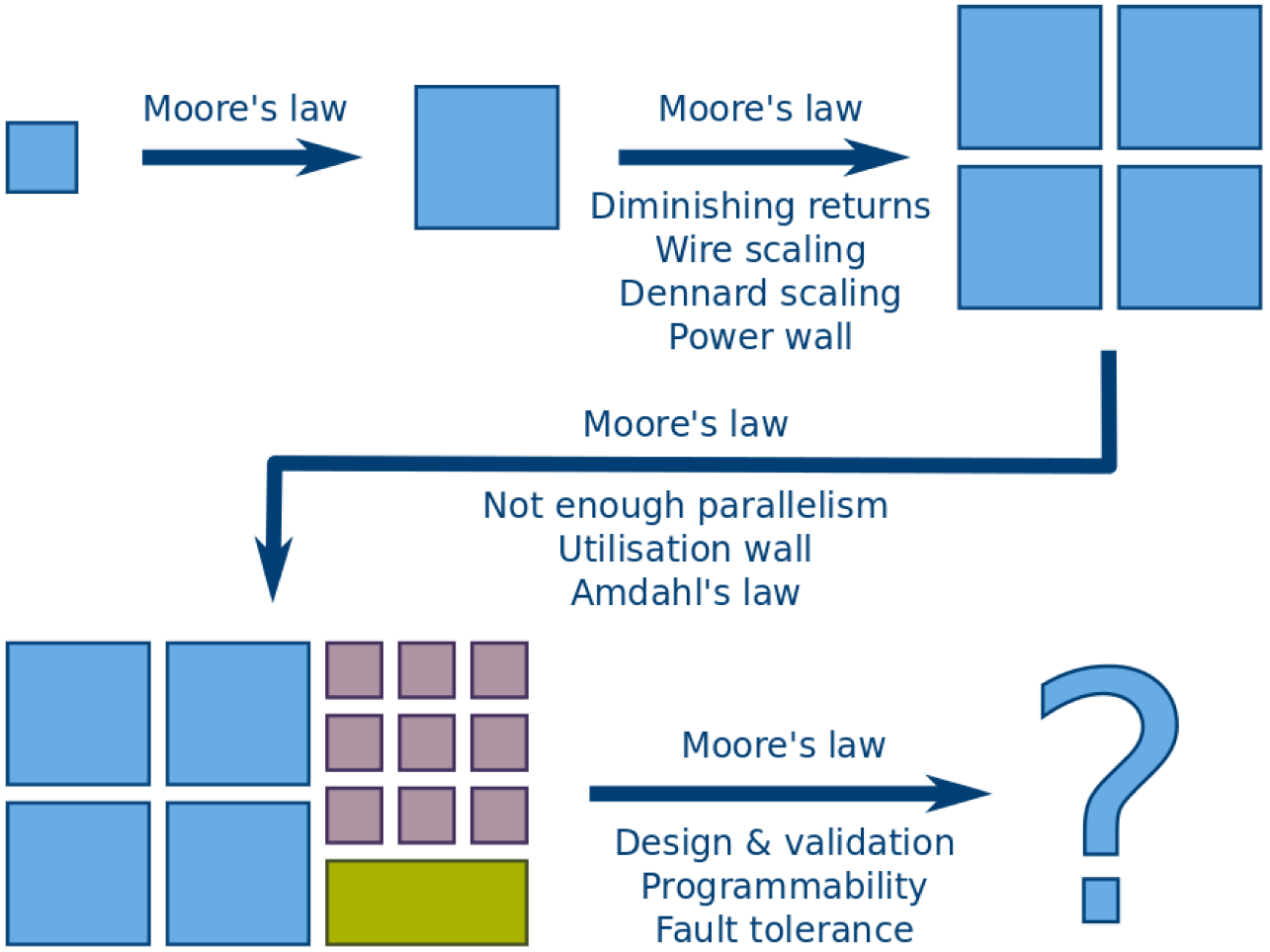
UNIVERSITY OF CAMBRIDGE

Exploiting Tightly-Coupled Cores

Daniel Bates, Alex Bradbury, Andreas Koltes and Robert Mullins

SAMOS XIII

Motivation

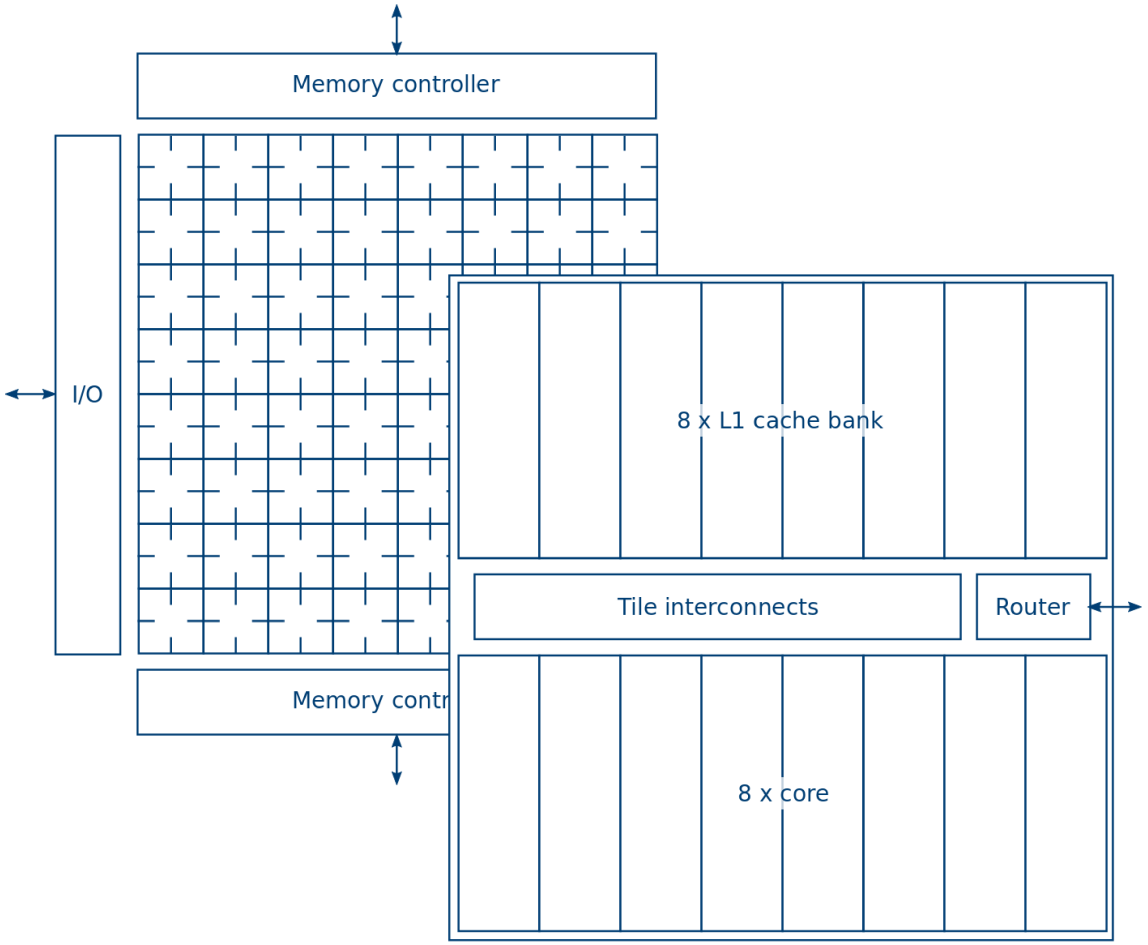

Moore's law
Moore's law
Diminishing returns
Wire scaling
Dennard scaling
Power wall
Moore's law
Not enough parallelism
Utilisation wall
Amdahl's law
Moore's law
Design & validation
Programmability
Fault tolerance

Introduction

- Today's processors have rigid boundaries
- We explore a different approach by exposing:
  - placement of data
  - placement of computation
  - communication
- *Virtual architectures* make a 500-core embedded processor feasible

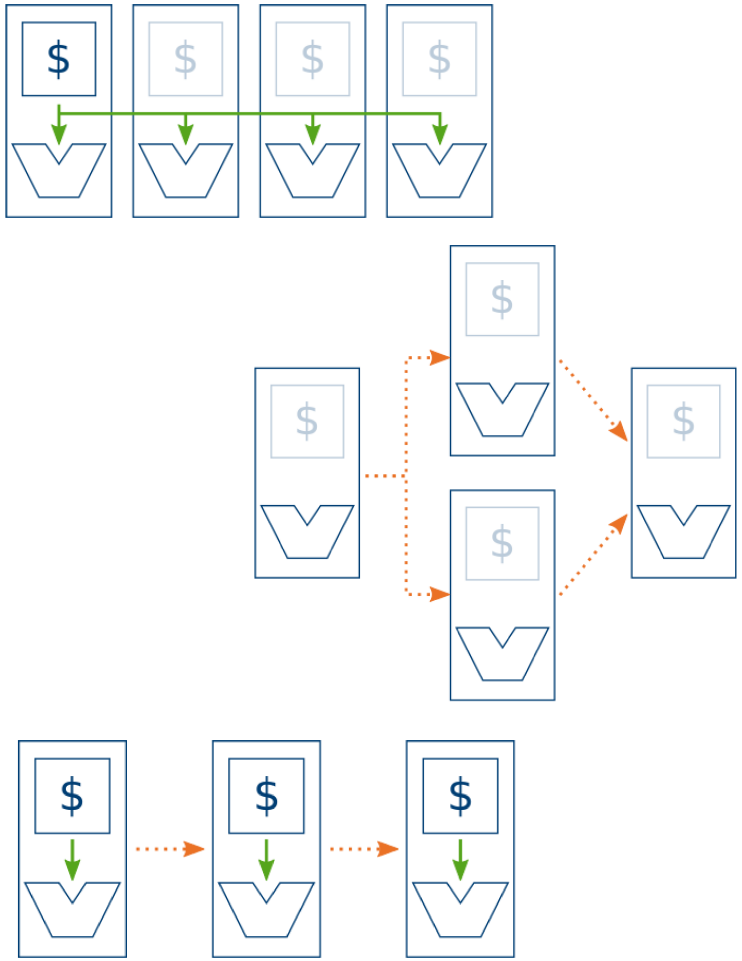
Loki chip

- Tiled, homogeneous design
- Hierarchical network
- Optimised for low latency and low energy communication
- Loki is unique in its flexibility



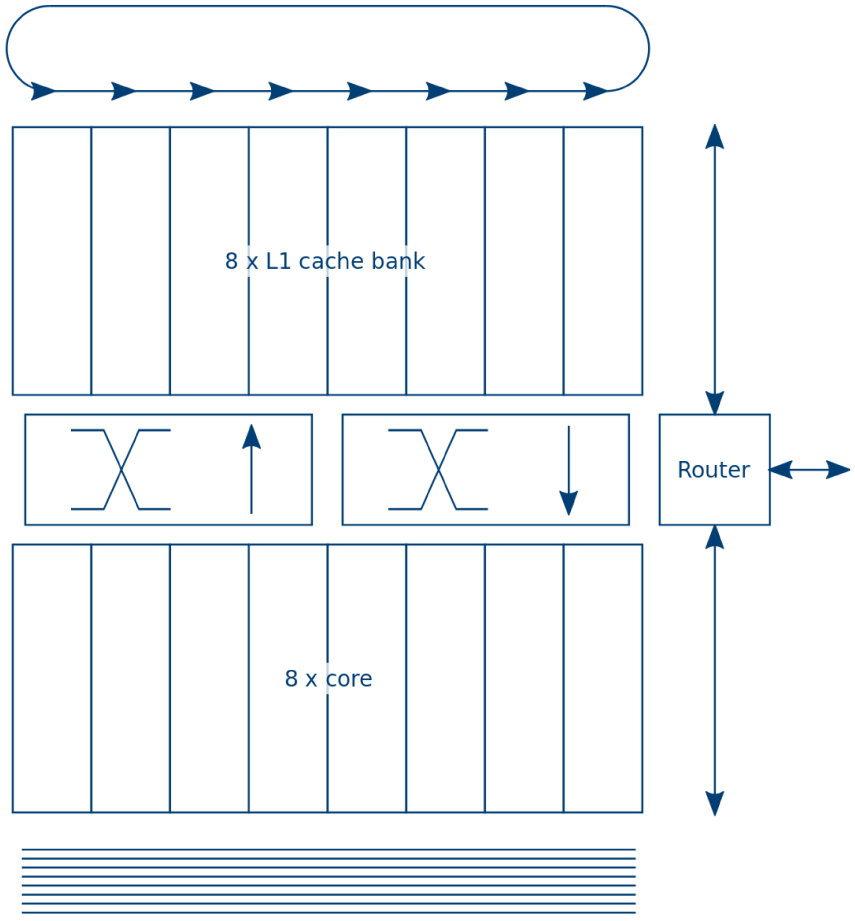
Execution patterns

- Similar to FPGA overlays
- Software specialisation
- Increase performance or reduce energy (or both!)
- Efficient communication is important
- A wide variety of available patterns is helpful



Loki tile

- Network communication is key
- Sub-networks optimised for different use cases
- Even L1 cache access is over the network
- Channel-based communication
- Cache is configurable

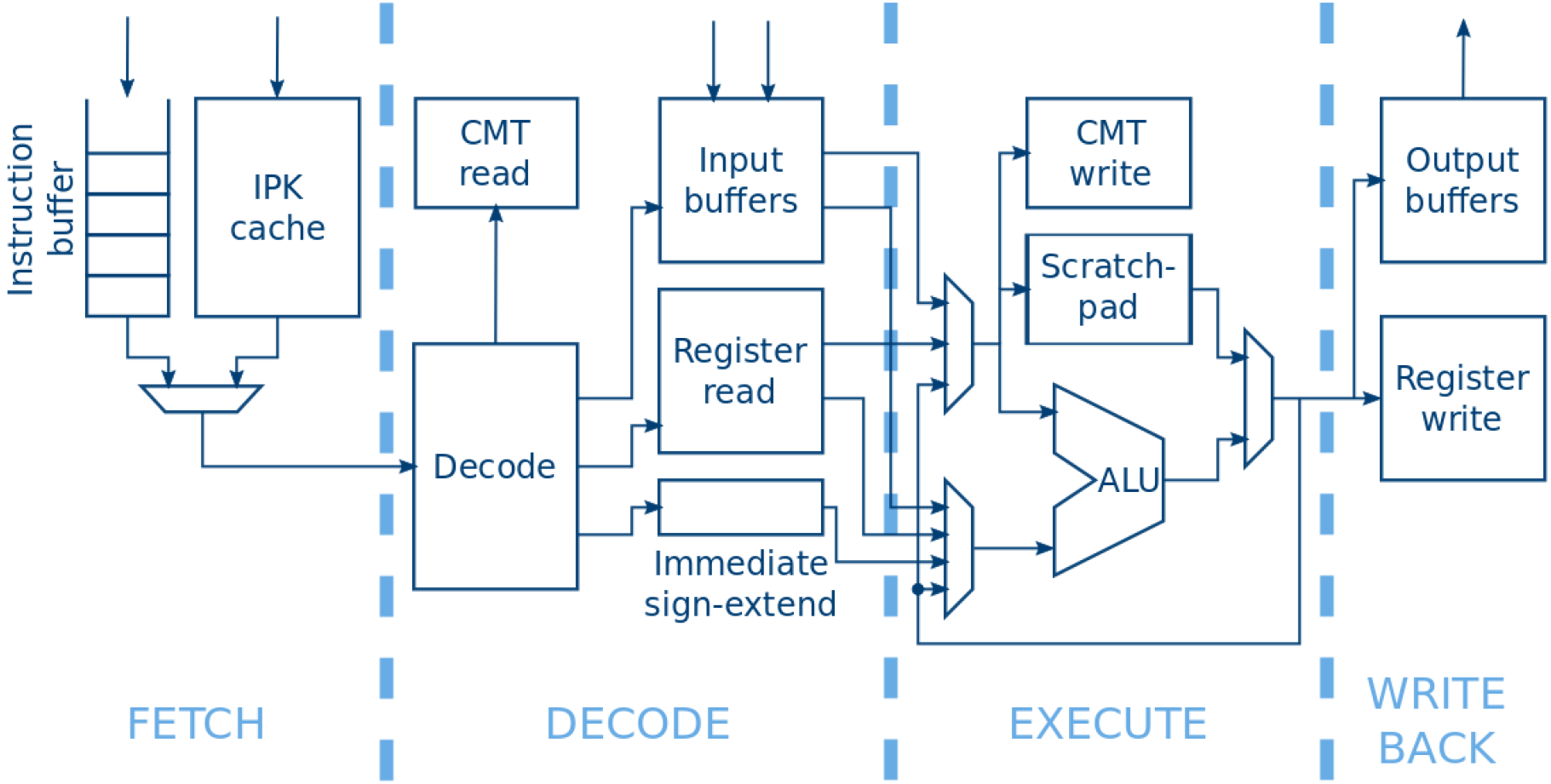


Code example

```
uint32_t updateCRC32(uint8_t ch,
                     uint32_t crc)
{
  return table[(crc ^ ch) & 0xff] ^
         (crc >> 8);
}
```

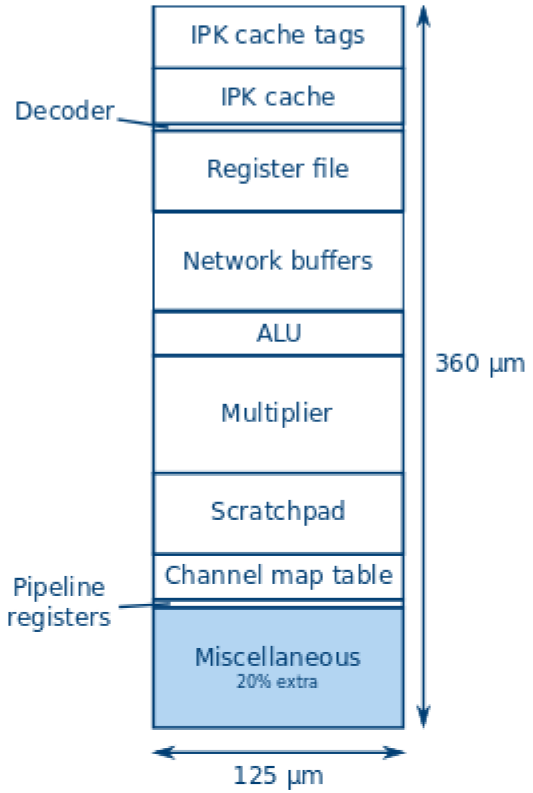
```
setchmapi 1,    r15
[...]
fetch       r10
xor         r11, r14, r13
lli         r12, %lo(table)
lui         r12, %hi(table)
andi        r11, r11, 255
slli        r11, r11, 2
addu        r11, r12, r11
ldw         0(r11) -> 1
srli        r12, r14, 8
xor.eop     r11, r2,  r12
```

Loki pipeline



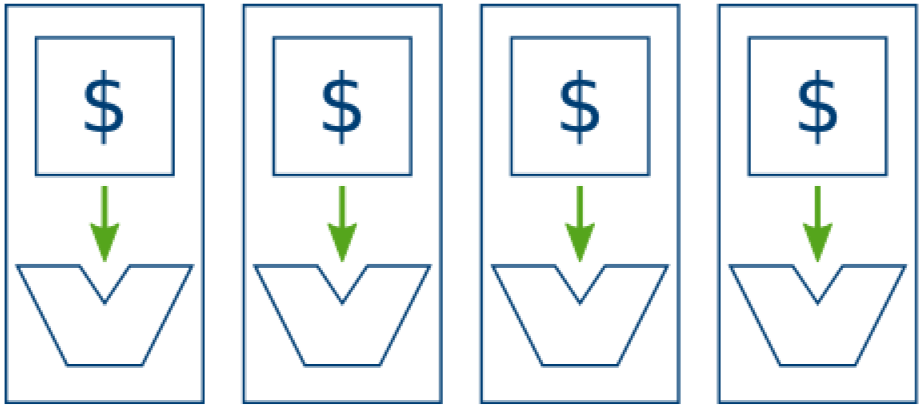
Single core behaviour

- Core area: 125x360µm
- Energy/instruction: 10-20pJ
- Compared to ARM1176 (same process):
  - 1.4-2.2x instruction count
  - 1.0-1.8x execution time
  - 7-15% energy/instruction



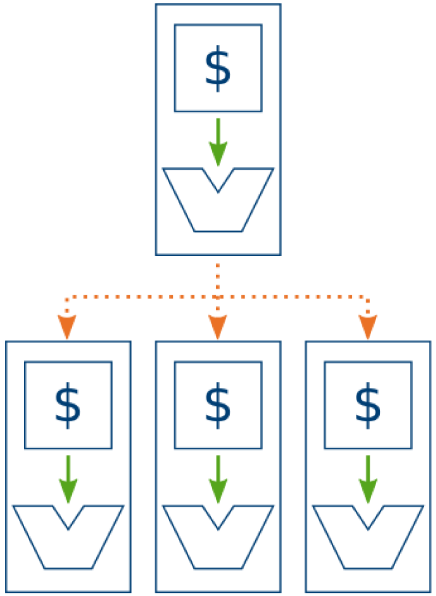
Data-level parallelism

- All cores execute the same code, but produce different parts of the output
- Improve performance by executing independent iterations simultaneously
- GPUs do this, and many modern processors have SIMD extensions
- DOALL vs. DOACROSS
- Linear speed-up, no energy increase

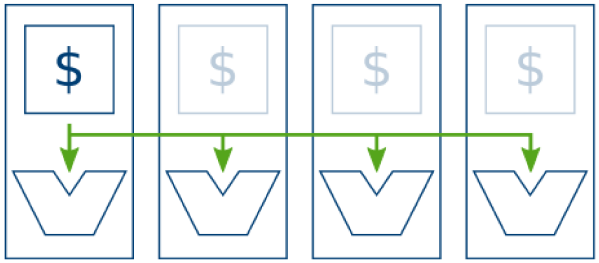


Data-level parallelism: modifications

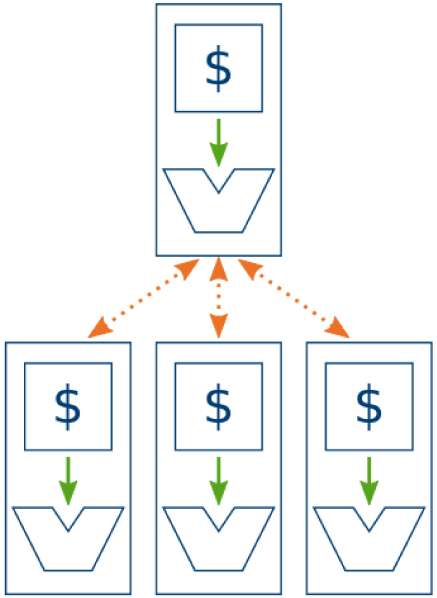
Scalarisation



Instruction sharing

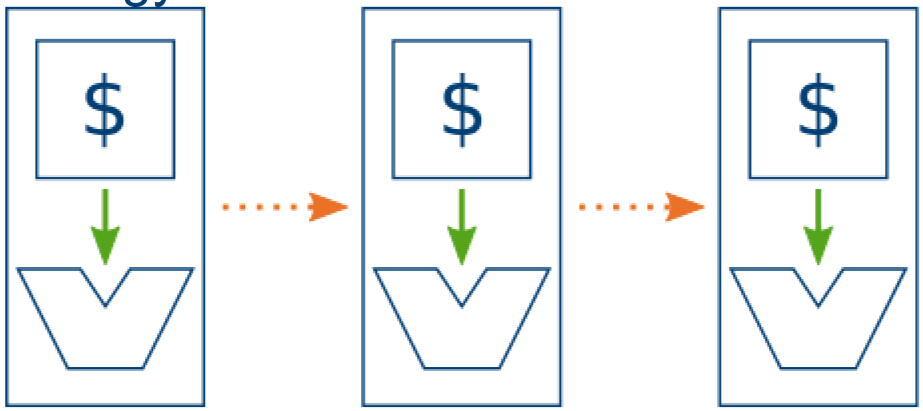


Worker farm



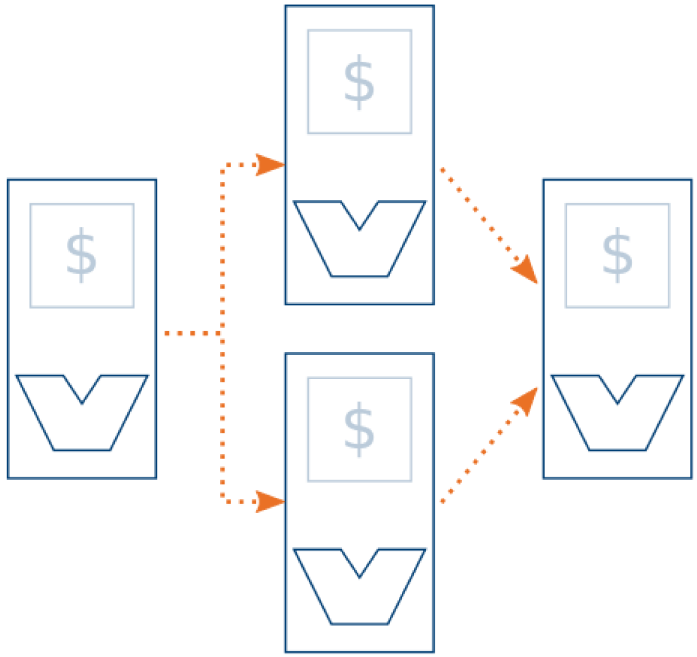
Task-level pipelines

- Each core processes the data before sending it on to the next core
- Primary aim: improve cache behaviour
- Potential to unlock more parallelism
- stringsearch: 6 cores, 4x speed-up, same energy
- adpcm: 2 cores, 2.5x speed-up, 1/3 energy



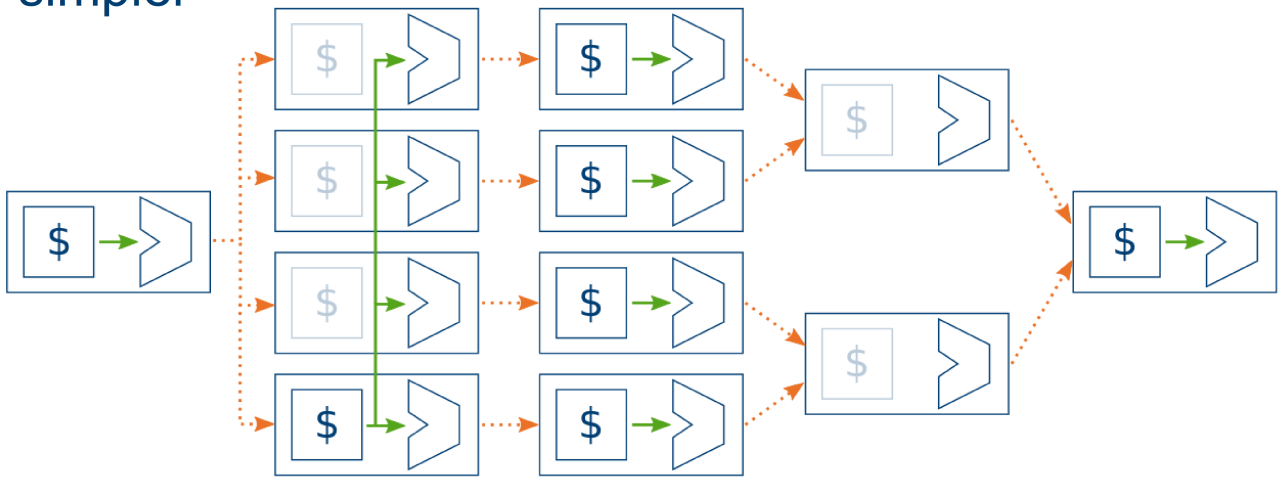
Dataflow

- The limit of task-level pipelining
- Aim to get each core down to a single instruction
- Clock gate large amounts of the pipeline
- Optimise:
  - Instruction set
  - Network
  - Pipeline
- 4.5pJ per instruction (>50% reduction)



Summary

- Loki: 10pJ per instruction, 8 cores per mm²
- Execution patterns can reduce energy and improve performance by eliminating unnecessary work
- Exposing the network to software is key
- Cores could be even simpler



Conclusion

- Step-change required
- Tightly-coupled cores allow many parallel execution patterns
- Execution patterns help to make use of resources and achieve efficiency
- Other group members: compilation, memory system, accelerators
- Future work: configurable network, OS support, fabrication!