groq™

ATPESC 2021

Reinventing Chips for AI

Dennis Abts
Groq Chief Architect & Fellow in Engineering
dabts@groq.com

Agenda

- Introduction
- Architecture
  Overview, Dataflow, Compute
- Impact
  Scaling, Determinism, Batch 1 performance, TCO
- Conclusions

Introducing Groq

Efficiently accelerating AI/ML and converged HPC

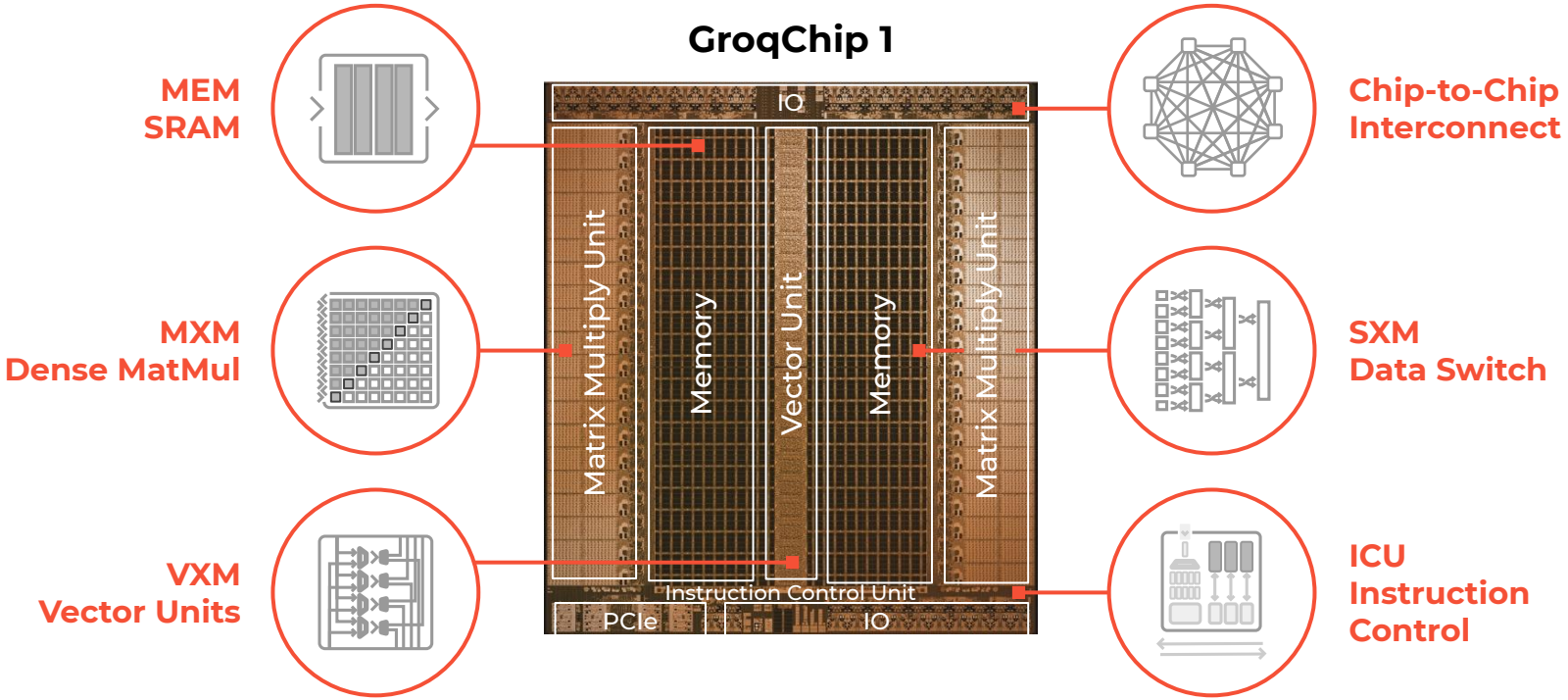
- Founded: 2016
- Headquarters: Mountain View, CA
- Flagship Product: GroqChip™ accelerator & Compiler for use in AI, Machine & Deep Learning, and converged HPC
- Availability: First generation chip shipping
- ML Systems at Groq: Implementation of ML models, optimization of ML models, & performance analysis of workloads on Groq hardware

ARCHITECTURE

Streams, Memory, ALUs, & ICUs

GroqChip™ at a Glance

Scalable Compute Architecture



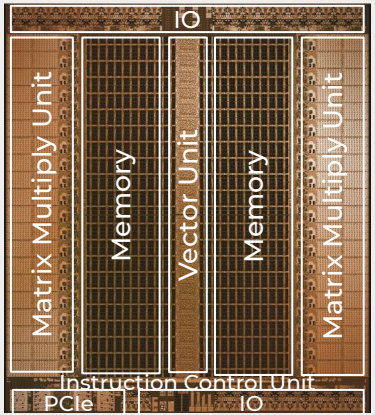
GroqChip 1 Overview

Scalable Compute Architecture

SRAM Memory
Massive concurrency
80 TB/s of BW
Stride insensitive

Groq TruePoint™ Matrix
Engines x4
320x320 fused dot product
Integer and floating point

Programmable Vector Units
5,120 Vector ALUs* for high performance



GroqChip 1

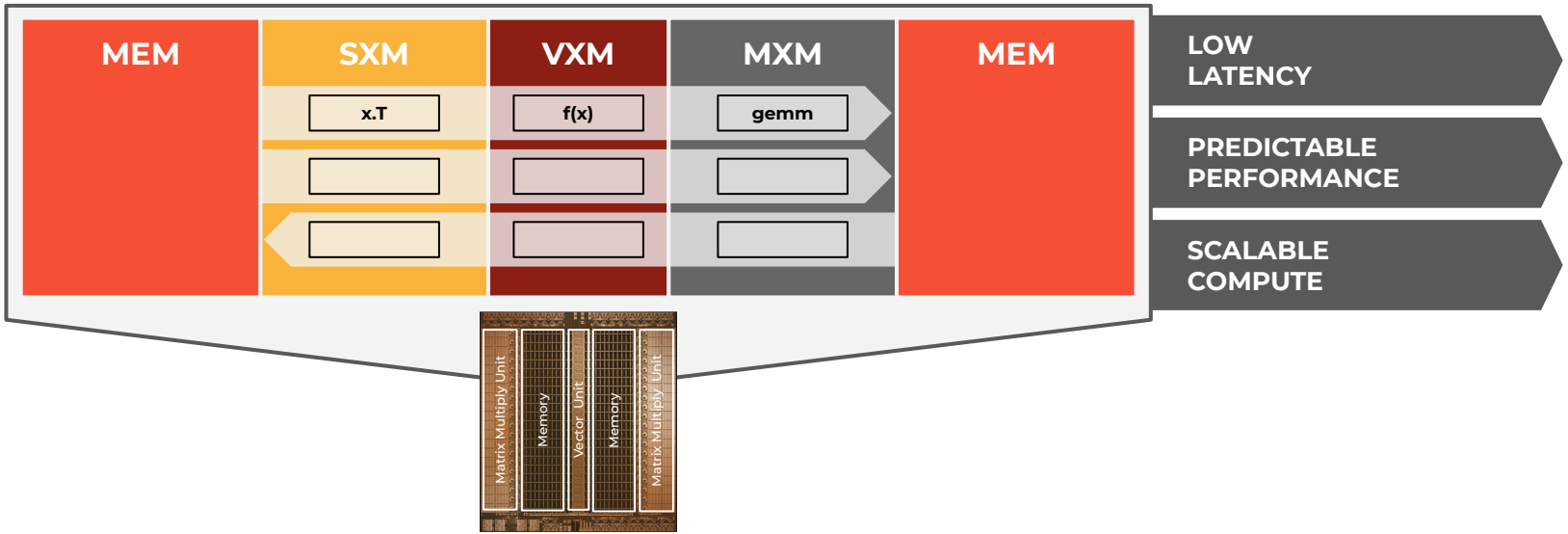
Networking
480 GB/s bandwidth
Extensible network scalability
Multiple topologies

Data Switch
Shift, Transpose, Permuter for improved data movement and data reshapes

Instruction Control
Multiple instruction queues for instruction parallelism

Tensor Streaming Dataflow

A simply efficient approach to compute architecture and data flow



Single core, spatial pipeline processing

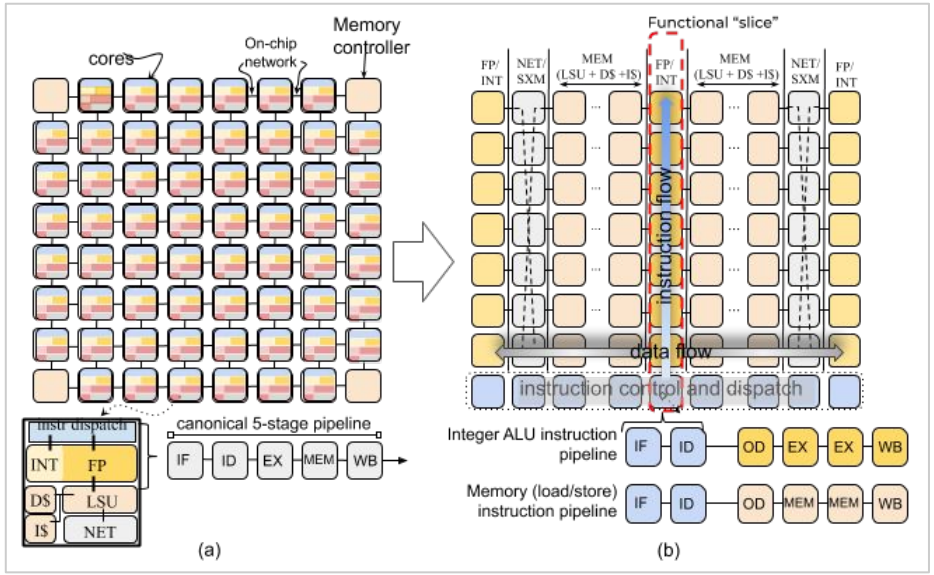
Simple tensor instruction set architecture

Stream programming of massive SIMD, concurrent streams

Large on-chip memory bandwidth

Deterministic, predictable performance scales to multi-chip

Disaggregated Compute & Control



Disaggregates Functional Units

- Processing elements arranged into "slices"
- Matrix, Vector, Switching (MXM, VXM, SXM)
- Exploit data and instruction parallelism
- Contrast to a many-core arch

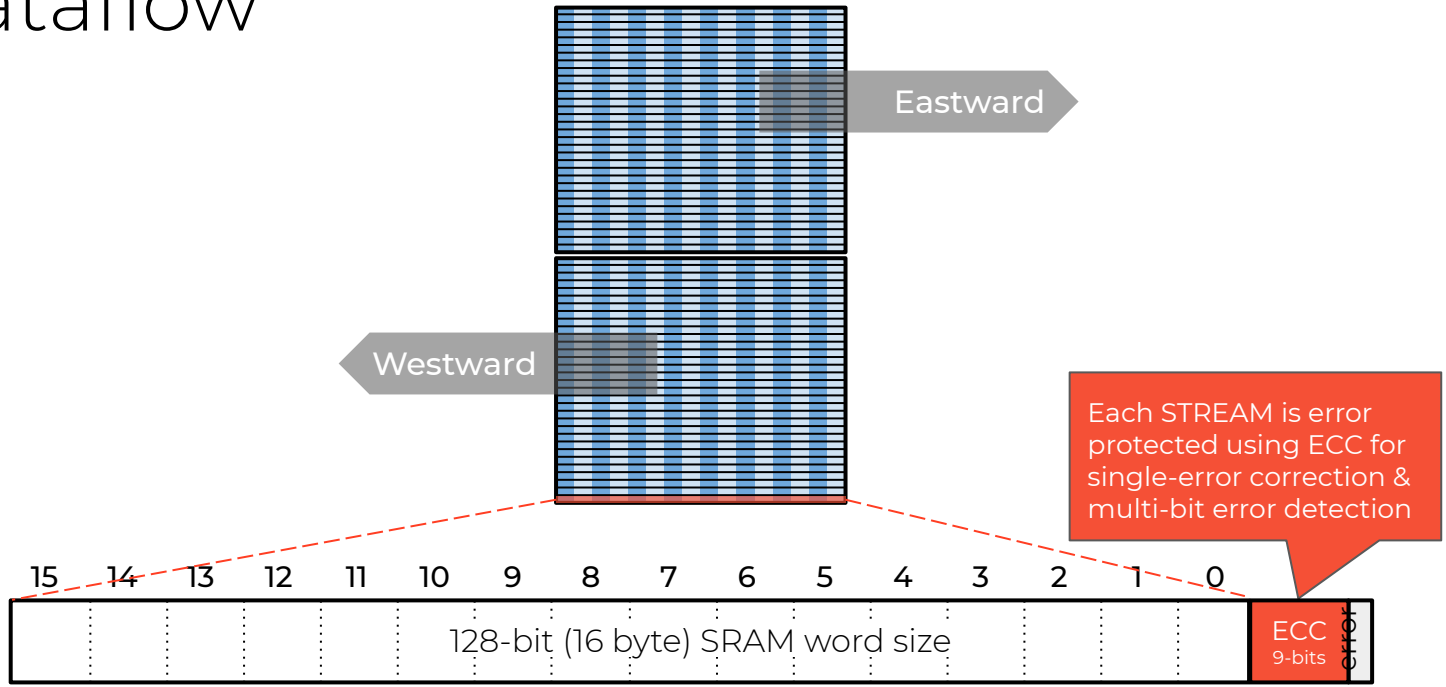
Data-Level Parallelism

- Instructions operate on 320 byte SIMD tensor while streaming

Instruction Level Parallelism (ILP)

- Up to 144-way from slices

Dataflow



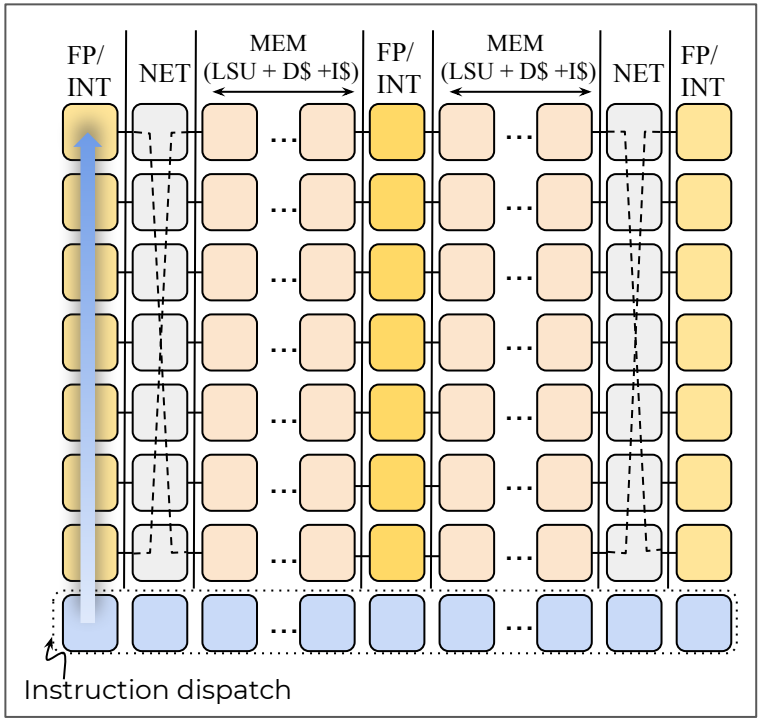
Streams composed of "superlanes" (one "row of the stream)

20 superlanes: Form a max vector length of 20x16=320 elements

Streams provide interface similar to a load/store architecture

All memory read/writes are produce/consumed to/from the streams

Compute



Executed in a SIMD manner

Same instruction executed on all 20 superlanes

Staggered in time

Instructions flow vertically

One superlane per cycle

20 cycles to operate on a vector

Compute, Memory, & IO

Compute

Matmult TOPs: INT8

Hemispheres * Planes/Hemisphere * (2*VL + 1) * MXM width * clock

2 * 2 * (2*320 + 1) * 320 * fclk

= 1024 TOps

(1 Peta Op per Second @1.25GHz)

Memory

Memory = Slices/hemisphere * Hemispheres * Addresses/Bank * Banks/slice * Bytes/address

44*2*4096*2*320 = 220MB

Memory-Compute Units BW

BW = Number of streams in parallel in both directions * Bytes/stream * Clock * hemispheres * Computation units

= 64 * 320 *900 MHz * 2 * 2 = 80TB/s

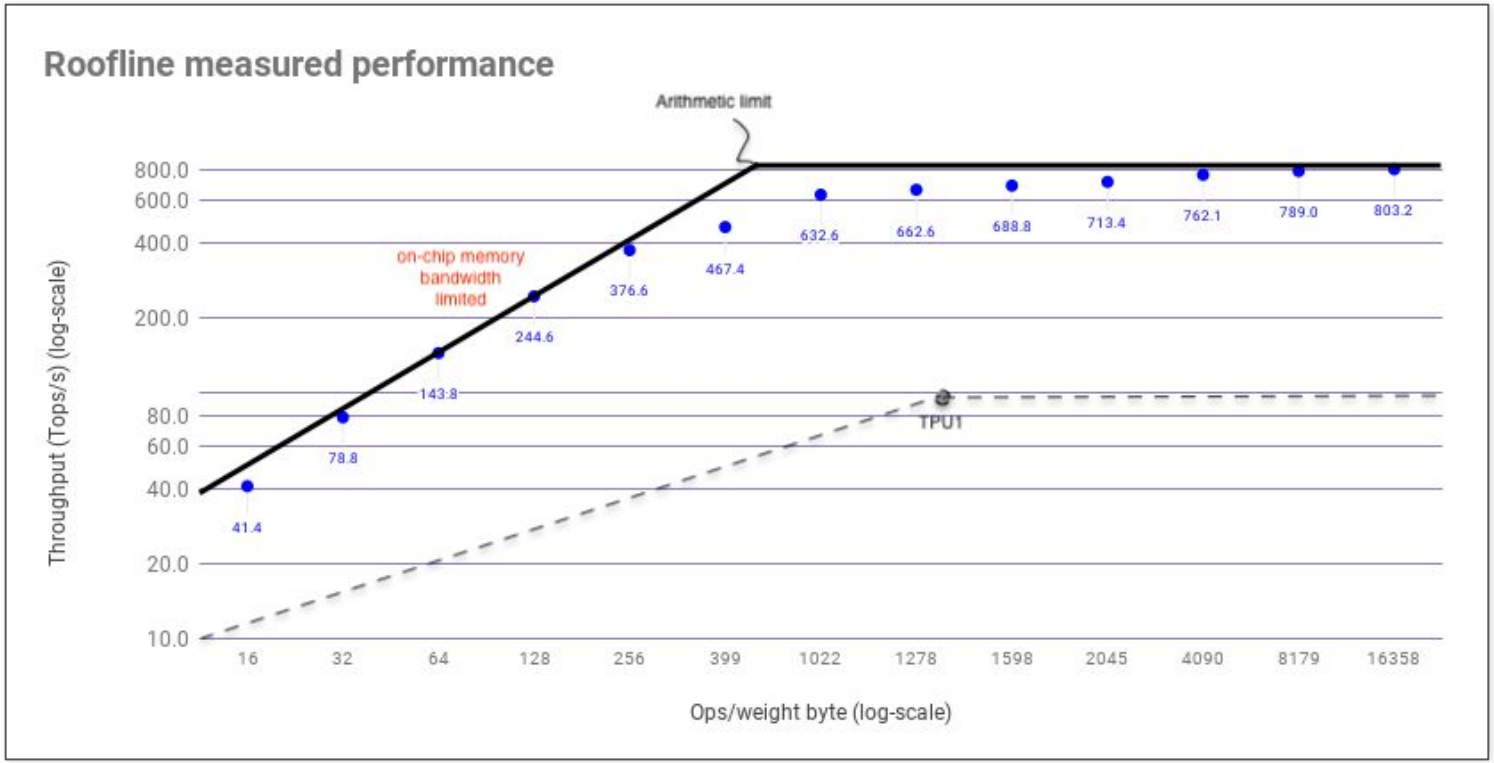
IO BW

PCI Gen 4: x16: 32GB/s

C2C BW

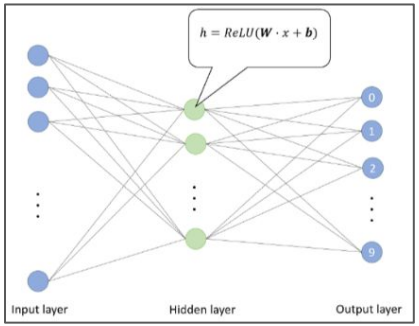
16 x4 links @ 30Gb/s x 2 (bidi) = 3.84 Tb/s =480 GB/s of total off chip BW

Roofline


Roofline measured performance
Arithmetic limit
on-chip memory bandwidth limited
TPU1
Throughput (Tops/s) (log-scale)
800.0
600.0
400.0
200.0
80.0
60.0
40.0
20.0
10.0
41.4
78.8
143.8
244.6
376.6
467.4
632.6
662.6
688.8
713.4
762.1
789.0
803.2
16
32
64
128
256
399
1022
1278
1598
2045
4090
8179
16358
Ops/weight byte (log-scale)

VXM Chain Example

Simple Dense Layer with Activation



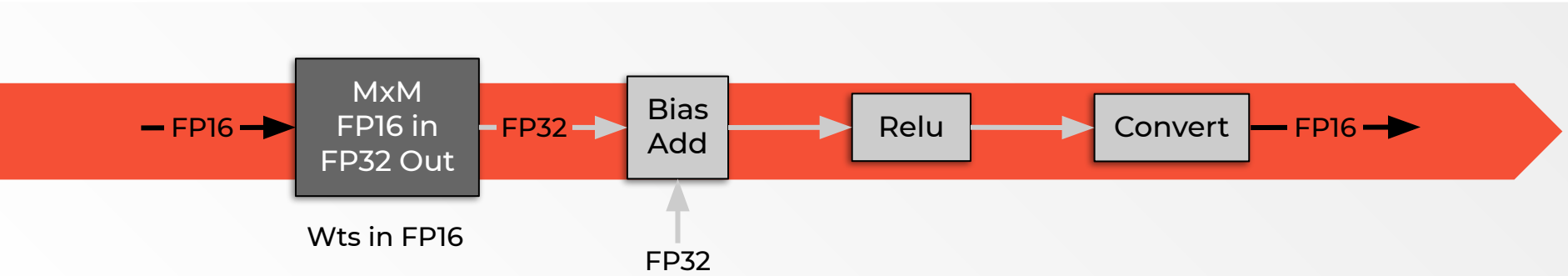
Would like to keep data in flight as long as possible without store to memory

MatMul returns partial products

Final accumulation done in the SXM or VXM

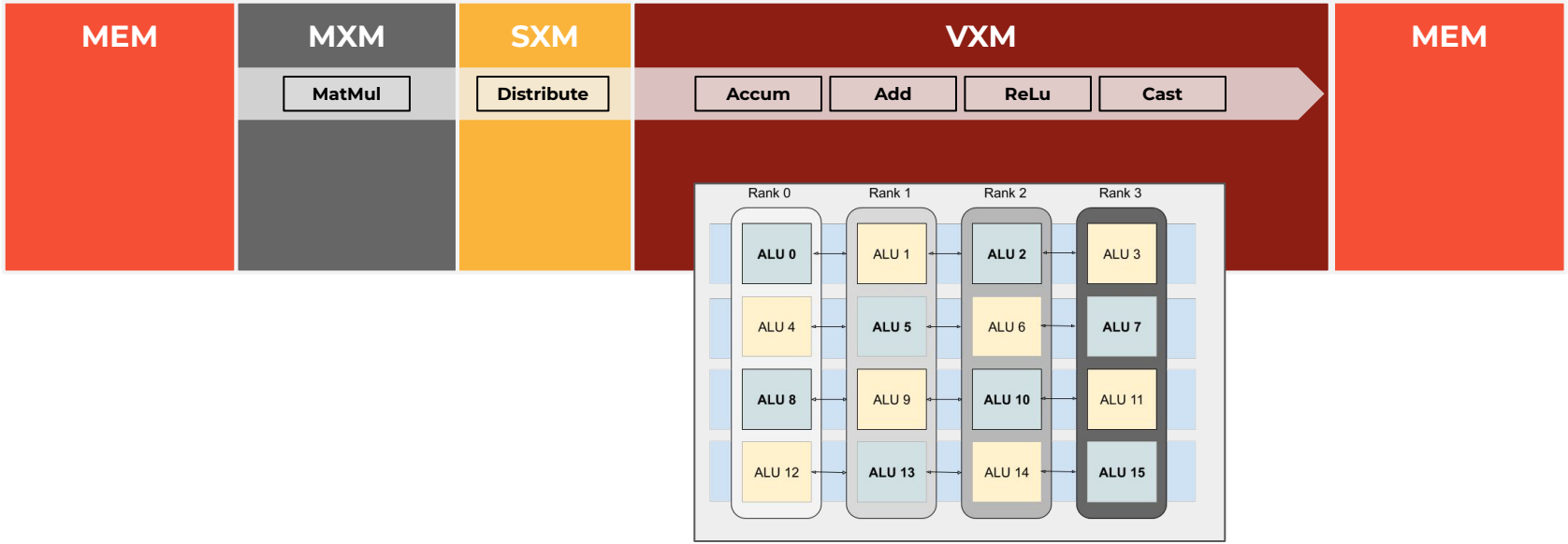
Continue chaining element-wise operations inside the VXM

Bias-Add, Relu, Re-Quantize can all be chained in the VXM



A Tensor Streaming Architecture

VXM Chaining for a Dense Linear Layer with Bias Add, Relu, and Quantize



Dataflow begins with memory Read onto Stream Tensor

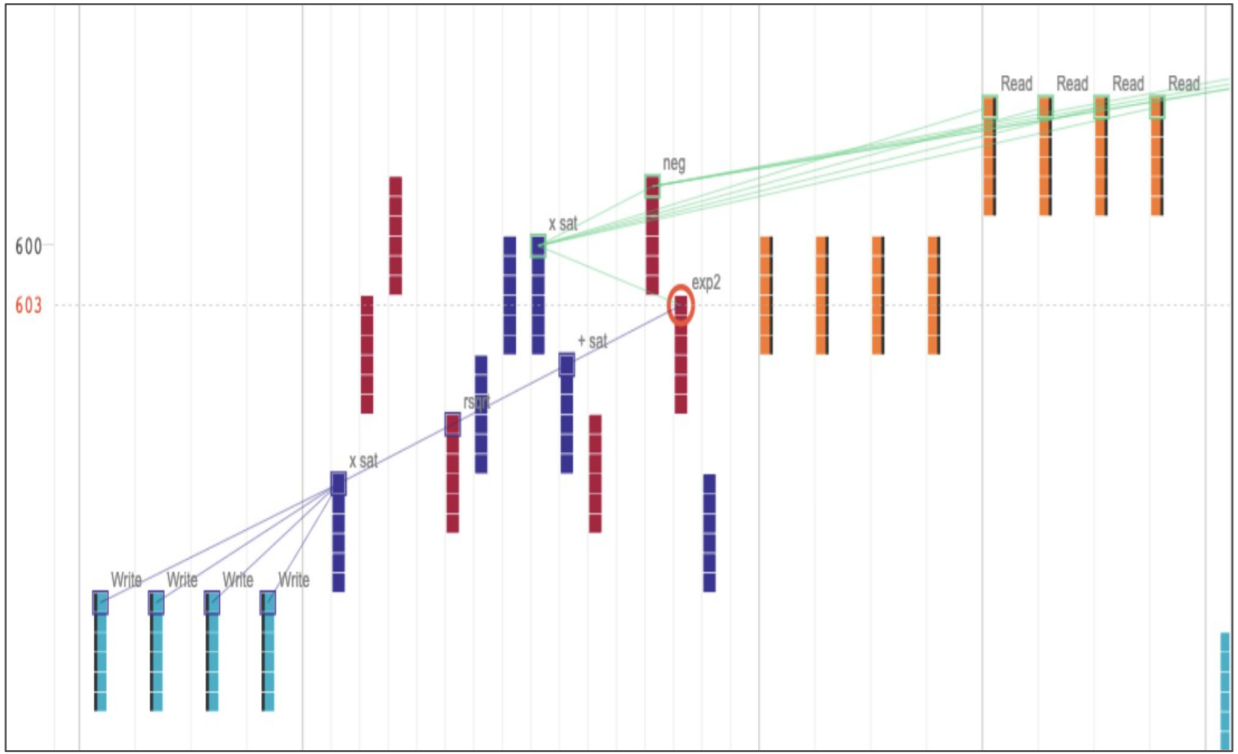
Many concurrent streams are supported in programming model

VXM provides a flexible and programmable fabric for Compute

Compute occurs on data locality of passing Stream Tensor

MEM bandwidth supports high concurrency

Visualizing Program “Schedule” on the Chip



Compute activity over time

Stream flow

Data concurrency

Performance & occupancy

Software Development at Groq

Compiler
Front-end optimization, vectorizing rewrites, and scheduling. Model parallel compilation flows and model segmentation

Assembler
Binary program generation and hardware abstractions

DNN Library
DNN llbrary development and packages using direct-to-architecture programming interface

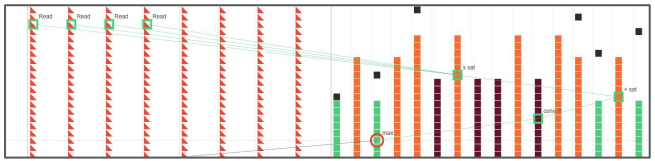
Data Scientist
ML exploration and algorithmic mapping onto GroqChip™ using Groq SDK and Tools

Tools
Visualization, user reports, and virtualization

onnx/**onnx-mlir**

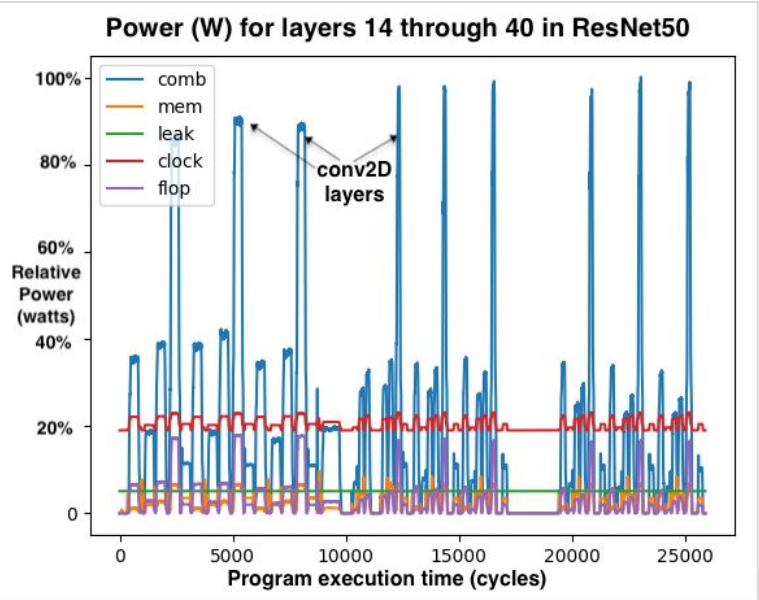
Representation and Reference Lowering of ONNX Models in MLIR Compiler Infrastructure

MLIR



elm

Compiler: Fine Grained Control



Compiler

Schedules functional units by controlling each instruction queue in the ICUs

Compiler inserts instruction prefetching instructions

Keep all 144 instruction queues fed

A functional slice never stalls to fetch more instructions.

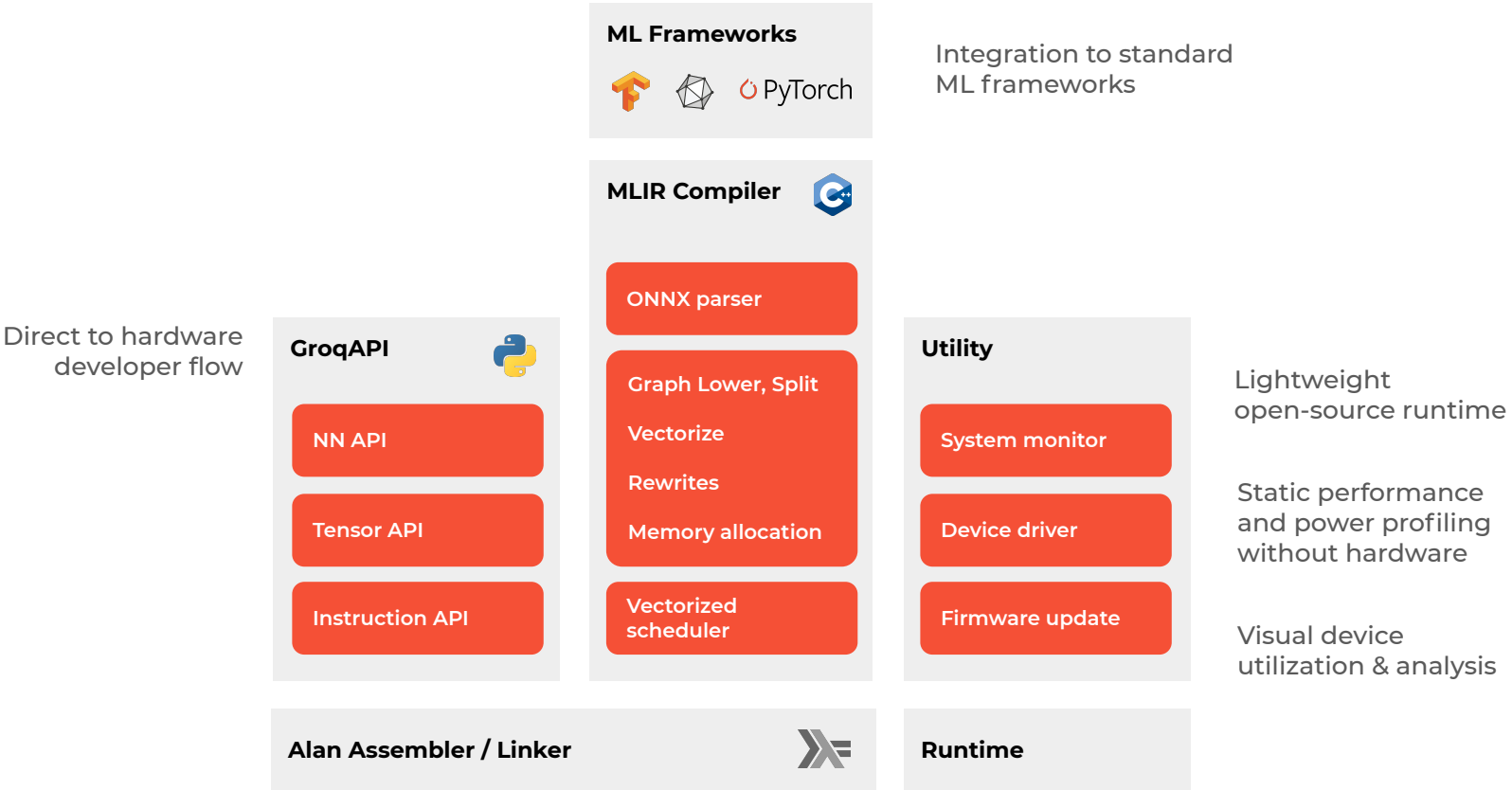
Compiler can trade-off latency & throughput for power savings

Add NOPs to “throttle” performance

Do it on a per functional unit basis

Measure performance at compile-time.

GroqWare™


ML Frameworks
PyTorch
Integration to standard ML frameworks
MLIR Compiler
ONNX parser
Graph Lower, Split
Vectorize
Rewrites
Memory allocation
Vectorized scheduler
Direct to hardware developer flow
GroqAPI
NN API
Tensor API
Instruction API
Utility
System monitor
Device driver
Firmware update
Lightweight open-source runtime
Static performance and power profiling without hardware
Visual device utilization & analysis
Alan Assembler / Linker
Runtime

Groq Delivers Greater Integration

Scalable compute architecture

Cards Scale to Nodes
Node contains 8 Cards

Nodes enabled for C2C compilation deliver more per-card value than single cards.

Nodes Scale to Racks
GroqRack: 8 Compute Nodes + 1 Redundant Node

Racks deliver more per-node value than single nodes, & four racks deliver still more than one.

Built for scale

GroqChip™

GroqCard™

GroqNode™

GroqRack™

IMPACT

Deterministic Execution, Batch Size 1, TCO, & Scalable Systems

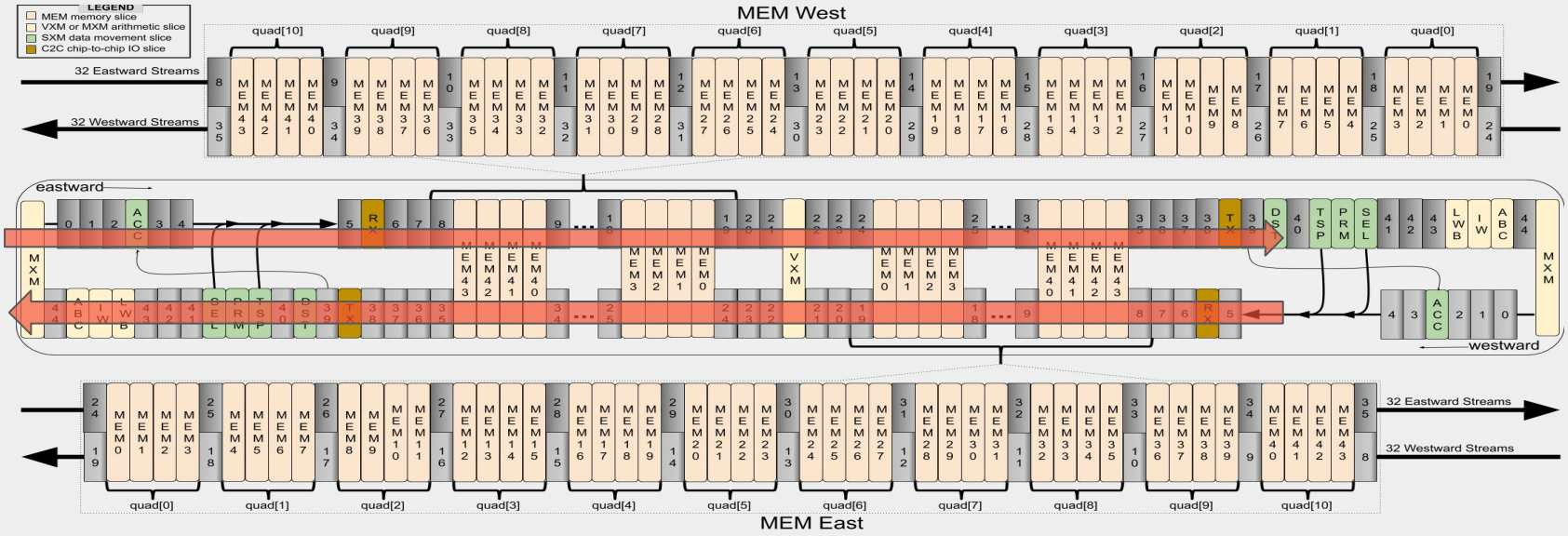
Determinism

Dataflow Model + Functional Units Ensures Determinism

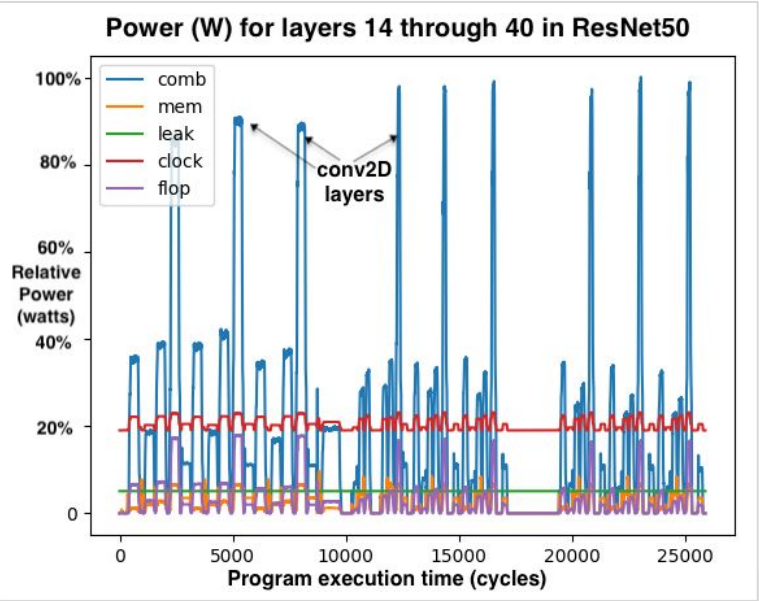
Built from the ground up

Model Latency Accurate Down to a Clock Cycle

No arbiters, caches, prediction



Importance of Determinism & Batch Size 1



Fine Grained Control of Power & Performance

At compile time

Easy to reason about chip performance

High Performance at Batch Size 1

Existing accelerators need large batch sizes to parallelize and achieve high throughput

High stream bandwidth to/from on chip memory allows high batch size 1 performance

Performance throughput and latency is independent of batch size

High Batch Size 1 performance reduces TCO for diverse SLAs

Conclusions

GroqChip™ 1	Deterministic HW architecture built from ground up Fine grained SW control
Determinism & Streaming Are Key to Architectural Benefits	Extract maximum parallelism from mode through data/instruction level parallelism Fine grained power, performance control at compile time Scalability
Batch Size 1 Performance Matters	Real-time (batch 1) applications put heavy strain on a datacenter deployment Batch size 1 lowers TCO

Questions?

For more information on Groq technology and products, contact us at

info@groq.com

Follow us on Twitter

@GroqInc

Connect with us on LinkedIn

https://www.linkedin.com/company/groq