The Quest for Ultra-Low Energy Computation

or

Opportunities for Architectures Exploiting Low-Current Devices

Jan M. Rabaey
http://www.eecs.berkeley.edu/~jan

A Historical Perspective (DEC/Compaq)

EV4
- 200MHz @100°C & 3.3V
- 16 gate delays per cycle
- 30W @200MHz & 3.3V
- 13.9mm x 16.8mm (233 mm²)
- 1.7 Million Transistors
  ~ 0.85 Million Logic Transistors

EV5
- 350MHz @100°C & 3.3V
- 14 gate delays per cycle
- 60W @350MHz & 3.3V
- 16.5mm x 18.1mm (298 mm²)
- 9.3 Million Transistors
  ~ 2.5 Million Logic Transistors

EV6
- 575MHz @100°C & 2.2V
- 12 gate delays per cycle
- 90W @575MHz & 2.2V
- 16.7mm x 18.8mm (314 mm²)
- 15.2 Million Transistors
  ~ 6 Million Logic Transistors

EV7
- Clock frequency >1.0GHz @ 1.5V
- 100W
- ~350mm²
- ~100 Million transistors

EV8
- Clock frequency range 1.0-2.0GHz (0.125 micron)
- <150W
- ~250 Million transistors

Slides Courtesy of Bill Herrick (Compaq)
**Micro-Architecture Trends**

- **Trends have included**
  - Wider super-scalar machines, deep pipelines
  - Larger register, L1 caches
  - On-chip L2 caches
  - Out of order execution
  - Sophisticated branch prediction, predication, speculation
  - Integrated memory and network controllers
  - SMT
  - Less idle logic but more bookkeeping logic

- **Future opportunities include**
  - Floating point performance improvements
  - Vectors
  - Thread-level speculation
  - More pipelining
  - Better on-chip communications
    - Banking, replicating structures
    - Clustering functional units
  - On-chip SMP

---

**Complexity Trends**

- **Process scaling has continued steadily**
- **Planarization has enabled an increase in the number of interconnect layers**
- **Transistor counts have increased dramatically with the L2 cache SRAMs**
- **Additionally, design team size has increased ~40% per generation**

**Opportunities to manage complexity and productivity**
- Fundamental understanding and modeling of process and circuit element behaviors
- High level design methods
- CAD
- Design reuse
- Micro-architecture
**Performance Trends**

- Performance has increased significantly (7x) faster than frequency.
- Performance tracks transistor count when L2 cache ignored.
  - Transistor budget has increased more than performance when L2 cache is considered (!!).
- Opportunities to continue performance improvements.
  - Continued scaling of devices, interconnect and dielectrics.
  - Clock distribution.
  - Micro-architecture.
  - System design.

**Power Dissipation Trends**

- Power consumption is increasing.
  - Power density increased with approximately factor 2 (0.2 -> 0.375 W/mm²).
  - Better cooling technology needed.
- Supply current is increasing faster.
  - mA/MIP is not scaling.
- On-chip signal integrity will be a major issue.
- Power and current distribution are critical.
- Opportunities to slow power growth.
  - Accelerate Vdd scaling.
  - Low κ dielectrics & thinner (Cu) interconnect.
  - SOI circuit innovations.
  - Clock system design.
  - Micro-architecture.
Challenging Design Trends

- Micro-architecture and logic design are stressed as frequency has increased faster than scaling
- Further reducing the number of gate delays per cycle will be difficult
- Cycles to communicate across chip track with frequency
- Clock edge rates are not scaling
- Opportunities to continue performance increases
  - Chip implementation design
  - Clock system design
  - Micro-architecture

Digital Processor Performance

Courtesy of Ravi Subramanian (Morphics)
The Law of Diminishing Returns

- More transistors are being thrown at improving general-purpose CPU and DSP performance

- Fundamental bounds are being pushed
  - limits on instruction-level parallelism
  - limits on memory system performance

- Returns per transistor are diminishing
  - new architectures realizing only 2-3 instructions/clock
  - increasingly large caches to hide DRAM latency

Some observations

- Von-Neuman style instruction set architectures were perceived when switching devices and interconnections were extraordinarily expensive, and multiplexing-in-time provided the most economical solution
  - Intel 4004: 2000 transistors, 1 MHz clock frequency, 1 metal layer

- This led to the “clock-speed” affixation, which in fact is only a secondary measure of performance

- Power is rapidly becoming a limiting factor
  - Newest processors are including thermal sensors and automatic slow-down (throttling) using pipeline bubbles and nop’s to combat overheating and meltdown
The Distributed Approach to Information Processing

Computation Bounded by Energy Limitations

The Changing Metrics

Power

Cost

Flexibility

Performance as a Functionality Constraint
(“Just-in-Time Computing”)
A Holistic Perspective on Low-Energy Design

Energy = upper bound on the amount of available computation

- Total Energy of Milky Way Galaxy: $10^{59}$ J
- Minimum switching energy for digital gate (1 electron@100 mV): $1.6 \times 10^{-20}$ J (limited by thermal noise)
- Upper bound on number of digital operations: $6 \times 10^{78}$
- Operations/year performed by 1 billion 100 MOPS computers: $3 \times 10^{24}$
- Energy consumed in 180 years assuming a doubling of computational requirements every year.

The Battery Limitation

- Energy cost of digital computation (embedded)
  - 1999 (0.25μm): 1pJ/op (custom) ... 1nJ/op (μproc)
  - 2004 (0.1μm): 0.1pJ/op (custom) ... 100pJ/op (μproc)
    - Factor 1.6 per year; Factor 10 over 5 years
  - Compared to minimum switching energy (for deterministic computing): $1.6 \times 10^{-20}$ J @ 300 °K (1 electron, 100mV)
- Assume: energy per digital operation (2004): 100 pJ
- Lithium-Ion: 220 Watt-hours/kg == 800 Joules/gr
- At 100 pJ/operation: 8 teraOps/gr!
  - Equivalent to continuous operation at 100 MOPS for 22 hours (@ average power dissipation of 10 mW)
### The Holy Grail: Energy Scavenging

<table>
<thead>
<tr>
<th>Energy Sources</th>
<th>Power (Energy) Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batteries (Zinc-Air)</td>
<td>1050 - 1560 mWh/cm³</td>
</tr>
<tr>
<td>Batteries (rechargeable Lithium)</td>
<td>300 mWh/cm³ (3 - 4 V)</td>
</tr>
<tr>
<td>Solar</td>
<td>15 mW/cm² - direct sun</td>
</tr>
<tr>
<td>Vibrations</td>
<td>0.05 - 0.5 mW/cm²</td>
</tr>
<tr>
<td>Inertial Human Power</td>
<td></td>
</tr>
<tr>
<td>Acoustic Noise</td>
<td>3E-6 mW/cm² at 75 Db</td>
</tr>
<tr>
<td>Non-Inertial Human Power</td>
<td>1.8 mW (Shoe inserts)</td>
</tr>
<tr>
<td>Nuclear Reaction</td>
<td>80 mW/cm³</td>
</tr>
<tr>
<td>One-Time Chemical Reaction</td>
<td></td>
</tr>
<tr>
<td>Fluid Flow</td>
<td>300 - 500 mW/cm²</td>
</tr>
<tr>
<td>Fuel Cells</td>
<td>~4000 mWh/cm³</td>
</tr>
</tbody>
</table>

**SOURCE:**
P. Wright & S. Randy
UC ME Dept.

---

### Example: MEMS Variable Capacitor

Out of the plane, variable gap capacitor

Up to 10 µW of power demonstrated
100 µW seems to be reasonable target

---

**Integrated Manufacturing Lab**

![Generator Diagram](image)

1. Up to 10 µW of power demonstrated
2. 100 µW seems to be reasonable target
• Voltage as a Design Variable
  – Match voltage and frequency to required performance
• Minimize waste (or reduce switching capacitance)
  – Match computation and architecture
  – Preserve locality inherent in algorithm
  – Exploit signal statistics
  – Energy (performance) on demand

✪ Easier accomplished in application-specific than programmable devices

---

Why not use fixed direct-mapped architectures?

• Move to deep sub-micron technology
  – Growing Design Cycle Times At Odds With Shrinking Product Cycle Times
    • rapidly increasing product integration cycles
    • increasingly constrained design resources
    • sharp increases in cost of “trying out” an idea- NRE
    • verification issues dominate design cycle time
  – Leads to “platform-based” design strategy
• After-fabrication flexibility an important asset
  – Reduces risks
  – Enables multi-standard / multi-function operation
  – Enables adaptation to environmental conditions -> leads to important system-level energy conservation
The Energy-Flexibility Gap

- **Dedicated HW**
  - Energy Efficiency: 0.4 MIPS/mW
- **Reconfigurable Processor/Logic**
  - Pleiades: 10-80 MOPS/mW
  - 2 V DSP: 3 MOPS/mW
- **Embedded Processors**
  - SA110: 0.4 MIPS/mW

Programming in Space: Merging Efficiency and Versatility

Spatially programmed connection of processing elements.

“Hardware” customized to specifics of problem.
- Direct map of problem specific dataflow, control.
- Circuits “adapted” as problem requirements change.
Spatial vs. Temporal Computing

Spatial

Example: FPGAs
The Basic Computational Element

Temporal

### Example: FPGAs

**The Basic Computational Element**

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

2-LUT
(look-up table)
FPGAs: The Architectural Model

Spatial/Configurable Benefits

- 10x raw density advantage over processors (and increasing)
- Energy efficiency (potentially)
- Locality, regularity, and predictability
- Ultimate distributed architecture
- Scalable with technology
  - Relies mostly on increase in computational density
  - Avoids most of the physics pitfalls threatening high-performance computing
**Spatial/Configurable Drawbacks**

- **Resource management**
  - Each compute/interconnect resource dedicated to single function
  - Must dedicate resources for every computational subtask
  - Infrequently needed portions of a computation sit idle --> inefficient use of resources
  - But … not a real issue when transistors are abundant

- **Potential mismatch between operations and operators**

- **Interconnect plays dominant role**
Example: Covariance Matrix Computation

```plaintext
for (i = 1; i <= length; i++) {
    for (k = i; k <= length; k++) {
        phi[i][k] = phi[i-1][k-1] +
                   in[NP-i] * in[NP-k] -
                   in[NA-1-i] * in[NA-1-k];
    }
}
```
Impact of Architectural Choice

Example: 16 point Complex Radix-2 FFT (Final Stage)

For Spatial Architectures

- Interconnect dominant
  - area
  - power
  - time

...so need to understand in order to optimize architectures
Spatial Efficiency

Interconnect also dominates power

XC4003A data from Eric Kusse (UCB MS 1997)
Interconnect can be managed!

Use of hierarchy, matching computational needs
Use of circuit techniques (enabled by predictable, regular structure)

Levels of interconnect targeting different connectivity lengths

Inverse Clustering

- Blocks further away are connected at the lowest levels
- Inverse clustering complements Mesh Architecture
Low-Energy Embedded FPGA

- Test chip
  - 8x8 CLB array
  - 5 in - 3 out CLB
  - 3-level interconnect hierarchy
  - 4 mm² in 0.25 µm ST CMOS
  - 0.8 and 1.5 V supply

- Results
  - 125 MHz Toggle Frequency
  - 50 MHz 8-bit adder
  - energy 70 times lower than comparable Xilinx

Source: Bill Dally (Stanford)

On-Chip Interconnection Networks

- Many modules, same global wiring
  - carefully optimized wiring
  - well characterized
  - optimized circuits
    - 0.1x power 0.3x delay

- Efficient protocols
  - static
  - statically scheduled
  - dynamic routing with pipelined control

- Standard interface

Source: Bill Dally (Stanford)
Circuits for On-Chip Networks

Uniform, well characterized lines enable custom circuits - 0.1x power, 3x velocity

H-bridge driver
100mV swing

Long, lossy RC lines
Regenerative Repeaters

Programming-in-Space - Summary

• Similar computational power/area, substantially lower switching speeds; substantially lower energy
• Where applicable?
  – “Printing of algorithms onto silicon”
  – Function oriented (as is the case in most embedded applications)
• Requirements
  – Dense integration of memory and function
  – Efficient implementation of programmable interconnect (switches/memory)
Spatial Computation: The Low-Current Device Window of Opportunity

- Massive number of cheap devices desirable
- No real need for the maximum switching speed
- Multiple active layers desirable (integrating switching into the interconnect fabric)
- Tight integration with sensing, displaying, and energy generation

\[ \Rightarrow \text{The self-contained integrated sensor-monitor-communication node} \]

Some Healthy Conclusions

Don’t use more transistors to stretch general-purpose performance, whether for CPUs, DSPs, or reconfigurable logic.

Don’t use more time to design dedicated hardwired solutions in cases where mass customization is what the market demands.

Spatial computation combines flexibility with efficiency, while being easy on switching speed.

\[ \Rightarrow \text{The window of opportunity for low-current electronics (TFT, organic, molecular)} \]