
Peer reviewed version

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research
General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
http://www.bristol.ac.uk/pure/about/ebr-terms
Implementing the model: Analysis

### Research question
- Can software activity be correlated with hardware energy consumption?

### State of the art
- Software energy modelling: ISA level\(^1\), device blocks\(^2\), library level\(^3\).

### Solution

#### Measurement HW/SW framework: XMProfile

Combination of XMOS hardware\(^4\), current sensor hardware and a custom software framework.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1V power supply</td>
<td>Volatile current sense&lt;br&gt;Sampling inertia: 2.19 μs&lt;br&gt;Voltage range: 1V</td>
</tr>
<tr>
<td>INA 219</td>
<td>Non-invasive current sense&lt;br&gt;Sampling inertia: 2.19 μs&lt;br&gt;Voltage range: 850mV</td>
</tr>
<tr>
<td>Control and Sync</td>
<td>XP20 parallel clock and asynchronous interconnect network</td>
</tr>
<tr>
<td>Test processor</td>
<td>Single instruction execution time: 78 ns&lt;br&gt;16KB L1 cache&lt;br&gt;64KB L2 cache&lt;br&gt;1Mbyte L3 cache</td>
</tr>
<tr>
<td>Test info, power data</td>
<td>Test vectors generated, power data measured</td>
</tr>
<tr>
<td>Host PC datastore</td>
<td>Data acquisition and analysis&lt;br&gt;Power consumption statistics calculated</td>
</tr>
<tr>
<td>XMProfile control SW</td>
<td>Custom software framework for data collection and analysis</td>
</tr>
</tbody>
</table>

#### Data collection: Test construction

**Tiwari method\(^5\):**

- Instruction overhead smaller than data overhead.
- Use instruction execution statistics, rather than trace, for speed: ~16x faster.
- Implement model per-instruction (standard) and by operand count (grouped).
- Consider concurrency levels (number of active threads).

**XS1-L multi-threaded pipeline**

<table>
<thead>
<tr>
<th>Step</th>
<th>1 thread</th>
<th>2 threads</th>
<th>3 threads</th>
<th>4 threads</th>
<th>5+ threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T0</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>T3</td>
<td>T4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>T0</td>
<td>T0</td>
<td>T0</td>
<td>T0</td>
<td>T0</td>
</tr>
</tbody>
</table>

#### Considerations for XS1-L\(^6\):
- Thread count.
- Idle periods (event waiting).
- Instruction overhead is between threads.
- Trace simulation is slow.

**Solution:**
- Tightly-coupled threads.
- Odd/even threads used to measure instruction overheads.
- Thread count (0 – 8) used to establish thread and idle costs.

**Additional XMProfile features:**
- Constrained random number generation.
- Auto-generate large sets of test loops (ALU).
- Minimise loop headtail overhead.

### Implementing the model: Analysis

**Multi-threaded model using simulation statistics:**

\[
E_p = P_{\text{burst}} N_{\text{burst}} + \sum_{i=0}^{N-1} \sum_{j=0}^{m} \left((M_i R_i O_i + P_{\text{burst}}) N_{i,j} T_{\text{calc}}\right)
\]

- Instruction overhead smaller than data overhead.
- Use instruction execution statistics, rather than trace, for speed: ~16x faster.
- Implement model per-instruction (standard) and by operand count (grouped).
- Consider concurrency levels (number of active threads).

### Test and evaluation

**A set of benchmarks – traditional and custom – were used to test model performance with various levels of concurrency.**

**Benchmarks were run through the XMProfile framework to acquire real device energy measurements.**

- Worst case error: 16% standard model, 26% grouped model.
- Average error: 7% standard model, 16% grouped model.
- ISA simulation\(^7\) runs ~100x slower than real time, statistics processing time is negligible.

### Continuing and future work

- Complete ISA model based on established base facts and more complex test kernels to improve accuracy.
- Swallow project: Many-core XS1 system grid (100s of cores). Incorporate commns costs into model.
- Contributing to ENTRA (ENergy TRANSParency) EU FP7 project.
- Static analysis of compiled code rather than simulation.
- Use model for design space exploration & guided optimisation (tool assisted & fully automated).