UNIVERSITY OF CAMBRIDGE

Spatial computation on a homogeneous, many-core architecture

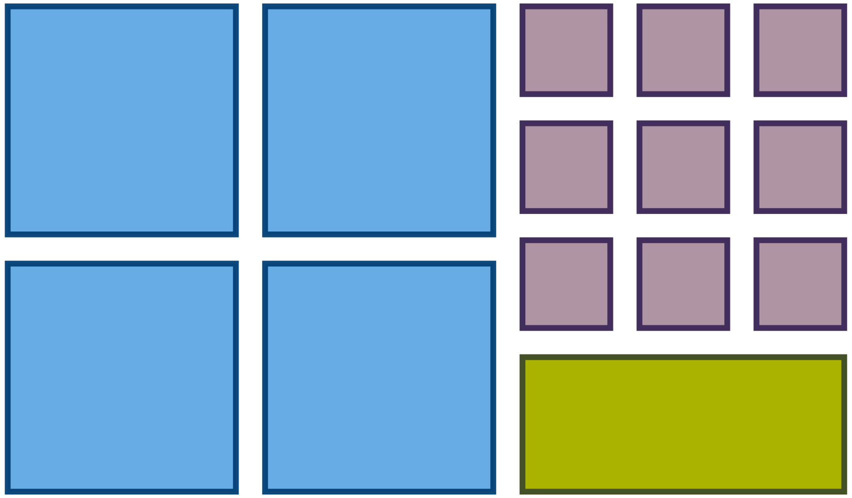
Daniel Bates, Alex Bradbury, Andreas Koltes and Robert Mullins

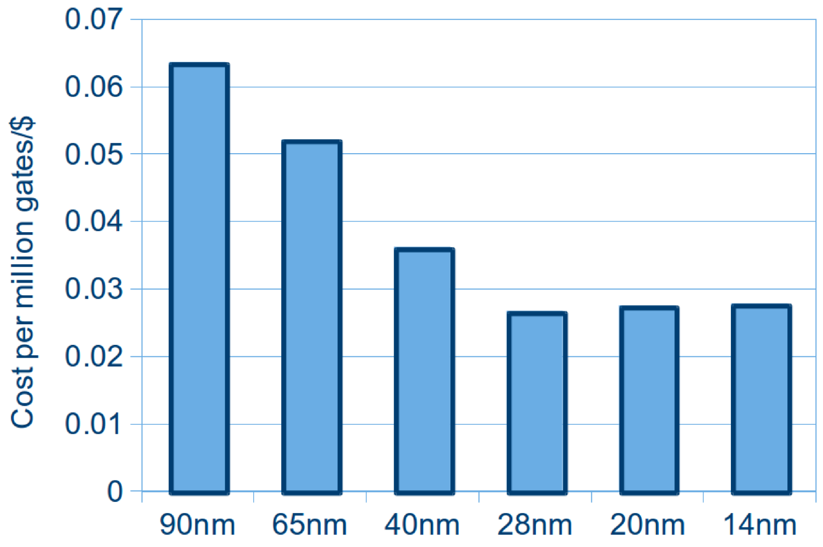
PRISM-2

Motivation: ever-changing tradeoffs

Heterogeneity won't be optimal forever

- Rising complexity
- Rising transistor costs





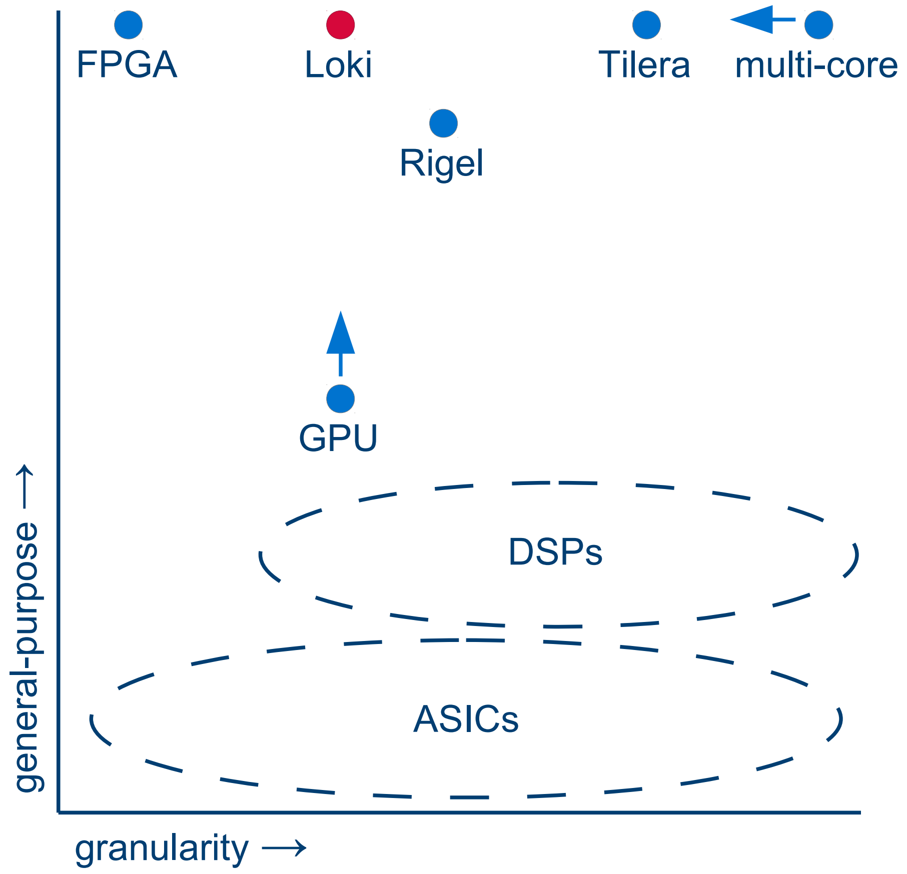
Source: *Feature dimension reduction slowdown*, by Handel Jones

Related work

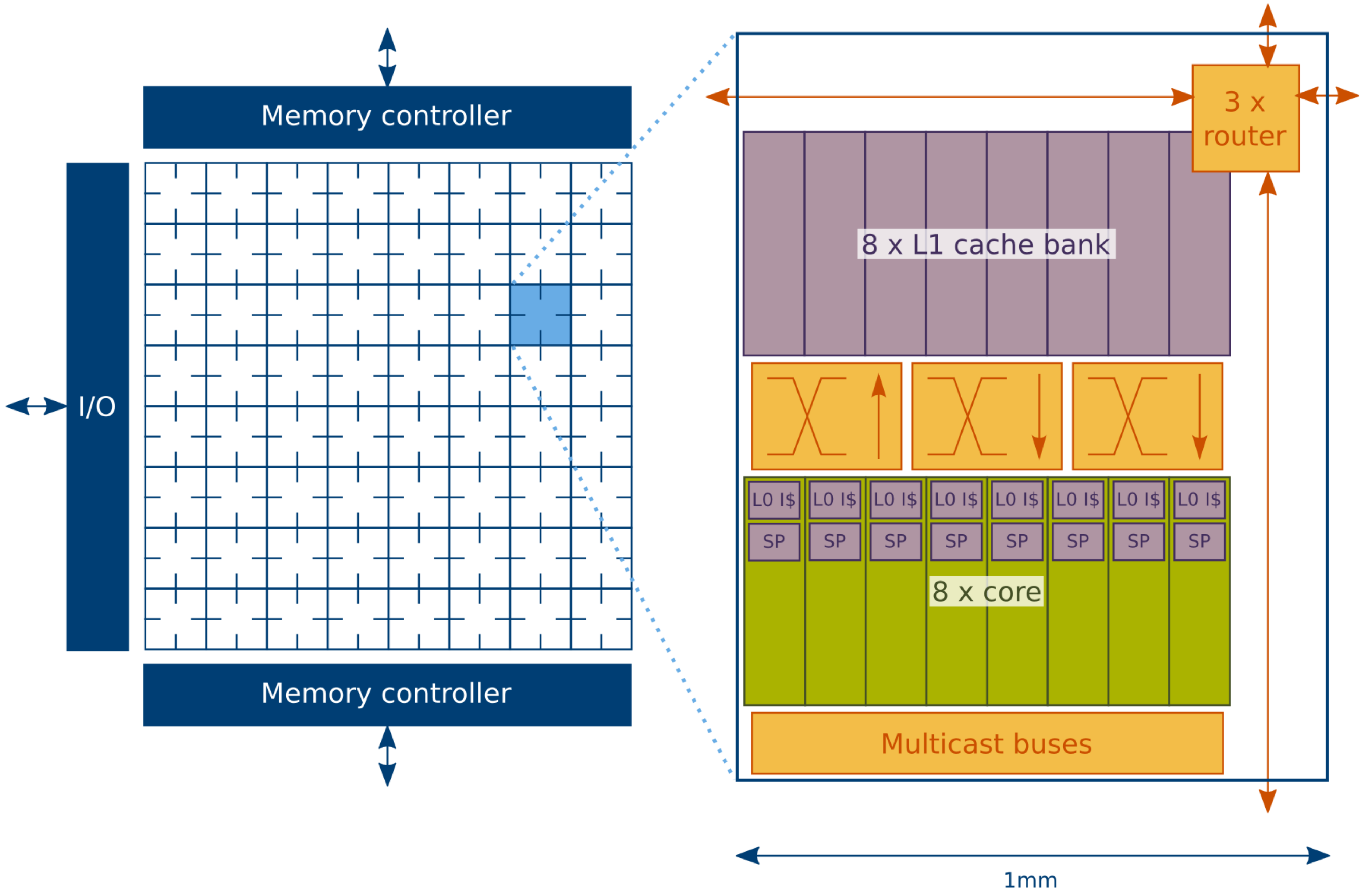
Many projects in this field:

- Elm
- Raw/Tilera
- Rigel
- TRIPS
- …

None have the same focus on flexibility and cooperation between cores.

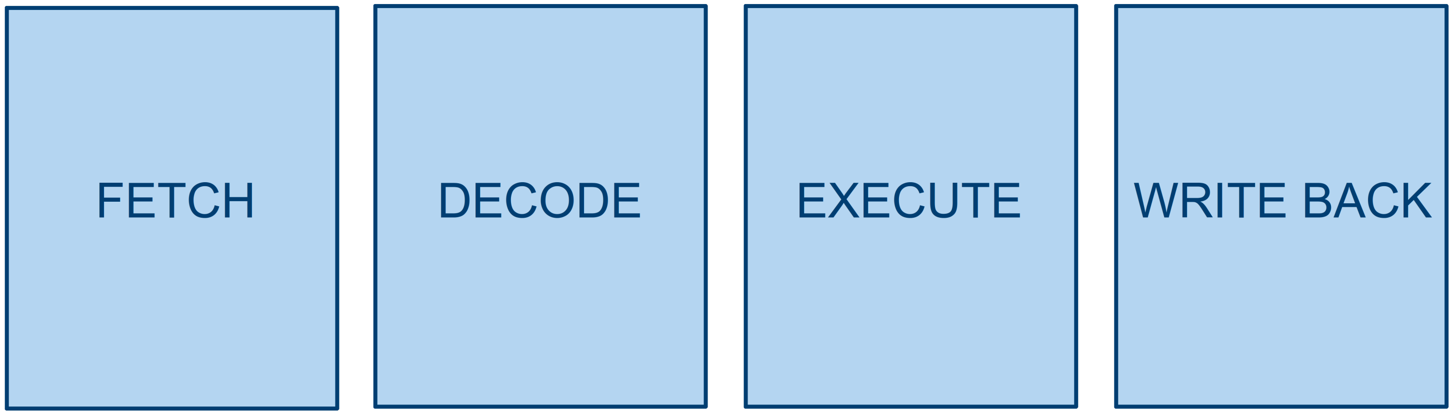


Loki architecture: chip and tile

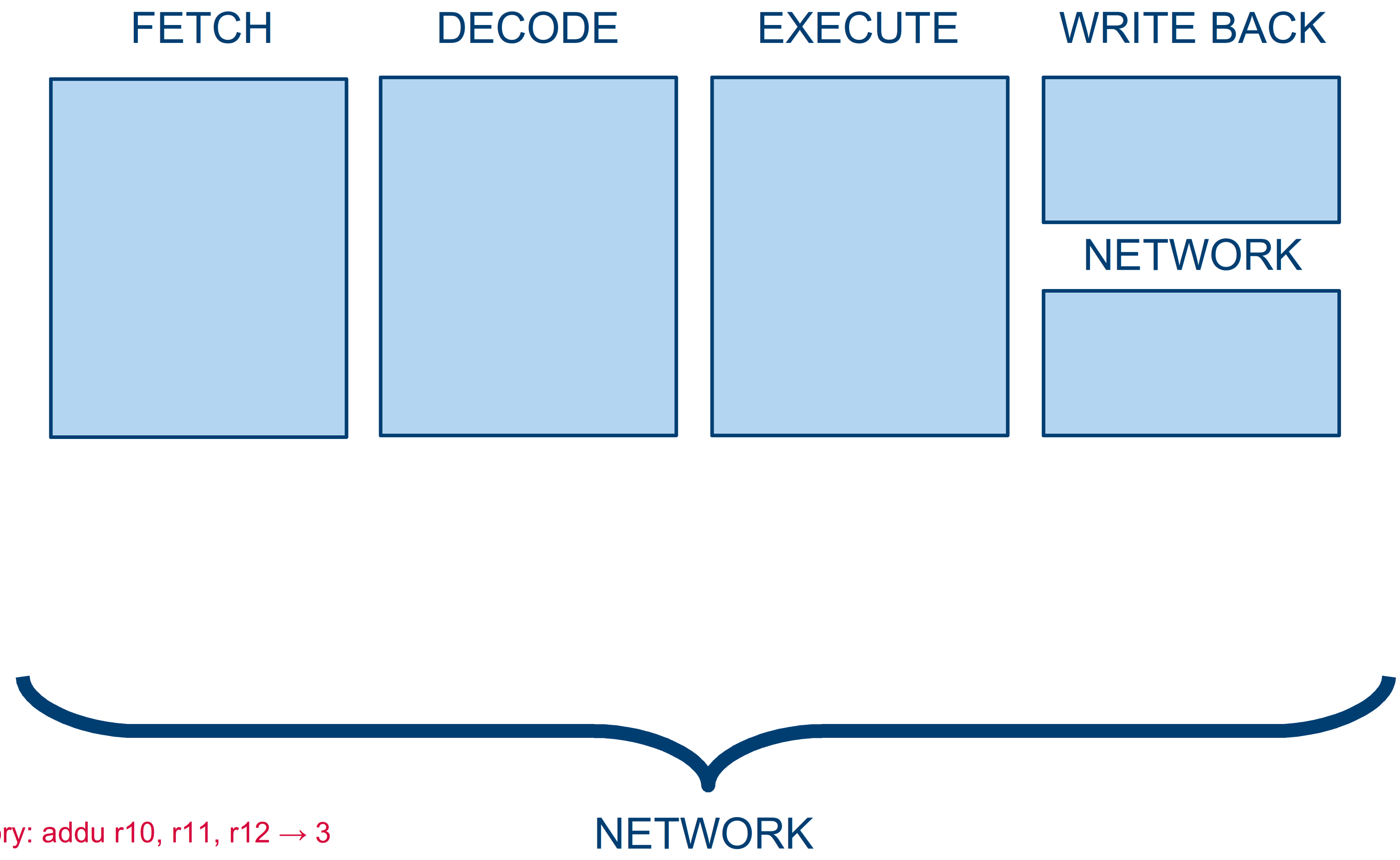


Loki architecture: key features

- 4-stage, in-order, single-issue pipeline
- Deep network integration
  - No memory access stage
- Indirect network access

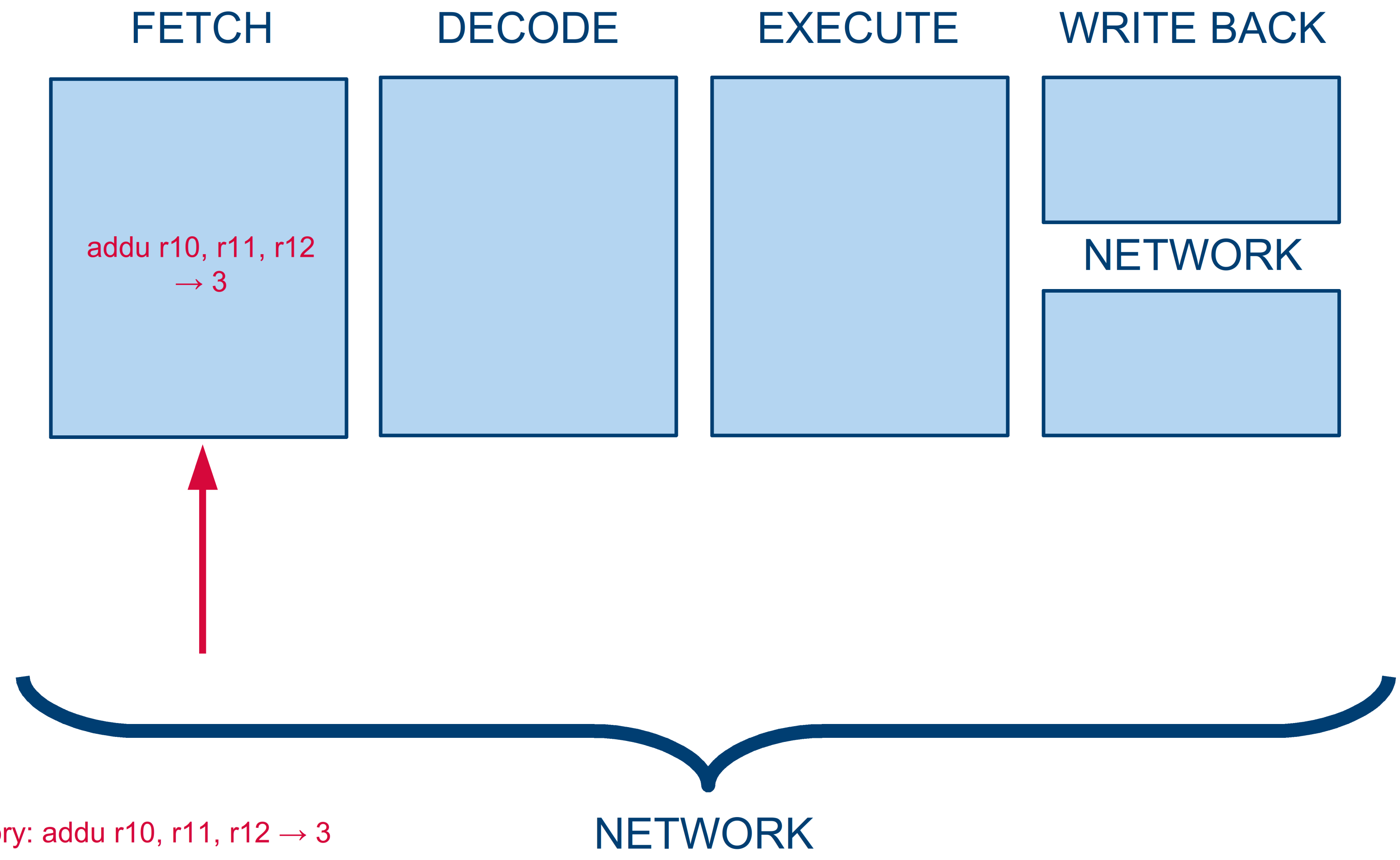


Loki architecture: key features



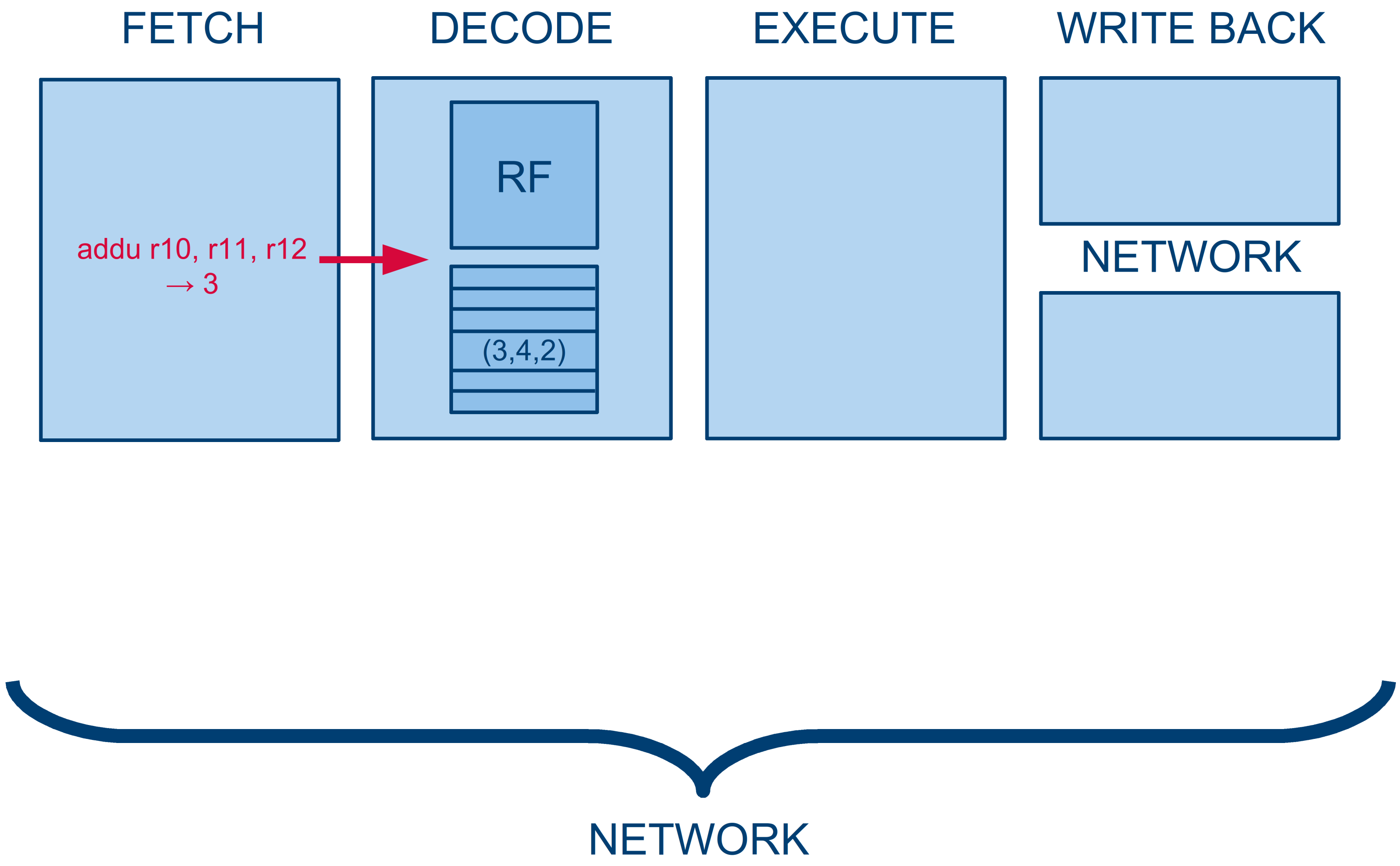
Memory: addu r10, r11, r12 → 3

Loki architecture: key features

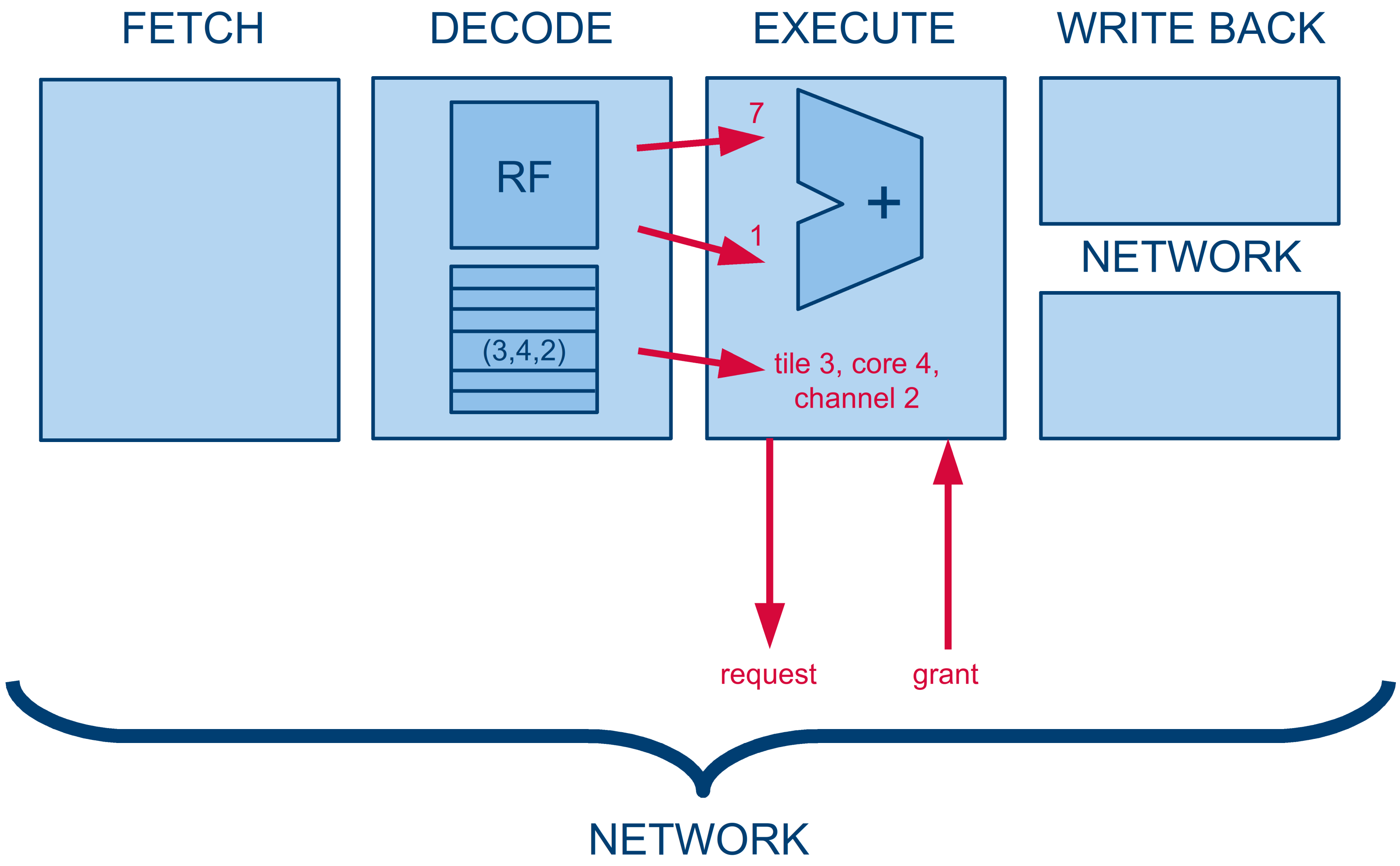


Memory: addu r10, r11, r12 → 3

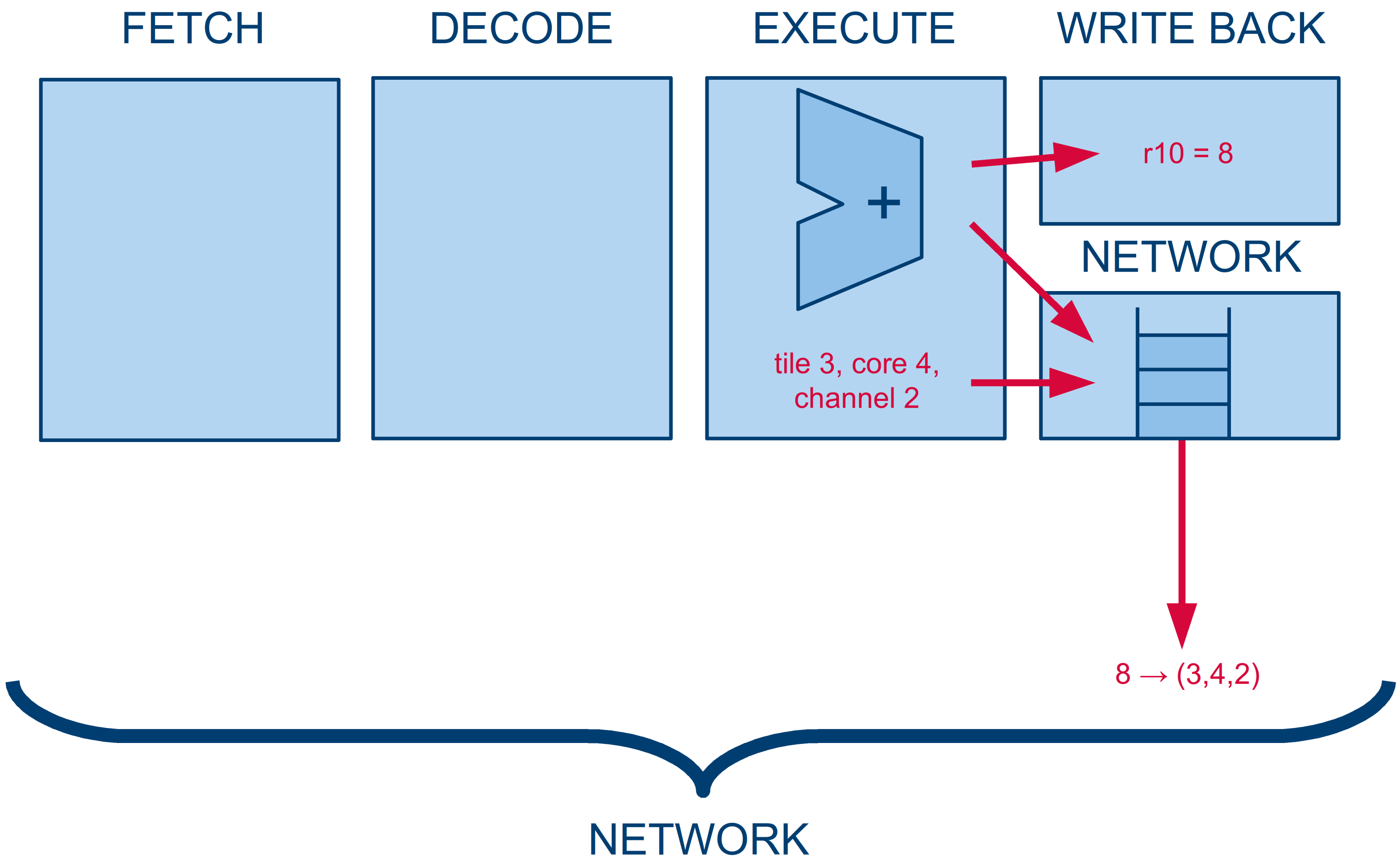
Loki architecture: key features



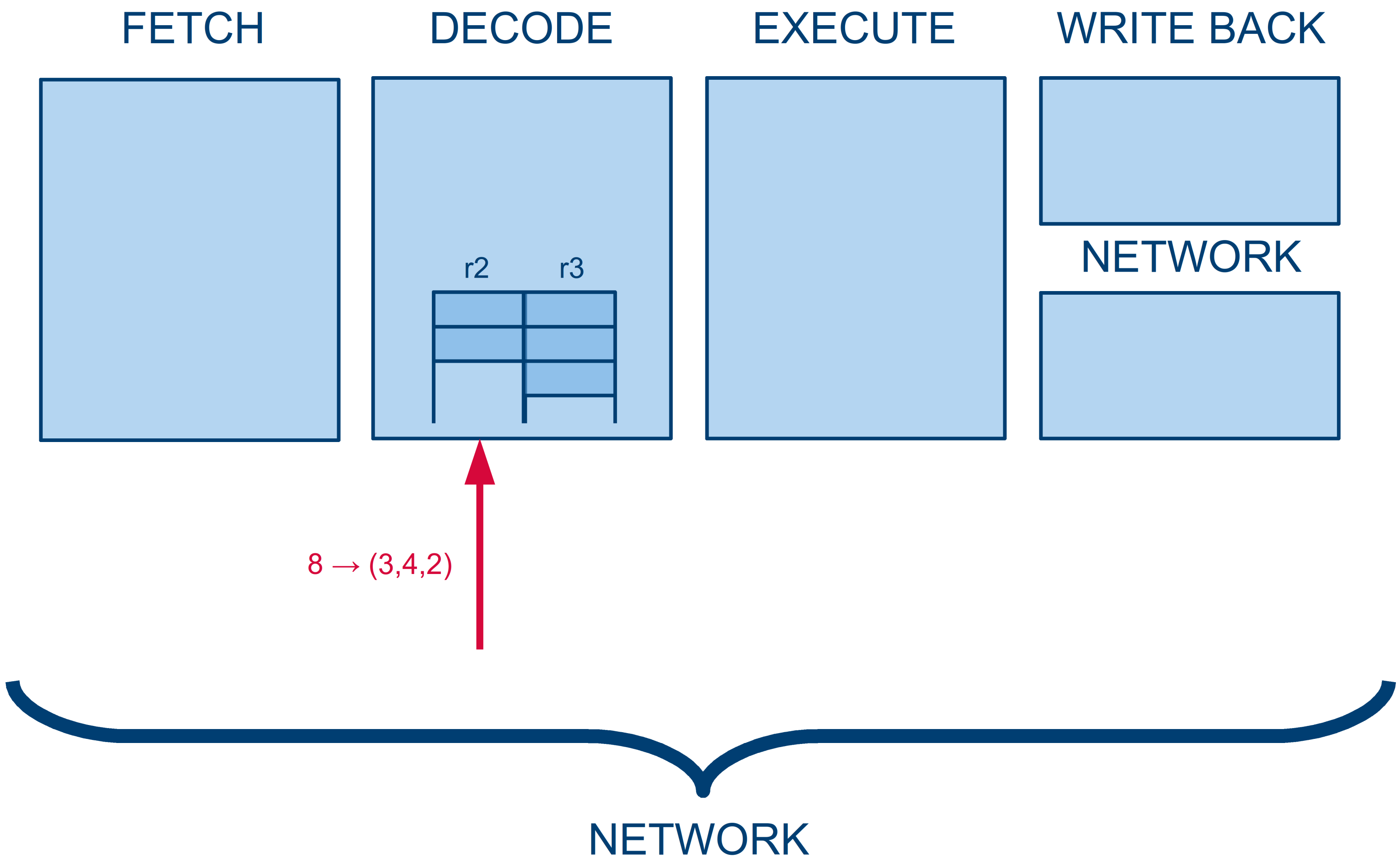
Loki architecture: key features



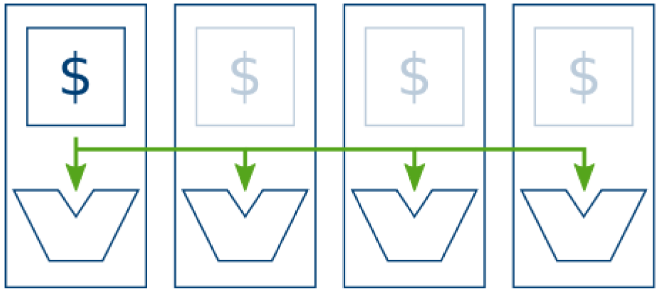
Loki architecture: key features

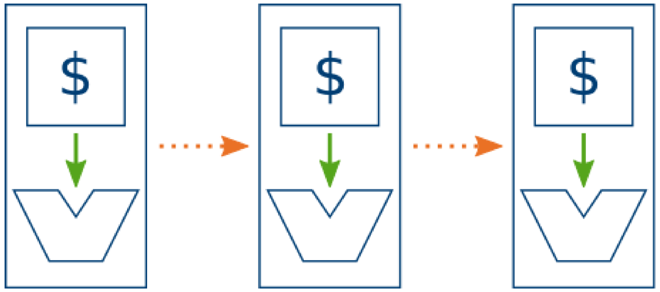


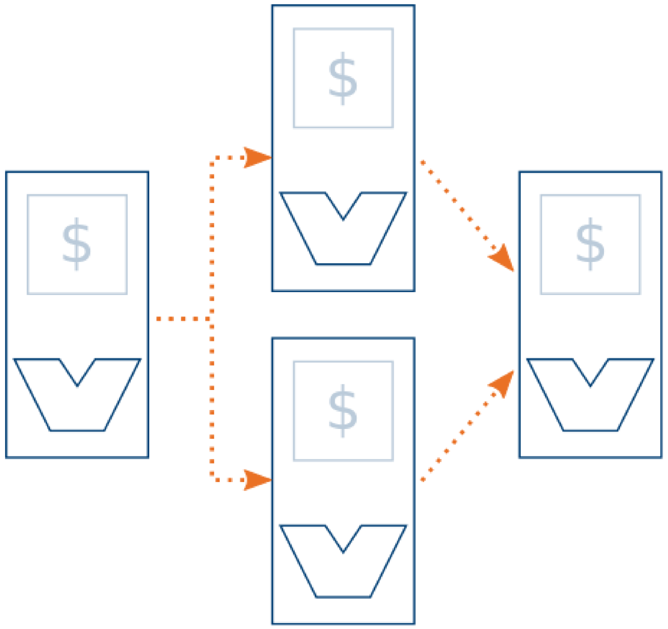
Loki architecture: key features



Previous work: execution patterns

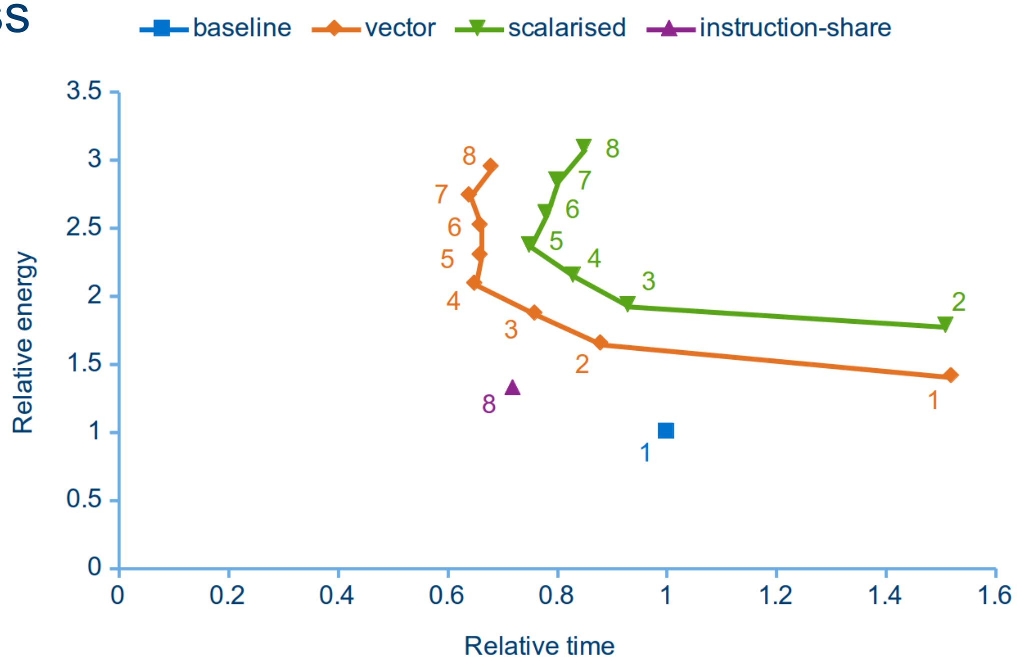






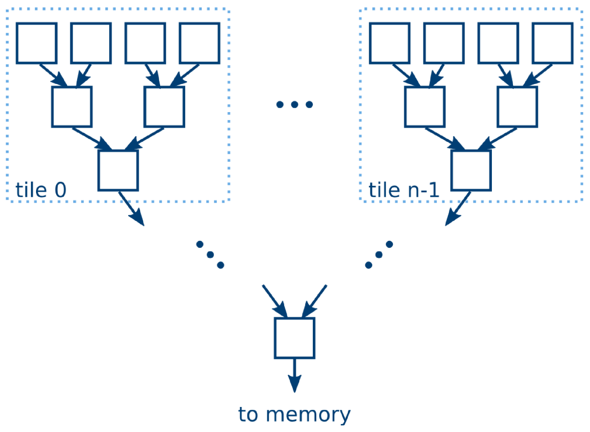
Previous work: flexibility

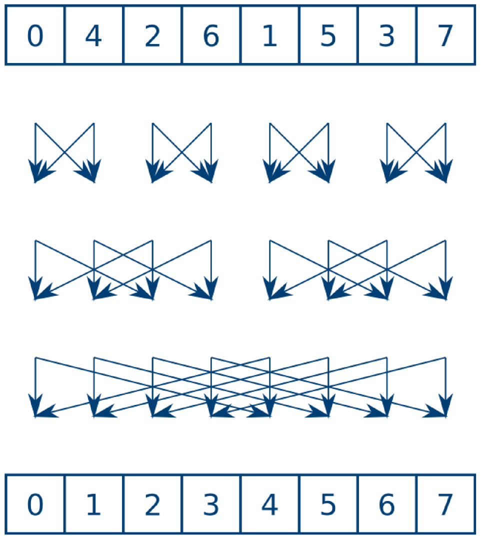
- Mappings within one tile (8 cores)
- Complementary to this work
- Now exploring mappings across multiple tiles

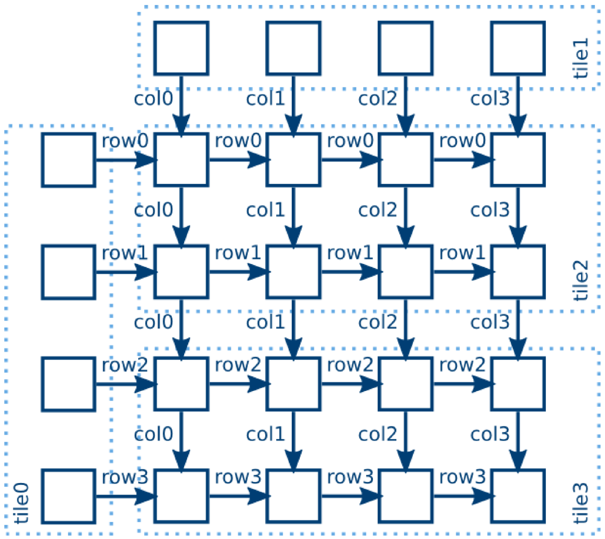


This work: case studies

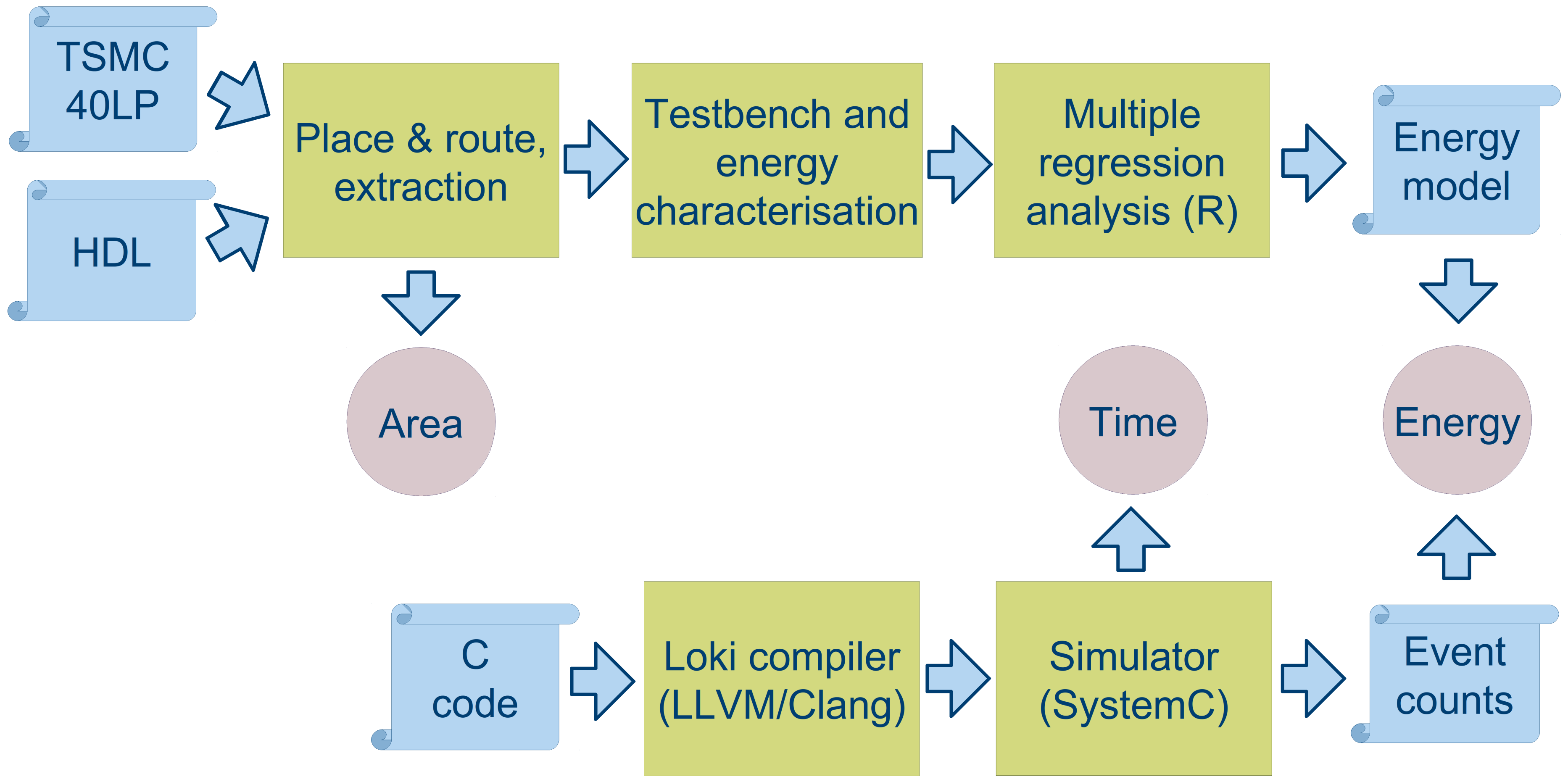
- Three kernels: sorting, FFT, matrix multiplication
- Explore different mappings
- Choose energy-performance-area tradeoff







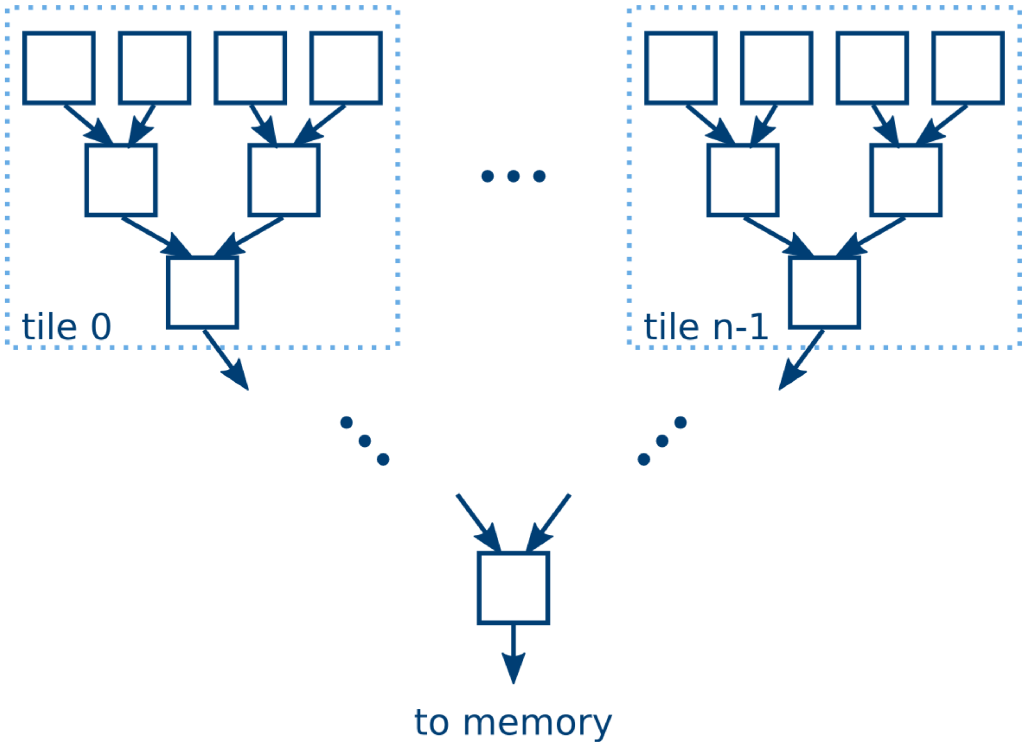
Methodology


TSMC 40LP
HDL
Place & route, extraction
Testbench and energy characterisation
Multiple regression analysis (R)
Energy model
Area
Time
Energy
C code
Loki compiler (LLVM/Clang)
Simulator (SystemC)
Event counts

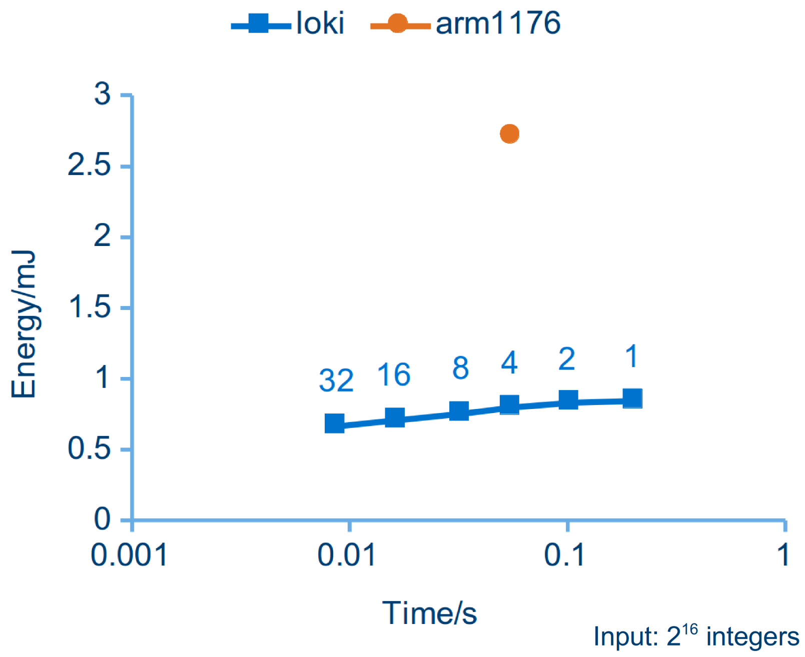
Comparison

	Loki	ARM1176
Technology	40nm LP	40nm LP
Frequency/MHz	435	700
Area/mm^2 (core + L1)	~0.125	~1
Energy per instruction/pJ	~5-20	~140
Floating point support	Software only	HW double-precision

Sorting



Critical path: 9 instructions



Sorting: reducing the bottleneck

Triggered instructions (Parashar et al, ISCA 2013)

```
compare:
        when (!p0 and !in1.finished and !in2.finished)
                p1 = in1.read() < in2.read()
                p0 = 1
send1:
        when (p0 and p1)
                out.write(in1.read())
                p0 = 0
send2:
        when (p0 and !p1)
                out.write(in2.read())
                p0 = 0
```

ISA improvements

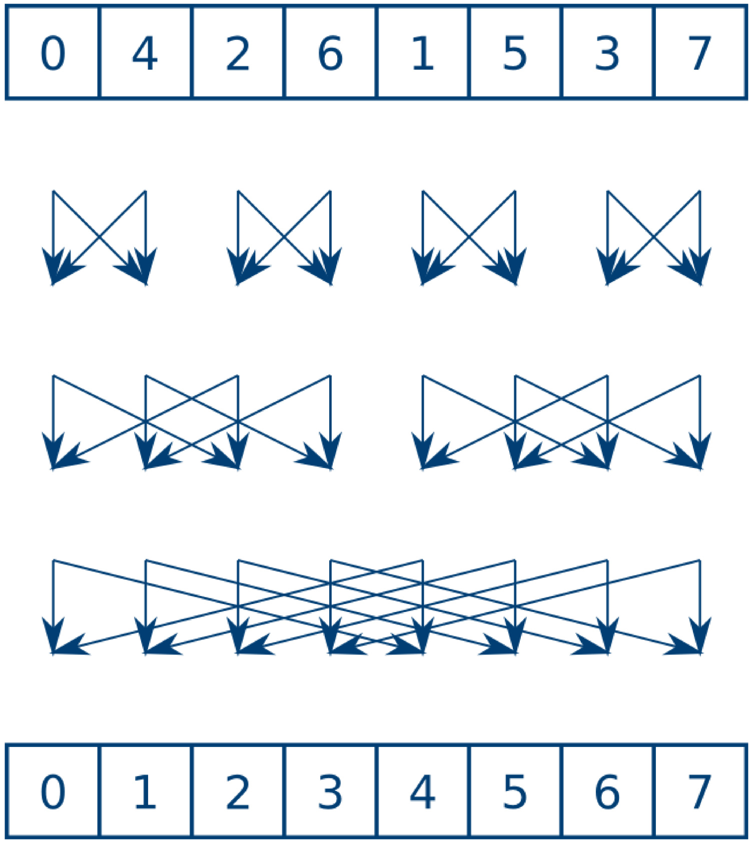
Features communication-centric architectures can borrow:

- Blocking communication
- Peek operation
- Data streams
- Triggered branch

Critical path: 4 instructions

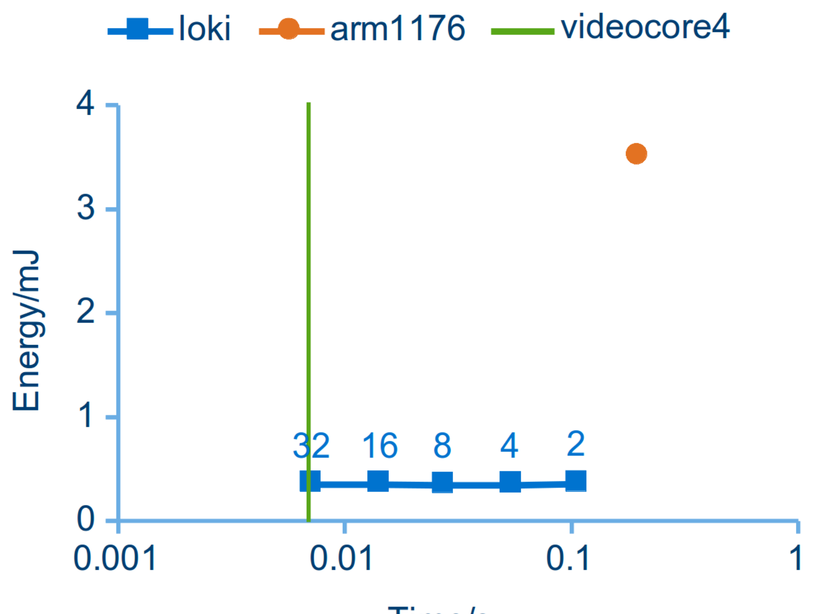
```
while true
  if in1.hasData and in2.hasData and out.hasSpace
    a = in1.read()
    b = in2.read()

    if numRead1 < listSize and numRead2 < listSize
      if a < b
        out.write(a)
        numRead1++
      else
        out.write(b)
        numRead2++

    ...
```

FFT



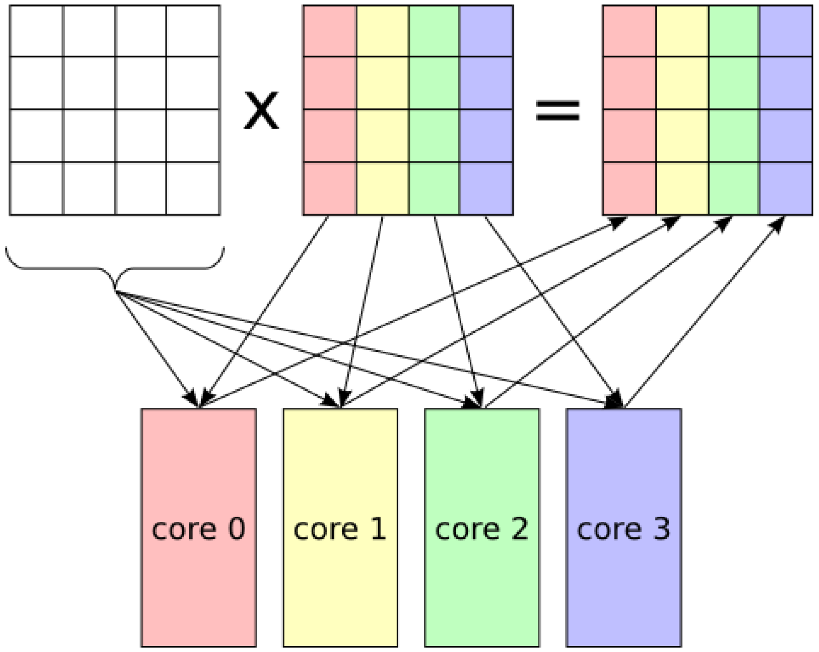
Software coherence

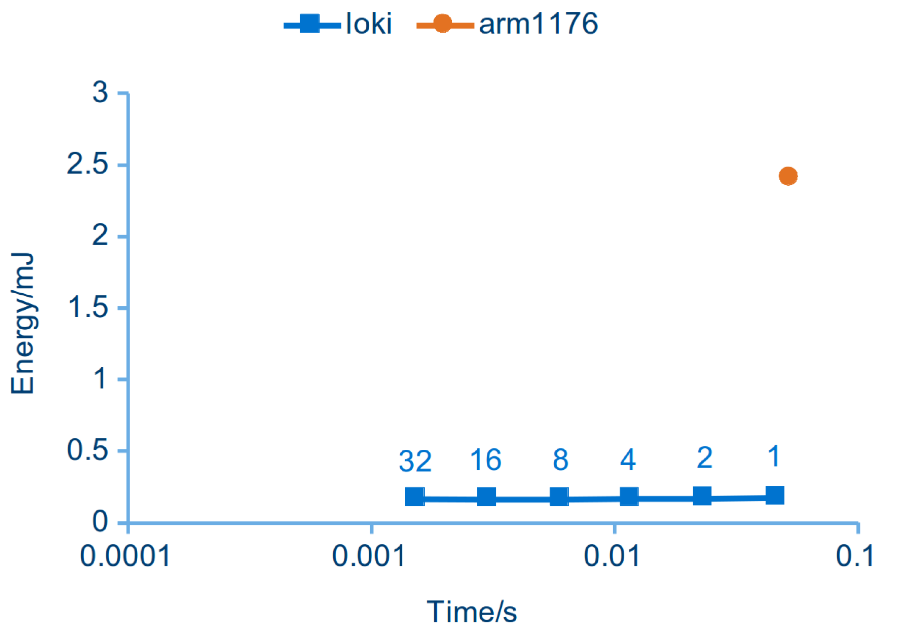
- Take the SWEL approach (Pugsley et al, PACT 2010)
  - Cache data in the lowest-level shared cache
  - Requires L1 bypass
- Also looking into configurable cache hierarchies



Input: 2^{16} complex fixed-point values

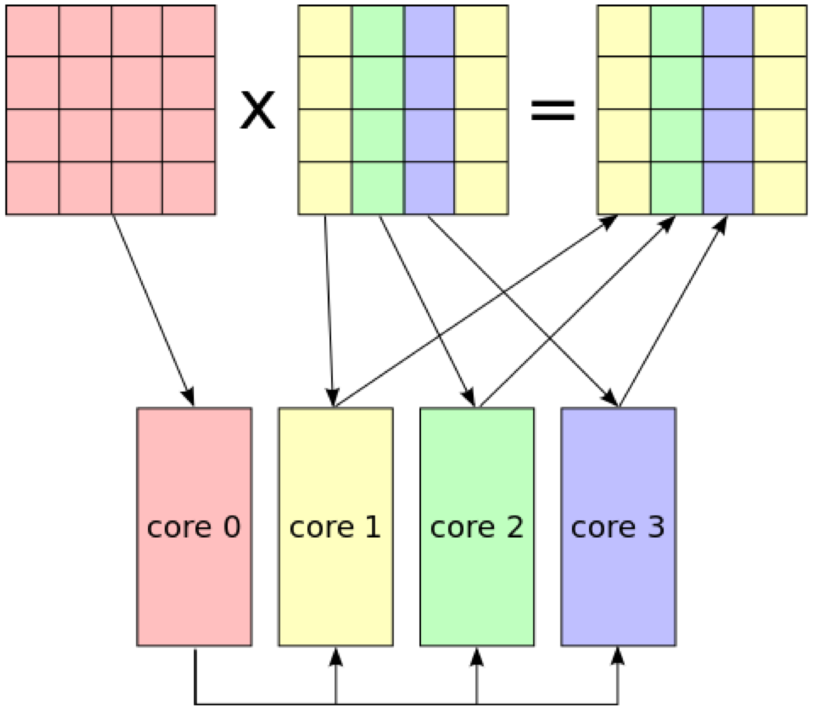
Matrix multiplication - vector

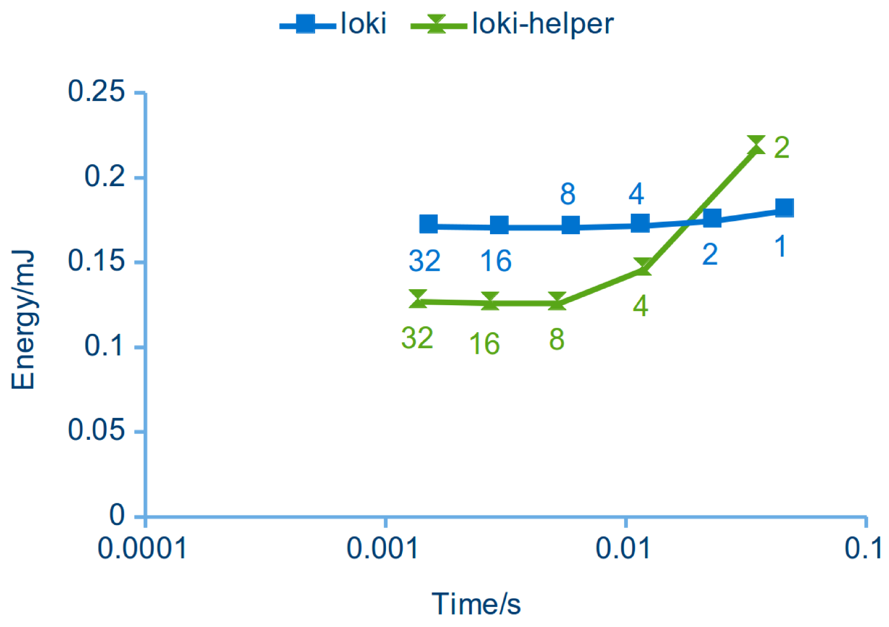




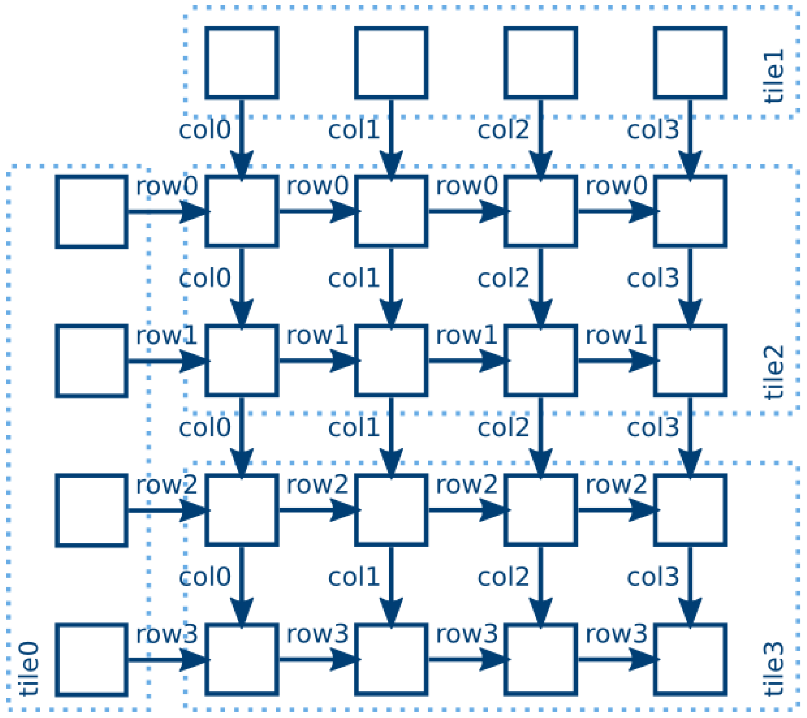
Input: two 128 x 128 integer matrices

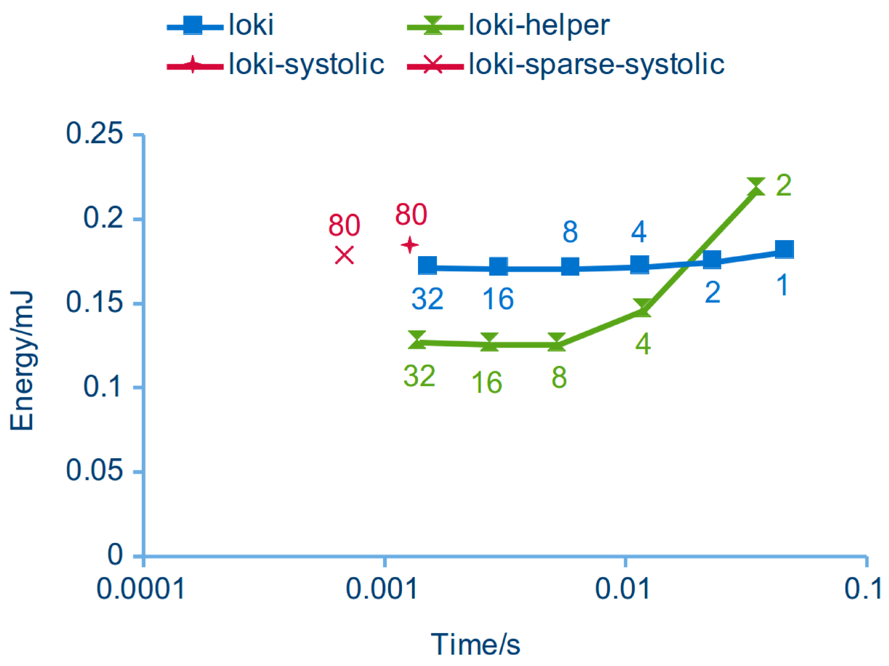
Matrix multiplication - scalarised





Matrix multiplication – systolic array

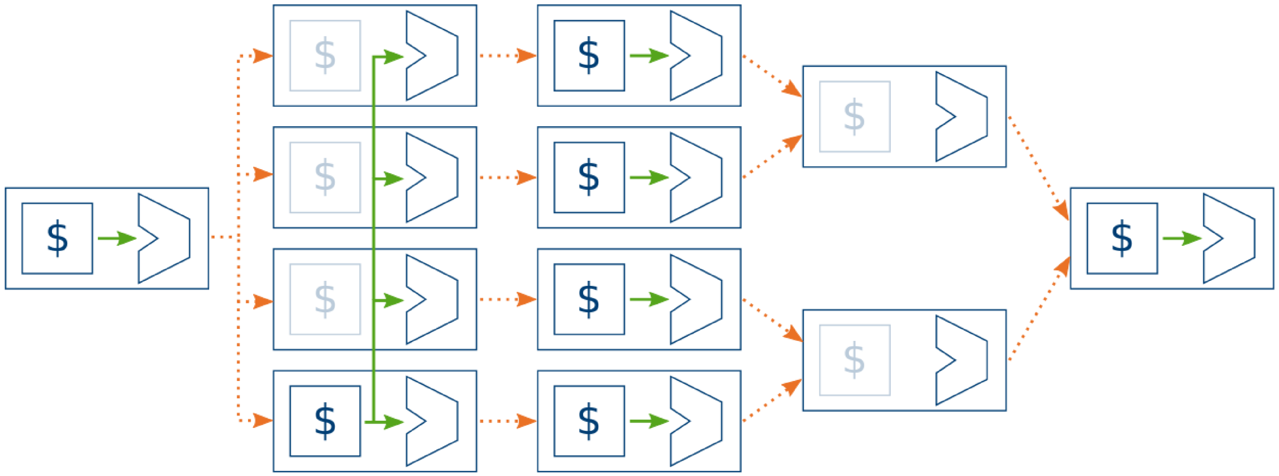




UNIVERSITY OF CAMBRIDGE

Summary

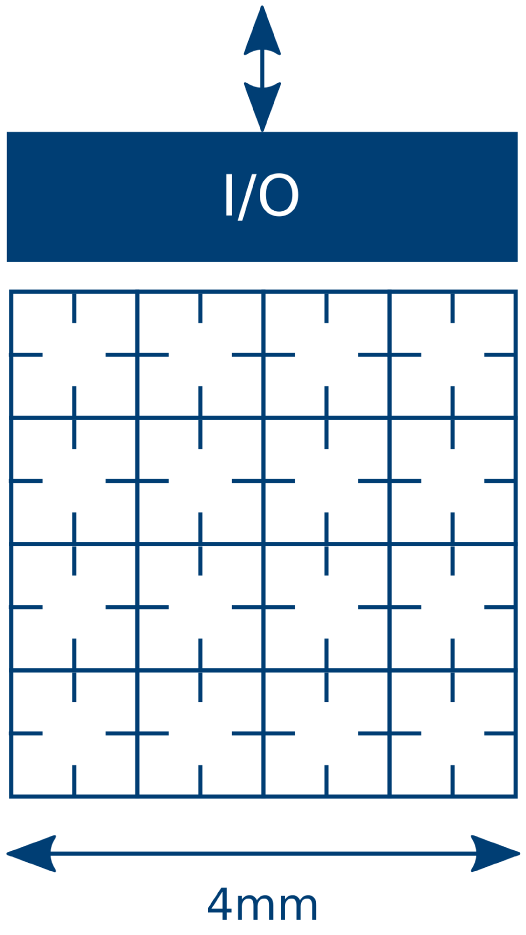
- We can use large numbers of cores effectively
- Promising performance and energy figures
- A step towards a fully-homogeneous system
- Flexible communication is key



UNIVERSITY OF CAMBRIDGE

What's next?

- Memory system
- Network protocols
- Compiler optimisations
- Programmable accelerators
- Test chip (2015)
  - 4 x 4 tiles x 8 cores/tile = 128 cores
  - Hoping to share development boards
- Operating system



UNIVERSITY OF CAMBRIDGE

Spatial computation on a homogeneous, many-core architecture

Daniel Bates, Alex Bradbury, Andreas Koltes and Robert Mullins

Daniel.Bates@cl.cam.ac.uk

www.cl.cam.ac.uk/~rdm34/loki/

PRISM-2