Gemmini: Enabling Systematic Deep-Learning Architecture Evaluation via Full-Stack Integration

DNNs are exploding in popularity...
Which means DNN ACCELERATORS are exploding in popularity...
Which means DNN accelerator GENERATORS are exploding in popularity...

However, they lack full-system and full-stack visibility
Full-System Visibility
Full-System Visibility: SoC

Performance Impacts:
Resource contention, etc.

Shared L2 Cache

IOs, Interconnects, etc.
Full-System Visibility: Memory Hierarchy

Performance Impacts
Cache coherence, miss rates/latencies, etc.
Full-System Visibility: Virtual Addresses

Performance Impacts
Page faults, TLB hits, etc.
Performance Impacts
Unaccelerated kernels, etc.
Full-System Visibility: Operating System

Performance Impacts
Interrupts, context switches, etc.
Full-Stack Visibility

- ONNX
- TensorFlow
- cuDNN
- cuBLAS

High

Medium

Low

Direct hardware configuration, low-level ISA
Gemmini

- DNN accelerator generator
- Flexible hardware template
- Full-stack
- Full-system
Gemmini: Spatial Array

- Parameters:
  - Dataflow
  - Dimensions
  - Pipelining
Gemmini: Spatial Array

- Parameters:
  - Dataflow
  - Dimensions
  - Pipelining
Gemmini: Spatial Array

- Parameters:
  - Dataflow
  - Dimensions
  - Pipelining
Gemmini: Non-GEMM Functionality

• Can be optimized out at elaboration-time
Gemmini: Local Scratchpad

- Parameters:
  - Capacity
  - Banks
  - Single- or dual-port
Gemmini: Global Memory

- Parameters:
  - Capacity
  - Banks
  - DRAM controller
Gemmini: Host CPU

- Parameters:
  - In-order/out-of-order
  - ROB capacity
  - L1 capacity
  - Branch predictor
Gemmini: Virtual Address Translation

- Parameters:
  - TLB capacity
  - TLB hierarchy
    - e.g. L2 TLB
Gemmini: Full SoC

SoC

Gemmini 1
- Spatial Array
- Controller
- Scratchpad
- DMA
- + Accumulator

CPU 1
- TLB
- RF
- Private L1$
- PTW

CPU 2
- TLB
- RF
- Private L1$
- PTW

Gemmini 2
- Spatial Array
- Controller
- Scratchpad
- DMA
- + Accumulator

Shared L2 Cache

IOs, Interconnects, etc.
Gemmini: Programming Model

- **High**
  - Direct hardware configuration, low-level ISA
  - Hand-tuned C library for DNNs
- **Medium**
  - configure_loads(...); configure_stores(...); preload_spatial_array(...); feed_spatial_array(...)
- **Low**
  - matmul(...); conv(...); residual_add(...); max_pool(...); global_averaging(...)
Performance: Evaluating Host CPUs

• “Im2col” runs on CPU, matmuls run on Gemmini

![Graph showing speedup for different models and CPU configurations]
Performance: Evaluating Optional Functional Units

- “Im2col” and matmuls both run on Gemmini

![Bar chart showing speedup for different models and configurations](chart.png)
Performance: Overall

- **DNNs:**
  - ResNet50: 22.8 FPS
  - AlexNet: 79.3 FPS
  - MobileNet: 18.7 FPS
  - BERT: 167x speedup

- **About 50% as fast as NVDLA**
Performance: Overall

- DNNs:
  - ResNet50: 22.8 FPS
  - AlexNet: 79.3 FPS
  - MobileNet: 18.7 FPS
  - BERT: 167x speedup

  - About 50% as fast as NVDLA

Performance Updates
ResNet50: 40.3 FPS
About 80% as fast as NVDLA
How Does the Full System and Full Stack Affect Performance?
Case Study: Virtual Memory for DNNs

Two-level TLB hierarchy

- Private Accelerator TLB
- Private CPU TLB
- Shared L2 TLB

TLB Misses for ResNet50

Graph showing the TLB miss rate over the last 1 million cycles.
Case Study: Virtual Memory for DNNs

- Small private TLB much more impactful
Case Study: Virtual Memory for DNNs

- Small private TLB much more impactful
- Low-cost optimizations:
  - Single-entry L0 TLB filters out consecutive TLB requests to same page

![Without L0 TLB](chart1)

![With L0 TLB](chart2)
Case Study: Memory Partitioning

SoC

Gemmini 1
- Spatial Array
- Controller
- DMA
- Scratchpad
- Accumulator

CPU 1
- TLB
- RF
- Private L1$
- PTW

CPU 2
- TLB
- RF
- Private L1$
- PTW

Gemmini 2
- Spatial Array
- Controller
- DMA
- Scratchpad
- Accumulator

Shared L2 Cache

IOs, Interconnects, etc.
Case Study: Memory Partitioning

- Single core
  - Private scratchpad more helpful
  - Much better for convs
Case Study: Memory Partitioning

- Single core
  - Private scratchpad more helpful
  - Much better for convs

- Dual core
  - Shared L2 more helpful
  - Much better for residual additions
Conclusion

• Gemmini is:
  • Full-system
  • Full-stack

• Enables DSE and hardware/software co-design
  • Layer composition vs. memory partitioning
  • Virtual address translation design

• Open-source!
  • github.com/ucb-bar/gemmini