Dynamic Power Management in Embedded Systems

Waltenegus Dargie
TU Dresden
Chair of Computer Networks
Outline

• Motivation
• Aspects of power consumption
• Selective switching
• Dynamic scaling
• Research Issues
Motivation

• Power dissipation
  – Due to circuit leakage
  – Unforeseen circumstances
  – Overhead of the operating system calls
  – Inefficient code
  – System resource utilisation

• This talk focuses on the last aspect only
Motivation

• Embedded system
  – Smart phones
  – Personal MP3 players
  – Point of sale mobile devices
  – Portable GPS navigators
  – Digital cameras
  – Digital video cameras
  – Pen digitizers
  – Handheld OCR
Motivation

- The global market for embedded systems is expected to increase from $92.0 billion in 2008 to an estimated $112.5 billion by the end of 2013, a compound annual growth rate (CAGR) of 4.1%.
- Embedded hardware was worth $89.8 billion in 2008 and is expected to grow at a CAGR of 4.1% to reach $109.6 billion in 2013.
- Embedded software generated $2.2 billion in 2008. This should increase to $2.9 billion in 2013, for a CAGR of 5.6%.

Source: BCC, April 2009
Motivation

Algorithmic Complexity (Shannon’s Law)

Processor Performance (Moore’s Law)

Battery Capacity

Source: EMUCO Project 2008
Outline

- Motivation
- Aspects of power consumption
- Selective switching
- Dynamic scaling
- Research Issues
Embedded Sys.: Architecture

Analogue Baseband

- I/O Interface
- ADC/DAC (Audio/Video)
- Filters and Synthesisers controllers
- Power/Voltage Amplifiers
- Receiver
- Synthesiser
- Modulator
- Flash Memory
- Microcontroller
- Memory Unit
- DSP

Digital Baseband

- Keyboard
- Display
- SIM card

RF

- Power/Voltage Amplifiers
Power Consumption

Source: Vargas, 2005

Source: Siemens
Power Consumption

Source: Siemens

Source: Vargas, 2005
Power Consumption

• Batteries
  – Specified by a rated current capacity, C, expressed in Ampere-Hour (mAh)
    • Drawing current at a rate greater than the discharge rate results in a current consumption rate higher than the rate of diffusion of the active elements in the electrolyte.
    • If this process continues for a long time, the electrodes run out of active material even though the electrolyte has not yet exhausted its active materials.
    • This situation can be overcome by intermittently drawing current from the battery.
## Power Consumption

<table>
<thead>
<tr>
<th>Discharge Rate</th>
<th>Battery Capacity Normalised to 1C Discharge Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/5</td>
<td>107%</td>
</tr>
<tr>
<td>C/2</td>
<td>104%</td>
</tr>
<tr>
<td>1C</td>
<td>100%</td>
</tr>
<tr>
<td>2C</td>
<td>94%</td>
</tr>
<tr>
<td>4C</td>
<td>86%</td>
</tr>
</tbody>
</table>

Source: Bellosa, 2000: Lithium-ion Battery
Outline

- Motivation
- Aspects of power consumption
- Dynamic Power Management
- Selective Switching
- Dynamic scaling
- Research Issues
• Fundamental premises about Embedded systems:
  – Predominantly event-driven
  – Experience non-uniform workload during operation time
• DPM\(^1\) refers to selectively shutting-off and/or slowing-down system components that are idle or underutilised
• A policy determines the type and timing of power transitions based on system history, workload and performance constraints

1. DPM: Dynamic power management
Concept

• It has been described in the literature as a linear optimisation problem
  – The objective function is the expected performance
    • Related to the expected waiting time and the number of jobs in the queue
  – The constraint is the expected power consumption
    • Related to the power cost of staying in some operation state and the energy consumption for the transfer from one server state to the next
Architecture

Scheduler

*task arrival rates, priority of tasks, task deadlines*

Task1

Workload and Energy monitoring

Task runtime, frequency and duration of accessed resources

Power mode adaptation

Hardware profile *operating points*
Outline

• Motivation
• Aspects of power consumption
• Dynamic Power Management
• Selective Switching
• Dynamic scaling
• Research Issues
Selective Switching

- Power state machine for the StrongARM-1100 processor

Source: Benini, 2000
### Selective Switching

<table>
<thead>
<tr>
<th>Sleep Mode</th>
<th>Active clock domains</th>
<th>Oscillators</th>
<th>Wake up sources</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>clk&lt;sub&gt;CPU&lt;/sub&gt;</td>
<td>clk&lt;sub&gt;FLASH&lt;/sub&gt;</td>
<td>clk&lt;sub&gt;IO&lt;/sub&gt;</td>
</tr>
<tr>
<td>Idle</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ADC noise red.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>power down</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power save</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>standby</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ext. standby</td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Active Clock Domains and Wake Up Sources in the Different Sleep Modes

Source: ATMEL, Atmega 128: 2008
Selective Switching

- Memory access

Source: Ellis, 2003
Selective Switching

- Memory access

Source: Ellis, 2003

1. MMU: Memory Management unit
## Selective Switching

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>StrongARM</th>
<th>Memory</th>
<th>MEMS &amp; ADC</th>
<th>RF</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₀</td>
<td>Sleep</td>
<td>Sleep</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>P₁</td>
<td>Sleep</td>
<td>Sleep</td>
<td>On</td>
<td>Off</td>
</tr>
<tr>
<td>P₂</td>
<td>Sleep</td>
<td>Sleep</td>
<td>On</td>
<td>RX</td>
</tr>
<tr>
<td>P₃</td>
<td>Idle</td>
<td>Sleep</td>
<td>On</td>
<td>RX</td>
</tr>
<tr>
<td>P₄</td>
<td>Active</td>
<td>Active</td>
<td>On</td>
<td>TX, RX</td>
</tr>
</tbody>
</table>

Source: Sinha and Chandrakasan, 2001
Selective Switching

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{\text{on}}$</td>
<td>10 W</td>
</tr>
<tr>
<td>$P_{\text{off}}$</td>
<td>0 W</td>
</tr>
<tr>
<td>$P_{\text{on} \rightarrow \text{off}}$</td>
<td>10 W</td>
</tr>
<tr>
<td>$P_{\text{off} \rightarrow \text{on}}$</td>
<td>40 W</td>
</tr>
<tr>
<td>$t_{\text{on} \rightarrow \text{off}}$</td>
<td>1 s</td>
</tr>
<tr>
<td>$t_{\text{off} \rightarrow \text{on}}$</td>
<td>2 s</td>
</tr>
<tr>
<td>$t_R$</td>
<td>25 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Policy</th>
<th>Energy</th>
<th>Avg. Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always on</td>
<td>250 J</td>
<td>1 s</td>
</tr>
<tr>
<td>Reactive greedy</td>
<td>240 J</td>
<td>3 s</td>
</tr>
<tr>
<td>Power-aware</td>
<td>140 J</td>
<td>2.5 s</td>
</tr>
</tbody>
</table>

Source: Pedram, 2003
Selective Switching

![Diagram showing power (W) vs. time (s) with points Pi and Pj and t_{th,j} highlighted]
Selective Switching

\[ E_j^{\text{saved}} = P_i \left( t_j + t_{i,j} + t_{j,i} \right) - \left[ P_{i,j} \times t_{i,j} + P_{j,i} \times t_{j,i} + P_j \times t_j \right] \]

\[ t_{th,j} \geq \max \left( 0, \frac{(P_i - P_{i,j}) t_{i,j}}{P_i - P_j} \right) \]
## Selective Switching

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>StrongARM</th>
<th>Memory</th>
<th>MEMS &amp; ADC</th>
<th>RF</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₀</td>
<td>Sleep</td>
<td>Sleep</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>P₁</td>
<td>Sleep</td>
<td>Sleep</td>
<td>On</td>
<td>Off</td>
</tr>
<tr>
<td>P₂</td>
<td>Sleep</td>
<td>Sleep</td>
<td>On</td>
<td>RX</td>
</tr>
<tr>
<td>P₃</td>
<td>Idle</td>
<td>Sleep</td>
<td>On</td>
<td>RX</td>
</tr>
<tr>
<td>P₄</td>
<td>Active</td>
<td>Active</td>
<td>On</td>
<td>TX, RX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>P (mW)</th>
<th>Trans. Latency (ms)</th>
<th>Threshold, $T_{th}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₄</td>
<td>1,040</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>P₄</td>
<td>400</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>P₂</td>
<td>270</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>P₁</td>
<td>200</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>P₀</td>
<td>10</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

Source: Sinha and Chandrakasan, 2001
Selective Switching

1. Process event and compute Resource Consumption
2. Observation time is up?
   - Yes: For c, c = 1, 2, ..C: Compute \( \lambda_c \);
       Compute \( t_{th,c} \);
       Calculate \( t_{agg} \).
     - No: \( m = M-1 \)
3. If \( m > 0 \):
   - Yes: If \( t_{agg} \geq t_{h,m} \):
     - Yes: power mode = \( P_m \);
       pmDirty = true;
     - No: \( m = M-1 \)
   - No: \( m = M-1 \)
4. If(pmdirty)
   - Yes: Set power;
     pmDirty = false;
   - No:

Diagram
Selective Switching

• Selective switching should be application dependent

• Weissel et al. demonstrate that long beacon periods in IEEE 802.11 wireless local area networks do not necessarily result in power saving for some applications
  – Continuous-aware mode, power-saving mode and adaptive power saving mode
    • Periodical activation to synchronise with the server
    • The length of the sleep interval is called beacon period (default value = 100 ms)
Selective Switching

Mode Transition | Time | Energy
---|---|---
PSP to CAM | 320 ms | 280 mJ
CAM to PSP | 317 ms | 283 mJ
Beacon interval adjustment | 333 ms | 300 mJ

Application Without PM With PM
---|---|---
Beacon = Not applicable | Beacon ≥ 100 ms; period
NFS | Task: Negligible runtime | Task: 250 ms runtime

Source: Weissel et al., 2004: The runtime and energy cost of a dynamic power Management: Application: NFS; OS: Linux; Network interface: Cisco Aironet
Selective Switching

• Mitigation
  – Power management should take into account the type of application, the send/receive characteristics and the user sensitiveness and tolerance

<table>
<thead>
<tr>
<th>Application-specific DPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average size of packets received</td>
</tr>
<tr>
<td>Ratio of average length of inactive to length of active periods</td>
</tr>
<tr>
<td><strong>Ratio of average size of packets received to size of packets sent</strong></td>
</tr>
<tr>
<td>Ratio of traffic volume received to traffic volume sent</td>
</tr>
<tr>
<td>Average size of packets sent</td>
</tr>
</tbody>
</table>
Outline

• Motivation
• Aspects of power consumption
• Dynamic Power Management
• Selective Switching
• Dynamic Scaling
• Research Issues
Dynamic Scaling

• Refers to runtime change in the supply voltage and clock frequency of the hardware components

\[ P \propto C_L V_{dd}^2 f \]

\[ \tau = k C_L \frac{V_{dd}}{(V_{dd} - V_{th})^2} \]

Where

- \( C_L \): Load capacitance
- \( V_{dd} \): Supply voltage
- \( f \): Clock frequency
- \( V_{th} \): Threshold voltage
- \( \tau \): Switching delay
- \( k \): Boltzmann’s constant
Dynamic Scaling

1. Normalized Energy

DPM without voltage scaling

Efficient DVS

DPM with ideal voltage scaling
Outline

• Motivation
• Aspects of power consumption
• Dynamic Power Management
• Selective Switching
• Dynamic scaling
• Research Issues
Research Issues

• The cost of power management
• Workload arrival estimation as a basis for decision making
• Side effects of DPM
Cost

• Performance cost
• Resource cost
  – For example, MMU requires associative memory that is accessed whenever memory is referenced
  – Size and computational cost
Workload Estimation

• Given an observation period, $T$, and an average normalized workload, $w(n)$, in the interval $(n-1)T \leq t \leq nT$, the task is to decide the power mode for the next cycle.
Workload Estimation

- **FIR Filter**

\[ W_p[n] = \sum_{k=0}^{N-1} h_n[k]w[n - k] \]

<table>
<thead>
<tr>
<th>Filter</th>
<th>Filter Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving Average (MAW)</td>
<td>( h_k(i) = \frac{1}{N} )</td>
</tr>
<tr>
<td>Exponential weighted average (EWA)</td>
<td>( h_k(i) = a^{-i} )</td>
</tr>
<tr>
<td>List Mean Square (LMS)</td>
<td>( h_{n+1}(k) = h_n(k) + \mu w_e(n) w(n - k) )</td>
</tr>
</tbody>
</table>

- **Expected workload state (EWS)**

\[ w[n_i] = E\{w[n_i]\} = \sum_{j=0}^{L} w_j p_{i,j} \]
Workload Estimation

Source: Sinha and Chandrakasan, 2001
Workload Estimation

• Often task estimation is made from within the system but should consider external factors as well
  – For example in audio and video streaming, knowledge of bandwidth and data rate should be helpful
• Rich context information (operation condition) is to tackle side-effects
Side Effects

• Scaling latency
  – Power supplies require a finite amount of time to settle to the new operating voltage
  – The delay is a function of load on the supply voltage
  – Comparatively, the clock-generator requires negligible time

• Unreliable operation during transition
  • The CPU should be halted during a transition
  • Requires an external hardware
Commercial Product

Source: Lattice Semiconductor (ispPAC-POWR1208P1), 2005
Conclusion

- Most of the subsystems of an embedded system are not equally active at the same time.
- Dynamic scaling is preferred over selective switching if the long term task arrival pattern is known.
- This requires a comprehensive model for task arrival rate estimation.
- The resource cost of power management is so far the least addressed issue.
Journals and Conferences

- ACM Transactions on Embedded Computing Systems
- IEEE Transaction on VLSI Systems
- IEEE Journal of Solid-State Circuit
- IEEE Transaction on Computer-Aided Design
- IEEE Design and Test of Computers
- DATE (2010 Dresden)
- DAC (US Dominated)
- ASP-DAC (Asia-Pacific)
- ARCS (2010 Hannover)
Thanks for Listening
Significant Contributions

• Sinha and Chandrakasan (IEEE Design and Test of Computers, 2001)
  – Mathematical model (DVFS) and task arrival estimation
• Su et al. (ISLPED\textsuperscript{1} 2003)
  – Leakage estimation under power supply and temperature variations
• Weissel et al. (ARCS\textsuperscript{2}, 2004)
  – Application-aware DPM
• Kang et al. (DAC\textsuperscript{3} 2007)
  – Variation resilient circuit design technique
• Jung and Pedram (DATE\textsuperscript{4} 2008)
  – Stochastic process model as a tuple (S (power), A (V-F value), O (temperature), T, Z, c)

1. ISLPED: International Symposium on Low Power Electronics and Design
2. ARCS: International conference on architecture of computing systems
3. DAT: Design automation conference
4. DATE: Design automation and test in Europe
Scheduling

• Most research concentrates on finding optimal combinations of tasks, but does not discuss how a multiprocessor scheduler in a real system can succeed in combining tasks accordingly.
Scheduling

• It does not pay-off to co-schedule memory-bound tasks in order to be able to profit from lower chip frequencies
  – Combining tasks in order to reduce resource contention is more important than combining tasks that share a common optimal frequency.

• Co-scheduling policy that avoids contention for bottleneck resources.

• Migration policy that balances resource utilization across execution contexts