Boosting ReRAM-based DNN by Row Activation Oversubscription

Mengyu Guo, Zihan Zhang, Jianfei Jiang, Qin Wang, and Naifeng Jing*

Department of Micro/Nano Electronics,
Shanghai Jiao Tong University

ASP-DAC 2022, Jan. 17-20

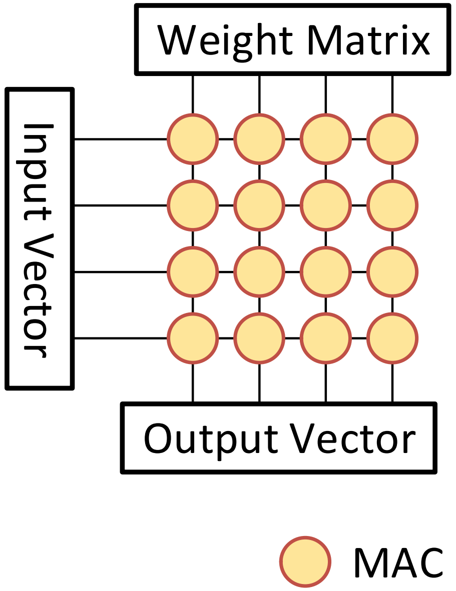
Outline

- **Background**
- **Motivation**
- **Proposed methods**
  - ***RAOS and Prediction***
  - ***Adequate RAOS Rate***
  - ***Architecture Design***
- **Experiment and Results**

ReRAM-based DNN

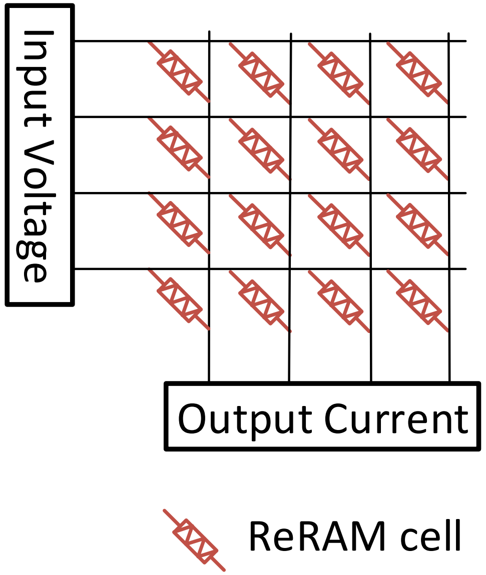
A digital way

$$Vector_{out} = Vector_{in} \times Matrix$$



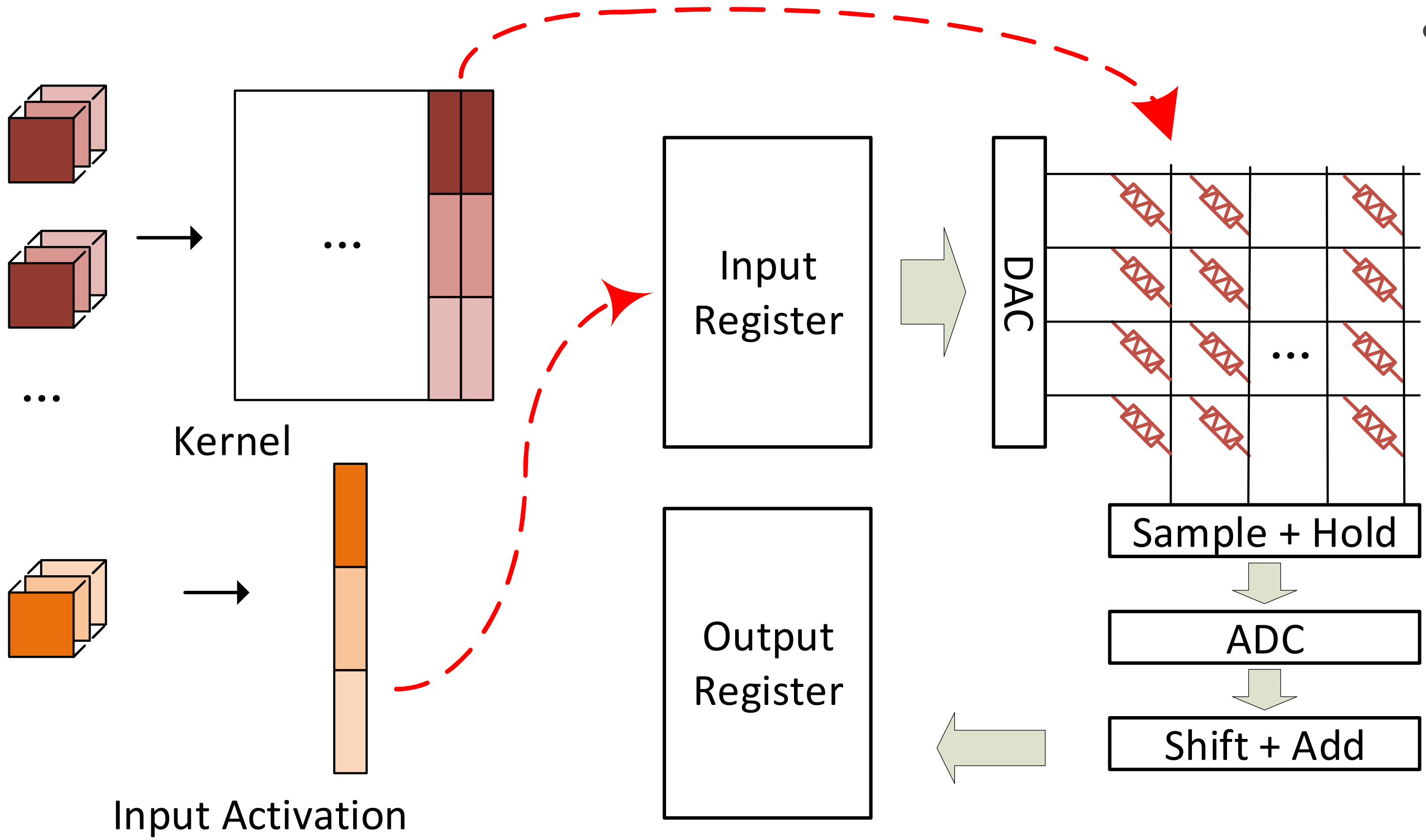
An analog way

$$Current_{out} = Voltage_{in} \cdot Cond_{ReRAM}$$

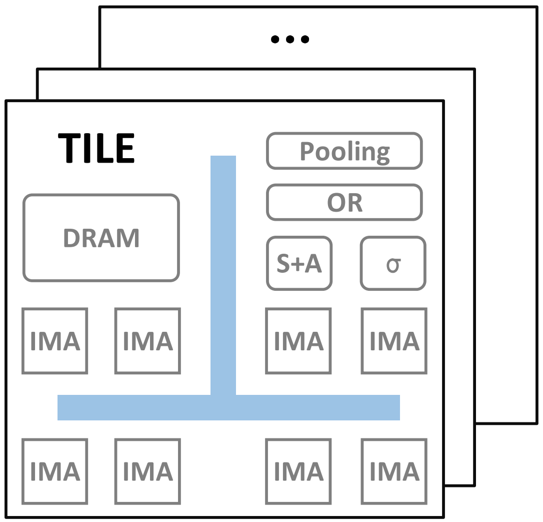


- An effective approach to the memory wall

ReRAM-based DNN



- Typical architecture
  - Hierarchical structure
  - Multi-stage pipeline



[A. Shafiee *et al*. ISCA'16]

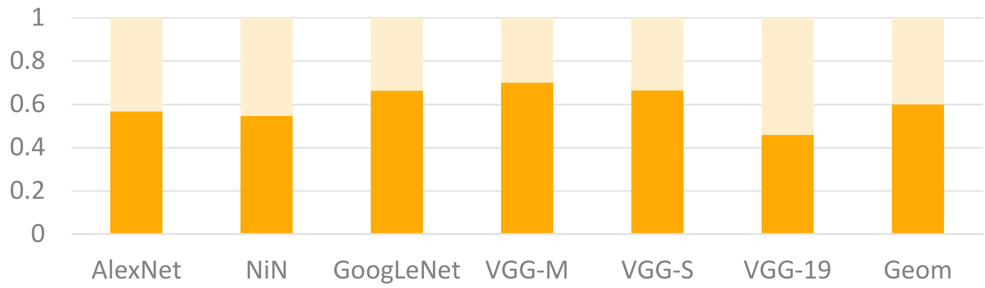
Motivation

- ADC's Huge overhead
  - High resolution

Component	Power (mW)	Area (mm^2)
ADC	**16**	**0.0096**
DAC	4	0.00017
Memristor	2.4	0.0002
IR	1.24	0.0021
OR	0.23	0.00077
Total	**24.08**	**0.0131**

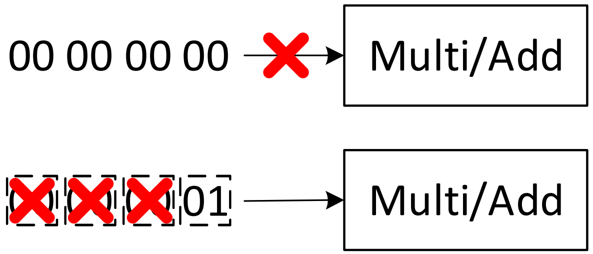
[A. Shafiee *et al*. ISCA'16]

- Useless calculations



[J. Albericio *et al*. MICRO'16]

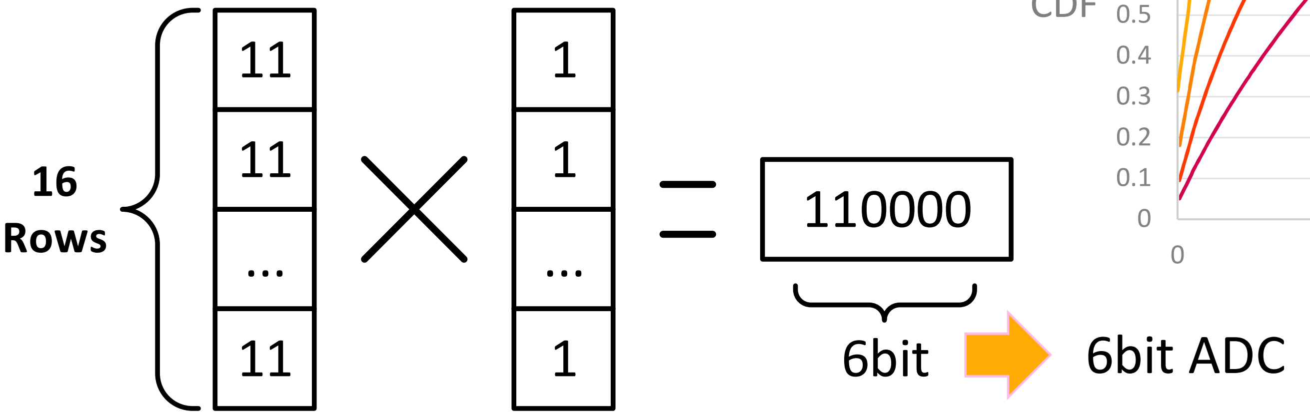
- Value
- Slice



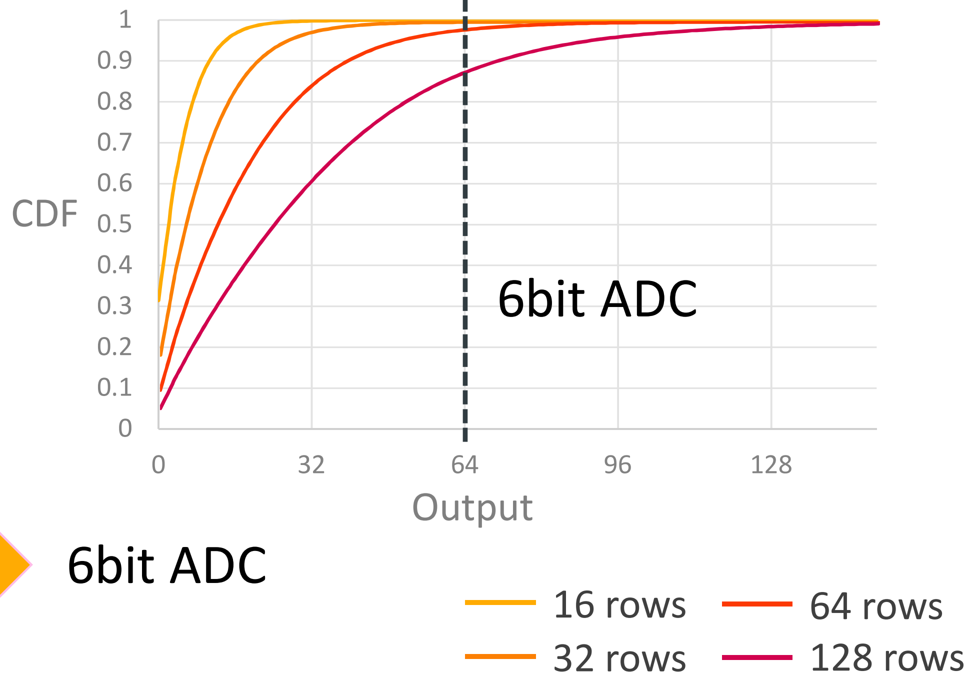
Motivation

- ADC designed for the worst case

$$Out_j = \sum_{i=0}^{R-1} In_i \times W_{ij} \leq 2^{r_{out}} - 1$$



- Test in ResNet50
  - 16 rows parallel
  - 32/64/128... rows parallel

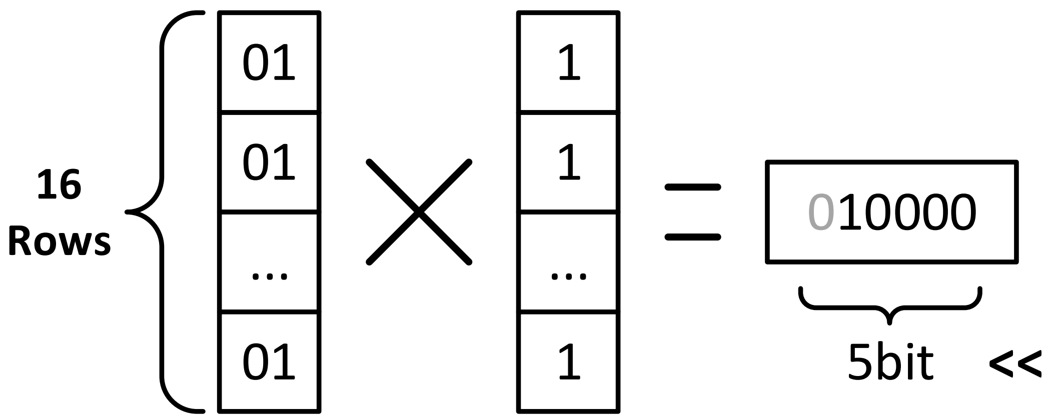


Explore greater computational parallelism by data sparsity

Motivation

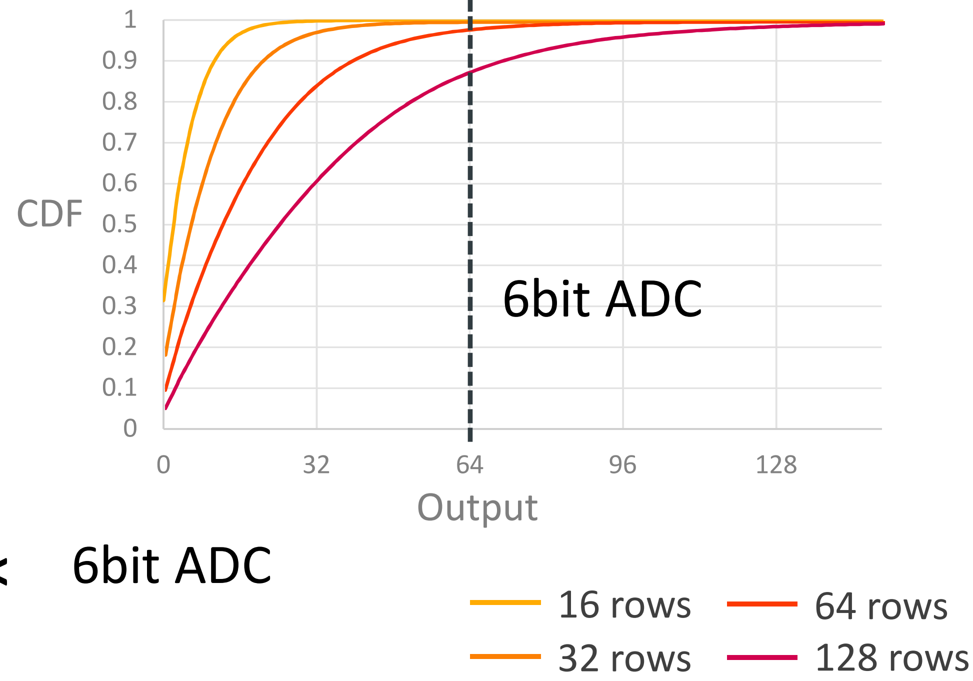
- ADC designed for the worst case

$$Out_j = \sum_{i=0}^{R-1} In_i \times W_{ij} \le 2^{r_{out}} - 1$$



<< 6bit ADC

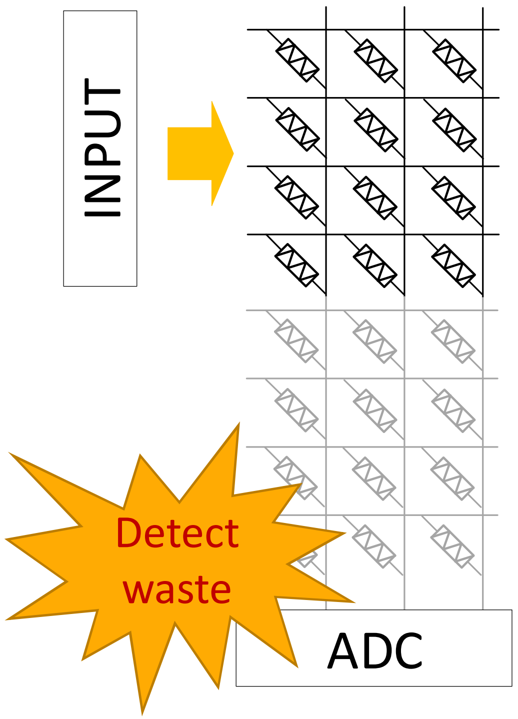
- Test in ResNet50
  - 16 rows parallel
  - 32/64/128… rows parallel



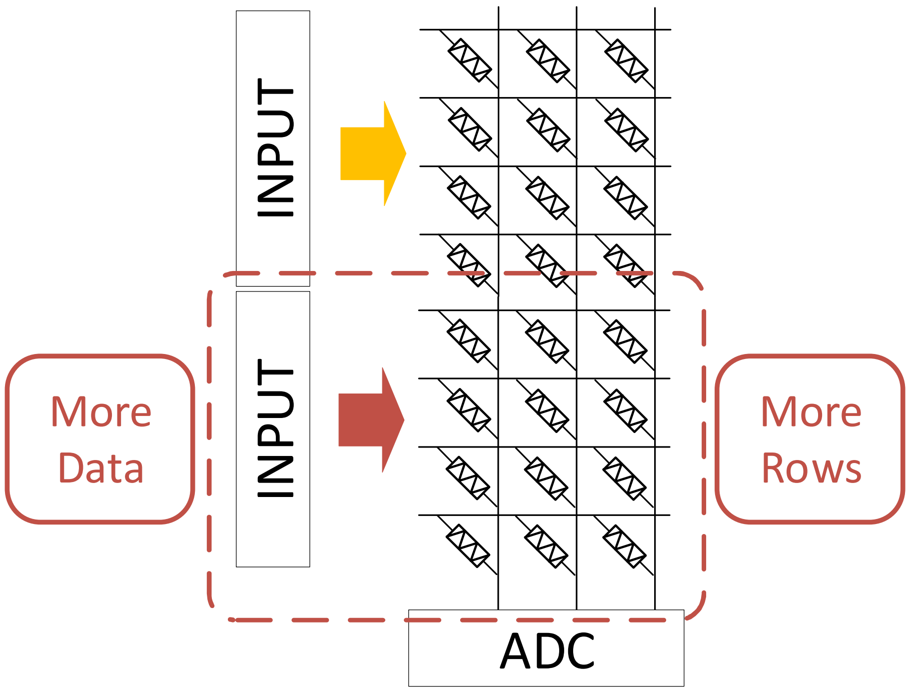
Explore greater computational parallelism by data sparsity

Motivation

- When ADC is going to be wasted



- Activate more rows
- Transfer more data



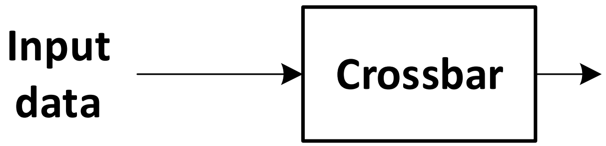
Row activation oversubscription(RAOS)

Our works

- **Explore greater computational parallelism by data sparsity**
  - **Proposed RAOS to dynamically increase the computing parallelism**
  - **Proposed a Predicting Unit to find the upper bound of the results without hurting MVM accuracy**
  - **Proposed two Prediction Schemes to maximize the oversubscription rate for MVM calculation**

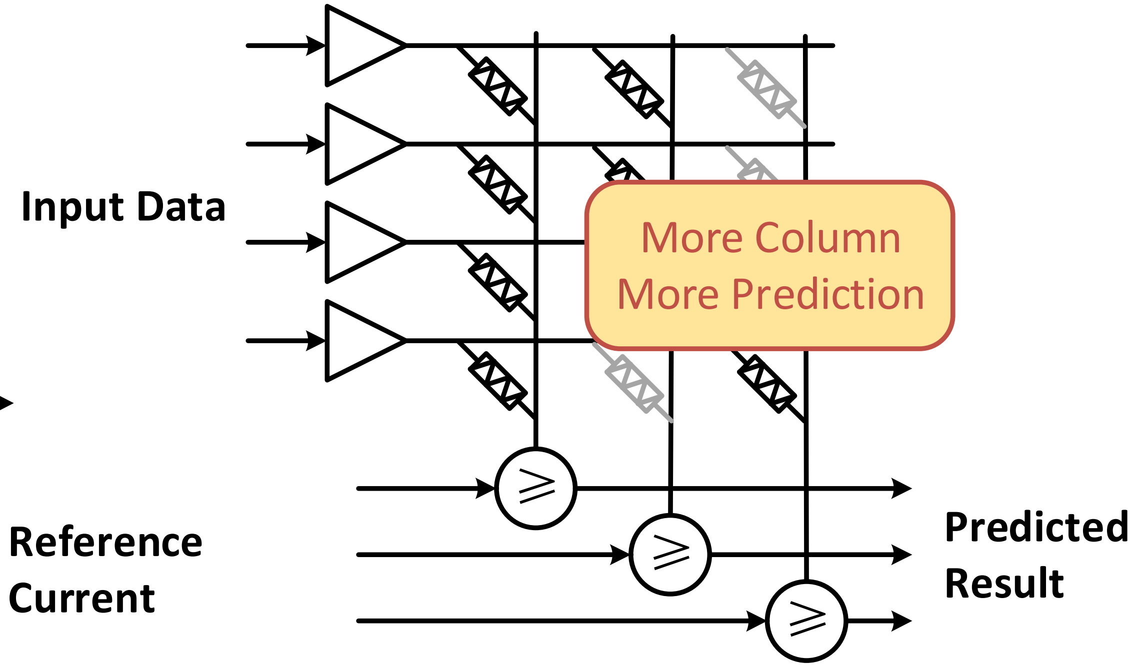
RAOS and Prediction

- Row activation oversubscription



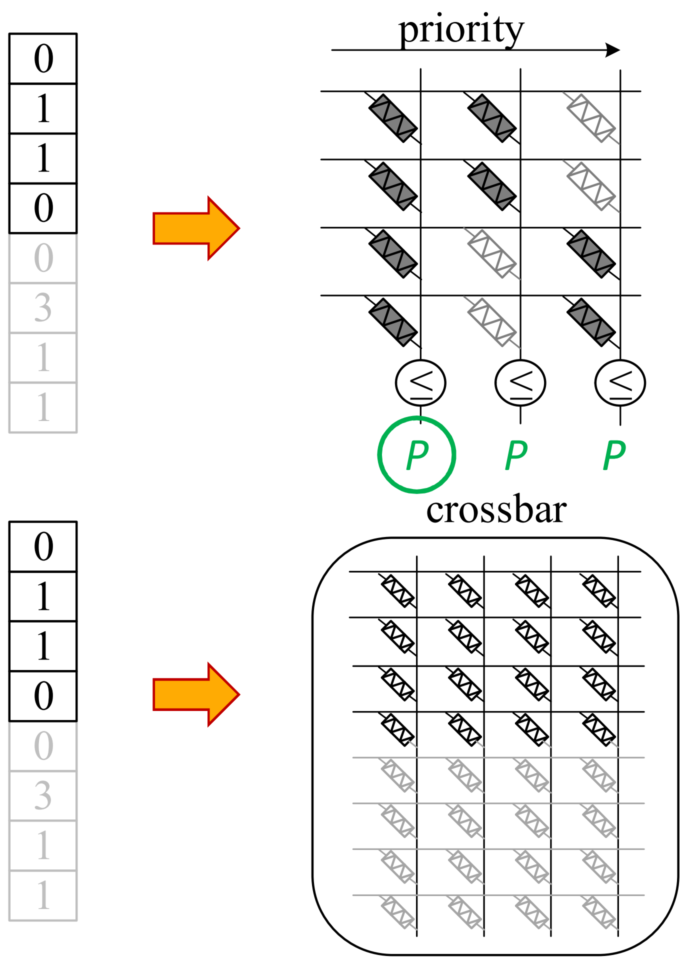
- How to predict?

$$Out \leq \max(W_i) \times \sum_{i=0}^{R-1} In_i = Out_{max}$$



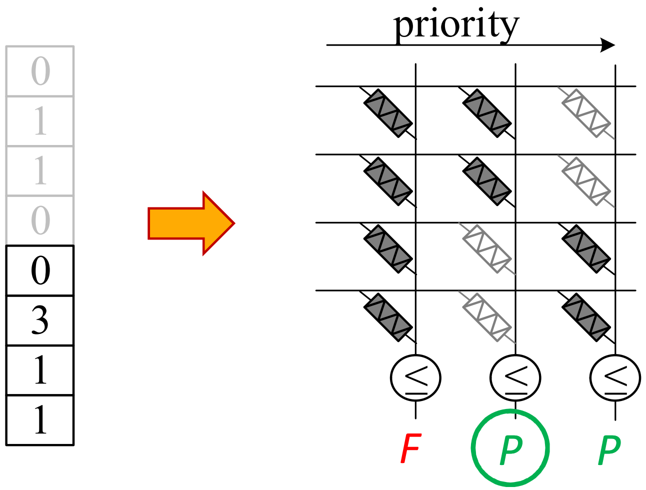
Adequate RAOS Rate — Binary Prediction

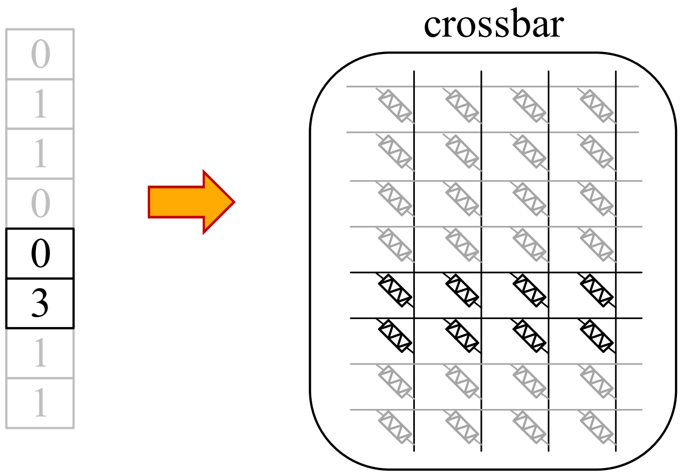
- Predicting Unit
  - Three predicting columns
  - Complementary columns
  - Work at the same time
- Pass
  - Compute in Crossbar
- Fail
  - Try to use lower RAOS rate
  - Compute in Crossbar
  - Compute the rest



Adequate RAOS Rate — Binary Prediction

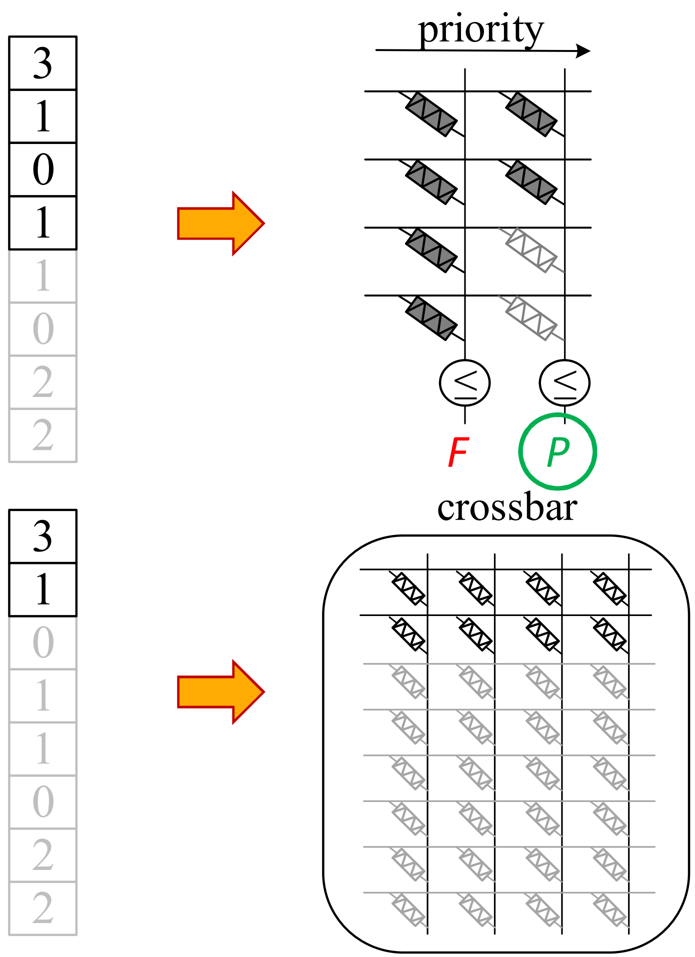
- Predicting Unit
  - Three predicting columns
  - Complementary columns
  - Work at the same time
- Pass
  - Compute in Crossbar
- Fail
  - Try to use lower RAOS rate
  - Compute in Crossbar
  - Compute the rest





Adequate RAOS Rate – Sliding Prediction

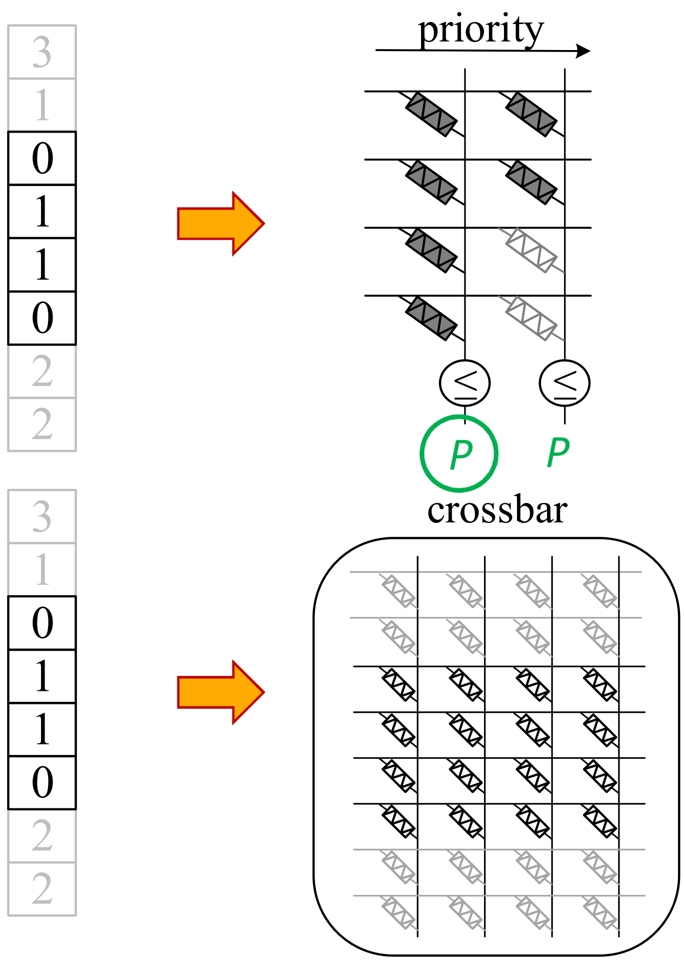
- Predicting Unit
  - Two predicting columns
  - One rate for one column
  - Work at the same time
- Fail
  - Try to use lower RAOS rate
  - Compute in Crossbar
  - Make next predictions



Adequate RAOS Rate – Sliding Prediction

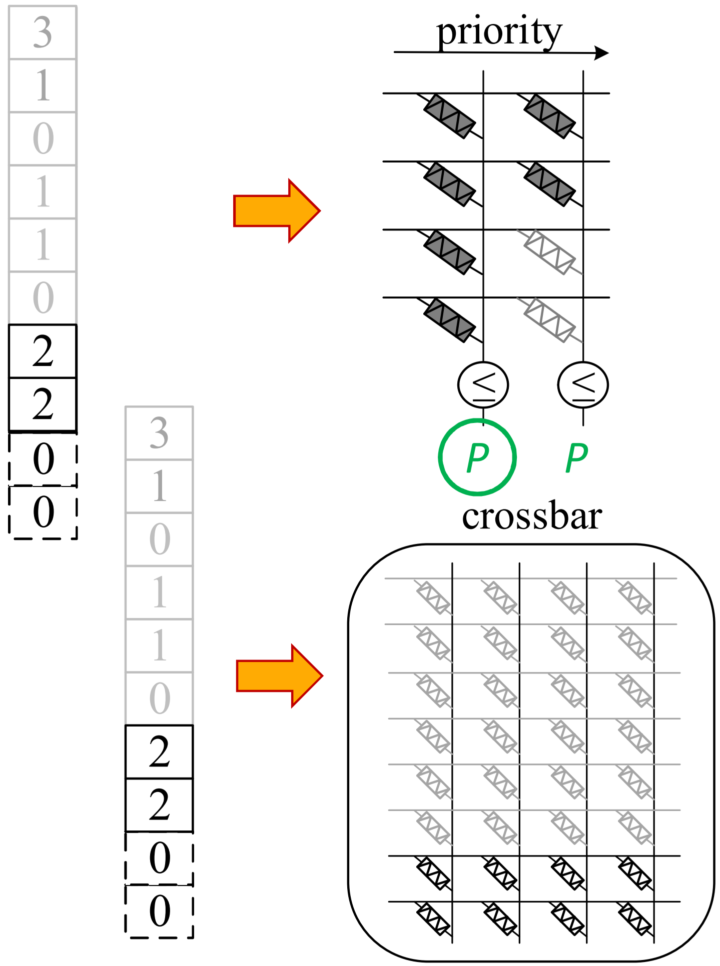
- Predicting Unit
  - Two predicting columns
  - One rate for one column
  - Work at the same time

- Fail
  - Try to use lower RAOS rate
  - Compute in Crossbar
  - Make next predictions



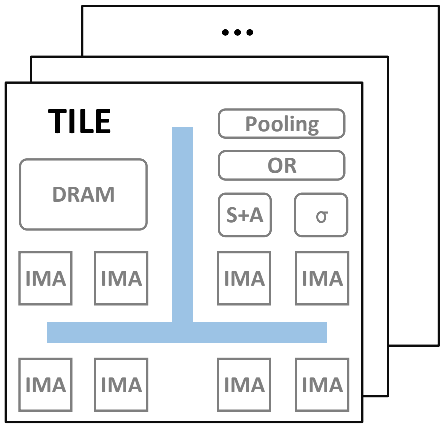
Adequate RAOS Rate – Sliding Prediction

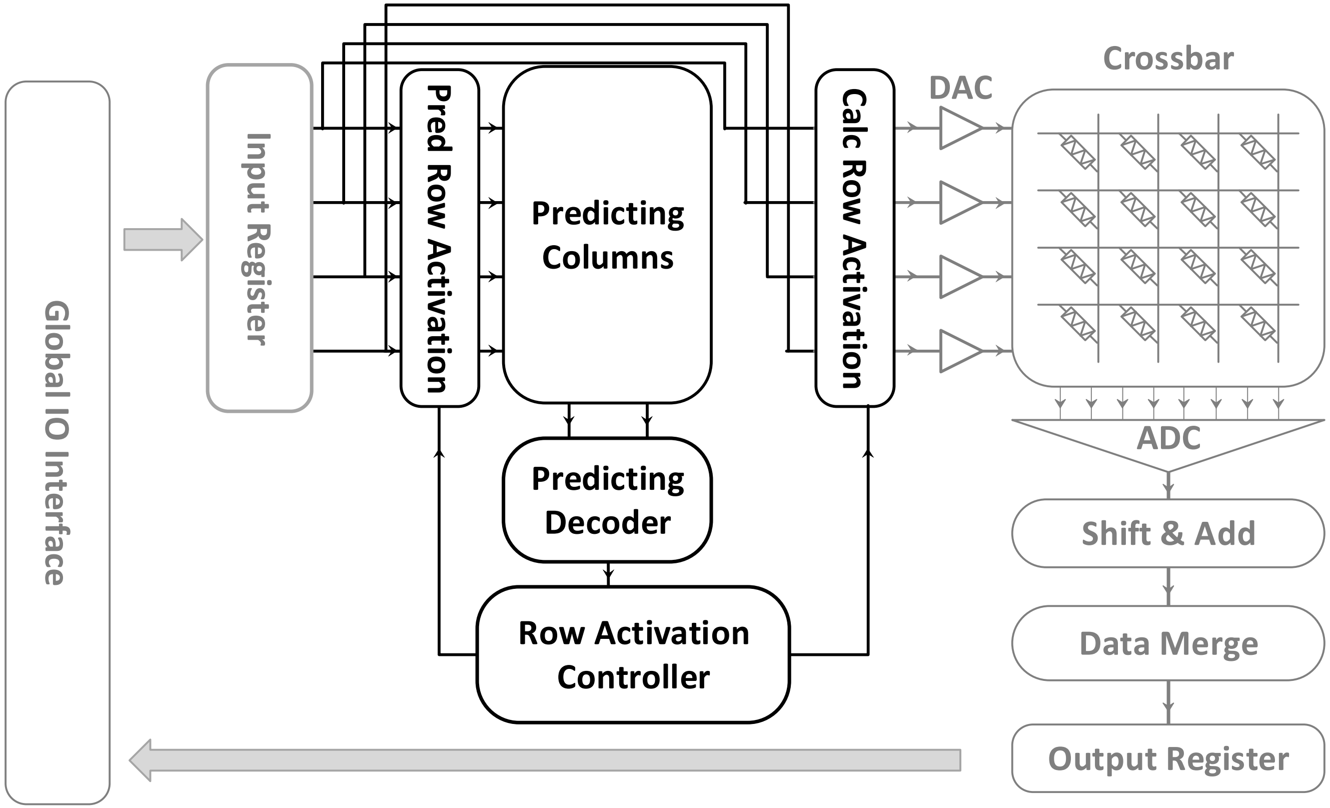
- Predicting Unit
  - Two predicting columns
  - One rate for one column
  - Work at the same time
- Fail
  - Try to use lower RAOS rate
  - Compute in Crossbar
  - Make next predictions
- Boundary conditions
  - Padding



Overall Architecture Design

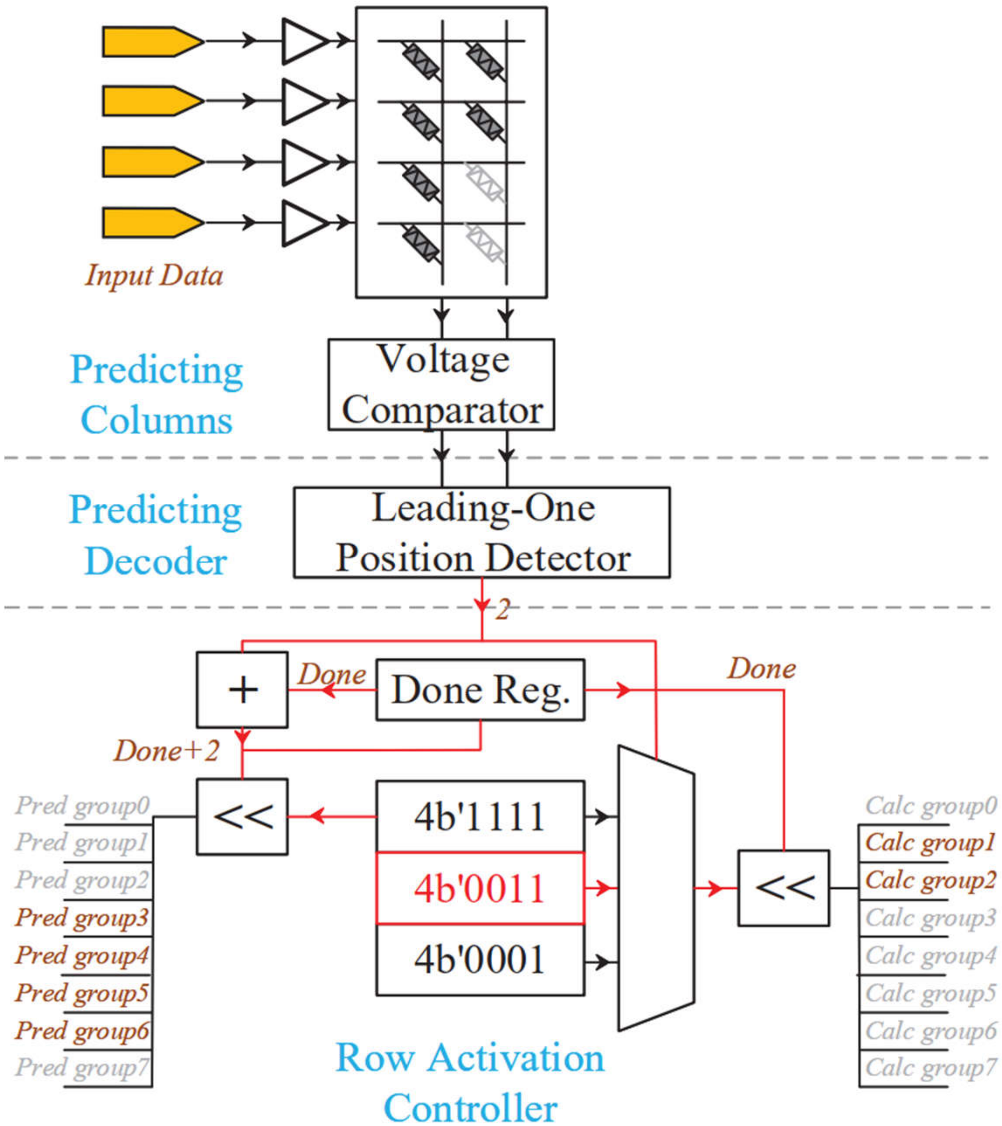
- RAOS mainly built upon a crossbar
  - Predicting columns
  - Predicting decoder
  - Row activation controller
- Hierarchical Architecture
  - Chips, Tiles, and IMA
  - Similar to ISAAC





Core Components

- Predicting columns
  - Voltage comparator
  - 2bit "pass/fail" output vector
- Predicting decoder
  - Predict vector ➡ RAOS rate
  - Leading-One Position Detector
- Row activation controller
  - Generate $Mask_{pred}$ & $Mask_{calc}$
  - Done register for update



Experimental setup

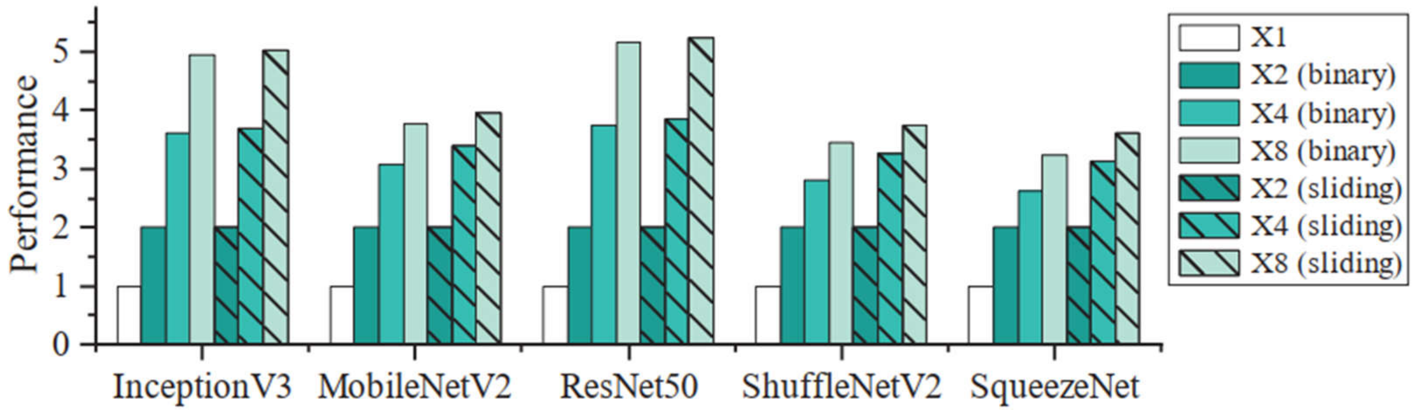
- Benchmarks
  - NN: ResNet50, InceptionV3, MobileNetV2, ShuffleNetV2, SqueezeNet
  - Dataset: ImageNet2012
  - We use the weight and feature map data without adding sparsity
- Simulation
  - Neuro-Sim
- Device Model
  - Crossbar size: 128x128
  - CACTI 6.5 RAM model@32nm
  - SAR-ADC @32nm
  - The same as ISAAC[1] and SRE[2]

ADC resolution designed for 16-row parallel computing

[1] A. Shafiee et al. ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars. ISCA, 2016.
[2] T. Yang et al. Sparse ReRAM Engine: Joint Exploration of Activation and Weight Sparsity in Compressed Neural Networks. ISCA, 2019.

Performance Evaluation

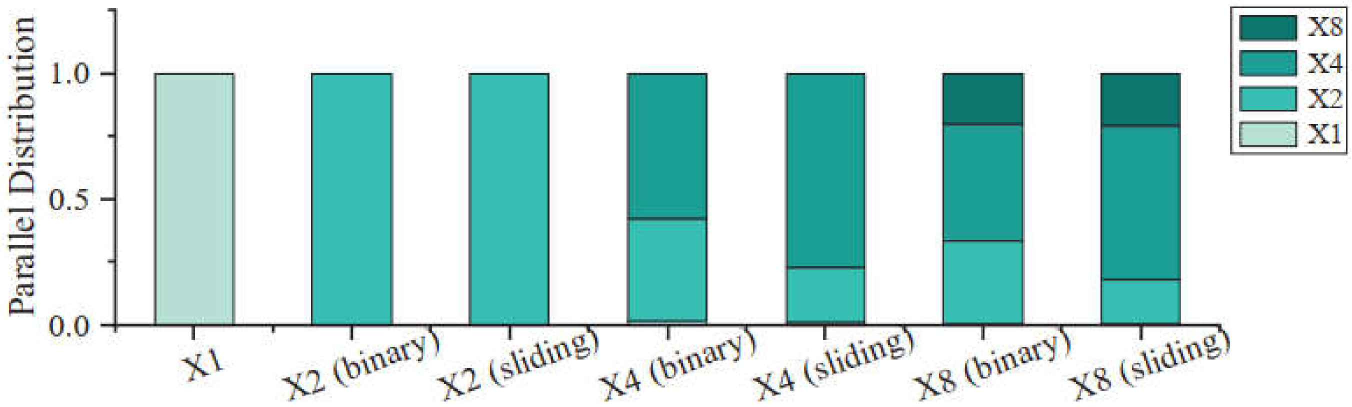
- Performance improvement using different RAOS rates
  - ×1 activates 16 rows without over-subscription



- Higher RAOS rate, higher performance
  - 1.97X(rate = 2), 5.1X(rate = 8) @ResNet50
- Sliding prediction is better!

Performance Evaluation

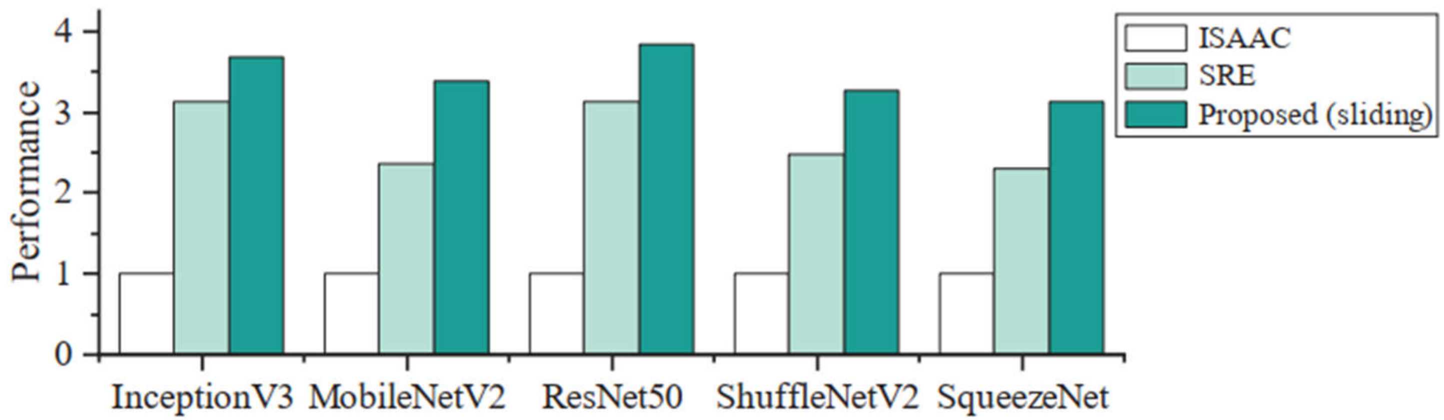
- Breakdown of the achieved rates using different prediction schemes
  - Network: ShuffleNetV2



- Sliding prediction results in higher RAOS rate

Performance Evaluation

- Performance comparison using sliding prediction
  - ×4 RAOS rate

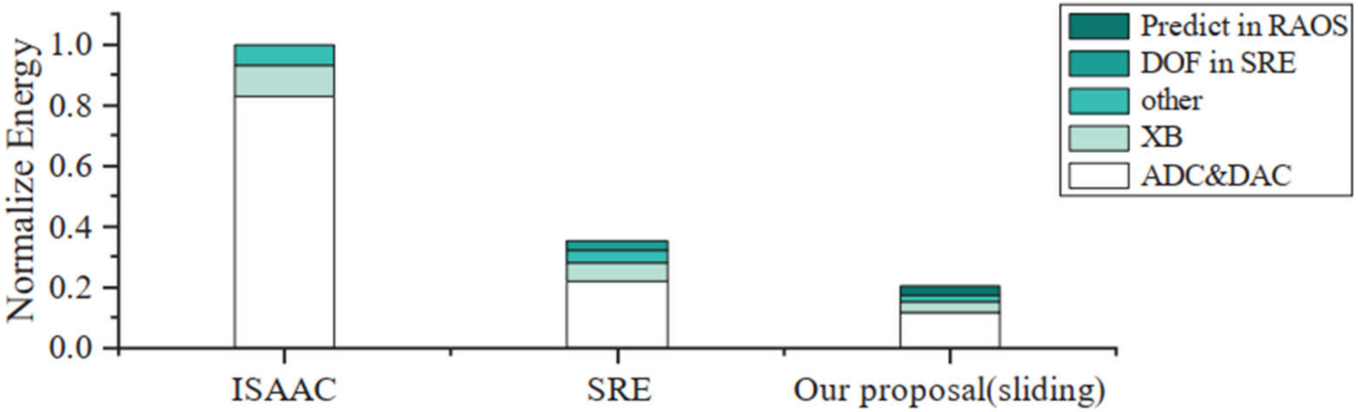


- More performance
  - *vs* ISAAC[1]: 3.1x ~ 3.8x
  - *vs* SRE[2]: 23% ~ 31%

[1] A. Shafiee et al. ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars. ISCA, 2016.
[2] T. Yang et al. Sparse ReRAM Engine: Joint Exploration of Activation and Weight Sparsity in Compressed Neural Networks. ISCA, 2019.

Energy Evaluation

- Energy consumption comparison
  - Network: ResNet50
  - Normalized to ISAAC



- Lower energy
  - *vs* ISAAC[1]: 4.9x
  - *vs* SRE[2]: 1.7x

[1] A. Shafiee et al. ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars. ISCA, 2016.
[2] T. Yang et al. Sparse ReRAM Engine: Joint Exploration of Activation and Weight Sparsity in Compressed Neural Networks. ISCA, 2019.

Conclusion

- RAOS for greater computational parallelism by data sparsity
  - A method for predicting output results
  - A method for dynamically adjusting computing resources
  - A method for reducing waste of ADC
- Compared with the state-of-the-art works
  - Performance: 3.8X (ISAAC), 1.2X (SRE)
  - Energy: 4.9X (ISAAC), 1.7X (SRE)

Boosting ReRAM-based DNN by Row Activation Oversubscription

Mengyu Guo, Zihan Zhang, Jianfei Jiang, Qin Wang,
and Naifeng Jing*

Thanks for your listening