A Heuristic and Greedy Weight Remapping Scheme with Hardware Optimization for Irregular Sparse Neural Networks Implemented on CIM Accelerator in Edge AI Applications

Lizhou Wu[1], Chenyang Zhao[1], Jingbo Wang[3], Xueru Yu[2], Shoumian Chen[2], Chen Li[2], Jun Han[1], Xiaoyong Xue[1], Xiaoyang Zeng[1]

[1] Fudan University, Shanghai, China

[2] Shanghai Integrated Circuits R&D Center, China

[3] Tongji University, China

Outline

- ❑ **Background & Motivation**
- ❑ **Proposed Irregular-Sparsity-Aware CIM Accelerator**
  - **Heuristic & Greedy Weight Remapping**
  - **Hardware Optimization: Zero-Skipping Circuit & Pipeline**
- ❑ **Experiment & Results**
- ❑ **Conclusion**

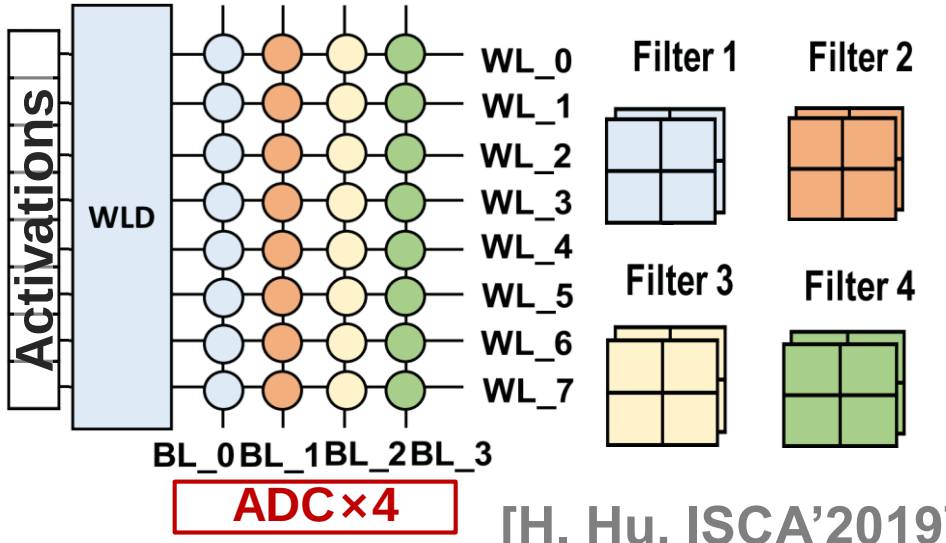
Outline

- ❑ **Background & Motivation**
- ❑ **Proposed Irregular-Sparsity-Aware CIM Accelerator**
  - **Heuristic & Greedy Weight Remapping**
  - **Hardware Optimization: Zero-Skipping Circuit & Pipeline**
- ❑ **Experiment & Results**
- ❑ **Conclusion**

Background: Analog CIM accelerator for NN

❑ Crossbar CIM Pattern

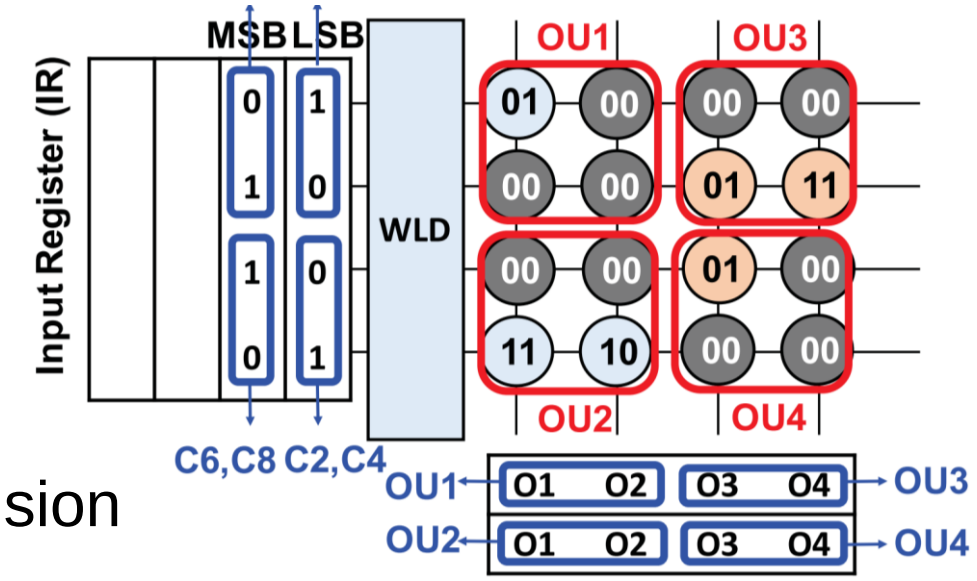
- Map weights to the memory crossbar
  MAC are performed on the bitline
- Drive activations from the wordline
  Inputs are shared by all the columns



[H. Hu, ISCA'2019]

❑ OU-based CIM Pattern

- **Full crossbar CIM is too ideal:**
  - ADC overhead / PVT deviatioin ☹
- **Operating on sub-units (OU):**
  - Practical / Easier for sparsity ☺



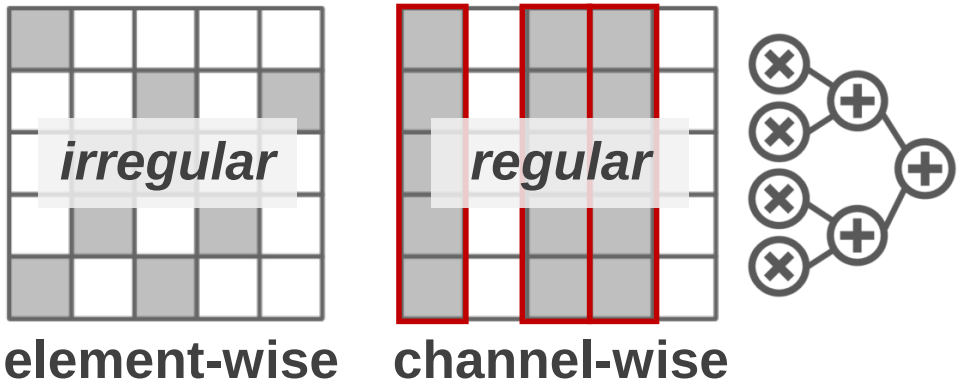
Restricted Column # ➡ Reduce ADC #

Restricted Row # ➡ Relieve ADC precision

Background: NN Sparsity & Pruning

❑ Neural Network Sparsity

- Activation (ReLU) & **Weight (Pruning)**
- Compact storage / Speedup / Efficiency



❑ Irregular Pruning

Our Target

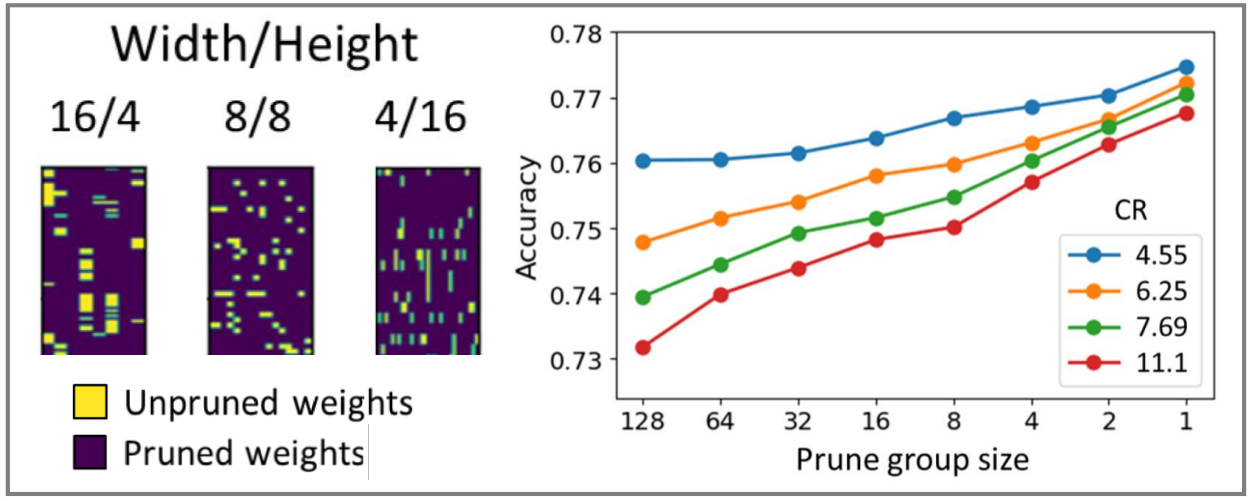
- **Merits: Low accuracy drop / Simplicity** — **Nice for Software**
- **Defects: Low utilization / High index cost** — **Hard for Hardware**

❑ Granularity Exploration

- Prune at fine-grained group

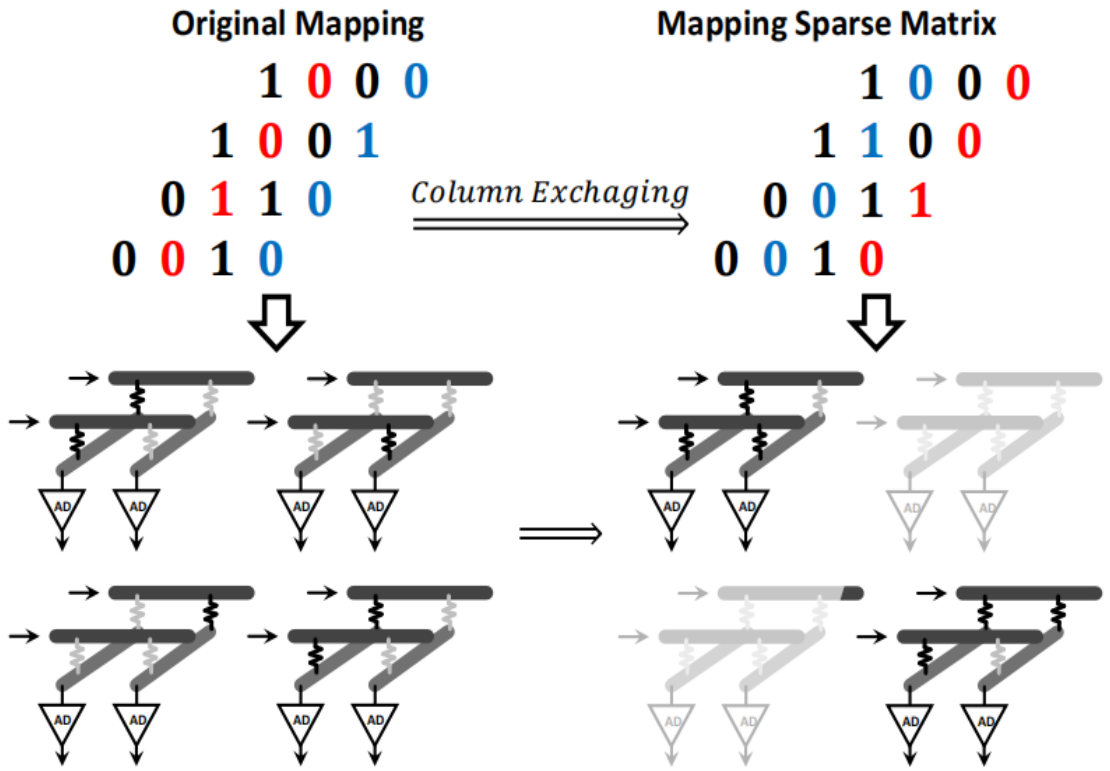
Model Accuracy ⟷ **trade-off** ⟷ Hardware Efficiency

[F. Meng, JETCAS'2022]



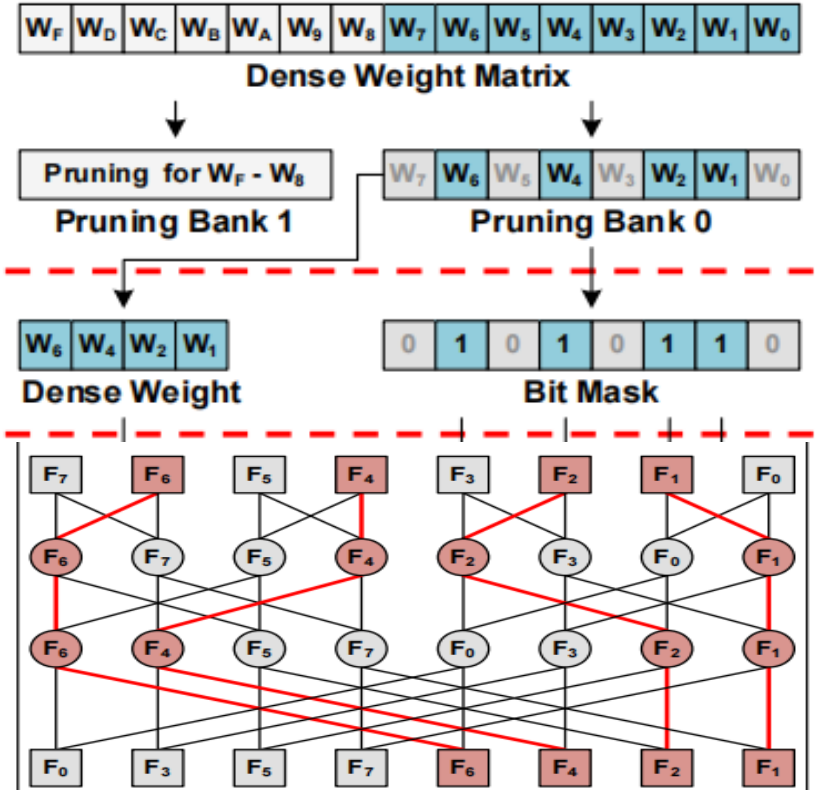
Background: Previous Work

- **Weight mapping based on k-means clustering to skip all-0-crossbars**



[J. Lin, ASPDAC'2019]

- **Weight compression with input-distributing network**



[S. Liu, ISSCC'2023]

Motivation: Irregular Sparsity Utilization

❑ Challenge —

Contradiction between the irregular sparsity & regular CIM

❑ Motivation

- ✓ Both concepts of OU & PG provide opportunites for sparsity utilization

  —— **Joint granularity exploration for the OU & PG**

- ✓ Scattered zero-elements can be aggregated by weight remapping

  —— **Apply heuristic method to efficiently search for the optima**

  70%-94% sparsity utilization (up to 1.6× improvement)

- ✓ Both the weight & actiavtion sparsity can be leveraged for acceleration

  —— **Optimize the hardware to realize pipeline acceleration**

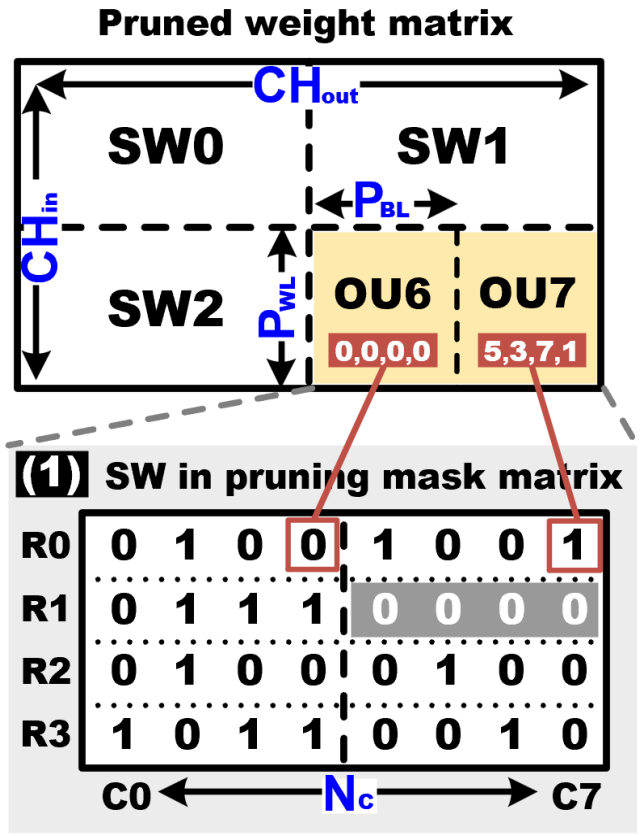
  3-7.6×speedup & 2.1-4.8×energy efficiency

Outline

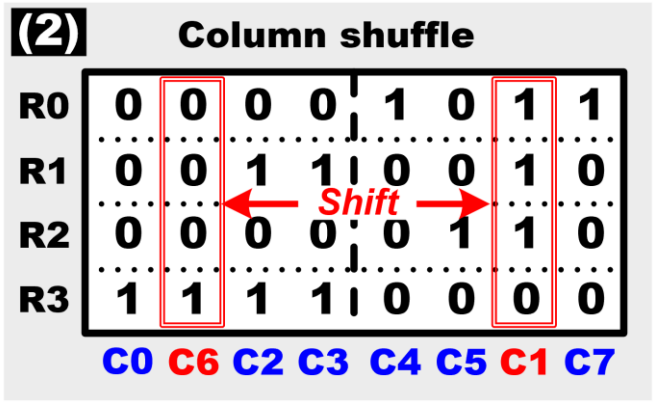
- ❑ Background & Motivation
- ❑ **Proposed Irregular-Sparsity-Aware CIM Accelerator**
  - **Heuristic & Greedy Weight Remapping**
  - Hardware Optimization: Zero-Skipping Circuit & Pipeline
- ❑ Experiment & Results
- ❑ Conclusion

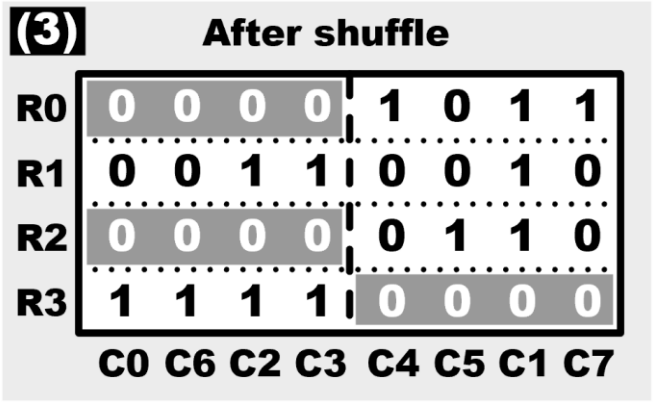
Solution 1: Heuristic & Greedy Weight Remapping

❑ Column shuffle remapping (CSR) for OU-based CIM

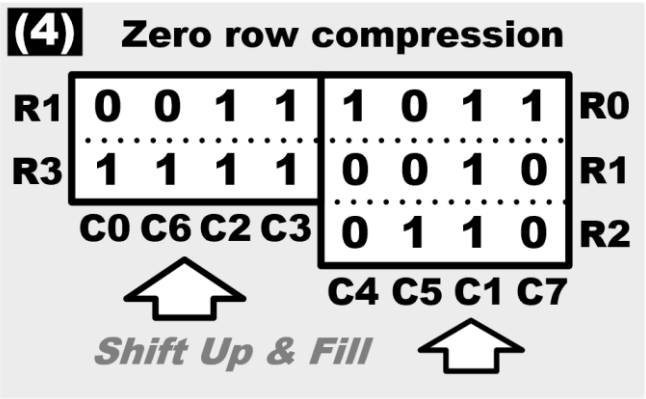


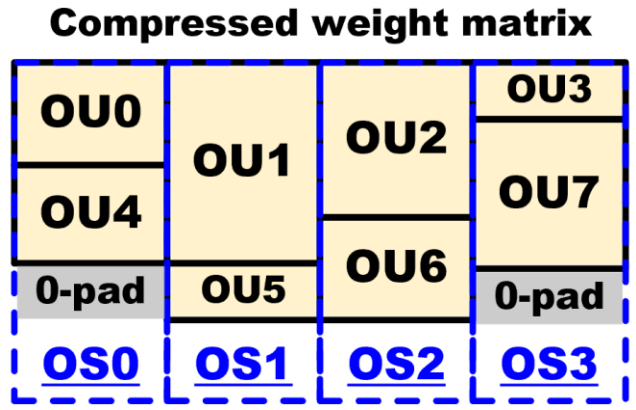
Prune the model and get the mask matrix





Shuffle columns to cluster all-0-rows





Compress OU-wise all-0-rows for compact storage

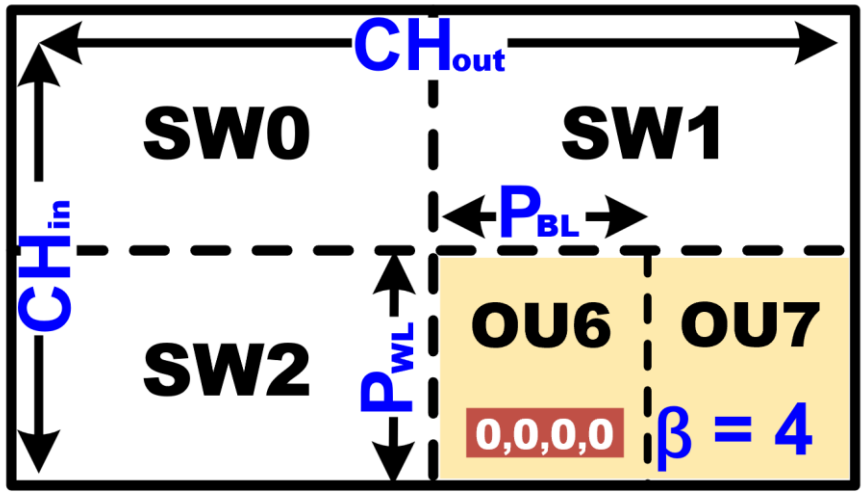
Solution 1: Heuristic & Greedy Weight Remapping

❑ Definition of the parameters in the CSR problem

- ✓ **CH_{in} & CH_{out}:** the shape of the flattened 2D weight matrix
- ✓ **P_{WL} & P_{BL}:** the parallelism of the MVM conducted in one OU
- ✓ **N_C:** The degree of the CSR freedom
- ✓ **Pruning Factor (β):**

  β elements in one row are pruned and shuffled as a block
- ✓ **Shuffle Window (SW):**

  α OUs in one row are grouped as a shuffle window

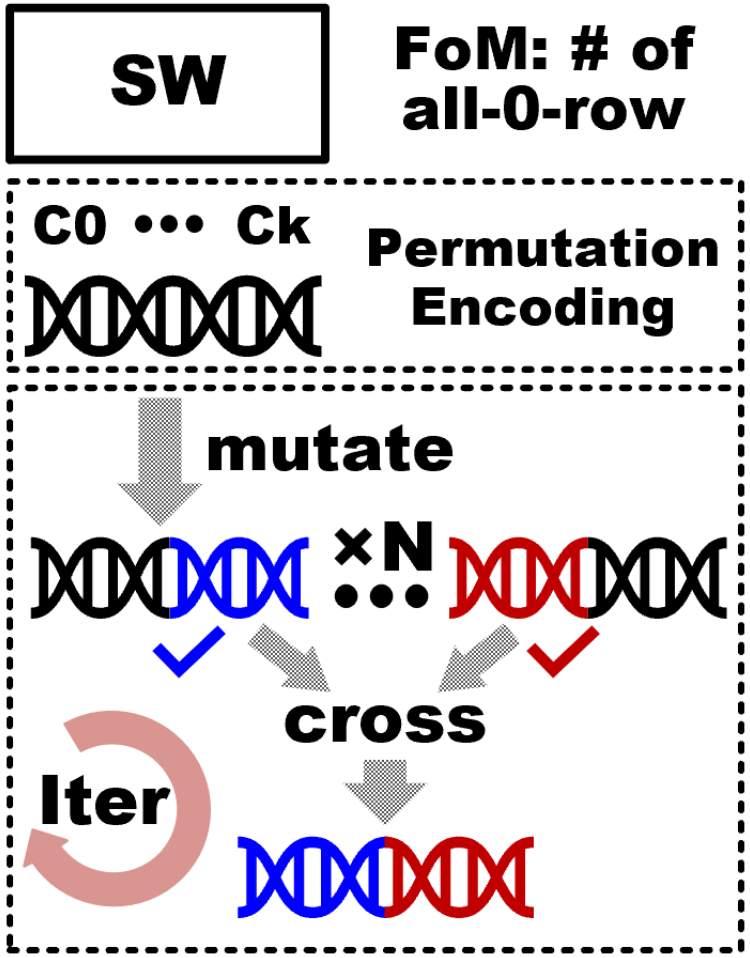


$$N_C = \alpha \cdot P_{BL} / \beta$$

CSR is restricted to SW to relieve the hardware overhead for shuffle $N_C log N_C - N_C/2$

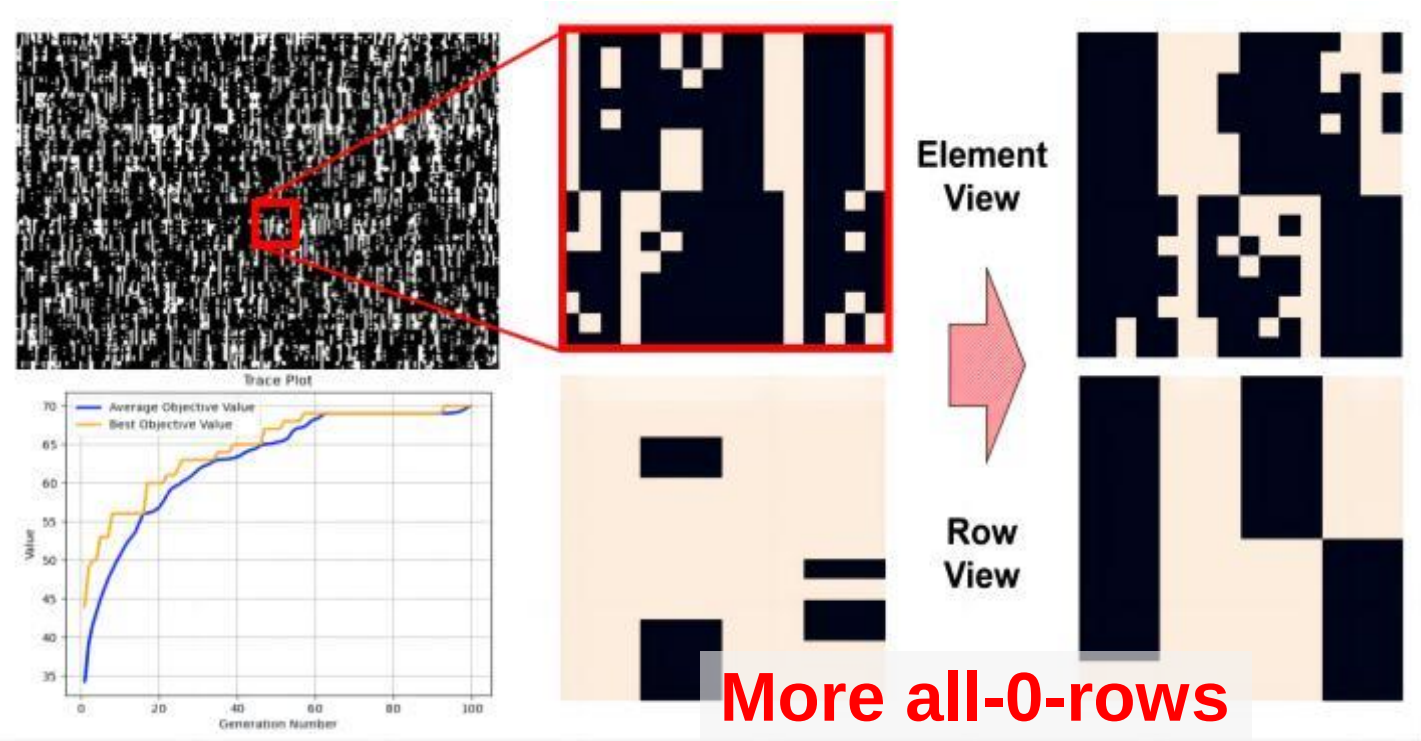
Solution 1: Heuristic & Greedy Weight Remapping

❑ Heuristic CSR exploration based on Genetic Algorithm



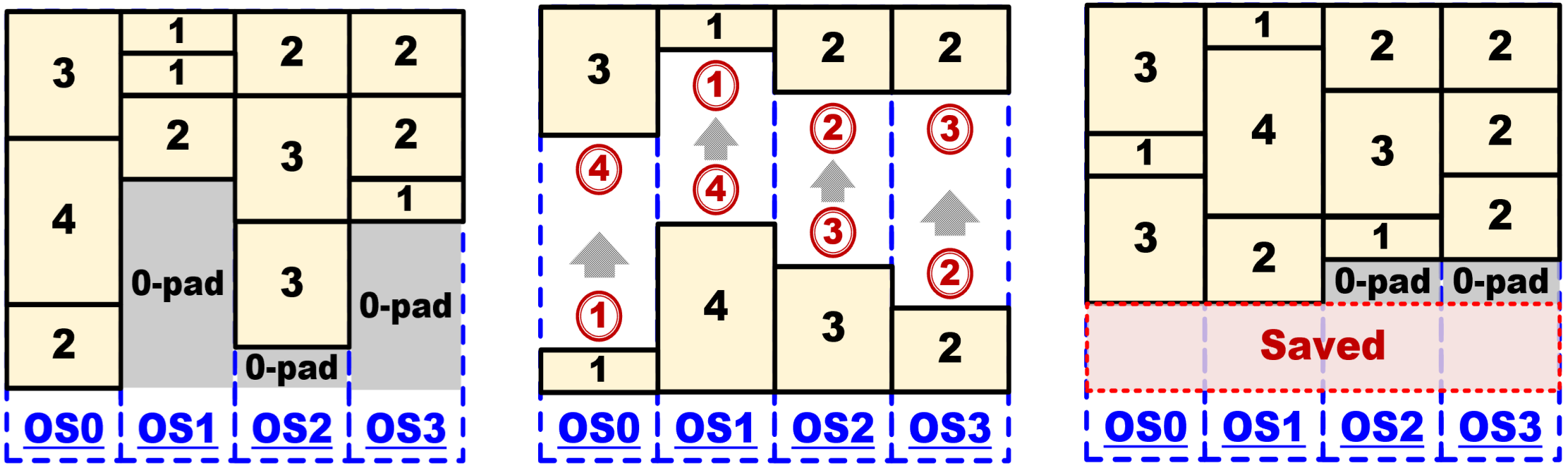
✓ Insensitivity to initial clustering centers

✓ Ability to search for the global optima



Solution 1: Heuristic & Greedy Weight Remapping

- **Storage compression based on Greedy Algorithm**
  - ✓ The maximum deviation between OUs is less than P_{WL}
  - ✓ Column shuffle can be merged with the heuristic process



Outline

- ❑ Background & Motivation
- ❑ **Proposed Irregular-Sparsity-Aware CIM Accelerator**
  - ▪ Heuristic & Greedy Weight Remapping
  - ▪ **Hardware Optimization: Zero-Skipping Circuit & Pipeline**
- ❑ Experiment & Results
- ❑ Conclusion

Solution 2: Sparsity Aware Circuit Design

❑ Zero-Skipping Circuit

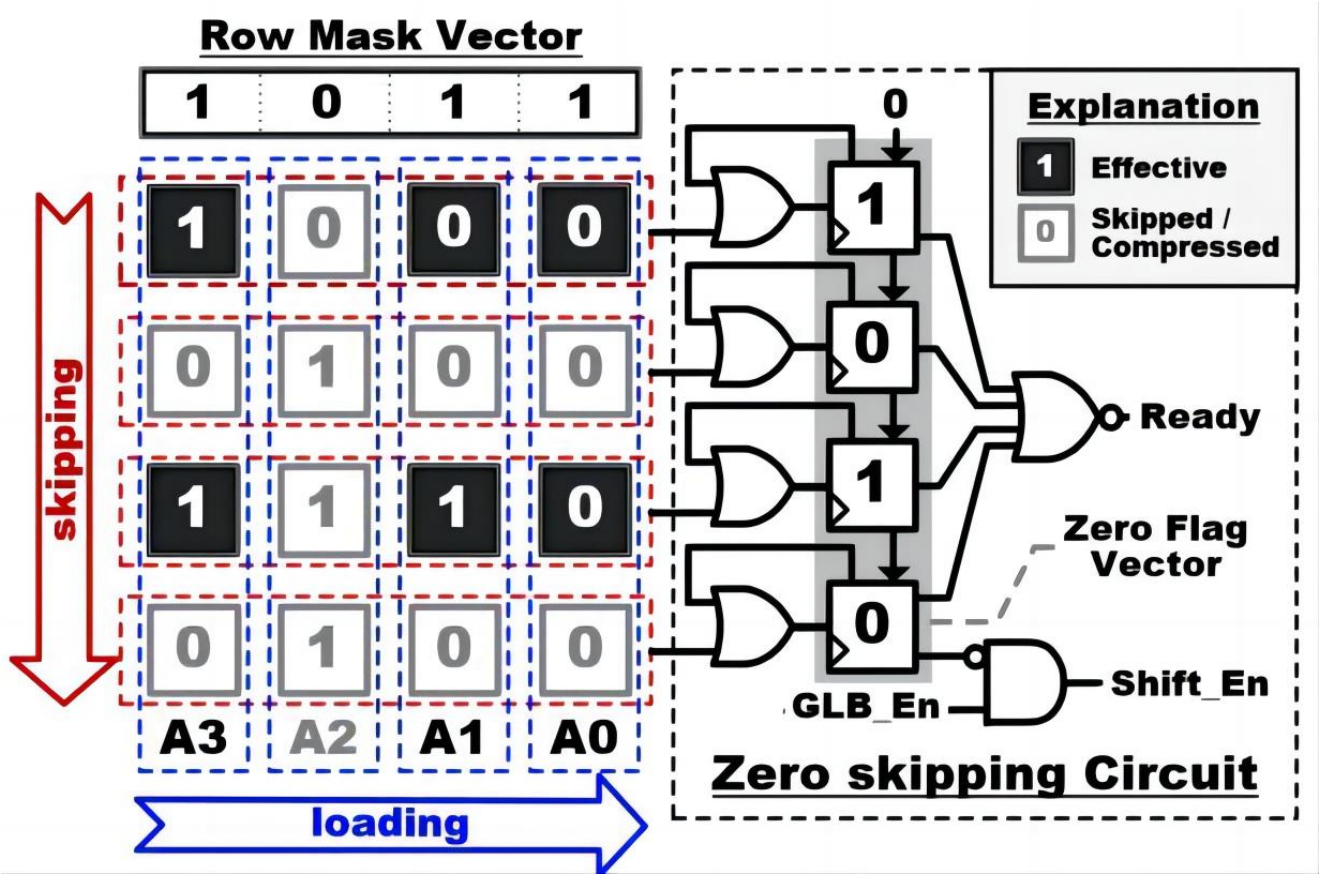
Joint the activations' bit-wise sparsity with weights' row-wise sparsity for acceleration

Loading (flattened act.):

- ✓ **Discard the activations** corresponding to the compressed rows.
- ✓ **Record the zero flag** for each activation bit.

Skipping (per act. bit):

- ✓ **Skip all-0-bits** according to the zero flag.
- ✓ **Fire the effective bits** to the word-line driver.



Solution 2: Sparsity Aware Circuit Design

❑ The overall architecture and dataflow

Cache (per OU row):

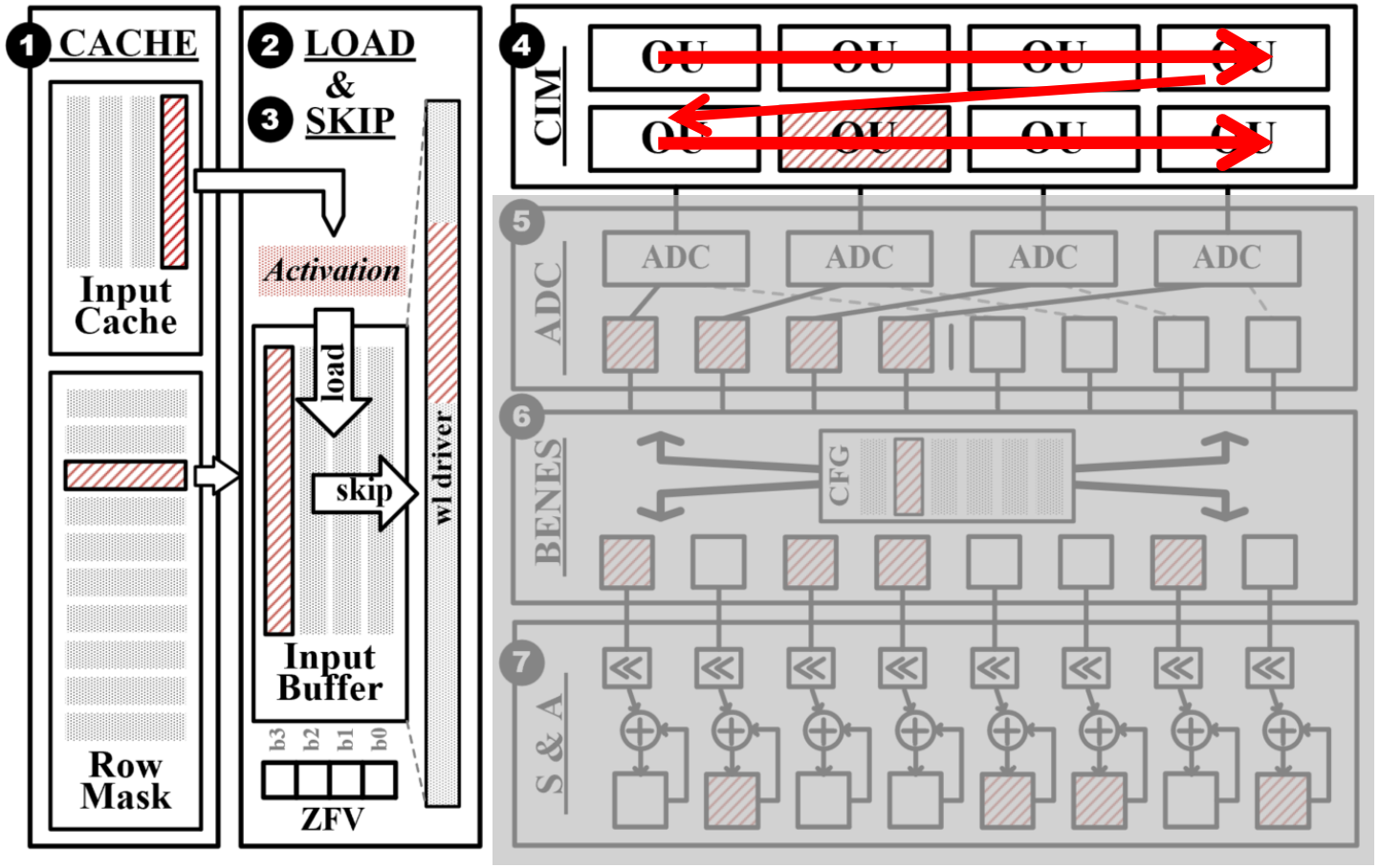
✓ Reuse the act. for the OUs in the same row

Load & Skip (per OU):

✓ Drive the effective bits of the act. to the OU

CIM (per OU):

✓ Traverse the OUs in the row-first order



Solution 2: Sparsity Aware Circuit Design

❑ The overall architecture and dataflow

ADC (per OU):

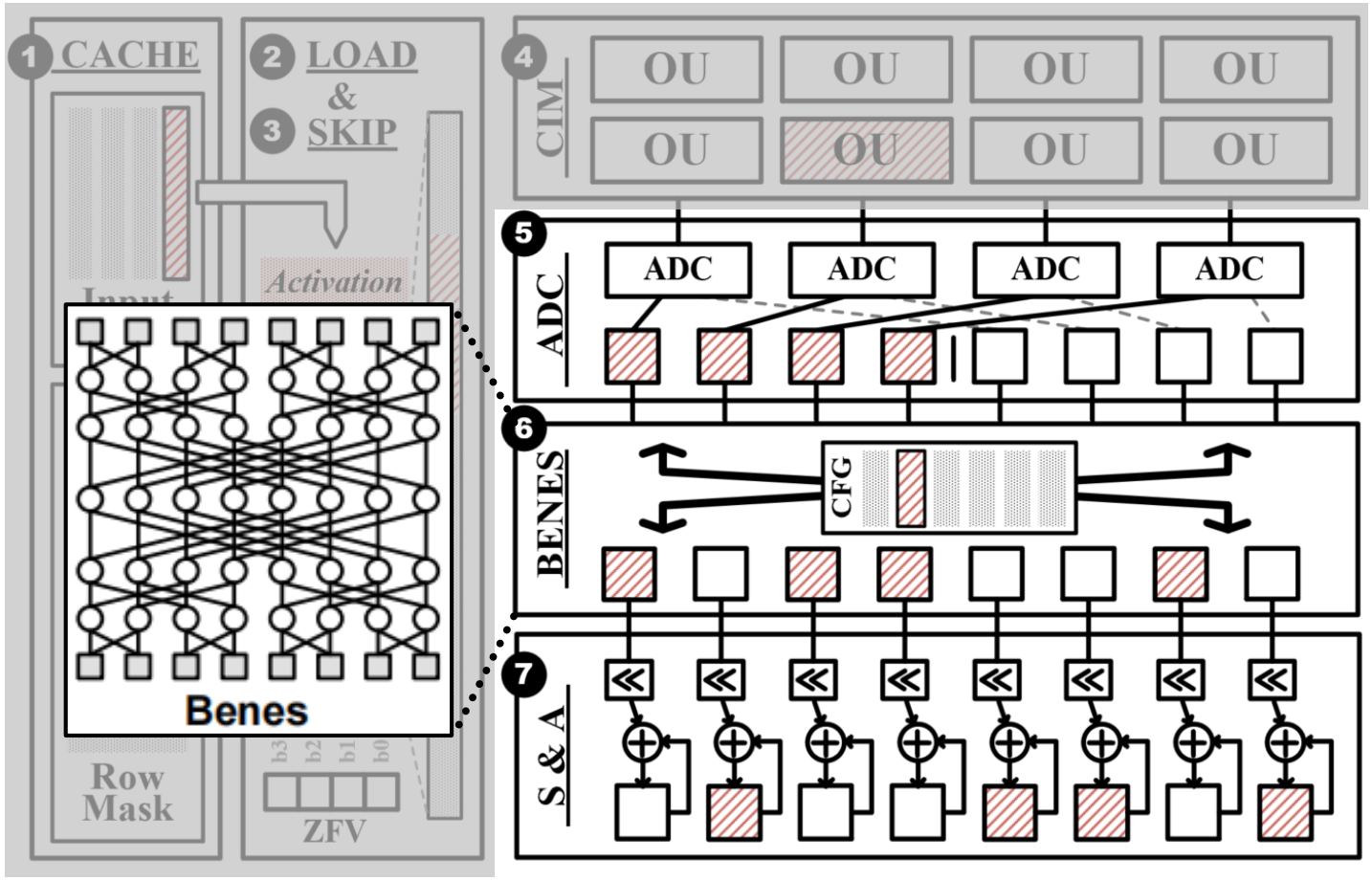
✓ Accumulate the bitwise multiplication results

BENES (per SW):

✓ Reorder the results to the normal sequence

S&A (per SW):

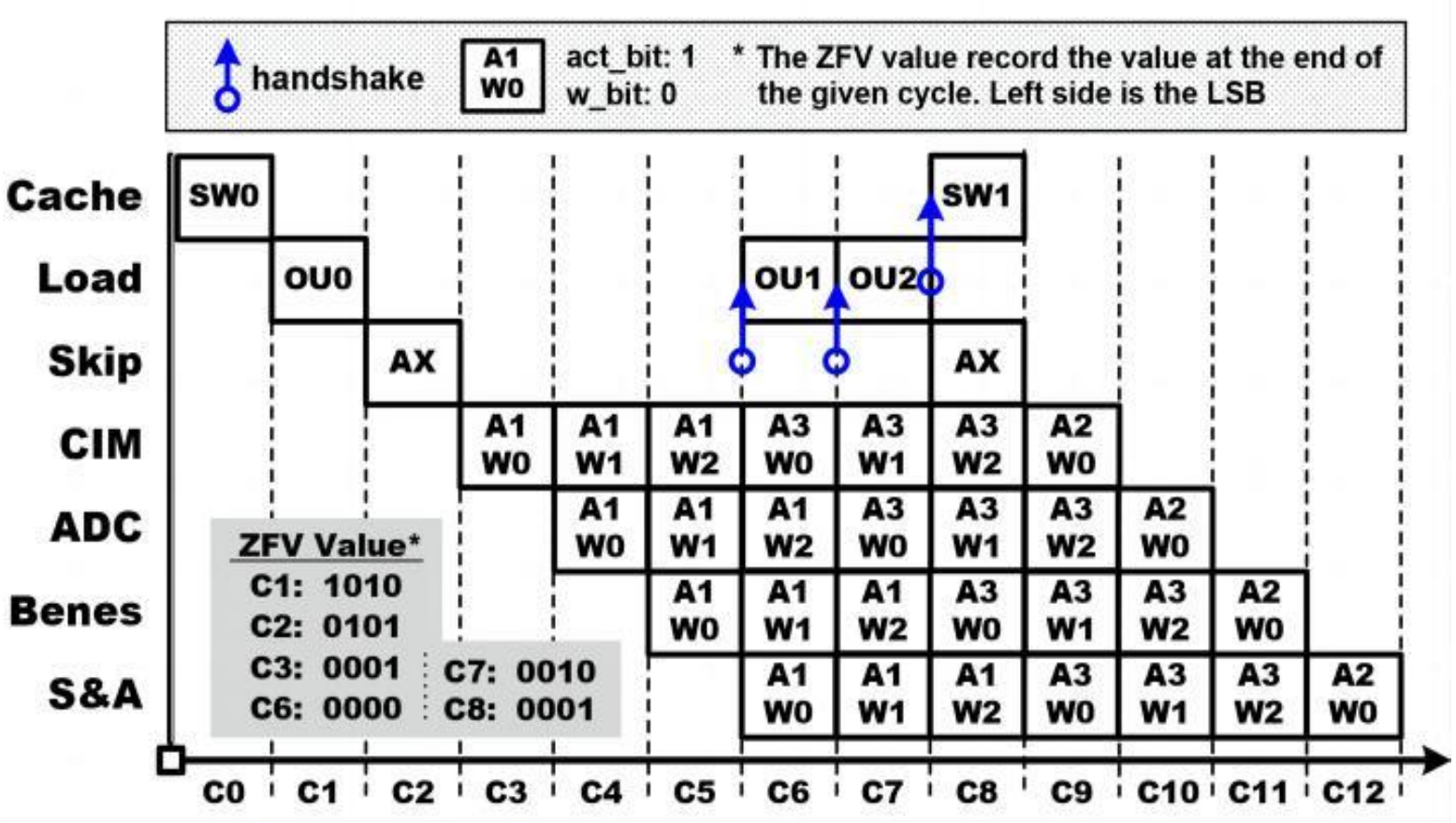
✓ Shift & add for weighted psum accumulation



Solution 2: Sparsity Aware Circuit Design

❑ The timing diagram of the zero-skipped pipeline

- ✓ The bit-serial pattern benefits directly from the proposed ZSC
- ✓ The Cache & Load stages are emitted by the handshake signal
- ✓ The skip of the all-0-OU can be hidden by the bit-serial process



Outline

- ❑ Background & Motivation
- ❑ Proposed Irregular-Sparsity-Aware CIM Accelerator
  - Heuristic & Greedy Weight Remapping
  - Hardware Optimization: Zero-Skipping Circuit & Pipeline
- ❑ **Experiment & Results**
- ❑ Conclusion

Experiment: Case Study on Granularity

- **Index Footprint: R_{cost} = $Index_{bits}$ / $Weight_{bits}$**

$$R_{cost} = \frac{1}{B_w \beta}\left[(\log_2 N_c - \frac{1}{2})\frac{1}{P_{wl}} + \frac{\beta}{P_{bl}}\right]$$

sharing coefficient

shuffle term

compression term

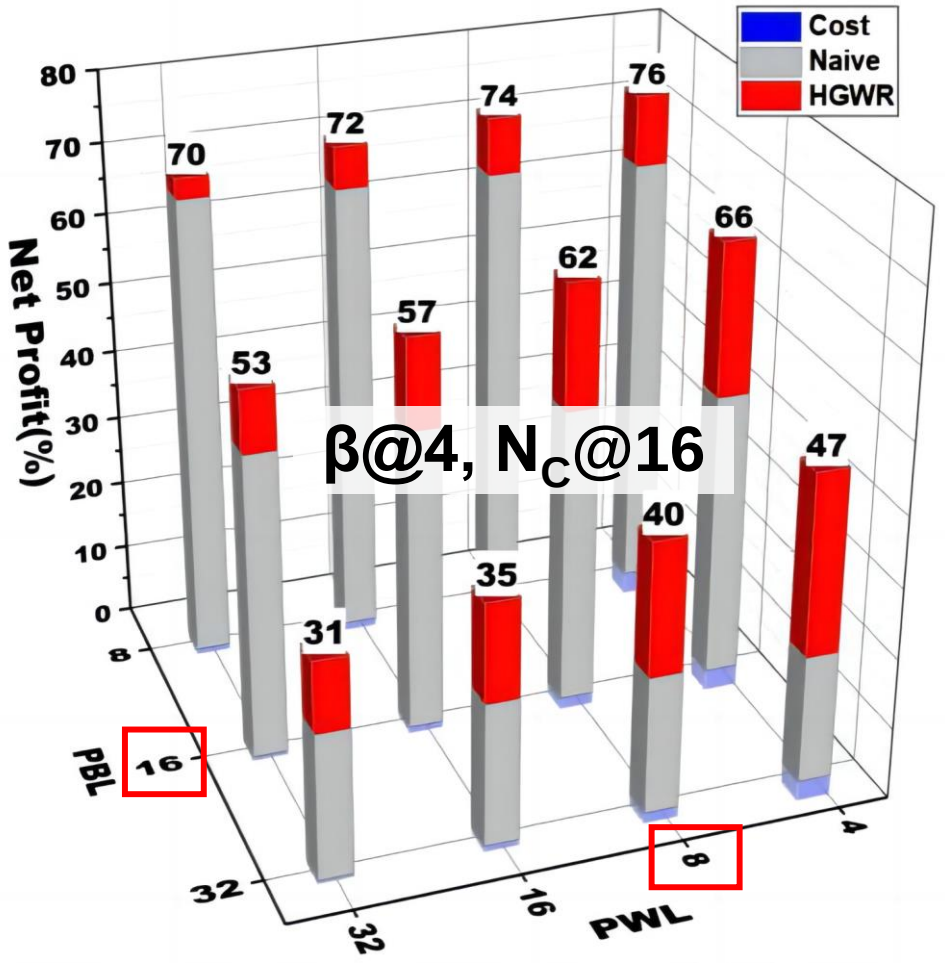
Larger P_{WL} & P_{BL} bring more index reuse —— **less footprint**

trade off

- **Compression Rate**

Smaller P_{WL} & P_{BL} put less constraints —— **higher compression rate**

81% Sparsed ResNet50



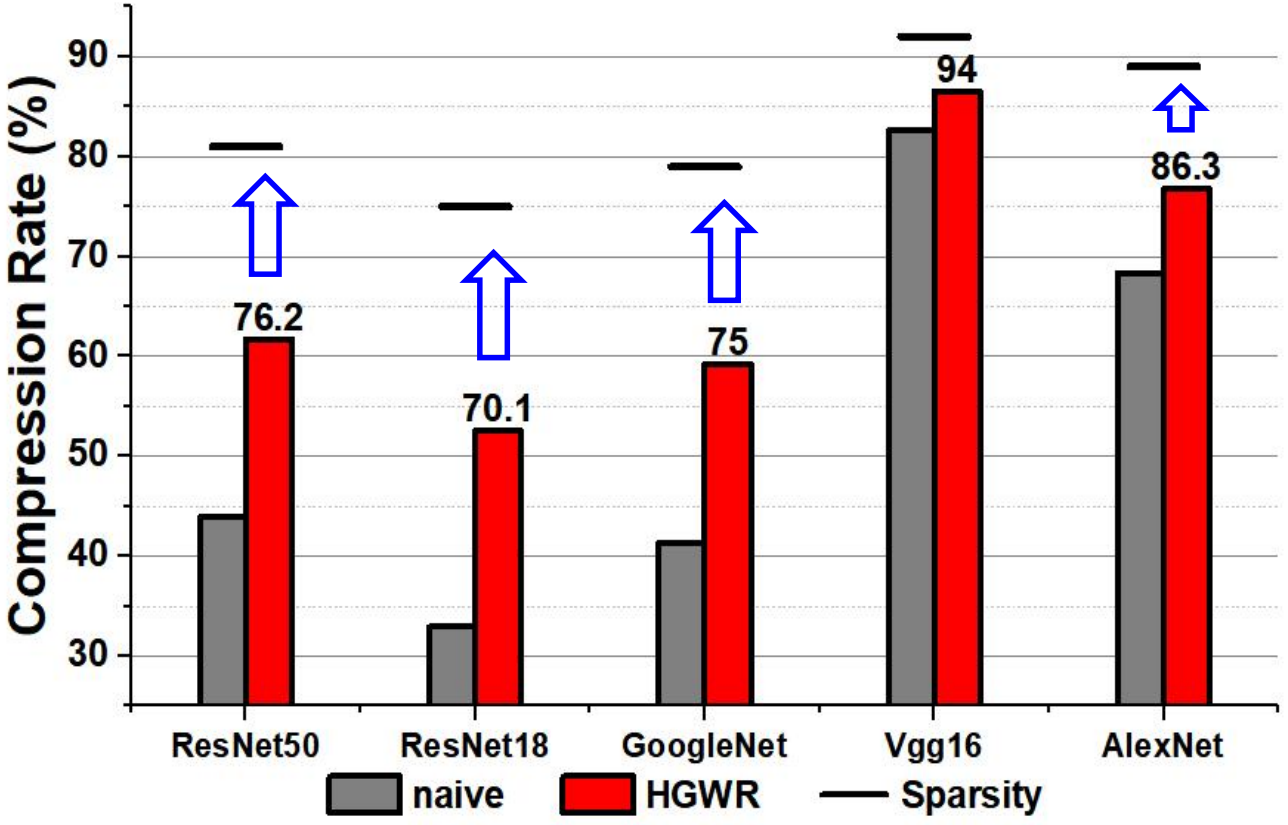
Experiment: Setup

- **Hardware Configuration**
  - Memory: 128×128 RRAM crossbar
  - OU: 16×8 sub-crossbar with 2-bits cell percision
  - Peripheral: 5-bits ADC×16 / 16to16 Benes Network
- **Pruning & Quantization**
  - Prune: Magnitude method based on L2 norm (β=4)
  - Quant: LSQ method — W@int8, A@uint8
- **Behavior-Level Simulator**
  - Record the per-layer activations during inference (on ImageNet1K)
  - Model the timing of the zero-skipped pipeline and the energy of the macro
  - Analyze the performance according to the configuration and runtime trace

Experiment: Compression Rate

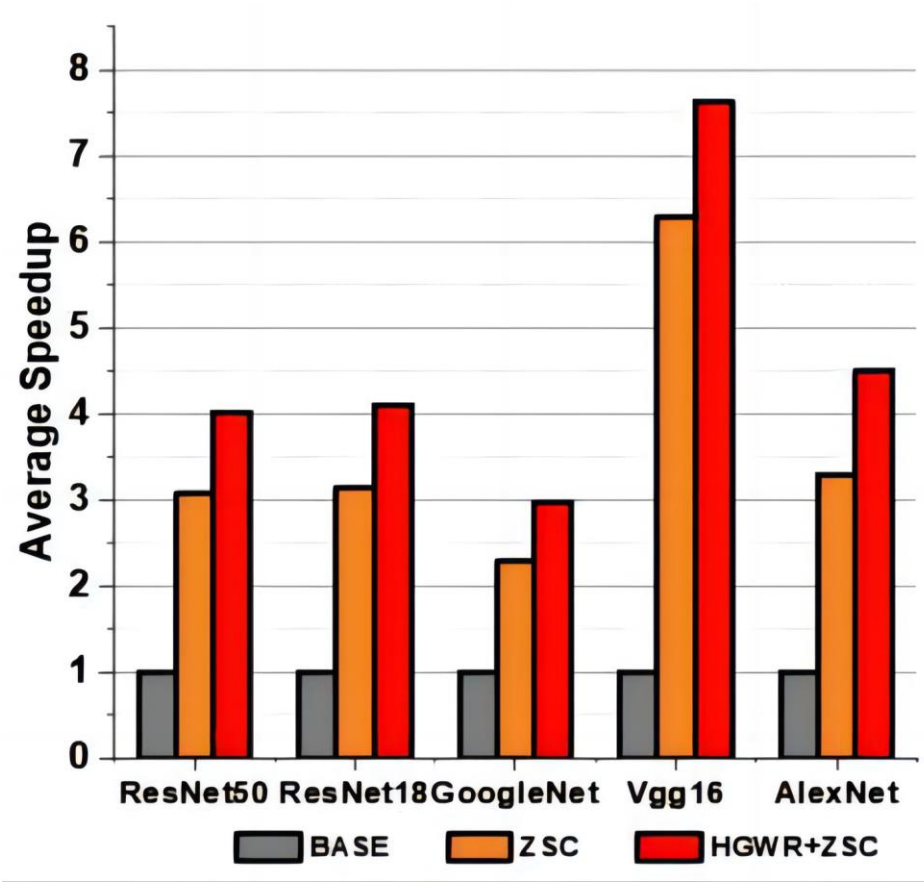
- 70%-94% sparsity utilization — up to 1.6× improv.

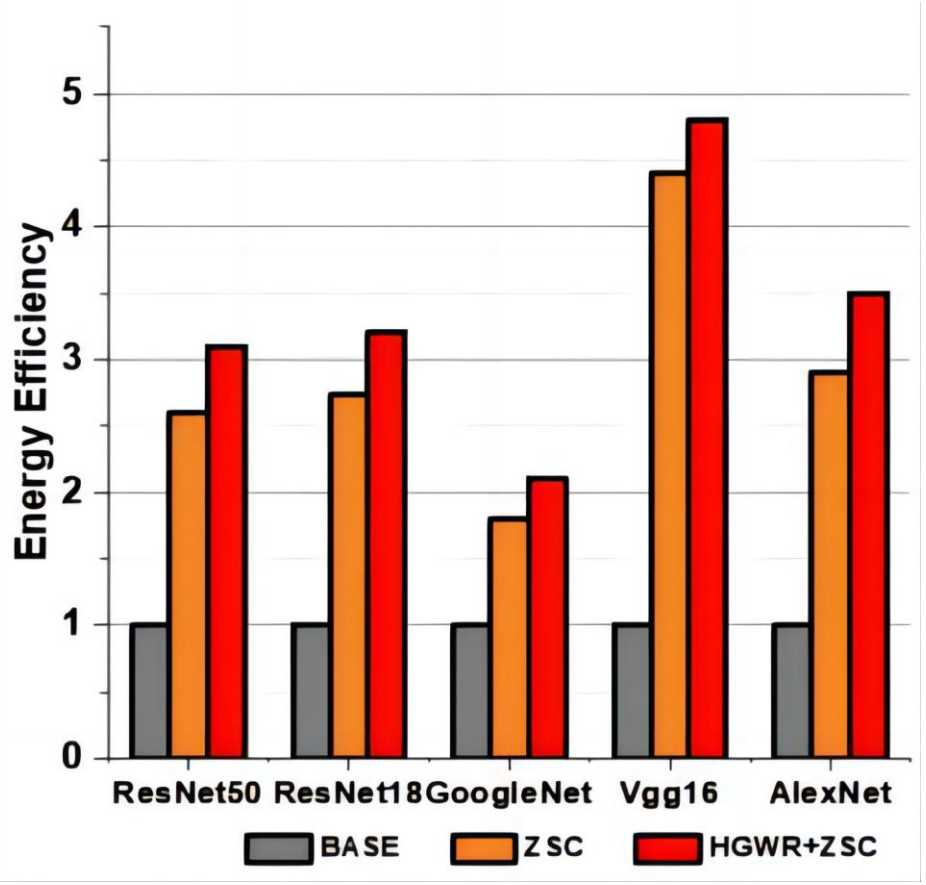
NN Model	Prune Ratio
ResNet18	75%
ResNet50	81%
GoogleNet	79%
Vgg16	92%
AlexNet	89%



Experiment: Speedup & Energy Efficiency

- 3-7.6×speedup & 2.1-4.8×energy efficiency





Outline

- ❑ Background & Motivation
- ❑ Proposed Irregular-Sparsity-Aware CIM Accelerator
  - Heuristic & Greedy Weight Remapping
  - Hardware Optimization: Zero-Skipping Circuit & Pipeline
- ❑ Experiment & Results
- ❑ **Conclusion**

Conclusion

❑ The Co-optimzied Irregular-Sparsity-Aware CIM Accelerator

- **Heuristic & Greedy Weight Remapping**

  ✓ Exploit the irregular weight sparsity for the regular CIM

- **Zero-Skipping Circuit & Pipeline**

  ✓ Utilize the exploited sparsity for speedup & energy saving

- **Joint exploration of the OU & prunning granularity**

  ✓ Strike a balance between the compression rate, index overhead and degree of parallelism

Thank You !