University of BRISTOL

XMOS

The challenges of modelling energy for multithreaded programs

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

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

Introduction

- Steve Kerrison
  - Micro Research Group, UoB
    - Electronics & Electrical Engineering
    - Computer Science
    - HPC, architectures, energy, verification
- Working with XMOS over the past year
  - Bristol-based semiconductor company
  - Multi-threaded, event driven, embedded processors
- **Exploring problems faced by energy-aware embedded software developers**

Contents

University of BRISTOL

XMOS

- Introduction
- **The problem**
- Traditional methods
- XMOS XS1 architecture
- Multi-threading breaks the model
- What can we do about it?
- Unanswered questions

The problem

University of BRISTOL

XMOS

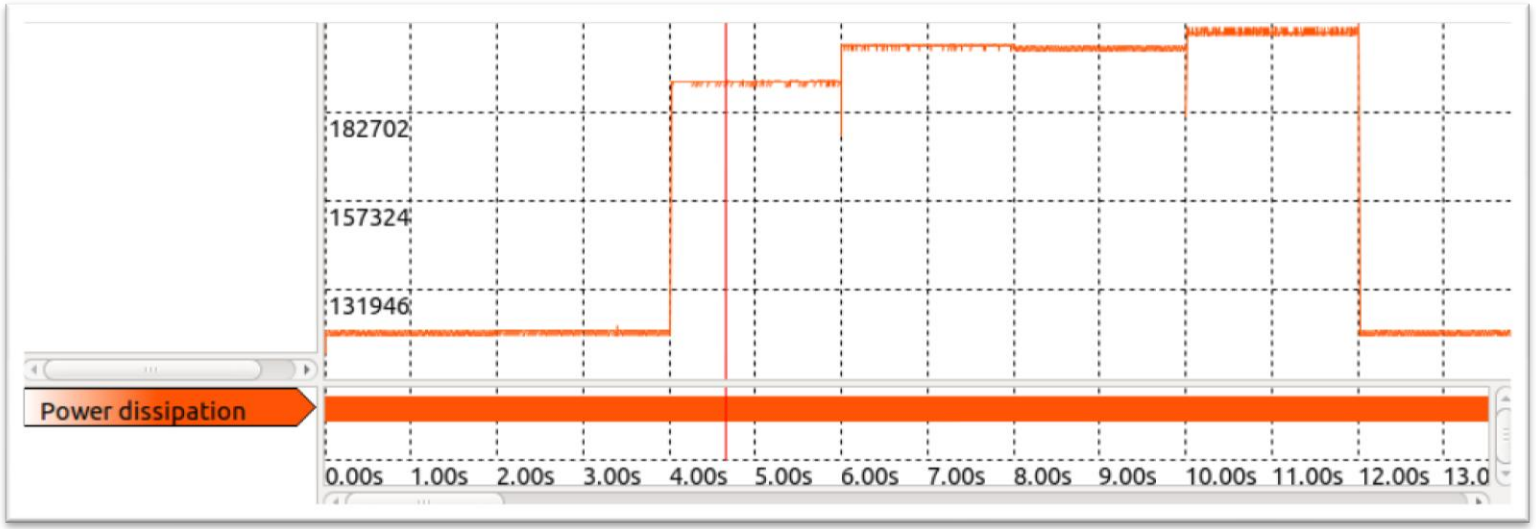
- Hardware power figures
  - Maximum? Minimum? Typical?
- What will that translate to **for my application?**
- **Where** is the energy being consumed?
- **What can I do** to improve this?
  - Reduce instantaneous power
  - Improve energy consumption over time
- How can **the compiler** help me?

The problem

- Mapping software onto hardware behaviour
  - The number of parameters grow
- Exposing & understanding HW power characteristics & capabilities
  - What should we expect? What can we influence directly?

The problem

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



Contents

University of BRISTOL

XMOS

- Introduction
- The problem
- **Traditional methods**
- XMOS XS1 architecture
- Multi-threading breaks the model
- What can we do about it?
- Unanswered questions

Traditional methods

- ISA level
  - A mapping of instructions onto power
  - Include switching between instructions
- Functional block level
  - Reduce ISA to set of blocks with defined power behaviour
- Software block level
  - Map function calls, library calls onto power

- All represent static + dynamic power dissipation over time *in some way*

$$P_{static} = V_{core} I_{leak}$$

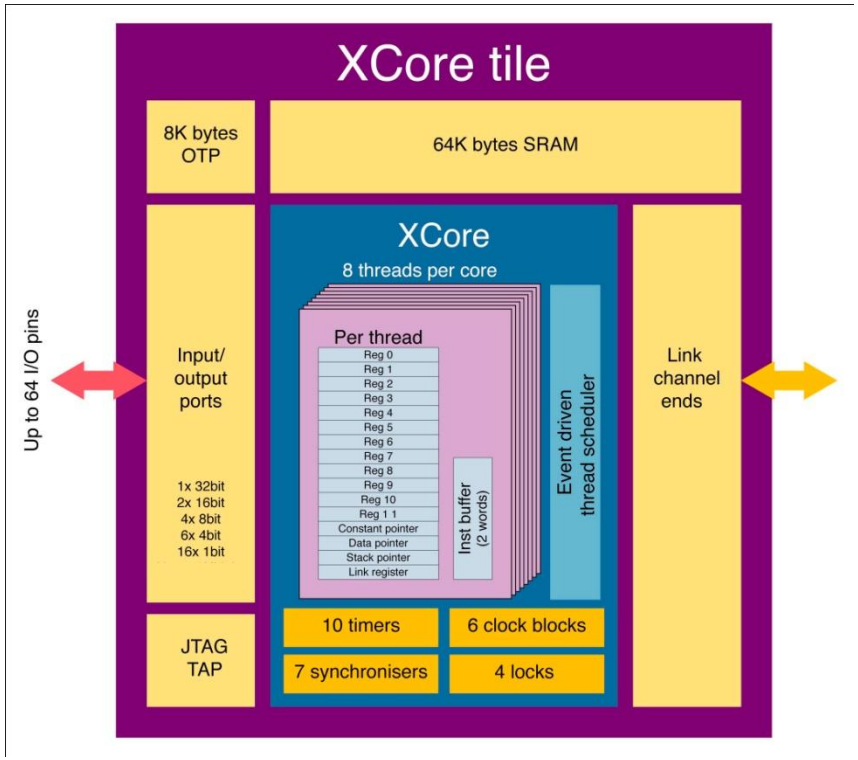
$$P_{dynamic} = A_{\mu} C_{sw} V_{core}^{2} F$$

$$E = T(P_{static} + P_{dynamic})$$

Contents

- Introduction
- The problem
- Traditional methods
- **XMOS XS1 architecture**
- Multi-threading breaks the model
- What can we do about it?
- Unanswered questions

XMOS XS1



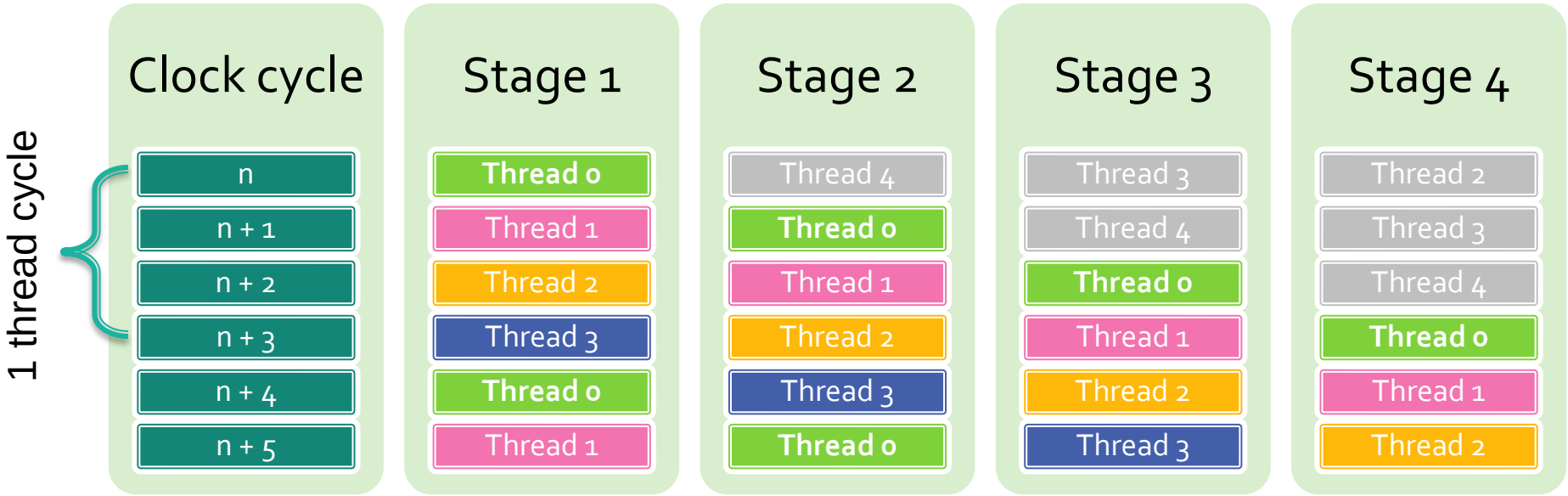
- Microprocessor for embedded designs
- ISA incorporates I/O instructions
- Event driven, real-time
- Channels for routing messages between threads, cores & devices
- 64KB single-cycle SRAM per core
- Deterministic execution with WCET analysis
- **Hardware multi-threaded pipeline**

XMOS XS1

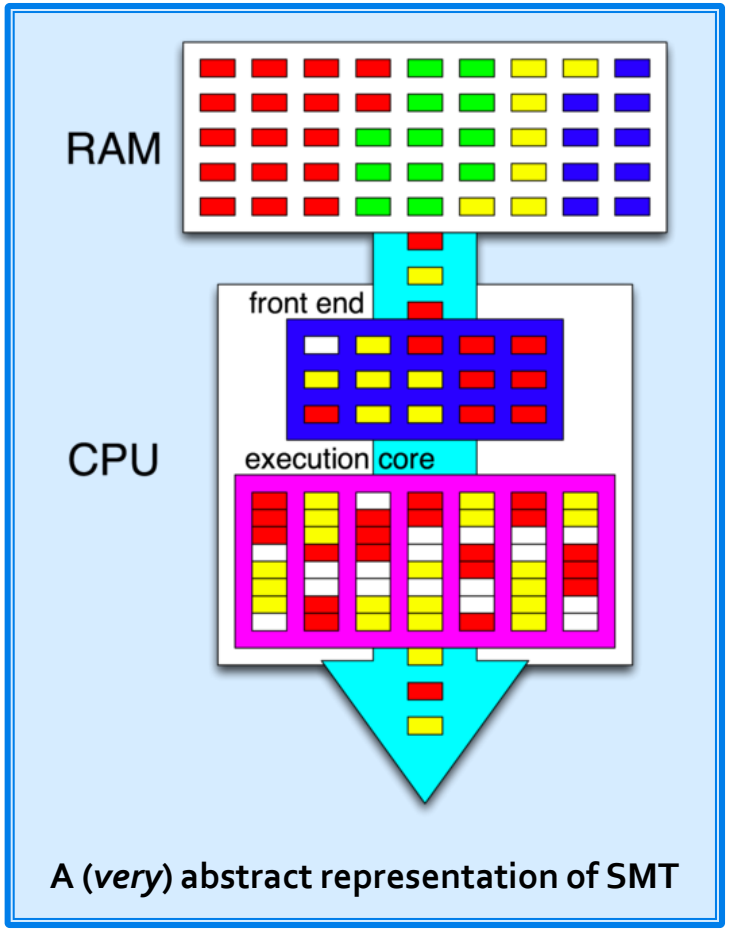
University of BRISTOL

XMOS

- Up to eight threads per core
- Four stage pipeline
- Simple scheduling
- At 500MHz, 125MIPS per thread for <= 4 threads



…and others



A (*very*) abstract representation of SMT

- **Consider also…**
- SMT "hyper-threaded" architectures
- Super-scalar CPUs
- Out-of-order execution
- **All affect switching in data paths of CPU**

Contents

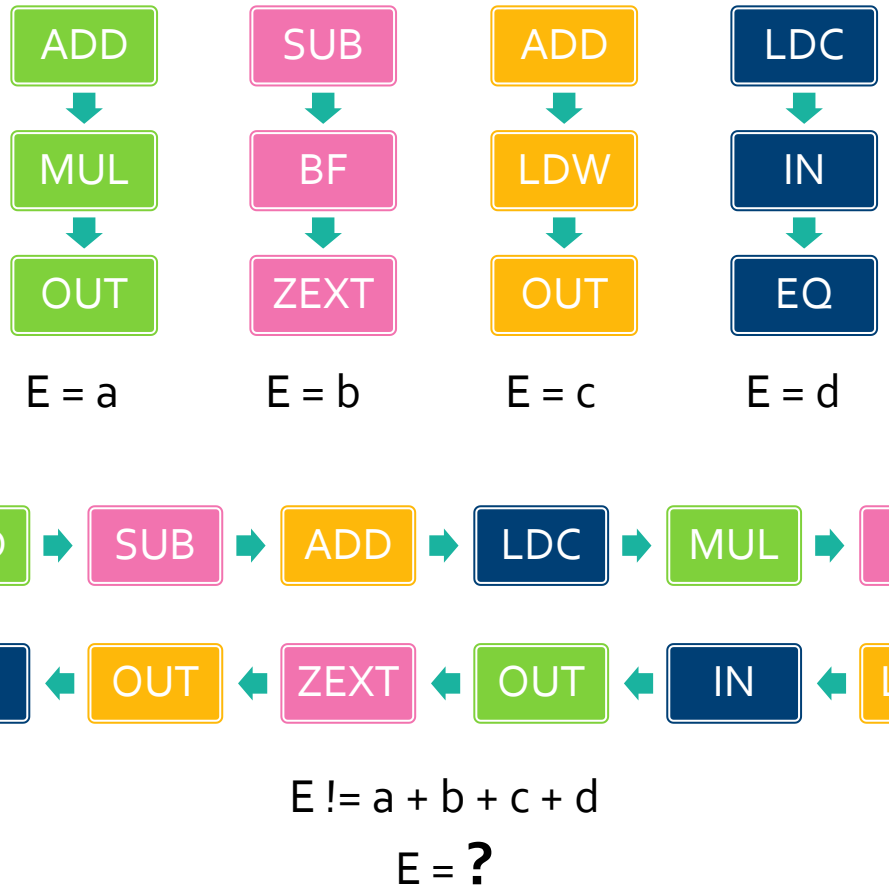
- Introduction
- The problem
- Traditional methods
- XMOS XS1 architecture
- **Multi-threading breaks the model**
- What can we do about it?
- Unanswered questions

Breaking the model

- Energy models and optimisations assume we know what's in the pipeline.
  - Instruction re-ordering
  - Data aware optimisation
- But multiple threads create an unpredictable instruction stream
  - …or do they?

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!**

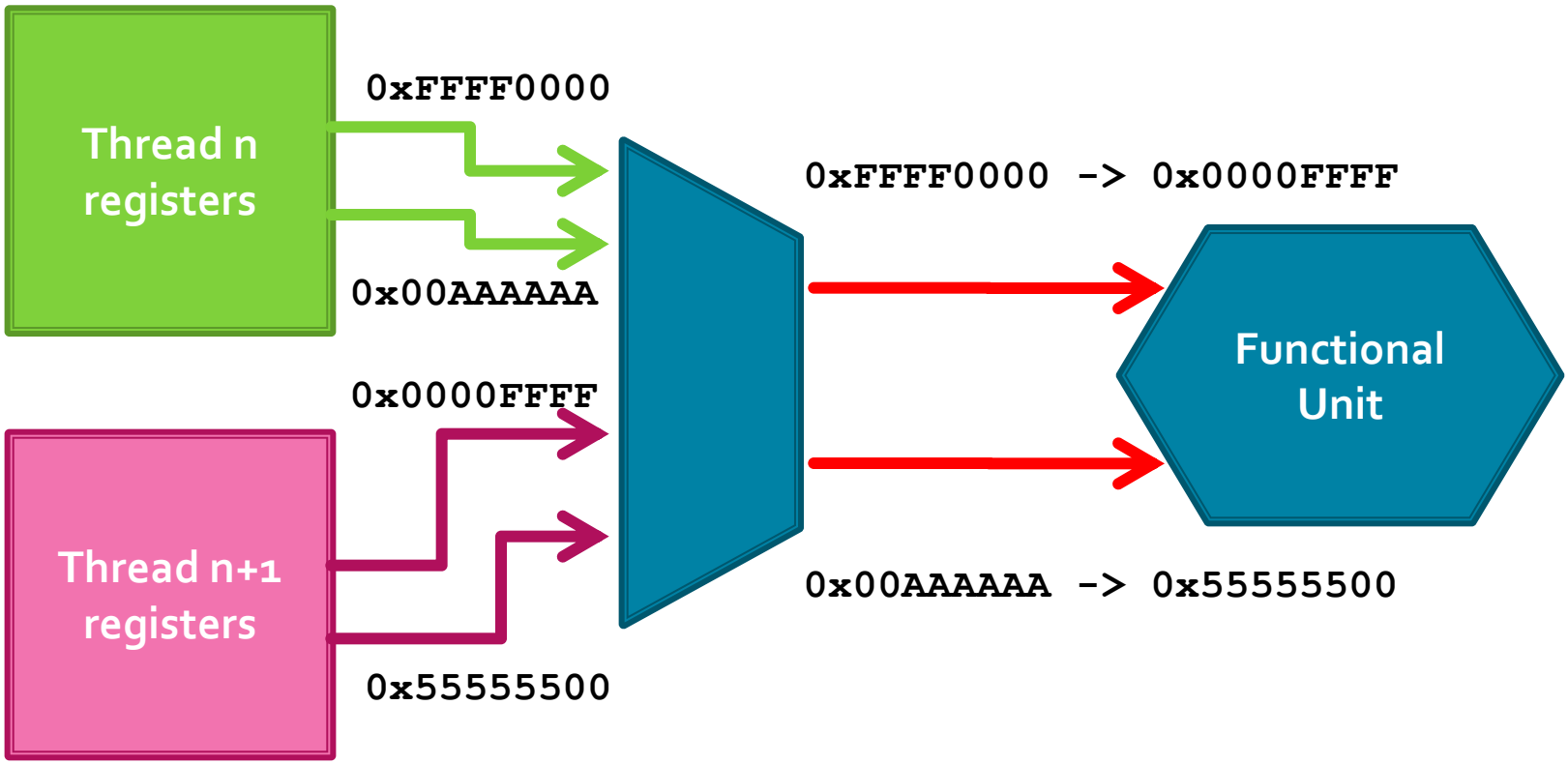


Breaking the model

University of BRISTOL

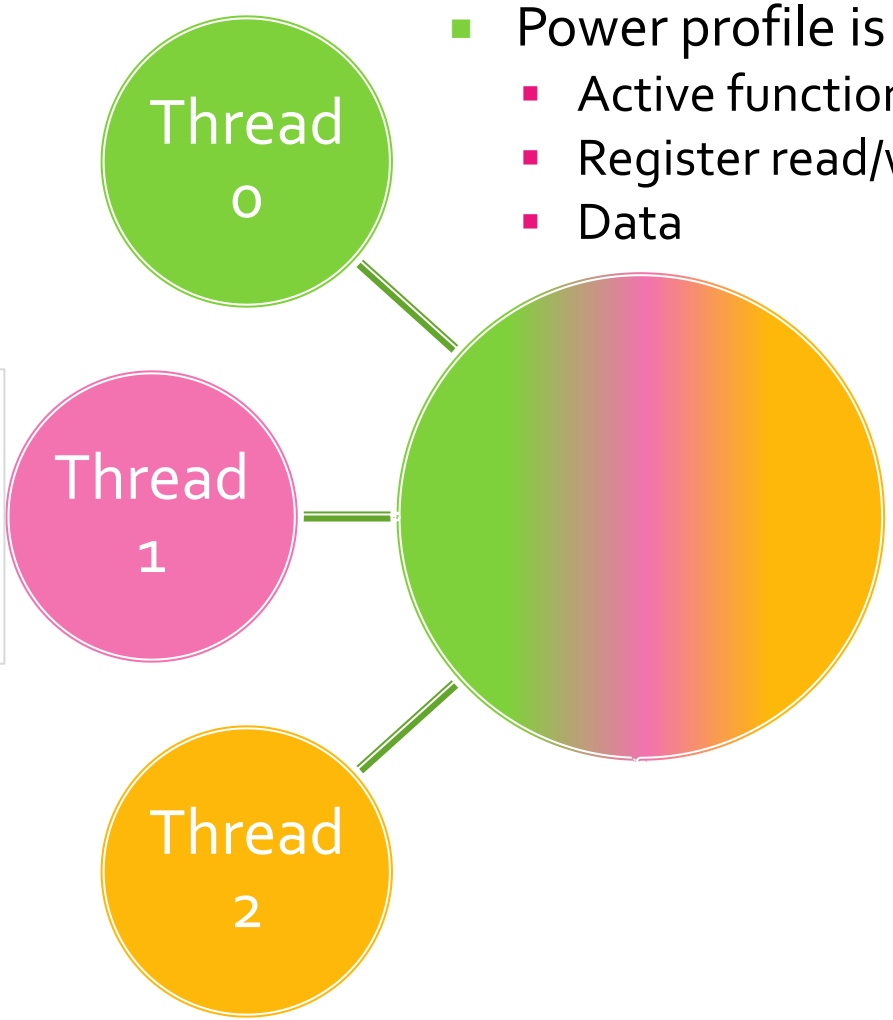
XMOS

Data-path context switch example



Breaking the model

- Power profile is less predictable:
  - Active functional units
  - Register read/write paths
  - Data



- Instruction
  - Functional unit
  - Read ports
  - Write ports
- State
  - Register contents
  - Values read/written

Contents

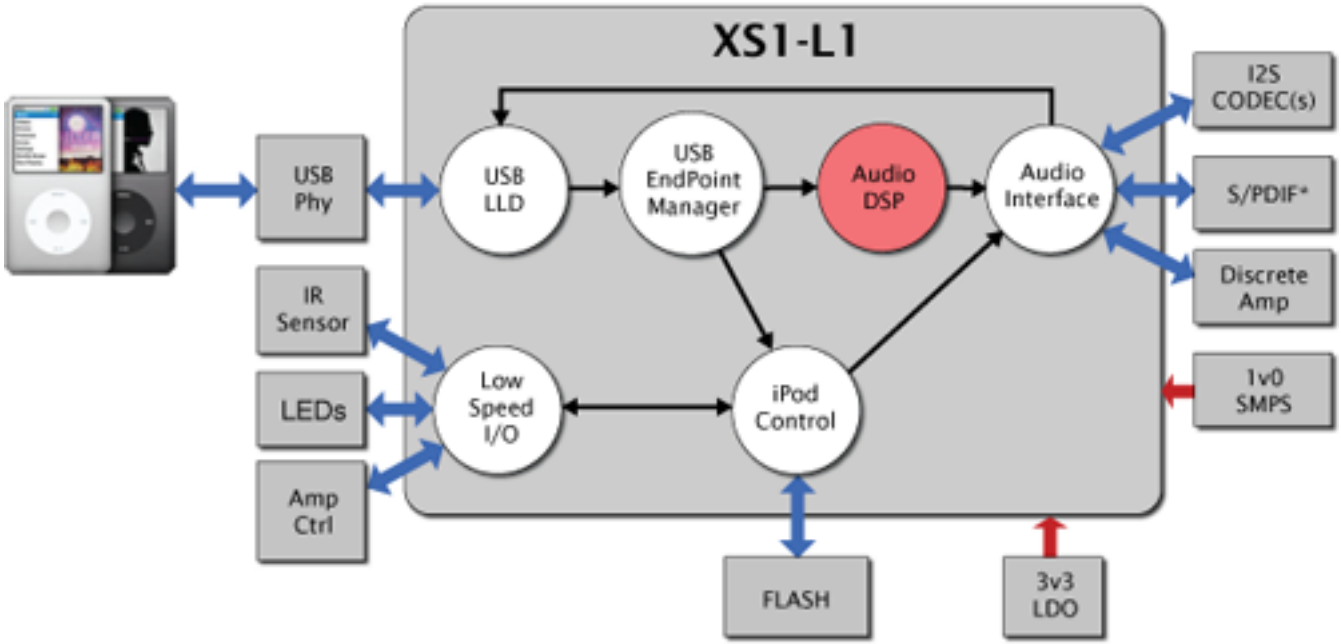
- Introduction
- The problem
- Traditional methods
- XMOS XS1 architecture
- Multi-threading breaks the model
- **What can we do about it?**
- Unanswered questions

Taking action

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

Taking action

- 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**.



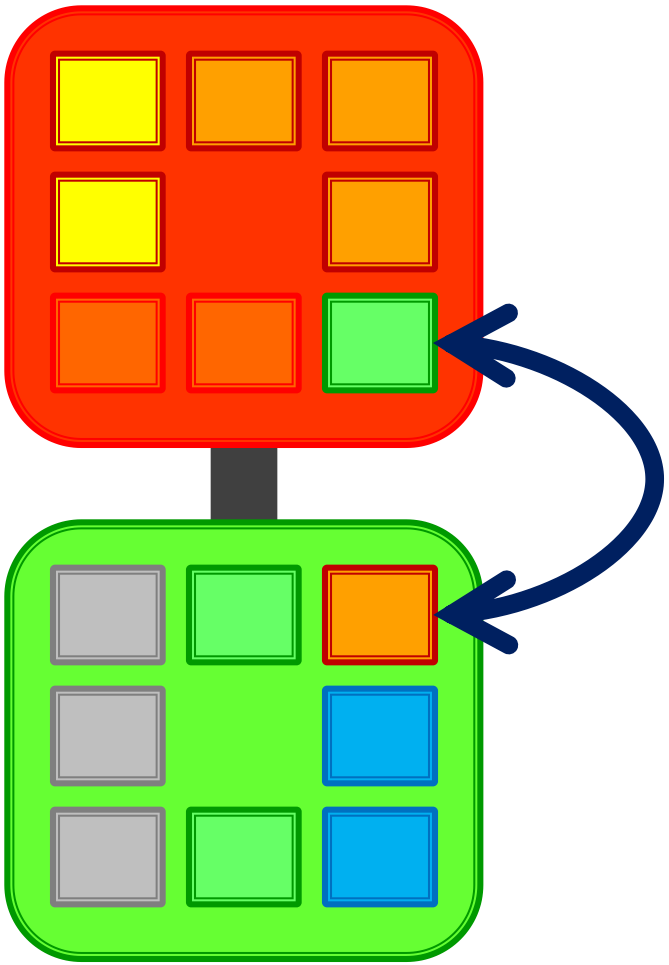
Taking action

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

Taking action

- Possible optimisations:
- Thread-sensitive instruction ordering
  - Group instructions by functional unit
  - …or by data attributes
    - Width
    - Signed/unsigned
- Energy aware thread-ordering
  - Tweak the ordering of threads to our advantage

Taking action



- What about multi-threaded, multi-core?
- Gives us a few more options:
  - 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.

Contents

- Introduction
- The problem
- Traditional methods
- XMOS XS1 architecture
- Multi-threading breaks the model
- What can we do about it?
- **Unanswered questions**

Unanswered questions

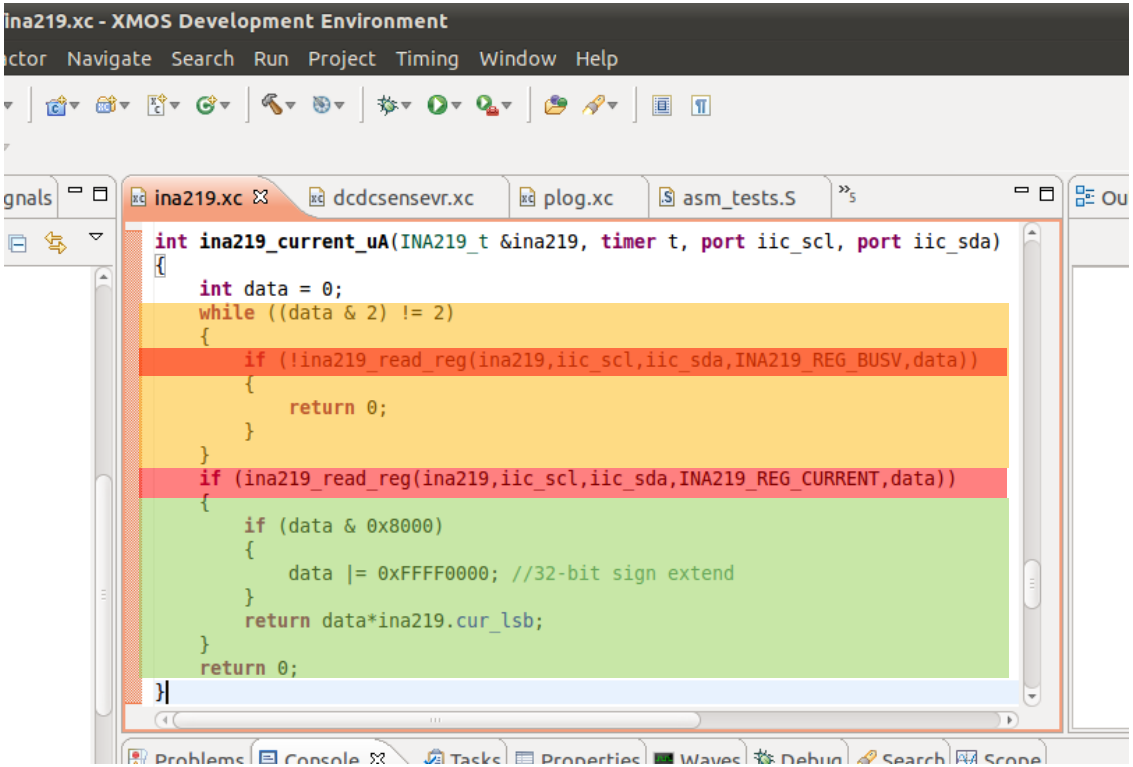
University of BRISTOL

XMOS

- 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