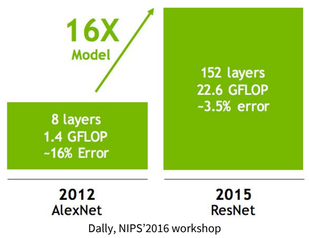
Design and Analysis of a Hardware CNN Accelerator

Stanford University

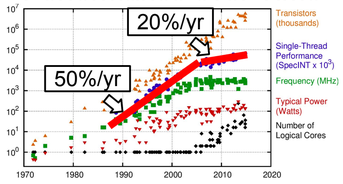
Kevin Kiningham, Michael Graczyk, Athul Ramkumar

Background

- New CNN models require massive computation

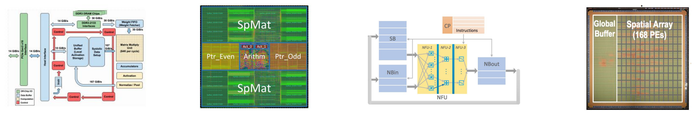


- Hardware Improvements have slowed



- One solution: specialized hardware

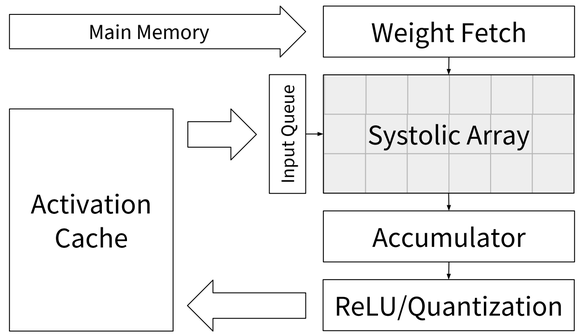
- Prior Work



Problem Statement

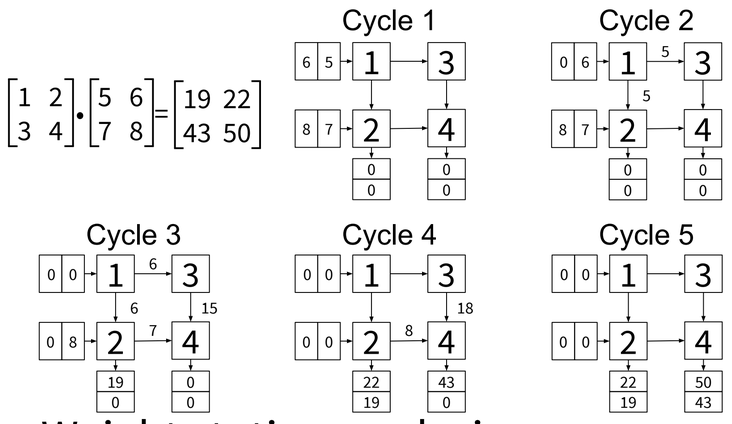
- Design and implement a hardware CNN accelerator
- Compare our design's power and throughput to TPU and a GPU

Design



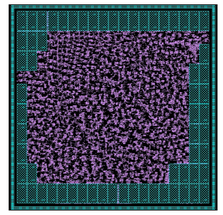
- Main component is systolic array matrix multiply
  - Conv, Batch Norm, FC, etc, can be transformed to dense matrix mult.
- Activations cached for fast reuse

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\bullet\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}=\begin{bmatrix}19 & 22\\43 & 50\end{bmatrix}$$



- Weight stationery design
  - Weights loaded from memory before multiply
  - Multiple inputs can reuse weights without reloading

Evaluation



Final layout
PnR in 32nm

- Evaluation on 8-bit quantized SqueezeNet (78.4 Top-5 ImageNet)

	Power (W peak)	Clock (MHz)	TOP/s		Area (mm^2)
			Int	FP	
K80/die	98	560	-	2.8	561
TPU/die	40	700	92	-	<300
Ours	**20.9**	**714**	**93.6**	-	**76.6**

- Power breakdown

Memory	3.2W	15.7%
Clock	1.1W	5.4%
Register	7.3W	35%
Combinational	9.1W	43%

Conclusion

- Our design efficiently accelerates CNN inference
  - 33x TOPS @ 0.21x power vs GPU
- Future: accelerate sparse matrix ops