PULP: A Parallel Ultra Low Power platform for next generation IoT Applications

Davide Rossi

Francesco Conti, Andrea Marongiu, Antonio Pullini, Igor Loi, Michael Gautschi, Giuseppe Tagliavini, Alessandro Capotondi, Philippe Flatresse, Luca Benini

1DEI-UNIBO, 2IIS-ETHZ, 3STMicroelectronics
How efficient do we need to be?

- $10^{12} \text{ops/J}$
- $1 \text{pJ/\text{op}}$
- $1 \text{GOPS/mW}$

[*RuchIBM11*]
System View

Sense

MEMS IMU
MEMS Microphone
ULP Imager
EMG/ECG/EIT

Analyze and Classify

µController
L2 Memory
IOs

Battery + Harvesting powered → a few mW power envelope

100 µW ÷ 2 mW

Transmit

Short range, BW
Low rate (periodic) data
SW update, commands
Long range, low BW

Idle: ~1µW
Active: ~ 50mW
### Near-Sensor Processing

#### INPUT BANDWIDTH  COMPUTATIONAL DEMAND  OUTPUT BANDWIDTH  COMPRESSION FACTOR

<table>
<thead>
<tr>
<th></th>
<th>Bandwidth</th>
<th>Demand</th>
<th>Bandwidth</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tracking:</td>
<td>80 Kbps</td>
<td>1.34 GOPS</td>
<td>0.16 Kbps</td>
<td>500x</td>
</tr>
<tr>
<td><em>Lagroce2014</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Voice/Sound</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speech:</td>
<td>256 Kbps</td>
<td>100 MOPS</td>
<td>0.02 Kbps</td>
<td>12800x</td>
</tr>
<tr>
<td><em>VoiceControl</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Inertial</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kalman:</td>
<td>2.4 Kbps</td>
<td>7.7 MOPS</td>
<td>0.02 Kbps</td>
<td>120x</td>
</tr>
<tr>
<td><em>Nilsson2014</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Biometrics</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM:</td>
<td>16 Kbps</td>
<td>150 MOPS</td>
<td>0.08 Kbps</td>
<td>200x</td>
</tr>
<tr>
<td><em>Benatti2014</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Extremely compact output (single index, alarm, signature)**

**Computational power of ULP µControllers is not enough**

**Parallel workloads**
PULP: pJ/op Parallel ULP computing

- **pJ/op** is traditionally the target of ASIC + µControllers
- Scalable: to many-core + heterogeneity
- Best-in-class LP silicon technology
- Programmable: OpenMP, OpenCL, OpenVX
- Open: Software & HW

*From ULP computing to parallel + heterogeneous ULP computing*

*1mW-10mW active power*
Near-Threshold Multiprocessing
Minimum Energy Operation

Near-Threshold Computing (NTC):

1. Don't waste energy pushing devices in strong inversion
2. Recover performance with parallel execution
3. Aggressively manage idle power (switching, leakage)

32nm CMOS, 25°C

4.7X

[VivekDeDATE2013]
The “best” Processor

- Single issue in-order is most energy efficient
- Put more than one + shared memory to fill cluster area

[AzizilSCA10]
Building PULP

**SIMD + MIMD + sequential**

Private per-core instruction cache

- 4-stage, in-order OpenRISC core
- DEMUX

1 Cycle Shared Multi-Banked L1 Data Memory + Low Latency Interconnect

"GPU like" shared memory \(\rightarrow\) low overhead data sharing

Near Threshold but parallel \(\rightarrow\) Maximum Energy efficiency when Active

+ strong power management for (partial) idleness

Double buffering

Tightly Coupled DMA
OR1ON: Extended OpenRISC Core

- 4-stage OpenRISC
- IPC ~ 1
- DSP extensions:
  - Hardware loops
    - Eliminates branching overhead
  - LD/ST + post-increment
    - Enhanced vector indexing
  - Small vector support (SIMD)
    - 2x 16-bit operations
    - 4x 8-bit operations
  - Unaligned memory accesses
    - To better exploit SIMD

UP TO 5x performance improvement and 3x reduction of energy!!!
Silicon Implementation
Technology For ULP

UTBB FD-SOI provides good features for ULP design:

- Good behavior at low voltage
- Body bias for power and variability management
Body biasing with UTBB FD-SOI technology

**RVT transistor** (conventional-well)

**LVT transistor** (flip-well)

**BODY BIAS WINDOWS**

- **RVT:** Regular Voltage Threshold
- **LVT:** Low Voltage Threshold
- **FBB:** Forward Body Bias
- **RBB:** Reverse Body Bias

Poly biasing allow to trade performance/leakage At design time

**RVT transistors: low leakage + flexible power management (FBB + RBB)**
Near Threshold + Body Biasing Combined

State retentive (no state retentive registers and memories)
Ultra-fast transitions (tens of ns depending on n-well area to bias)
Low area overhead for isolation (3µm spacing for deep n-well isolation)
Thin grids for voltage distribution (small transient current for wells polarization)
Simple circuits for on-chip VBB generation (e.g. charge pump)

But even with aggressive RBB leakage is not zero!

FBB vs. FREQUENCY

RBB vs. LEAKAGE

RVT transistors
Body Biasing for Variability Management

Process variation

120° C

Thermal inversion

100x @0.5V

-40° C

25 MHz ± 7 MHz (3σ)

RVT transistors
FBB/RBB

FREQUENCY + LEAKAGE COMPENSATION WITH ± 0.2 BB

FREQUENCY + LEAKAGE COMPENSATION WITH -1.8 to +0.5 BB
**ULP memory implementation:**

**latch-based SCM**

- **“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
  - Controlled P&R mitigates area overhead

![256x32 6T SRAMS vs. SCM](image)
Architectural Technology Awareness
Exploiting body biasing

- The cluster is partitioned in separate clock gating and body bias regions
- Body bias multiplexers (BBMUXes) control the well voltages of each region
- Each region can be **active** (FBB) or **idle** (deep RBB → low leakage!)

**State-Retentive + Low Leakage + Fast transitions**
Core shut-down sequence:
1) Disable fetching
2) Wait outstanding transactions
2) Clock gating
3) Reverse Body Biasing

Private, per core port
→ single cycle latency
→ no contention

GOALS:
→ Reduce parallelization overhead
→ Accelerate common OpenMP and OpenCL patterns (e.g. Task creation)
→ Automatically manage shut down of idle cores
Power Management: External Events

Programming sequence:
1) Set events mask
2) Program transfer
3) Trigger transfer
4) Shut down cores

48 maskable events
→ General purpose
→ DMA
→ Timers
→ Peripherals (SPI, I2C, GPIO...)

GOALS:
→ Automatically manage shut down of cores during data transfers
Heterogeneous Memory Architecture + Management

Shared I$ to recover SCMs area overhead
Private L0 buffers to reduce pressure on shared I$

SCM on I$ and part of TCDM to widen the operating range

Reconfigurable Pipeline Stages for SRAMs degradation @ low VDD

MMU (logical/physical add map):
1) Interleaved/private addresses
2) Shutdown of SRAM banks
The PULP “Family”
## CHIP FEATURES

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>28nm FDSOI (RVT)</td>
</tr>
<tr>
<td>Chip Area</td>
<td>3mm²</td>
</tr>
<tr>
<td># Cores</td>
<td>4xOpenRISC</td>
</tr>
<tr>
<td>I$</td>
<td>4x1kbyte (private)</td>
</tr>
<tr>
<td>TCDM</td>
<td>16 kbyte</td>
</tr>
<tr>
<td>L2</td>
<td>16 kbyte</td>
</tr>
<tr>
<td>BB regions</td>
<td>6</td>
</tr>
<tr>
<td>VDD range</td>
<td>0.45-1.2V</td>
</tr>
<tr>
<td>VBB range</td>
<td>-1.8V - +0.9V</td>
</tr>
<tr>
<td>Perf. Range</td>
<td>1 MOPS-1.9GOPS</td>
</tr>
<tr>
<td>Power Range</td>
<td>100 µW -127 mW*</td>
</tr>
<tr>
<td>Peak Efficiency</td>
<td>60 GOPS/W@0.5V*</td>
</tr>
</tbody>
</table>

*Does not include IOs*
Measured Results

Maximum Energy Efficiency @ 0.5V + 0.5V FBB → 60GOPS/W

~10mW @ 100 MHz, 0.75V

1.8GOPS

Low leakage (< 2%)

Wide operating range

Peak GOPS/W competitive with best-in-class near-threshold (16bit) ULP microcontrollers, plus more than x100 peak GOPS!
= PULPv1 + 2 DVFS regions (SoC + CLUSTER) + Event Unit + Peripherals
= PULPv2 + Extended cores + HW Synch + Shared Cache + HWCE + Shared IOs
# PULP’s Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>PULPv1</th>
<th>PULPv2</th>
<th>PULPv3</th>
</tr>
</thead>
<tbody>
<tr>
<td># of cores</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>L2 memory</td>
<td>16 kB</td>
<td>64 kB</td>
<td>128 kB</td>
</tr>
<tr>
<td>TCDM</td>
<td>16kB SRAM</td>
<td>32kB SRAM</td>
<td>32kB SRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8kB SCM</td>
<td>16kB SCM</td>
</tr>
<tr>
<td>Reconf. pipe. stages</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>I$</td>
<td>4kB SRAM private</td>
<td>4kB SCM private</td>
<td><strong>4kB SCM shared</strong></td>
</tr>
<tr>
<td>Body bias regions</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>DVFS</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>I/O connectivity</td>
<td>JTAG</td>
<td>full</td>
<td>full multiplexed</td>
</tr>
<tr>
<td>Extended processor</td>
<td>no</td>
<td>no</td>
<td><strong>Yes</strong></td>
</tr>
<tr>
<td>Event unit</td>
<td>no</td>
<td>yes</td>
<td>yes+ HW synchro</td>
</tr>
<tr>
<td>Debug unit</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td><strong>silicon proven</strong></td>
<td><strong>post tape out</strong></td>
<td><strong>pre tape out</strong></td>
</tr>
<tr>
<td><strong>Technology</strong></td>
<td>FD-SOI 28nm conventional-well</td>
<td>FD-SOI 28nm flip-well</td>
<td>FD-SOI 28nm conventional-well</td>
</tr>
<tr>
<td>Voltage range</td>
<td>0.45V - 1.2V</td>
<td>0.3V - 1.2V</td>
<td>0.5V - 0.7V</td>
</tr>
<tr>
<td>BB range</td>
<td>-1.8V - 0.9V</td>
<td>0.0V - 1.8V</td>
<td>-1.8V - 0.9V</td>
</tr>
<tr>
<td><strong>Max freq.</strong></td>
<td>475 MHz</td>
<td>1 GHz</td>
<td>200 MHz</td>
</tr>
<tr>
<td><strong>Max perf.</strong></td>
<td>1.9 GOPS</td>
<td>4 GOPS</td>
<td>1.8 GOPS</td>
</tr>
<tr>
<td><strong>Peak en. eff.</strong></td>
<td><strong>60 GOPS/W</strong></td>
<td><strong>135 GOPS/W</strong></td>
<td><strong>385 GOPS/W</strong></td>
</tr>
</tbody>
</table>

*equivalent 32-bit RISC operations*
Breaking the GOPS/mW wall
Recovering more silicon efficiency

GOPS/W

1

3

6

> 100

General-purpose Computing

Throughput Computing

ULP parallel Computing

CPU

GPGPU

Accelerator Gap

SW

Mixed

HW

1GOPS/mW

Closing The Accelerator Efficiency Gap with Agile Customization
Fractal Heterogeneity

Fixed function accelerators have limited reuse... how to limit proliferation?
Brain-inspired (deep convolutional networks) systems are high performers in many tasks over many domains.

- Human: 85% (untrained), 94.9% (trained)
- CNN: 93.4% accuracy

Image recognition
[RussakovskyIMAGENET2014]

Speech recognition
[HannunARXIV2014]

Flexible acceleration: learned CNN weights are “the program”
PULP CNN Performance

Average performance and energy efficiency on a 32x16 CNN frame

**PERFORMANCE**

- **8 GOPS**
- **61x**

**ENERGY EFFICIENCY**

- **6500 GOPS/W**
- **47x**

**PULPv3 ARCHITECTURE, CORNER:** tt28, 25° C, VDD = 0.5V, FBB = 0.5V
Thanks for your attention!!!
References


How Big is the IoT?

Data of the Internet of Things

- **BrontoByte**: The digital universe of tomorrow
- **ZettaByte**: In 2016, 1.3 ZB will cross our digital networks daily
- **PetaByte**: The CERN LHC generates 1 PB per second
- **YottaByte**: The digital universe today: 250 trillion DVDs
- **ExaByte**: At the moment, every day 1 EB of data is created on the internet. That is the equivalent of 250 million DVD’s. The Square Kilometer Array Telescope will produce around 1 EB per day.

How much energy to process (1 op. per Byte) one BB?
Microcontrollers Landscape

*not exhaustive
Parallel NTC

High Workloads

Low Workloads

```
<table>
<thead>
<tr>
<th>Target Workload [MOPS]</th>
<th>1-Core Energy Efficiency (ideal) [MOPS/mW]</th>
<th>4-Cores Energy Efficiency (ideal) [MOPS/mW]</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>43</td>
<td>55</td>
<td>1.3x</td>
</tr>
<tr>
<td>200</td>
<td>33</td>
<td>50</td>
<td>1.5x</td>
</tr>
<tr>
<td>400</td>
<td>18</td>
<td>43</td>
<td>2.4x</td>
</tr>
</tbody>
</table>
```

*Measured on our first prototype*
Parallel NTC + Race to Halt

**SINGLE-CORE @ MAX FREQUENCY** (e.g. 200MHz)

- **Core power**
- **System power**

*Low Workload (duty cycled)*

**ACTIVE PERIOD**

**MULTI-CORE @ MAX FREQUENCY** (e.g. 200 MHz)

- **Core power**
- **System power**

*Ideally same energy of single-core solution*

*Deep sleep*

*Saved energy*

*Active period*

Going faster allows to integrate system power over a smaller period

The main constraint here is the power envelope
Back to SRAMs

SRAM performance rapidly degrades at low voltage

SRAM VDDMIN is higher than logic (and SCM)