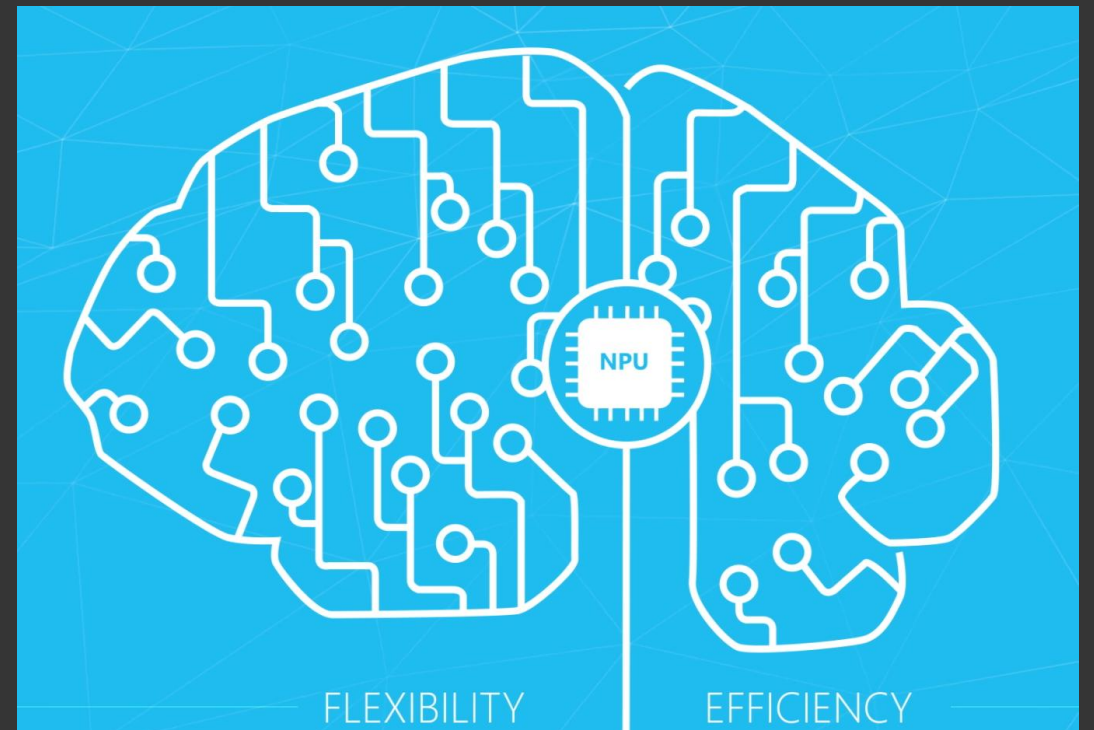


# Global-Scale FPGA-Accelerated Deep Learning Inference with Microsoft's Project Brainwave

Gabriel Weisz  
Bing Engineering  
Microsoft



# Over 1 Million Catapult FPGAs in Our Data Centers



Machine Learning



Accelerated Networking

# Catapult FPGA Servers

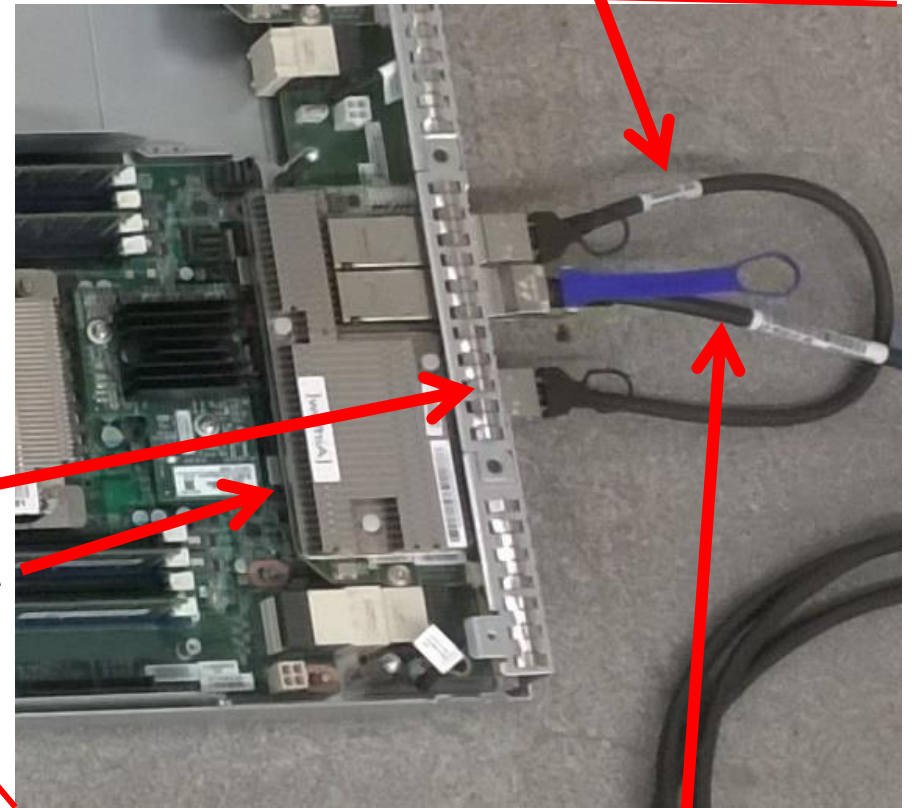


FPGA is between the CPU and the Top-of-Rack (TOR) switch

FPGAs are included in every server  
Microsoft has deployed since 2015

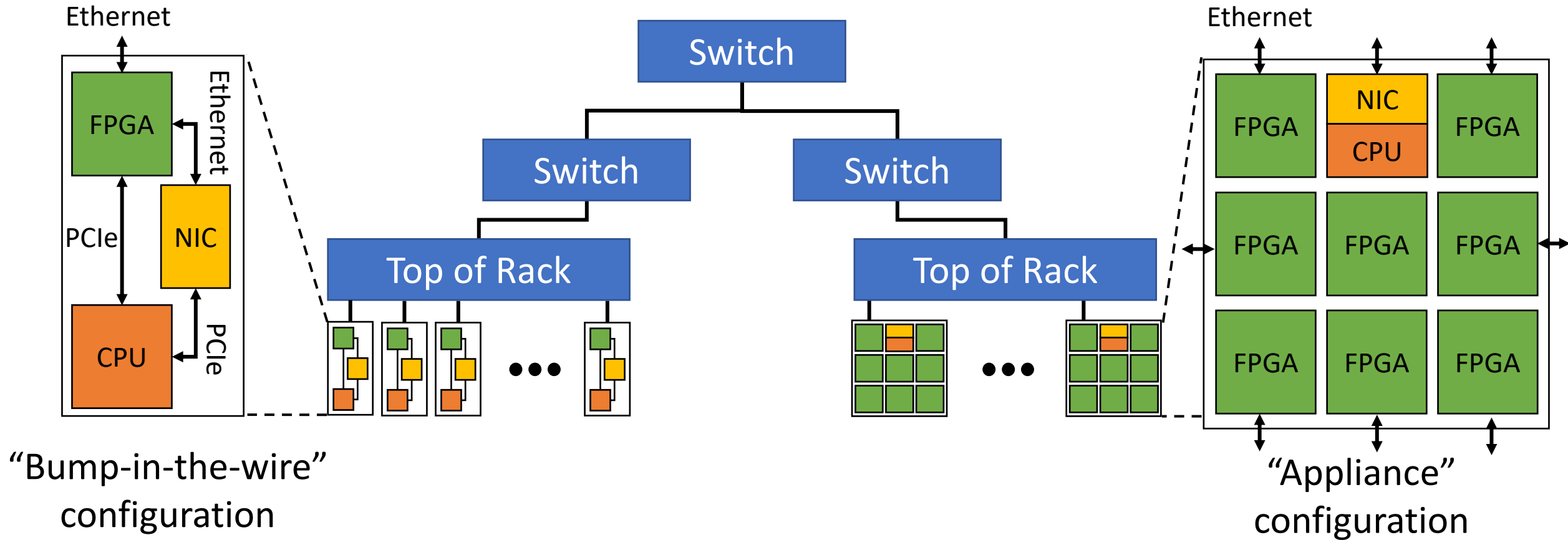
CPU NIC  
FPGA

0.5m QSFP cable from NIC to FPGA

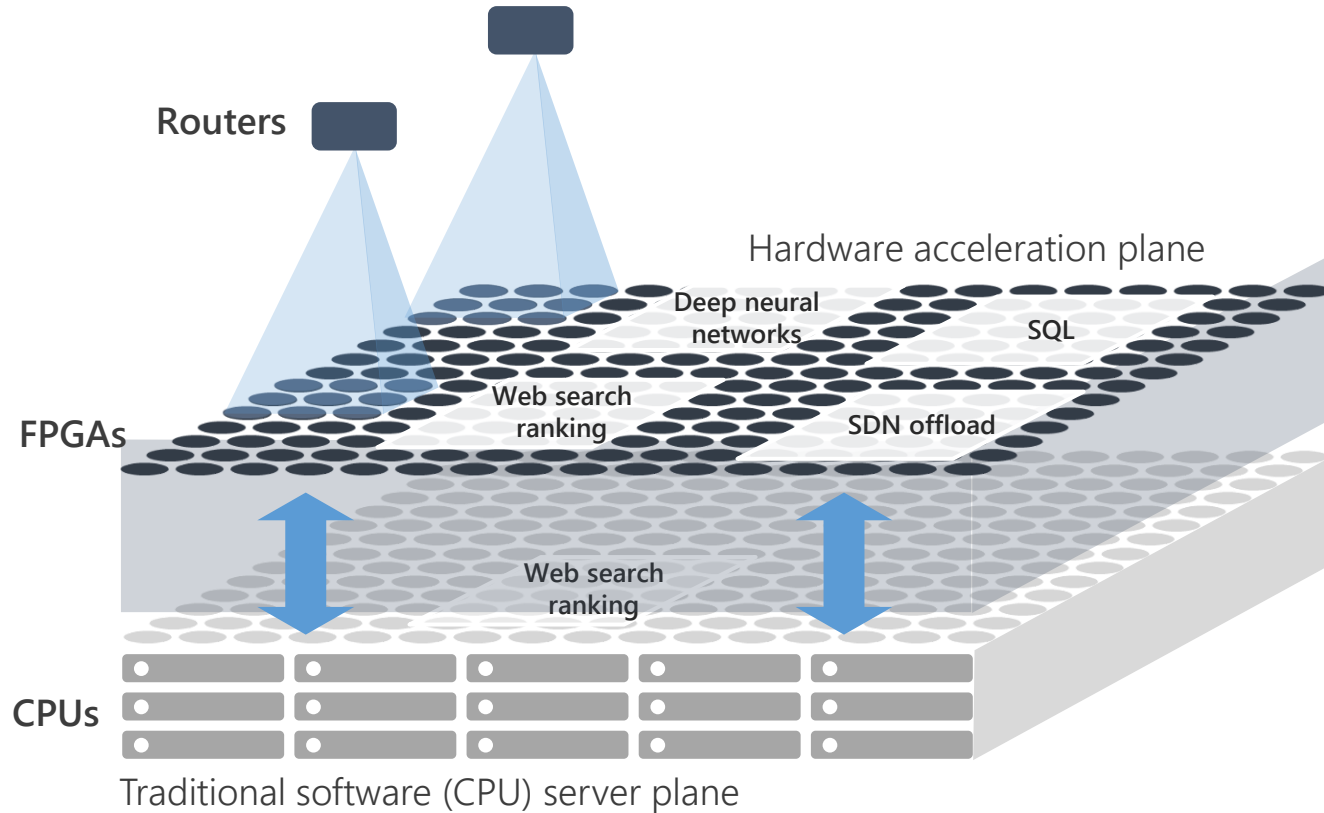


~3m QSFP cable from FPGA to TOR  
[Slide courtesy Andrew Putnam]

# Catapult in the Data Center



# Catapult + Software = Hardware Microservices



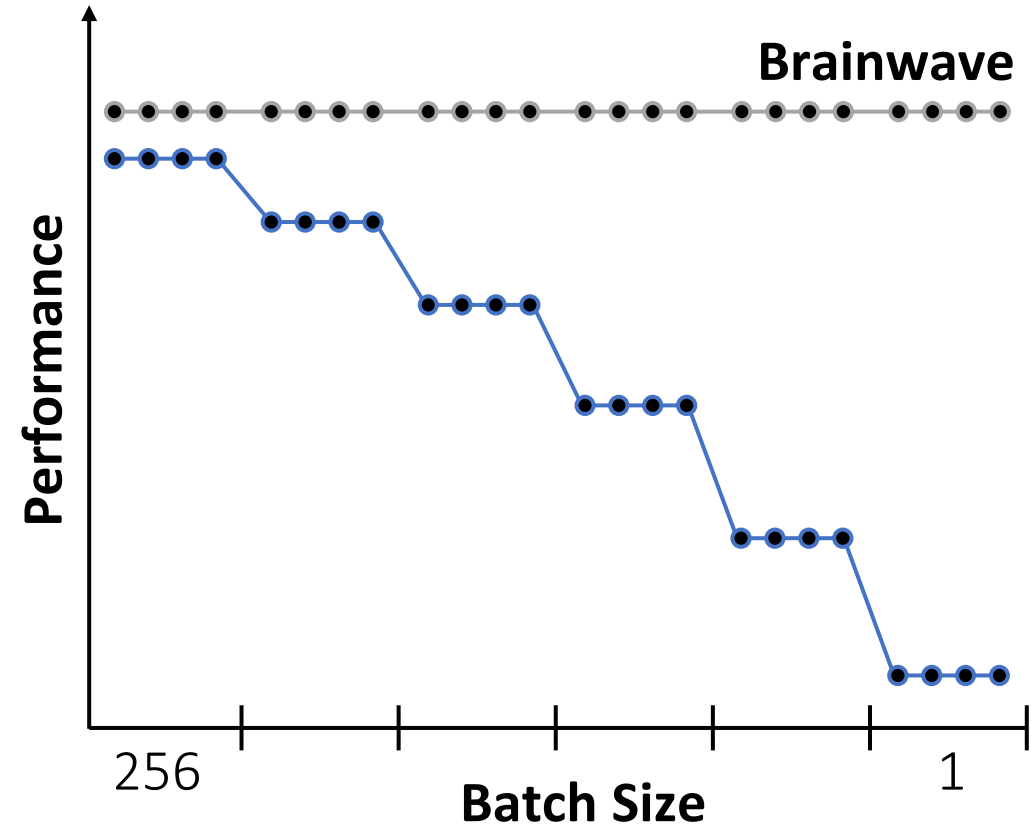
- **Interconnected FPGAs form a separate plane of computation**
- **FPGAs are used and managed independently from the CPU**
- **Applications are mapped across multiple FPGAs and CPUs**

# Hardware Microservices for Real-Time AI

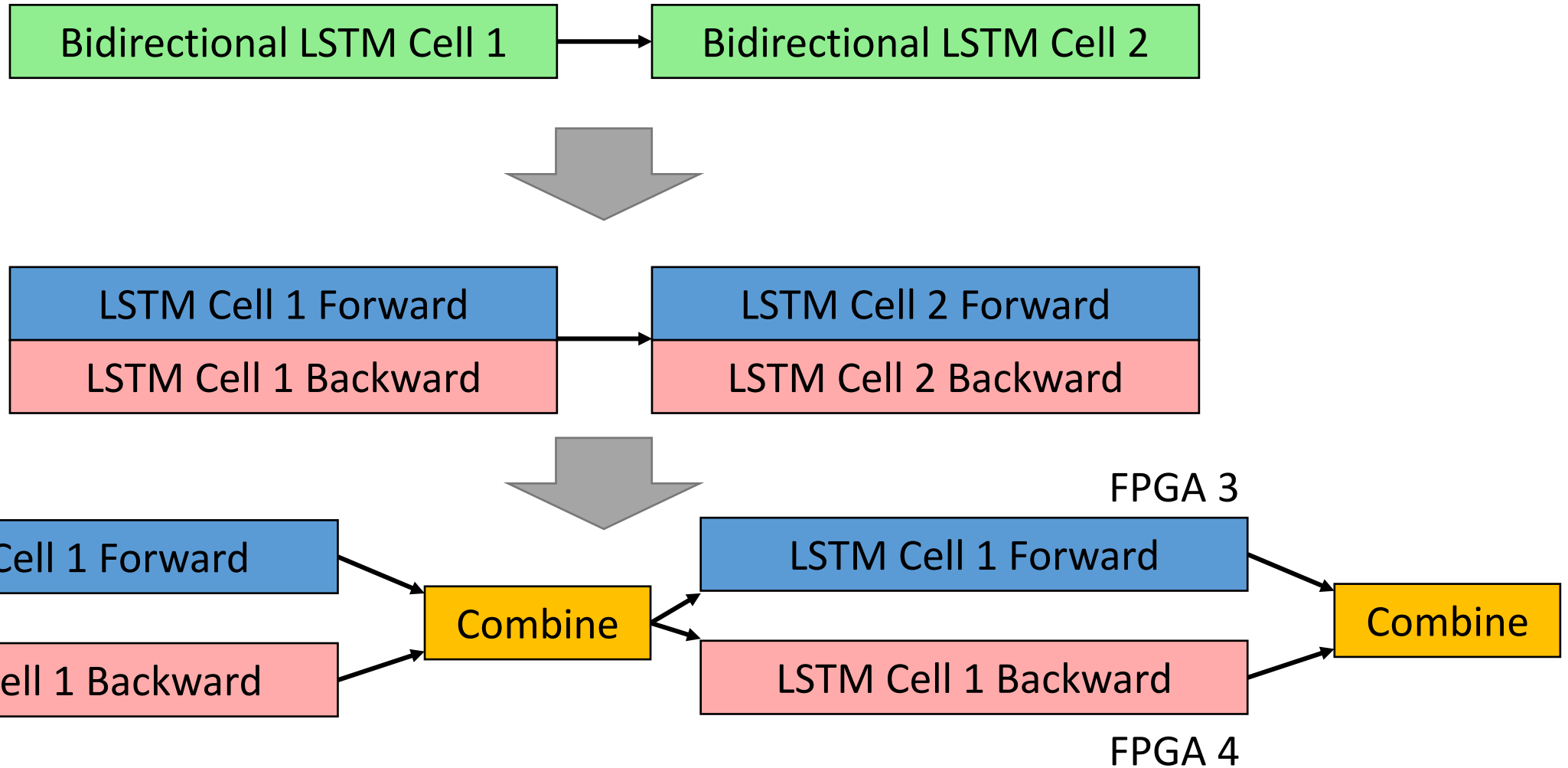
Real-Time AI = low latency without batching

Brainwave maps neural network models to multiple network-attached FPGAs

Weights are pinned to registers for low latency



# Mapping a Model Across FPGAs





# Bing Intelligent Search Powered By Brainwave

DECEMBER  
13  
2017



Bing launches new intelligent search features, powered by AI

Today we announced new [Intelligent Search](#) features for Bing, powered by AI, to give you answers faster, give you more comprehensive and complete information, and enable you to interact more naturally with your search engine.

Intelligent answers:

Intelligent answers leverage the latest [state of the art machine reading comprehension](#), backed by [Project Brainwave running on Intel's FPGAs](#), to read and analyze billions of documents to understand the web and help you more quickly and confidently get the answers you need.

Bing now uses deep neural networks to validate answers by aggregating across multiple reputable sources, rather than just one, so you can feel more confident about the answer you're getting.

 when did pembroke college in Rhode Island change names 

AllImagesVideosMapsNewsShopMy saves

281,000 ResultsAny time ▾

1928

Consolidated from multiple sources

In **1928**, the Women's College was renamed "Pembroke College in Brown University" in honor of Pembroke College at the University of Cambridge in England. Roger Williams, one of the founders of Rhode Island, was an alumnus of Cambridge's Pembroke.

[Pembroke College in Brown University - Wikipedia](#)  
[en.wikipedia.org](#)

Similar answer at: [brown.edu](#)

FPGA-Accelerated model is **much** faster even though it is more complicated

Bing TP1			
	CPU-only	Brainwave-accelerated	Improvement
Model details	GRU 128x200 (x2) + W2Vec	LSTM 500x200 (x8) + W2Vec	Brainwave-accelerated model is > 10X larger and > 10X lower latency
End-to-end latency per Batch 1 request at 95%	9 ms	0.850 ms	
Bing DeepScan			
	CPU-only	Brainwave-accelerated	Improvement
Model details	1D CNN + W2Vec (RNNs removed)	1D CNN + W2Vec + GRU 500x500 (x4)	Brainwave-accelerated model is > 10X larger and 3X lower latency
End-to-end latency per Batch 1 request at 95%	15 ms	5 ms	

CPU vs Stratix V performance on production models



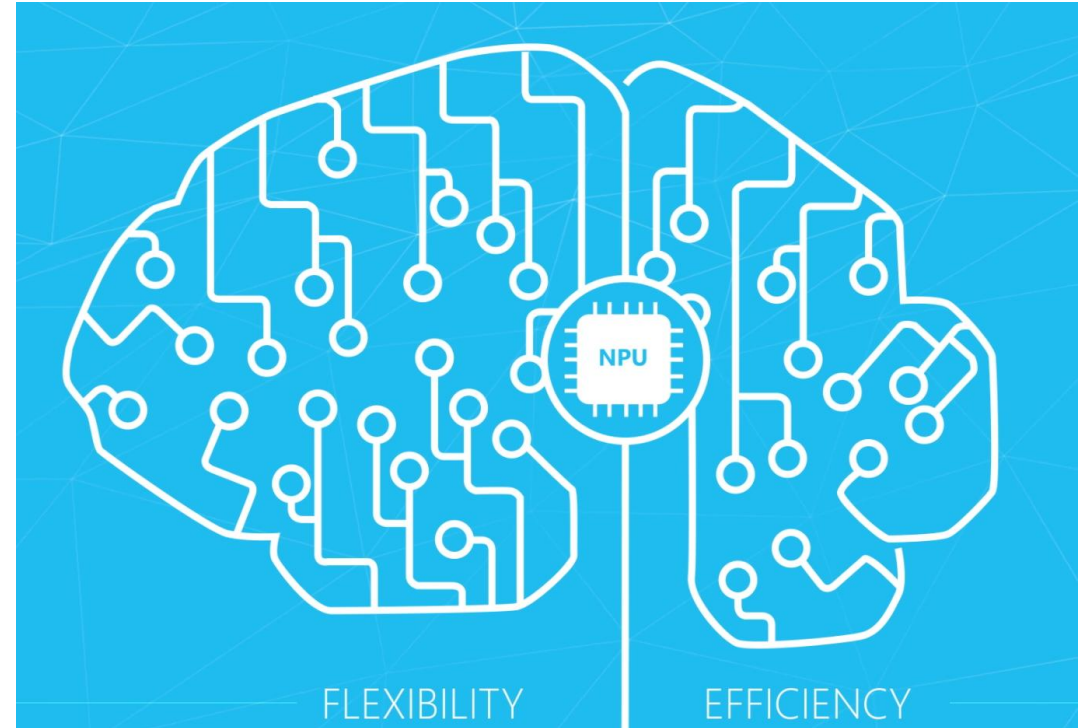
# Brainwave Components

## FPGA-based overlay (“NPU”)

- Highly parameterized
- Supports multiple FPGA device generations
- Run-time programmable

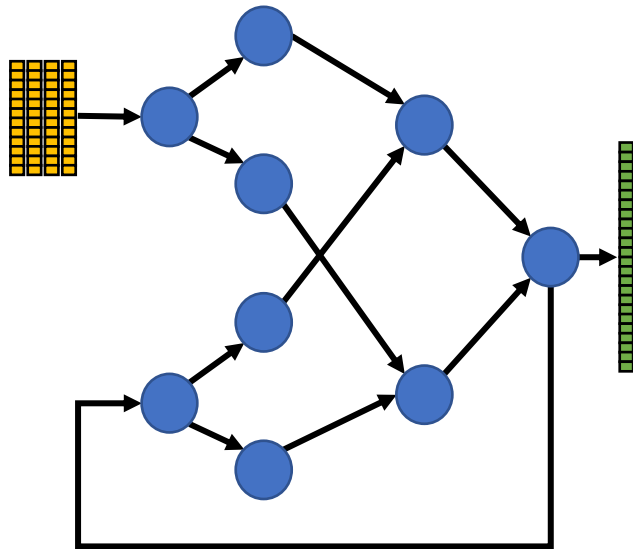
## Enterprise-grade software stack

- FPGA management
- Orchestration of computations
- Model compiler

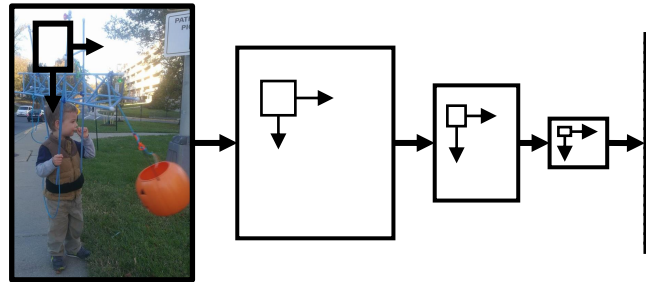


**This talk focuses on the overlay**

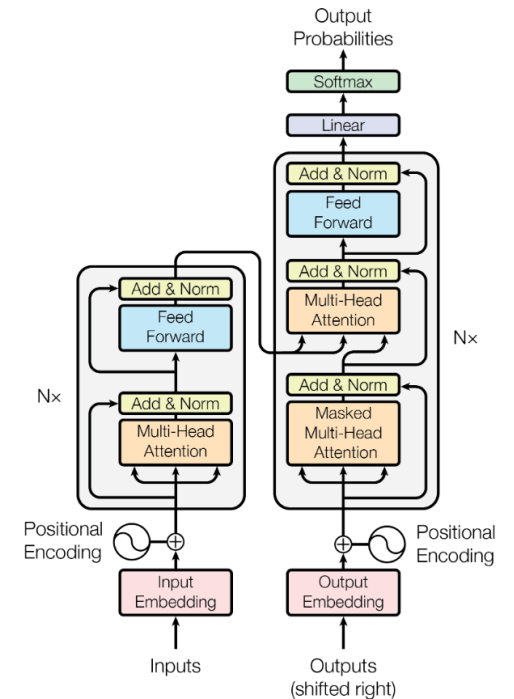
# Deep Learning Network Topologies



Recurrent  
Networks



Convolutional  
Networks

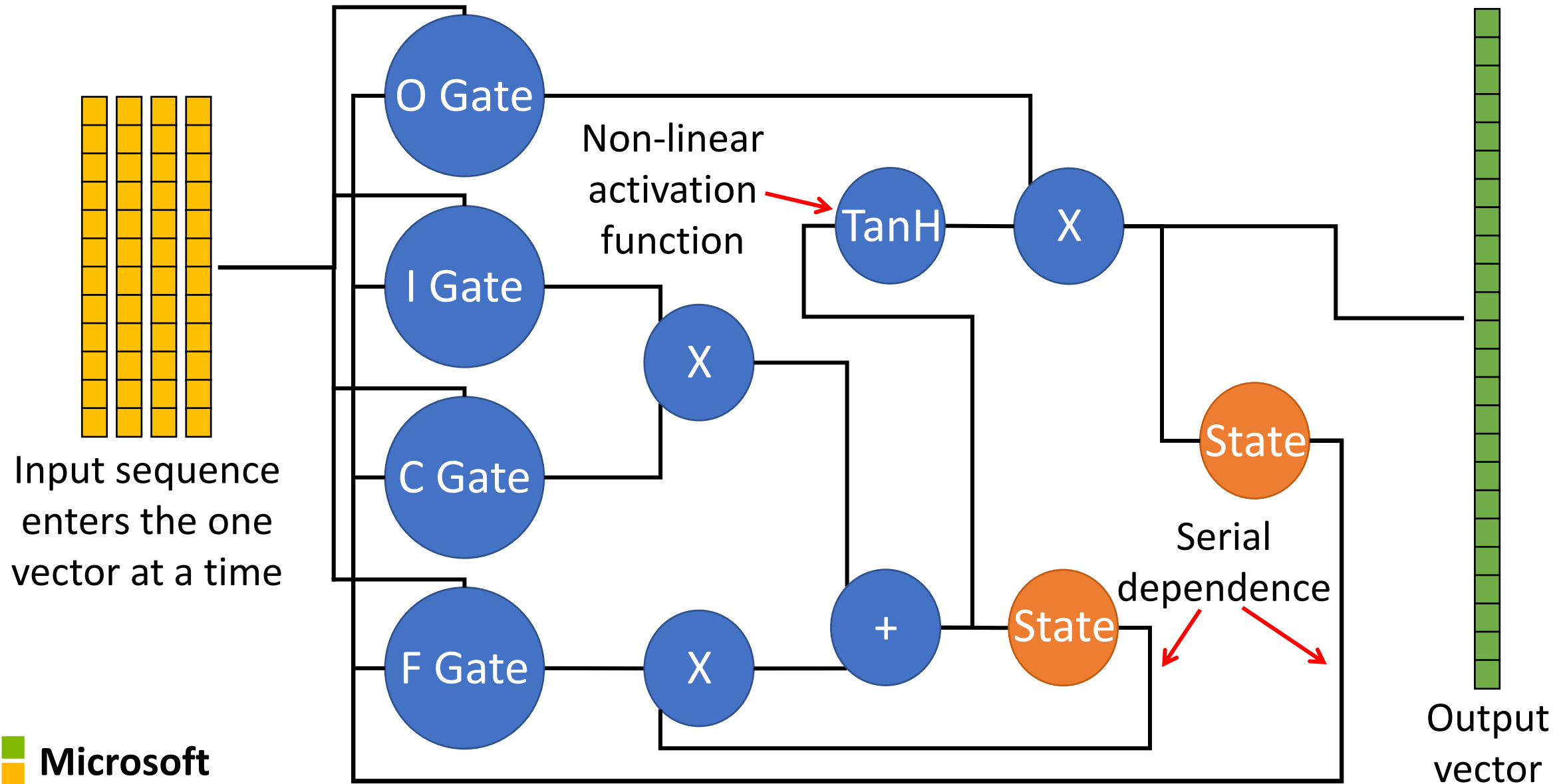


[Vaswani+, "Attention is all You Need", arXiv]

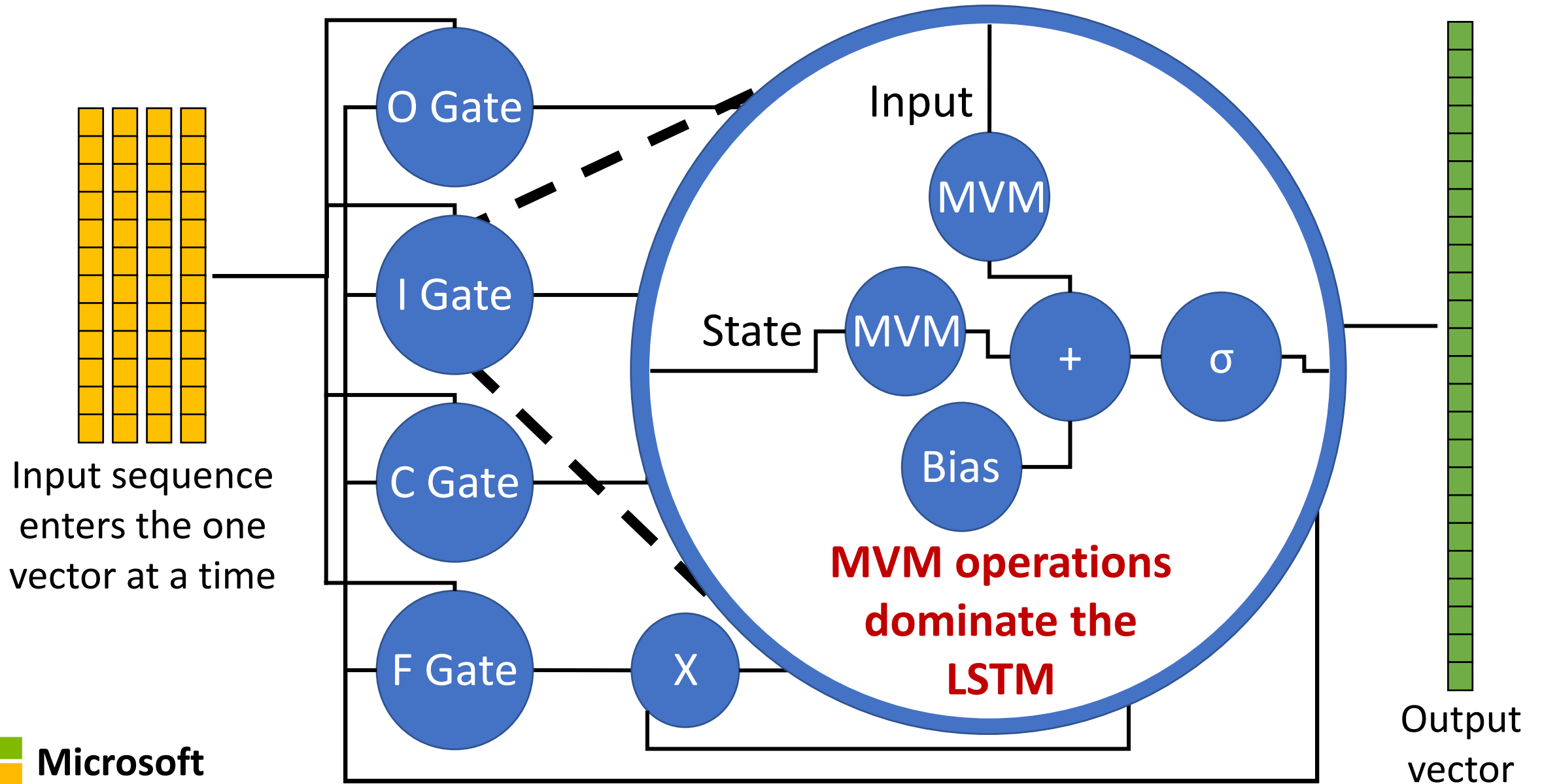
Transformer  
Networks

**What computations do we need to support?**

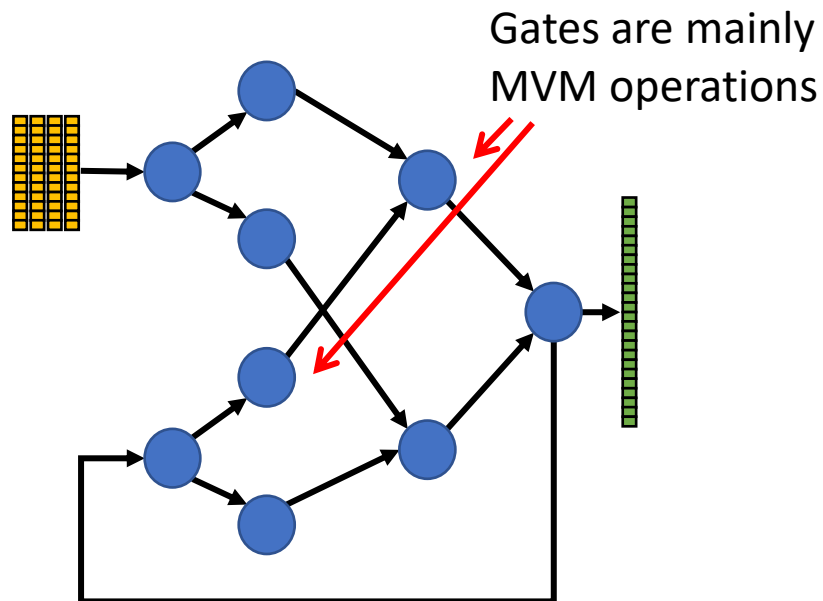
# Example RNN: Long Short-Term Memory (LSTM)



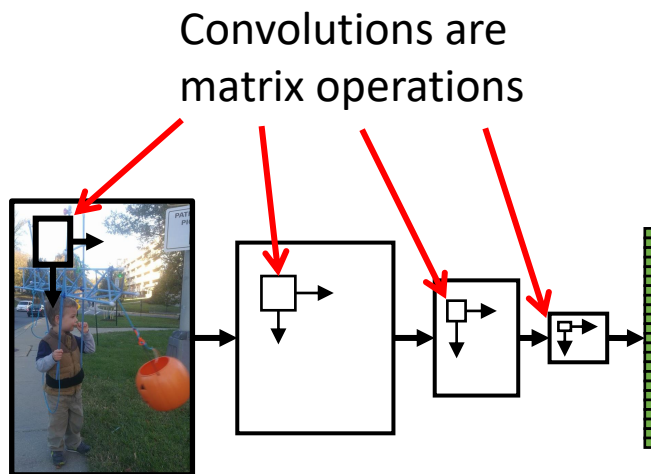
# Example RNN: Long-Short Term Memory (LSTM)



# (Almost) Everything is a Matrix Operation

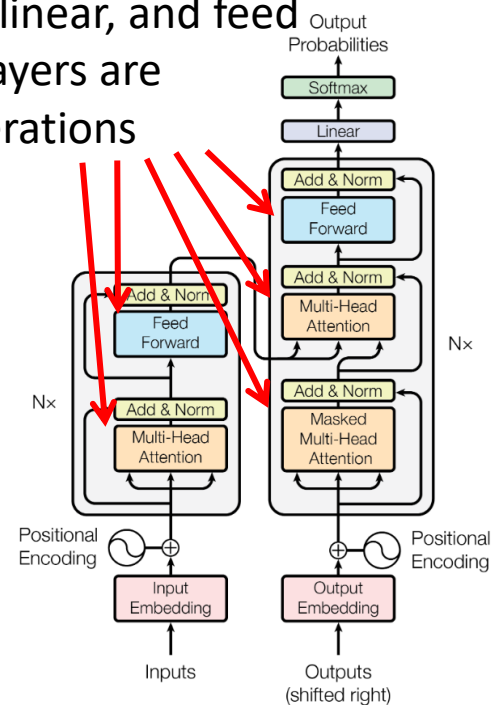


Recurrent  
Networks



Convolutional  
Networks

Attention, linear, and feed forwards layers are matrix operations



[Vaswani+, "Attention is all You Need", arXiv]

Transformer  
Networks

**How should we compute matrix operations?**

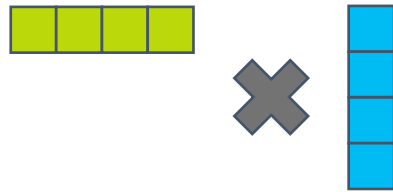
# Primitives for Matrix Operations

## Primitive

## Input Reuse

## Natural Fit For

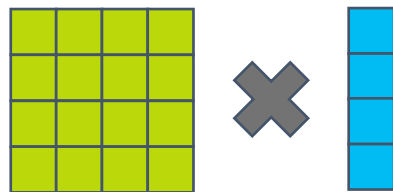
Vector \*  
Vector



*Left:  $O(1)$*   
*Right:  $O(1)$*

Convolutional Layers  
Recurrent Layers  
Fully-Connected Layers

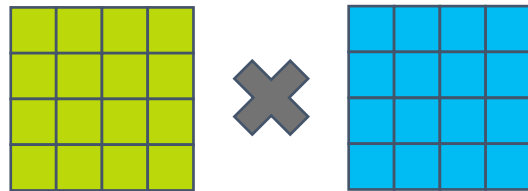
Matrix \*  
Vector



*Left:  $O(1)$*   
*Right:  $O(n)$*

Convolutional Layers  
Recurrent Layers  
Fully-Connected Layers

Matrix \*  
Matrix



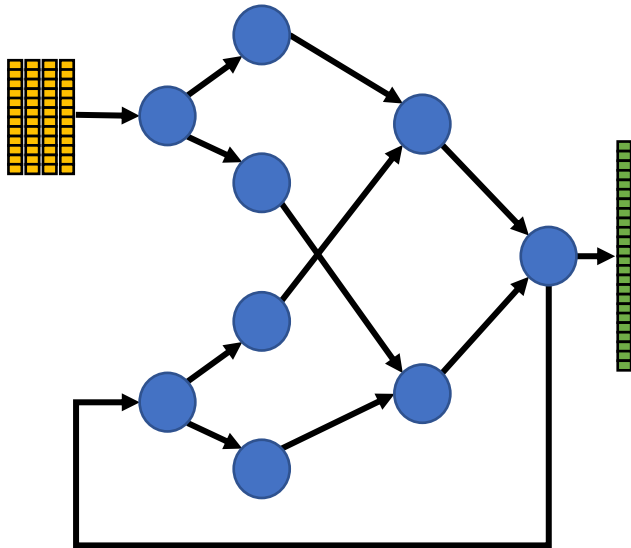
*Left:  $O(n)$*   
*Right:  $O(n)$*

Convolutional Layers  
**Batched** Recurrent Layers  
**Batched** Fully-Connected Layers

**Brainwave uses Matrix-Vector Multiply**

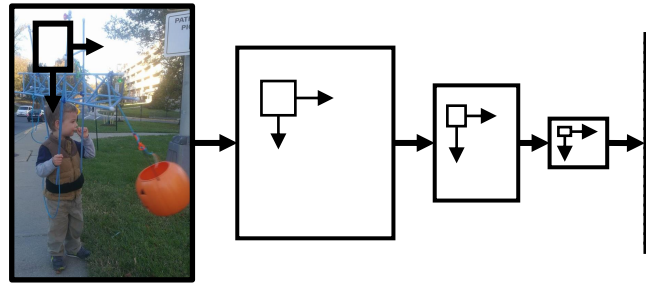
# What Else Has to Run on the FPGA?

Non-linear Activation Functions  
Vector-vector operations



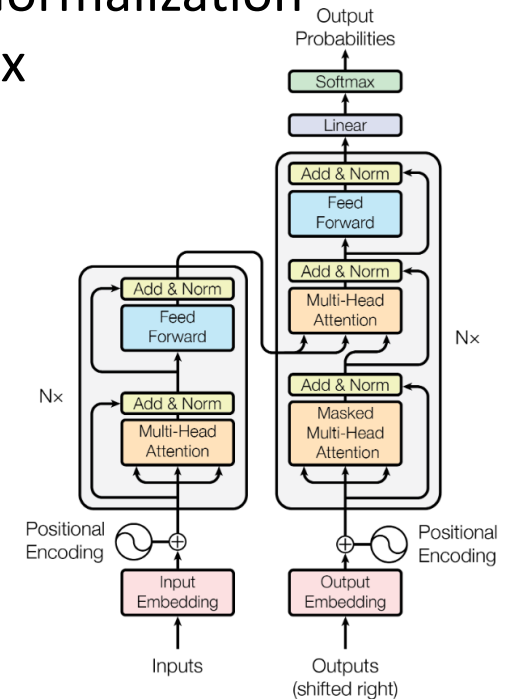
Recurrent  
Networks

Pooling operations  
Batch Normalization



Convolutional  
Networks

Layer Normalization  
SoftMax



[Vaswani+, "Attention is all You Need", arXiv]

Transformer  
Networks

**Neural networks are not just matrix multiply**



# Brainwave Overlay Design Principles

## Objectives

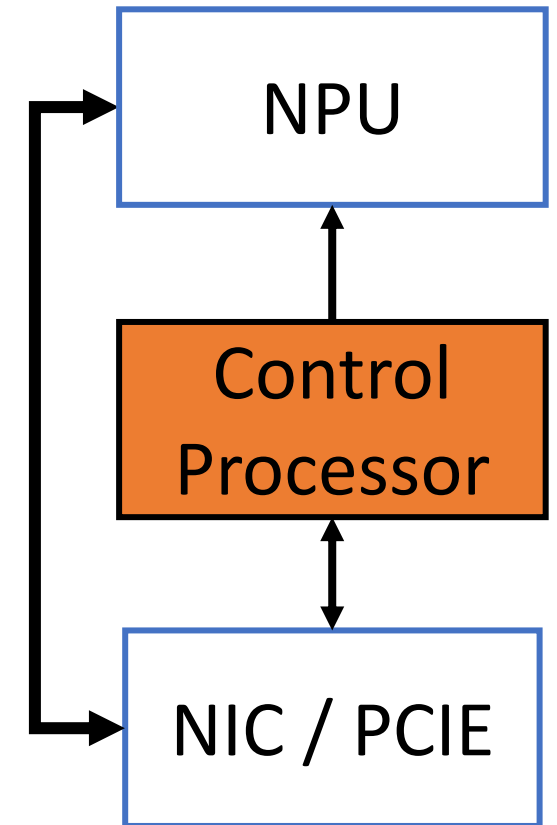
- Fast inferencing without batching
- Simple programmability with a single thread of control

## Balance NPU complexity, instruction granularity, and flexibility

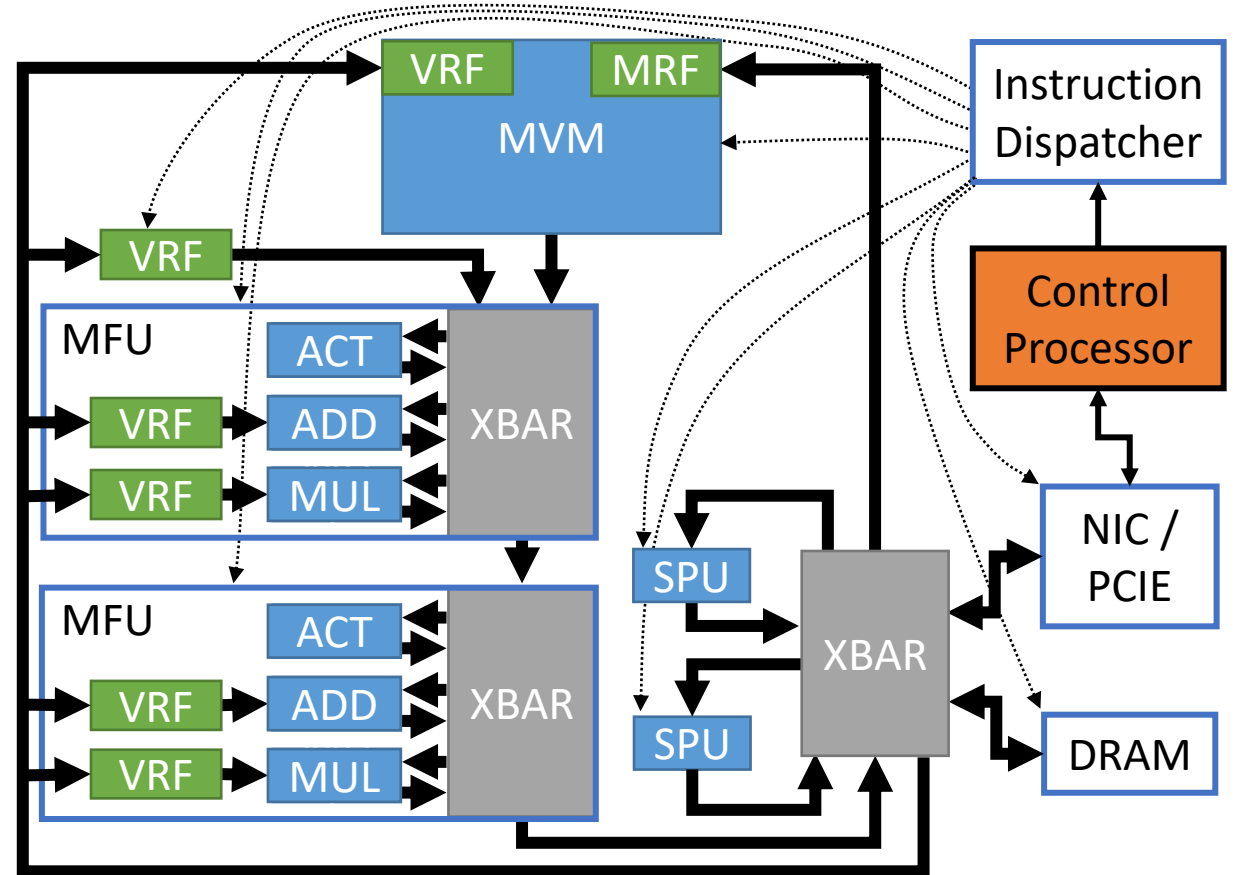
- All instructions operate on vectors of some native dimension
- Compute model is MVM and vector operations
- Neural networks decomposed into these operations

## Instruction Chaining

- Optimizes for matrix operation followed by vector operations
- Reduces the need for dependency analysis and multi-ported register files
- Allows a compact instruction encoding



# Brainwave Overlay Microarchitecture



VRF	Vector Register File	MFU	Vector Multi-Function Unit
MRF	Matrix Register File	ACT	Activation Unit
MVM	Matrix Vector Multiply	ADD	Vector Elementwise Add
SPU	Special Purpose Unit	MUL	Vector Hadamard Product

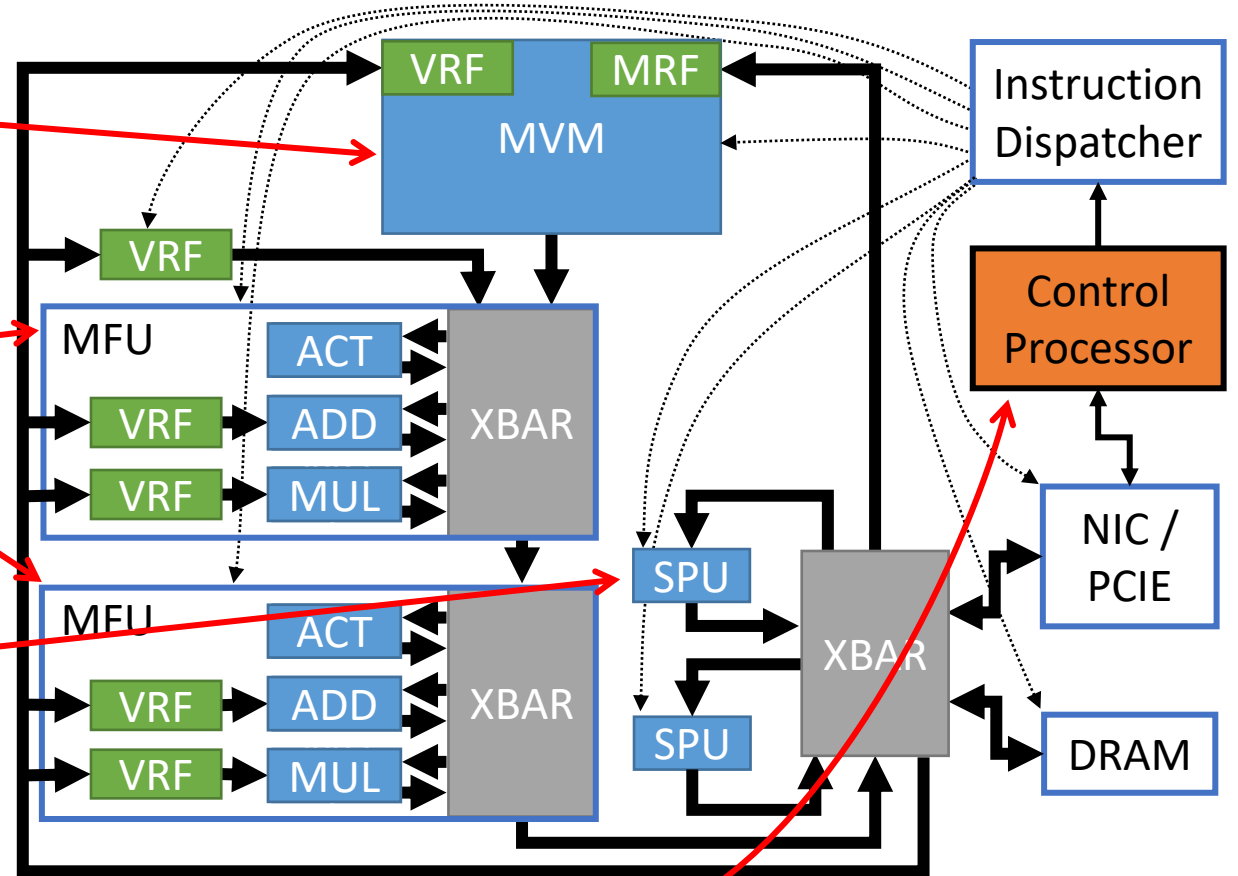
# Brainwave Overlay Microarchitecture

Most of the computation is in the MVM

Data flows from the MVM to MFUs that containing several compute units that can be used in any order

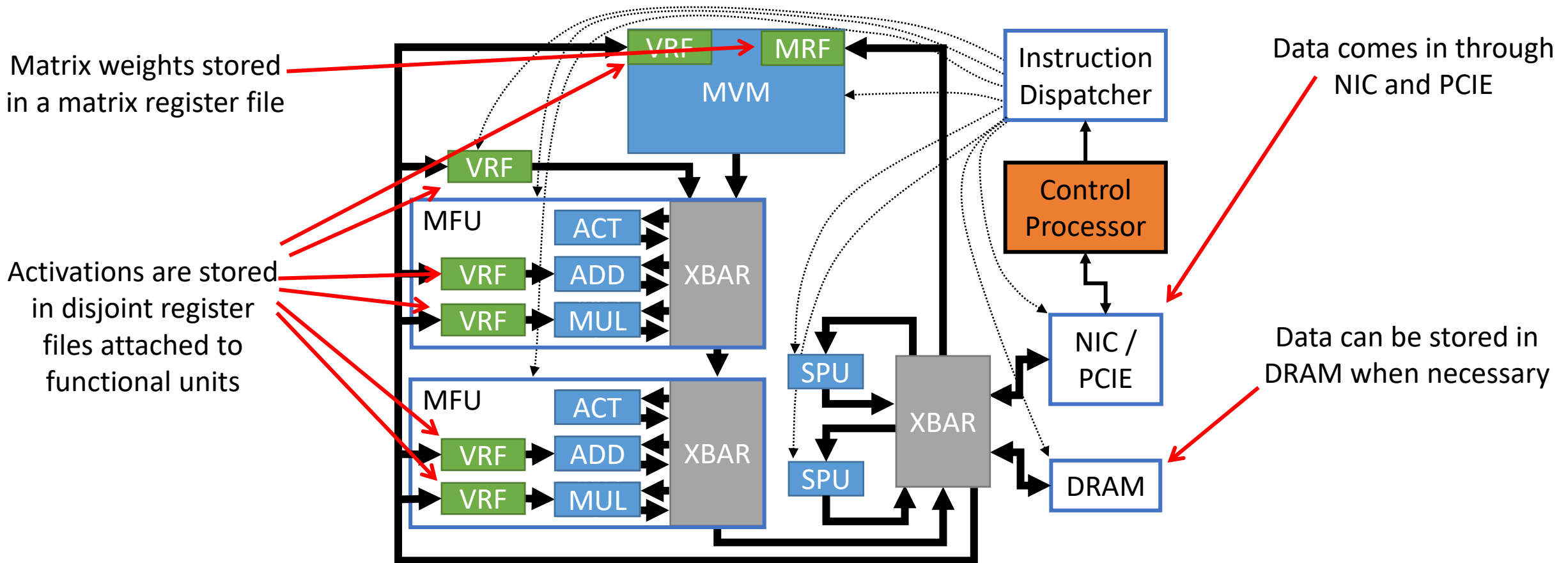
Special processing units allow sub-vector operations

Control processor runs model-specific firmware



VRF	Vector Register File	MFU	Vector Multi-Function Unit
MRF	Matrix Register File	ACT	Activation Unit
MVM	Matrix Vector Multiply	ADD	Vector Elementwise Add
SPU	Special Purpose Unit	MUL	Vector Hadamard Product

# Brainwave Data Management

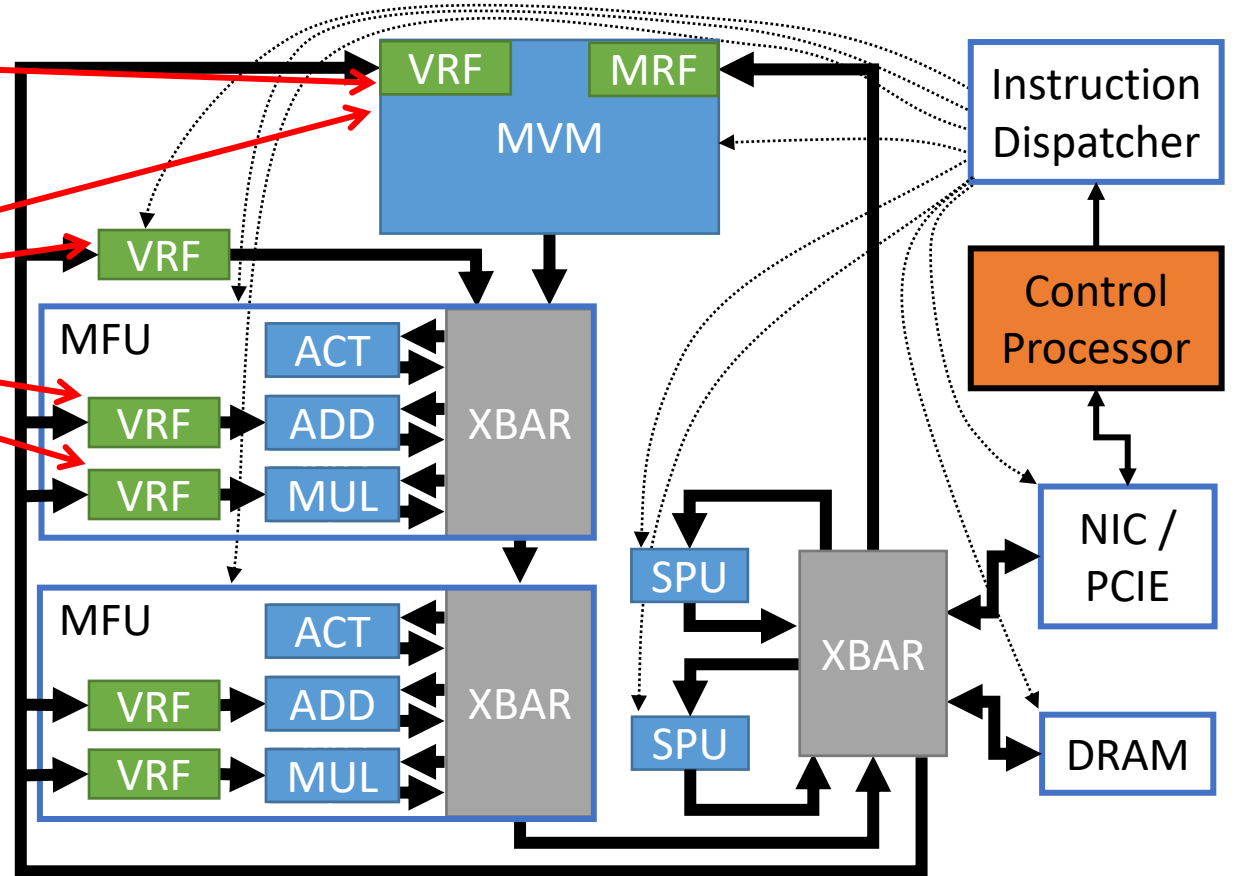


VRF	Vector Register File	MFU	Vector Multi-Function Unit
MRF	Matrix Register File	ACT	Activation Unit
MVM	Matrix Vector Multiply	ADD	Vector Elementwise Add
SPU	Special Purpose Unit	MUL	Vector Hadamard Product

# Overlay Specialization

Configurable parallelism – one large MVM or several parallel MVMs (with parallel vector data paths)

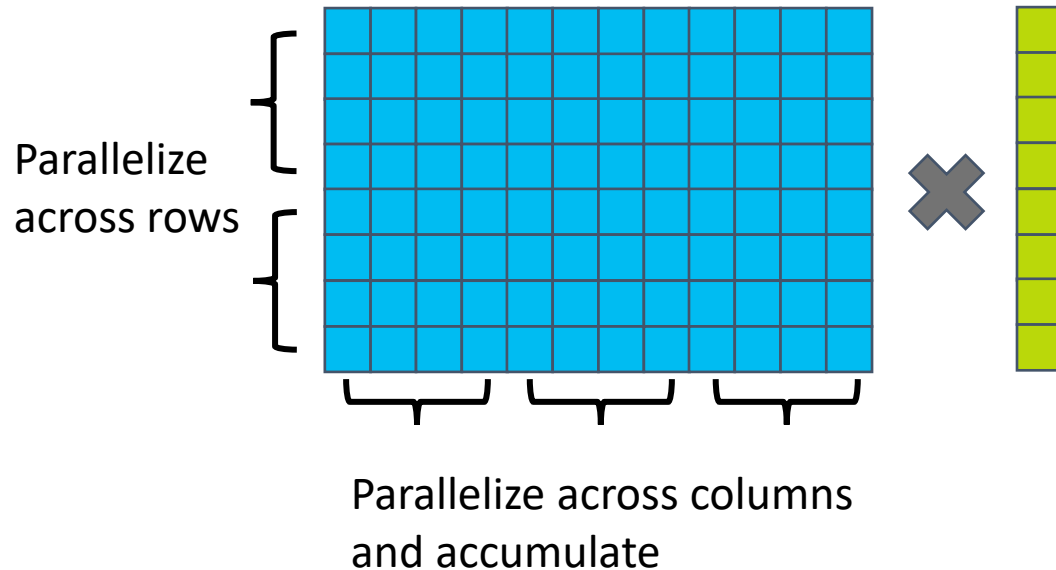
Vector sizes, element sizes, in-vector parallelization, VRF sizes are configurable



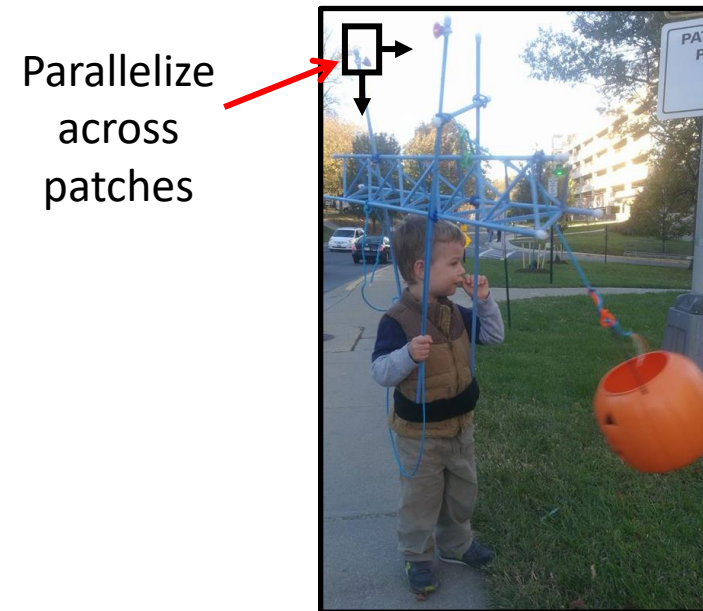
VRF	Vector Register File	MFU	Vector Multi-Function Unit
MRF	Matrix Register File	ACT	Activation Unit
MVM	Matrix Vector Multiply	ADD	Vector Elementwise Add
SPU	Special Purpose Unit	MUL	Vector Hadamard Product

# Optimizing for Different MVM Operations

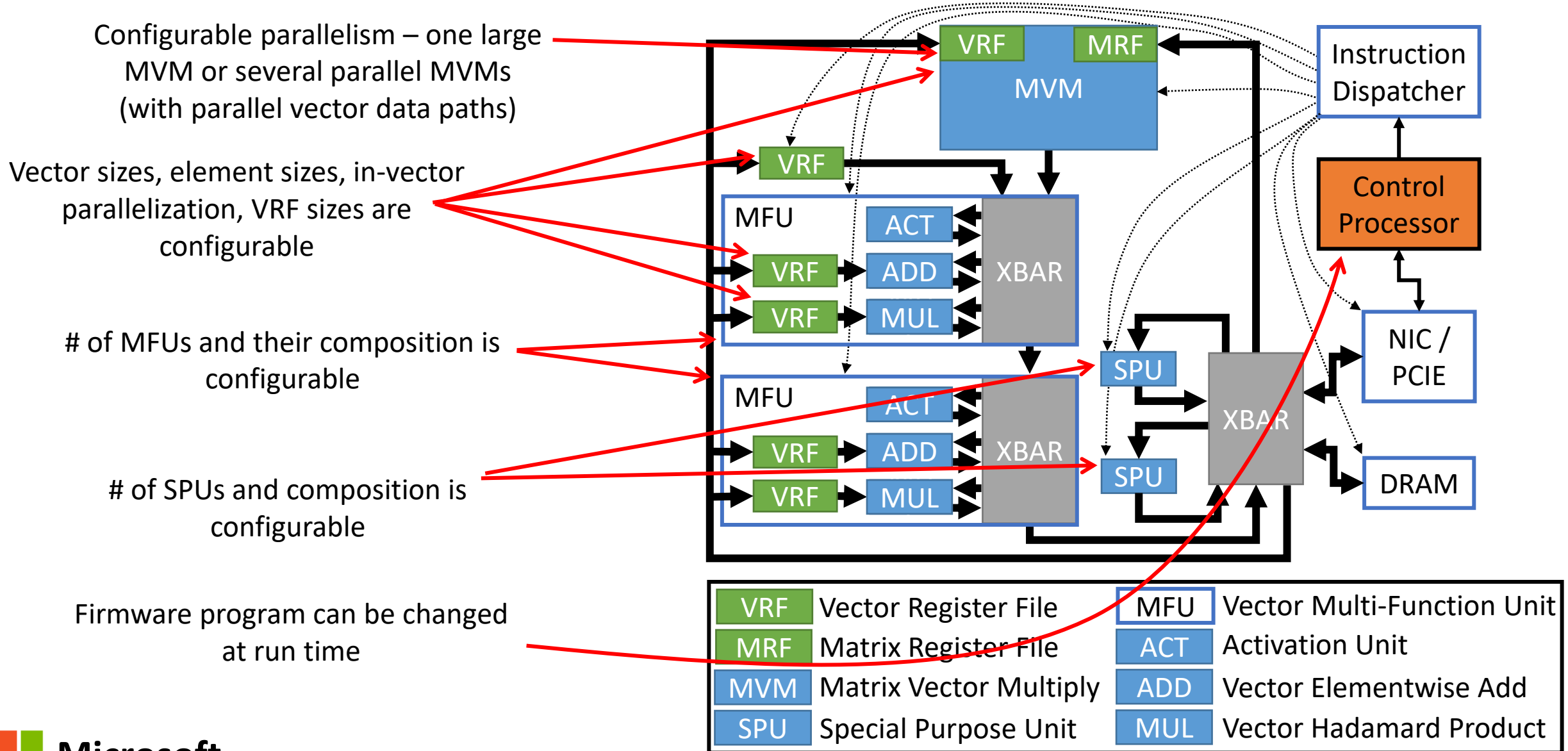
Recurrent network  
with large matrices



Convolutional network with  
many small filter operations



# Overlay Specialization



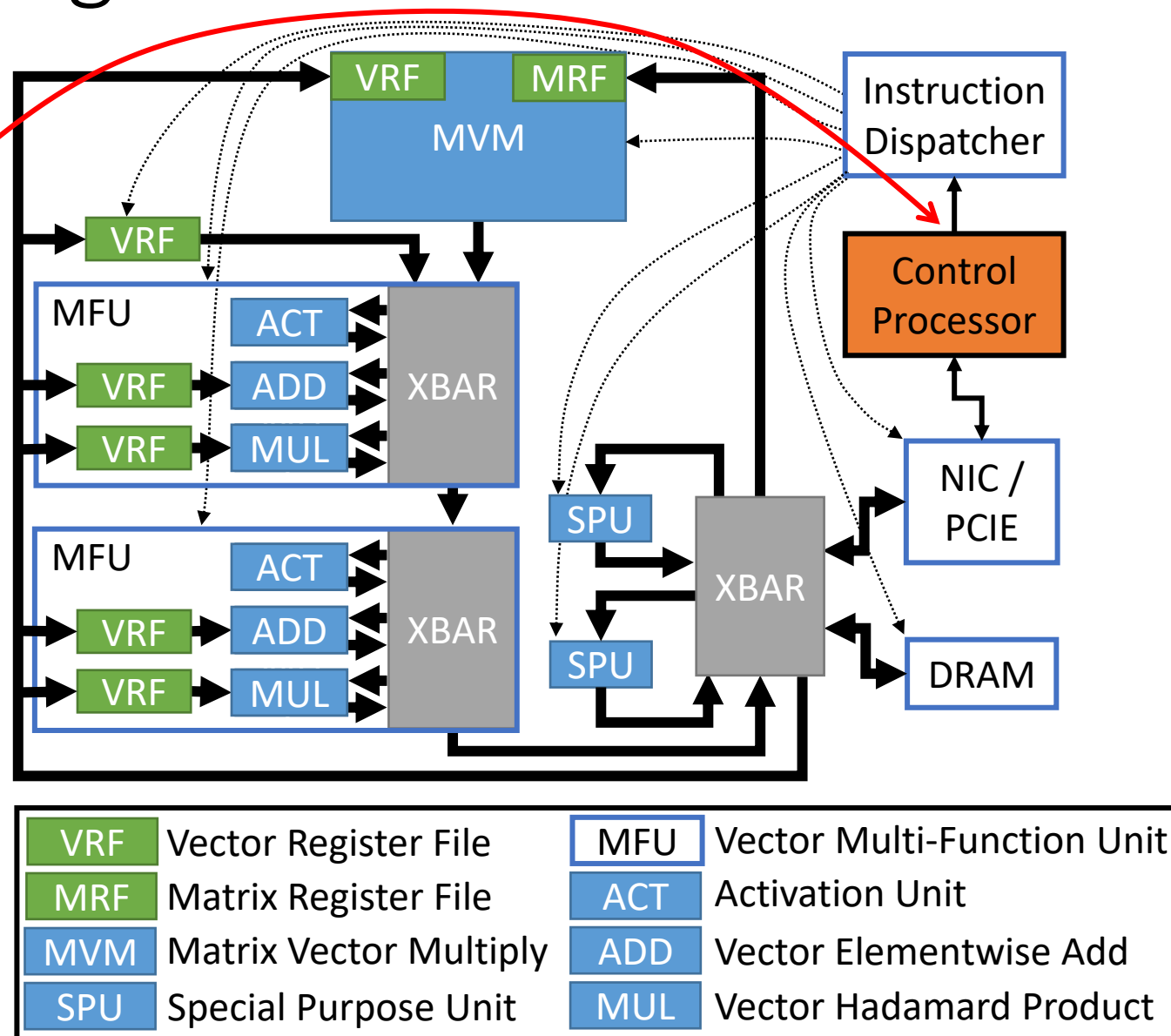


# Brainwave Firmware Programs

Firmware is a C program that runs on the control processor and makes the accelerator execute each particular neural network

Firmware manages control flow and data movements

Firmware maps the network's operations to chains of operations that the accelerator supports



# DNN Operators and Brainwave

## **Operations common in Deep Learning Networks**

LSTM	Scale
GRU	Max Pool
Convolution	Batch Norm
SoftMax	Sigmoid
Bias	TanH

## **Operations supported by the Brainwave accelerator**

MVM  
Vector add/sub/max  
Hadamard product  
Sigmoid  
TanH  
Square root  
Inverse

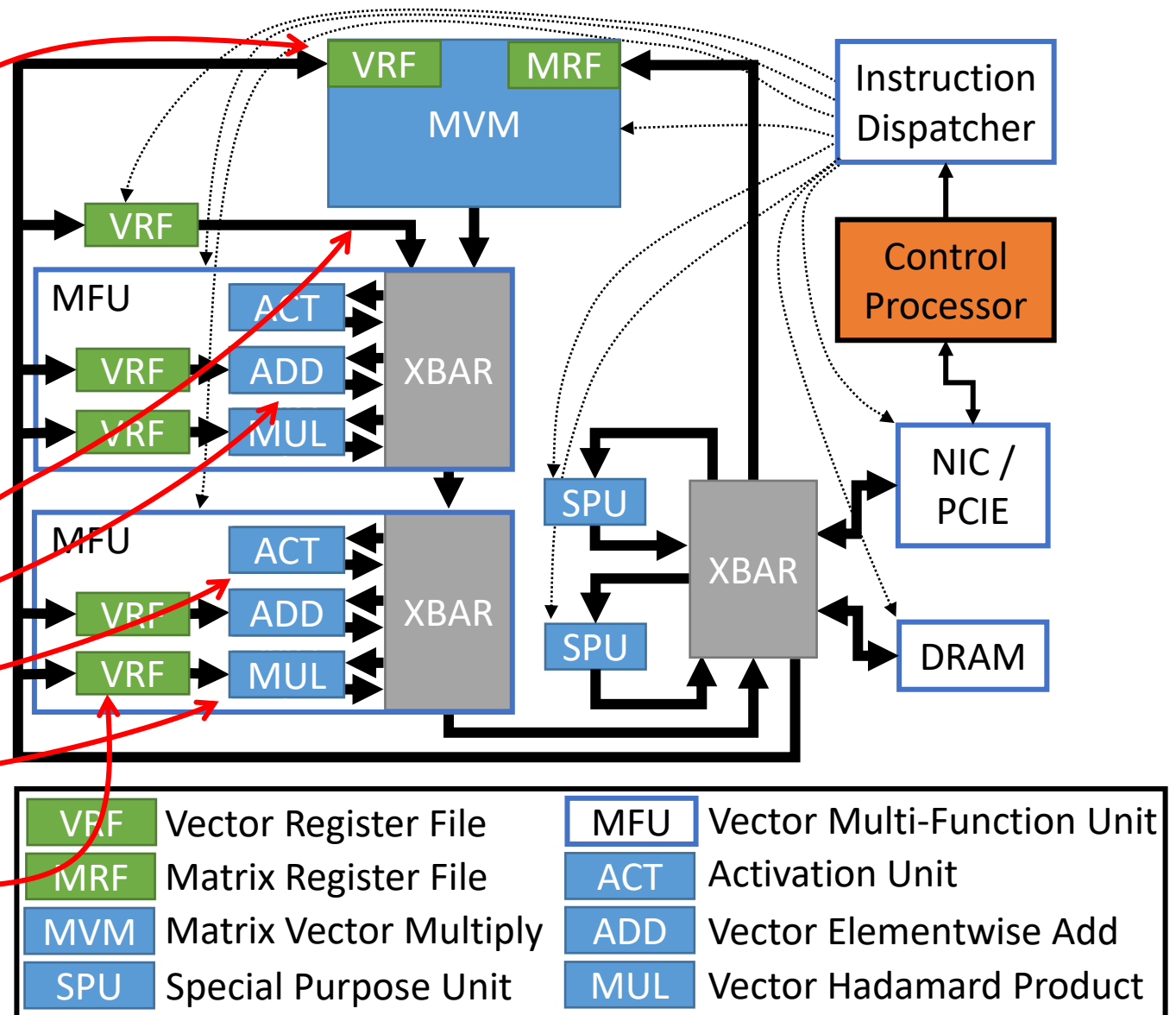
# LSTM Sketch in 29 Lines of Firmware Code

Loop over sequence	{	1. void LSTM(int steps) {	15. v_rd(InitialVrf, h_prev);	Process Hidden State	
Read Input		2. for (int t = 0; t < steps; t++) {	16. mv_mul(Uc);		
Process Current Input	{	3. v_rd(InputQ);	17. vv_add(xWc);		
		4. v_wr(InitialVrf, xt);	18. v_tanh();		
		5. v_rd(InitialVrf, xt);	19. vv_mul(it);		
		6. mv_mul(Wf);	20. vv_add(ft_mod);		
Process Hidden State	{	7. vv_add(bf);	21. v_wr(MultiplyVrf, c_prev);		Compute Next Hidden State and Output
		8. v_wr(AddSubVrf, xWf);	22. v_wr(InitialVrf, ct);		
		9. v_rd(InitialVrf, h_prev);	23. v_rd(InitialVrf, ct);		
		10. mv_mul(Uf);	24. v_tanh();		
		11. vv_add(xWf);	25. vv_mul(ot);		
		12. v_sigm();	26. v_wr(InitialVrf, h_prev);		
		13. vv_mul(c_prev);	27. }		
		14. v_wr(AddSubVrf, ft_mod);	28. v_wr(OutputQ);		
		29. }			

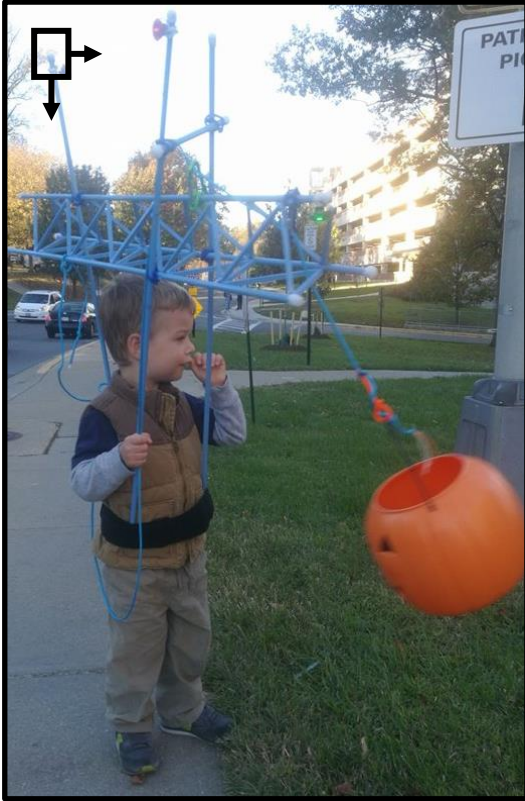
Firmware includes instruction chains that direct each functional unit

# Mapping LSTM Chains to the Accelerator

```
1. void LSTM(int steps) {  
2.   for (int t = 0; t < steps; t++) {  
3.     v_rd(InputQ);  
4.     v_wr(InitialVrf, xt);  
5.     v_rd(InitialVrf, xt);  
6.     mv_mul(Wf);  
7.     vv_add(bf);  
8.     v_wr(AddSubVrf, xWf);  
9.     v_rd(InitialVrf, h_prev);  
10.    mv_mul(Uf);  
11.    vv_add(xWf);  
12.    v_sigm();  
13.    vv_mul(c_prev);  
14.    v_wr(AddSubVrf, ft_mod);
```

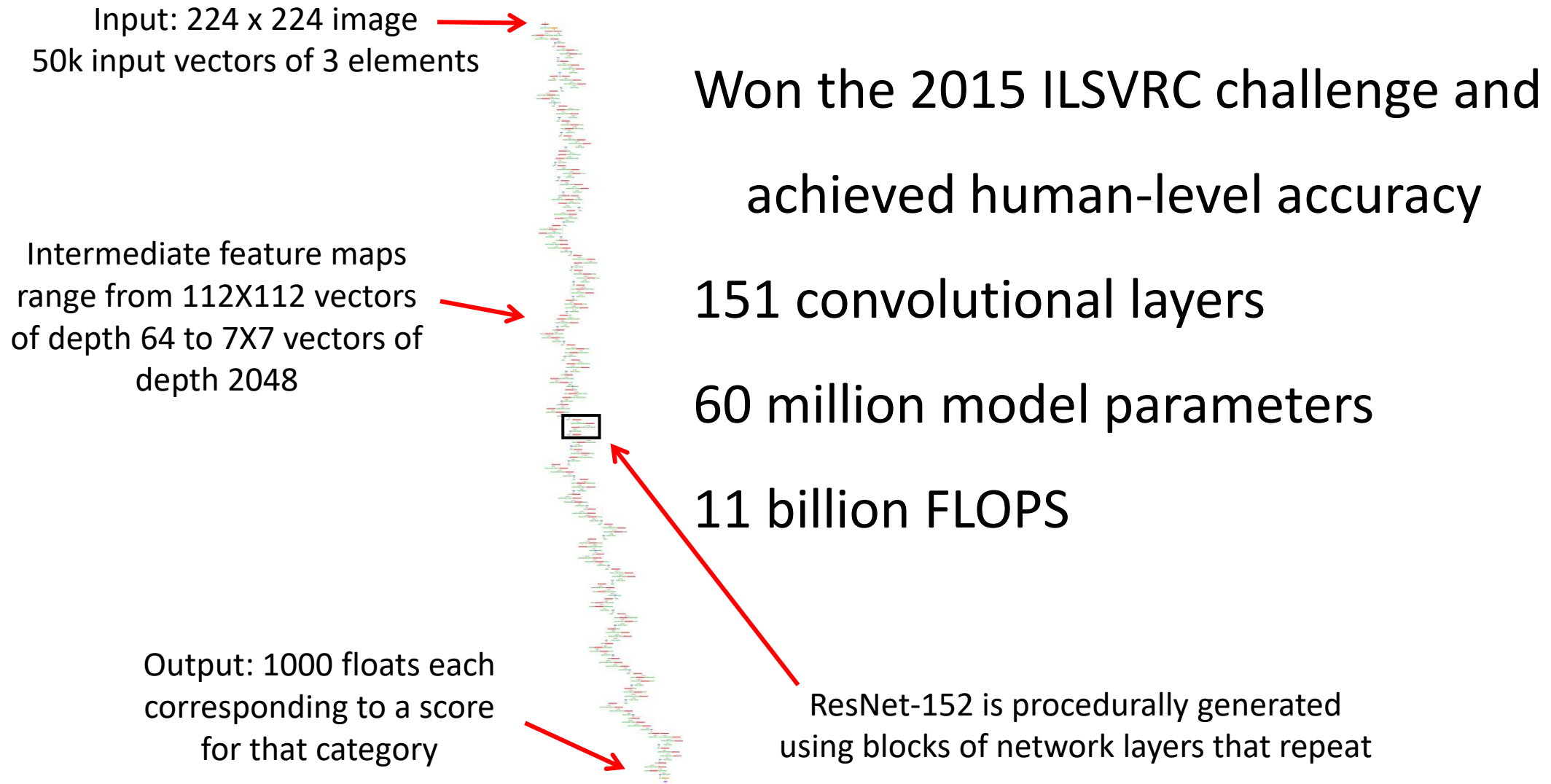


# Convolutional Networks



- Convolutions:
  - Convolutions slide a window over the image
  - The set of input data at each location is called a “patch”
  - The convolution computes a dot product between each patch and a set of filters.
  - The output of the convolution operation is a 2D array of vectors each containing one element per filter
- Batch normalization reduces the range of the activation values, reducing covariate shift
- Pooling operations reduce the size of the feature maps

# ResNet-152: A Convolutional Neural Network

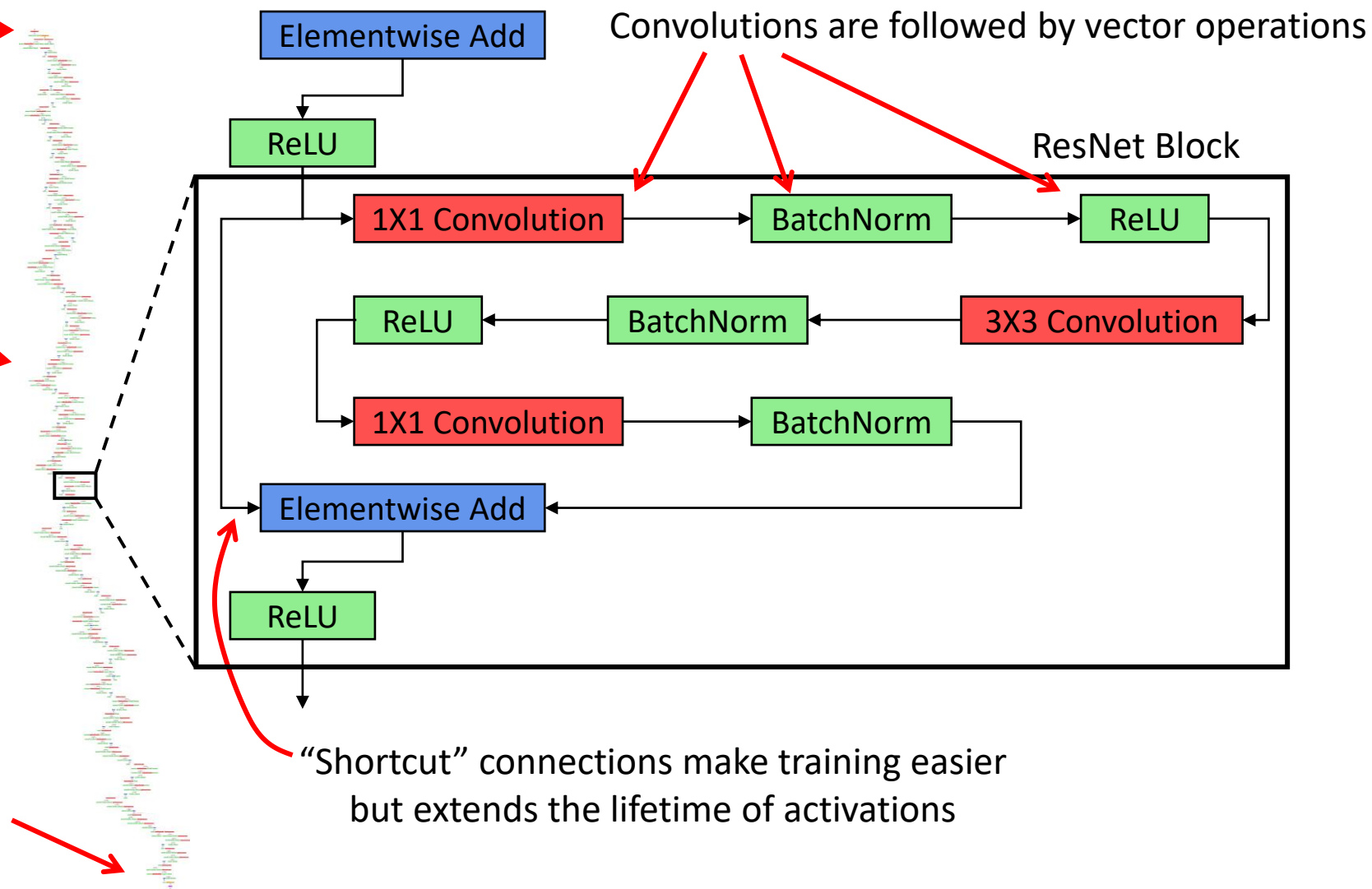


# “Res” = “Residual” Learning with Shortcuts

Input: 224 x 224 image  
50k input vectors of 3 elements

Intermediate feature maps  
range from 112X112 vectors  
of depth 64 to 7X7 vectors of  
depth 2048

Output: 1000 floats each  
corresponding to a score  
for that category



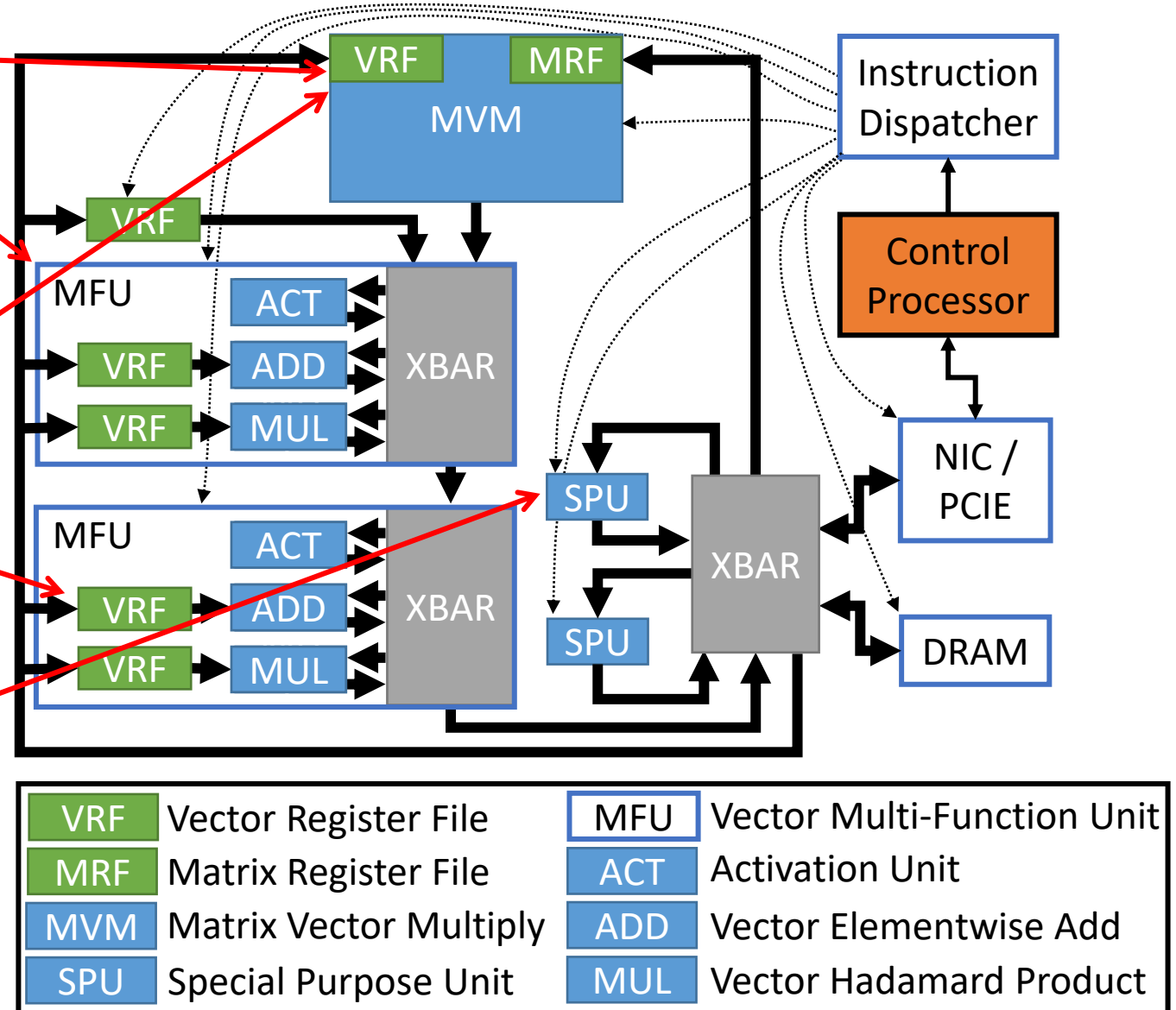


# Specializing Brainwave for ResNet-152

We configure the MVM and MFUs for parallel computations

We instantiate large vector register files to store convolution inputs and shortcut data

We instantiate an “image convolution patch generator” to process the input layer more efficiently

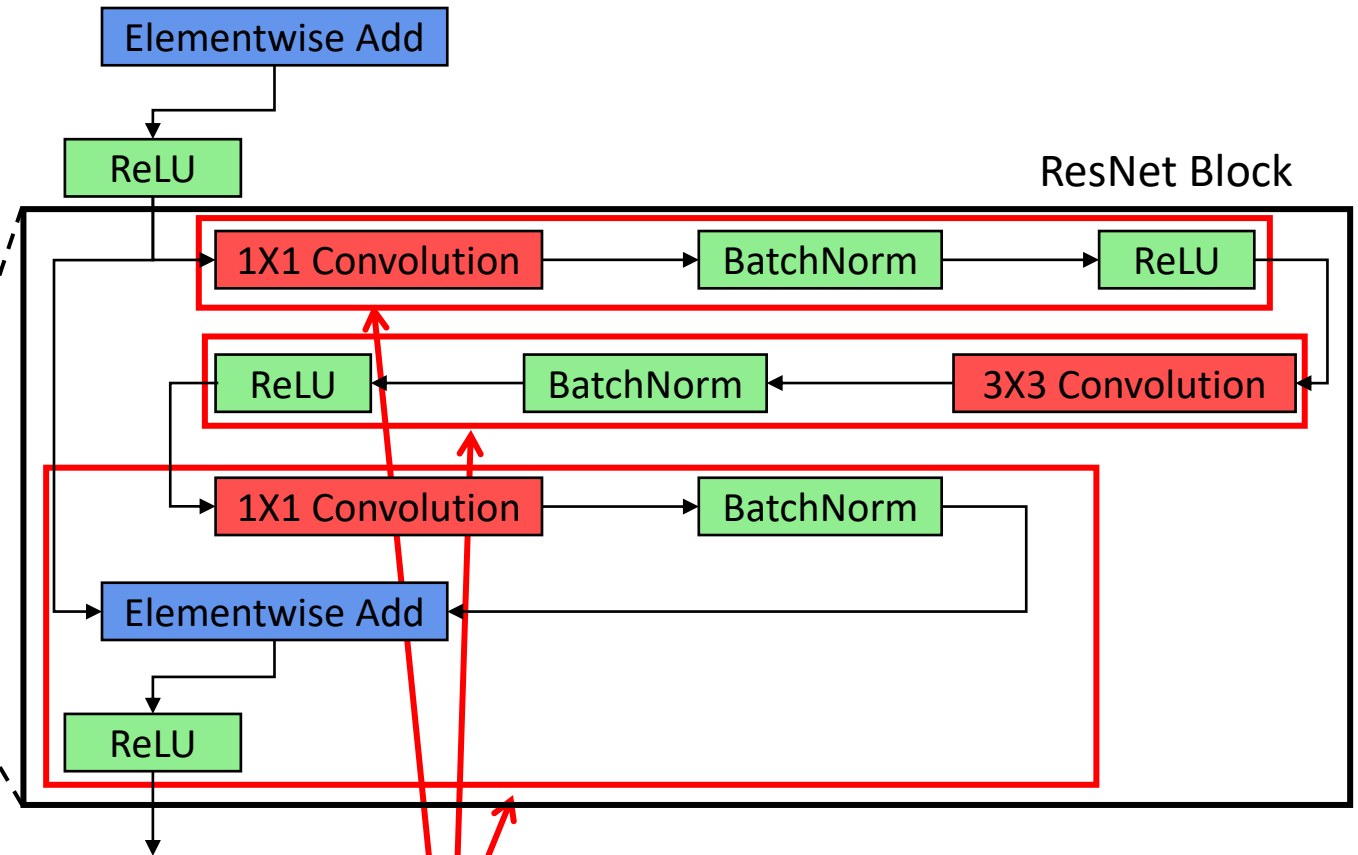


# Model Porting and Firmware Generation

Model compiler creates code for each operation in the model

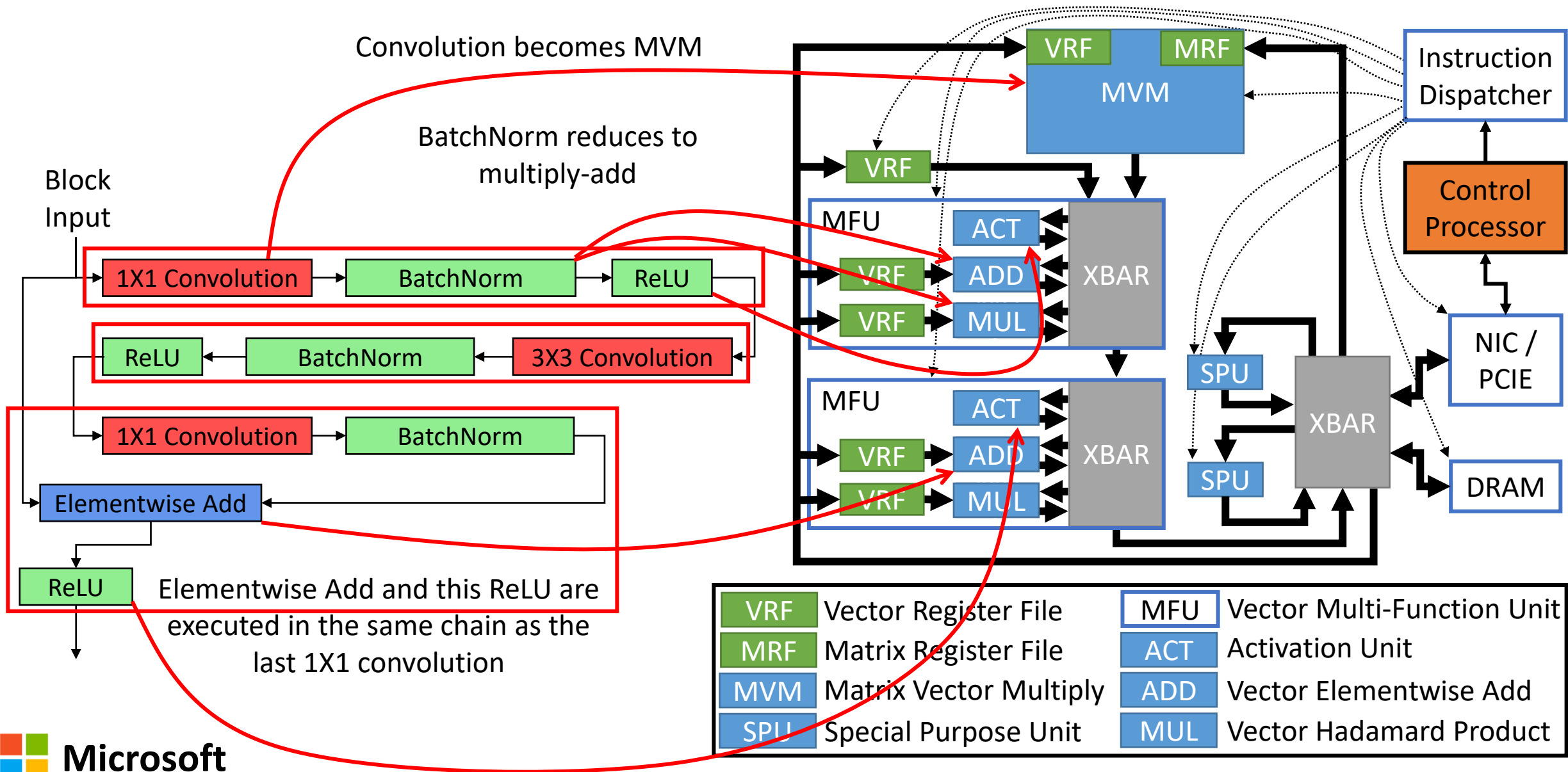
60 million model parameters don't fit in registers and are loaded in bulk from DRAM

Computations overlap with data transfers even within a block

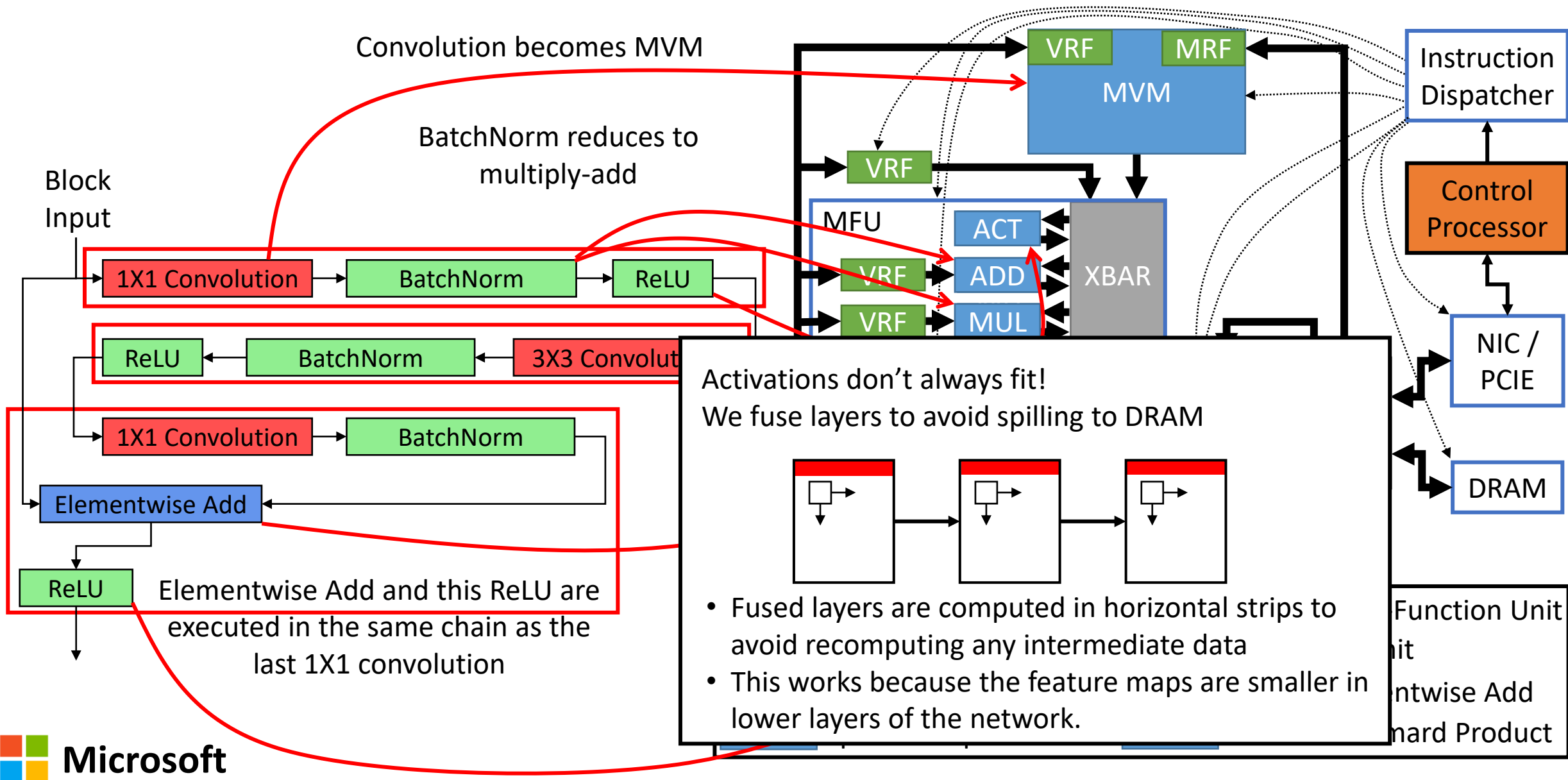


Operations are grouped into chains during code generation based on the accelerator configuration

# Mapping ResNet-152 to Brainwave



# Mapping ResNet-152 to Brainwave



# ResNet-152 and ResNet-50 Performance

- All convolution layers run on the FPGA
- Experiments use a batch size of 1
- Classifier runs on host computer
- Results are for the layers running on the FPGA and include data transfers
- Results on Arria 10 GX 1150 running at 300 MHZ

ResNet variant	ResNet-152	ResNet-50
Convolution Layers	151	49
Inference Latency (ms)	4	1.65
Top-1 Accuracy (%)	75.4	73.3
Reference Top-1 (%)*	77	75.3
Top-5 Accuracy (%)	92.4	91.1
Reference Top-5 (%)*	93.3	92.2

\* [[github.com/KaimingHe/deep-residual-networks](https://github.com/KaimingHe/deep-residual-networks)]

ResNet-152 reference results:

[Ma+ ISCAS 17]: 72 ms on the Arria 10 GX 1150

[Aziz+ HPCA 2019]: 35 ms on the Virtex-7 485T

ResNet-50 reference results:

[Chen+ FPGA 2019]: 8 ms on VU9P

Our experiments: 25% faster than an NVIDIA P40

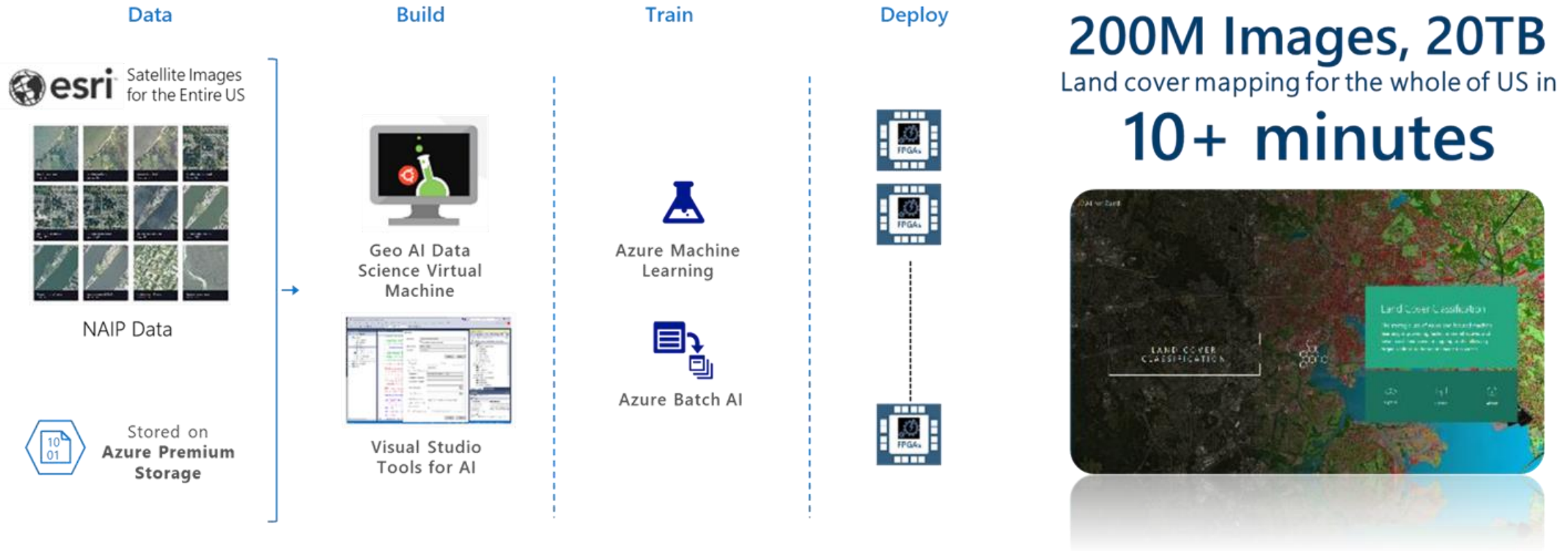
# FPGA-Accelerated CNNs in Azure

5 well-known convolutional neural networks

- ResNet-152
- ResNet-50
- DenseNet-121
- VGG-16
- SSD-VGG

The system includes an SDK, web-based GUI,  
and tutorials [<https://aka.ms/aml-real-time-ai>]

# Azure-Hosted ResNet-50 Based Land Classification



Created a national land cover map in about 10 minutes using \$42 of compute time

[<https://blogs.microsoft.com/green/2018/05/23/achievement-unlocked-nearly-200-million-images-into-a-national-land-cover-map-in-about-10-minutes/>]



# Azure-Hosted ResNet-50 for Particle Physics

## FPGA-accelerated machine learning inference as a service for particle physics computing

Javier Duarte · Philip Harris · Scott Hauck · Burt Holzman ·  
Shih-Chieh Hsu · Sergo Jindariani · Suffian Khan · Benjamin Kreis ·  
Brian Lee · Mia Liu · Vladimir Lončar · Jennifer Ngadiuba · Kevin  
Pedro · Brandon Perez · Maurizio Pierini · Dylan Rankin · Nhan  
Tran · Matthew Trahms · Aristeidis Tsaris · Colin Versteeg · Ted W.  
Way · Dustin Werran · Zhenbin Wu

Received: - / Accepted: -

**Abstract** Large-scale particle physics experiments face challenging demands for high-throughput comput-

ing resources both now and in the future. New heterogeneous computing paradigms on dedicated hardware with increased parallelization, such as Field Programmable Gate Arrays (FPGAs), offer exciting solutions with large potential gains. The growing applications of machine learning algorithms in particle physics

J.D., B.H., S.J., B.K., M.L., K.P., N.T., and A.T. are supported by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy.

[<https://arxiv.org/pdf/1904.08986.pdf>]

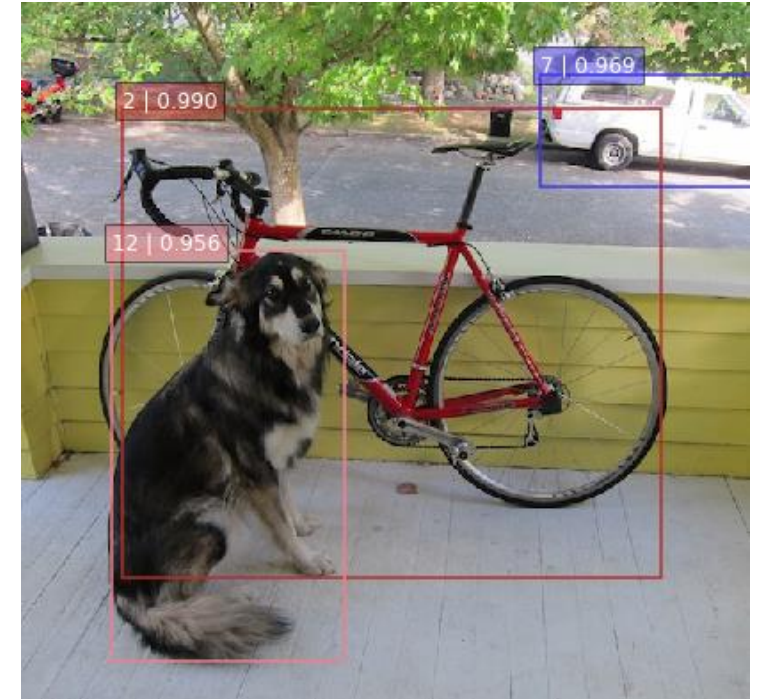
18 Apr 2019  
a-an]

# FPGA-Accelerated CNNs in Azure

## 5 well-known convolutional neural networks

- ResNet-152
- ResNet-50
- DenseNet-121
- VGG-16
- SSD-VGG

This one localizes objects in the image



The system includes an SDK, web-based GUI,  
and tutorials [<https://aka.ms/aml-real-time-ai>]

# SSD-VGG For Empty Shelf Detection at the Edge



KROGER CORPORATE > INVESTOR RELATIONS > PRESS RELEASES > PRESS RELEASE

## Kroger and Microsoft Partner to Redefine the Customer Experience and Introduce Digital Solutions for the Retail Industry

- America's largest grocery retailer and global technology company partnering to pilot two connected experience stores
- Companies will jointly bring to market Retail as a Service product for retailers and present the solution at NRF 2019: Retail's Big Show

Company Release - 1/7/2019 6:30 AM ET

CINCINNATI and REDMOND, Wash., Jan. 7, 2019 /PRNewswire/ -- The Kroger Co. (NYSE: KR) and Microsoft Corp. (Nasdaq: MSFT) today announced a collaboration to redefine the customer experience using Kroger Technology products powered by Microsoft Azure, the retailer's preferred cloud platform for Retail as a Service (RaaS). Through this innovative partnership, Kroger will pilot a connected store experience and together with Microsoft, jointly market a commercial RaaS product to the industry.

[<http://ir.kroger.com/file/Index?KeyFile=396285733>]

# Research Topics in DNN Inference Acceleration

- Number format
- Sparse Networks
- Dynamic Networks
- Overlay Sharing Between Models

# Takeaways

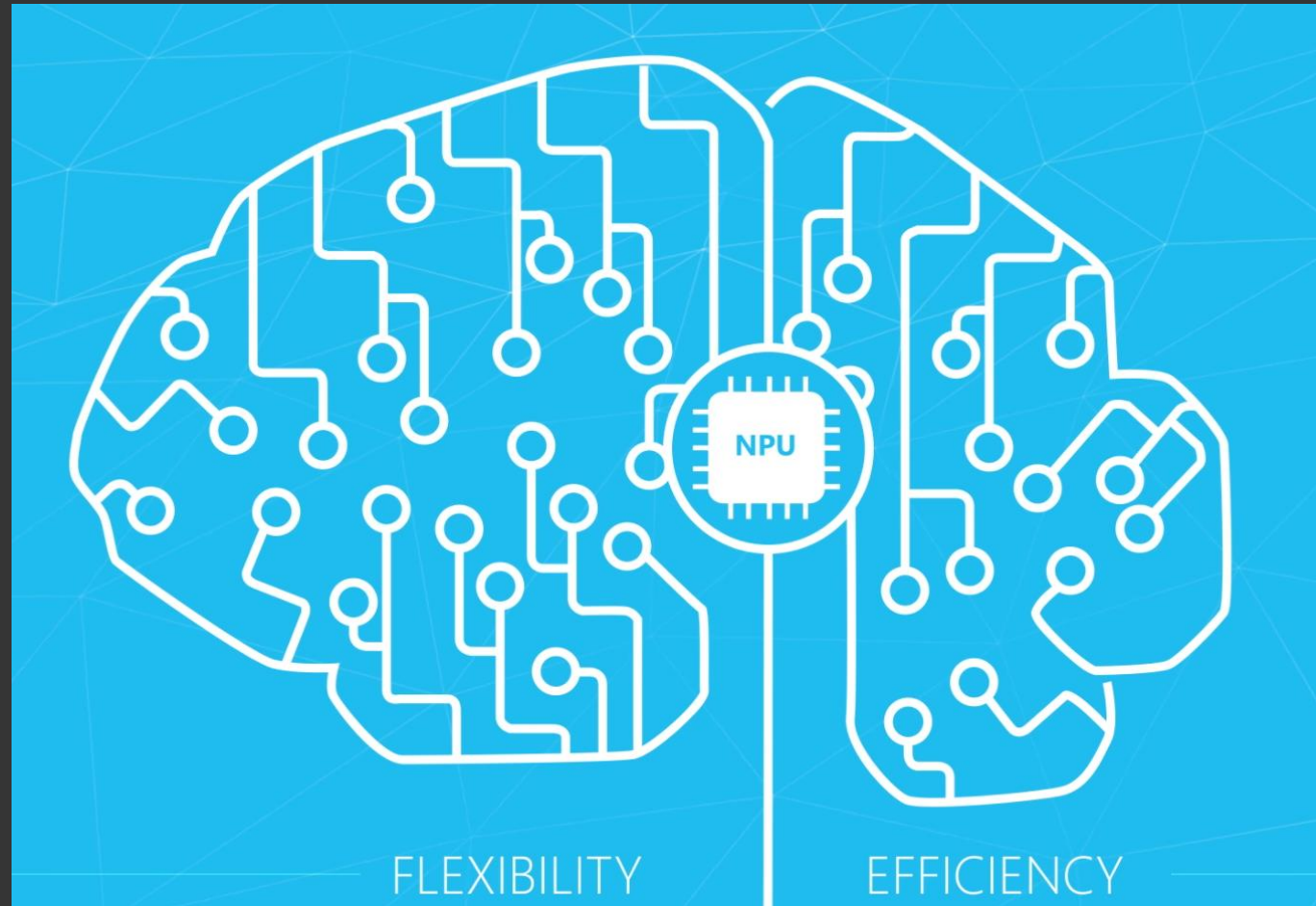
FPGAs are great for neural networks because we can specialize the overlay for the network and update the overlay in place

- Can switch any FPGA to a different configuration for load balancing
- Neural networks keep changing and FPGAs allow us to keep up

Brainwave is co-designed across hardware and software to take advantage of this flexibility to perform neural network inference at a massive scale for 1<sup>st</sup> party models on Bing and 3<sup>rd</sup> party models on Azure

Brainwave is still under development and we're scaling it to better FPGA hardware and bigger models





<https://www.microsoft.com/en-us/research/project/project-brainwave/>  
<https://www.microsoft.com/en-us/research/project/project-catapult/>  
<https://aka.ms/aml-real-time-ai>