Cycle-Accurate Simulation of Energy Consumption in Embedded Systems

T. Simunic, L. Benini and G. De Micheli
Stanford University
Bologna University
Hewlett-Packard Laboratories
Motivation

SmartBadge

- Small, portable design
- High performance
- Long battery life
- Discrete components
Problem

- Original SmartBadge running MPEG video decode
  - low performance: 6 frames/sec; need at least 20 frames/sec
  - high energy consumption → very low battery life
  - both hardware and software redesign needed

- Problem:
  - Hardware redesign requires evaluating multiple architecture options
  - Software design directly depends on the hardware architecture
  - Multiple iterations of board design are costly and slow
  - Verilog simulations are still too slow
  - Instruction-level simulator has only performance models
Cycle-accurate energy consumption simulator within 5% accuracy of hardware measurements at speed comparable to the instruction-level simulator.
Previous Work

- Architecture-level power modeling [eg. Landman et al., Liu]
  - analysis done at the netlist level
  - technology files required

- Energy and performance models of design components
  - cache [Kamble et al., Wilton and Jouppi]
    - capacitance and resistance values from technology files
    - run time statistics for hit/miss and read/write counts
  - RAM [Itoh et al.] requires technology parameters and netlist
Instruction-level power analysis [Tiwari et al., Wan]
- measure energy consumption of each assembly instruction
- measure energy consumption of non-ideal execution (e.g., stall)
- off-line analysis of assembly code gives total energy consumed
- average power measured for instructions on StrongARM:
  - 200 mW at 170 MHz for most instructions
  - 260 mW at 170 MHz for loads and stores
Previous Work (cont.)

- System-level energy simulation [Benini et al.]
  - system described as a state machine
  - each component has multiple power and performance states
  - very high level

- System-level SOC energy simulation [Li et al., Kapoor] for processor, cache and memory
  - each component analyzed separately
  - technology information needed for modeling
Solution

- Extend instruction-level simulator with energy models for all design components
  - component energy models are based on the data sheets
  - components are analyzed dynamically on cycle-accurate basis
    - very fast
    - many software and hardware architectures can be easily evaluated
  - plots of energy consumption vs. time are available
    - peak energy consumption analysis
    - detailed algorithm analysis
System Model

- Processor
  - CPU
  - L1 cache
    - I-cache
    - D-cache
  - L2 cache
- RAM
- Flash
- Burst RAM
- DC-DC Converter
- Battery

Connections:
- Processor Supply
- Memory Supply
- Data flow
- Address flow
System Simulation Setup

Original Instruction-Level Simulator

1. Write a functional model for each design component (eg. memory)
2. Annotate the functional model for each design component with cycle-accurate timing (eg. mem. access time)
3. Cross-compile and load application software (eg. MPEG decode in C)
4. Obtain simulation output (eg. time)

Enhanced Instruction-Level Simulation of Energy Consumption

1. Write a functional model for each design component (eg. memory)
2. Annotate the functional model for each design component with cycle-accurate timing and energy (eg. mem. access time)
3. Cross-compile and load application software (eg. MPEG decode in C)
4. Obtain simulation output (eg. energy)
Simulator Architecture

\[ E_{\text{Cycle}} = E_{\text{CPU}} + E_{\text{Memory}} + E_{L2\text{Cache}} + E_{\text{Interconnect}} + E_{\text{DC/DC}} \]
Processor Energy Model

- Modeled ARM processor family
  - any processor family can be modeled with appropriate instruction-level simulator
- Estimated active and NOP processor energy per cycle from the values given in the data sheets:

\[
\begin{align*}
E_{\text{CPU, Active}} &= C_{\text{CPU, Active}} V_{cc}^2 \\
C_{\text{CPU, Active}} &= \frac{P_{\text{Active}}}{V_m^2 f_m} \\
E_{\text{CPU, NOP}} &= C_{\text{CPU, NOP}} V_{cc}^2 \\
C_{\text{CPU, NOP}} &= \frac{P_{\text{NOP}}}{V_m^2 f_m}
\end{align*}
\]
Memory Energy Model

- Model burst or normal RAM, FLASH and L2 Cache
- Number of wait cycles estimated by:
  \[ N_{\text{wait}} = \frac{T_{\text{mem}}}{T_{\text{CPU}}} \]
- Estimated active and idle memory energy per cycle from the values given in the data sheets:

**Active Energy**

\[ E_{\text{Mem,Active}} = \frac{C_{\text{Mem,Active}} V_{cc}^2}{N_{\text{Wait}} + 1} \]

\[ C_{\text{Mem,Active}} = \frac{P_{\text{Mem,Active}}}{V_{m}^2 f_{m}} \]

**Idle Energy**

\[ E_{\text{Mem,Idle}} = T_{\text{Cycle}} \sum_{\text{idle}=0}^{n} P_{\text{idle}} \rho_{\text{idle}} \]
Interconnect and Pins Energy Model

- Estimated total switched capacitance from the number of lines switching, interconnect cross-section, line length and pin capacitance

\[
C_{\text{Line}} = \begin{cases} 
L_{\text{Stripline}} C_{\text{Stripline}} \\
L_{\text{Microstrip}} C_{\text{Microstrip}}
\end{cases}
\]

\[
C_{\text{Switch}} = \sum_{\text{Switch} = 0}^{n} (C_{\text{Line}} + C_{\text{Pins}})
\]

- Total energy per cycle depends on the switched capacitance, frequency of access and voltage swing:

\[
E_{\text{Interconnect,Active}} = \frac{C_{\text{Switch}} V_{cc}^2}{N_{\text{Wait}} + 1}
\]
Total energy per cycle is the difference between the energy supplied by the DC/DC converter to the portable system and energy supplied by the battery.

\[
E_{\text{DC/DC}} = I_{\text{Bat}} V_{\text{Bat}} T_{\text{Cycle}} - E_{\text{DC/DOut}}
\]

\[
I_{\text{Bat}} = \frac{I_{\text{DC/DOut}}}{\text{Efficiency}}
\]
Validation of Simulation Methodology

Dhrystone benchmark energy simulations and hardware measurements on SmartBadge are within 5% tolerance.
### Hardware Configurations

<table>
<thead>
<tr>
<th>Name</th>
<th>Instruction Memory</th>
<th>Data Memory</th>
<th>L2 Cache Present</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>FLASH</td>
<td>SRAM</td>
<td>no</td>
</tr>
<tr>
<td>L2 Cache</td>
<td>FLASH</td>
<td>BSRAM</td>
<td>yes</td>
</tr>
<tr>
<td>Burst SRAM</td>
<td>BFLASH</td>
<td>BSRAM</td>
<td>no</td>
</tr>
<tr>
<td>Burst SDRAM</td>
<td>BFLASH</td>
<td>BSRAM</td>
<td>no</td>
</tr>
</tbody>
</table>

### Memory Architectures

<table>
<thead>
<tr>
<th>Name</th>
<th>Units</th>
<th>Initial Access (ns)</th>
<th>Burst Access (ns)</th>
<th>Active Power (mW)</th>
<th>Idle Power (mW)</th>
<th>Interconnect Capacitance (pF/line)</th>
<th>I/O Pin Capacitance (pF/pin)</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLASH</td>
<td>80</td>
<td>N/A</td>
<td>75</td>
<td>0.5</td>
<td>4.8</td>
<td>10</td>
<td>Intel</td>
<td></td>
</tr>
<tr>
<td>BFLASH</td>
<td>80</td>
<td>40.00</td>
<td>600</td>
<td>2.5</td>
<td>4.8</td>
<td>10</td>
<td>TI</td>
<td></td>
</tr>
<tr>
<td>SRAM</td>
<td>90</td>
<td>N/A</td>
<td>185</td>
<td>0.1</td>
<td>8</td>
<td>8</td>
<td>Toshiba</td>
<td></td>
</tr>
<tr>
<td>BSRAM</td>
<td>90</td>
<td>45.00</td>
<td>365</td>
<td>1.7</td>
<td>8</td>
<td>8</td>
<td>Micron</td>
<td></td>
</tr>
<tr>
<td>BSDRAM</td>
<td>30</td>
<td>15.00</td>
<td>430</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>Micron</td>
<td></td>
</tr>
<tr>
<td>L2 Cache</td>
<td>20.00</td>
<td>10</td>
<td>1985</td>
<td>330</td>
<td>3.2</td>
<td>5</td>
<td>Motorola</td>
<td></td>
</tr>
</tbody>
</table>

- ARM processor with 64KB L1 cache running at 200 MHz, 400 mW active and 170 mW idle power consumption.
Hardware Design Exploration Results

- Data memory speed limits energy and performance efficiency, but instruction memory speed is not a limitation.
- The most energy and performance efficient design uses fast and power hungry burst SDRAM.
- L2 cache is neither energy nor performance efficient.

Energy Consumption

- Data memory speed limits energy and performance efficiency, but instruction memory speed is not a limitation.
- The most energy and performance efficient design uses fast and power hungry burst SDRAM.
- L2 cache is neither energy nor performance efficient.
MPEG Decode Software Design Exploration

- MPEG input data format
  - I-frame
    - jpeg encoded frame
  - P-frame
    - differences between the current and the previous frame
  - B-frame
    - differences between the current and both the previous and the future frames
  - combinations of I,P,B frames in a group of pictures (GOP)

- Decoding speed
  - faster decoding speed means less data

### Software Configurations

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Speed (frames/s)</th>
<th>I-frames (number)</th>
<th>P-frames (number)</th>
<th>B-frames (number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPBB @</td>
<td>30</td>
<td>2</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>IP @</td>
<td>30</td>
<td>2</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>IPPI @</td>
<td>30</td>
<td>4</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>IPPI @</td>
<td>25</td>
<td>4</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>
Software Design Exploration Results

- Combination of I and P-frames performs best for energy consumption and execution time
- B-frame decoding is not energy or time efficient
- Faster decoding speed gives energy savings

Energy Consumption and Execution Time

![Graph showing energy consumption and execution time for different frame combinations]
Peak energy consumption can be more than two times larger than the average so DC-DC converter, battery and thermal design have to be specified accordingly.
Conclusions and Future Work

- A methodology for cycle-accurate energy consumption simulation of discrete component designs has been presented.
- Designer’s extensions to the instruction-level simulator are minimal.
- Simulation is within 5% accuracy of the hardware measurements.
- Design exploration of both hardware and software architectures:
  - MPEG decode design exploration example.
- Plots of energy consumption over time are available.
- Future extensions:
  - Model of wireless link input.
  - Model the video output.