The challenges of modelling and optimising energy for multithreaded programs

Creating energy efficient software for multi-threaded, multi-core processors

Steve Kerrison
μ Research Group
University of Bristol
steve.kerrison@bris.ac.uk
Contents

- The problem
  - XMOS XS1 architecture
  - Multi-threading breaks existing models
  - Fixing the model
  - New tricks
  - Unanswered questions
The problem

- Hardware power figures
  - Maximum? Minimum? Typical?
- What will that translate to for my application?
- How can the compiler help me?
The problem

- Furnish the workflow with this information
  - The developer
  - The compiler
- Packaged in a fast, safe, easy to support way
Traditional methods

- Map processor activity onto some energy metric.
- Requires some cost of executing an instruction.
- **Plus** the cost of switching data & switching between instructions.

\[ E = \sum T \left( P_i N_i + P_{i,j} N_{i,j} \right) \]
Traditional methods

- What are we trying to save?

\[ P_{\text{static}} = V_{\text{core}} I_{\text{leak}} \]

\[ P_{\text{dynamic}} = A \mu C_{\text{sw}} V_{\text{core}}^2 F \]

\[ E = T\left( P_{\text{static}} + P_{\text{dynamic}} \right) \]

- Turn off
- Switch less
- Lower frequency/Voltage
- Take less time
Contents

- The problem
- XMOS XS1 architecture
  - Multi-threading breaks existing models
  - Fixing the model
  - New tricks
- Unanswered questions
- 32-bit micro-processor for embedded systems.
- ISA incorporates I/O instructions
- Event driven, real-time
- Message passing
- 64KB fast SRAM per core
- Deterministic WCET analysis
- Hardware multi-threaded pipeline
- Up to eight threads per core
- Four stage pipeline
- Simple scheduling
- At 500MHz, 125MIPS per thread for $\leq 4$ threads
...and others

Consider also...

- SMT “hyper-threaded” architectures
- Super-scalar CPUs
- Out-of-order execution

All affect switching through the CPU
Contents

- The problem
- XMOS XS₁ architecture
  - Multi-threading breaks existing models
- Fixing the model
- New tricks
- Unanswered questions
Breaking the model

- ISA-level example
- Model four threads
  - Get power figures
  - Determine energy consumption
- Run at architectural level
  - Interleaving in pipeline changes switching behaviour
  - For better or worse?
  - Regardless, less accurate!

\[ E = a \]
\[ E = b \]
\[ E = c \]
\[ E = d \]

\[ E \neq a + b + c + d \]
\[ E = ? \]
Data-path context switch example

Thread n registers

0xFFFFF0000
0x00AAAAAA
0x0000FFFF
0x55555500

Thread n+1 registers

0x0000FFFF
0x00AAAAAA
0x55555500

0xFFFFF0000 -> 0x0000FFFF
0x0000FFFF -> 0x55555500
0x00AAAAAA -> 0x55555500
Contents

- Introduction
- The problem
- XMOS XS1 architecture
- Multi-threading breaks existing models
- Fixing the model
- New tricks
- Unanswered questions
Fixing the model

- Exactly how unpredictable is the pipeline?
- Independent unsynchronised threads – very unpredictable
- Interacting threads – more predictable
  - Known points of synchronisation
    - Explicit sync
    - Communication
    - Events
Fixing the model

- Thread interaction example
  - I/O & protocols govern activity
  - More detail gives a higher accuracy model.
  - The more we can predict, the more energy we can save.
Fixing the model

- Incorporate thread sync/communication.
- Use protocol timing to inform thread schedule.
- Find sections of threads that execute together.
- Identify optimisation candidates.
The problem

XMOS XS1 architecture

Multi-threading breaks existing models

Fixing the model

New tricks

Unanswered questions
Trick 1: Migrate

- Assign less demanding tasks to low frequency cores.
- Move tasks that interfere with other optimisation efforts onto a different core.
- We have to deal with the timing implications of doing this.
Trick 2: Data width

- Example: 32-bit XCore hardware.
- The full width of the data path is always active.
  - No architectural specialisation for narrower data.
- Let’s compare the switching with 32-bit and 16-bit values.
**Trick 2: Data width**

- **10-25% improvement for some instruction pairings**

Processed/2011-04-12/16.c.dat.png

Processed/2011-04-12/32.c.dat.png
Trick 2: Data width

- Per-bit entropy sampled over the **lifetime** of a register.

![Bar chart](image)
Trick 2: Data width

- Per-bit entropy sampled over the lifetime of a register.
Trick 2.5: Precision

- Do we really need all 12-bits of that ADC sample?
- If we reduce precision of some data, what will the cumulative error be?
- How do we choose an appropriate trade-off between error, accuracy and energy?
Trick 3: Format

- Think of a sequence of audio samples.
- How often do we cross zero?
- What happens in twos complement when we do this?
  - 10 -> -10
  - 00001010 -> 11110110
  - 7 of 8 bits switch!
Trick 3: Format

- Think of a sequence of audio samples.
- How often do we cross zero?
- What happens in twos complement when we do this?
  - 10 -> -10
  - 00001010 -> 11110110
  - 7 of 8 bits switch!

- So what about sign-magnitude?
  - 10 -> -10
  - 00001010 -> 10001010
  - 1 bit switch!
  - 5-15% reduction seen by changing format.

- Unfortunately:
  - Redundant zero representation
  - No hardware support.
The problem

XMOS XS1 architecture

Multi-threading breaks existing models

Fixing the model

New tricks

Unanswered questions
Unanswered questions

- How do we knit together timing data, thread schedules and the energy cost of code segments?

- To what extent can inter-thread optimisation be performed?
  - How great are the benefits and when do they become worth the effort?
Unanswered questions

- What’s the best way to present energy data to the developer?
- How do we minimise the amount of additional annotation required?
  - We want to improve workflow, not complicate it
Thank you
Questions welcomed...

More information

XMOS: http://www.xmos.com

UoB Micro research group: http://www.cs.bris.ac.uk/Research/Micro

Me: steve.kerrison@bris.ac.uk