Low Power Systems on a Chip – Today's Challenge

Trevor Mudge
Bredt Professor of Engineering
The University of Michigan, Ann Arbor
July 2004
Outline

- Why low power? – Trends
- Dynamic power management
- Static power management
- Power and design uncertainty
- Conclusions
Why does power matter?

<table>
<thead>
<tr>
<th>Year</th>
<th>Model</th>
<th>Battery life</th>
<th>Battery</th>
<th>Weight</th>
<th>Features</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1983</td>
<td>Motorola DynaTAC 8000X</td>
<td>0.5h - 1h talk, 8h standby</td>
<td>Lead Acid, 500g</td>
<td>800g</td>
<td>Talk for brokers</td>
<td>$3995</td>
</tr>
<tr>
<td>1995</td>
<td>Nokia 232</td>
<td>1h talk, 13h standby</td>
<td>NiMh, 100g</td>
<td>205g</td>
<td>Talk for the masses</td>
<td>$500</td>
</tr>
<tr>
<td>2003</td>
<td>Nokia 6600</td>
<td>4h talk, 240h (&gt;1 week) standby</td>
<td>Li-Ion, 21g</td>
<td>125g</td>
<td>Talk, play, web, snap, video, organize</td>
<td>$500</td>
</tr>
</tbody>
</table>

- The disappearing battery – despite only incremental capacity improvements: the rest of the system has become more power efficient
- Power has major impact on form factor, features, and cost
First-class design issue: Power

- What the end-users really want: supercomputer performance in their pockets...
  - Untethered operation, always-on communications
  - Driven by applications (games, positioning, advanced signal processing, etc.)

- Technology scaling trends are not in our favour:
  - Need creative ways of dealing with increasing leakage power
  - New processes are expensive
  - Diminishing performance gains from process scaling
  - Dynamic power remains high

- Solutions need to cut across traditional boundaries (SW / architecture / microarch / circuits)
Technology Trends

- Silicon is likely to be the technology of choice for the next 4-5 generations
  - 130nm ➔ 90nm ➔ 60nm ➔ 45nm ➔ 30nm
  - 2 or 3 years between generations
  - ~10 ± 2 Years
  - after 2015 there may be a paradigm shift to non-Si technology
- Consequences for power ➔

Normalized Total Chip Power Dissipation vs. Technology node

- Dynamic Power
- Sub-threshold Leakage
- Gate-oxide Leakage

Possible trajectory if high-k dielectrics reach mainstream production

Normalized to data from ITRS 2001 roadmap
Dynamic Power

- Voltage scaling – by now a familiar story
  - relies on the quadratic law (also helps leakage)
  - familiar but not widely implemented
    - LongRun 1 & 2, SpeedStep, DPM (dynamic power mgmt/IBM)

- The quadratic law implies that parallelism is good
  - only true if leakage is not considered
CMOS Power and Energy

- Power and Energy consumption trends of a workload running at different frequency and voltage levels.
- DFS: frequency scaling only, DVS: frequency & voltage scaling

\[ f \sim \frac{(v_{dd} - v_t)\alpha}{v_{dd}} \]
\[ \alpha \approx 1.3 \]
\[ v_t / v_{max} \approx 0.3 \]

\[ P = C v_{dd}^2 f + v_{dd} I_{leak} \]
Avg. power ~ heat

\[ E = \int P dt \]
Need DVS to save energy

Must reduce voltage to save energy and extend battery life
Performance scaling for energy efficiency

- Reduced processing rate enables more efficient operation
  - Use dynamic voltage scaling (DVS) and threshold scaling (ABB)
Commercial Example: IEM

4 performance (frequency and voltage) levels available in benchmarked system

Closest available performance level of system

Performance level requested by algorithm

2 seconds

ARM Intelligent Energy Manager™
IEM test chip: AM926EJ-S core
DVS926: Power Analysis

- Room Temp
- Cached workload (Dhrystone)
- CPU core V/I measurements
- Four run-time frequency divisions: 100%, 75%, 50%, 25%
- Boot time PLL settings:
  - 216MHz
  - 228MHz
  - 240MHz
  - 252MHz
  - 264MHz
  - 276MHz
  - 288MHz
  - 300MHz

Core power vs CORECLK [Room Temp]
Static Power

- Growing importance
- Reducing activity no longer works
- Voltage scaling still works
- Powering off works if state is not an issue
- Memory is important

![Graph showing normalized total chip power dissipation over time with various labels for different power components and technology nodes.](image-url)
Cache Leakage Power

- On-chip caches are becoming bigger
  - 2x64KB L1 / 1.5MB L2 for Alpha 21464
  - 256KB L2 / 3MB(6MB) L3 for Itanium 2

- Increasing on-chip cache leakage power
  - Feature size shrinking / $V_{TH}$ decreasing
  - Increasing fraction of leakage power by L2 & L3 caches
    - Consuming constant leakage power
    - Less frequent access (less dynamic power)

- We can maintain cache performance by trading cache size for power
  - Counter intuitive: larger caches consuming less power
Cache Miss Statistics

- SPEC 2000 INT/FP – average all 25 benchmarks
- Using “sim-cache” in SimpleScalar suite
- L2 miss rate is local miss rate
Optimizing L2 leakage at fixed L1 size

Based on fast 16KB L1

- Constraint – maintaining the same AMAT
- Optimization – use larger but less leaky L2 caches

Graph showing leakage (W) versus AMAT (ns/access) for cache sizes 256KB, 512KB, and 128KB, with corresponding percentages of improvement: 69% and 85%.
L2 Leakage Saving at Fixed L1 Size

**Conclusion**

- Cost-effective # of VTH for cache leakage reduction
  - depending on the target access time, but 1 or 2 high VTH's is enough for leakage reduction

- Cache leakage
  - another design constraint in processor design
  - trade-off among delay / area / leakage

---

<table>
<thead>
<tr>
<th>L2 size</th>
<th>128KB</th>
<th>256KB</th>
<th>512KB</th>
<th>256KB</th>
<th>512KB</th>
<th>1024KB</th>
<th>512KB</th>
<th>1024KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16KB</td>
<td>100%</td>
<td>31.3%</td>
<td></td>
<td>100%</td>
<td>14.5%</td>
<td></td>
<td>100%</td>
<td>10.9%</td>
</tr>
<tr>
<td>32KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.4%</td>
</tr>
<tr>
<td>64KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.7%</td>
</tr>
</tbody>
</table>

Trevor Mudge
Advanced Computer Architecture Lab
The University of Michigan
L1 Leakage Reduction: Drowsy Caches

Instead of being sophisticated about predicting the working set, reduce the penalty for being wrong

Algorithm:
- Periodically put all lines in cache into drowsy mode.
- When accessed, wake up the line.

- Optimize across circuit-microarchitecture boundary:
  - Use of the appropriate circuit technique enables simplified microarchitectural control.

- Requirement: state preservation in low leakage mode.
Drowsy Memory Using DVS

- Low supply voltage for inactive memory cells
  - Low voltage reduces leakage current too!
  - Quadratic reduction in leakage power 😷 $P_{\downarrow\downarrow} = I_{\downarrow} \times V_{\downarrow}$

![Diagram showing low supply voltage for normal and drowsy modes](diagram.png)
Drowsy Cache Line Architecture

- VDD (1V)
- VDDLow (0.3V)
- drowsy signal
- drowsy bit
- drowsy (set)
- wake up (reset)
- voltage controller
- drowsy
- power line
- SRAMs
- word line
- word line gate
- row decoder
- word line driver
Energy Reduction

High leakage: lines have to be powered up when accessed

Drowsy circuit
- Without high $v_t$ device (in SRAM): 6x leakage reduction, no access delay.
- With high $v_t$ device: 10x leakage reduction, 6% access time increase.
Power and Design Uncertainty

- Increasing uncertainty with process scaling
  - Inter- and intra-die process variations
  - Temperature variation
  - Power supply drop
  - Capacitive and inductive noise

- Impact on traditional design:
  - Addressing worst-case variation in design requires *large safety margins*
  - Higher energy / lower performance
  - Reduced yield
  - Difficulty in design closure

- Key Observation: worst-case conditions also highly improbable
  - Significant gain for circuits optimized for common case
  - Efficiency mechanisms needed to tolerate infrequent worst-case scenarios
Reducing Voltage Margins with Razor

**Goal:** reduce voltage margins with *in-situ* error detection and correction for delay failures

**Proposed Approach:**
- Tune processor voltage based on error rate
- Eliminate safety margins, purposely run *below* critical voltage
  - Data-dependent latency margins
  - Trade-off: voltage power savings vs. overhead of correction
- Analogous to wireless power modulation & power/reliability trade-offs in DSP
Razor Flip-Flop Implementation

- Compare latched data with *shadow-latch* on delayed clock

- Upon failure: place data from shadow-latch in main latch
  - Ensure shadow latch always correct using conservative design techniques

- Key design issues:
  - Maintaining pipeline forward progress
  - Recovering pipeline state after errors
  - Short path impact on shadow-latch
  - Meta-stable results in main flip-flop
  - Power overhead of error detection and correction
Centralized Pipeline Recovery Control

- Once cycle penalty for timing failure
- Global synchronization may be difficult for fast, complex designs
Razor Prototype Implementation

- 4 stage 64-bit Alpha pipeline
  - 200MHz expected operation in 0.18\(\mu\)m technology, 1.8V, ~500mW

- Razor overhead:
  - Total of 192 Razor flip-flops out of 2408 total (9%)
  - Error-free power overhead:
    - Razor flip-flops: < 1%
    - Short path buffer: 2.1%
  - Recovery power overhead:
    - Razor latch power overhead: 2% at 10% error rate
    - Additional power overhead due to re-execution of instructions
Error Rate Studies – Empirical Results

18x18-bit Multiplier Block at 90 MHz and 27 C

- 35% energy savings with 1.3% error
- 30% energy saving
- 22% saving

Environmental-margin @ 1.69 V  Safety-margin @ 1.63 V  Zero-margin @ 1.54 V

Supply Voltage (V)

Error rate

once every 20 seconds!
Multiplier Bit-flip Analysis

- Error rates similar to initial experiment (right)
- Occasional multi-bit flips require more complex error correcting schemes
- Razor overhead much lower than other schemes for full fault coverage
Summary for Razor

- **Razor benefits:**
  - Efficient in-situ timing error detection and correction for worst-case timing failures
  - Eliminate process, environmental, and safety margins necessary in DVS
  - Data dependent speculation for sub-critical voltage operation
  - Allow design for common case – “better than worst case design”

- **Other applications:**
  - Over-clocking for performance improvement (2x shown among hobbyists)
  - Clock skew tuning to off-set process / ambient variations
  - Automatic adjustment to process variation
Conclusions

- Reducing power is the #1 issue facing designers of digital systems
  - true even if they are not mobile
- Dynamic power will continue to be a challenge
- Static power will to although there are partial technological solutions on the horizon
  - high-k dielectrics – oxide leakage
  - finFETs – subthreshold leakage
  - Feature size reductions – dynamic
- Trends that may help reduce power will also introduce uncertainty in the process of manufacturing chips – *the next major challenge*