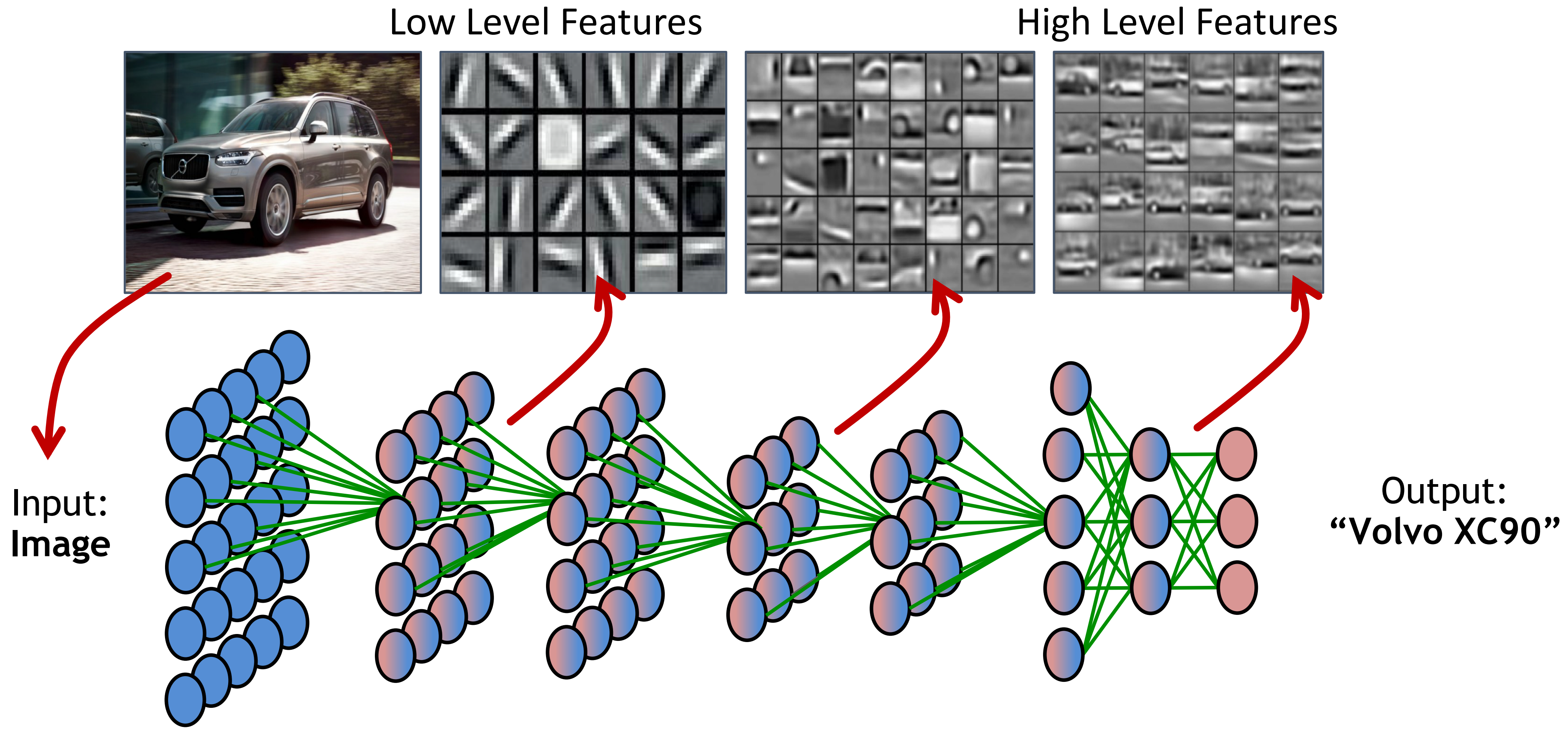
Design Considerations for Efficient Deep Neural Networks on Processing-in-Memory Accelerators

Tien-Ju Yang, Vivienne Sze

Massachusetts Institute of Technology

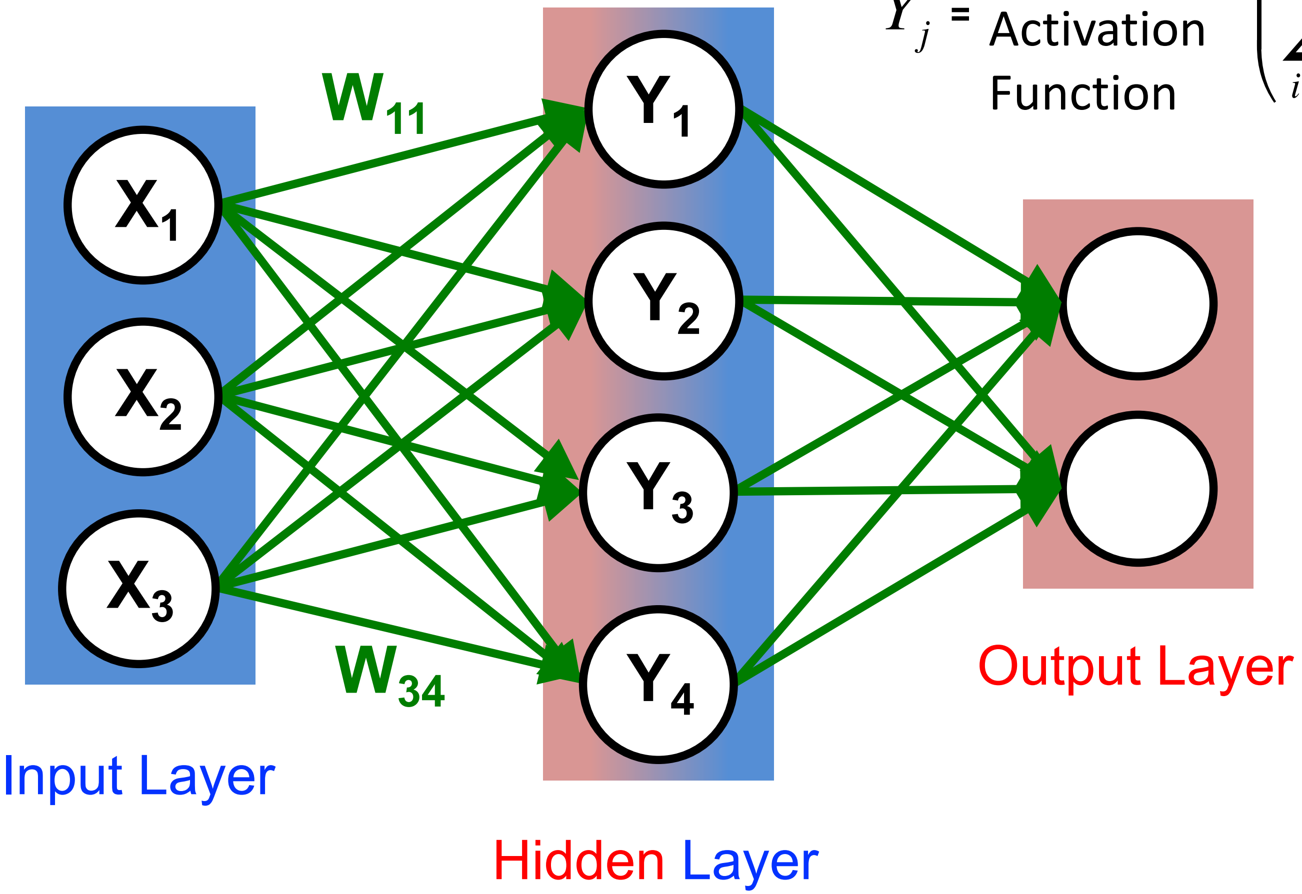
email: sze@mit.edu
twitter: @eems_mit
website: http://sze.mit.edu

What are Deep Neural Networks (DNNs)?



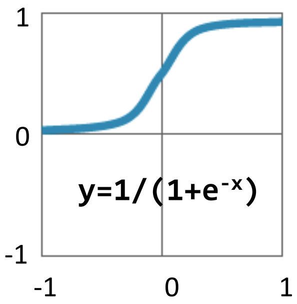
Modified Image Source: [**Lee**, *CACM* 2011]

Weighted Sums



$$Y_j = \text{Nonlinear Activation Function}\left(\sum_{i=1}^{3} W_{ij} \times X_i\right)$$

Sigmoid



Rectified Linear Unit (ReLU)

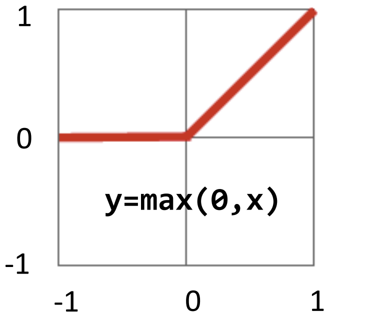
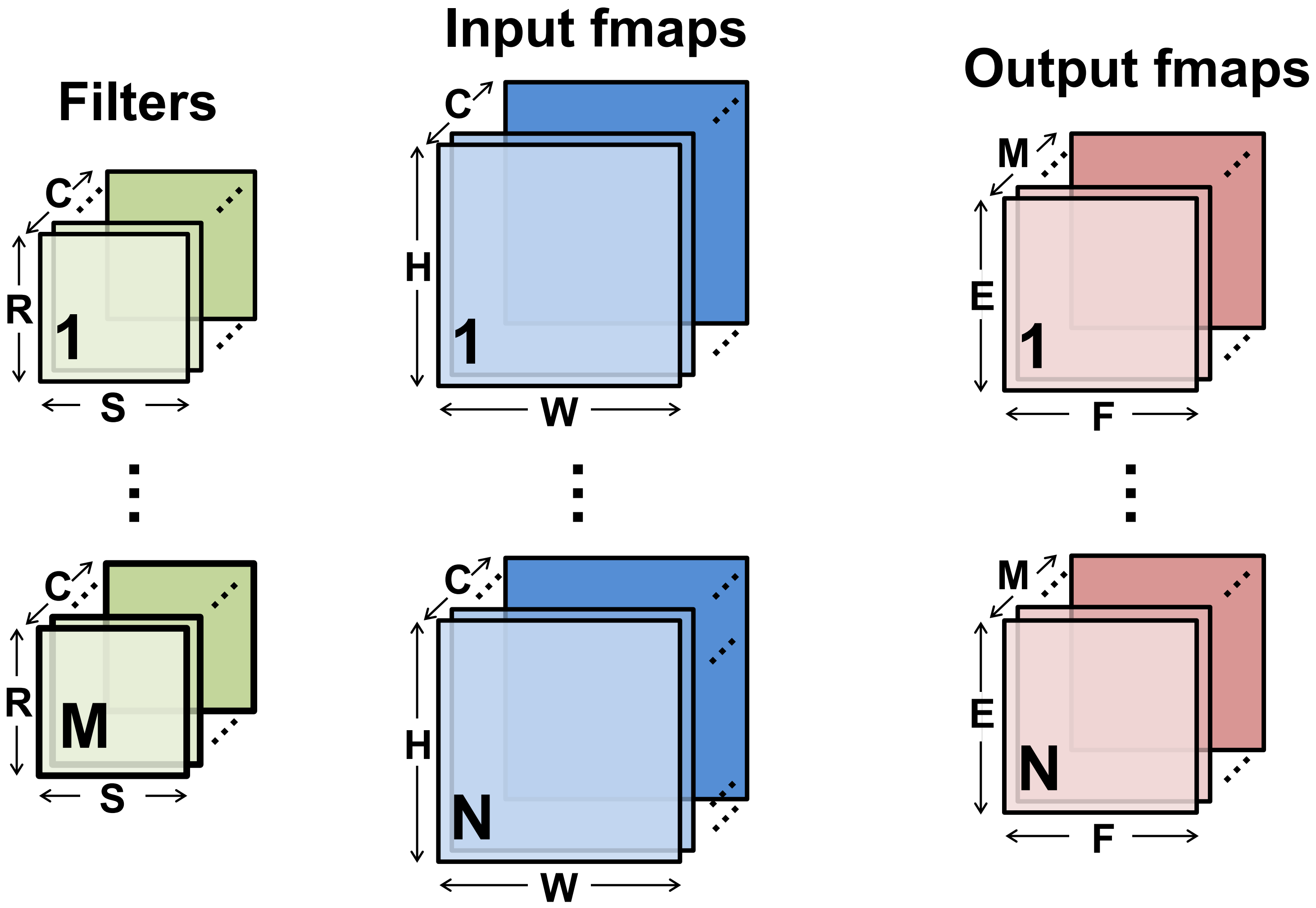


Image source: Caffe tutorial

Key operation is
multiply and accumulate (MAC)
Accounts for > 90% of computation

Define Shape for Each Layer

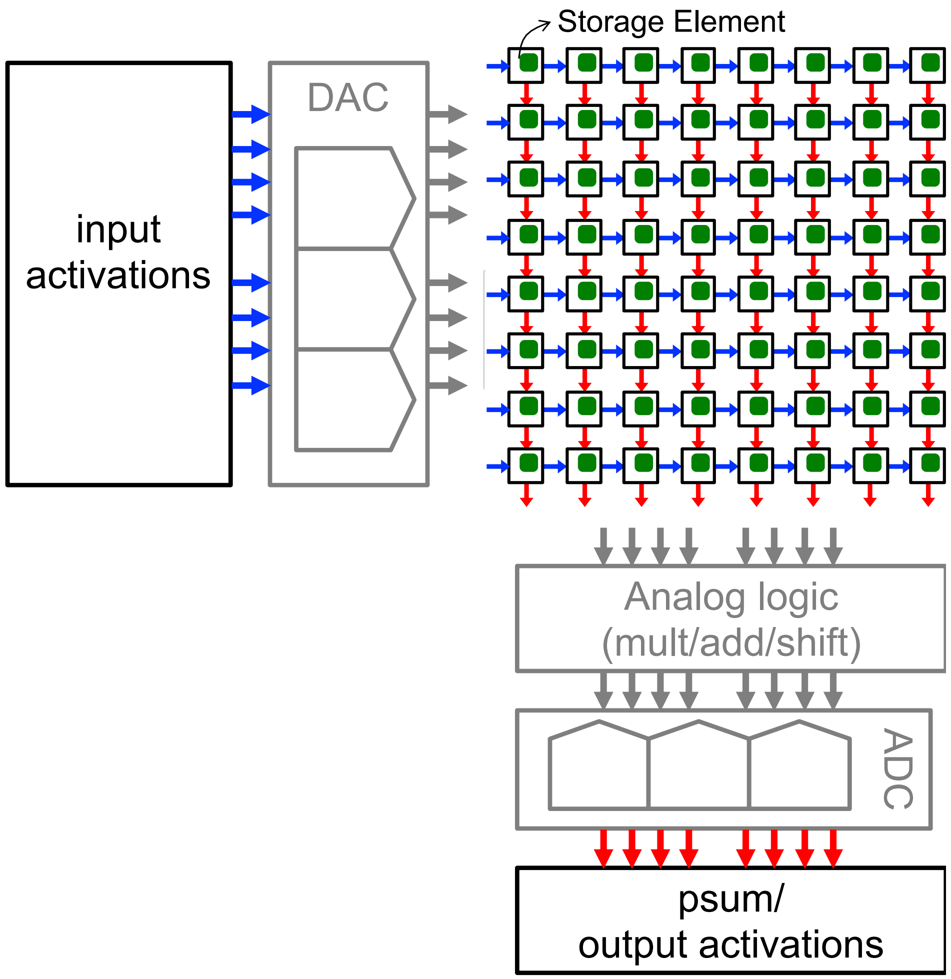


Shape **varies** across layers

H – Height of input fmap (activations)
W – Width of input fmap (activations)
C – Number of 2-D input fmaps /filters (channels)
R – Height of 2-D filter (weights)
S – Width of 2-D filter (weights)
M – Number of 2-D output fmaps (channels)
E – Height of output fmap (activations)
F – Width of output fmap (activations)
N – Number of input fmaps/output fmaps (batch size)

Processing-in-Memory (PIM) Accelerators

- Emerging approach for processing DNNs
- Implement as **matrix-vector multiply**
- **Reduce weight data movement** by moving compute into the memory
- **Increase weight bandwidth and amount of parallel MACs**



Design Considerations for PIM Accelerators

- **Prediction Accuracy**
  - **non-idealities of analog compute**
    - Solution: per chip training → expensive in practice
  - **lower bit widths for data and computation**
    - Solution: multiple devices per weight → decrease area density
    - Solution: bit serial processing → increase cycles per MAC
- **Hardware Efficiency**
  - **Data movement into/from array**
    - A/D and D/A conversion increase energy consumption and reduce area density
  - **Array utilization**
    - Large array size can amortize conversion cost → increase area density and data reuse → DNNs need to take advantage of this property

Our Contributions

- **The design of the DNN network architecture** (i.e., layer shape, and # of layers) for PIM is less studied than training DNN weights for PIM
- We evaluate the accuracy and efficiency of **state-of-the-art DNNs** on PIM accelerators with the **large-scale ImageNet Dataset**
- We show that approaches for designing accurate and efficient DNNs for traditional digital accelerators may not apply for PIM

Key takeaway: Need to rethink the design of the DNN network architecture for PIM for improved accuracy and efficiency

Prediction Accuracy

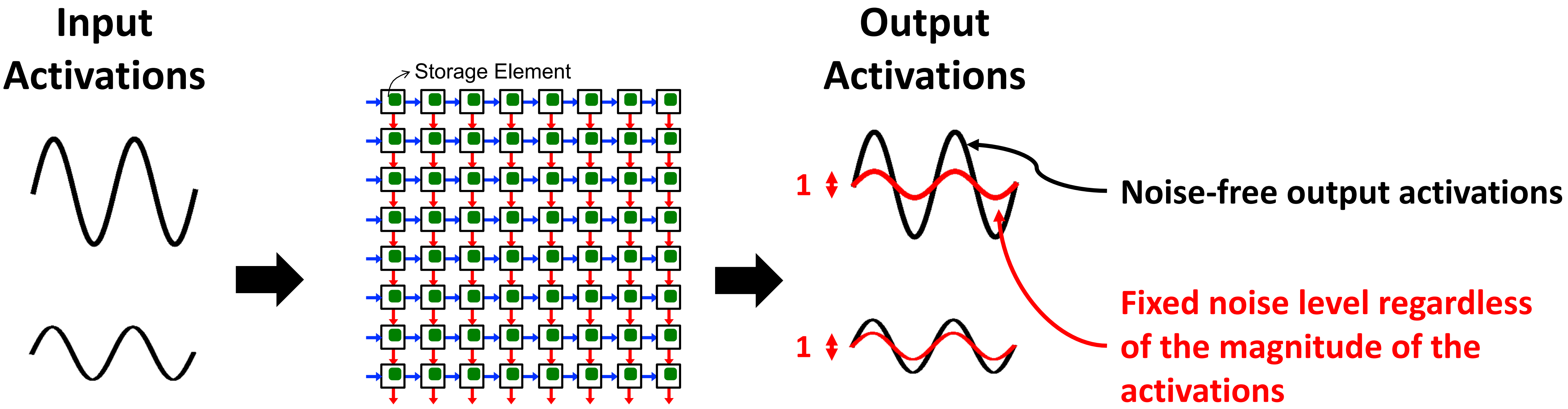
- Noise resilience
- Low precision computation

Noise Resilience

- Non-idealities in PIM cause the weights and activations to deviate from their intended values
- Accuracy under these non-idealities should be considered
- Evaluate noise resilience of various DNNs
  – Inject zero-mean Gaussian noise into the output activations to account for the noise in the input activations, weights, and computation
  – The weights are not retrained

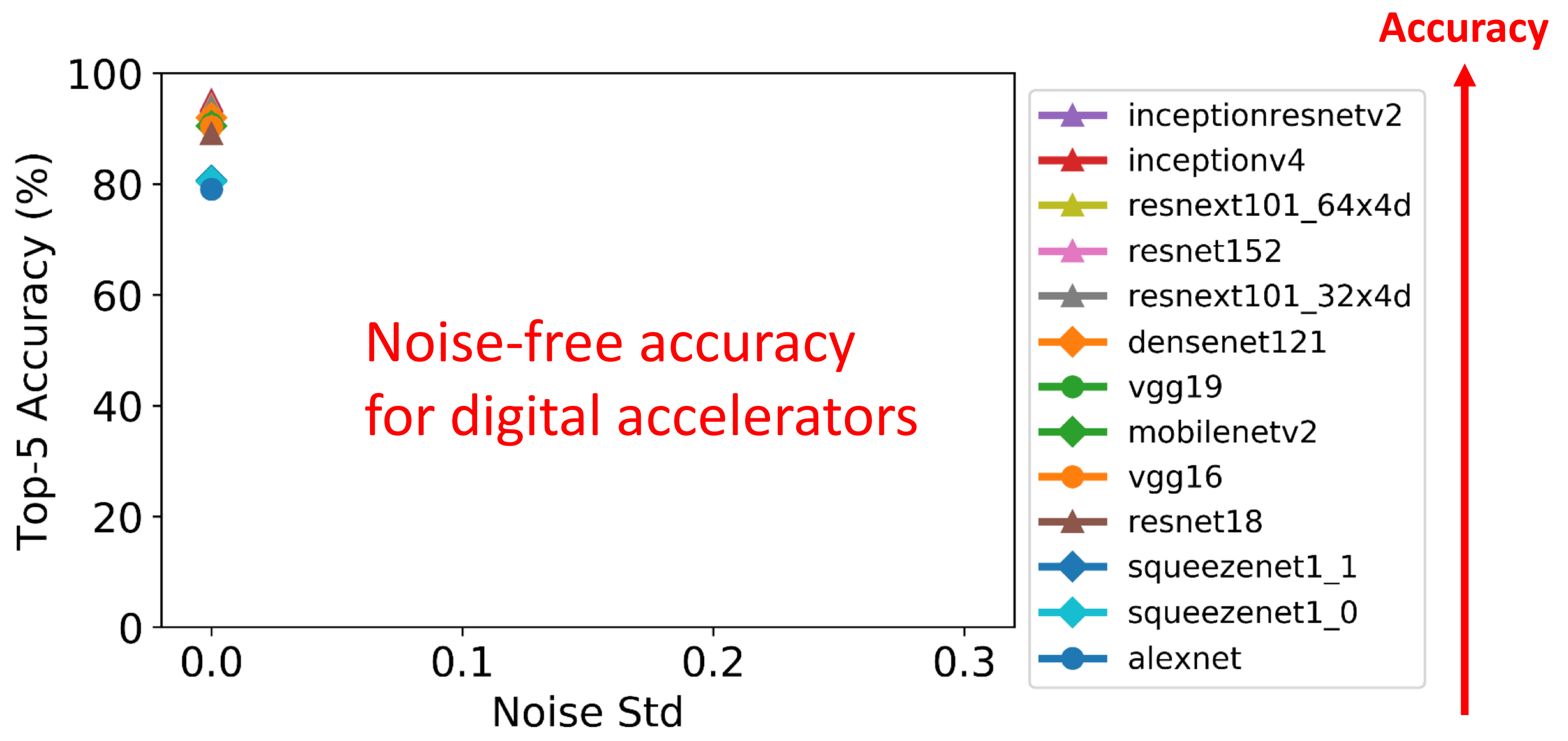
Noise Resilience

Fixed noise: Noise has fixed standard deviation and does not change with magnitude of the activations



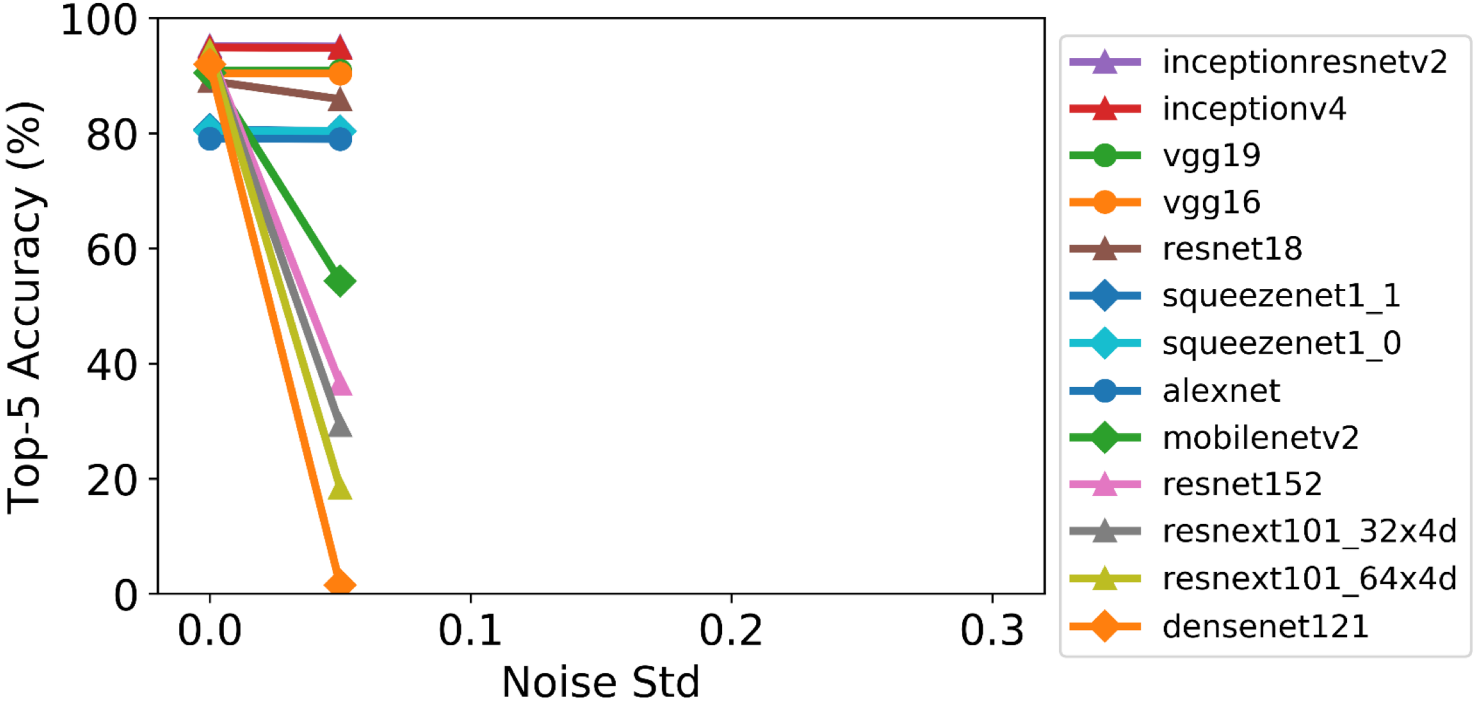
Fixed Noise Resilience

- Different DNNs have different sensitivities to noise



Fixed Noise Resilience

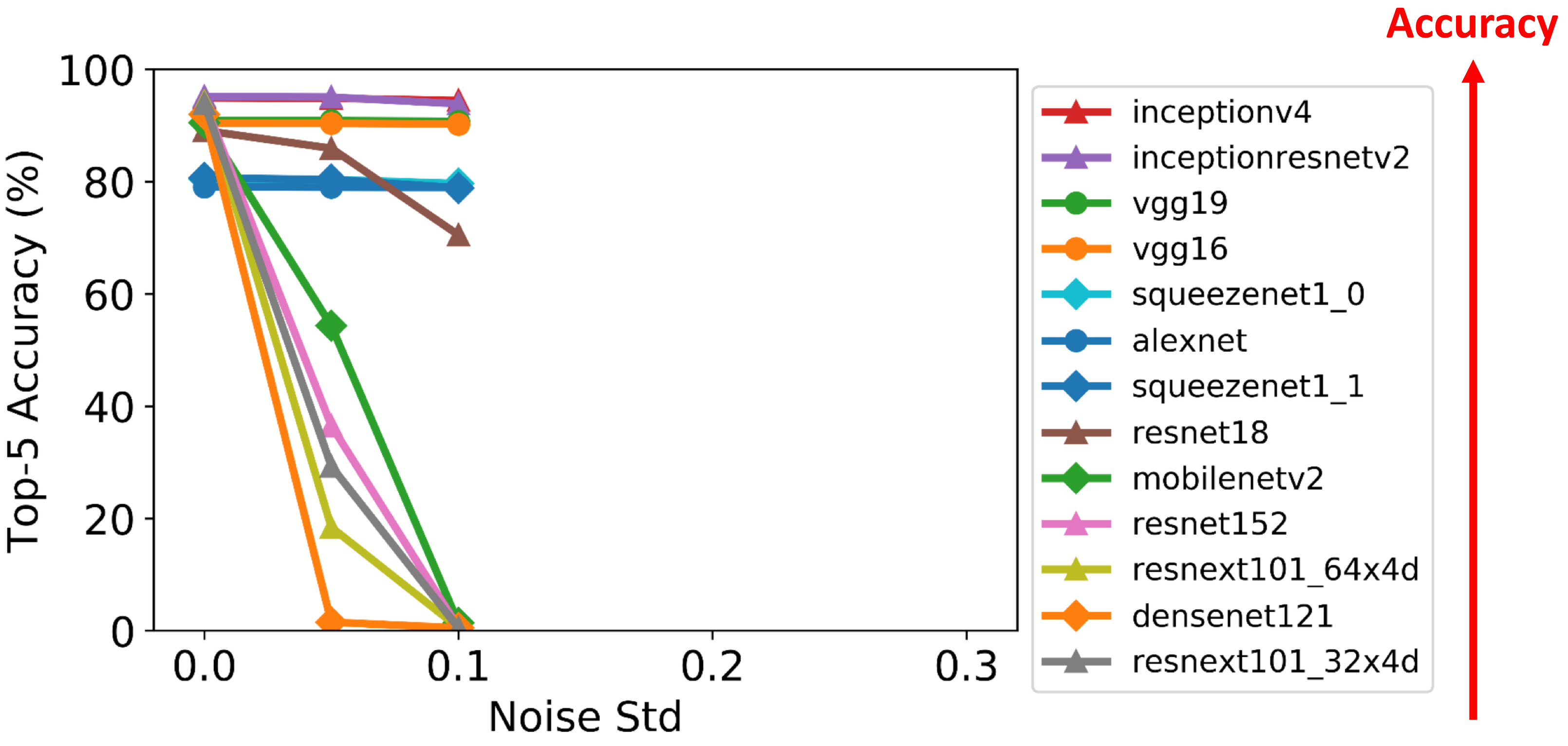
- Different DNNs have different sensitivities to noise



Accuracy

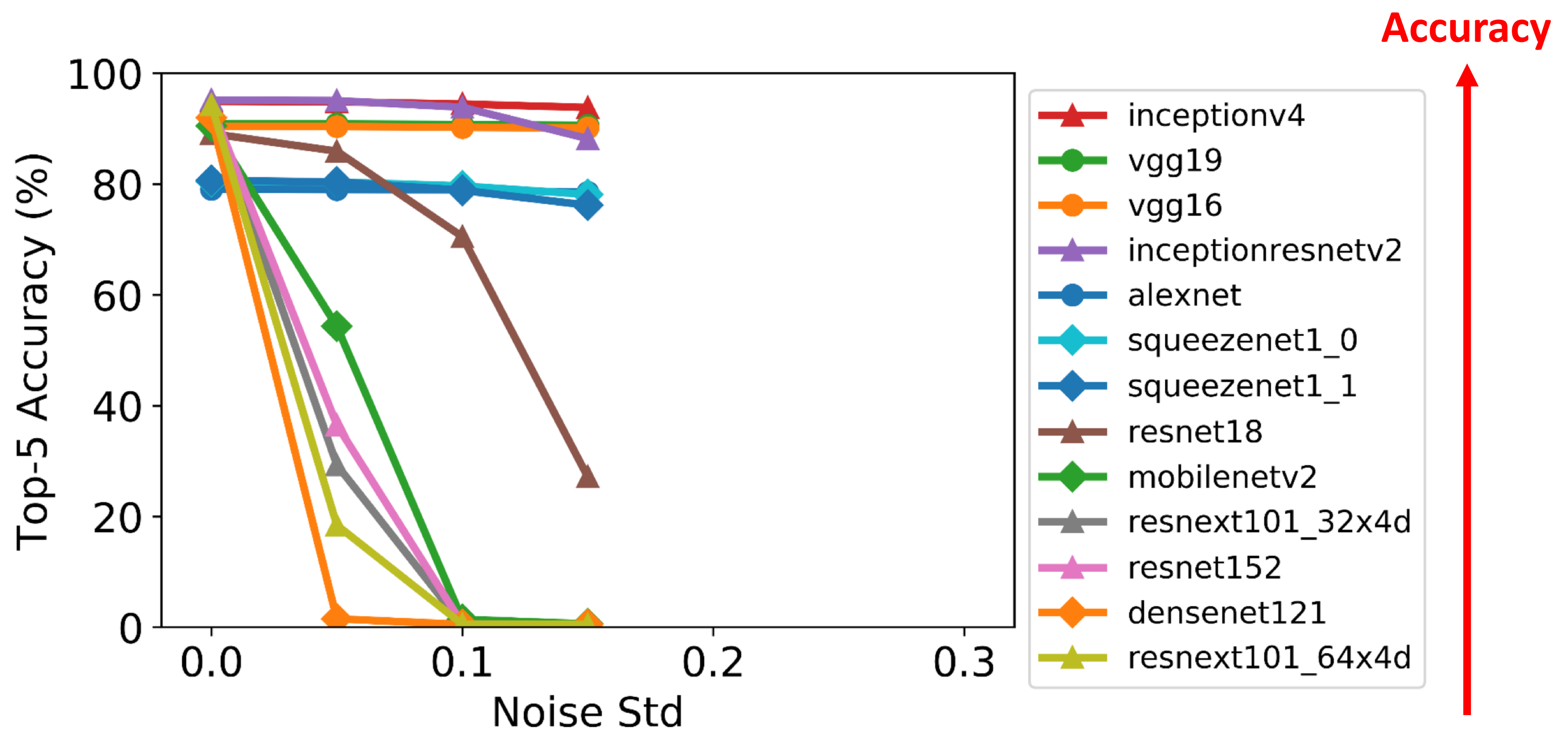
Fixed Noise Resilience

- Different DNNs have different sensitivities to noise



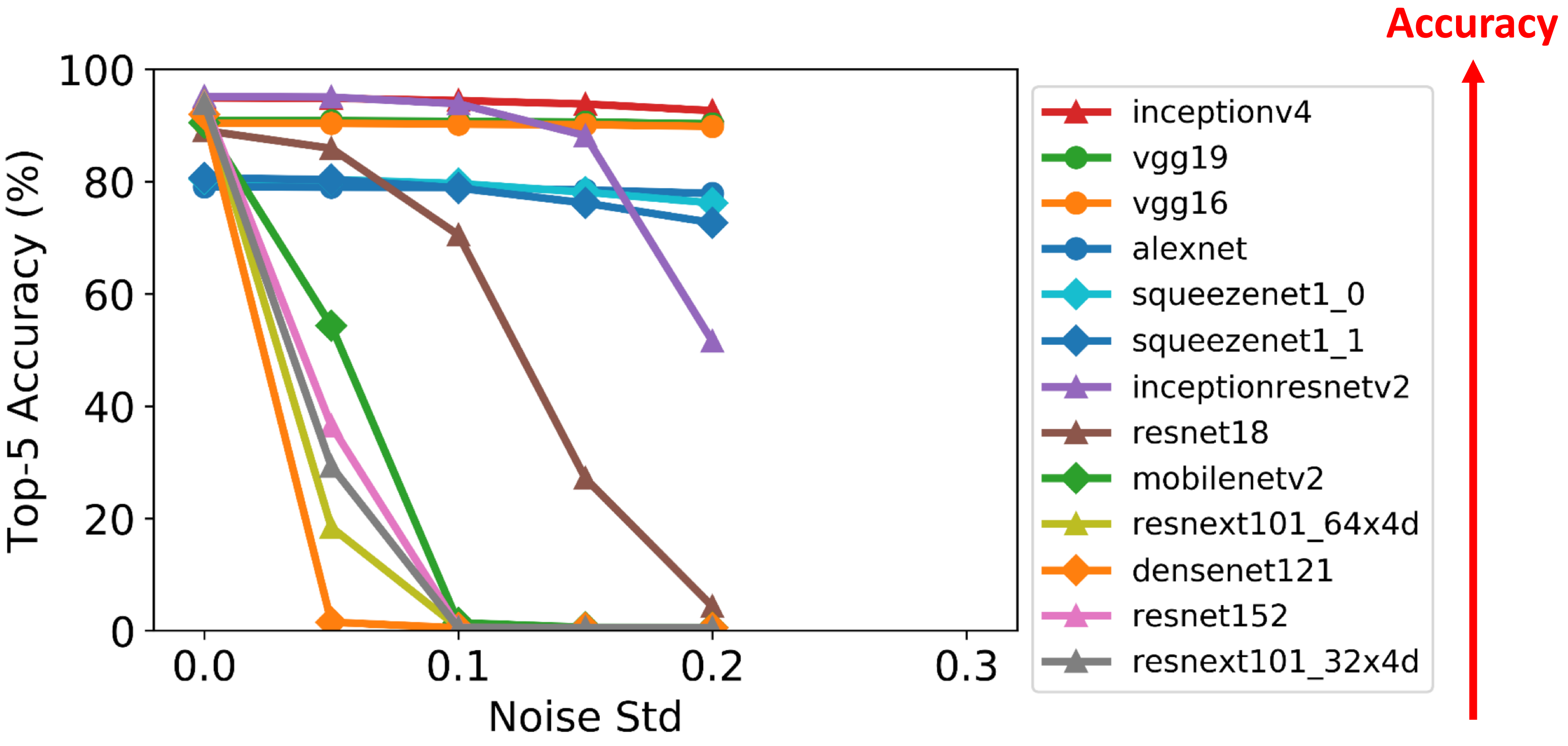
Fixed Noise Resilience

- Different DNNs have different sensitivities to noise



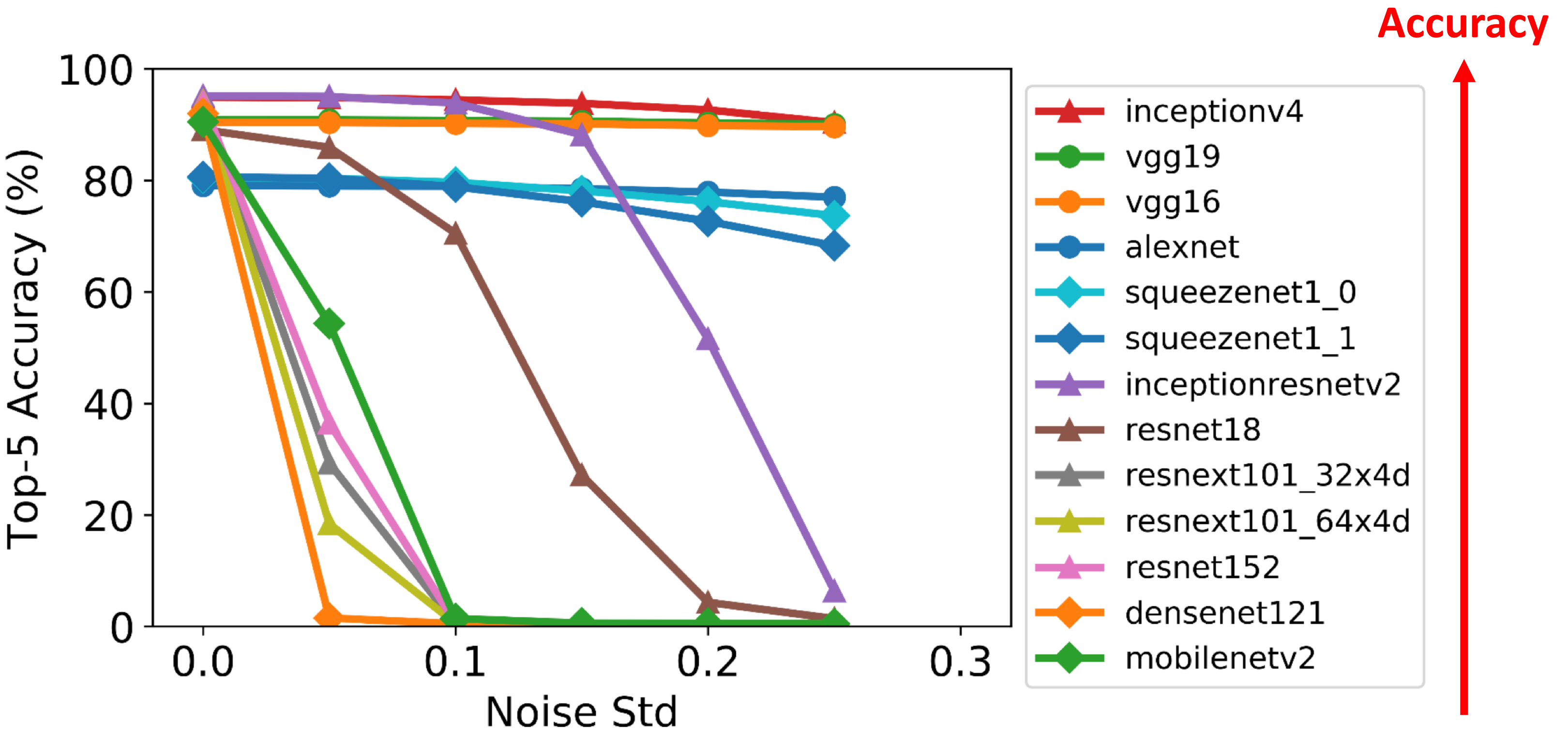
Fixed Noise Resilience

- Different DNNs have different sensitivities to noise



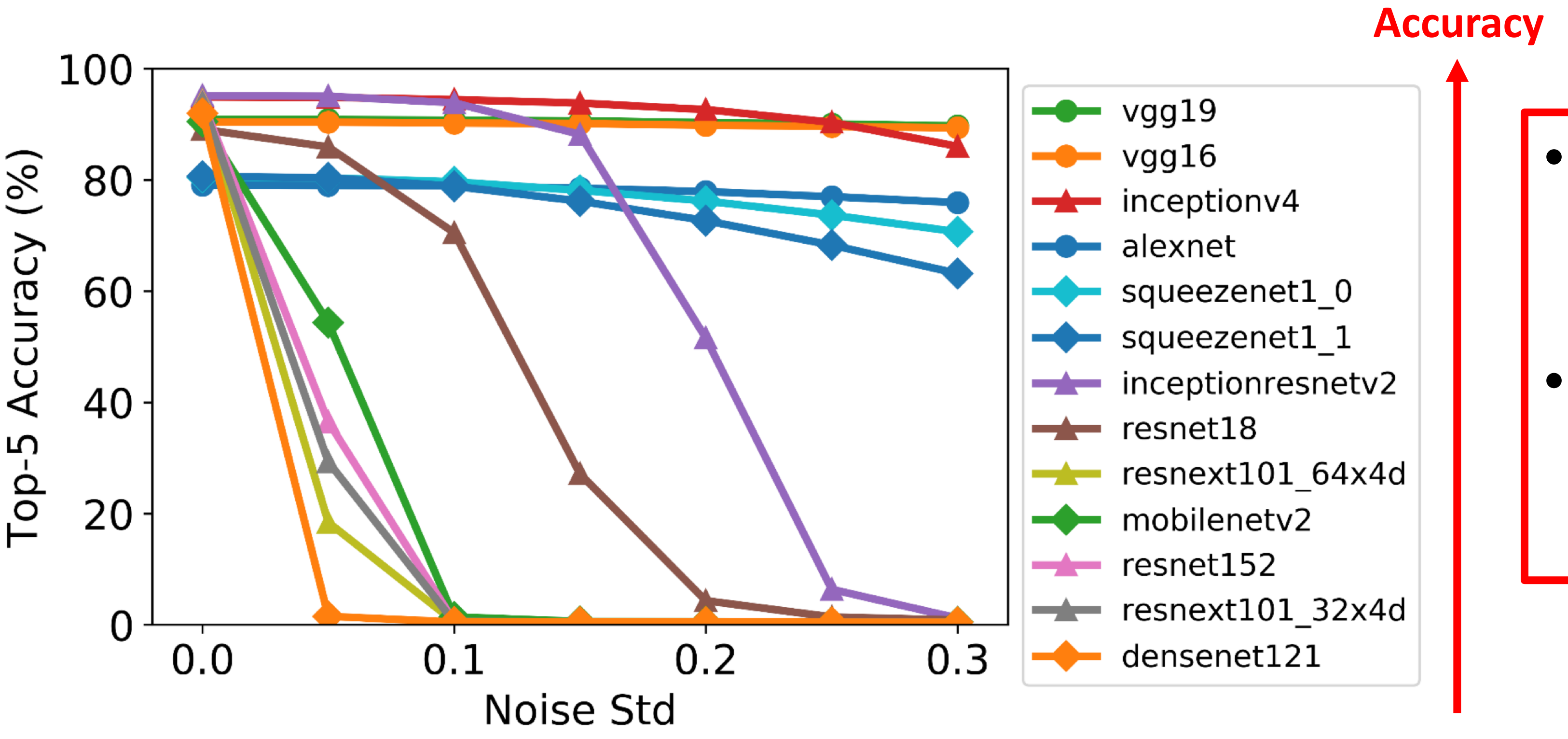
Fixed Noise Resilience

- Different DNNs have different sensitivities to noise



Fixed Noise Resilience

- Different DNNs have different sensitivities to noise

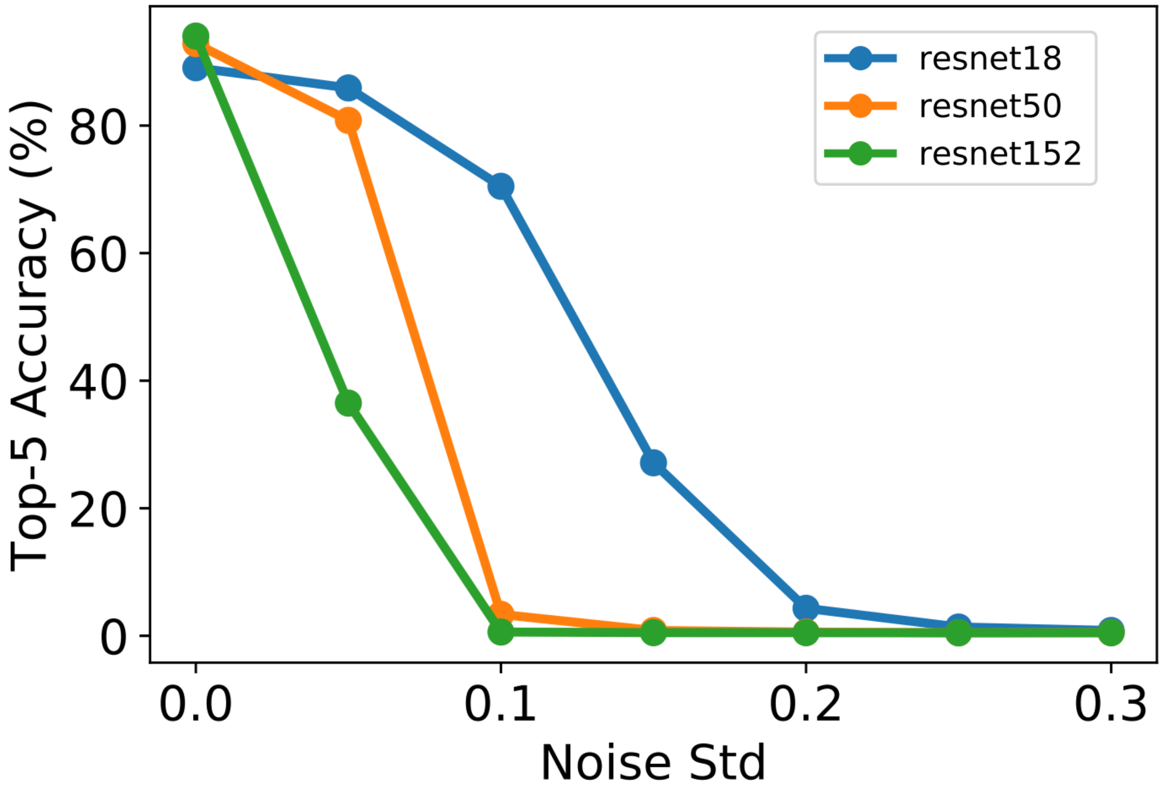


- Rank of accuracy changes with amount of noise
- The most accurate DNN for digital accelerators ***may not*** be the most accurate for PIM

Fixed Noise Resilience – Network Depth

Recent trend for designing DNNs that run on digital accelerators:

– **Increase number of layers (network depth)** + reduce filter size



As the depth increases,

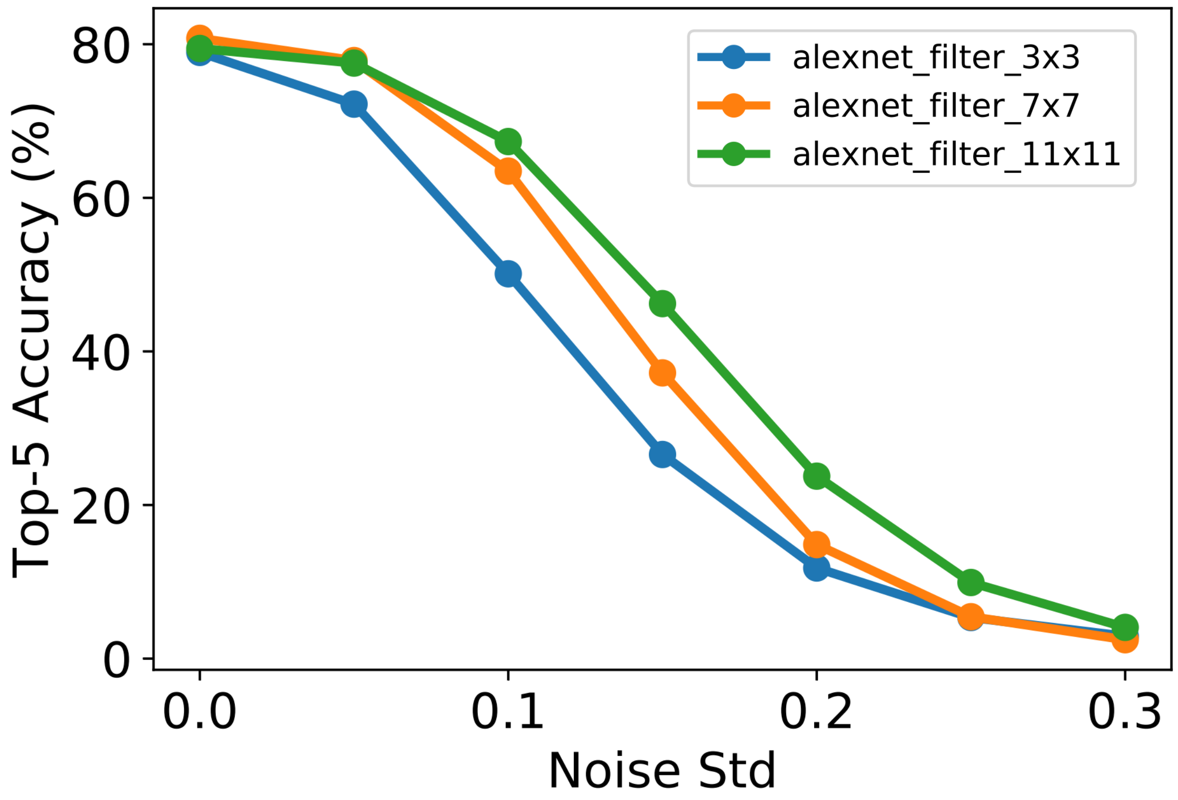
- Ideal (noise-free) accuracy increases
- However, the accuracy **decreases faster** with increasing noise

Hypothesis: Shallower DNNs have less accumulated errors across layers

Fixed Noise Resilience – Filter Size

Recent trend for designing DNNs that run on digital accelerators:

– Increase number of layers + **reduce filter size**



As the filter size increases,

- Accuracy **decreases slower** with increasing noise

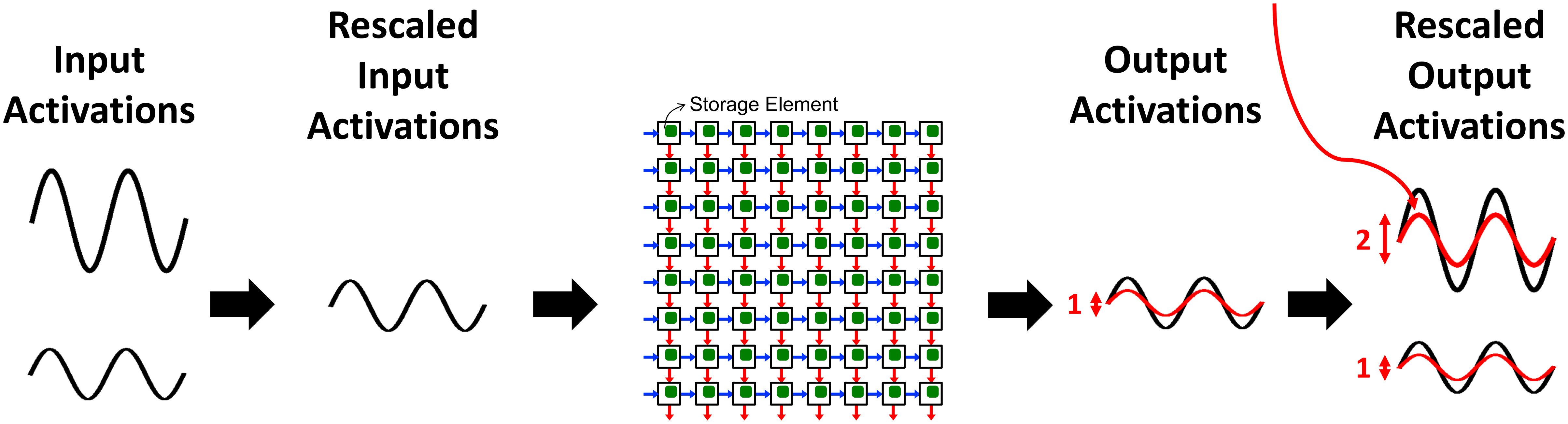
Hypothesis: Larger filters have more redundancy and are more robust to noise

Noise Resilience

Rescaled noise: Standard deviation of noise scales with respect to the maximum magnitude of the activations

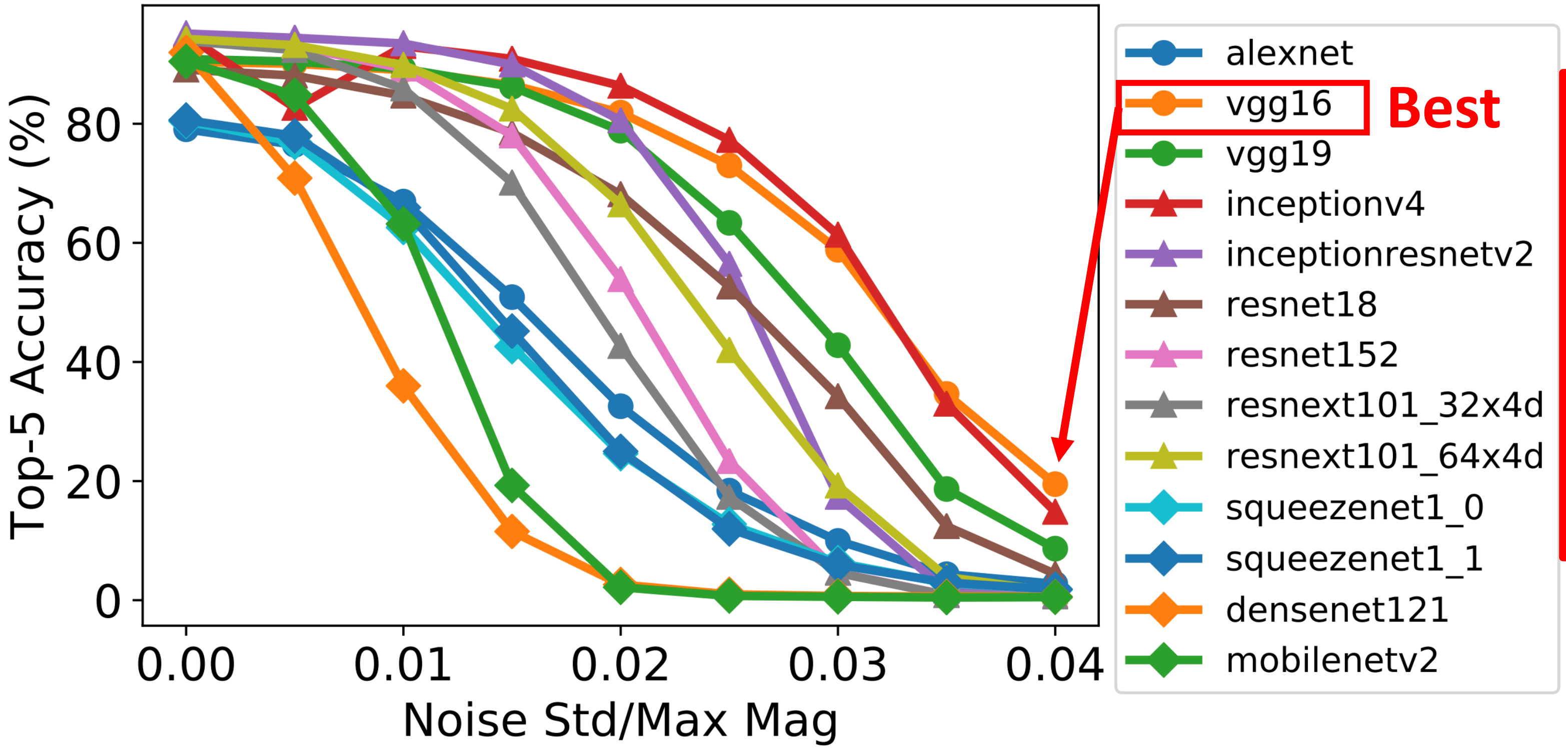
[**Gokmen**, *Frontiers in Neuroscience* 2016]

The noise level varies with respect to the maximum magnitude of the activations



Rescaled Noise Resilience

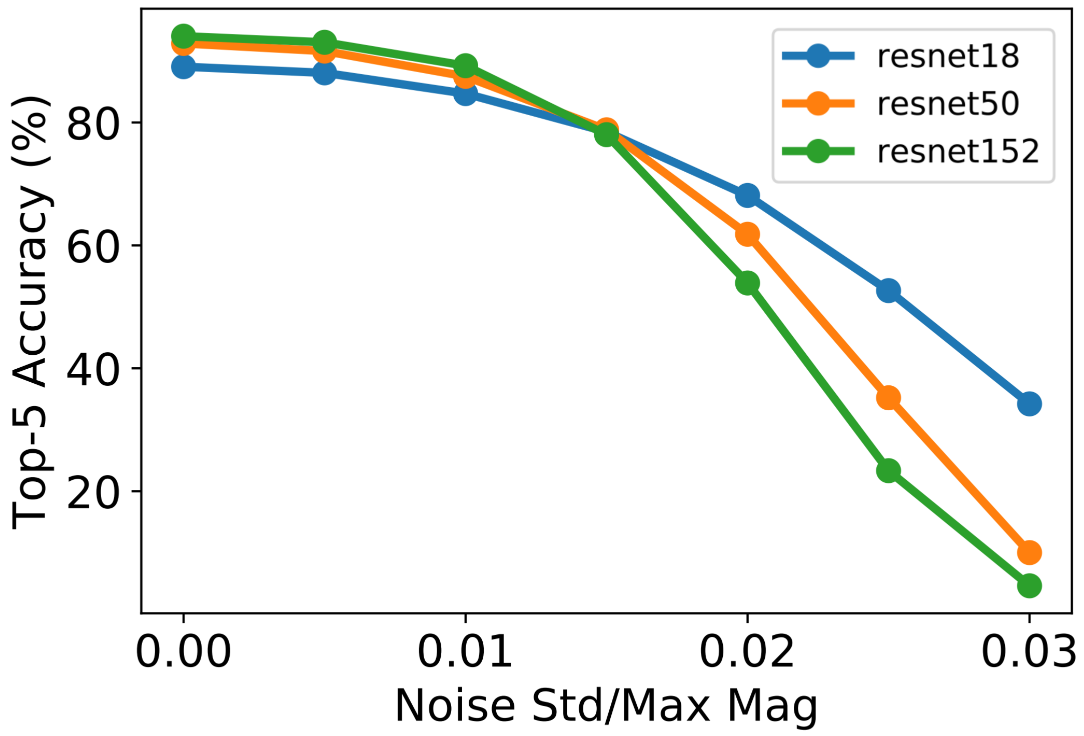
- Different DNNs have different sensitivities to noise



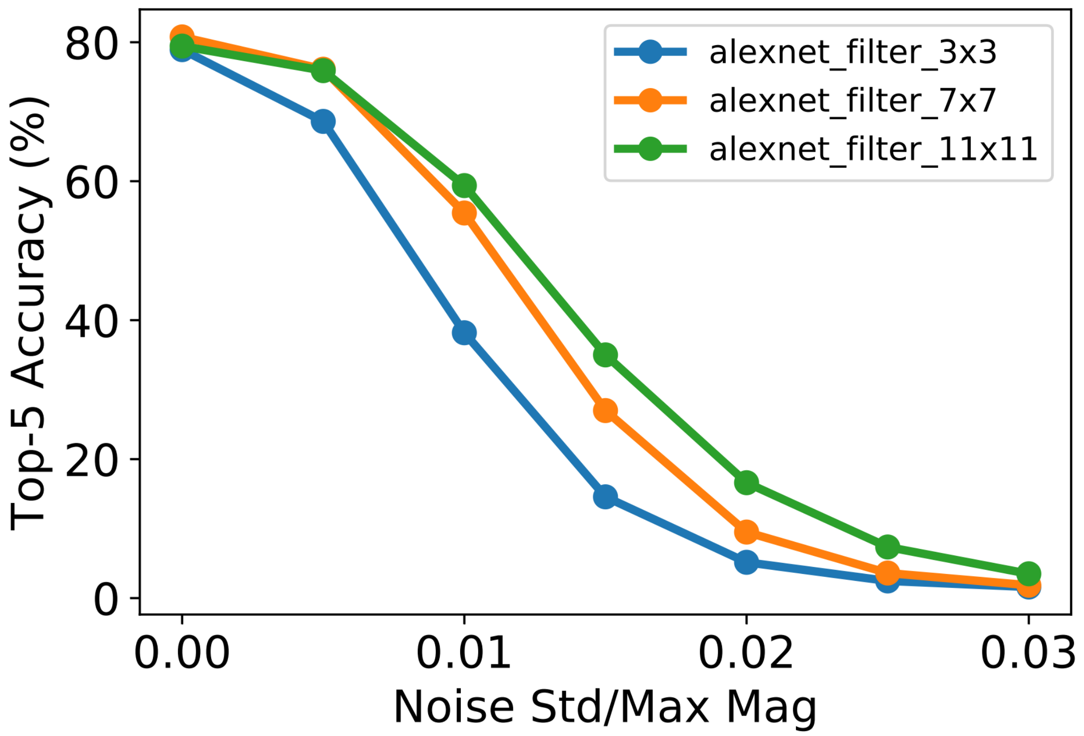
- Rank of accuracy changes with amount of noise
- The most accurate DNN for digital accelerators ***may not*** be the most accurate for PIM

Same trend as fixed noise

Rescaled Noise Resilience – Network Depth/Filter Size



▲ Network Depth



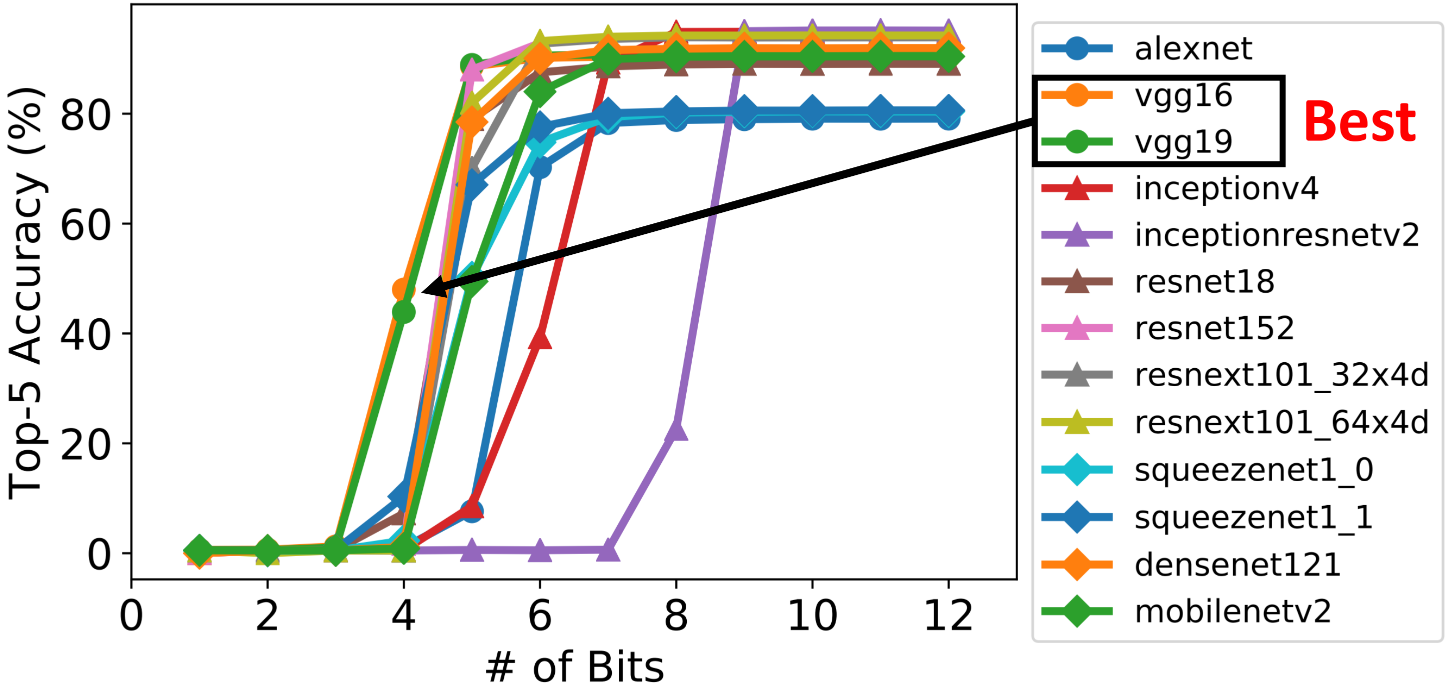
▲ Filter Size

Reducing depth or increasing filter size may make DNN more robust

Same trend as fixed noise

Low Precision Computation

- Different DNNs have different sensitivities to the bit width of weights



- Rank of accuracy changes with different bit widths of weights
- Shallower DNNs with larger filters (e.g., VGG) achieve the highest accuracy at 4 bits

Prediction Accuracy – Short Summary

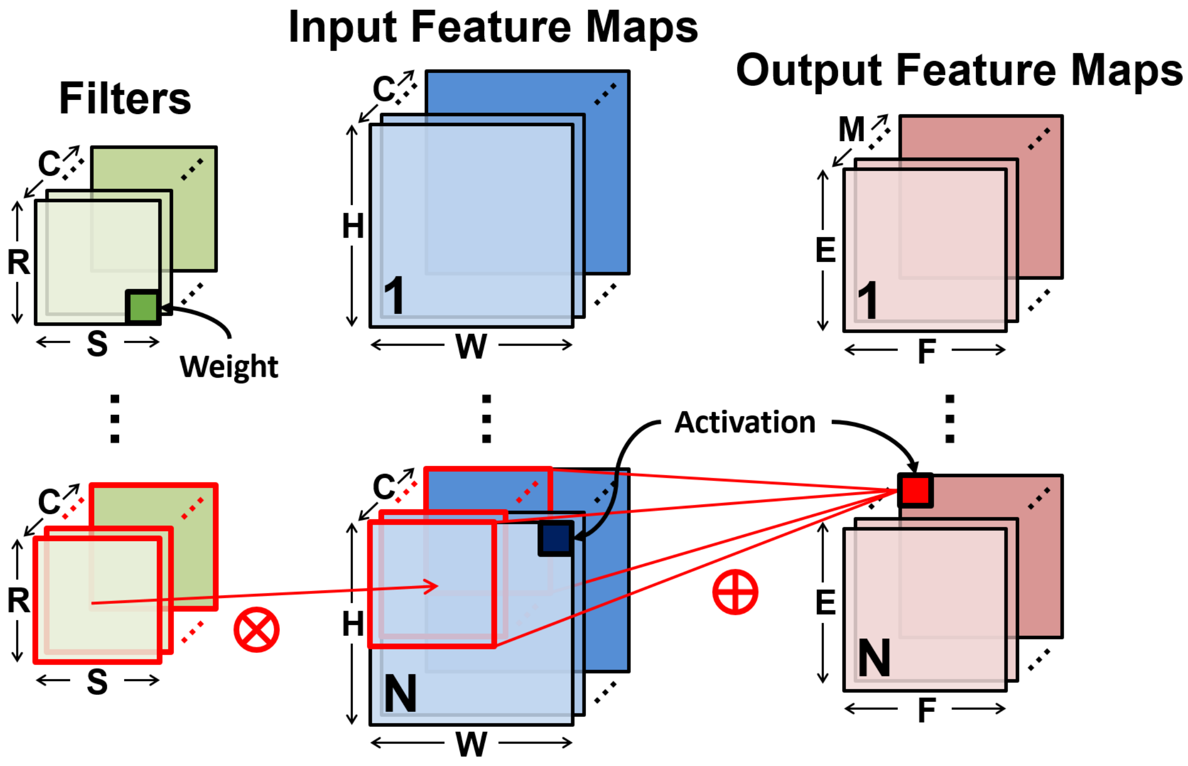
- DNNs that achieve high accuracy on digital accelerators **may not** have high accuracy on PIM due to noise and lower bit width
- Need to rethink the DNN network architecture design approach for PIM accelerators to maximize the accuracy
- Retraining the weights to further increase the robustness for PIM accelerators is still an open area of research

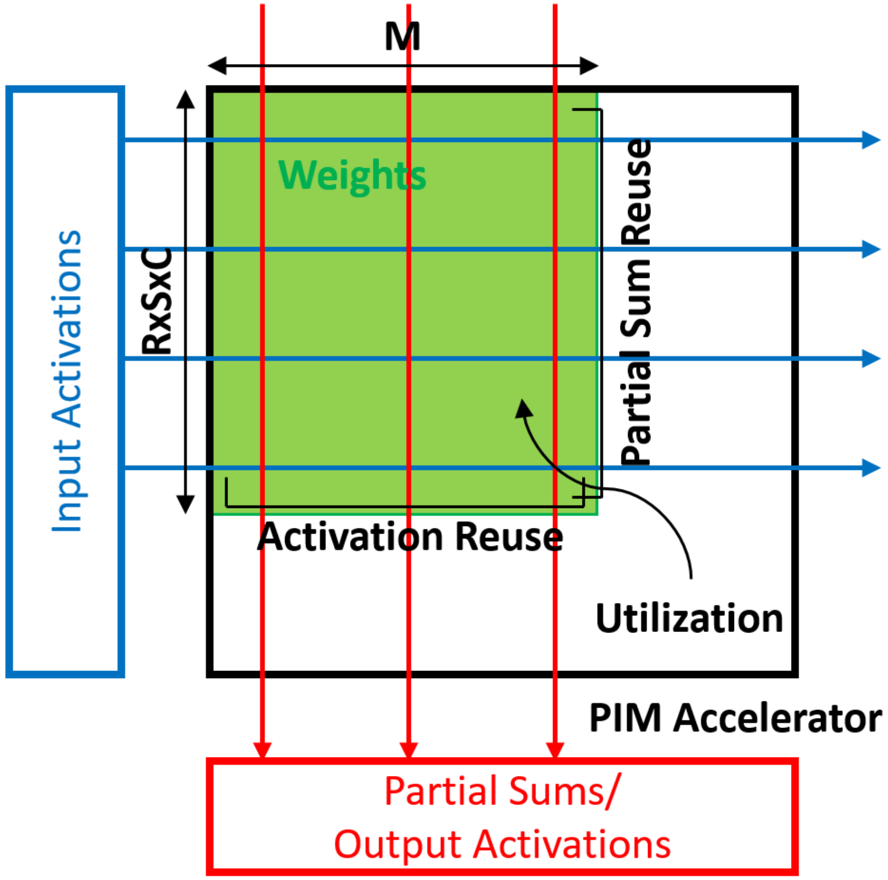
Hardware Efficiency

- Data movement of activations
- Impact of array size on utilization

Data Reuse

- **Reuse:** number of times a value (e.g., weight, activation) is used when it moves into the array
- PIM accelerators maximize the reuse of weights





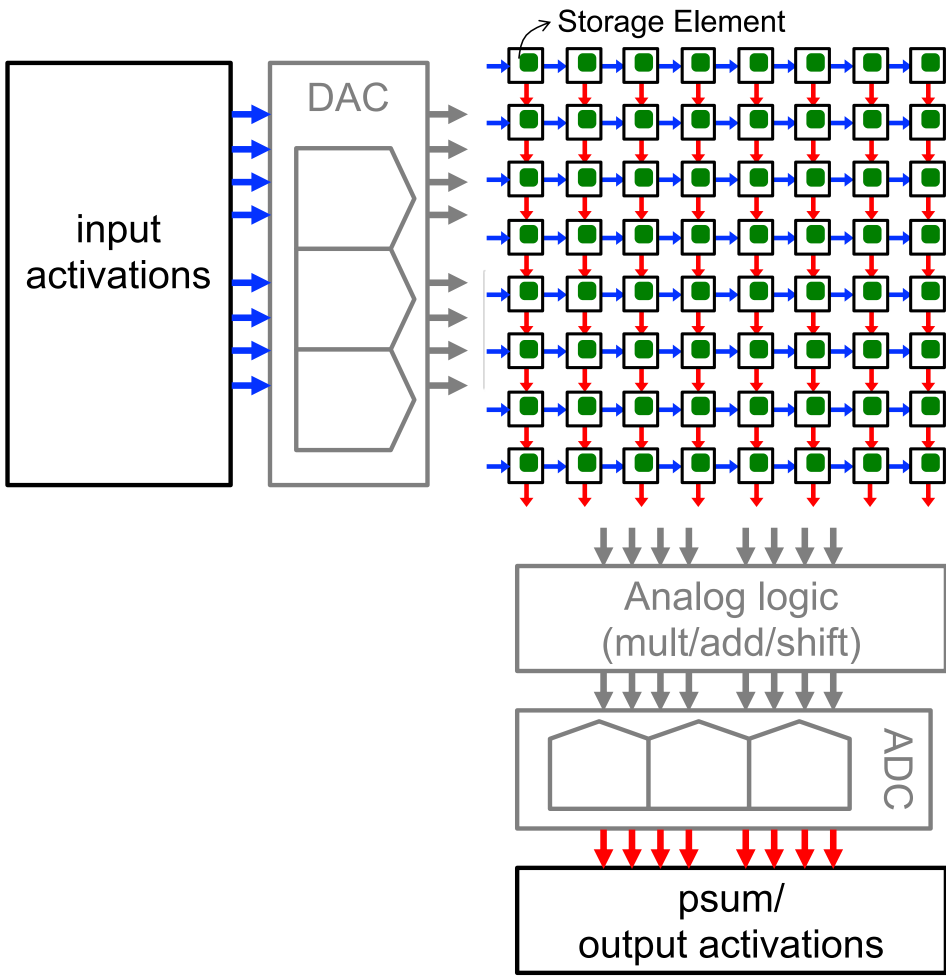
Weight-stationary dataflow of PIM accelerators [**Chen**, *ISCA* 2016]

Data Movement of Activations

- Weight-stationary dataflow trades the movement of **weights** for the movement of **activations**

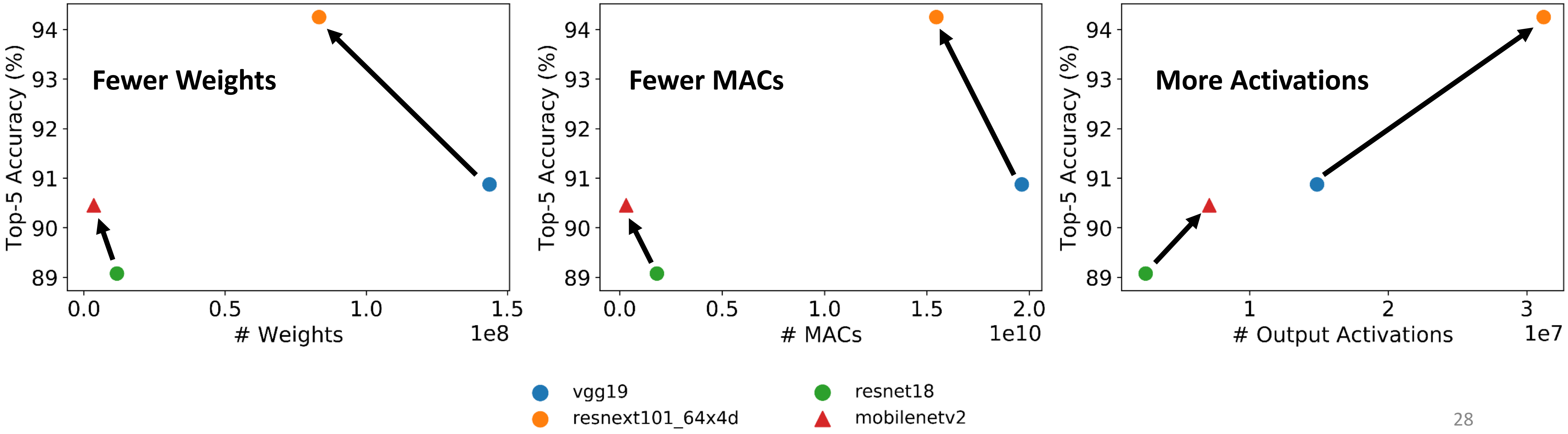
- Movement of activations can dominate energy consumption of PIM accelerators due to the **costly peripheral circuits**

- Two key factors for energy consumption:
  - Number of activations
  - Data reuse: array utilization (discussed next)



Data Movement of Activations

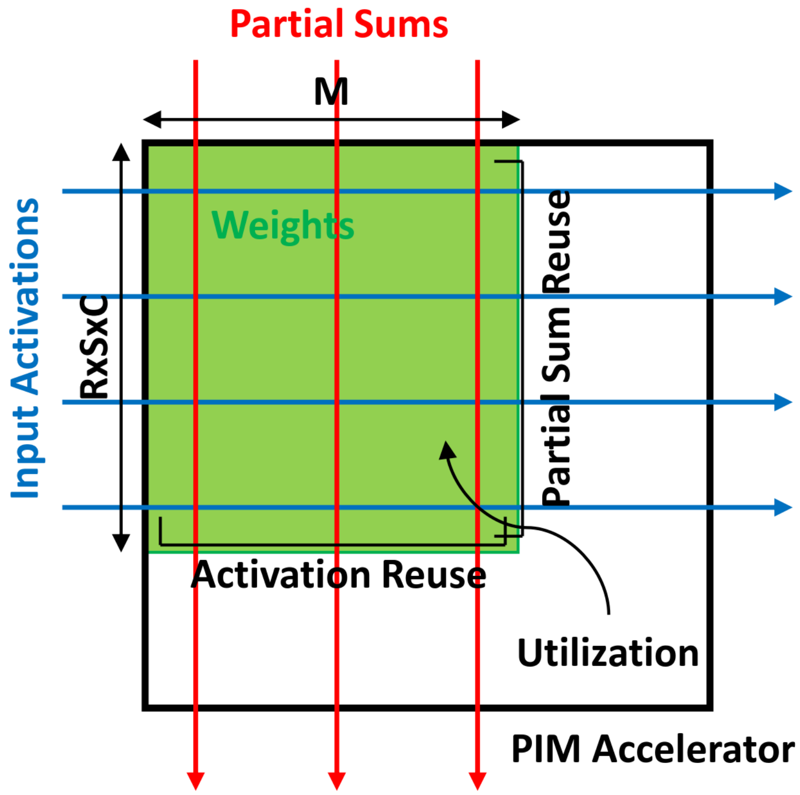
- Recent DNNs achieve higher accuracy with fewer MACs and weights
- However, the decrease in MACs and weights can be accompanied by an increase in the number of activations
  - Activations are much more expensive than weights and MACs in PIM!



Impact of Array Size on Utilization

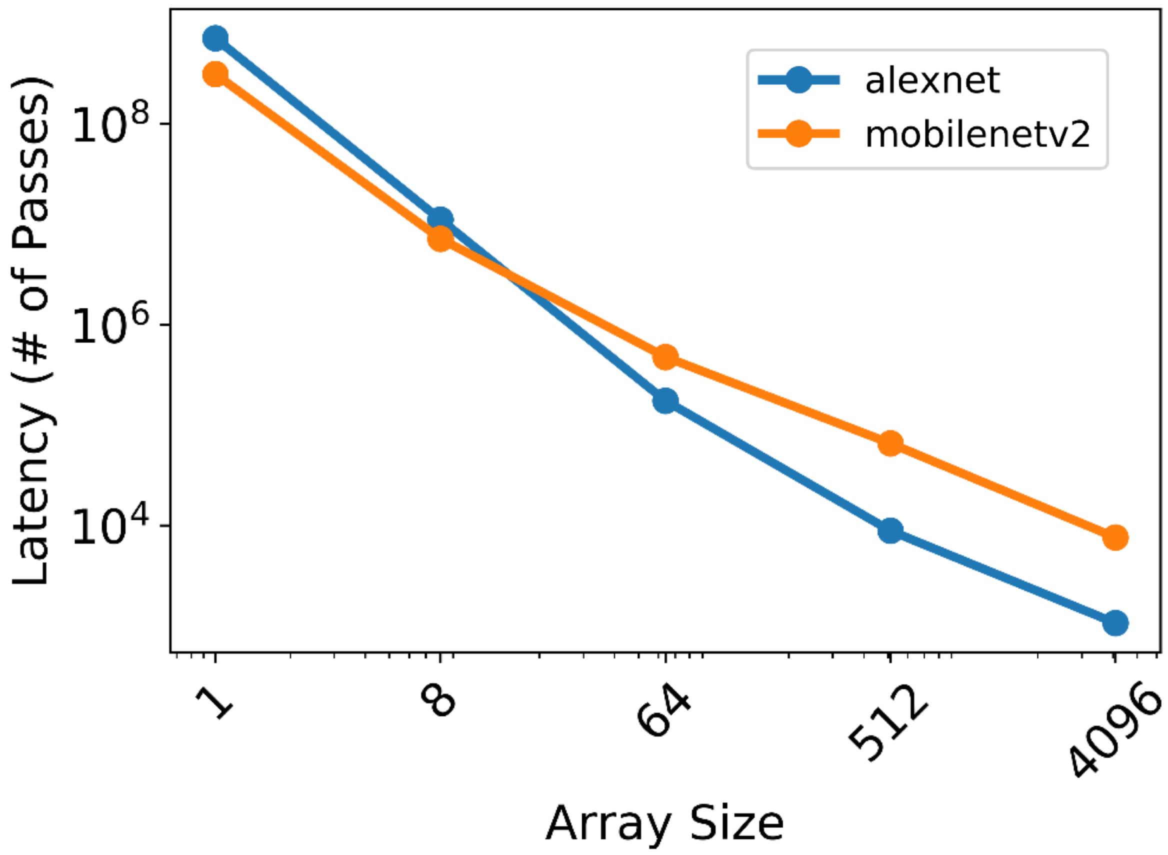
- PIM accelerators often have a large array size to amortize cost of peripheral circuits
  - Digital: 16x16 → 128x128
  - PIM: 128x128 → 4096x4096

- Number of MACs used in array depends on filter size
  - Recent DNNs have smaller filters
  - However, smaller filter means lower utilization!



Impact of Array Size on Utilization

- Lower utilization causes
  - Fewer MACs are processed in parallel → **Increased latency**
  - Reduce data reuse of activations → **Increased energy consumption**

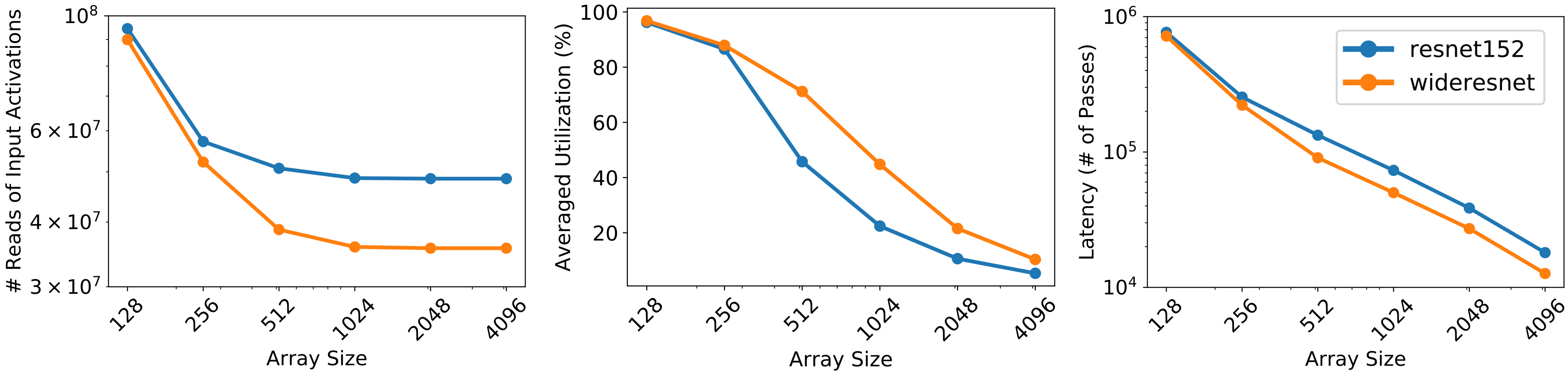


Shallower DNNs with larger layers may benefit more from the large array in PIM

This goes against recent trend in the design of DNNs for digital accelerators

Hardware Efficiency – Trade-Off

- Without lowering the accuracy, reducing the depth and increasing the filter size can increase the hardware efficiency
- Example: Wide ResNet [**Zagoruyko**, *BMVC 2017*] versus ResNet152 [**He**, *CVPR 2016*]



Summary

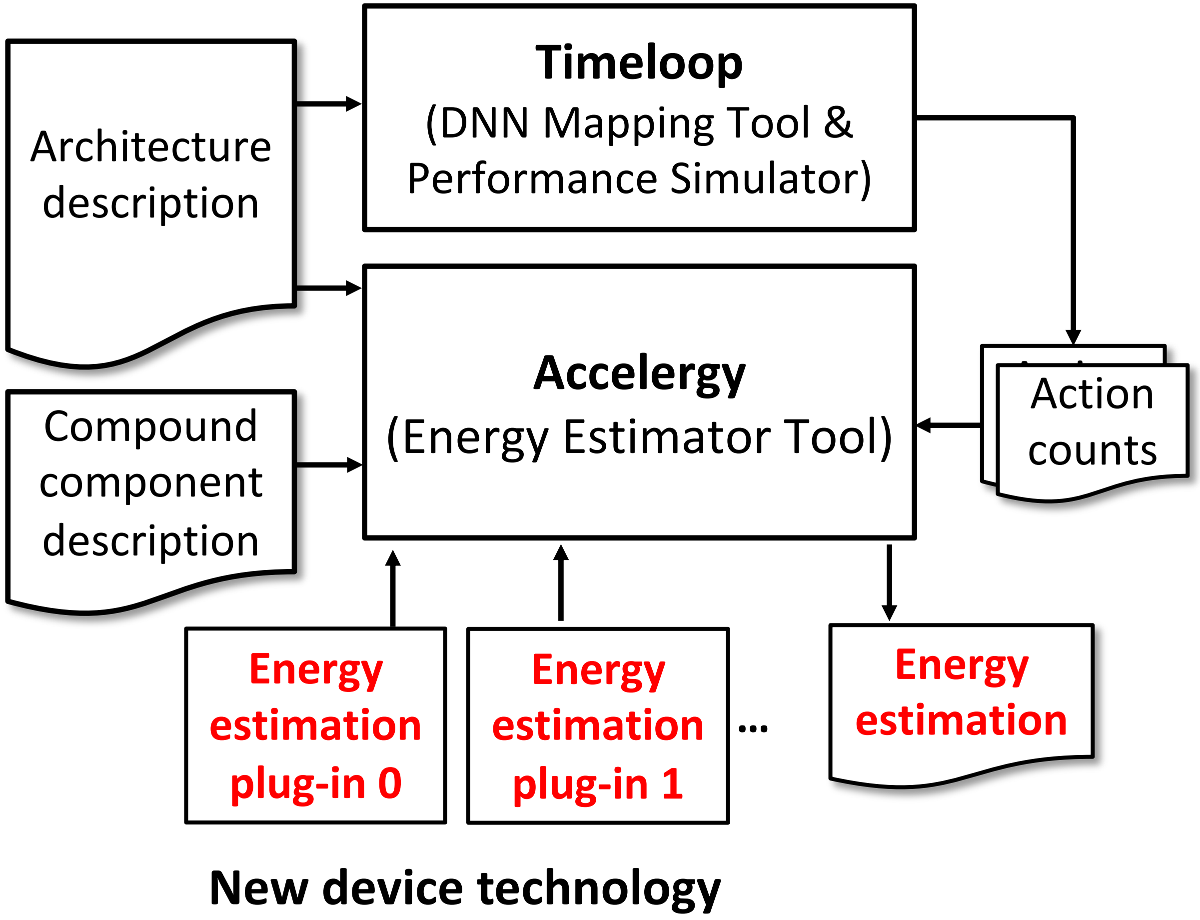
- Need to rethink design of network architecture of DNNs for PIM
  - Design approaches that achieve high accuracy and efficiency on digital accelerators does ***NOT*** necessarily translate to PIM
- In addition to the number of weights, MACs, and noise-free accuracy, design of DNN for PIM should consider
  - the sensitivity to non-idealities and lower bit widths
  - the movement of activations
  - the array utilization
- New line of research – design new DNN network architectures for PIM
  - e.g., Making DNNs shallower with larger layers may be preferable

We thank Tayfun Gokmen for helpful discussions. This work was supported by the MIT-IBM Watson AI Lab, the MIT Quest for Intelligence, and by the NSF E2CDA 1639921

DNN Processor Evaluation Tools

Evaluate Impact of Emerging Devices

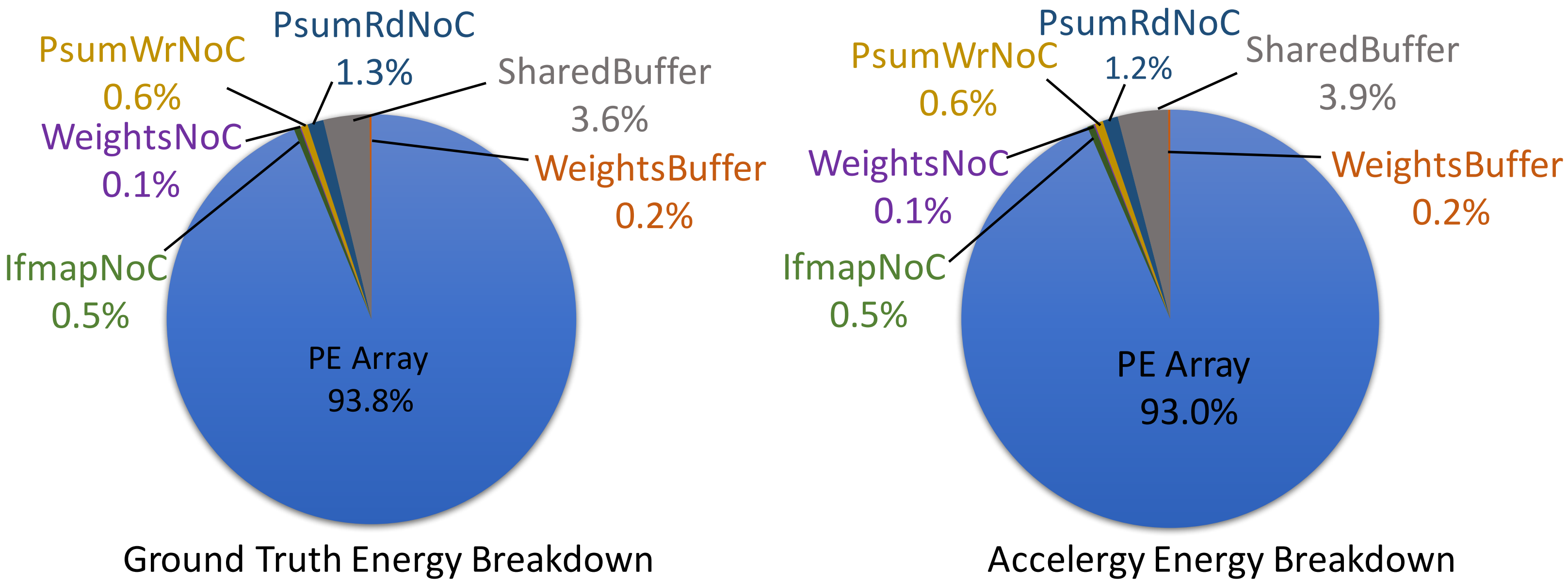
- Require systematic way to
  - Evaluate and compare wide range of DNN processor designs
  - Rapidly explore design space
- **Accelergy** [**Wu**, *ICCAD* 2019]
  - Early stage energy estimation tool at the architecture level
  - Evaluate architecture level energy impact of emerging devices
- **Timeloop** [**Parashar**, *ISPASS* 2019]
  - DNN mapping tool
  - Performance Simulator → Action counts



Open-source code available at: http://accelergy.mit.edu

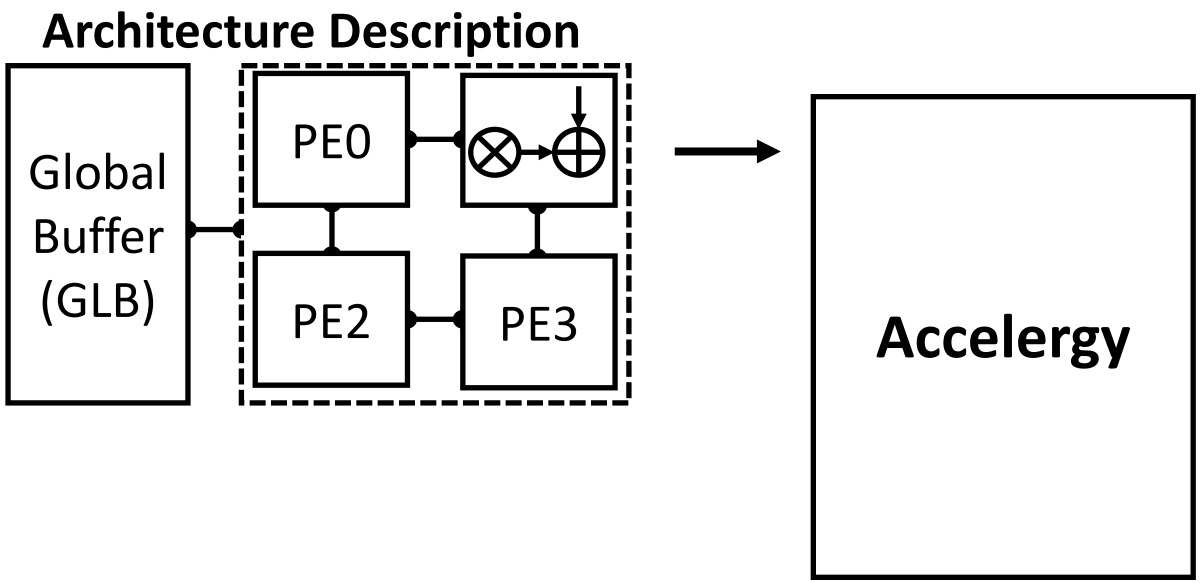
Accelergy Estimation Validation

- Validation on Eyeriss [**Chen**, *ISSCC* 2016]
  - Achieves 95% accuracy compared to post-layout simulations
  - Can accurately captures energy breakdown at different granularities



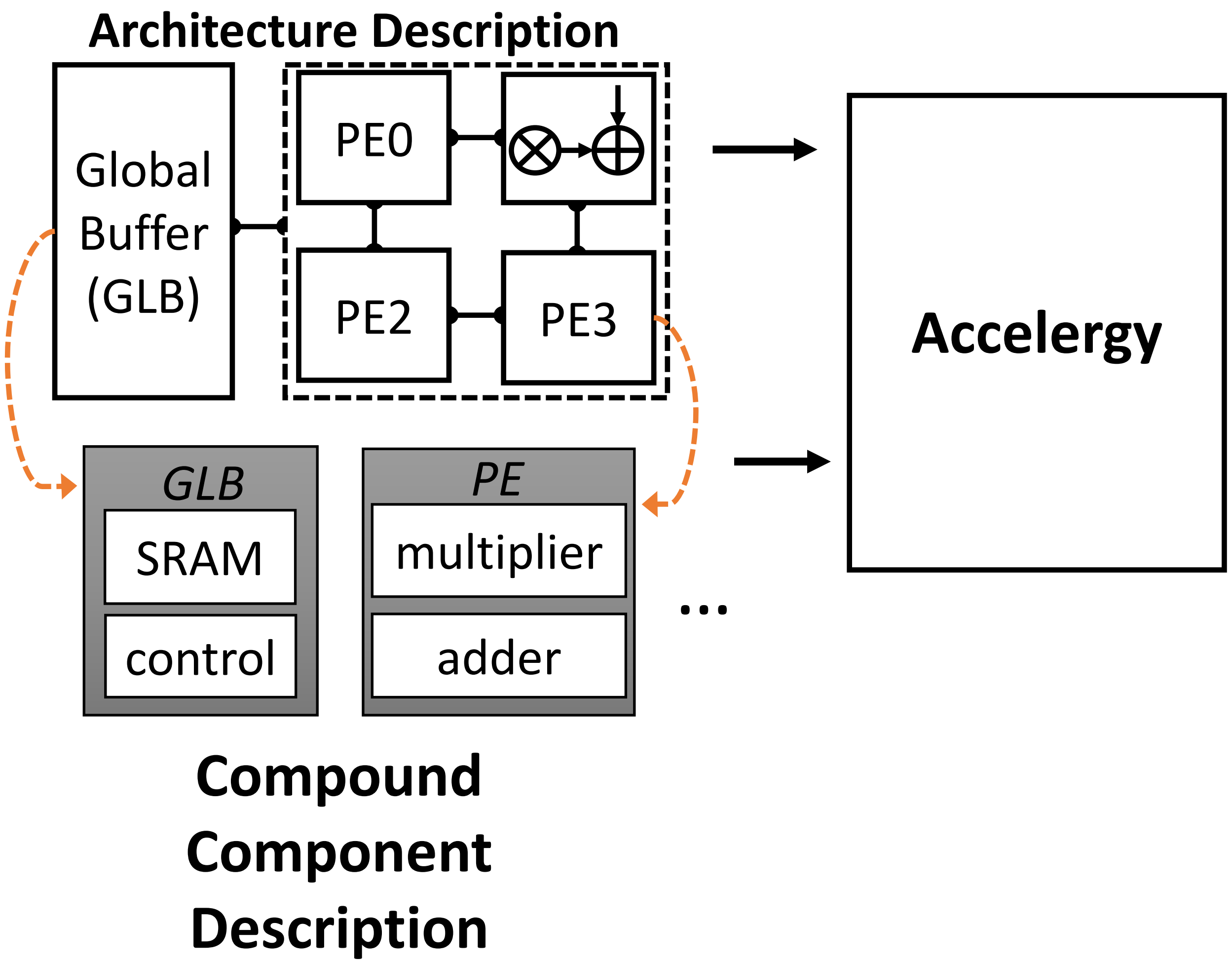
Open-source code available at: http://accelergy.mit.edu

Accelergy Infrastructure



Open-source code available at: http://accelergy.mit.edu

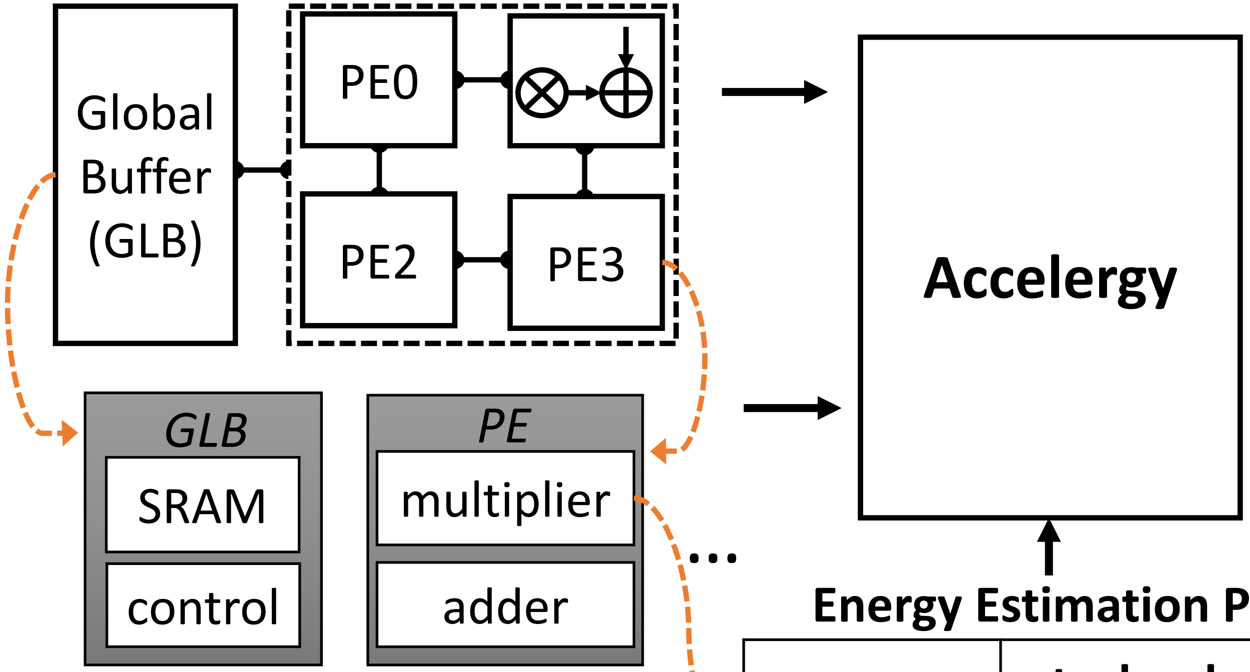
Accelergy Infrastructure



Open-source code available at: http://accelergy.mit.edu

Accelergy Infrastructure

Architecture Description



Compound
Component
Description

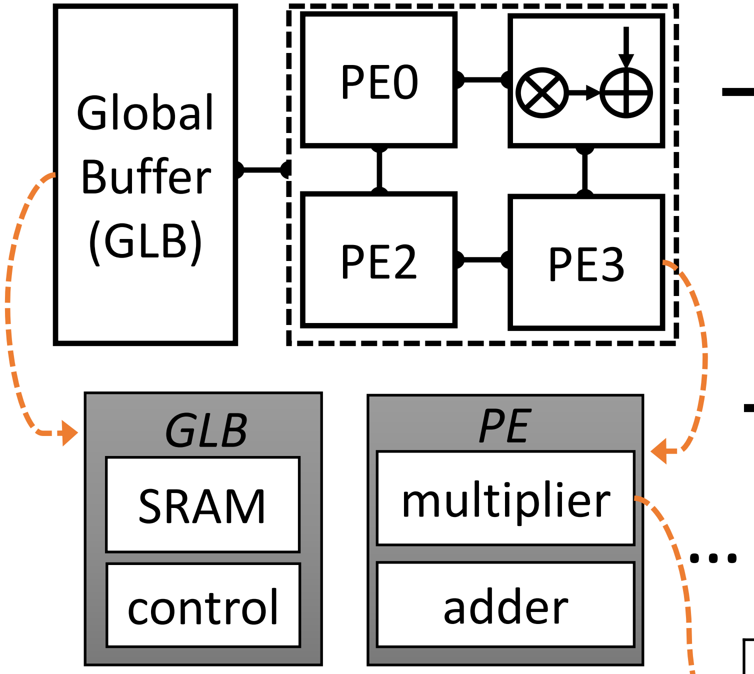
Energy Estimation Plug-in

name	technology	width	action	energy (pJ)
multiplier	65nm	16	multiply	0.8
adder	…			

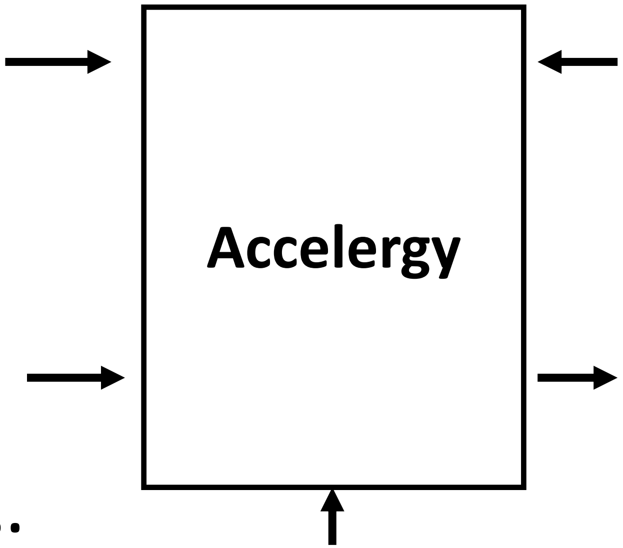
Open-source code available at: http://accelergy.mit.edu

Accelergy Infrastructure

Architecture Description



Compound Component Description



Action Counts

name	action	count
PE0	compute	500
PE1	…	

Energy Estimation

name	energy (pJ)
PE0	1500
PE1	…

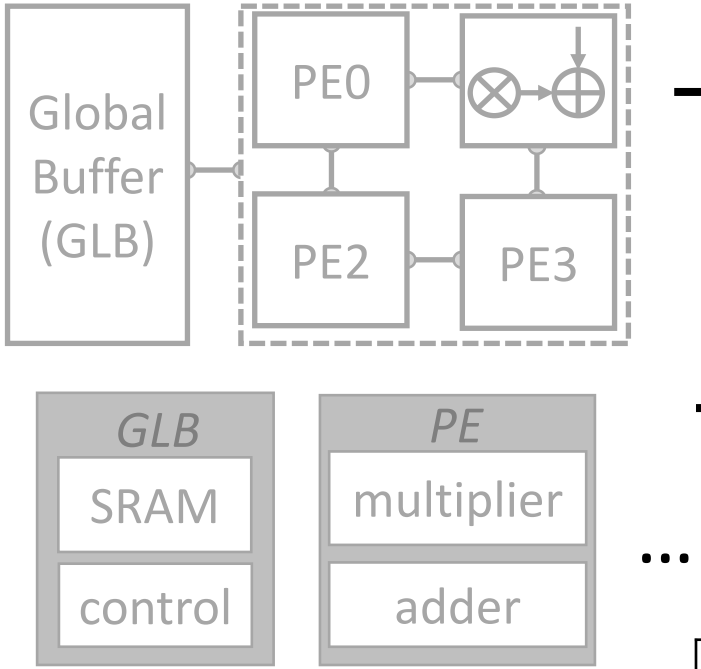
Energy Estimation Plug-in

name	technology	width	action	energy (pJ)
multiplier	65nm	16	multiply	0.8
adder	…			

Open-source code available at: http://accelergy.mit.edu

Estimation for a Different Process Technology

Architecture Description



Compound Component Description

Accelergy

Action Counts

name	action	count
PE0	compute	500
PE1	...	

Energy Estimation

name	energy (pJ)
PE0	~~1500~~ 600
PE1	...

Energy consumption impacts are reflected here

Energy Estimation Plug-in

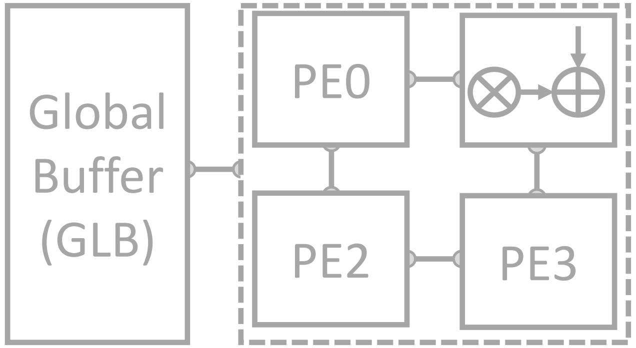
name	technology	width	action	energy (pJ)
multiplier	~~65nm~~ 45nm	16	multiply	~~0.8~~ 0.4
adder	...			

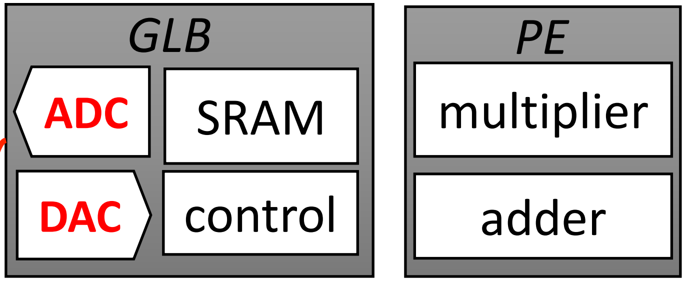
Simple updates in the original table

http://accelergy.mit.edu

Estimation for PIM Accelerators

Architecture Description





…

Compound Component Description

Redefine compound component

Accelergy

Action Counts

name	action	count
PE0	compute	500
PE1	…	

Energy Estimation

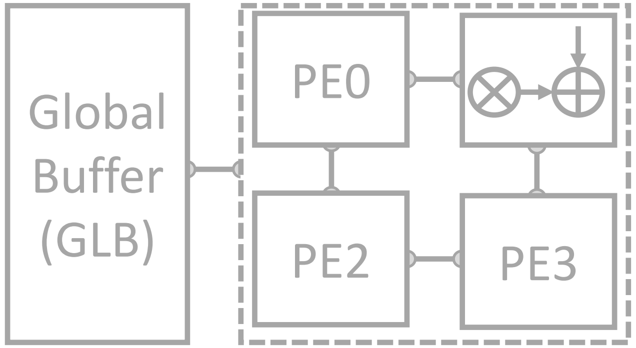
name	**energy (pJ)**
PE0	~~1500~~ E_{total}
PE1	…

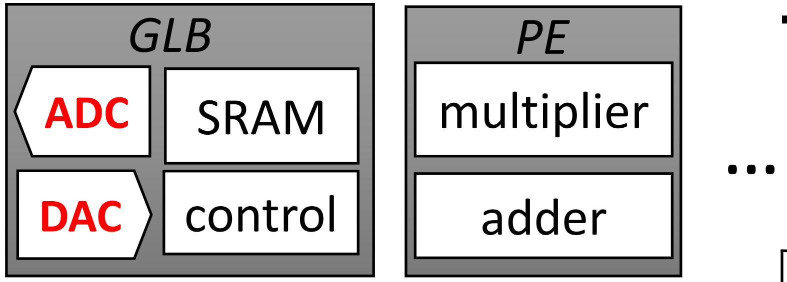
Energy Estimation Plug-in

name	**technology**	**width**	**action**	**energy (pJ)**
multiplier	~~65nm~~ **memristor**	16	multiply	~~0.8~~ E_{mult}
adder	…			
ADC	…			
DAC	…			

Estimation for PIM Accelerators

Architecture Description





Compound Component Description

Update the original table with additional building blocks

Accelergy

Action Counts

name	action	count
PE0	compute	500
PE1	...	

Energy Estimation

name	energy (pJ)
PE0	~~1500~~ E_{total}
PE1	...

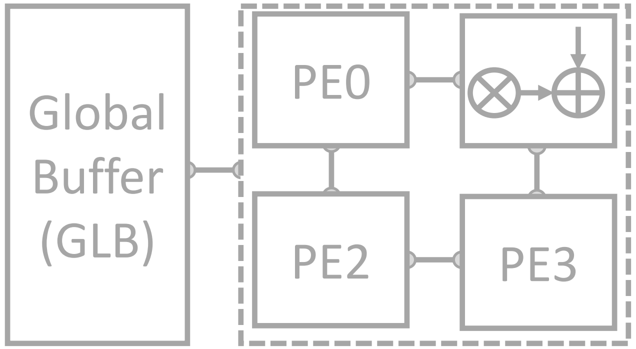
Energy Estimation Plug-in

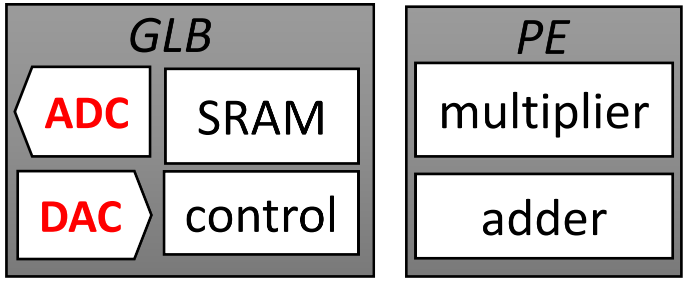
name	technology	width	action	energy (pJ)
multiplier	~~65nm~~ memristor	16	multiply	~~0.8~~ E_{mult}
adder	...			
ADC	...			
DAC	...			

http://accelergy.mit.edu

Estimation for PIM Accelerators

Architecture Description





...

Compound Component Description

Accelergy

Action Counts

name	action	count
PE0	compute	500
PE1	...	

Energy Estimation

name	energy (pJ)
PE0	~~1500~~ E_{total}
PE1	...

*Populate the multiplication energy of **your technology** here!*

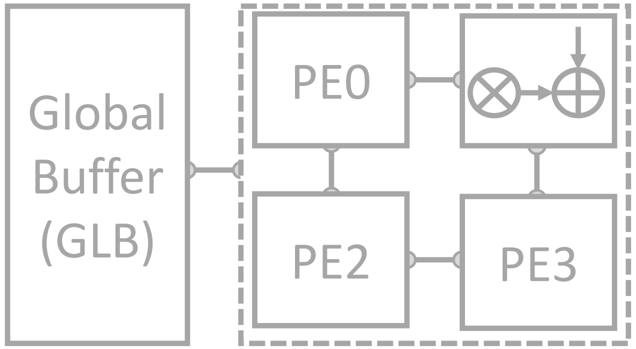
Energy Estimation Plug-in

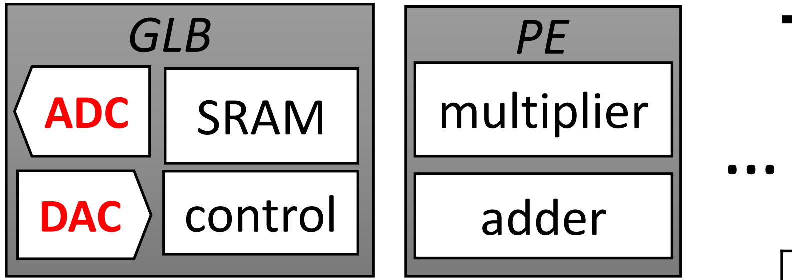
name	technology	width	action	energy (pJ)
multiplier	~~65nm~~ memristor	16	multiply	~~0.8~~ E_{mult}
adder	...			
ADC	...			
DAC	...			

http://accelergy.mit.edu

Estimation for PIM Accelerators

Architecture Description





Compound Component Description

Accelergy

Action Counts

name	action	count
PE0	compute	500
PE1	…	

Energy Estimation

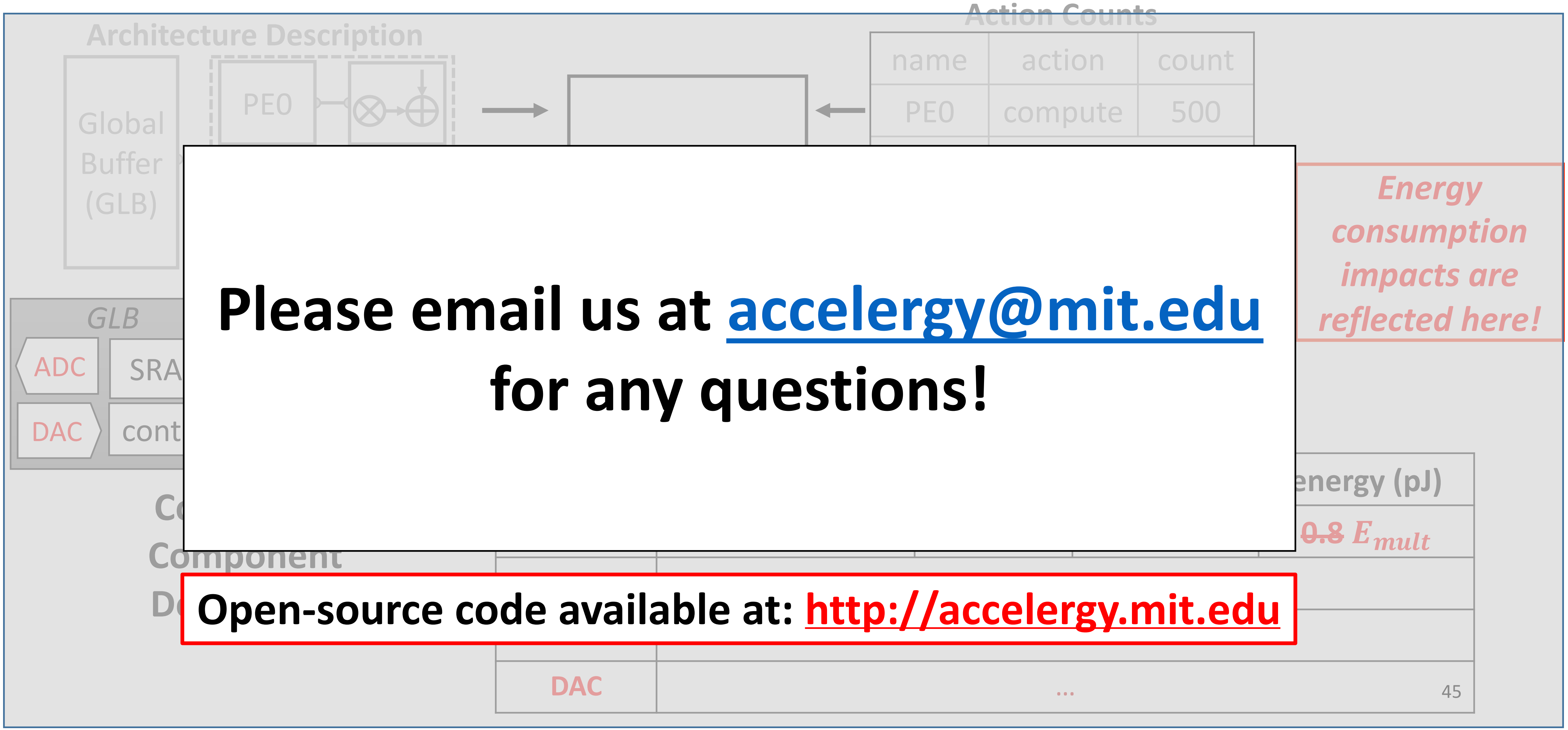
name	**energy (pJ)**
PE0	~~1500~~ E_{total}
PE1	**…**

Energy consumption impacts are reflected here!

Energy Estimation Plug-in

name	**technology**	**width**	**action**	**energy (pJ)**
multiplier	~~65nm~~ memristor	16	multiply	~~0.8~~ E_{mult}
adder	…			
ADC	…			
DAC	…			

Estimation for PIM Accelerators



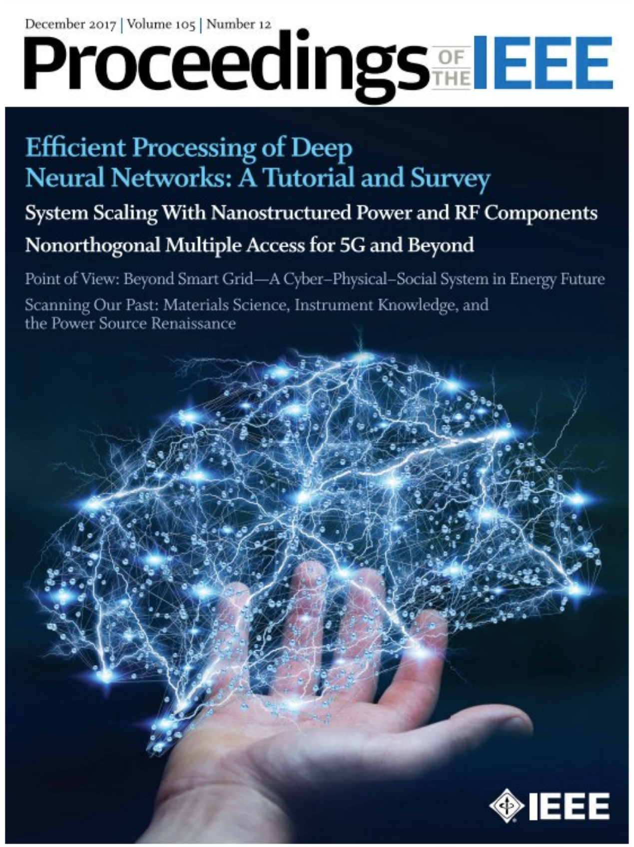
Please email us at accelergy@mit.edu for any questions!

Open-source code available at: http://accelergy.mit.edu

Resources

- Today's slides available at http://sze.mit.edu
- Efficient Processing of Deep Neural Networks
  http://eyeriss.mit.edu/tutorial.html
- NeurIPS tutorial: https://slideslive.com/38921492
- MIT Professional Education Course on "**Designing Efficient Deep Learning Systems**"
  http://professional-education.mit.edu/deeplearning
- For Research Updates

Follow @eems_mit



Book Coming Soon!

Join EEMS news mailing list http://mailman.mit.edu/mailman/listinfo/eems-news