CS 758: Advanced Topics in Computer Architecture

Lecture #13: Systolic

Professor Matthew D. Sinclair

Some of these slides were developed by Tushar Krishna at Georgia Tech
Slides enhanced by Matt Sinclair

Announcements

- **HW2 Due Friday** (tomorrow)
  - None of jobs should run for more than ~2 hours in baseline GPGPU-Sim
  - See Piazza for various suggestions
    - Especially: I suggest setting “stream_output” flag in your sub files so you can see output (e.g., stdout) as job progresses
      - Alternatives (e.g., condor_tail) on Piazza
- Doug Berger Talk 3 PM on Friday (tomorrow)

Question for Today

- Why Systolic Arrays for GEMM / CNN?

CONV Layer Tensor Computation

Output fmaps (O) Biases (B) Input fmaps (I) Filter weights (W)

$$\mathbf{O}[n][m][x][y] = \mathrm{Activation}(\mathbf{B}[m] + \sum_{i=0}^{R-1} \sum_{j=0}^{S-1} \sum_{k=0}^{C-1} \mathbf{I}[n][k][Ux+i][Uy+j] \times \mathbf{W}[m][k][i][j]),$$

$$0 \le n < N, 0 \le m < M, 0 \le y < E, 0 \le x < F,$$

$$E = (H - R + U)/U, F = (W - S + U)/U.$$

Shape Parameter	Description
N	fmap batch size
M	# of filters / # of output fmap channels
C	# of input fmap/filter channels
H/W	input fmap height/width
R/S	filter height/width
E/F	output fmap height/width
U	convolution stride

Google TPU

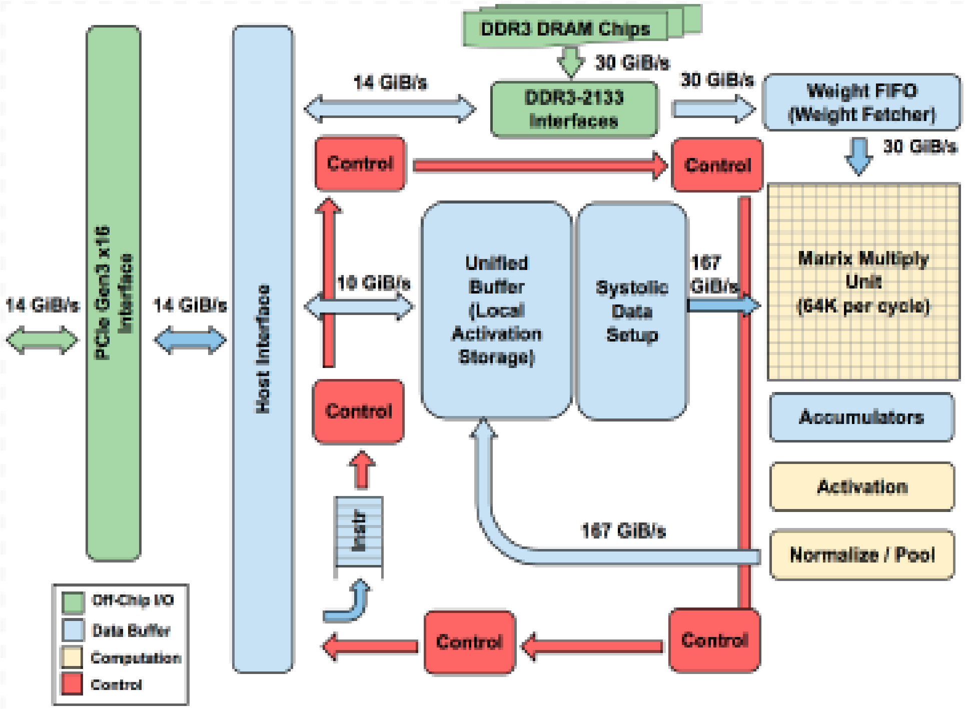
- Why ASIC chip?
- Specs
  - 256x256 = 64K 8-bit MAC
  - Peak throughput: 92 TOPS
  - Software Managed On-Chip Memory: 28 MB
- Highlights
  - 15-30X faster than K80 GPU for Inference
  - Bandwidth limited for 6 out of 8 workloads

Name	*LOC*	*Layers*					*Nonlinear function*	*Weights*	*TPU Ops / Weight Byte*	*TPU Batch Size*	*% of Deployed TPUs in July 2016*
		FC	*Conv*	*Vector*	*Pool*	*Total*					
MLP0	100	5				5	ReLU	20M	200	200	61%
MLP1	1000	4				4	ReLU	5M	168	168	
LSTM0	1000	24		34		58	sigmoid, tanh	52M	64	64	29%
LSTM1	1500	37		19		56	sigmoid, tanh	34M	96	96	
CNN0	1000		16			16	ReLU	8M	2888	8	5%
CNN1	1000	4	72		13	89	ReLU	100M	1750	32	

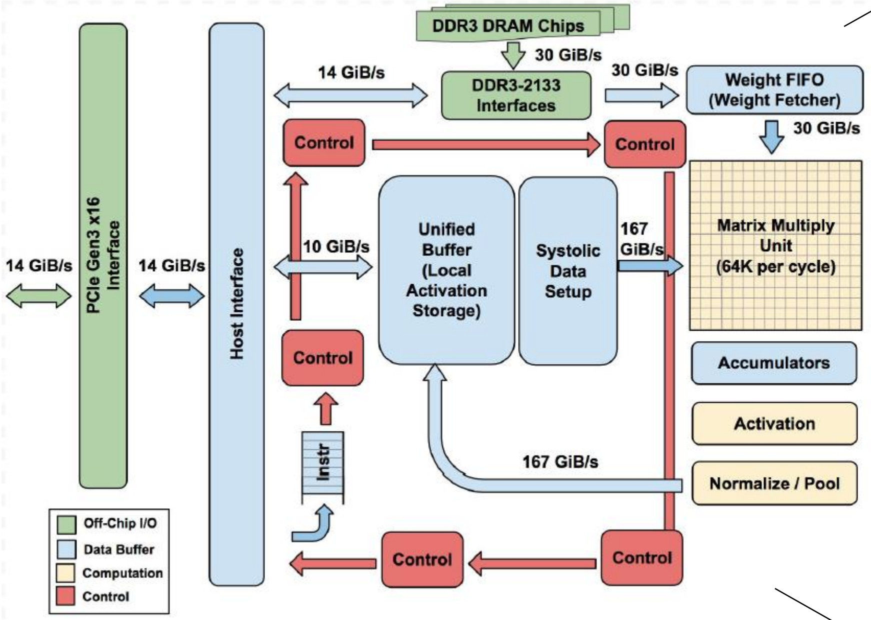
Table 1. Six NN applications (two per NN type) that represent 95% of the TPU's workload. The columns are the NN name; the number of lines of code; the types and number of layers in the NN (FC is fully connected, Conv is convolution, Vector is self-explanatory, Pool is pooling, which does nonlinear downsizing on the TPU; and TPU application popularity in July 2016. One DNN is RankBrain [Cla15]; one LSTM is a subset of GNM Translate [Wu16]; one CNN is Inception; and the other CNN is DeepMind AlphaGo [Sil16][Jou15].

Google TPU Summary

- Focus: **supervised** learning
- <10 CISC instructions!
- Lots and lots of MACs
  - Reduced precision (8/16 bits)
  - Common for ML workloads
- Large memories
  - Bandwidth is the key limiter
  - MACs only saturated with > 1000 batch size
  - Accumulators locally collect values



TPU Microarchitecture



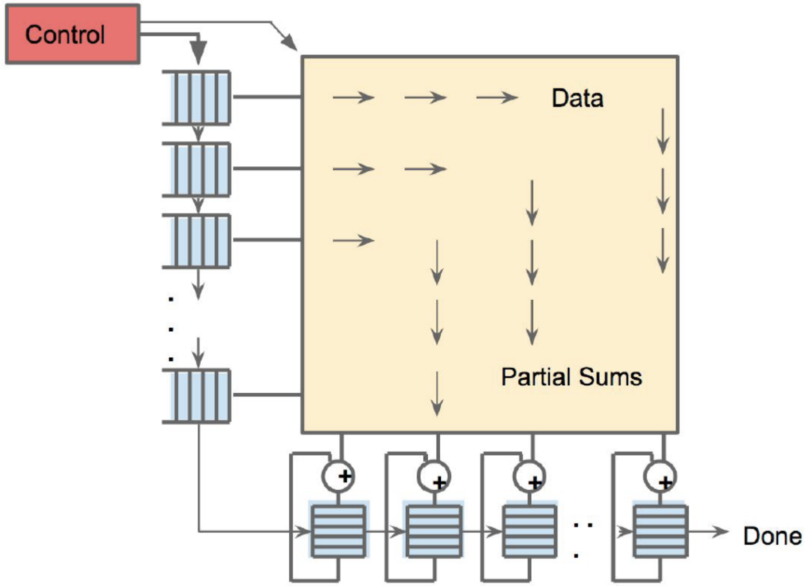
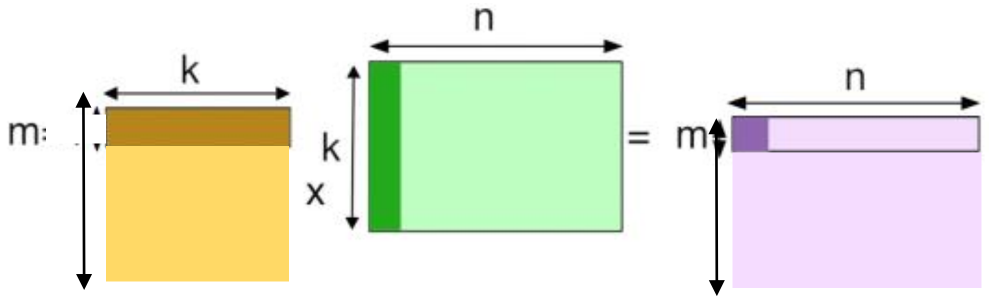


Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Why systolic arrays?

- Compute vs I/O
- How many computations are required to compute each matrix element?



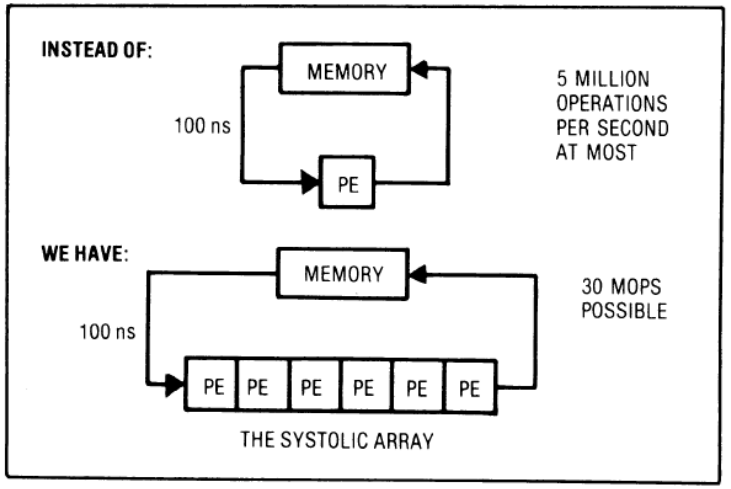
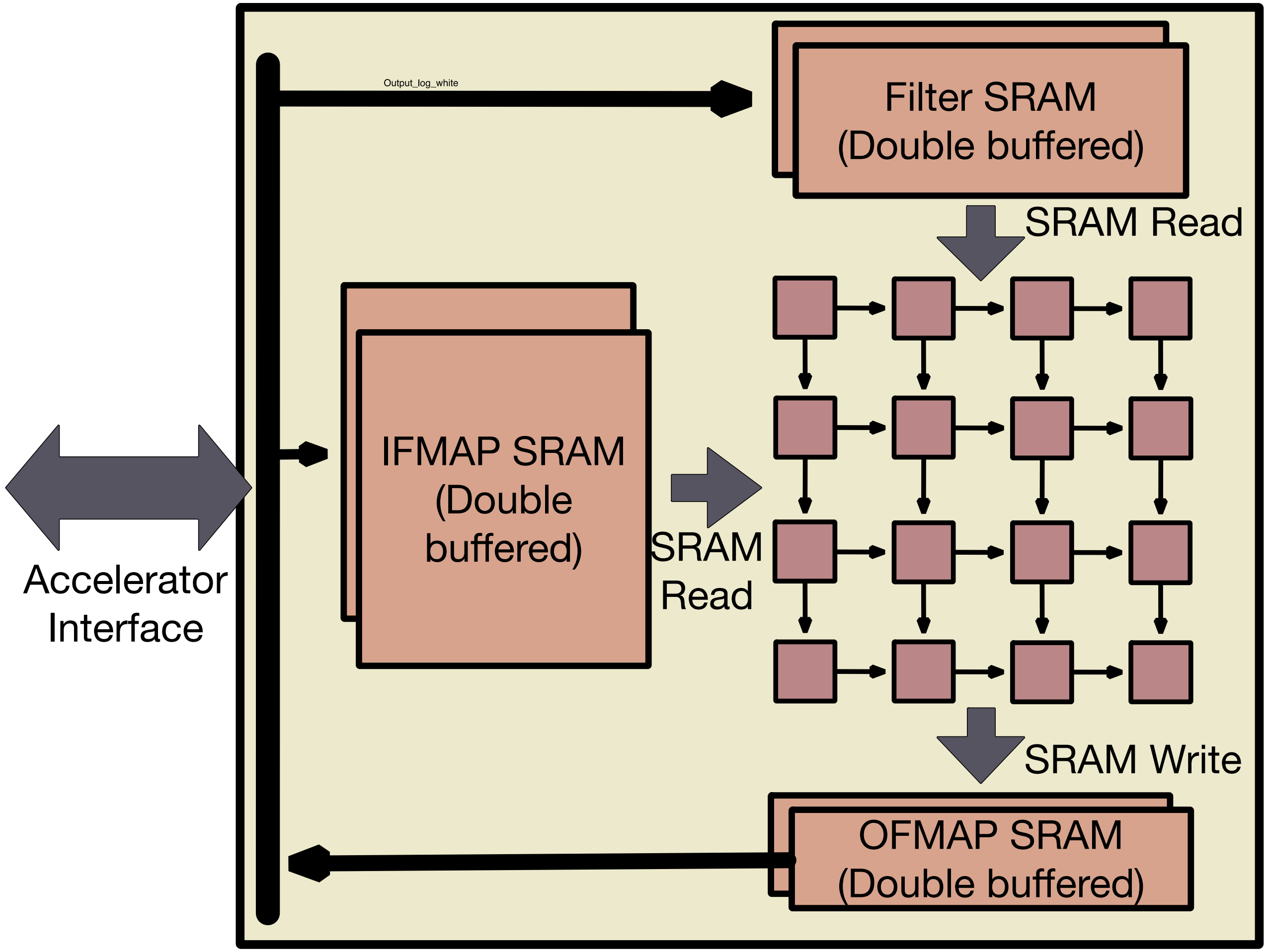


Figure 1. Basic principle of a systolic system.

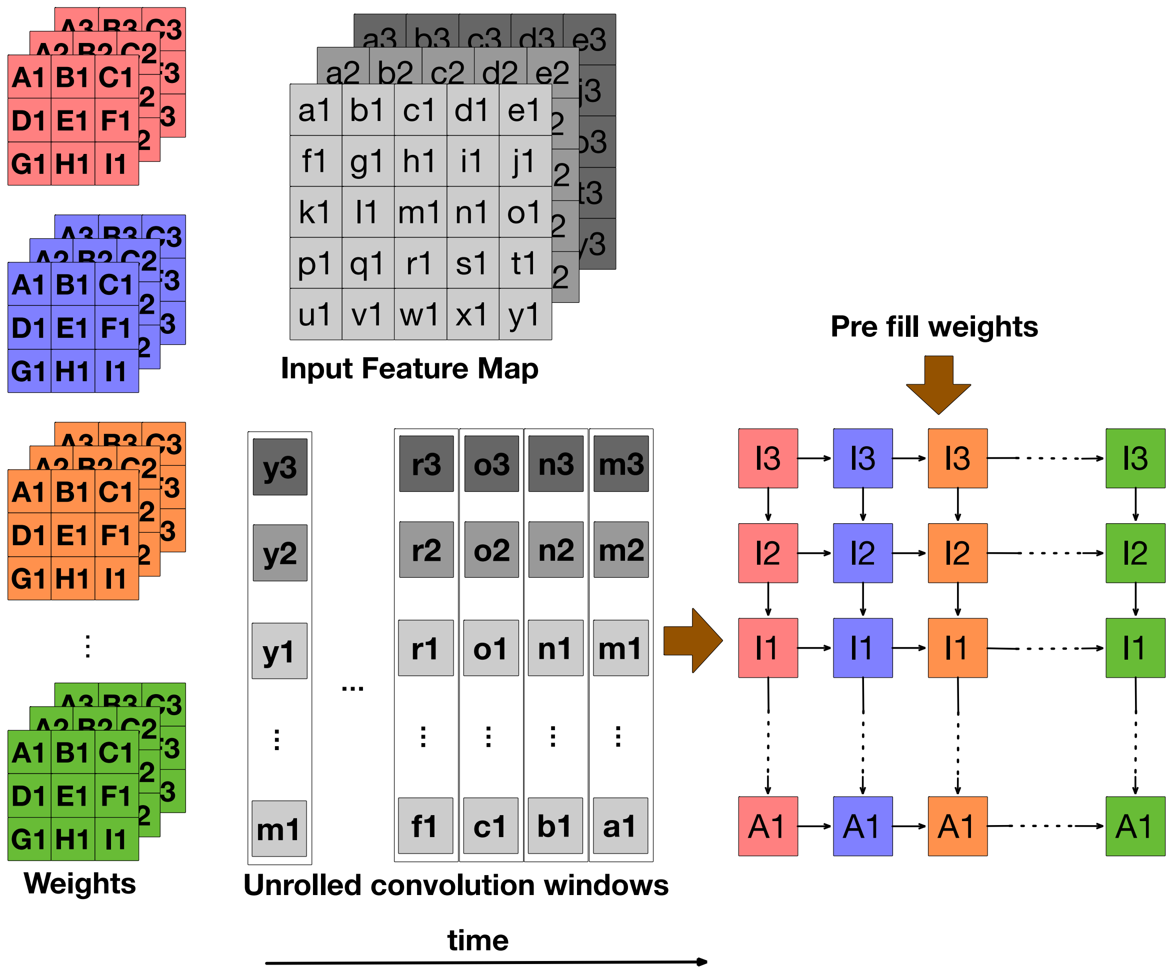
Source: HT Kung, 1982

How does a systolic array work?



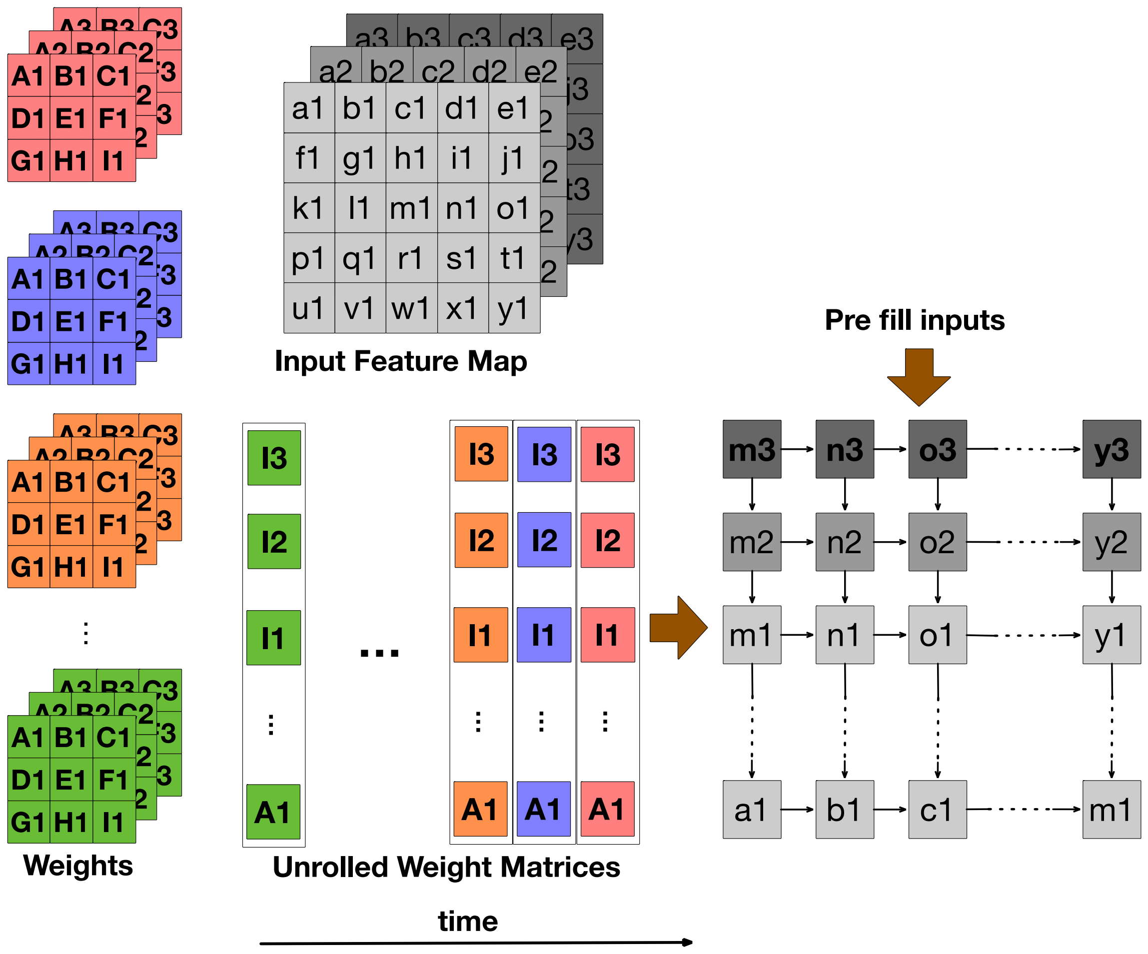
Design 1

- weights stationary, broadcast inputs, move results



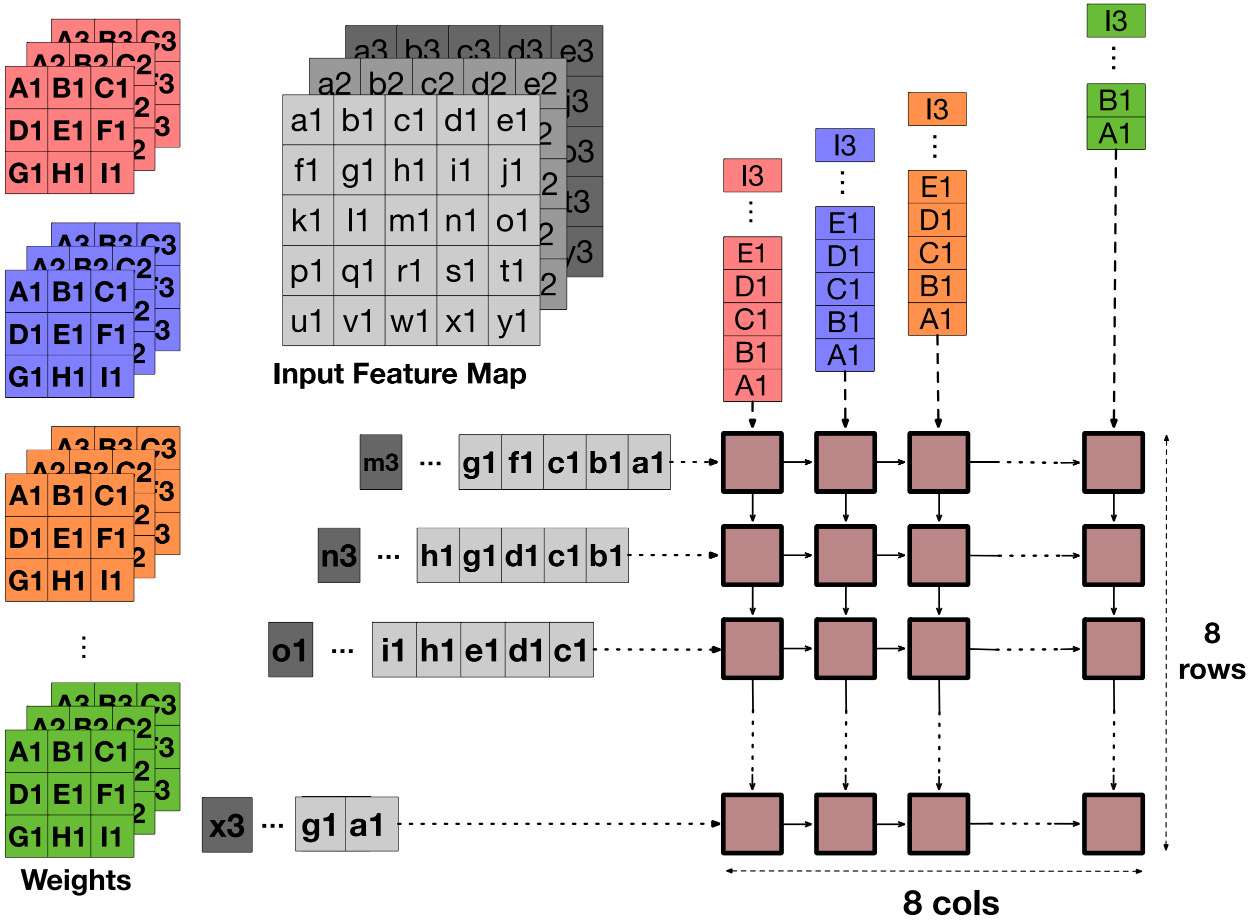
Design 2

- Inputs stationary, broadcast weights, move results

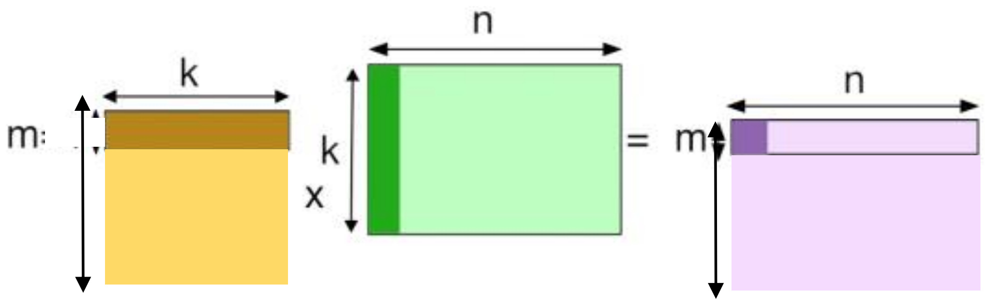


Design 3

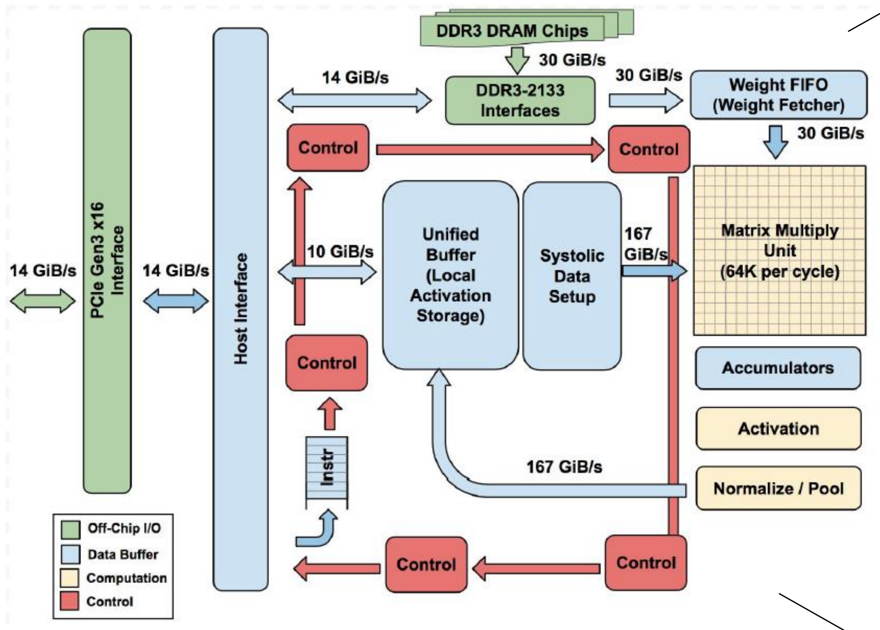
- Outputs stationary, broadcast weights, broadcast inputs



What about MLP and LSTMs?



TPU Microarchitecture



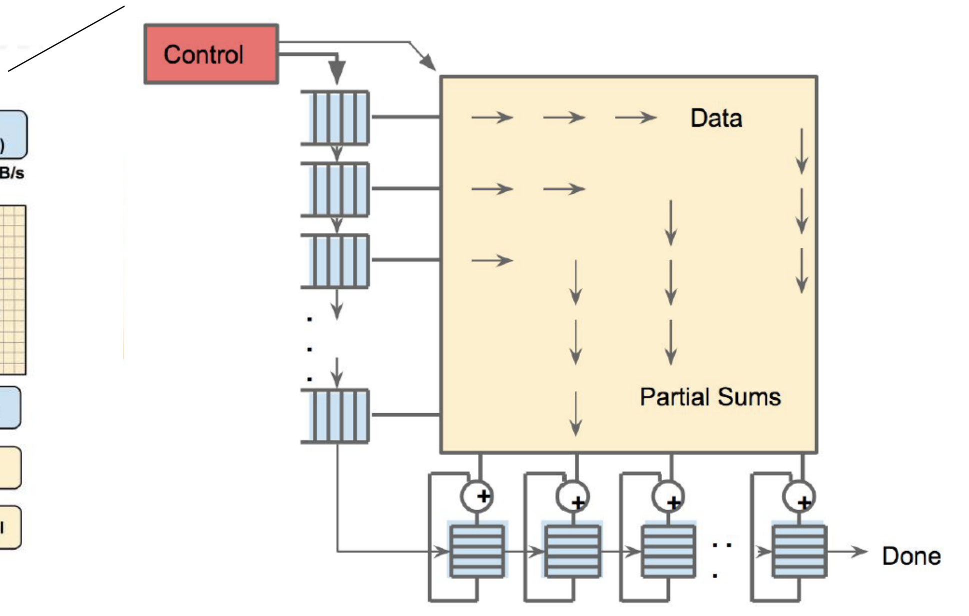
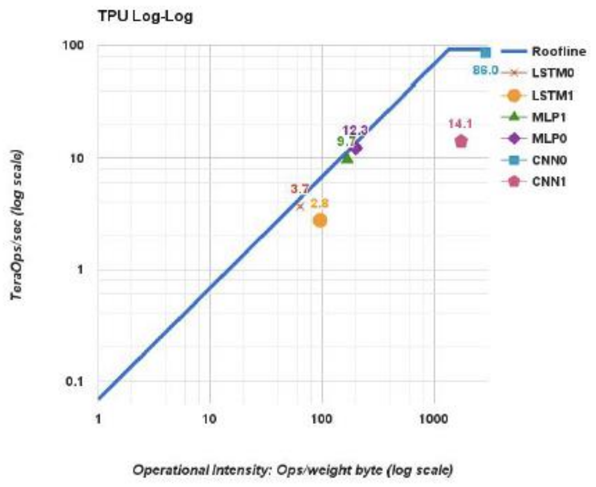
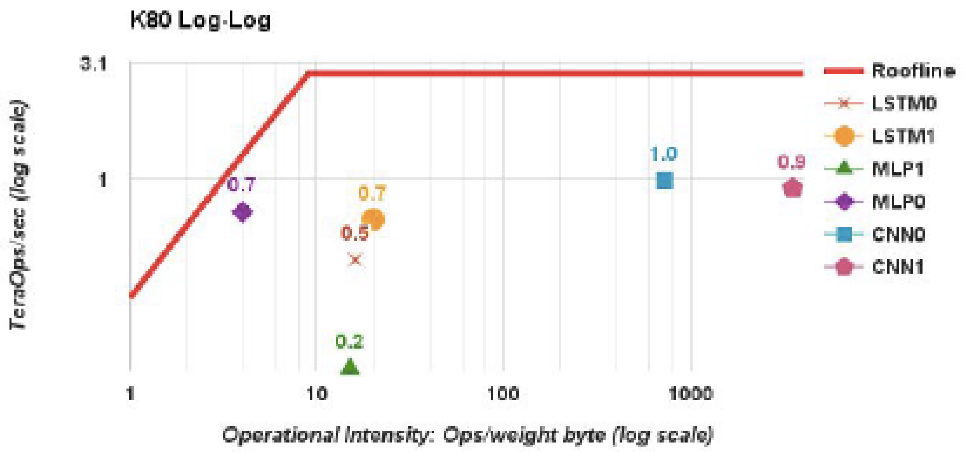


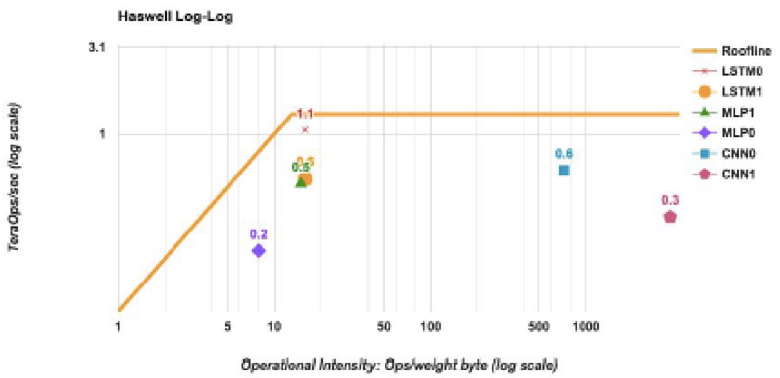
Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

TPU Performance

What are Roofline Plots?





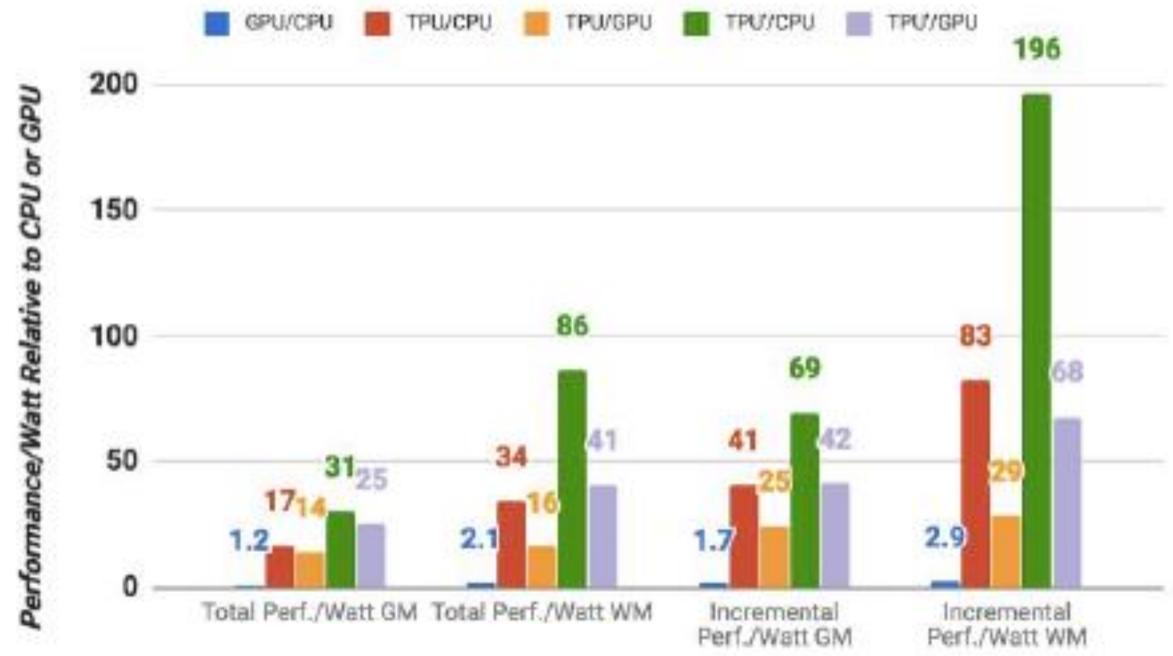


TPU Performance

Application	*MLP0*	*MLP1*	*LSTM0*	*LSTM1*	*CNN0*	*CNN1*	*Mean*	*Row*
Array active cycles	12.7%	10.6%	8.2%	10.5%	78.2%	46.2%	28%	1
Useful MACs in 64K matrix (% peak)	12.5%	9.4%	8.2%	6.3%	78.2%	22.5%	23%	2
Unused MACs	0.3%	1.2%	0.0%	4.2%	0.0%	23.7%	5%	3
Weight stall cycles	53.9%	44.2%	58.1%	62.1%	0.0%	28.1%	43%	4
Weight shift cycles	15.9%	13.4%	15.8%	17.1%	0.0%	7.0%	12%	5
Non-matrix cycles	17.5%	31.9%	17.9%	10.3%	21.8%	18.7%	20%	6
RAW stalls	3.3%	8.4%	14.6%	10.6%	3.5%	22.8%	11%	7
Input data stalls	6.1%	8.8%	5.1%	2.4%	3.4%	0.6%	4%	8
TeraOps/sec (92 Peak)	12.3	9.7	3.7	2.8	86.0	14.1	21.4	9

Table 3. Factors limiting TPU performance of the NN workload based on hardware performance counters. Rows 1, 4, 5, and 6 total 100% and are based on measurements of activity of the matrix unit. Rows 2 and 3 further break down the fraction of 64K weights in the matrix unit that hold useful weights on active cycles. Our counters cannot exactly explain the time when the matrix unit is idle in row 6; rows 7 and 8 show counters for two possible reasons, including RAW pipeline hazards and PCIe input stalls. Row 9 (TOPS) is based on measurements of production code while the other rows are based on performance-counter measurements, so they are not perfectly consistent. Host server overhead is excluded here. The MLPs and LSTMs are memory-bandwidth limited but CNNs are not. CNN1 results are explained in the text.

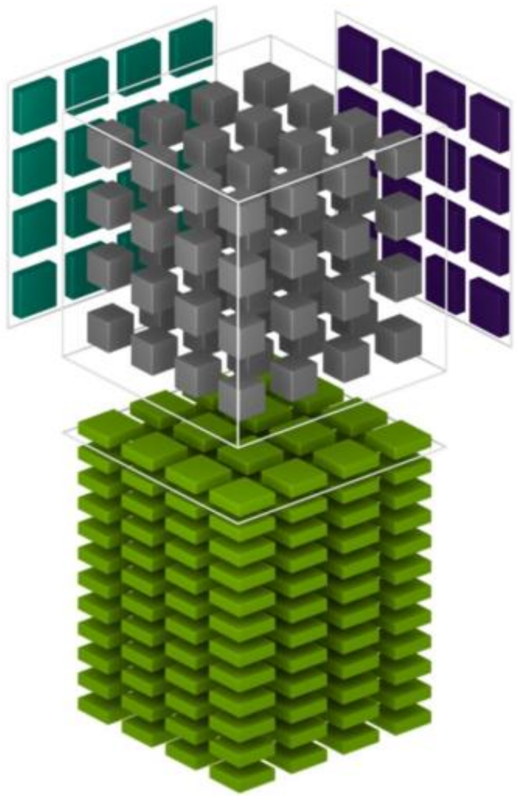
Performance/Watt

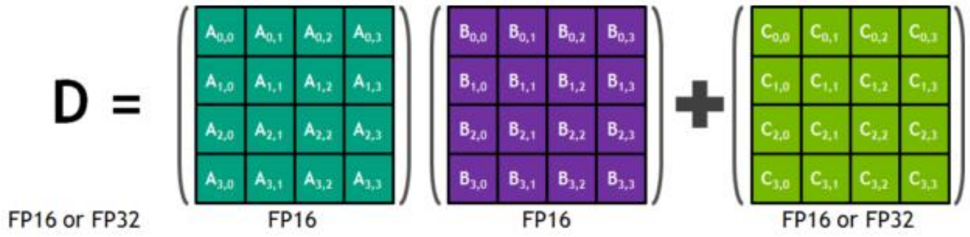


NVIDIA's response: TensorCores!

	K80 2012	TPU 2015	P40 2016
Inferences/Sec <10ms latency	$^{1}/_{13}$X	1X	2X
Training TOPS	6 FP32	NA	12 FP32
Inference TOPS	6 FP32	90 INT8	48 INT8
On-chip Memory	16 MB	24 MB	11 MB
Power	300W	75W	250W
Bandwidth	320 GB/S	34 GB/S	350 GB/S

NVIDIA Volta – TensorCores





Google response to NVIDIA's response!

- TPU v2
  - Some apps needed more precision – 16 bits instead of 8 bits
  - Added training
  - Slightly more general purpose
  - Can connect multiple TPUs together to form larger cluster
    - "pod" – supercomputer class machine
  - HBM memory – 30X improvement

- TPU v3 – liquid cooled
  - Enables even more scaling

- All in the name of running larger and larger ML workloads

TPU Next Steps

- Open research questions
  - Compress ML models to fit on a smaller devices
    - Current hot area of research
  - Find alternatives to backprop (important for training)
    - “Brains don’t do backprop”
  - Look at other kinds of ML beyond CNNs
    - Unsupervised learning
  - Remove barriers between CPUs and TPUs
    - CPU-GPU redux?