MemPool: A Shared-L1 Memory Many-Core Cluster with a Low-Latency Interconnect

Matheus Cavalcante$^1$, Samuel Riedel$^1$, Antonio Pullini$^2$, Luca Benini$^{13}$
matheusd at iis.ee.ethz.ch

$^1$ETH Zürich, Zürich, Switzerland
$^2$Greenwaves Technologies, Grenoble, France
$^3$Università di Bologna, Bologna, Italy
• Shared-L1 clusters are an extremely common pattern
  • Simple programming model
  • Only scales to a few tens of cores

• Our proposal: MemPool
  • Many-core cluster with 256 32-bit RISC-V cores
  • Low-latency shared view of 1 MiB of L1 memory

• Physical-aware design with GlobalFoundries’ 22FDX
  • 700 MHz at typical conditions, critical path 53 gates long
MemPool’s hierarchy: the Snitch core

- Ultra-small RV32IMA core
  - 44 kGE
- Functional units outside the main core pipeline
  - Pipelineable complex instructions
- Latency-tolerant memory interface

For more information, refer to arXiv:2002.10143 [cs.AR]
MemPool’s hierarchy: the tile

• Four Snitch cores
• 2 KiB shared L1 instruction cache
  • 4-way set associative
  • L0 cache private to core
• 16 L1 SPM Banks
  • 16 KiB
  • Accessible from local cores within one cycle
• Each tile has K ports to access banks in remote tiles
  • Traffic concentration
  • Possible throughput bottleneck!

• Remote request/response interconnects:
  • $4 \times K$ fully-connected logarithmic crossbars

• Request and response interconnects
  • $(4+K) \times 16$ fully-connected logarithmic crossbars
MemPool cluster: assembling the 64 tiles

• Approach #1 (Top₁)
  • K = 1 master and slave ports to access banks in remote tiles
  • Single $64 \times 64$ butterfly network
  • Traffic concentration bottleneck

• Approach #2 (Top₄)
  • K = 4 master and slave ports to access banks in remote tiles
  • Four $64 \times 64$ butterfly networks
  • Routing congestion bottleneck
  • Homogeneous 5-cycle latency
• 16 tiles compose a **group**
• Tiles in the same group can be accessed within 3 cycles
• Directional ports to access remote groups
  • North, Northeast, East
  • Remote groups can be accessed within 5 cycles
• Radix-4 16x16 Butterflies
Throughput and Latency Analysis

• Cores replaced with synthetic traffic generators
  • Uniformly random access pattern
• \( \text{Top}_1 \) saturates fast
  • Traffic concentration issue
• \( \text{Top}_4 \) and \( \text{Top}_H \) are pretty much equivalent
  • \( \text{Top}_H \) has slightly lower latency
  • Latency below 6 cycles for a load of 0.25 req/core/cycle
• Baseline: idealized Top$_X$
  • Fully connected logarithmic crossbar between 256 cores and 1024 banks

• Cycle-accurate RTL simulation
  • matmul
    • Multiplication of two $64 \times 64$ matrices
  • 2dconv
    • 2D Convolution with a $3 \times 3$ kernel
  • dct
    • 2D Discrete Cosine Transform on $8 \times 8$ blocks in local memory

• Top$_H$ has a performance penalty of at most 20%, on all kernels
Back-end results: tile implementation in GF 22FDX

• Synopsys flow:
  • DesignCompiler 2019.12 for synthesis and IC Compiler II 2019.12 for PnR

• Dense and compact tile:
  • 425 μm × 425 μm (908 kGE)
    • 72.8% utilization
  • Routed with six layers
    • Four layers for above-the-tile routing
MemPool’s implementation in GF 22FDX

- Two limiting factors:
  - Routing congestion:
    - 4 interconnects competing for routing resources
  - Propagation delay:
    - Wires need to cross long distances → high utilization of upper routing layers
- Top₄ is physically unfeasible
- Top₄: 4.6 × 4.6 mm macro
  - 55% of it occupied by tiles
  - 700 MHz at typical conditions
Power Analysis

• Switching activities extracted from running `matmul`
• Extraction with PrimeTime 2019.12 at typical conditions
  • 500 MHz, TT, 0.80 V, 25 °C

• Each tile consumes 20.9 mW
  • The interconnects consume 1.7 mW, <10% of the total consumption

• MemPool consumes 1.55 W
  • The tiles are responsible for 86% of that
  • The global interconnect consumes 211 mW, 14% of the total consumption
Breakdown of the energy consumption per instruction:

- **Add** consumes 3.7 pJ
- **Mul** consumes 7.0 pJ
- **Local load** consumes 2.1 pJ, 4.5 pJ, and 1.8 pJ

- Local loads consume about as much energy as a *mul*
  - About half of it, 4.5 pJ, by the interconnect
- Remote loads consume twice the energy of a local load
  - Despite crossing the whole cluster, twice!
• What is working:
  • A 256-core shared L1 cluster, with all banks accessible within 5 cycles
  • Performance within 20% of the ideal baseline, on key benchmarks
  • 700 MHz at typical conditions (GF 22FDX)

• What is next:
  • Increment the core with DSP functional units
  • Develop the software environment and extend the benchmarks
    • Halide
  • Scale up the number of cores
MemPool: A Shared-L1 Memory Many-Core Cluster with a Low-Latency Interconnect

Matheus Cavalcante\textsuperscript{1}, Samuel Riedel\textsuperscript{1}, Antonio Pullini\textsuperscript{2}, Luca Benini\textsuperscript{13}

matheusd at iis.ee.ethz.ch

\textsuperscript{1}ETH Zürich, Zürich, Switzerland
\textsuperscript{2}Greenwaves Technologies, Grenoble, France
\textsuperscript{3}Università di Bologna, Bologna, Italy