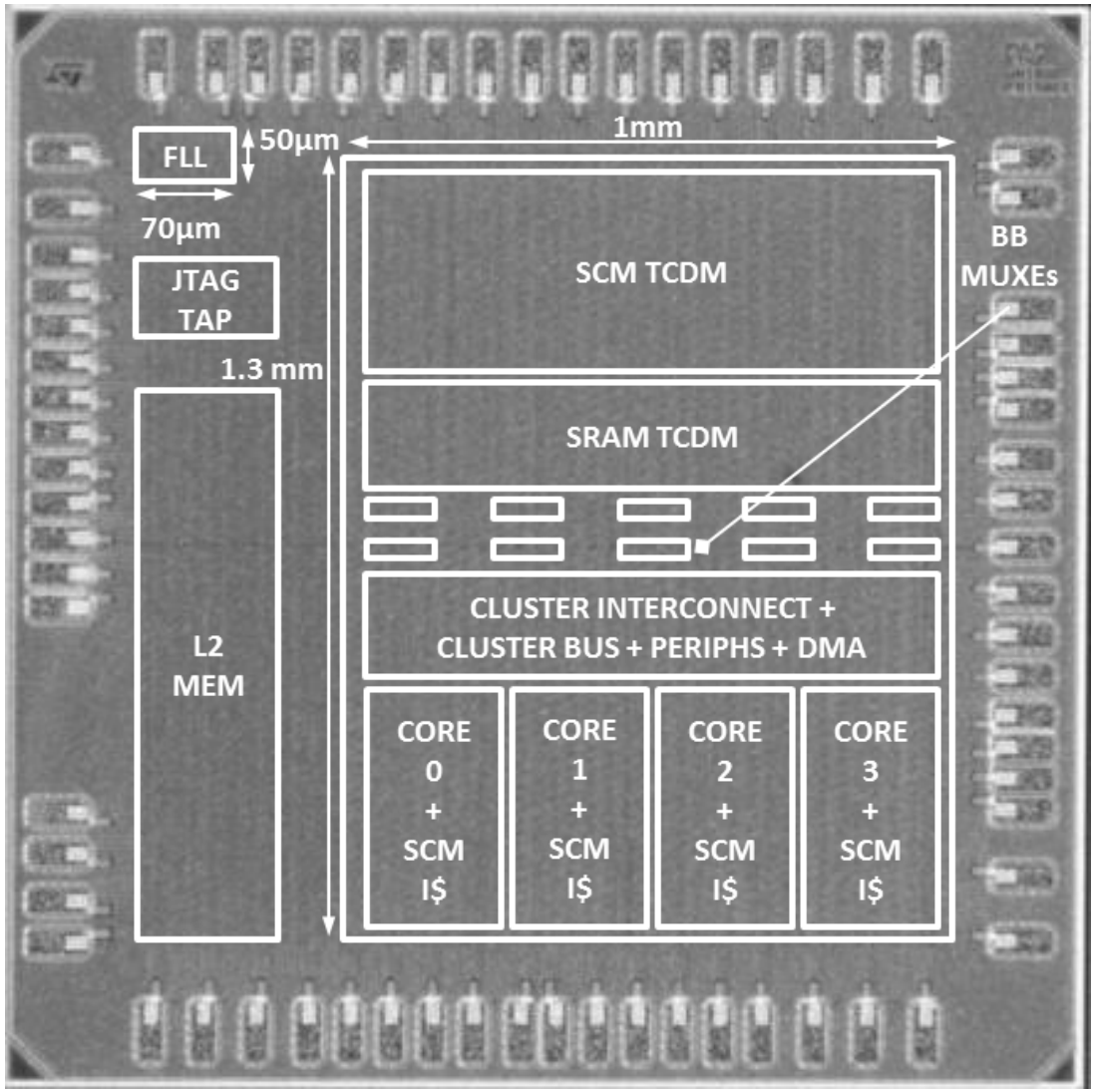


Low workloads:
→Leakage dominated region

High Workloads:
→Dynamic dominated region

Sequential Processing, Temperature=25° C, FBB=1.75V

Silicon Results

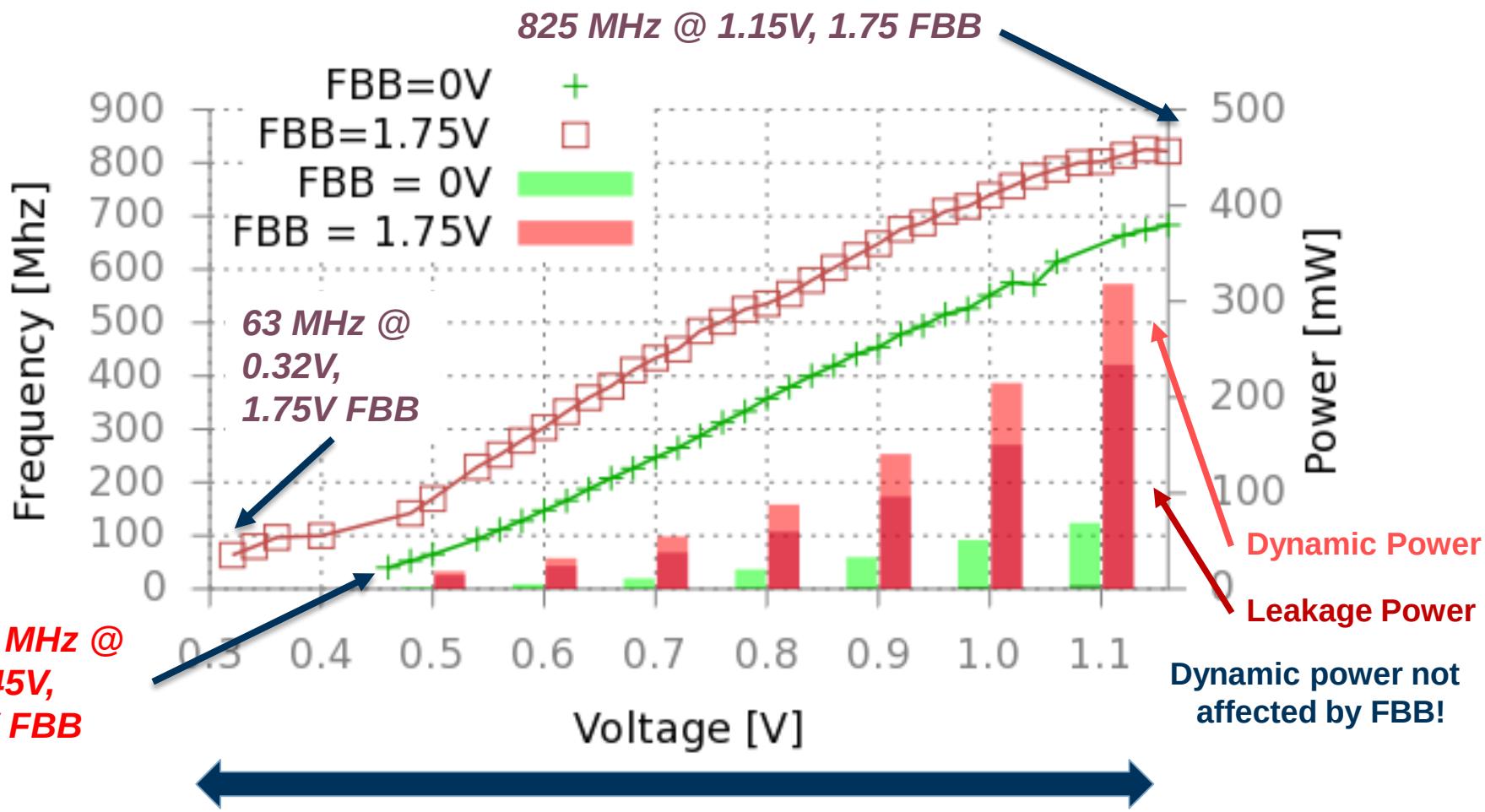


Technology	**UTBB FD-SOI 28nm**
Transistors	**Flip well** **L = 24 nm**
Cluster area	**1.3 mm²**
VDD range (memories)	**0.32V - 1.15V** **(0.45 – 1.15V)**
BB range	**0V - 1.75V**
SRAM macros	**8 x 32 kbit (TCDM)**
SCM macros	**16x4 kbit (TCDM)** **4x 2x4 kbit (I$)**
Gates	**200K**
Frequency range	**NO BB: 40.5-710 MHz** **MAX FBB: 63.5 - 825 MHz**
Power range	**NO FBB: 0.56 - 85 mW** **MAX FBB: 6.9 - 480 mW**

Hot Chips 15 V1 → Cool Chips 16 V2

193MOPS/mW @162MOPS → ~5pJ/OP

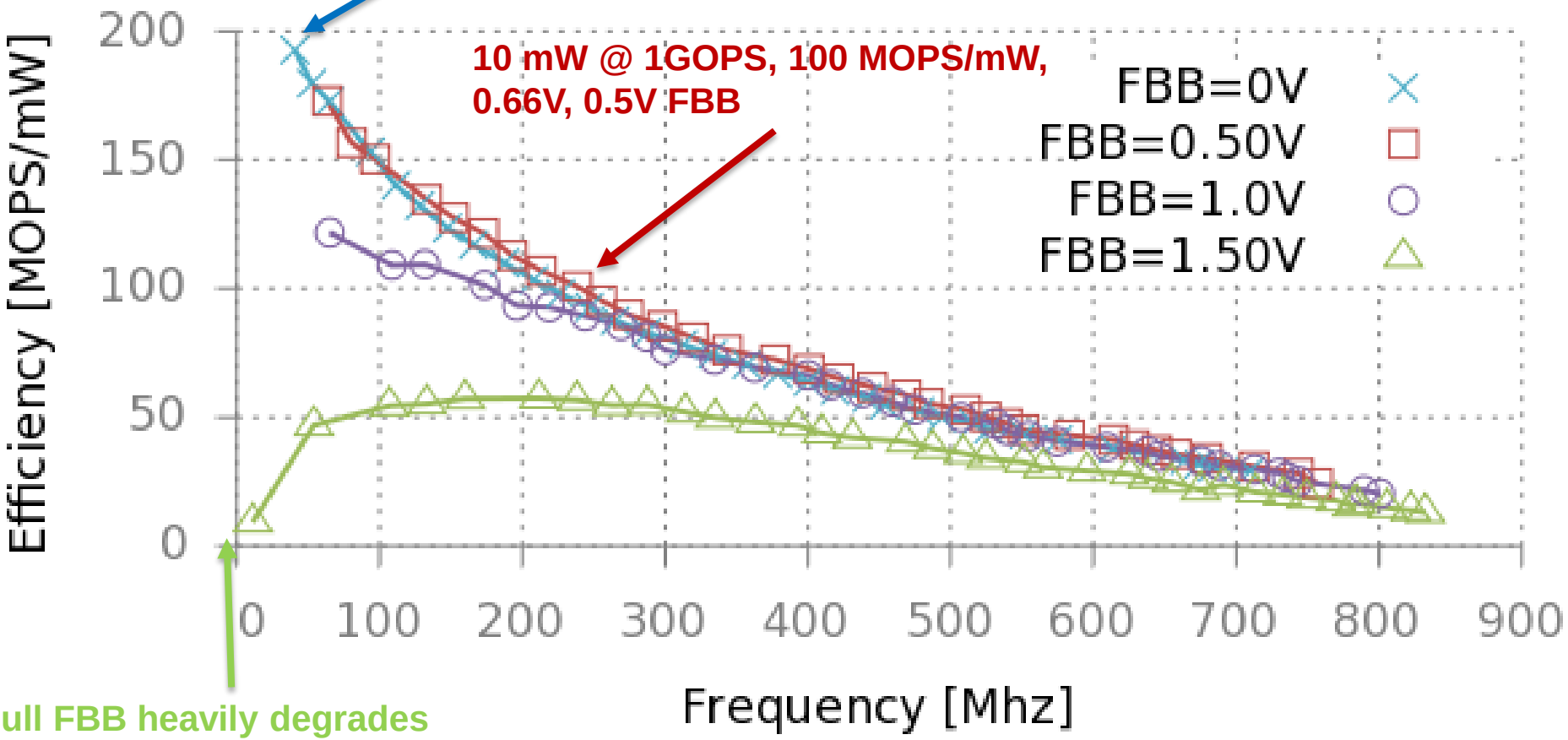
Cluster Performance



Ultra-wide voltage range (0.32V – 1.15V)

Cluster Energy Efficiency

193 MOPS/mW @ 40.5MHz, 0.46V, 0V FBB, 840 µW



Full FBB heavily degrades energy efficiency at low voltage due to high Leakage!

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Approximate Computing to the Rescue

Approximate?

Less-than-perfect results perceived as correct by the users
e.g. image processing (filtering)

RGB to GRAYSCALE

RGB to GRAYSCALE (+ 10% error)

Approximation is not always acceptable
→ **Application and program phase dependent!**

Approximate Storage?

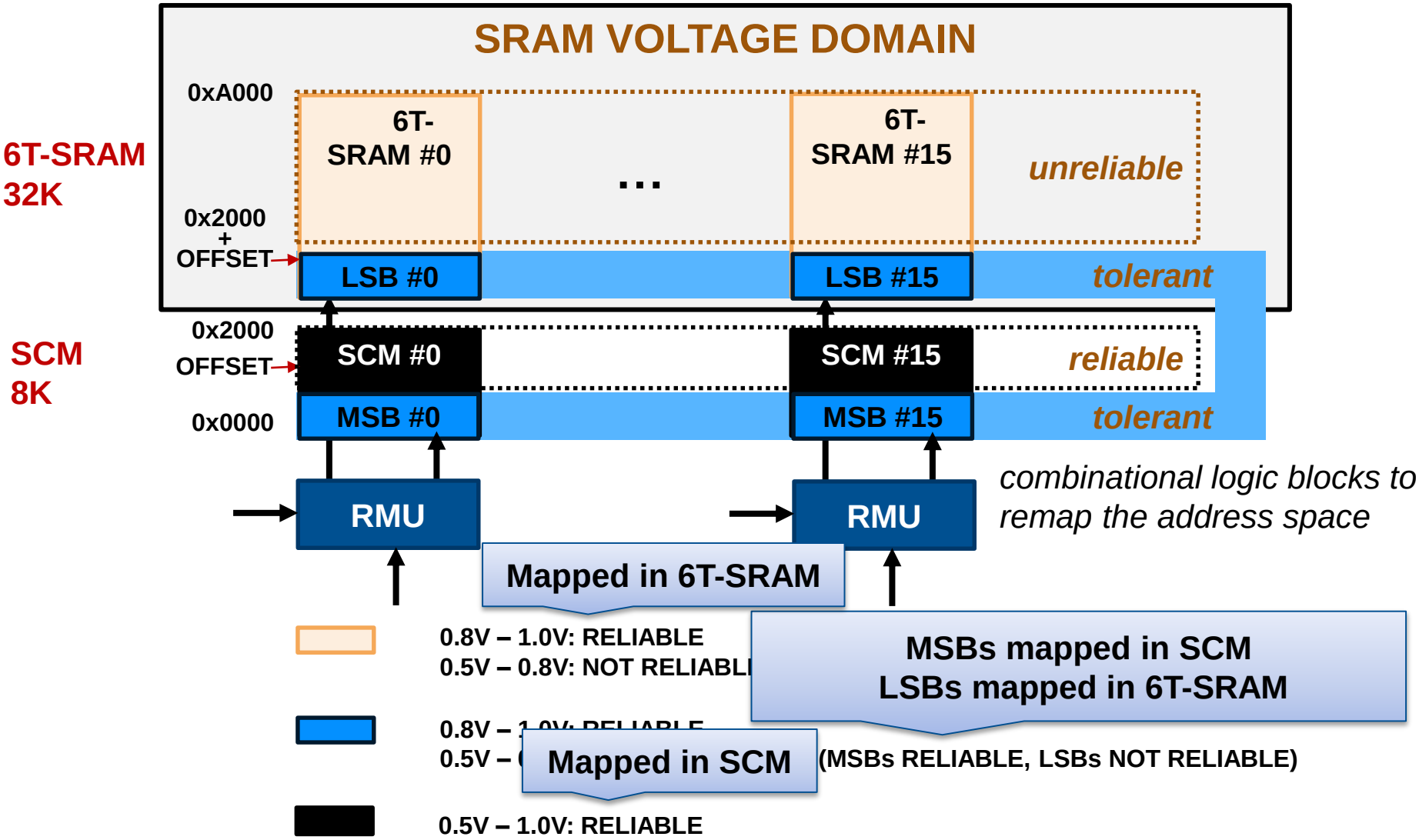
- **Retention voltage**

	Retention
SCM	0.25V
6T-SRAM	0.29V

- Probability of **flip-bit error** on a single bit during read/write operations

Voltage (V)	0.50	0.55	0.60	0.65	0.70	0.75	0.80
P(flip-bit) SCM	0.0	0.0	0.0	0.0	0.0	0.0	0.0
P(flip-bit) 6T	**0.0037**	**0.0012**	**0.0003**	**5.24e-5**	**4.35e-6**	**4.16e-8**	0.0

TCDM layout & address space



Programming model

- To conveniently control energy-efficient data mapping with an approximation model, we propose an ***extension to OpenMP***

- Two additional directives:
  - **#pragma tolerant var** → coupled to a *variable declaration* to specify it tolerates approximation
  - **#pragma tolerant computation** → coupled to a *program statement* to specify the memory addressed therein can be accessed in tolerant (low-voltage) mode

Example


```
int main()
{
  #pragma omp tolerant var
  int sparse_A[];
  int i=0, index;

  while(func(i))
  {
    …
    index = compute_index(i);
    #pragma omp tolerant computation
    sparse_A[index] = compute_element(sparse_A, index);
    …
    update(i);
  }
  use_value(sparse_A);
}
```

Marking index as tolerant would probably lead to fatal errors!

Tolerant computation here would probably lead to fatal errors!

Savings and Accuracy

- System-level Energy savings

	1 domain	2 domains		4 domains	
		Average	Best	Average	Best
Area (μm^2)	2943648	2968768		3062400	
Normalized energy	1.00	0.91	0.80	0.88	0.72
Normalized area	1.00	1.01		1.04	
Norm en. × area	1.00	0.92	0.81	0.92	0.81

10-30%

- Accuracy «Max MSE» → for max end-to-end loss <1%

Benchmark	Zero-ing	Flip-bit	Max MSE
Color Tracking	676	64	225
HOG	2.12E+13	58564	2814663
CNN	17114769	26244	36864
Health	6867734	378	25000
Navi	1681	36	100

650x

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

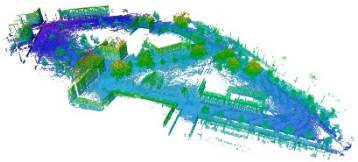
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Approximate ≠ Low Quality!!

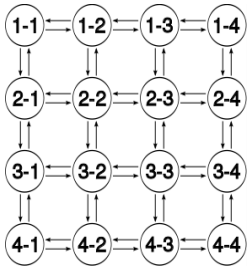
A new viewpoint on approximation

- A practical example: real-time UAV path planner

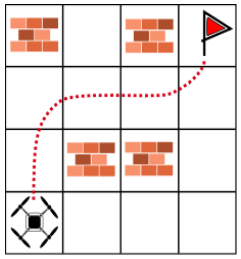
UAV with sensors



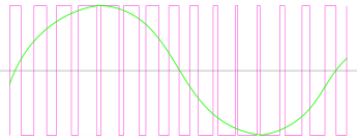
Occupancy Grid



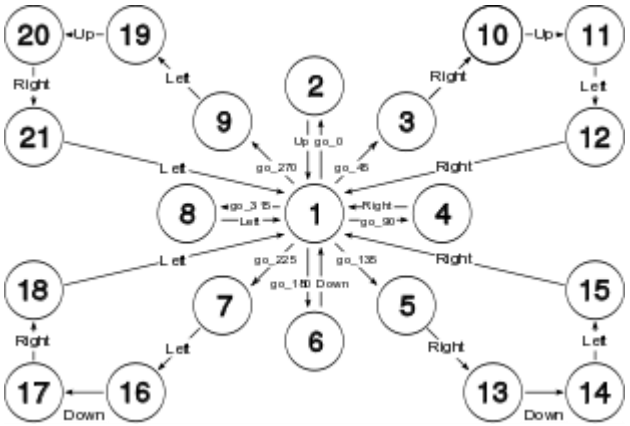
Oriented Graph



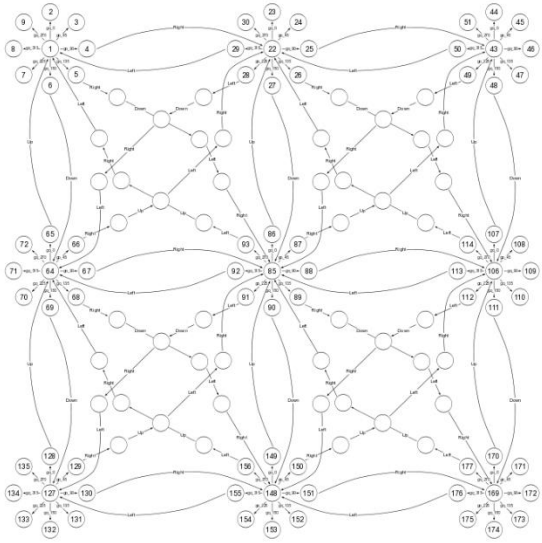
Path Planning



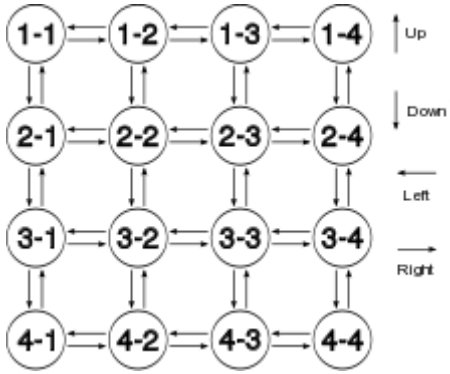
PWM Signal



Quadrotor Automaton



Automaton Synchronous Composition (map 3x3)

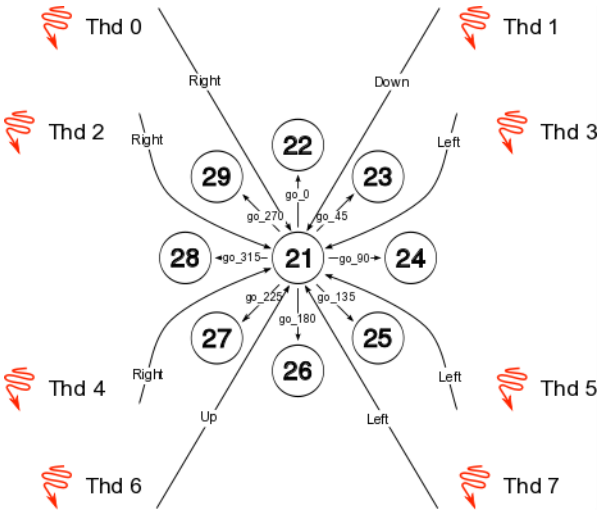


Map Automaton

Optimal Trajectory Computation

- Parallel Dijkstra algorithm
  - Each thread (in parallel)
    - handles several nodes “to be explored” (reference nodes)
    - for each reference node, explores all the neighbors
    - for each neighbor, eventually, updates both cost and predecessor id

Potential race condition → non-optimal and **non-deterministic** path



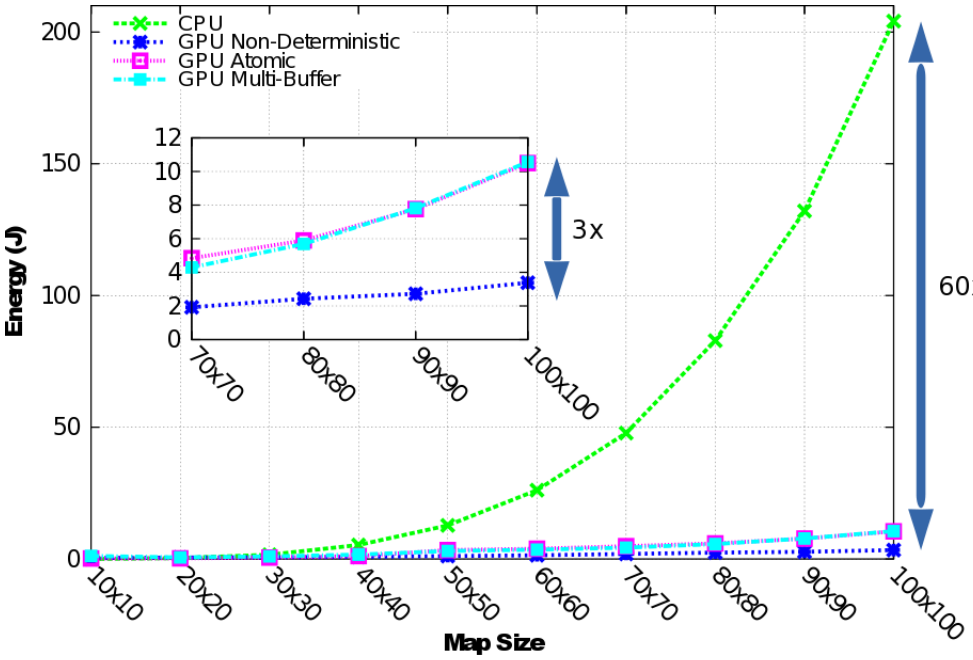
3 different approaches:

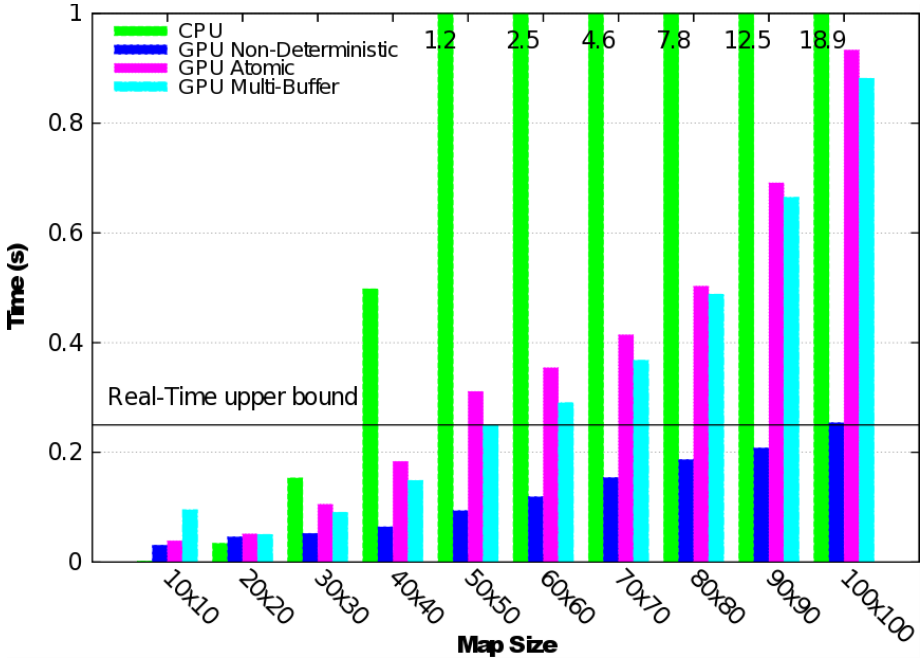
1. Data packing + keep non-determinism: *Non-Determistic version*
2. Data packing + OpenCL atomic operation: *Atomic version*
3. Data packing + extra-size cost buffer, selecting minimal value in a second stage: *Multi-Buffer version*

Approximate is better!

- Sequential << Parallel << Parallel+Approximate

Of course energy is better, but also…





Plan quality is much better if the surveyed area (within the RT bound) is larger!

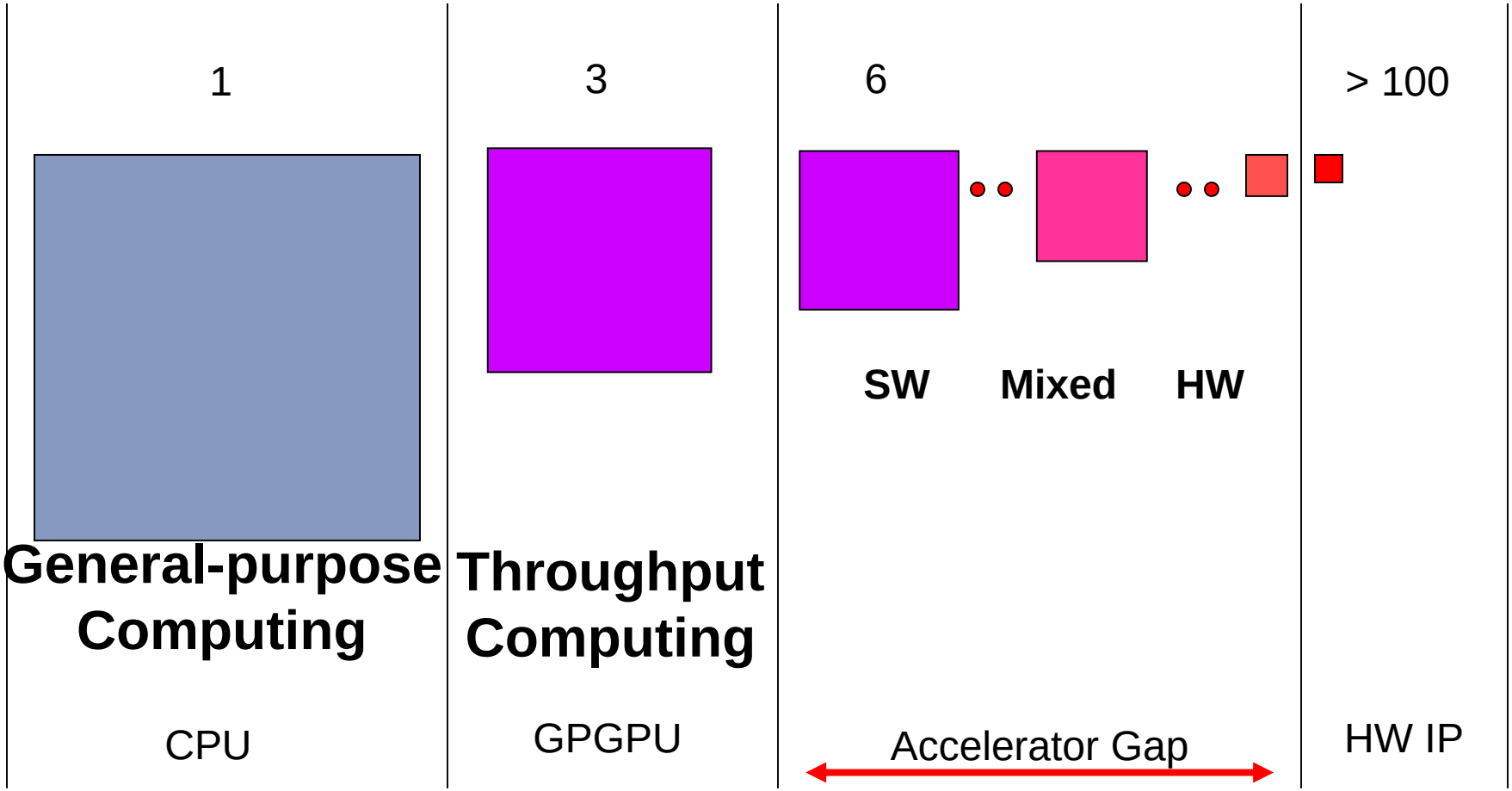
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Pushing Beyond pJ/OP

Recovering more silicon efficiency

GOPS/W



Closing The Accelerator Efficiency Gap with Agile Customization

Learn to Accelerate

- Brain-inspired (deep convolutional networks) systems are high performers in many tasks over *many domains*

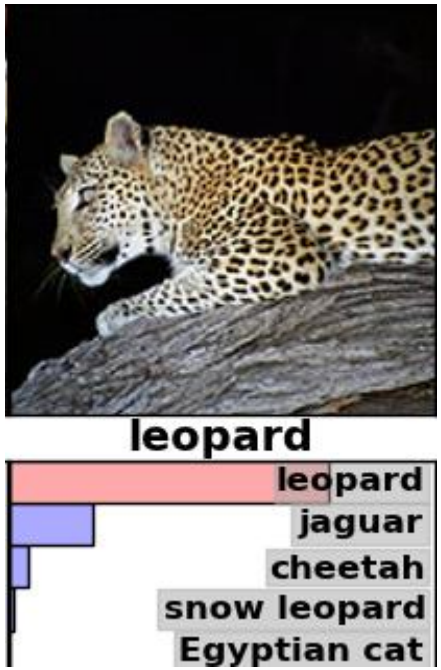
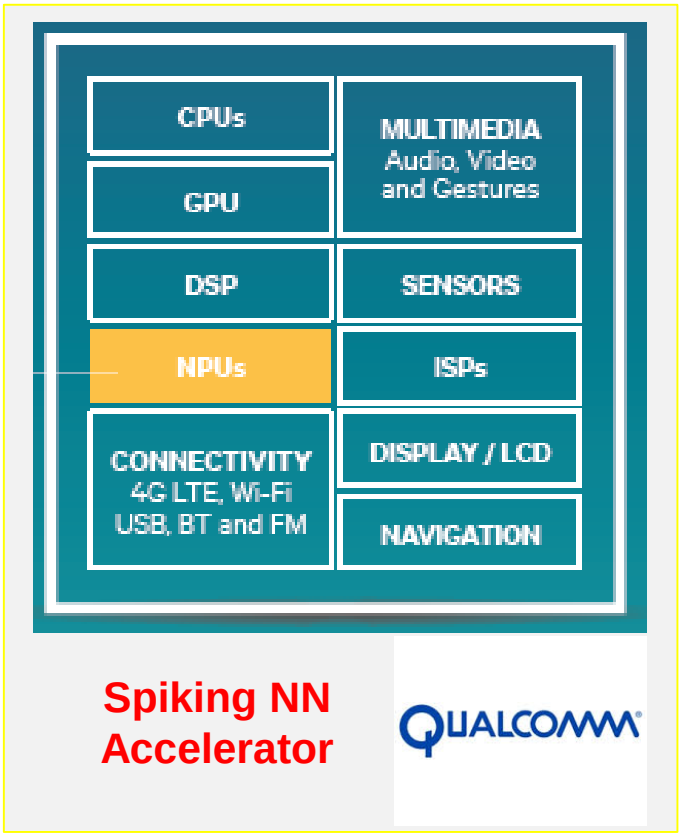
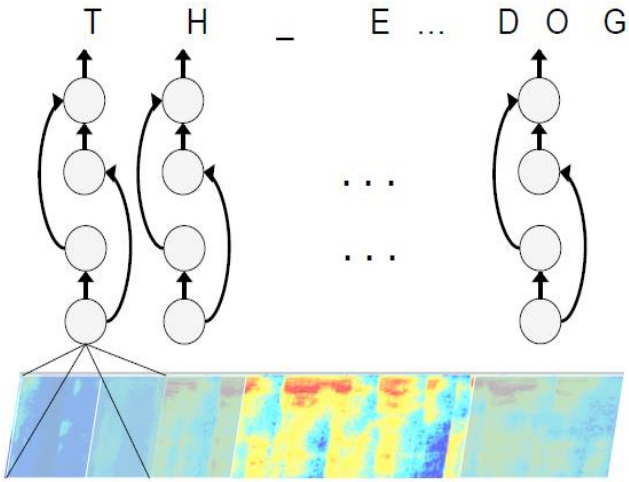


Image recognition
[Russakovsky et al., 2014]



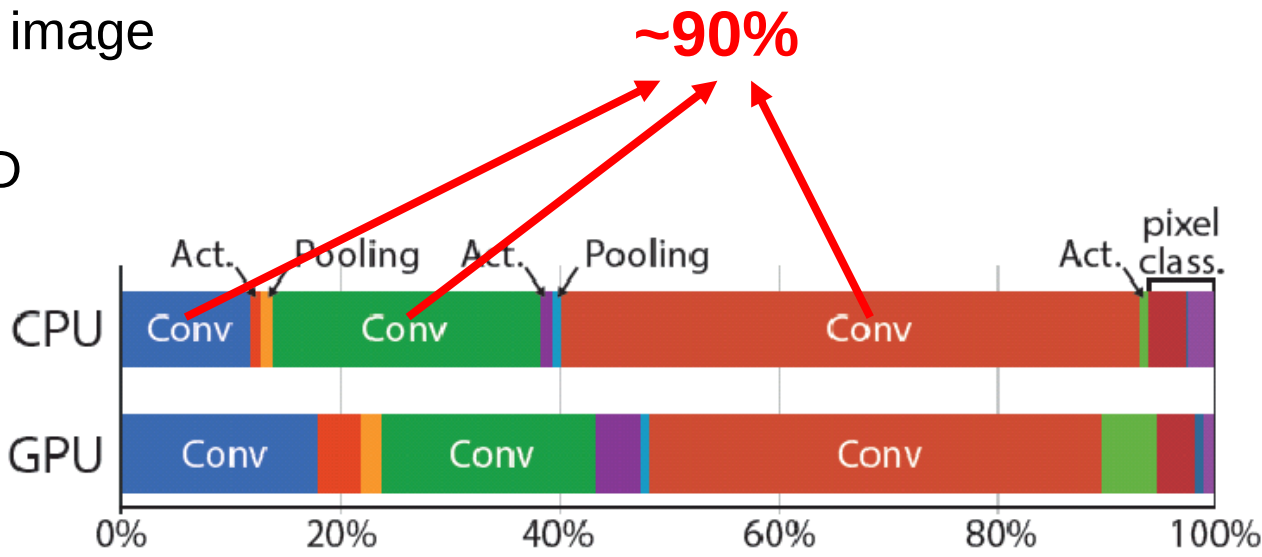


Speech recognition
[Hannun et al., 2014]

- **Flexible** acceleration: learned CNN weights are “the program”

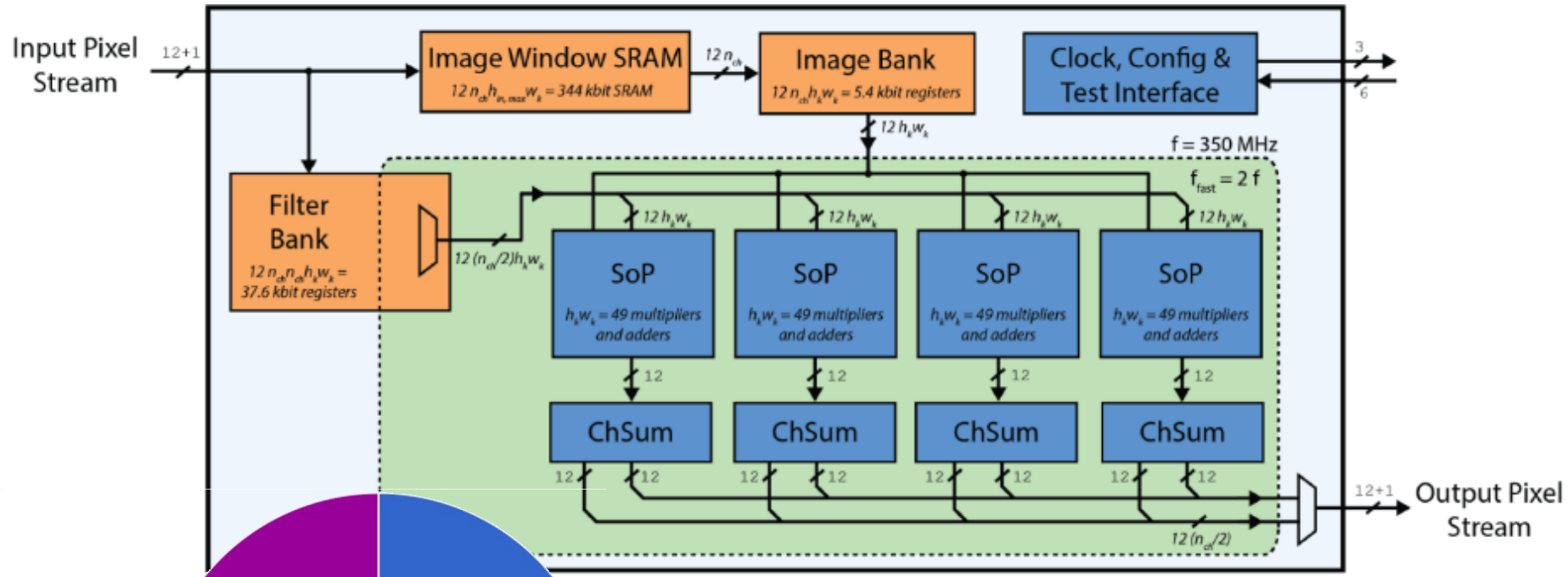
Computational Effort

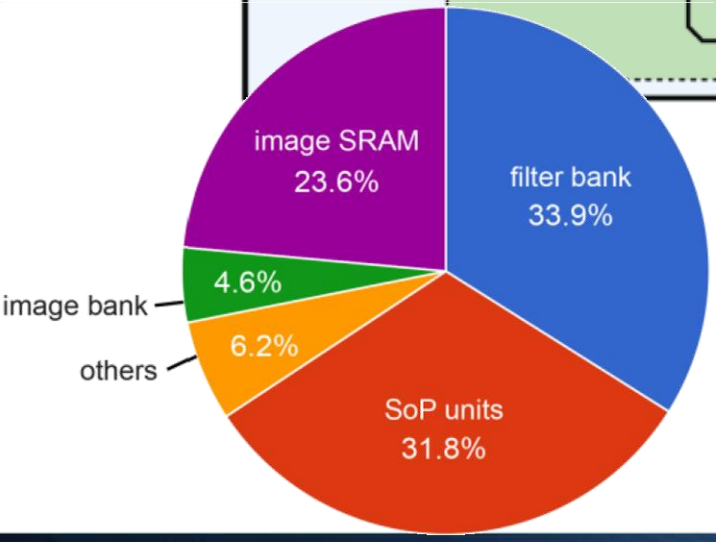
- Computational effort
  - 7.5 GOp for 320x240 image
  - 260 GOp for FHD
  - 1050 GOp for 4k UHD



Origami chip

Origami: A CNN Accelerator





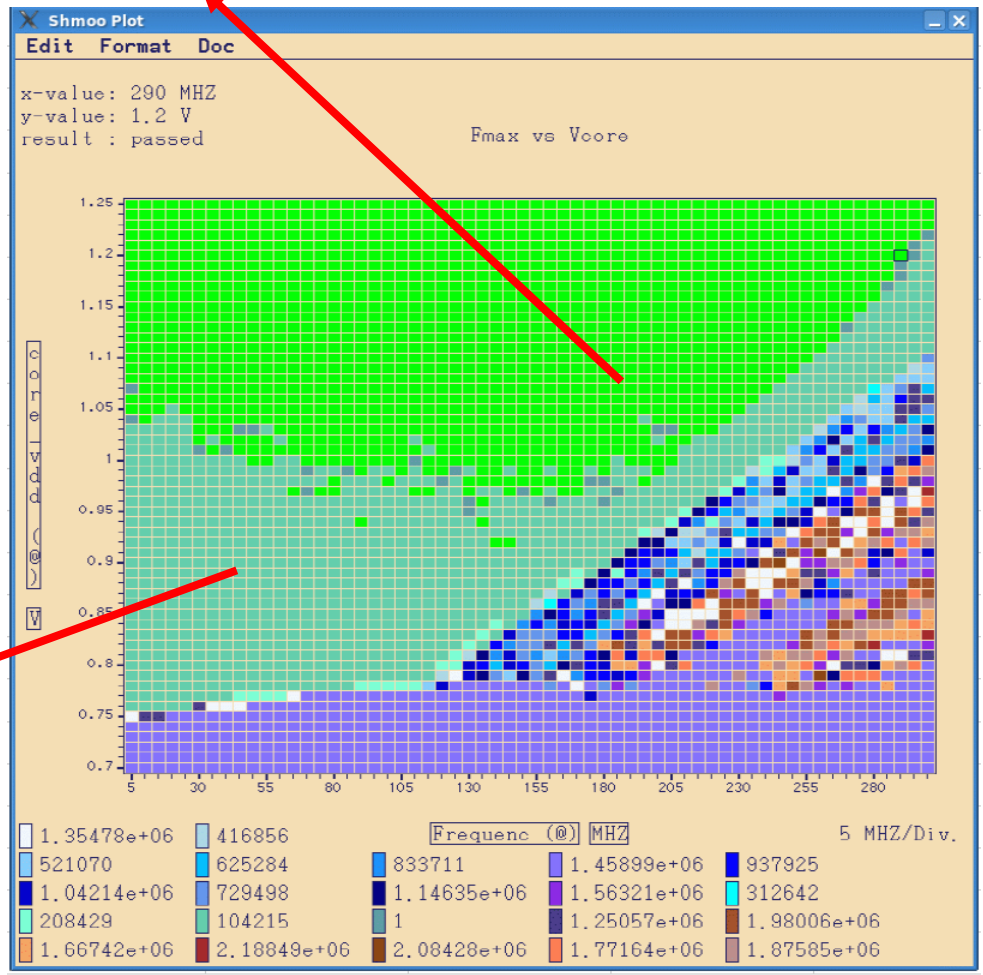
- FP needed?
  - 12-bit signals sufficient
  - Input to classification double-vs-12-bit accuracy loss $< 0.5\%$ (80.6% to 80.1%)

Sub pJ/OP?

437 GOPS/W @1.2V

803 GOPS/W @0.8V

0% bit flips

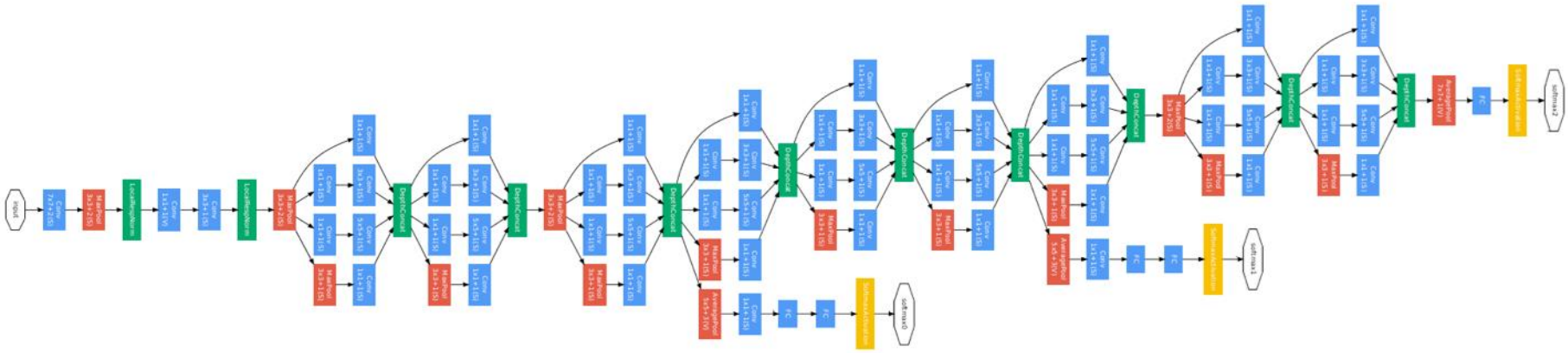


1% bit flips

1.84x energy improvement

1.2pJ/OP

CNNs: typical workload



Example: **GoogLeNet** [ILSVRC 2014 winner]

~**7x10^6** parameters

~**2.3x10^9** MAC operations on a **320x240** RGB image

Realtime (10 fps): ~**23 GMAC/s** performance

Realtime & Low-Power (10 fps @ 10mW): ~**2300 GMAC/s/W** efficiency

Origami core in 28nm FDSOI → GoogLeNet with ~10mW

Pushing Further: YodaNN[1]

- Approximation at the algorithmic side →Binary weights
- BinaryConnect [Courbariaux, NIPS15]
  - Reduce weights to a binary value -1/+1
  - Stochastic Gradient Descent with Binarization in the Forward Path

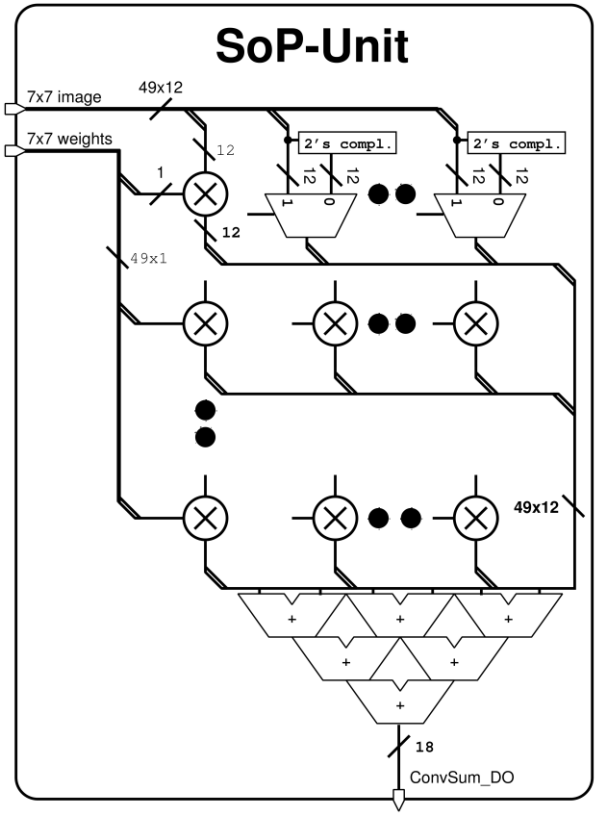
$$w_{b,stoch} = \begin{cases} -1 & p_{-1} = \sigma(w) \\ 1 & p_1 = 1 - p_{-1} \end{cases} \qquad w_{b,det} = \begin{cases} -1 & w < 0 \\ 1 & w > 0 \end{cases}$$

  - Learning large networks is still an issue with binary connect…

- Ultra-optimized HW is possible!
  - Power reduction because of arithmetic simplification
  - Major arithmetic density improvements
    - Area can be used for more energy-efficient weight storage
  - SCM memories for lower voltage → E goes with $1/V^2$

[1]After the Yedi Master from Star Wars - “Small in size but wise and powerful” cit. www.starwars.com

SoP-Unit Optimization



Equivalent for 7x7 SoP

ImageBank

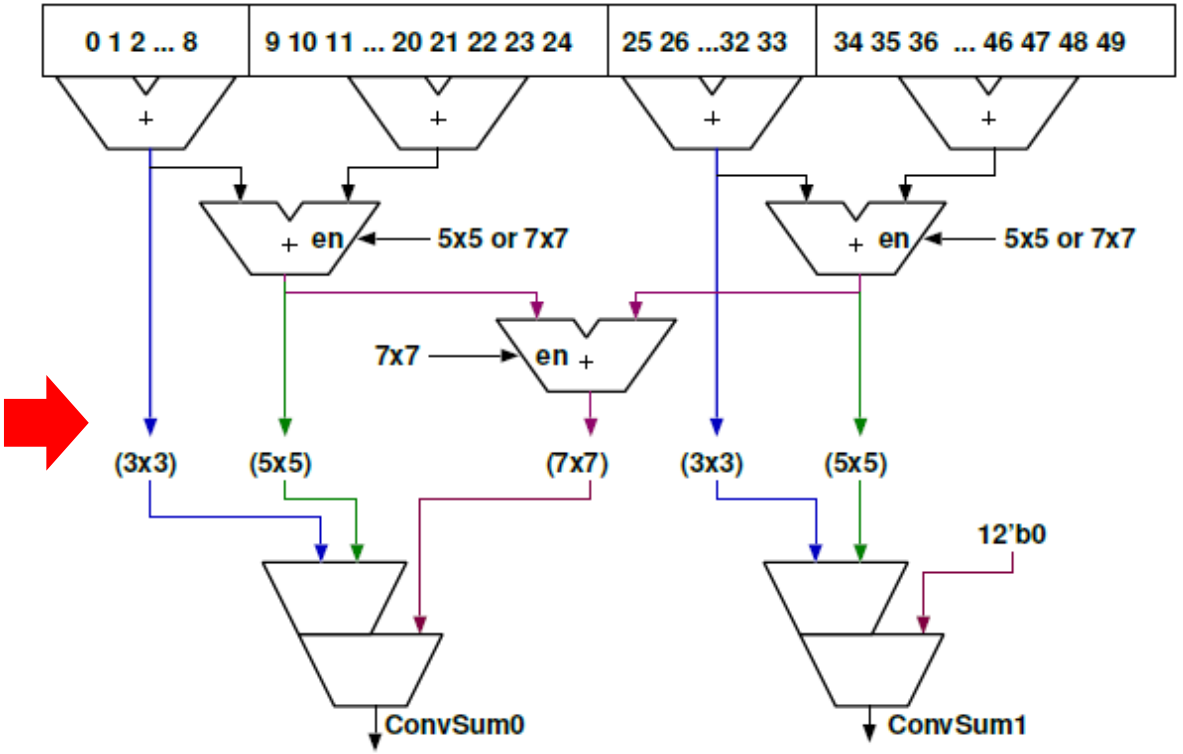
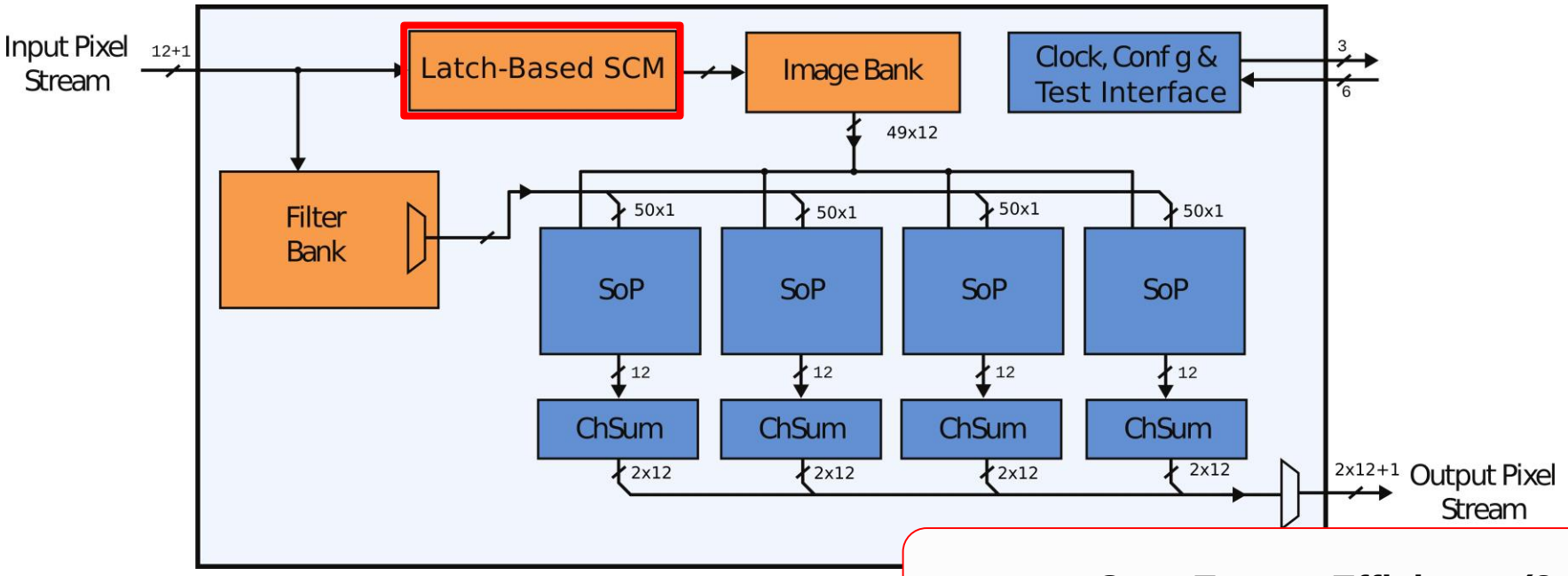


Image Mapping (3x3, 5x5, 7x7)

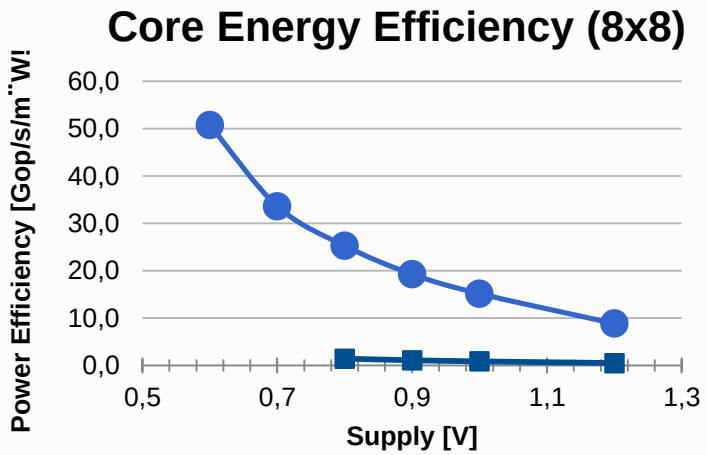
SCM for Energy efficiency

SCM (down to 0.6V in 65nm)



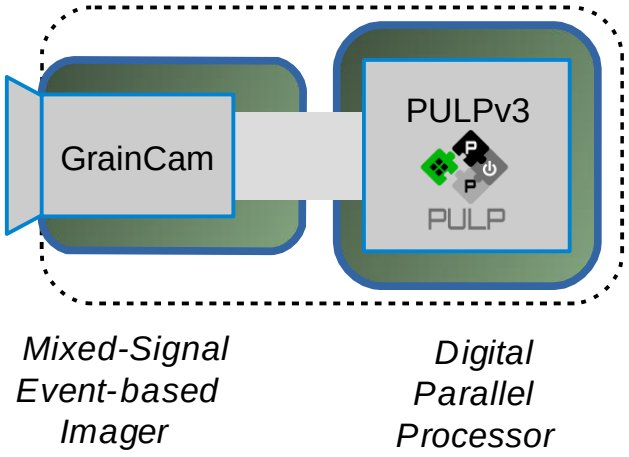
Same area 8→32 SoP units + all SCM

16x Energy efficiency improvement:
0.5pJ/OP →50GOPS/mW



Back to System-Level

Smart Visual Sensor→ idle most of the time (nothing interesting to see)

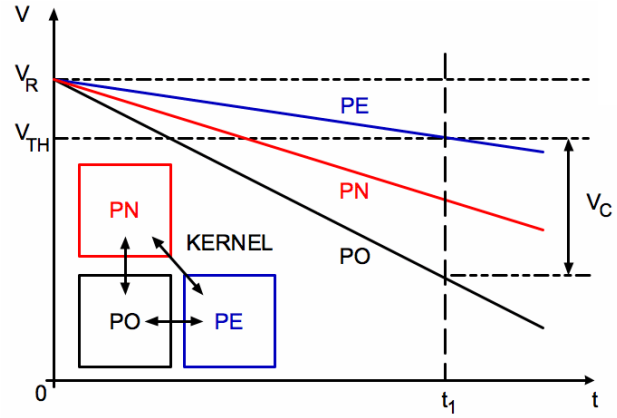


- **Event-Driven Computation**, which occurs only when relevant events are detected by the sensor
- **Event-based sensor interface** to minimize IO energy (vs. Frame-based inteface
- **Mixed-signal event triggering** with an ULP imager with internal processing AMS capability

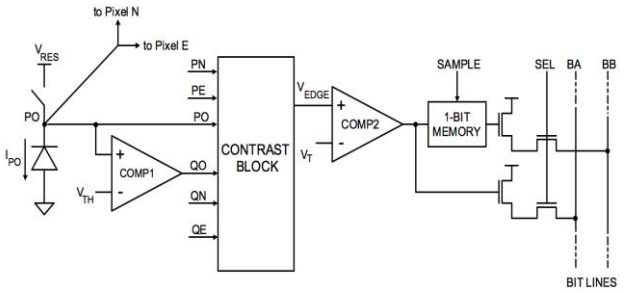
Doing *nothing* well is also very important!

GrainCam Imager (FBK)

Pixel-level spatial-contrast extraction



$$V_C = V_{PE}(t_1) - V_{PO}(t_1) = (V_R - V_{TH}) \left(\frac{I_{PO} - I_{PE}}{I_{PE}} \right)$$



Analog internal image processing

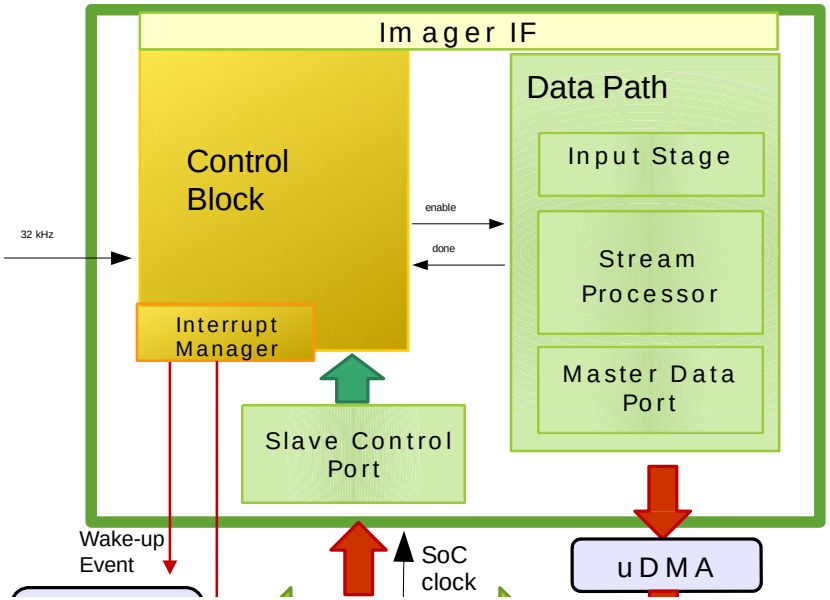
- Contrast Extraction
- Motion Extraction, differencing two successive frames
- Background Subtraction, differencing the reference image, stored in the memory, with the current frame

Energy-Efficient Sensor IF

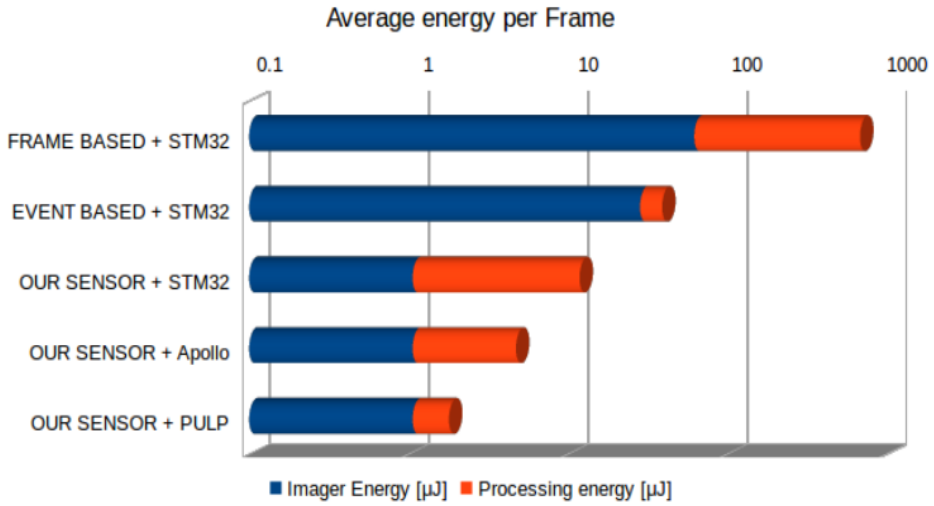
Event-based + Prefiltering + Buffering

Control Block:

- Always-on 32KHz controller for continuously handling sensor timing signals (e.g. integration, readout periods) and control signals (e.g. readout mode, reference frame sampling)
- Activates data-path block during imager readout phase
- Triggers events

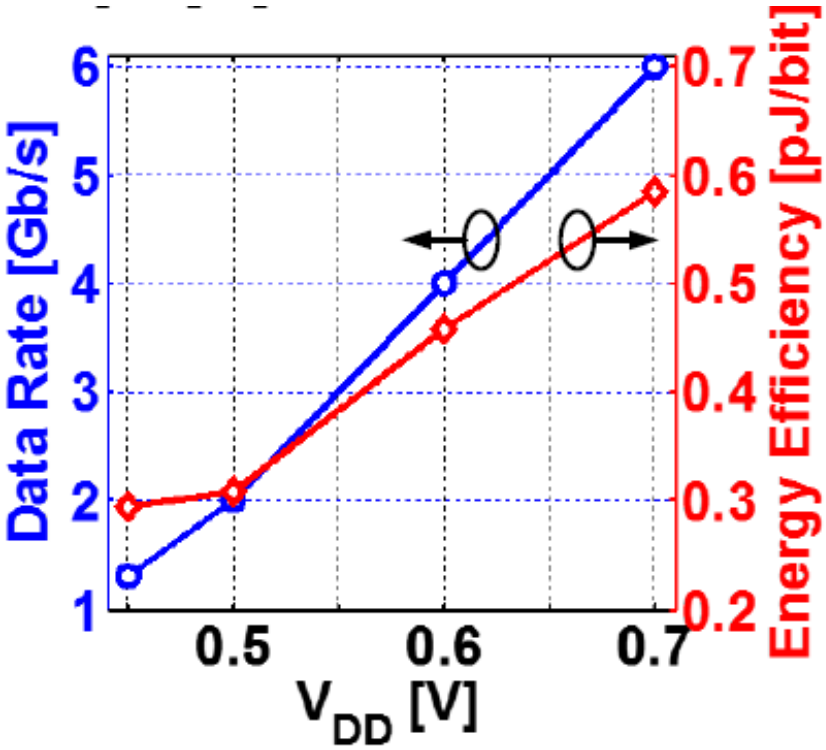


Data Path: stream processing on sensed post-triggering data
u-DMA: autonomously transfers pixel-events to L2 memory during imager readout, without requiring cluster action
Sensor IF Peripheral as transfer initiator.
Slave Port interface to set-up sensor interface parameters



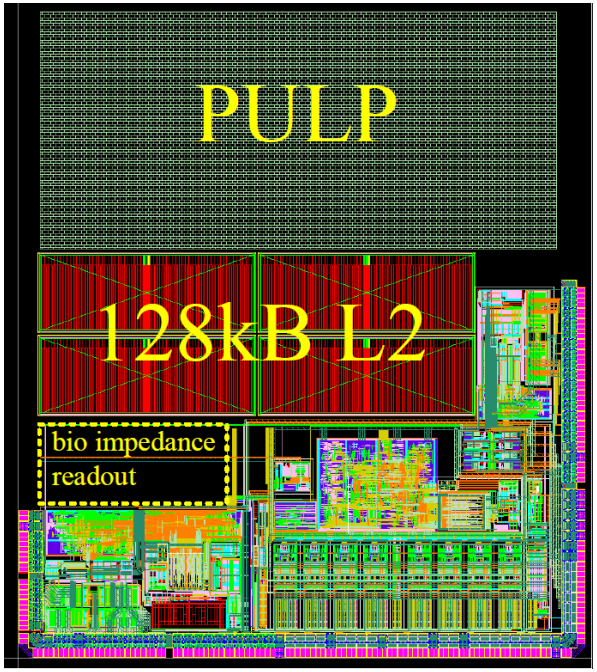
Integrating & communicating?

Sub pJ/b Chip2chip link



Source-synchronous, pseudo-differential, unterminated, Voltage Mode, 200mVpp, 1/8 rate CLK, self-calibrating PLL-based phase generator [ISSCC15]

VivoSoC2 – full AFE integration



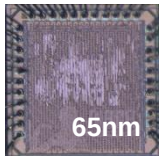
Multiple Biomedical Interfaces:
8 ExG channels with lead-off detection, Photoplethysmography (PPG), Bio-impedance monitoring, temperature sensing, 6 channels nerve blocker and stimulator, standard digital interfaces

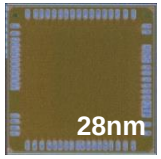
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Open Source
Parallel ULP computing for the IoT

(sub)-pJ/op computing platform **- let's make it Open!**





PULP
Parallel Ultra Low Power

OpenCores
www.opencores.org

RISC-V

GCC

LLVM
COMPILER INFRASTRUCTURE

MicroPython
Python for microcontrollers

freeRTOS

OpenMP

OpenCL

OpenVX

Processor & Hardware IPs

Compiler Infrastructure

Virtualization Layer

Programming Model

cea leti

POLITECNICO MILANO

LELAND STANFORD JUNIOR UNIVERSITY

EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

life.augmented

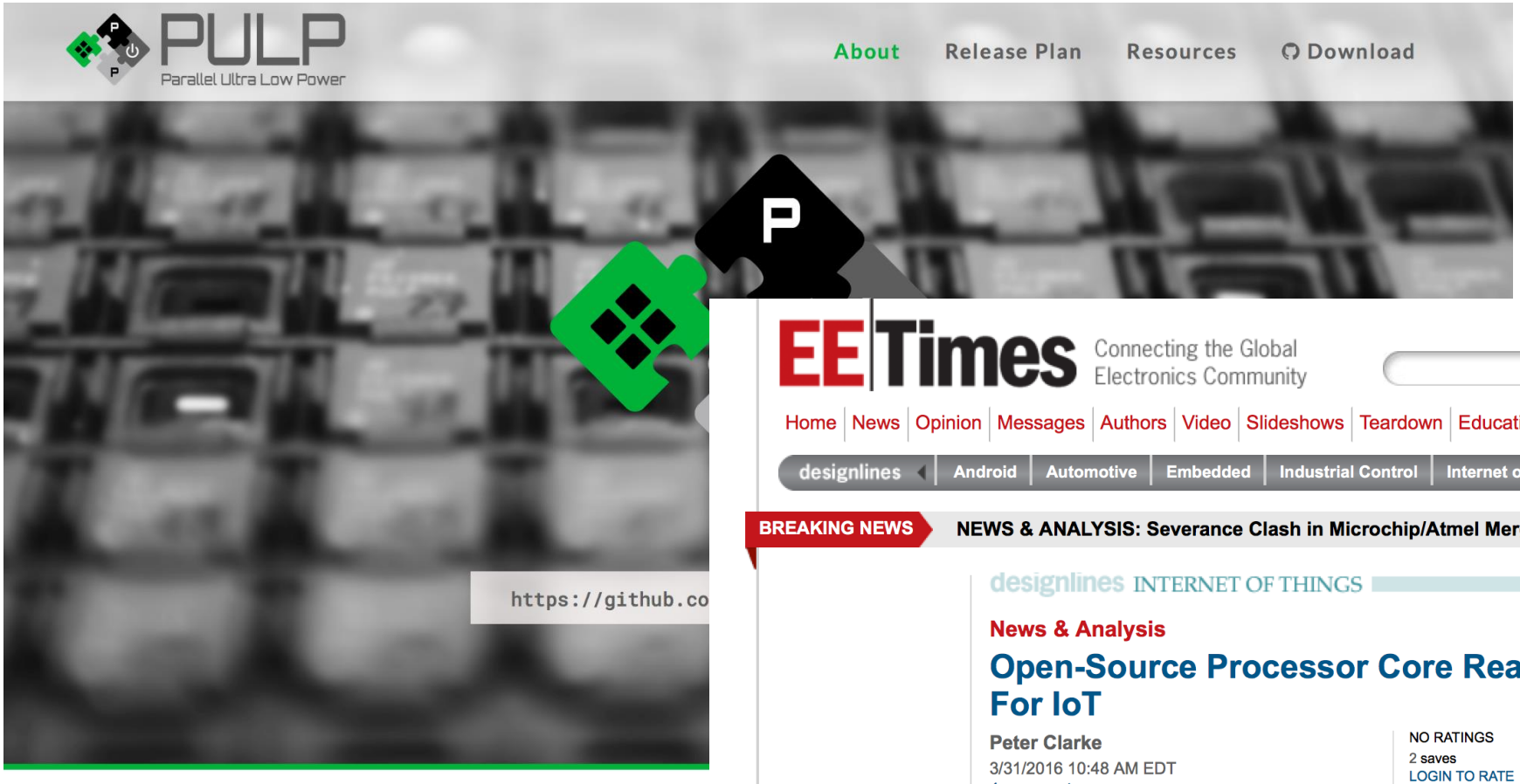
UNIVERSITY OF ILLINOIS

lowRISC

ETHZ and UNIBO released PULPino

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



EE Times — Connecting the Global Electronics Community

Home | News | Opinion | Messages | Authors | Video | Slideshows | Teardown | Education | EEL

designlines | Android | Automotive | Embedded | Industrial Control | Internet of Things

BREAKING NEWS NEWS & ANALYSIS: Severance Clash in Microchip/Atmel Merger

designlines INTERNET OF THINGS

News & Analysis

Open-Source Processor Core Ready For IoT

Peter Clarke
3/31/2016 10:48 AM EDT
4 comments

NO RATINGS
2 saves
LOGIN TO RATE

Like 105 Tweet Share 165 G+1 20

Researchers at ETH Zurich (Swiss Federal Institute of Technology in Zurich) and the University of Bologna have developed PULPino, an open-source processor optimized for low power consumption and application in wearables and the Internet of Things (IoT).

Hundreds of git forks in a few weeks!

Conclusions

- IoT a Challenge and an opportunity
  - Computing Everywhere!
- Energy efficiency requirements: pj/OP and below
  - Technology scaling alone is not doing the job for us
  - Ultra-low power architecture and circuits are needed, but not sufficient in the long run
  - Most promising technologies: 3D integration, low-leakage, non-volatile silicon-compatible mem-computing devices, but adoption rate will be SLOW
- Approximate computing to the rescue
  - Holistic approach needed, math, number systems, algorithms, tools, runtime, architecture, circuits, devices → from approximate to **transprecision** computing
- Non-Von-Neumann + Transprecision… Jackpot potential!
- Open Source HW & SW approach for building an innovation ecosystem

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Thanks for your attention!!!

PULP
Parallel Ultra Low Power

www.pulp-platform.org
www-micrel.deis.unibo.it/pulp-project
iis-projects.ee.ethz.ch/index.php/PULP