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

Quick Stats

• Best Paper at DAC 2021
• Tutorial at IISWC 2021
• DNN accelerator generator
  • Full-system, full-stack visibility
• Taped out
  • Intel22FL
    • ESSCIRC 2021
  • GF12
  • TSMC16
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.
Full-System Visibility: Host CPUs

Performance Impacts
Unaccelerated kernels, etc.
Full-System Visibility: Operating System

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

- ONNX
- TensorFlow
- Direct hardware configuration, low-level ISA
- cuDNN
- cuBLAS

High
Medium
Low
Gemmini

- DNN accelerator generator
  - RTL
  - Simulations
  - Runs Linux
- Flexible hardware template
- Full-stack
- Full-system
Gemmini: Spatial Array

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

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

- Parameters:
  - Dataflow
  - Datatypes
  - 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: Programming Model

ONNX

High

Medium

Low

matmul(...); conv(...); residual_add(...);
max_pool(...); global_averaging(...)

configure_loads(...); configure_stores(...)
preload_spatial_array(...); feed_spatial_array(...)

Hand-tuned C library for DNNs

Direct hardware configuration, low-level ISA
Performance: Evaluating Host CPUs

• “Im2col” runs on CPU, matmuls run on Gemmini
Performance: Evaluating Optional Functional Units

• “Im2col” and matmuls both run on Gemmini

![Speedup (log scale)]

- Small In-Order CPU + Gemmini with On-The-Fly Im2Col
- Large OoO CPU + Gemmini with On-The-Fly Im2Col
Performance: Overall

- DNNs:
  - ResNet50: 40.3 FPS
  - AlexNet: 79.3 FPS
  - MobileNet: 18.7 FPS
  - BERT: 167x speedup

- About 80% as fast as NVDLA
How Does the Full System and Full Stack Affect Performance?
Case Study: How Does Virtual Memory Affect DNN Accelerator 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 TLB miss rate over 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
Case Study: Memory Partitioning Schemes for Multi-Accelerator SoCs
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

Funded by DARPA RTML program
(contract FA8650-20-2-7006)