Near- and Sub-Threshold Design for Ultra-Low-Power Embedded Systems

Andreas Burg,
P. Meinerzhagen, A. Dogan, J. Constantin, M. M. Sabry Ali,
G. Karakonstantis, D. Atienza, L. Benini
EPFL, Switzerland

Telecommunications Circuits Laboratory
Energy autonomous systems with low- to moderate computing requirements

- (Wireless) sensor nodes
- Biomedical applications (wearable and implanted)
Ultra Low Power Design with Voltage-Frequency Scaling

System Level Strategies for Ultra-Low-Power

- Single core architectures
- ULP Multi core architectures

Low-Power Memories for Scaled Voltages

A New Way of Dealing with Reliability at Scaled Voltages
Sources of Power Consumption in CMOS Circuits

Dynamic power consumptions
- Switching & short-circuit power
- Depends on circuit activity and clock frequency

\[ P_{dyn} \sim \alpha C_{tot} V_{DD}^2 / T_{clk} \]

Leakage power consumptions
- Independent of the activity

\[ P_{leak} \sim k I_0 V_{DD} \]

Supply voltage scaling
- reduces active and leakage power consumption
- but also reduces speed
Energy per operation as a metric of efficiency

\[ E \sim \alpha C_{tot} V_{DD}^2 + k I_0 V_{DD} T_{clk} \]

- Operation at critical path speed minimizes leakage

Above VT \((V_{DD} > V_T)\)

- \(T_{clk}\) increases linearly as \(V_{DD}\) decreases
- Active energy dominates
- Voltage scaling improves energy efficiency significantly
Tradeoff between energy and speed

Below VT ($V_{DD} > V_T$)
- $T_{clk}$ increases exponentially as $V_{DD}$ decreases
- Very high cost for energy reduction
- Poor energy-delay product

Near-VT operation: balance between energy and delay
Energy per operation as a metric of efficiency

\[ E \sim \alpha C_{tot} V_{DD}^2 + k I_0 V_{DD} T_{clk} \]

Near/below VT \((V_{DD} < V_T)\)
- Exponential delay increase outweighs the reduced \(V_{DD}\)
- Leakage starts to dominate

Energy minimum voltage (EMV): balance between leakage and active energy

J. Rodrigues, PATMOS 2011, Keynote
Real-time embedded system requirements

- Handle a given workload with lowest power consumption

Optimum solution

- Operation at the energy minimum voltage with power gating during idle periods to avoid leakage

- But, power gating is only effective when idle periods are long and memories can often not be power gated and are the major source of leakage
Maximum Voltage/Frequency Scaling is Optimal

For a given architecture, and a given workload

- **Lower voltage** results in **lower power** consumption
- **But processing requirements** determine the **minimum operating voltage**

Need to minimize processing requirements to reduce active power and also leakage

[Diagram showing voltage versus power consumption]
System Level Strategies for Ultra-Low-Power
Shimmer: developed by the Digital Health Group at Intel

- Careful algorithm design and optimization enables decent lifetimes, e.g., for ECG monitoring [Rincon et al., ITAB, 2011]

- **TI MSP430 CPU**
  - 16-bit operation
  - 10 kB RAM, 48 kB Flash
  - Up to 8 MHz
  - 8-channel, 12-bit ADC

- Problems with state-of-the-art WBSN processors
  - Few widely used discrete components (e.g., TI MSP430 series)
  - Often outdated, complex architectures with significant legacy burden
  - Low operating frequencies even at nominal voltage
  - No or very limited voltage-frequency scaling for energy reduction
Memories Consume a Considerable Amount of Power

For embedded processors, memories occupy a large percentage of the silicon area

- No system power gating for many systems (retain contents)
  - Significant contribution to power consumption through leakage, especially at low voltages/frequencies

Achieve optimum energy efficiency:
Reduce complexity AND memory requirements
Application-Specific Processors to the Rescue

ASIP for Low-Power Embedded Systems

Source: Prof. T. Noll, RWTH Aachen
A plethora of different factors influence energy efficiency

- Software (coding style) and compilation tools
- System architecture
- ISA + processor core architecture
- RTL and gate level (synthesis & constraints)
- Floorplan and physical design
Increasing speed improves potential for voltage scaling but usually comes also at the expense of power

- Better processors reduce execution cycles but are more complex and require longer cycle times
- Higher maximum clock frequencies require more complex circuits and more area (more leakage)

Need to consider leakage as well at scaled voltages

- Memories are a major source of leakage and often dominate power consumption. Nevertheless, processing requirements are often reduced at the cost of memory

Need to reduce execution time and memory requirements without adding too much overhead.
ASIP core architecture described in LISA (PD)

• Iterate and optimize core architecture for lowest power consumption

Automatic generation:

• Software tool-chain
• Cycle-accurate ISS
• Synthesizable HDL

Tool-based processor architecture exploration and tools chain generation
How to evaluate total system power accurately?

- **Accurate power analysis using vcd-based post-layout gate-level simulations**

For small ULP systems, iterations through the complete design flow are possible.
Very simple, true RISC ISA

- 16-bit Harvard architecture, 3-stage pipeline, 24-bit instruction words
- 11 Single word, single cycle instructions
- Minimalistic ALU
  - ADD, SUB, AND, OR, XOR, Shift, Mult.
- Simplified data memory interface
- Addressing modes for efficient execution of DSP applications
- Immediate branching + full data bypass

- Less than 5% of an embedded platform (< 10 kGE)
- Operates up to 180 MHz for voltage scaling
- Near-VT computing: Only 6pJ per Op at 0.5V, running at ~1 MHz
Real-Time Operating System Support

Processor model includes hardware support for:
- **Interrupt** handling
- Context / **task switching**

**FreeRTOS** available on TamaRISC
- C-compiler support for 32-bit operations
- FreeRTOS compiled from **original C-source**

Custom LISA simulation model (ISS) with support for
- **Sleep mode** through clock-gating of the processing core: essential for ULP operation
- **Peripherals**
  - Timer
  - ADC
  - Interrupt controller
## Performance Results: PIC24 vs TamaRISC

<table>
<thead>
<tr>
<th></th>
<th>MSP430 COTS</th>
<th>PIC24 LISA</th>
<th>TamaRISC LISA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Clock [MHz]</td>
<td>8</td>
<td>122</td>
<td>180</td>
</tr>
<tr>
<td>Core Area [kGE]</td>
<td>N.A.</td>
<td>33.5</td>
<td>14.1</td>
</tr>
<tr>
<td>Memory Area [kGE]</td>
<td>N.A.</td>
<td>58.2</td>
<td>58.2</td>
</tr>
<tr>
<td>Total Area [kGE]</td>
<td>N.A.</td>
<td>91.7</td>
<td>72.3</td>
</tr>
<tr>
<td>Application CS [kCycles]</td>
<td>800.0</td>
<td>99.9</td>
<td>99.8</td>
</tr>
<tr>
<td>Application DWT [kCycles]</td>
<td>4700</td>
<td>1714.9</td>
<td>1834.3</td>
</tr>
<tr>
<td>Execution Time CS [ms]</td>
<td>100</td>
<td>0.819</td>
<td>0.554</td>
</tr>
<tr>
<td>Execution Time DWT [ms]</td>
<td>588</td>
<td>14.06</td>
<td>10.19</td>
</tr>
</tbody>
</table>

*in 180 nm CMOS technology*

ISA + core architecture simplicity is key, TamaRISC outperforms PIC24 and MSP430 in all criteria!
Energy-limited wireless sensor nodes

- Efficient data compression scheme (for communication and/or storage)
- ULP computing / circuit design (sub-threshold operation)

Low complexity sensor data compression: Compressed Sensing

- Data independent
- Low complexity
Samples of a sparse signal can be compressed efficiently while sensed

Matrix-vector multiplication:

\[ y = \Phi x \]

Compress vector of input samples \( x \), to sampled data vector \( y \)

Random sensing matrix (e.g. with uniformly distributed entries)

Sensing matrix can be very sparse! [M11]

- Few non-zero entries per column
- Entries can be one

**Algorithm 1 Pseudocode of Compressed Sensing Algorithm**

1: for \( i := 1 \rightarrow n \) do
2: \( sample := \text{getSample}() \)
3: for \( j := 1 \rightarrow \#\text{ones} \) do
4: \( index := \text{getRandomIndex}(1..k) \)
5: \( \text{buffer}[index] := \text{buffer}[index] + sample \)
6: end for
7: end for

- Efficient random number generation (indices)
- Hardware support: instruction set extension!

Random Number Generation (RNG)

Pseudo-random index sequence, as memory addressing offsets

Precomputation: storage in data memory
- 6 Kbyte per sensing matrix (e.g. for 512 samples with 50% compression)
- Large data memory footprint: large area, **high leakage power**!

Computation at runtime: LFSR based
- Software calculation: computational effort of **1k cycles / sample**

ULP operation in the sub-threshold regime significantly reduces core clock frequency: 1–20 kHz

Insufficient sampling rates: 1–20 Hz

ASIP with dedicated ISE is crucial!
CS Accumulation (CSA) instruction
- Compression buffer address
- Sample data
**Address random element in buffer**
- New random memory offset **each cycle**
**ISE hardware overhead:**
< 3% of total area
**Multiple LFSR steps per cycle**
- Reduce index correlation!

**ECG data reconstruction quality**
- 50% data compression (at 125 Hz sampling rate)
- **TamaRISC-CS with ISE**
  - 16.5 cycles/sample
  - 2.1 kHz clock freq.
  - 30.6 nW power
- Without ISE (std. RISC)
  - 1025.5 cycles/sample
  - 128 kHz clock freq.
  - 355 nW power
- **Reference impl. [Ma11]:**
  - storage approach (large memory): 390.5 cycles/sample

**Power consumption improved by 11.6x through CS ISE!**
TamaRISC-SHA3 for Authentication & Hashing

- SHA-3: set of cryptographic hash functions
- Equally important design goals: hashing speed, memory footprint
- Custom, algorithm-specific instruction set extensions

Average speedup: 172% Average memory savings: 40%

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE</td>
<td>155.2</td>
<td>102.9</td>
<td>-34%</td>
<td>~ 0%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-88%</td>
<td>+10%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-17%</td>
<td>+10%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-30%</td>
<td>~ 0%</td>
</tr>
<tr>
<td>Skein</td>
<td>157.6</td>
<td>112.6</td>
<td>-29%</td>
<td>~ 0%</td>
</tr>
</tbody>
</table>

Speedup → active power improvement: 1.2x - 8x

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Data PIC24 [byte]</th>
<th>Data +ISE</th>
<th>Text PIC24 [byte]</th>
<th>Text +ISE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE</td>
<td>488</td>
<td>-59%</td>
<td>1,028</td>
<td>-20%</td>
</tr>
<tr>
<td></td>
<td>-78%</td>
<td>2,619</td>
<td>-69%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-87%</td>
<td>4,649</td>
<td>-53%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-21%</td>
<td>3,480</td>
<td>-31%</td>
<td></td>
</tr>
<tr>
<td>Skein</td>
<td>242</td>
<td>0%</td>
<td>5,734</td>
<td>-18%</td>
</tr>
</tbody>
</table>

Leakage power savings: 40%

Constantin et al., ASAP, 2012
Multi-core systems are on the rise to solve thermal issues in desktop and HPC

- Very high clock frequency
- Performance limited by technology and TDP
- Active power still dominates

Ultra-low-power design

- Leakage is a significant contributor
- Performance is usually not technology limited
- Voltage scaling limitations due to reliability issues

Are multi-core architectures power efficient for embedded ultra-low-power designs?
Medical grade ECG recording requires multiple (4-8) leads

Multi-channel signal analysis is often embarrassingly parallel
- Filtering
- Baseline removal
- Data compression

Applications are well suited for multi-core platforms: process individual leads on multiple cores in parallel

Low-power design concept: parallel processing enables more aggressive voltage-frequency scaling
A simple Multi-core Architecture For Embedded Applications

- Several ULP cores sharing multi-bank DM
- Multiple instruction, multiple data execution (MIMD)
- Logarithmic interconnect [R11]
- Simplified Overall Design
  - Simplified Clock network
  - Single Supply Voltage
- No need for multi-port DM
  - Low leakage consumption

However,
- Occasional stalls for cores
  - Clock gating for reduced active power


Dogan et al., IETCDS, 2012
Multi-core meets workload requirements at lower supply voltages
Multicore is the only viable solution for workloads higher than 50.1 MOps/s.
Between 1.3–50.1 MOps/s workloads multi-core is more power efficient, up to 66%
The power consumptions become equal at 1.3 MOps/s. Multi-core has reached voltage-scaling-limit at 5.58 MOps/s.
Single-core is more power efficient for workloads lower than 1.3 MOps/s.
Multi-Core Power Bottleneck Analysis

- High workloads: > 50% of power due to instruction fetch
- Low workloads: > 90% of the power due to leakage in memories

Instruction fetch and instruction memory responsible for most of the power consumption

Reduce number of memory accesses by exploiting application characteristics
Reduce amount of (active) memory

Dogan et al., DATE, 2012
Multi-core architecture with voltage scaling for ultra-low-power embedded DSP, utilizing advanced memory organization [D12b]

Fully shared data and instruction memories

Logarithmic interconnects [R11] between cores and memories

Simple low-power processing cores (extended TamaRISC)


Application characteristics (multi-lead ECG)

- Significant parts of the code execute on multiple data
- Little or no data-dependent control flow

Synchronize cores before running identical code segments in lockstep
Code segments for virtual SIMD execution

- Mapped to a single memory bank
- **Instructions broadcasting** reduces memory access power
- **Power gating of unused memory banks** reduces leakage

Dogan et al., DATE, 2012
Data Memory Organization

Address translation handled by a simple MMU
- Address space partitioning can be changed at runtime

Focus on minimizing data access conflicts

Private segments: mapped into separate memory banks
- No conflicts for accesses to private data

Shared data: interleaved across banks to avoid conflicts
- Data broadcasting avoids conflicts in SIMD mode

Dogan et al., DATE, 2012
Reference: processing cores with dedicated instruction memories

- 86% power savings on instruction fetch
- Xbars consume less than 8% of the total power
- Broadcasting entails no noticeable power overhead

Dogan et al., DATE, 2012
For an embedded biosignal compression application:

- Exploited SIMD operations
  - Only 4% execution time increase
  - Up to 45.7% power saving at high workloads
  - Up to 38.8% power saving at low workloads
- Unused memory banks are power gated

Thanks to Virtual SIMD execution, a considerable amount of active power is saved at high workloads. Furthermore, leakage power is reduced dramatically for low workloads by partial power gating of memories.

Dogan et al., DATE, 2012
Virtuel SIMD execution highly relies on synchronous code execution

- **Lockstep execution** not guaranteed even for embarrassingly parallel applications
  - Example: For a multiple-input biosignal filtering application, SIMD execution is exploited less than 5% of instructions

- What may bring synchronous cores out of lockstep code execution?
  - Shared DM accesses may lead stalls of some cores
  - Conditional code sections in executed algorithms
Smart interconnects
- Enhanced serving policy avoids synchronization loss due to DM access conflicts

Lightweight software-directed hardware synchronizer
- Resynchronization after each sections of conditional code
- Dedicated Instructions keep overhead low
Synchronization: Power and Performance Results

- Power reduction
  - Up to 2.3x increase for SIMD operations
  - 64% power savings when voltage scaling is considered
- Performance gains
  - Up to 2.4x speed-up due to better access coordination
Low-Power Memories for Scaled Voltages
Increasing need for embedded memories in SoCs & ASICs [ITRS’11]
Memories are responsible for most of the active and leakage power in ULP embedded systems
Memories are the first point of failure under scaled voltages and limit the yield

SRAMs for scaled voltages (especially near- and sub-VT) are tedious handcrafted custom designs

Memory size in SoCs
50x in 25 years
[ITRS’11]
At low voltages, 6T SRAM suffers from:

- Read failures (bit-flips)
- Write contention (inability to flip bitcell)
- Hold failures (vanishing static noise margin)
- Access time failures

Many new SRAM bitcells for robust sub-VT operation

- 8T [1], 10T [2], and many more ...

Complex write and read assist techniques

Memory compilers for scaled and sub-VT voltages are not available

Standard-cell based memories (SCMs)

- Synthesize from standard-cell libraries, or
- from few specialized custom standard cells

SCMs have many advantages

- Immediately functional across a large voltage range
  - From sub-VT to medium-performance near-VT
- Merge with logic (where appropriate) ➔ power reduction
- Naturally implemented as two-port memories (higher bandwidth)
- Fine-granular organizations with any dimension
- Minimum design effort
  - Generic description in any HDL
  - Modifications at design time
  - Portability (unless custom cells)
  - Avoids complex power routing
Best Architectural Choices for Above/Near-\(V_T\) and Sub-\(V_T\)

Write Logic
- Clock-gates (b): smaller and less power than enable flip-flops (a)

Read Logic
- Above/Near-VT
  - Multiplexers (c): smaller, faster, and less power than tri-state buffers
- Sub-VT
  - Tri-state buffers (d): less leakage (energy) than multiplexers

Array of Storage Cells
- Latch arrays smaller than flip-flop arrays, but longer write-address setup time

Results are true for different technology nodes, fabs, and library providers

Meinerzhagen et al., MWSCAS’10; Meinerzhagen et al., JETCAS’11
Silicon area: latch arrays and flip-flop arrays vs SRAM macrocells

Up to ~1kb, SCMs smaller than SRAM macrocells

For >1kb, SCMs have higher area cost than SRAM

Meinerzhagen et al., MWSCAS’10
WID process variations in DSM CMOS compromise reliability
Under gradual voltage scaling

- **Sequential cells** fail earlier than **combinational CMOS gates** [1]

**Most reliability issues of 6T SRAM are avoided by latch**

- **Write failures**: unusually strong keeper $\leftrightarrow$ feedback disabled
- **Read failures**: degradation of SNM $\leftrightarrow$ isolated output
- **Hold failures**: SRAM bitcell = latch in non-transparent phase
  - Still **good SNM at** $V_{DD}=300\text{mV}$

**Latch resembles very much a robust 10T SRAM cell**

**SNM degradation 6T SRAM** [2]
- Transistor stacking and stretching for leakage reduction
- 3-state buffers used for read operation further reduce leakage, area, and routing

Meinerzhagen et al., ESSCIRC 2012
4kb SCM Test Chip in LP-HVT 65nm CMOS

Chip microphotograph and zoomed-in layout picture

Area cost of 12.7 $\mu m^2$ per bit (including peripherals)

Scan-chain test interface

Functionality verification: W/R random and checker-board patterns

Oven to control temperature: 27 or 37°C
Comparison with Prior-Art Sub-$V_T$ Memories

Benefits of designing 1 custom standard cell

- **Leakage power reduced by 50%** (at no area increase) w.r.t. commercial standard cell latch [Meinerzhagen et al., JETCAS’11]

**Considered work:** Full macro, measured, 65nm node

<table>
<thead>
<tr>
<th></th>
<th>[1]</th>
<th>[2]</th>
<th>[3]</th>
<th>[4]</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DDmin}$ [mV]</td>
<td>380</td>
<td>250</td>
<td>700</td>
<td>350</td>
<td>420</td>
</tr>
<tr>
<td>$V_{DDhold}$ [mV]</td>
<td>230</td>
<td>250</td>
<td>500</td>
<td>250</td>
<td>220</td>
</tr>
<tr>
<td>$E_{tot/bit}$ [fJ/bit]</td>
<td>54 (0.4V)</td>
<td>86 (0.4V)</td>
<td>-</td>
<td>55</td>
<td>14 (0.5V)</td>
</tr>
<tr>
<td>$P_{leak/bit}$ [pW/bit]</td>
<td>7.6 (0.3V)</td>
<td>6.1</td>
<td>6.0, 1.0$^a$</td>
<td>-</td>
<td>0.5</td>
</tr>
</tbody>
</table>

$^a$ Leakage-power of bitcell only


- Lowest leakage-power/bit ever reported in 65nm CMOS
- Lowest active energy/bit-access ever reported in 65nm CMOS
- Reduce leakage in array and periphery!

[D. Sylvester, ISCAS’11] has lower leakage power in 180nm CMOS

Meinerzhagen et al., ESSCIRC 2012
Significance Driven Data Protection: A New Way of Dealing with Reliability Issues at Scaled Voltages
Technology scaling and near threshold computing improves energy efficiency

Voltage scaling compromises reliable operation due to
- Process variations (permanent faults)
- Timing violations (data dependent faults)
- Single event upsets (temporary faults)

BUT, biomedical systems must continue to operate within reliability limits

Need for recovery mechanisms to restore «sufficiently reliable» operation

---

Dreslinski et al. IEEE, Feb 2010
Protection against errors can be done in hardware of software

**Hardware protection**
- Error correcting codes (ECC)
- Robust
- High energy and area cost

**Firmware/software protection**
- Backward error correction
- Forward error correction (SW-based ECC)

State-of-the-art methodology: protect everything for **100% reliable operation**

Do we need to protect all data and operations equally?

---

Kim et al. MICRO-40
Main idea: Computations do not contribute equivalently to the QoS

Equivalent computation vulnerability ✗
- Protection overhead is significant

Significance-based vulnerability ✓
- Few critical computations: guaranteed protection
- Many non-critical: best-effort protection

Significance-Driven Computation can achieve large energy savings while ensuring minimal QoS degradation
Data compression on the sensor node: reduce data to the essence to reduce power for radio communication!

Need for low-power data compression on sensor node

Courtesy of ESL@EPFL
Case Study: DWT-based ECG Monitoring

ECG data compression using Wavelet transform

- High compression ratio (up to 80%)
- High SNR (up to 25 dB)

Signal representation in time- and wavelet domain:
- Wavelet: energy concentrated in few coefficients → Hint to unequal significance
Significance sensitivity analysis
- Black box approach
- Inject error \( k \) and observe faulty output \( Y_k \)

Sensitivity metric: Percentage root-mean square difference (PRD)
- \( PRD < 2\% \) “very good signal”
- \( 2\% \leq PRD \leq 9\% \) “good signal”
- \( PRD > 9\% \) “not suitable signal”

Simulation setup/conditions
- Sensor: 3 lead ECG
- ECG database: MIT-BIH
- Processor: TamaRISC @ 2MHz
- Data memory: 32KB (2 Byte words)
DWT Selected Significance cases

Significant data: full protection
- I: 12.5% significant
- II: 25% significant
- III: 37.5% significant
- IV: 50% significant

Significant data
- Full software based error correction

Less significant data
- Only detection of errors
- Remove erroneous data
Error injection

- Rate: $1e^{-6}$ to $1e^{-9}$ bit/cycle
- Effective error occurrence: <40% of less significant data

Case I and II (12.5/25% significant)
- 80% less overhead, 37% energy savings
- But can not meet QoS requirements

Case III performance (37.5/50% sign.)
- 43% less overhead, 20% energy savings
- Guarantee QoS

Significance driven computation achieves 20% lower energy consumption, while guaranteeing good quality signal
Conclusions

Voltage frequency scaling all the way down to sub-VT is a key technology for achieving ultra-low-power embedded systems.

Optimization of software-programmable architectures for low-power consumption across all layers of abstraction:
- Consider the right balance between complexity and voltage scaling
- Memory is often as important as logic, especially at scaled voltages

Application specific processors bring significant benefit for memory reduction and enabling more voltage scaling.
Conclusions

Multi-core architectures are an efficient means for ULP operation provided that

- application characteristics are properly exploited
- Systems are optimized to exploit synergies between cores

Finally, reliability issues must be addressed. New design paradigms such as deviation from 100% correct operation are promising ideas.
Collaborators and Partners at

... Embedded Systems Laboratory at EPFL (Prof. David Atienza)
... Micrel Lab at University of Bologna (Prof. Luca Benini)
... Lund University (Prof. J. Rodrigues)
Thank you for your attention!

Q & A

andreas.burg@epfl.ch