Manticore

A 4096-core RISC-V Chiplet Architecture for Ultra-efficient Floating-point Computing

Florian Zaruba[*], zarubaf@iis.ee.ethz.ch
Fabian Schuiki[*], fschuiki@iis.ee.ethz.ch
Luca Benini[*†], lbenini@iis.ee.ethz.ch

[*] Integrated Systems Laboratory, ETH Zurich
[†] University of Bologna

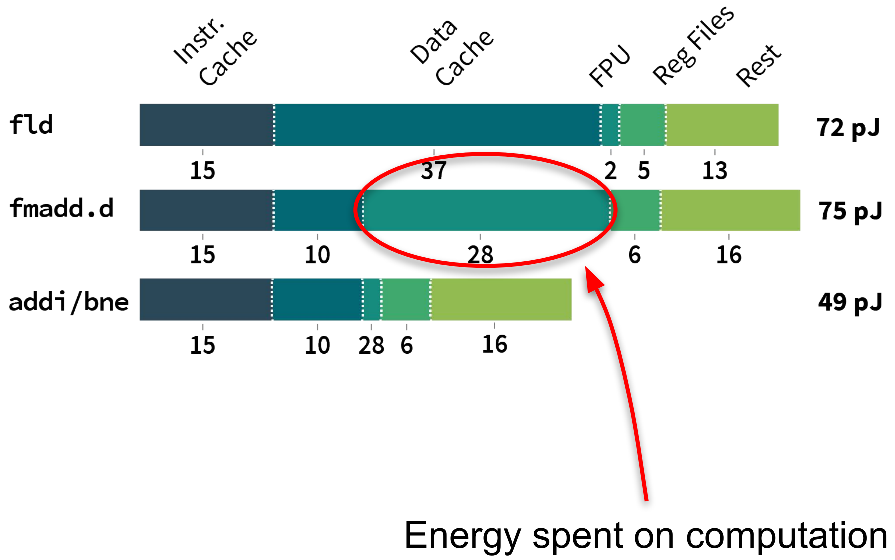
ETH zürich

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Introduction

- Ever growing demand for floating-point operations:
  - data-analytics, machine learning, scientific computing
- Tight energy-efficiency constraints
  - ➢ Node shrink: Increasing power density
  - ➢ Thermal design power limits the amount of active compute units
- Precision still counts (≥ fp32):
  - ➢ Stencils, linear differential equations
- Domain-specific architectures are hard to adjust to algorithmic changes
- Most of the energy spent on control:
  - Instruction cache, out-of-order execution
  - Von-Neumann bottleneck

Application-class processor (Ariane):
22nm FDX, 0.8V, 1 GHz, DGEMM



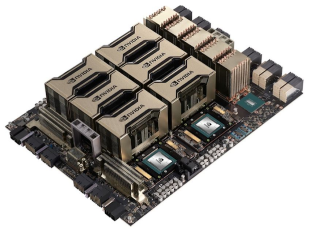
A Taste of Where Supercomputing is Heading

Fujitsu A64FX

- TSMC 7nm, CoWoS, ~8.7bn transistors
- Armv8-A SVE
- Many-core architecture
- **Wide per-core SIMD data path (better FPU/Control power ratio)**

NVIDIA A100

- TSMC 7nm, CoWoS, ~54bn transistors
- AMPERE architecture
- SIMT data path
- **Per-thread program counter; finer-grained threads**
- **Larger data-path (tensor cores)**

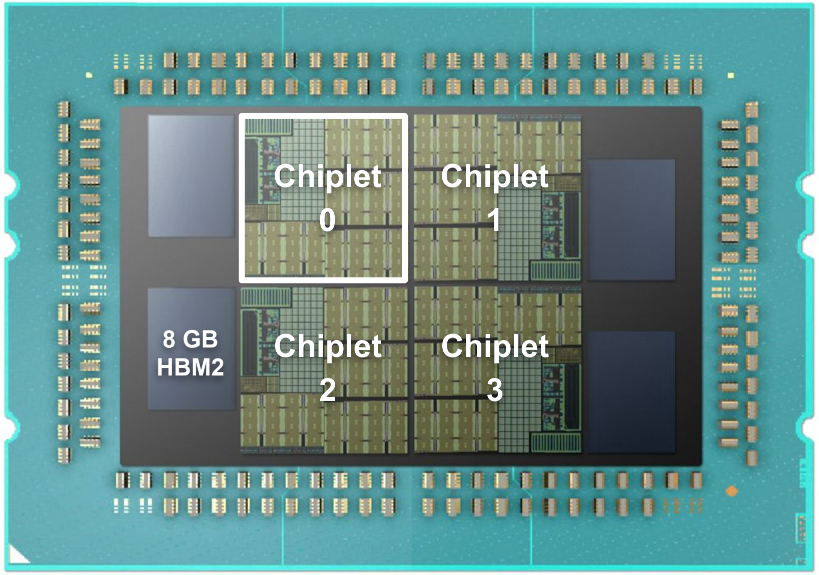


Maximize computation data path with respect to control!

"Fujitsu High Performance CPU for the Post-K Computer", HotChips 30

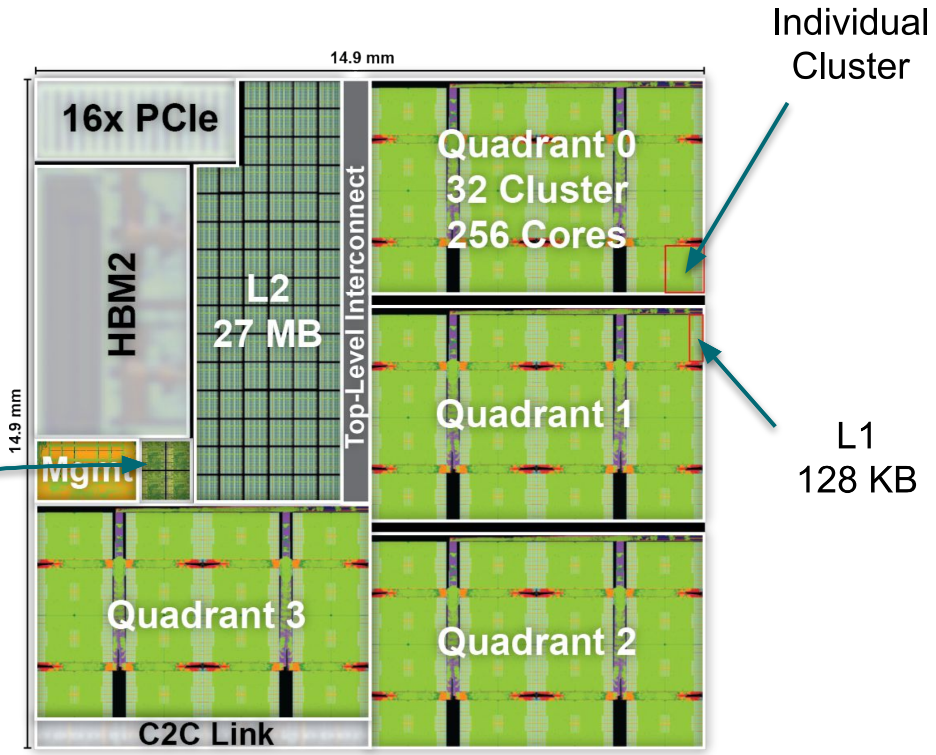
The Manticore Multi-Chiplet Concept

- Four chiplets:
  - 222mm^2 (14.9 x 14.9mm)
  - Estimated in a 22nm process
  - ➢ Yield and cost improvements
- Three die-to-die links:
  - Each die has short-range, multi-channel, die-to-die links to each sibling
  - ➢ Efficient inter-die synchronization
  - ➢ D2D non-uniform memory access
- Private **8GB HBM2** per die
  - ➢ SoA BW and efficiency
- 16x PCIe Endpoint
  - ➢ Flexible host communication
  - ➢ Industry standard



Chiplet Architecture

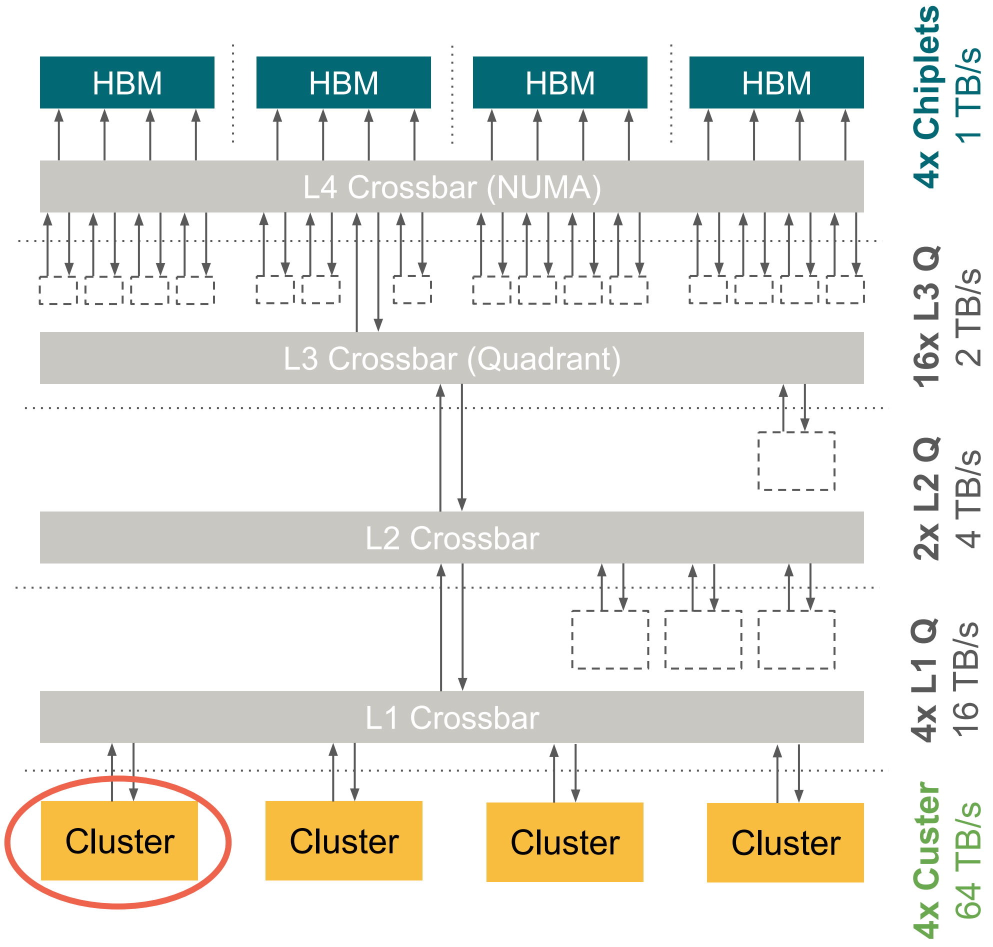
- Methodology:
  - Quadrants/L2/Ariane placed and routed
  - HBM2, PCIe estimated and scaled to tech
- **Four quadrants of 32 clusters**
  - 8 Snitch cores per cluster
  - 16 DP and 32 SP flops per cycle
  - 1 GHz operating frequency
  - Peak **> 4 Tdpflop/s** per chiplet
- Four Ariane "manager" cores
  - ➢ Run Linux
  - ➢ Management and offloading
- 27 MB L2 Memory
  - ➢ Balance thermal budget
  - ➢ Utilize die area



Quadrant Subdivision

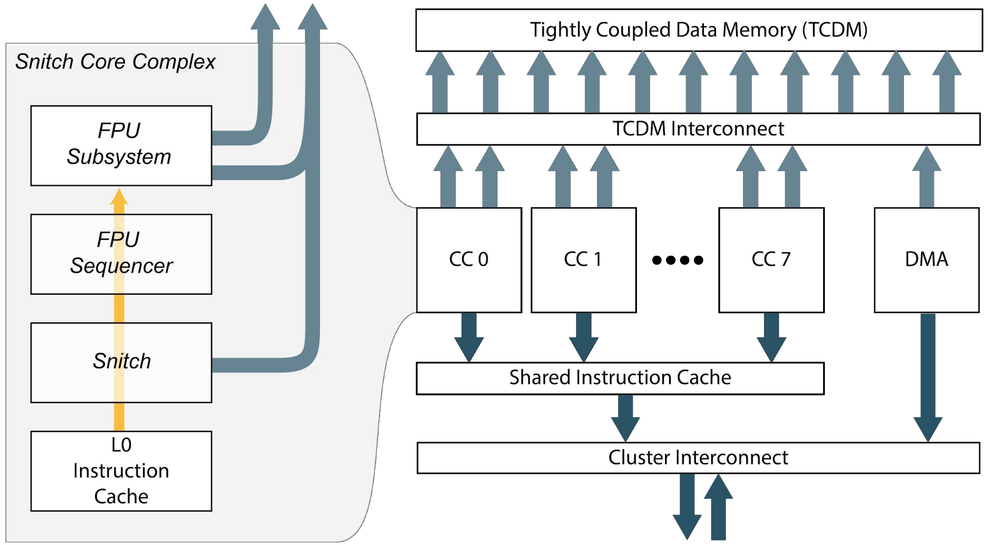
- System subdivided hierarchically into multi-level quadrants (as a tree)
  - ➢ **High aggregate bandwidth** of up to **64 TB/s** among quadrants at **lower levels**
  - ➢ Thinned to **sustainable HBM bandwidth** of **1 TB/s** at higher levels
- Low diameter and high BW

- **4x clusters** share instruction cache and uplink to L1 quadrant
- **4x L1** quadrants share instruction cache and uplink to L2
- **2x L2** quadrants share uplink to L3
- **4x HBMs** connected to **16x L3** quadrants



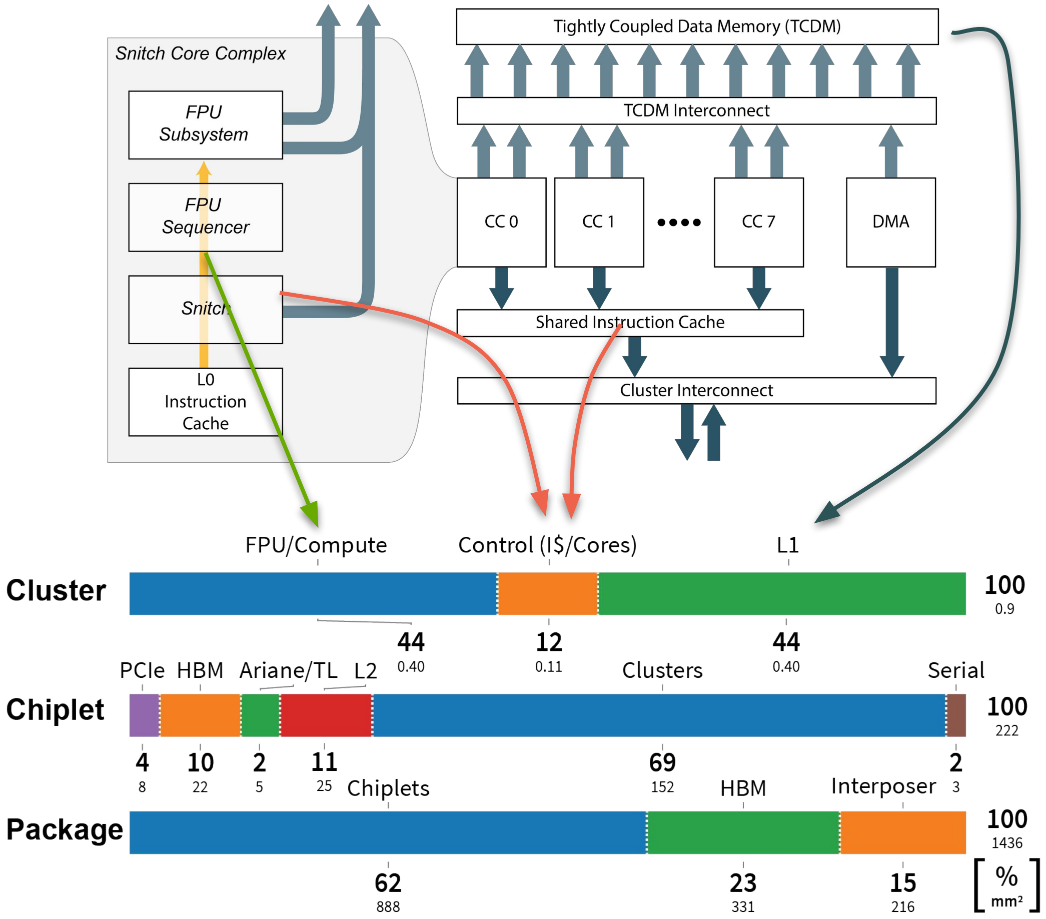
Compute Cluster

- **8x RV32G Snitch** cores
  - Optional w/ 16 GPRs
  - Single-stage
  - ➢ Area-efficient: 9-22 kGE
- **8x Large FPU**
  - Decoupled and heavily pipelined
  - Multi-format FPU (+SIMD) (half-precision, bfloat, custom fp8)
  - ➢ Source of useful compute!
- **128 kB TCDM**
  - ➢ Scratchpad for predictable memory accesses
- **DMA** w/ 512 bit data interface
  - ➢ Efficient data movement
- **Custom ISA extensions**
  - Xssr and Xfrep



Focus on Compute

- **Goal:**
  Maximize compute/control ratio
- **Small cores** with **large FPUs**
  - ➢ SSRs/FREP allow for small cores with high compute utilization
- **Multi-banked** scratchpad memories (TCDM):
  - ➢ High sustainable, element-wise, BW
  - ➢ Similar to a register file in a VPU/GPU
  - ➢ Efficient access with SSRs
  - ➢ Fine-grained inter-cluster synchronization
- Async. **data movement** with **DMA**
  - ➢ Efficient bulk data transfer



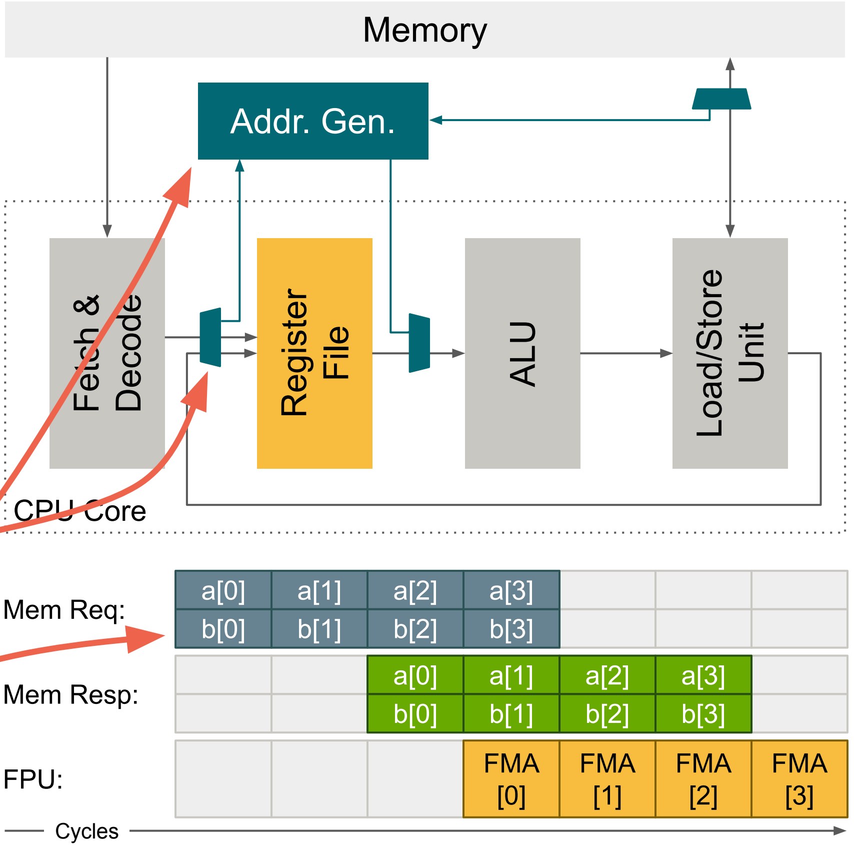
Taming the Beast

Stream Semantic Registers (Xssr)

- Turn register read/writes into **implicit** memory loads/stores
- Elides many **explicit** load/store instructions
    - Increases FPU/ALU utilization by **~3x**
    - Towards **100%** in many cases

```
loop:                         scfg 0, %[a], ldA
fld r0, %[a]                  scfg 1, %[b], ldB
fld r1, %[b]       ──────→    loop:
fmadd r2, r0, r1              fmadd r2, ssr0, ssr1
```

- Extension around the core's register file
    - Subset of registers have stream semantics
    - Accesses routed out of the core
- Address generation hardware
    - Assigns affine addresses to accesses
    - Up to 4 nesting levels
- SSRs ≠ memory operands
    - Perfect prefetching, highly latency-tolerant



Floating-Point Repetition (Xfrep)

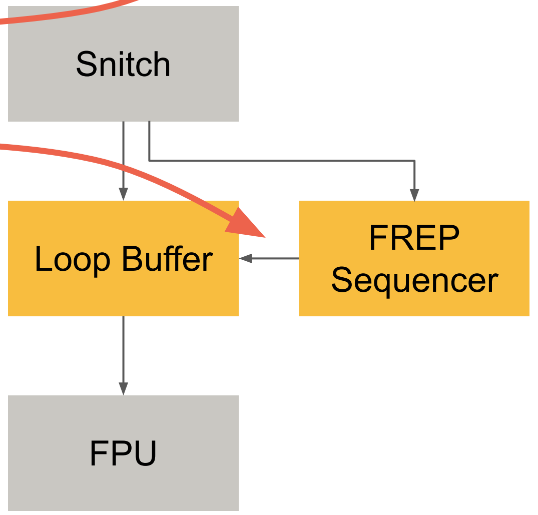
- Programmable micro-loop buffer
- Allows offloading of inner loop bodies

- Custom instruction indicates start of hardware loop block
- Sequencer steps through the buffer, issues instructions to the FPU

- Integer core operates in parallel: **Pseudo-dual Issue**
- Synchronization on FPU ⇄ INT ops
- RISC-V floating-point ISA makes this easy

```
  mv    r0, zero
loop:
  addi  r0, 1
  fmadd r2, ssr0, ssr1
  bne   r0, r1, loop
```

```
  frep  r1, 1
loop:
  fmadd r2, ssr0, ssr1
```



Typical SSR/FREP Execution

- SSRs enable float-only hardware loops
- FREP marks loop
- Example: Reduction Operation
  - E.g. matmul, stencils, convolution
  - Unrolled four-fold

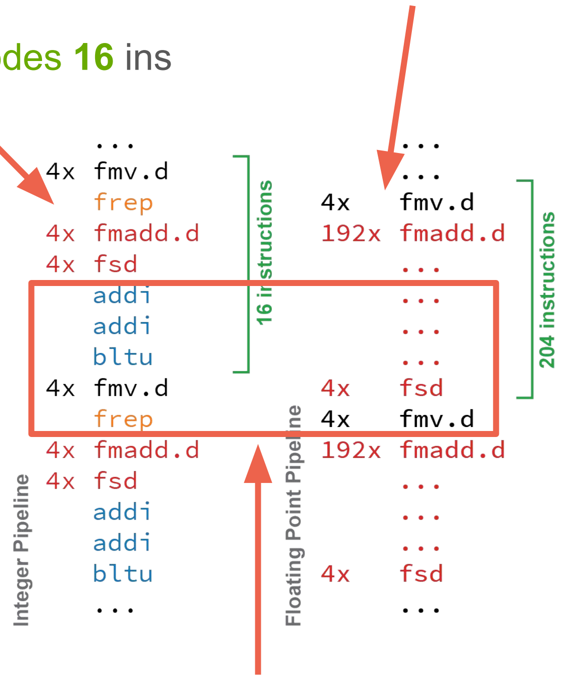
```
constexpr int N = 48;
for (int i = 0; i < N; i++) {
  double a = 0.0;
  for (int j = 0; j < N; j++) {
    a += A[i][j] * B[i][j];
  }
  C[i] = a;
}
```

```
loop:
  fmv.d     fa5,fa1
  fmv.d     fa2,fa1
  fmv.d     fa3,fa1
  fmv.d     fa4,fa1
  frep      t0, 4
  fmadd.d   fa5, ft0, ft1, fa5
  fmadd.d   fa2, ft0, ft1, fa2
  fmadd.d   fa3, ft0, ft1, fa3
  fmadd.d   fa4, ft0, ft1, fa4
  fsd       fa5,0(a5)
  fsd       fa2,8(a5)
  fsd       fa3,16(a5)
  fsd       fa4,24(a5)
  addi      a4, a4, 4
  addi      a5, a5, 32
  bltu      a4, a1, loop
```

16 instructions

FPU executes **204** ins

Core fetches/decodes **16** ins



Bookkeeping ops in integer ALU **overlaps** with FPU operation

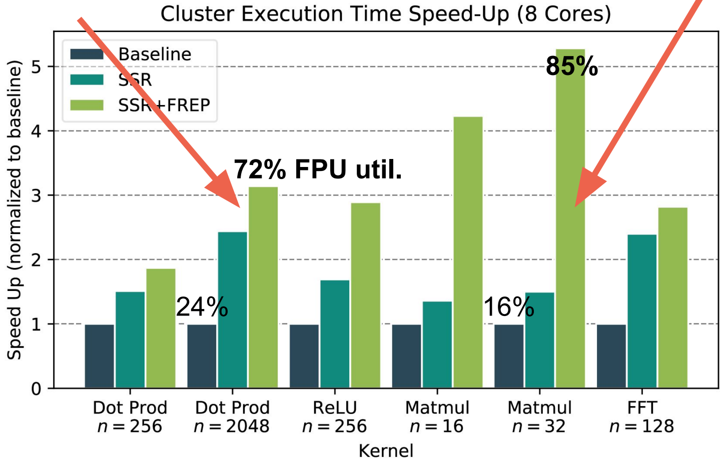
The Benefit of SSRs and FREP

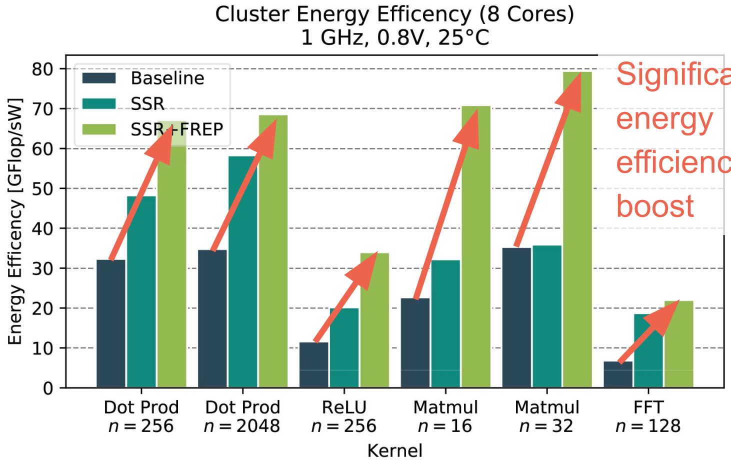
- A single-issue core can completely saturate an FPU
- **IPC > 1**, few inst. fetched, many executed
- **Reduces** the von Neumann bottleneck

- Integer and float **pseudo-dual-issue**
- Single-cluster energy efficiency of **80 DP-Gflop/sW**, **>5x** speedup
- SSR and FREP are also applicable to less regular problems, such as FFT or sorting

Even mem-bound dot product benefits

>5x speedup for matmul





Significant energy efficiency boost

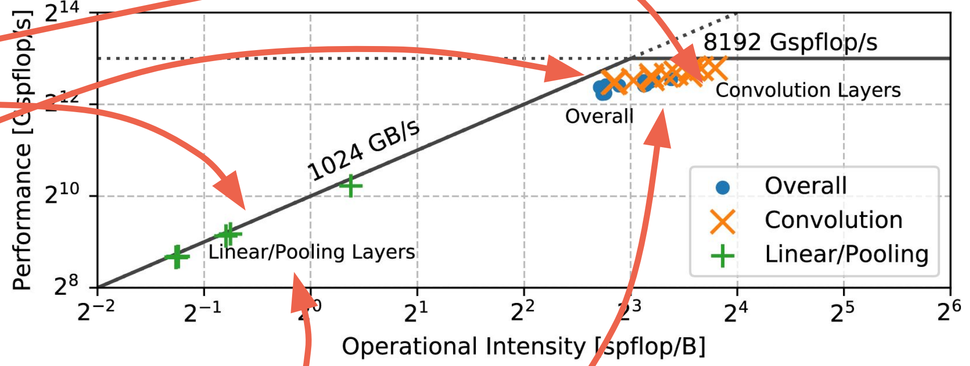
Performance Roofline

- Workloads from a **DNN training step**
  - Convolution-only (compute-bound)
  - Linear/pooling-only (memory-bound)
  - Full mix of kernels (conv.-dominated)

- Estimated based on:
  - **Cycle-accurate** hardware simulation
  - **Architectural model** of full system
  - **Silicon measurements** of prototype system

- **>80%** of peak bandwidth for memory-bound kernels
- **>90%** of peak performance for compute-bound kernels

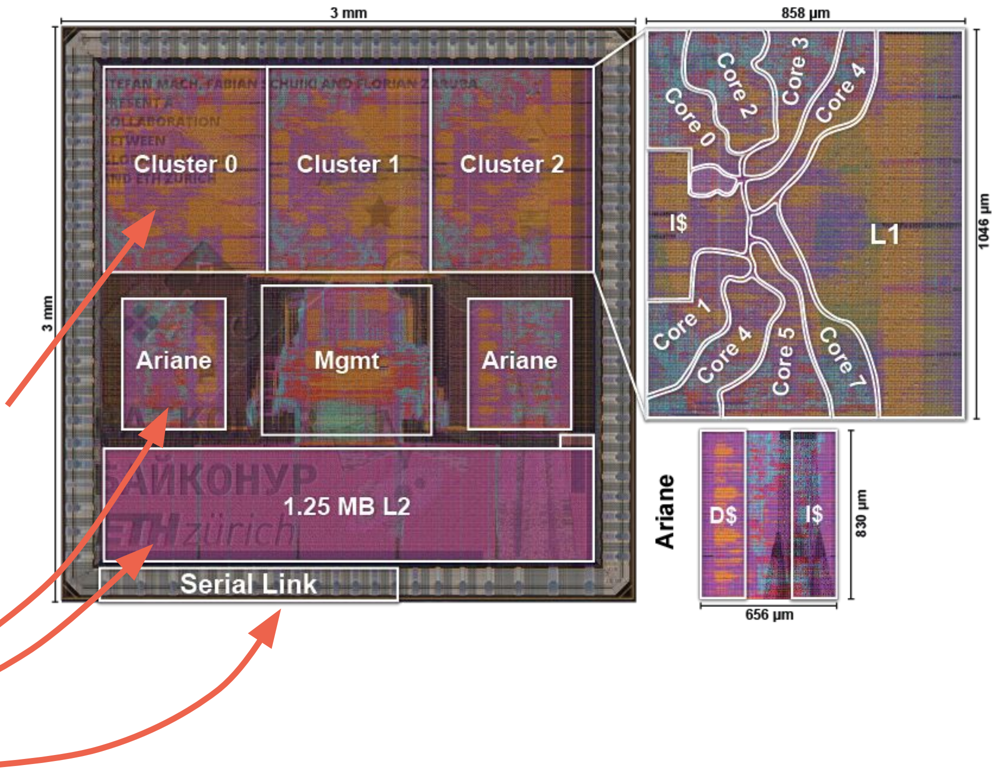
- **Close tracking of the roofline**



Silicon Prototype

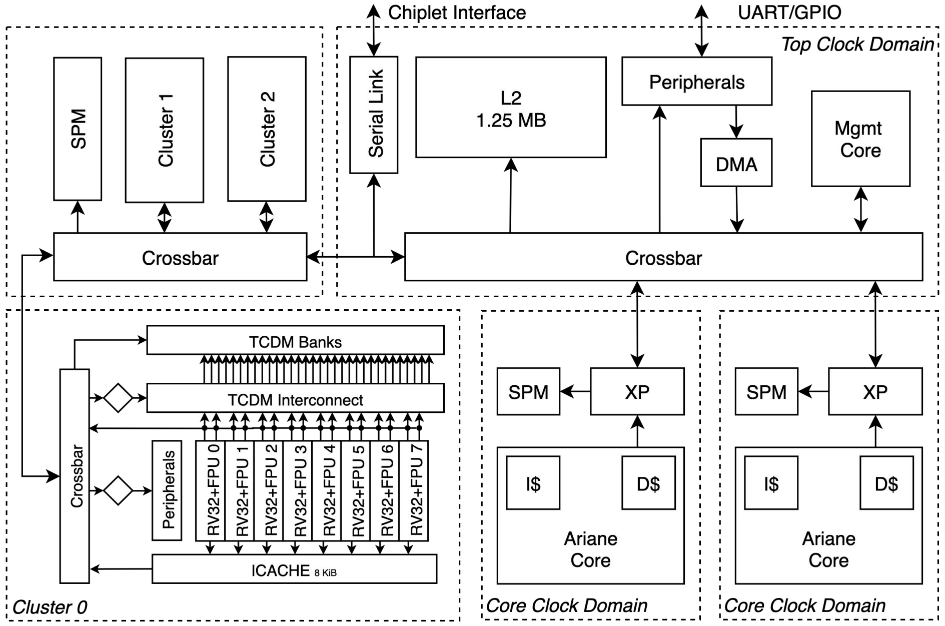
Floorplan

- 9 mm^2 prototype of chiplet architecture
- Manufactured in **22nm**
  - Globalfoundries 22FDX
  - Evaluates different standard cell flavors and threshold voltages
  - Forward Body Biasing
- Testbed for key architecture components
- Number-crunching **Snitch** cores (RV32G)
- Clusters laid out around memory system:
  - Cores arranged in star shape around Tightly Coupled L1 Data Memory
  - Instruction frontends close to L1 I-cache
- Application-class **Ariane** cores (RV64G)
- 1.25 MB of L2 memory
- Serial link tightly coupled to pad frame



Architecture

- 3 Octa-Core Snitch Clusters (RV32G)
  - 8 kB L1 instruction cache
  - 128 kB L1 data memory (in 32 banks)
  - 8 Snitch cores
  - 8 multi-format FPUs
  - ➢ Good multi-core speed-ups
  - ➢ Suitable unit for “hardening” as a macro
- 2 Ariane Cores (RV64G)
  - 16/32 kB L1 instruction/data cache
  - 8 kB local scratchpad memory
  - ➢ Linux-capable, controller core
- 1 Snitch Management Core
- 800 MHz / 2.56 Gbit/s all-digital chip-to-chip link prototype
  - ➢ portable, “easy” to bring-up
- DMA Unit / Peripherals / 4 FLLs



Silicon Performance

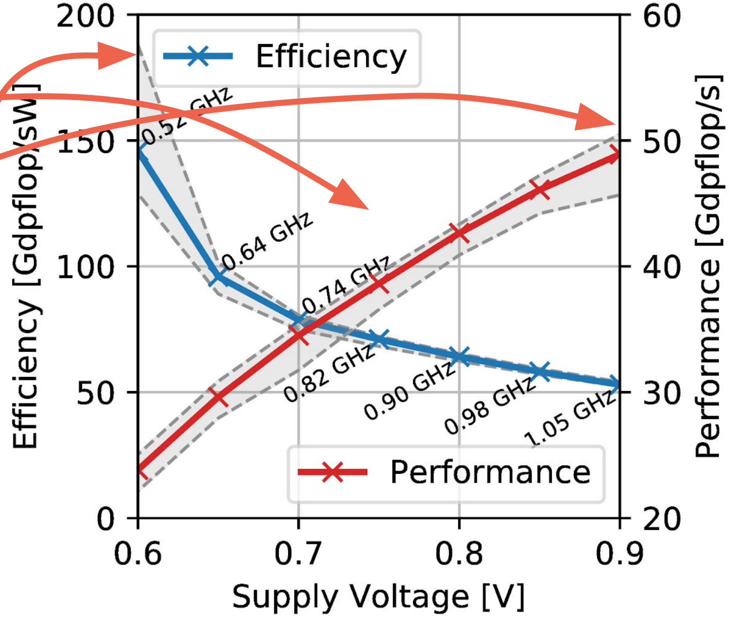
Silicon Performance / Efficiency

- **>90%** FPU utilization thanks to Snitch cores with Xfrep/Xssr

- Wide range of operating points
- Choice of performance/efficiency tradeoff

- On prototype up to **54 DP-Gflop/s** across 24 cores
- On full Manticore system **>27 DP-Tflop/s** across 4096 cores
- Both in **22nm** (**GF 22FDX**)

- Compute efficiency up to **188 DP-Gflop/sW**
- Compute density up to **20 DP-Gflop/smm^2**

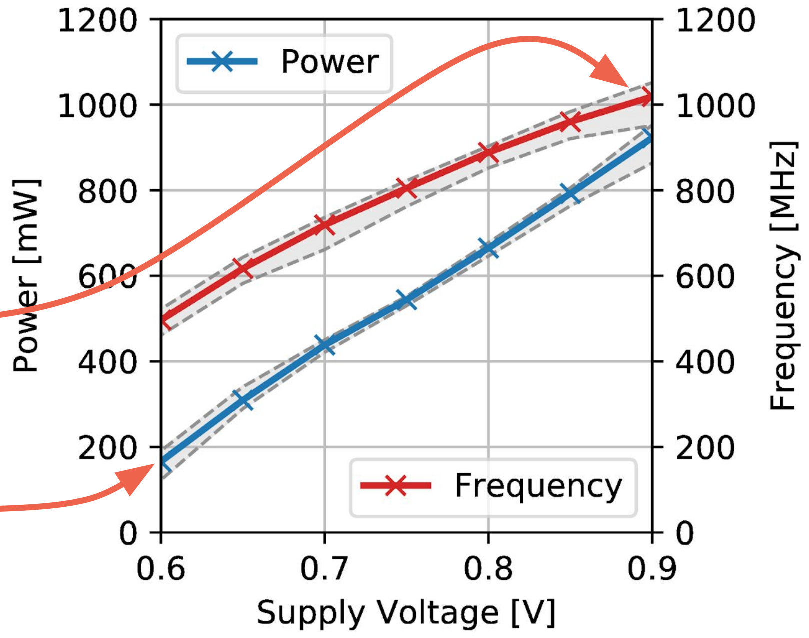


Silicon Power / Speed

- Silicon supports wide power/frequency envelope
- DVFS based on operational intensity of current workload
- Adjust roofline to match:

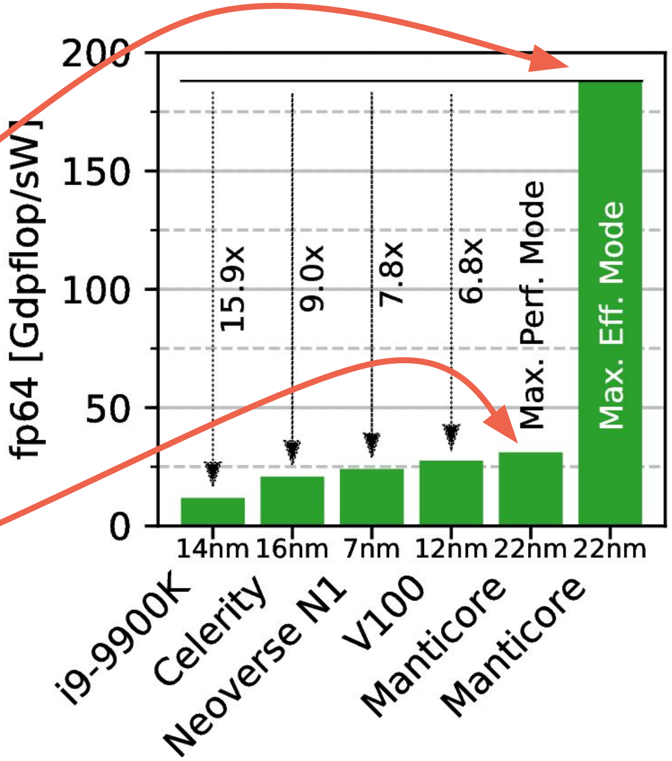
- High-performance mode: **>54 DP-Gflop/s**
  - **0.9 V** supply
  - **>1 GHz** sustained compute

- High-efficiency mode: **>188 DP-Gflop/sW**
  - **0.6 V** supply
  - **0.5 GHz** sustained compute



Efficiency on fp64

- Industry-leading fp64 efficiency
- Assuming 90% of peak performance sustainable for competing architectures
- In maximum-efficiency mode:
  - **15x** more efficient than i9-9900K
  - **9x** more efficient than Celerity (RISC-V)
  - **7x** more efficient than N1
  - **6x** more efficient than V100
  - **5x** more efficient than A100[1]
- In maximum-performance mode:
  - **Competitive** with N1 / V100 / A100[1]



[1] Preliminary results for Ampere A100 based on whitepaper

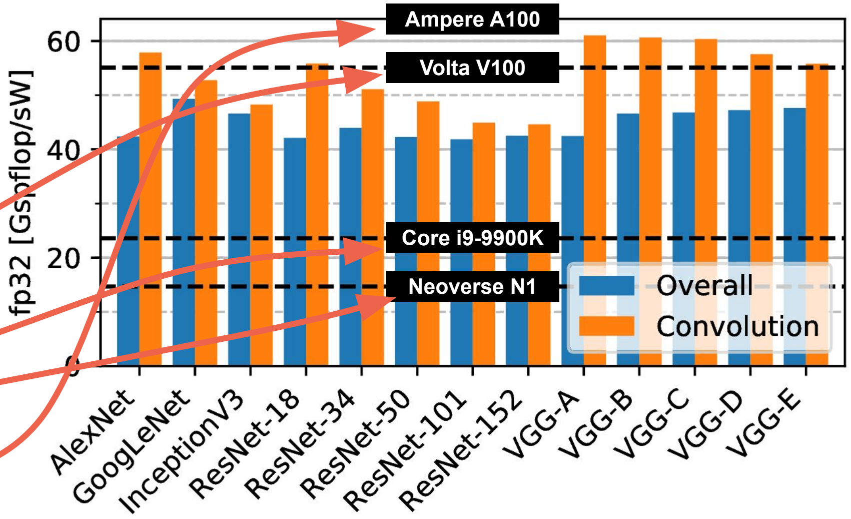
Efficiency on fp32

Manticore:	22nm FD-SOI @0.6V
Volta V100:	12nm FinFET @1V
Volta A100:	7nm FinFET @1V
Core i9-9900K:	14nm FinFET @1V
Neoverse N1:	7nm FinFET @1V

- Competitive fp32 efficiency
- Assuming 90% of peak performance sustainable for competing architectures

- Comparison on DNN Training workload
  - Convolution-only (highly compute-bound)
  - Full layer mix (intermittently memory-bound)

- **Competitive** with V100
- **>2x** better efficiency than i9-9900K
- **>3x** better efficiency than N1
- **25%** lower than A100[1] (but 22nm vs 7nm)



[1] Preliminary results for Ampere A100 based on whitepaper

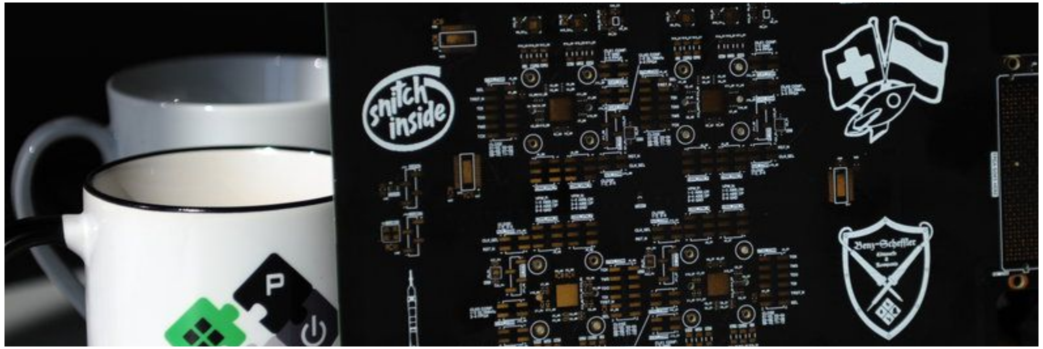
In Closing

- Small core, large FPU
- SSR/FREP combats von Neumann bottleneck
- Extreme efficiency on fp64
- Competitive on fp32

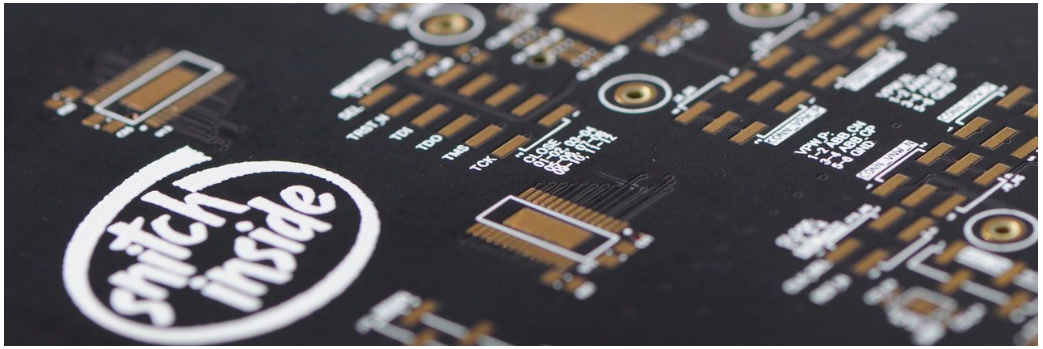
Next steps:

- “Quad-chiplet” prototype board
- Larger prototype silicon (FinFET)
- Looking for industrial partners for D2D/HBM/DDR PHY integration

Thanks!

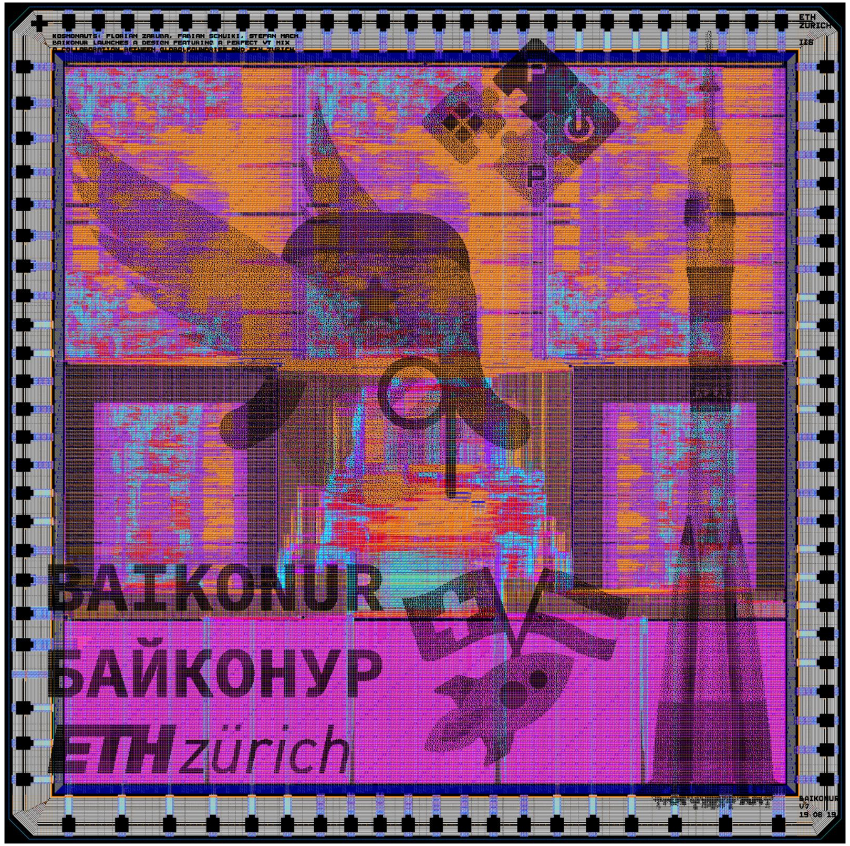


@pulp_platform / github.com/pulp-platform / pulp-platform.org



Physical Design

- Manufactured in Globalfoundries 22FDX
  - 10-metal ULP SLVT/LVT
- Components laid out as hard macros
- Allows us to mix:
  - Different poly-silicon pitches
  - Different routing grids
  - Different threshold voltages
- Evaluates different standard cell flavors and threshold voltages
  - 7.5T vs. 8T
- Forward Body Biasing
  - ➢ Speed improvements
- Evaluated on industry-grade silicon testing equipment at ETH Zurich


BAIKONUR
БАЙКОНУР
ETH zürich