

Analog Computation in Flash Memory for Datacenter-scale AI Inference in a Small Chip

Dave Fick, CTO/Founder

Mike Henry, CEO/Founder


About Mythic

- ✔ Focused on high-performance Edge AI
 - Full stack co-design: device physics to new algorithms and applications
- ✔ Founded in 2012 by Mike Henry and Dave Fick
 - While working with the Michigan Integrated Circuits Lab (MICL)
- ✔ Raised \$55M from top-tier investors: DFJ, SoftBank, Lux, DCVC
 - Offices in Redwood City & Austin
 - 60+ employees

DNNs are Largely Multiply-Accumulate

Primary DNN Calculation is Input Vector * Weight Matrix = Output Vector

Input Data	Neuron Weights	Outputs Equations
$[X_0 \quad X_1 \quad \dots \quad X_N]$	$\begin{bmatrix} A_0 & B_0 & C_0 \\ A_1 & B_1 & C_1 \\ \dots & \dots & \dots \\ A_N & B_N & C_N \end{bmatrix}$	$= \begin{bmatrix} Y_A = X_0A_0 + X_1A_1 + X_2A_2 \\ Y_B = X_0B_0 + X_1B_1 + X_2B_2 \\ Y_C = X_0C_0 + X_1C_1 + X_2C_2 \end{bmatrix}$

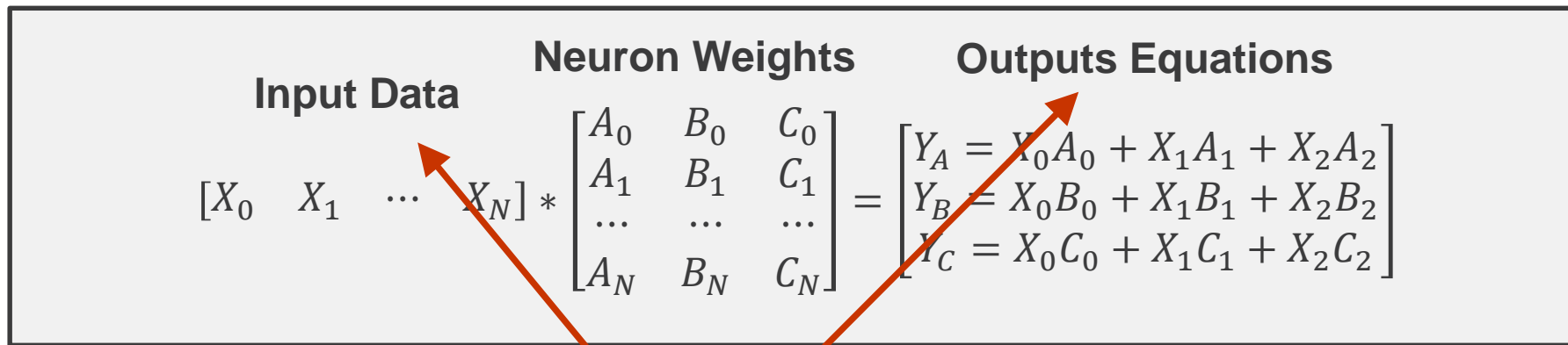


Key Operation: Multiply-Accumulate, or “MAC”

Figure of Merit: How many picojoules to execute a MAC?

Memory Access Includes Weight Data and Intermediate Data

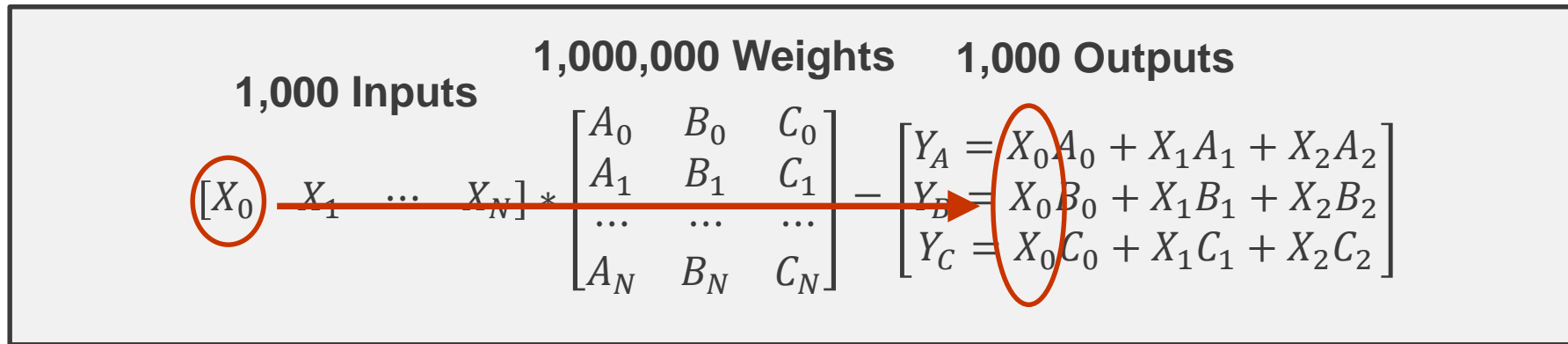
“Weight Data”



“Intermediate Data”

Intermediate Data Accesses are Naturally Amortized

For a 1000 input, 1000 neuron matrix....



Intermediate data accesses are amortized **64-1024x**
since they are used in many MAC operations

Weight Data Accesses are Not Amortized

For a 1000 input, 1000 neuron matrix....

1,000 Inputs	1,000,000 Weights	1,000 Outputs
$[X_0 \quad X_1 \quad \dots \quad X_N]$	$\begin{bmatrix} A_0 & B_0 & C_0 \\ A_1 & B_1 & C_1 \\ \dots & \dots & \dots \\ A_N & B_N & C_N \end{bmatrix}$	$\begin{bmatrix} Y_A = X_0A_0 + X_1A_1 + X_2A_2 \\ Y_B = X_0B_0 + X_1B_1 + X_2B_2 \\ Y_C = X_0C_0 + X_1C_1 + X_2C_2 \end{bmatrix}$

Note: The weight matrix and the resulting output equations are circled in red in the original image.

Weight data could need to be stored in *DRAM*, and it does not have the same amortization as the intermediate data

DNN Processing is All About Weight Memory

- ✓ 10+M parameters to store
- ✓ 20+B memory accesses
- ✓ How do we achieve...
 - High Energy Efficiency
 - High Performance
 - “Edge” Power Budget (e.g., 5W)

Network	Weights	MACs	...@ 30 FPS
AlexNet ¹	61 M	725 M	22 B
ResNet-18	11 M	1.8 B	54 B
ResNet-50	23 M	3.5 B	105 B
VGG-19 ¹	144 M	22 B	660 B
OpenPose ²	46 M	180 B	5400 B

**Very hard to fit this
in an Edge solution**

¹: 224x224 resolution
²: 656x368 resolution

Common Techniques for Reducing Weight Energy Consumption

Weight Re-use

✓ Focus on CNN

- Re-use weights for multiple windows
- Can build specialized structures

☹ ***Not all problems map to CNN well***

✓ Focus on Large Batch

- Re-use weights for multiple inputs

☹ ***Edge is often batch=1***

☹ ***Increases latency***

Weight Reduction

✓ Shrink the Model

- Use a smaller network that can fit on-chip (e.g., SqueezeNet)

☹ ***Possibly reduced capability***

✓ Compress the Model

- Use sparsity to eliminate up to 99% of the parameters
- Use literal compression

☹ ***Possibly reduced capability***

✓ Reduce Weight Precision

- 32b Floating Point => 2-8b Integer

☹ ***Possibly reduced capability***

Key Question: Use DRAM or Not?

Benefits of DRAM

- 😊 Can fit arbitrarily large models
- 😊 Not as much SRAM needed on chip

Drawbacks of DRAM

- 😞 Huge energy cost for reading weights
- 😞 Limited bandwidth getting to weight data
- 😞 Variable energy efficiency & performance depending on application

Common NN Accelerator Design Points

	Enterprise With DRAM	Enterprise No-DRAM	Edge With DRAM	Edge No-DRAM
SRAM	<50 MB	100+ MB	< 5 MB	< 5 MB
DRAM	8+ GB	-	4-8 GB	-
Power	70+ W	70+ W	3-5 W	1-3 W
Sparsity	Light	Light	Moderate	Heavy
Precision	32f / 16f / 8i	32f / 16f / 8i	8i	1-8i
Accuracy	Great	Great	Moderate	Poor
Performance	High	High	Very Low	Very Low
Efficiency	25 pJ/MAC	2 pJ/MAC	10 pJ/MAC	5 pJ/MAC

Mythic is Fundamentally Different

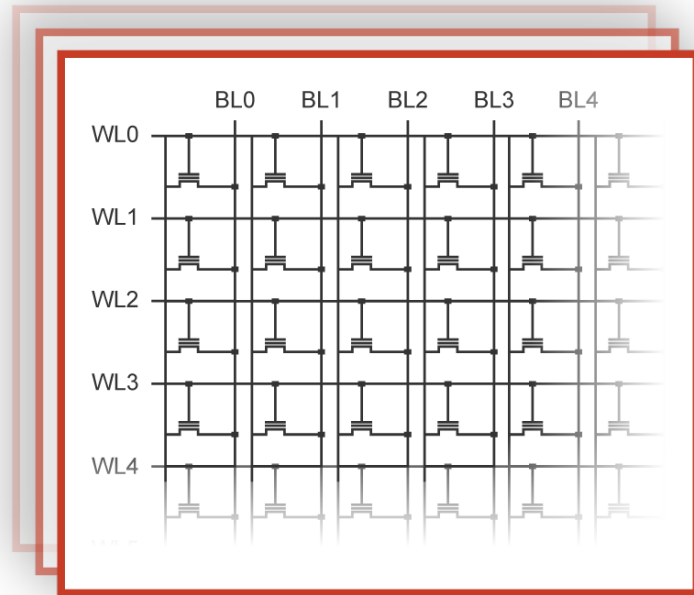
	Enterprise With DRAM	Enterprise No-DRAM	Edge With DRAM	Edge No-DRAM	Mythic NVM
SRAM	<50 MB	100+ MB	< 5 MB	< 5 MB	< 5 MB
DRAM	8+ GB	-	4-8 GB	-	-
Power	70+ W	70+ W	3-5 W	1-3 W	1-5 W
Sparsity	Light	Light	Moderate	Heavy	None
Precision	32f / 16f / 8i	32f / 16f / 8i	8i	1-8i	1-8i
Accuracy	Great	Great	Moderate	Poor	Great
Performance	High	High	Very Low	Very Low	High
Efficiency	25 pJ/MAC	2 pJ/MAC	10 pJ/MAC	5 pJ/MAC	0.5 pJ/MAC

Mythic is Fundamentally Different

	Enterprise With DRAM	Enterprise No-DRAM	Edge With DRAM	Edge No-DRAM	Mythic NVM
SRAM	<50 MB	100+ MB	< 5 MB	< 5 MB	< 5 MB
DRAM	8+ GB	-	4-8 GB	-	-
Power	7 W	Also, Mythic does this in a 40nm process, compared to 7/10/16nm			1-5 W
Sparsity	L				None
Precision	32f / 16f / 8i	32f / 16f / 8i	8i	1-8i	1-8i
Accuracy	Great	Great	Moderate	Poor	Great
Performance	High	High	Very Low	Very Low	High
Efficiency	25 pJ/MAC	2 pJ/MAC	10 pJ/MAC	5 pJ/MAC	0.5 pJ/MAC

Mythic's New Architecture Merges Enterprise and Edge

- ✓ Mythic introduces the ***Matrix Multiplying Memory***
 - Never read weights
- ✓ This effectively makes weight memory access ***energy-free*** (only pay for MAC)
- ✓ And eliminates the need for...
 - Batch > 1
 - CNN Focus
 - Sparsity or Compression
 - Nerfed DNN Models



*Made possible with
Mixed-Signal Computing
on embedded flash*

Revisiting Matrix Multiply

Primary DNN Calculation is Input Vector * Weight Matrix = Output Vector

Input Data	Neuron Weights	Outputs Equations
$[X_0 \quad X_1 \quad \dots \quad X_N]$	$\begin{bmatrix} A_0 & B_0 & C_0 \\ A_1 & B_1 & C_1 \\ \dots & \dots & \dots \\ A_N & B_N & C_N \end{bmatrix}$	$= \begin{bmatrix} Y_A = X_0A_0 + X_1A_1 + X_2A_2 \\ Y_B = X_0B_0 + X_1B_1 + X_2B_2 \\ Y_C = X_0C_0 + X_1C_1 + X_2C_2 \end{bmatrix}$

Flash Transistors

Analog Circuits Give us the MAC We Need

Flash transistors can be modeled as **variable resistors** representing the weight

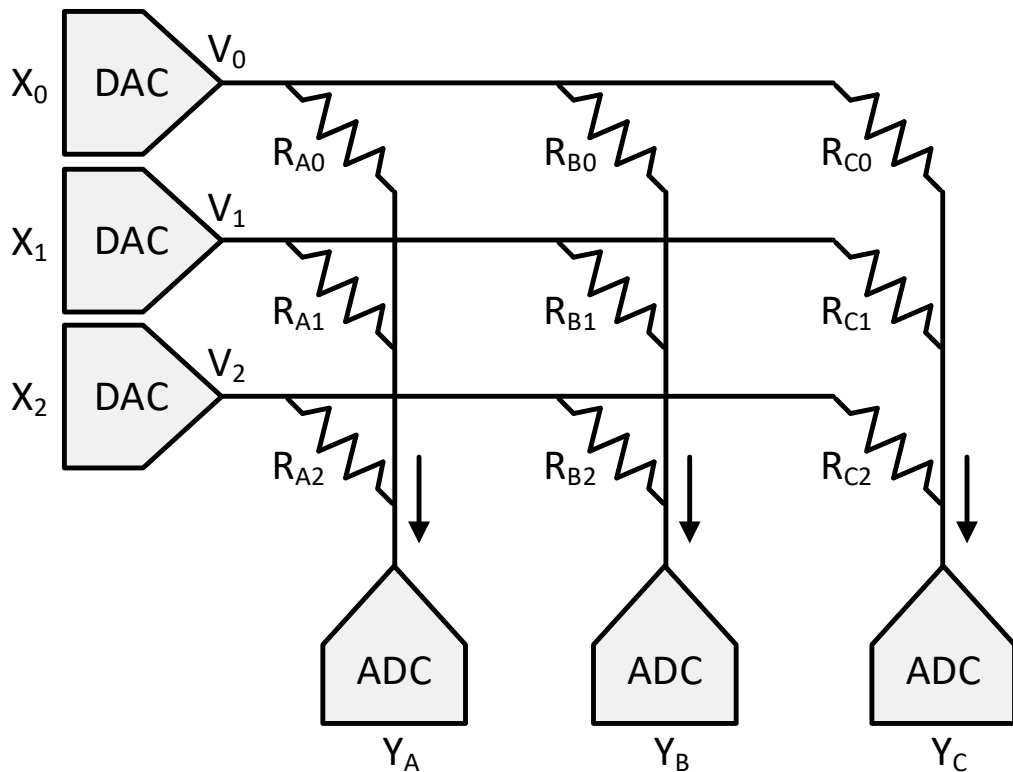
The $V=IR$ current equation will achieve the math we need:

Inputs (X) = DAC

Weights (R) = Flash transistors

Outputs (Y) = ADC Outputs

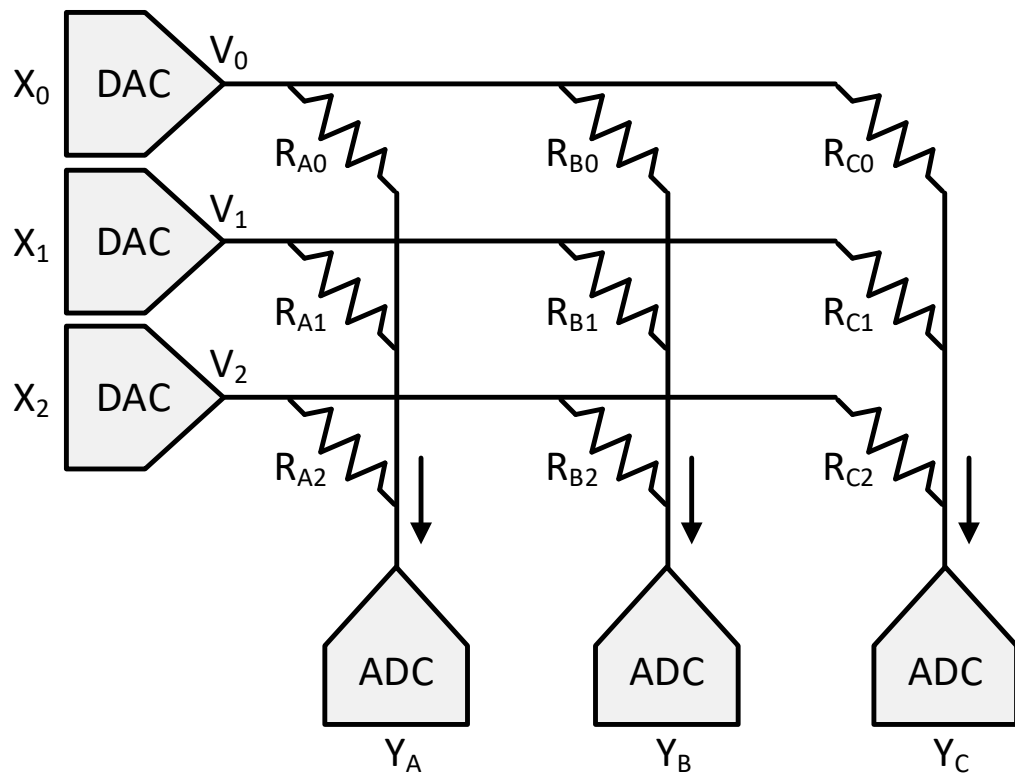
The ADCs convert current to digital codes, and provide the non-linearity needed for DNN



DACs & ADCs Give Us a Flexible Architecture

We have a **digital** top-level architecture:

- ✓ Interconnect
- ✓ Intermediate data storage
- ✓ Programmability
(XLA/ONNX => Mythic IPU)

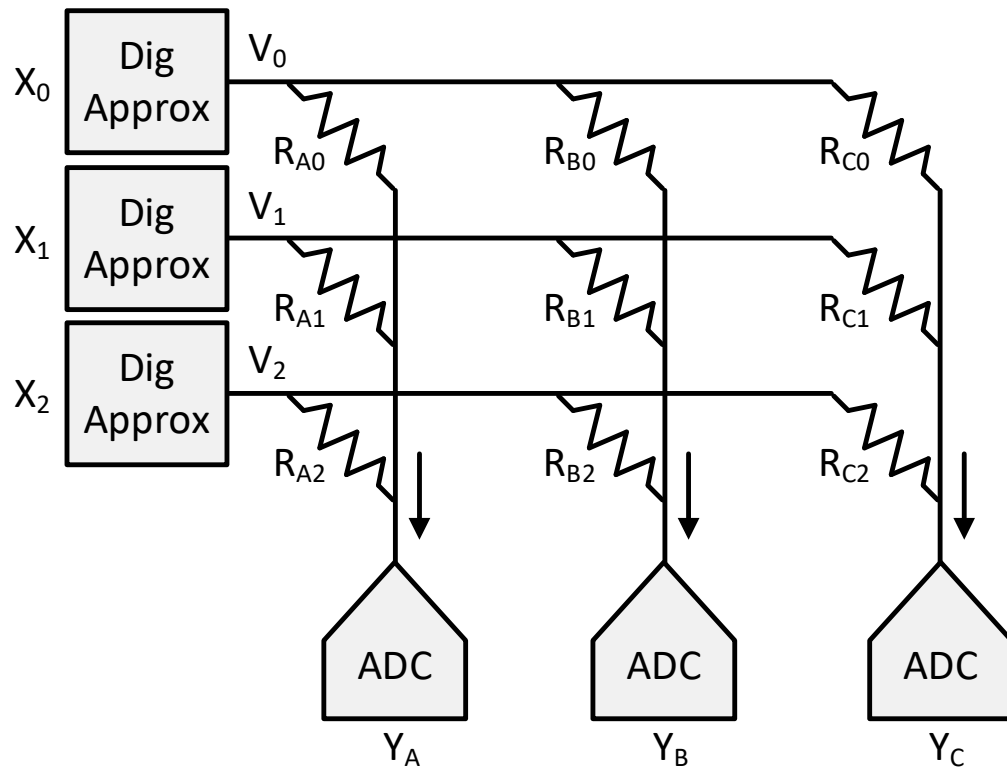


To Simplify we use Digital Approximation

To improve time-to-market, we have left the Input DAC as a future endeavor

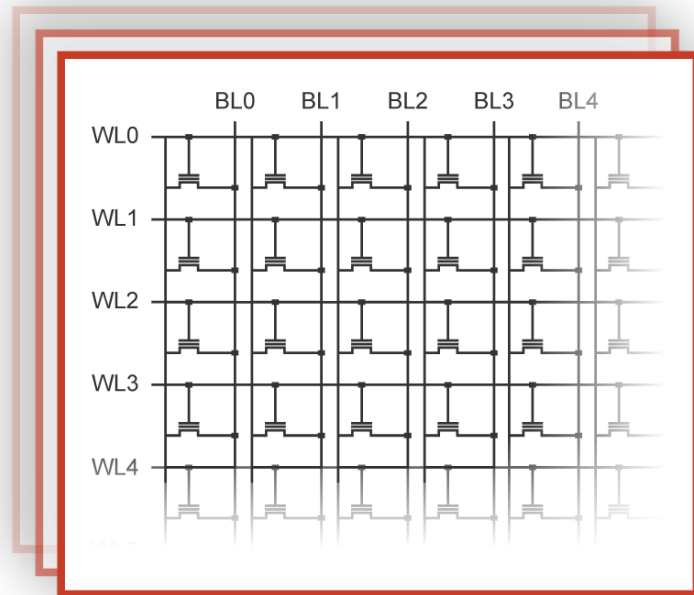
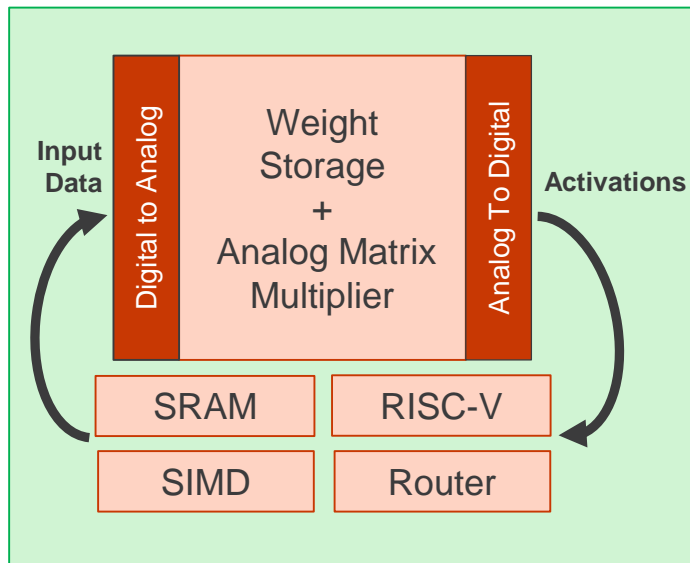
We achieve the same result through digital approximation

Silver lining: we have future improvements available



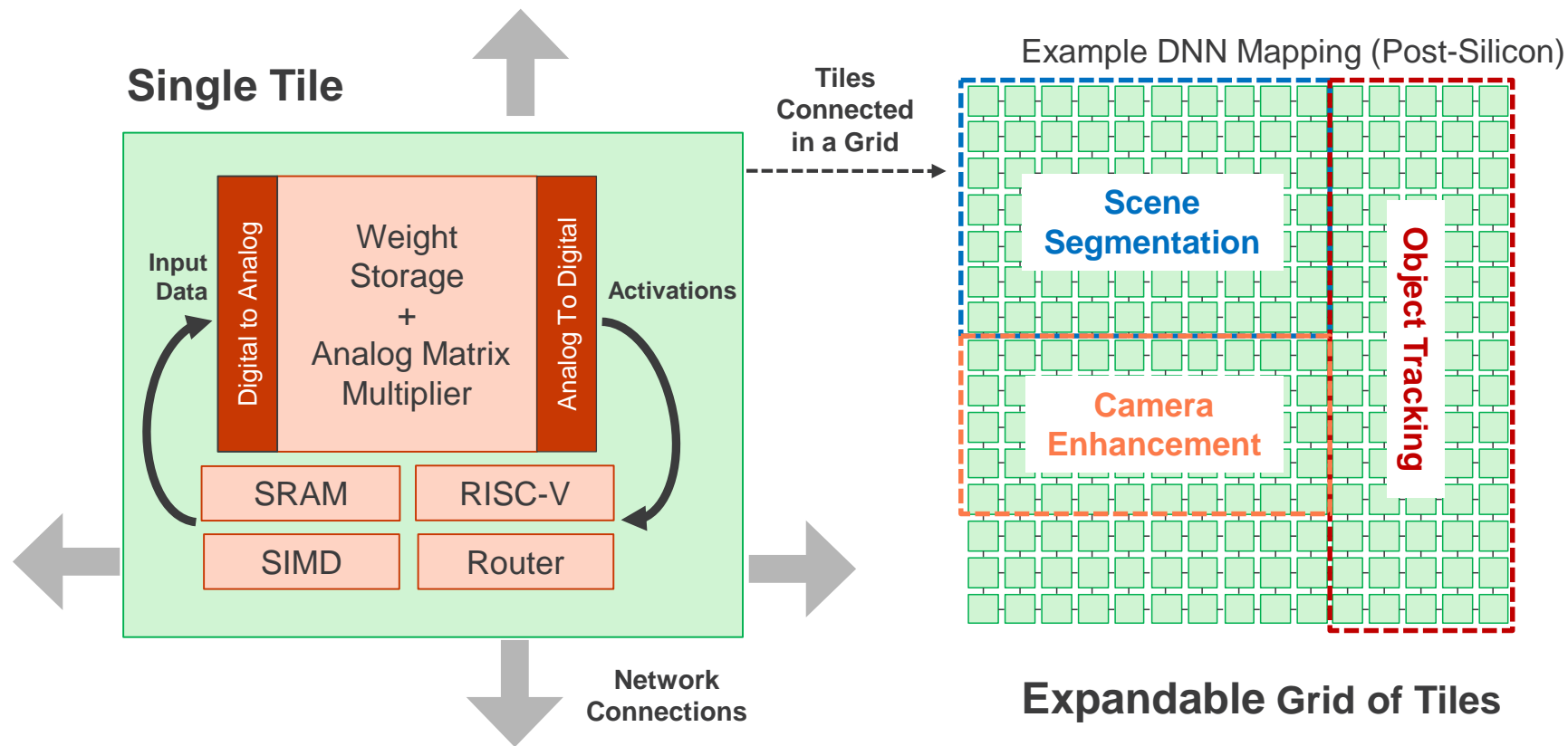
Mythic Mixed-Signal Computing

Single Tile



*Made possible with
Mixed-Signal Computing
on embedded flash*

Mythic Mixed-Signal Computing



System Overview

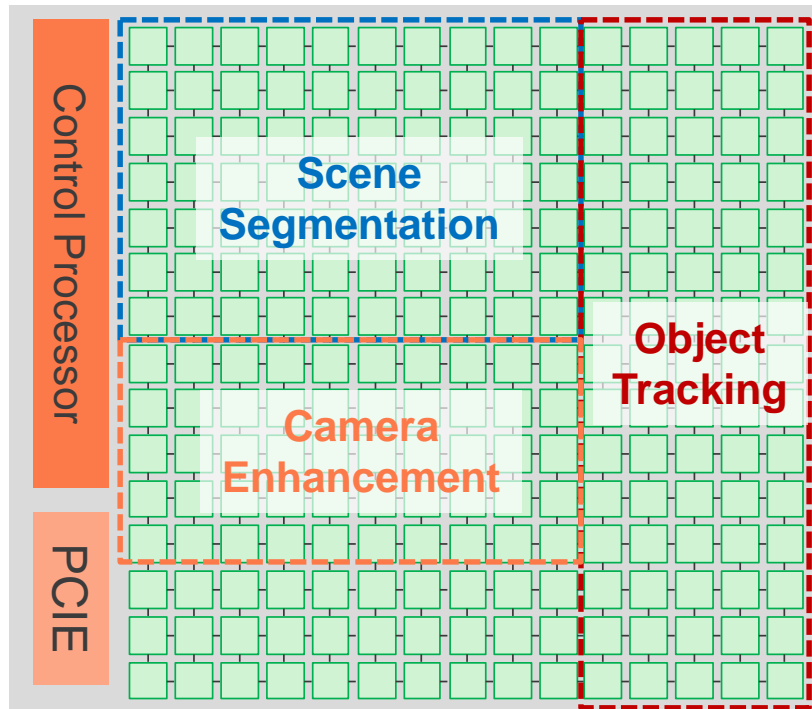
Initial Product

- 50M weight capacity
- PCIe 2.1 x4
- Basic Control Processor

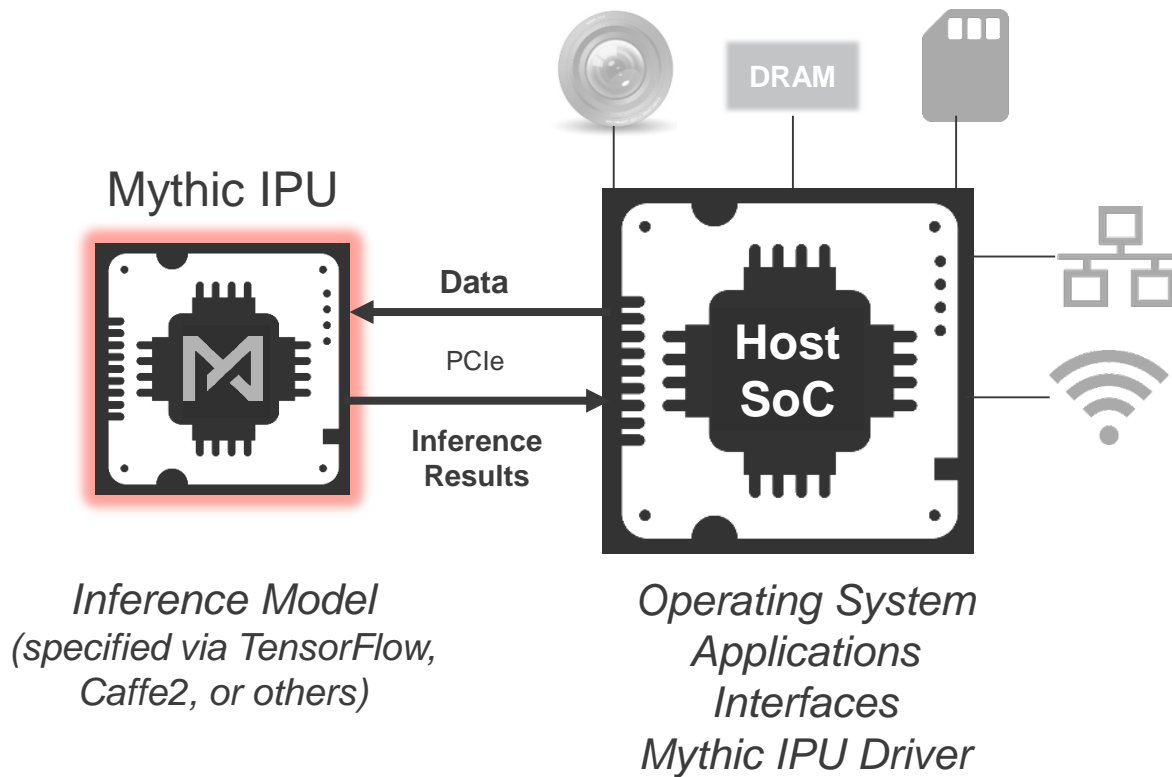
Envisioned Customizations (Gen 1)

- Up to 250M weight capacity
- PCIe 2.1 x16
- USB 3.0/2.0
- Direct Audio/Video Interfaces
- Enhanced Control Processor (e.g., ARM)

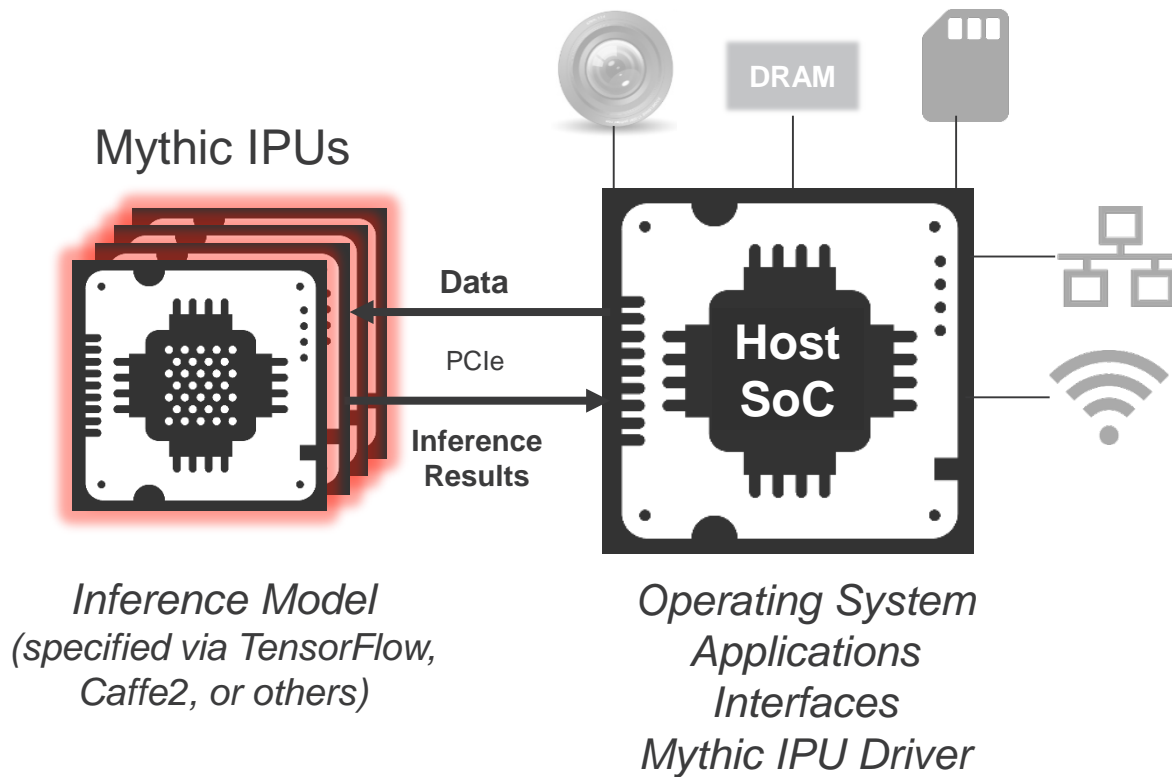
Intelligence Processing Unit (IPU)



Mythic is a PCIe Accelerator



We Also Support Multiple IPUs

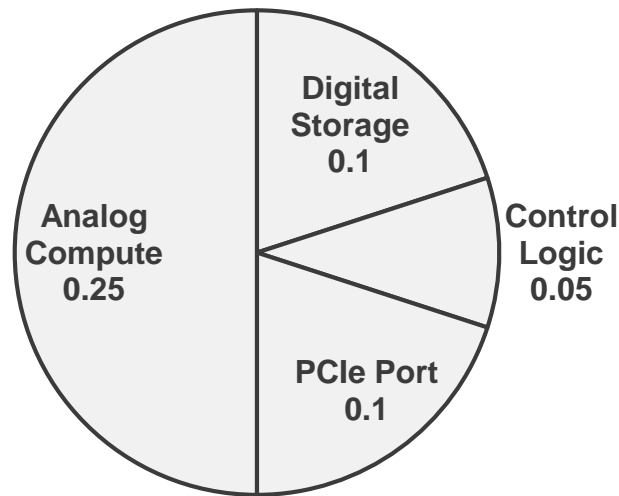


We Account For All Energy Consumed

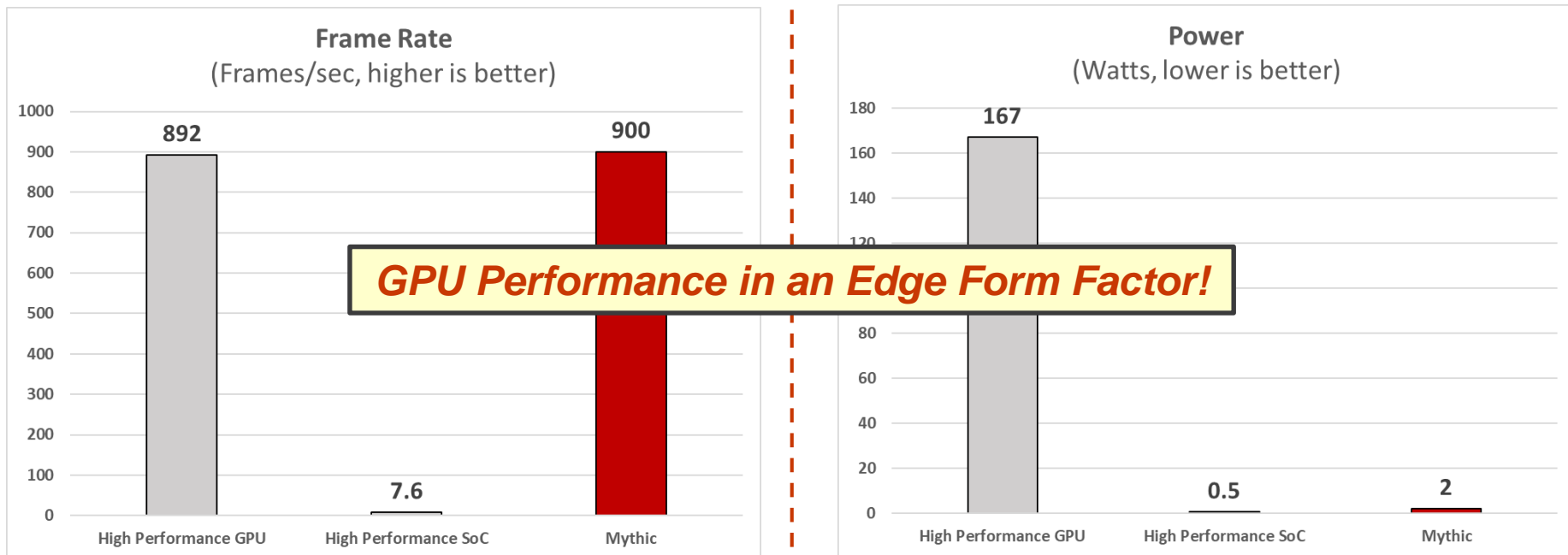
- ✓ Numbers are for a typical application, e.g. ResNet-50
 - Batch size = 1
 - We are relatively application-agnostic (especially compared to DRAM-based systems)
- ✓ 8b analog compute accounts for about half of our energy
 - We can also run lower precision
 - Control, storage, and PCIe accounts for the other half

Energy (pJ/MAC)

Total = 0.5

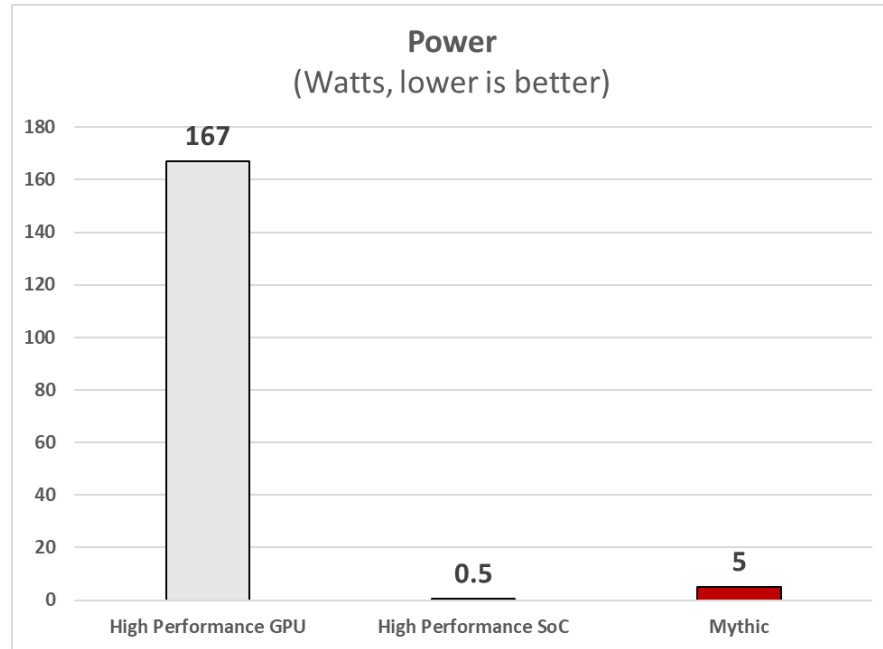
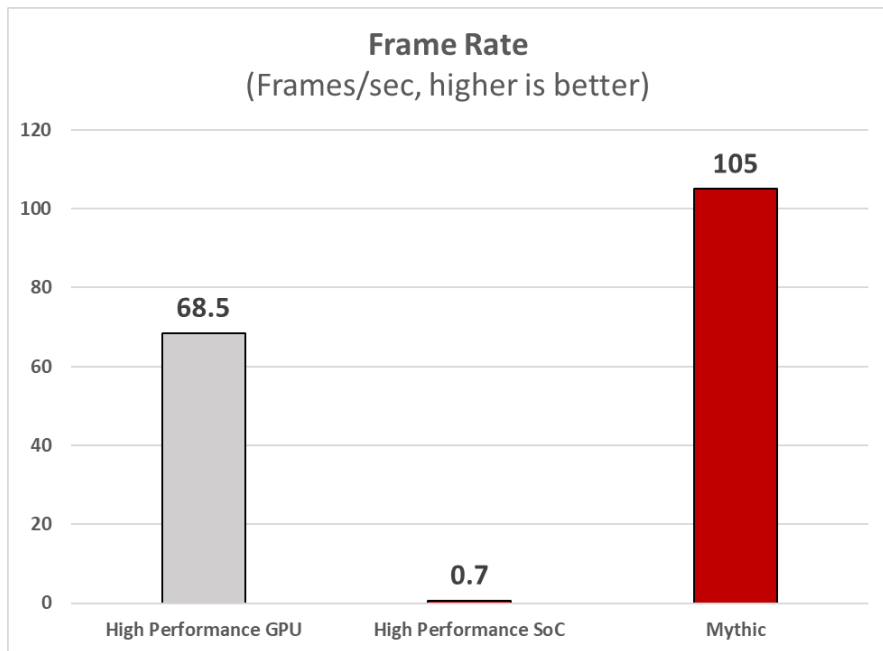


Example Application: ResNet-50



Running at 224x224 resolution. Mythic estimated, GPU/SoC measured

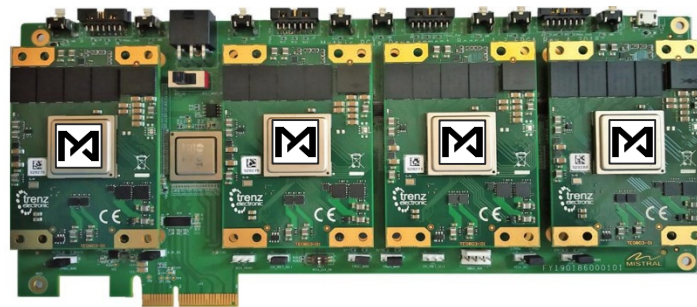
Example Application: OpenPose



Running at 656x368 resolution. Mythic estimated, GPU/SoC measured

Timeline

- ✓ Alpha release of software tools and profiler: Late 2018
- ✓ External samples: Mid 2019
 - PCIe Dev Boards (1, 4 IPUUs)
- ✓ Volume shipments: Late 2019
 - BGA Chips
 - PCIe Cards (1, 4, 16 IPUUs)



Four IPU PCIe card

Mythic IPU Overview

✓ Low Latency

- Runs batch size = 1
- E.g., single frame delay

✓ High Performance

- 10's of TMAC/s

✓ High Efficiency

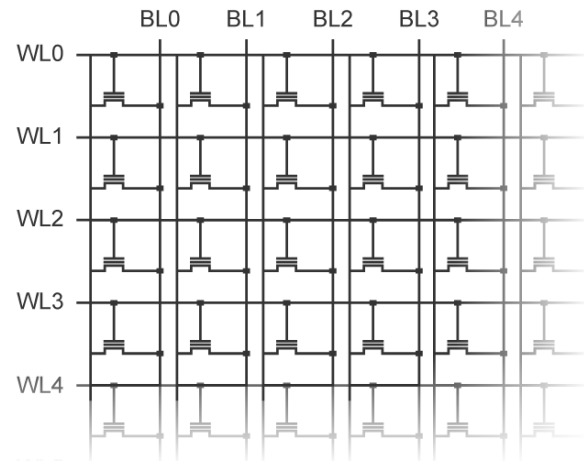
- 0.5 pJ/MAC aka 500mW / TMAC

✓ Hyper-Scalable

- Ultra low power to high performance

✓ Easy to use

- Topology agnostic (CNN/DNN/RNN)
- TensorFlow/Caffe2/etc supported



*Made possible with
Mixed-Signal Computing
on embedded flash*

Thank you for listening!

Questions?