Challenges and Opportunities for Computing-in-Memory Chips

Xiang Qiu

School of Integrated Circuits,

East China Normal University

& Shanghai Flash Billion Semiconductor Inc.

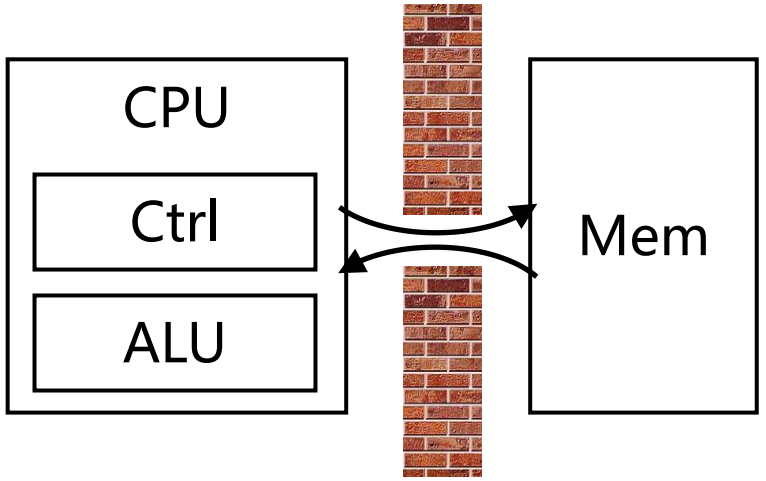
ECNU

Outline

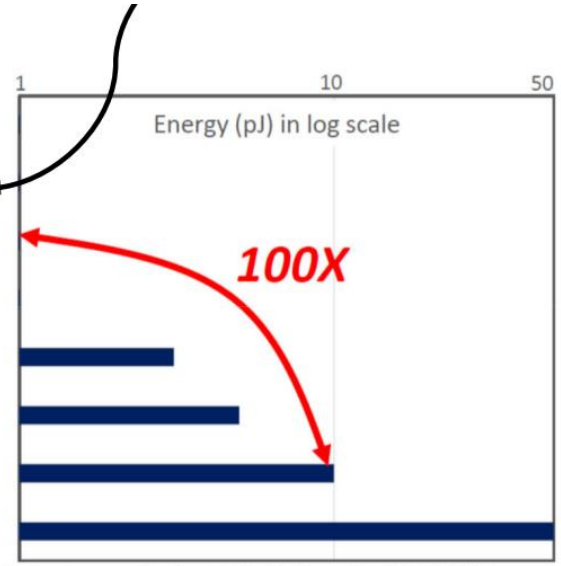
- Computing-In-Memory Overview
- A NOR-Flash based CIM chip
- Analog Computing Accuracy Challenge
- Neural Network Model Deployment on CIM chips
- Conclusions

Why Computing-In-Memory?

- Von Neumann architecture hitting the power wall
  - Data access energy far greater than computation
  - Neural network computation is data-centric



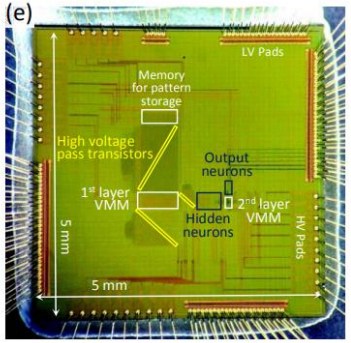
Operation	Energy (pJ)
Integer ADD (8b)	0.03
Integer ADD (16b)	0.05
Integer ADD (32b)	0.1
Integer MULT (8b)	0.2
Integer MULT (32b)	3.1
8KB SRAM Read (32b)	5
32KB SRAM Read (32b)	10
1MB SRAM Read (32b)	50



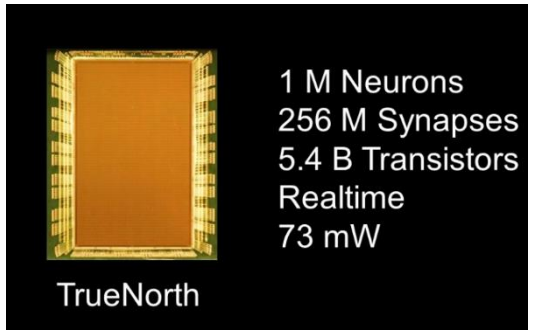
"Computing's Energy Problem (and what we can do about it)", M. Horowitz, ISSCC 2014

How to do Computing-in-Memory?

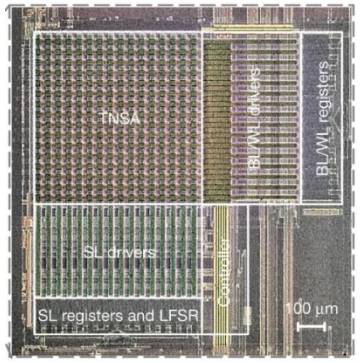
- Advanced memory technology + new architectures



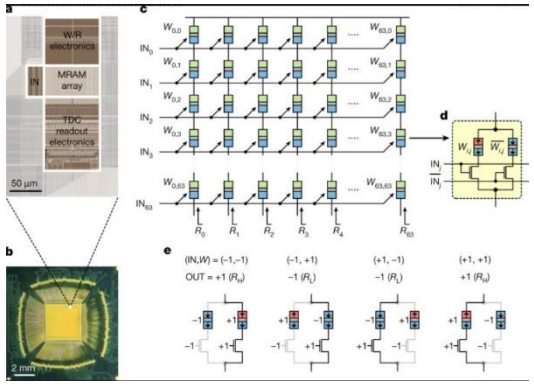
NOR-Flash based MAC Unit, [X. Guo et al, IEDM, 2017]



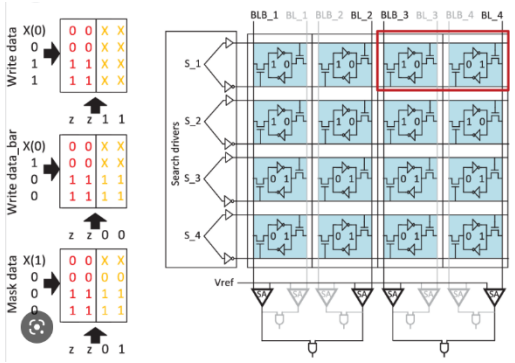
PCRAM based SNN chip [IBM, 2014]



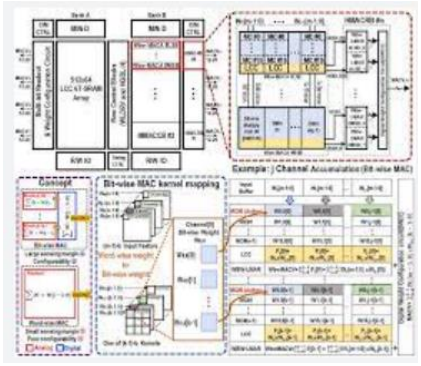
RRAM-based NeuRRAM [W. Wan, et al, Nature, 2022]



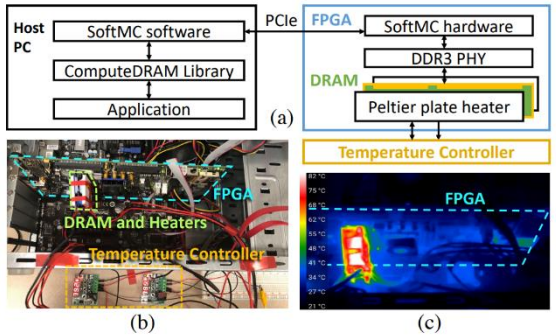
MRAM-based MAC [J. Sung, et al, Nature, 2022]



SRAM-based logic [S. Jeloka, et al, JSSC, 2016]



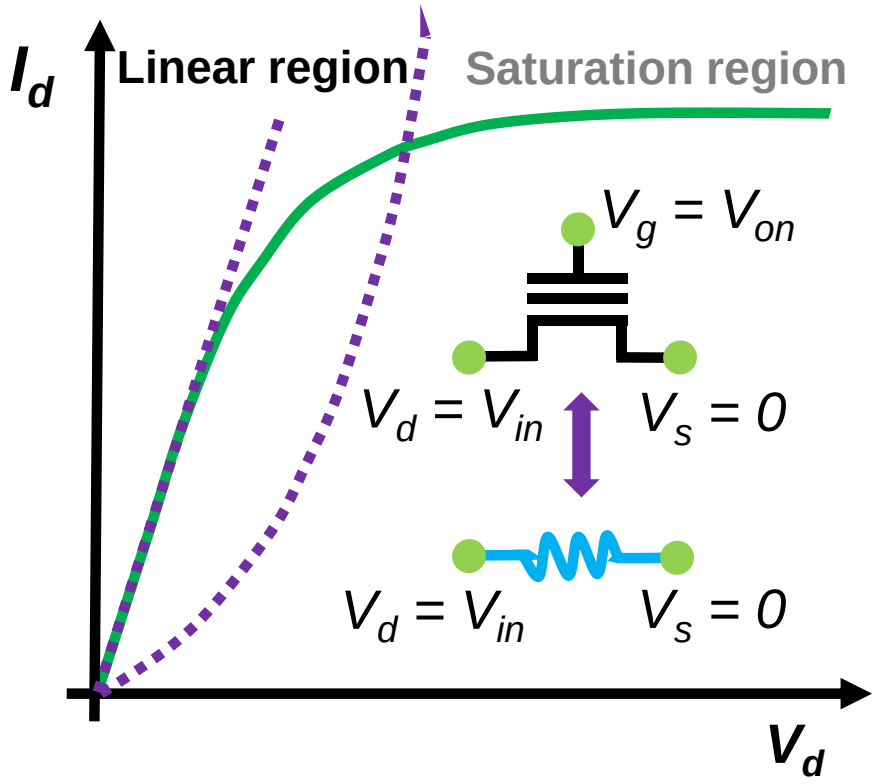
SRAM-based MAC [X. Shi et al, ISSCC, 2020]

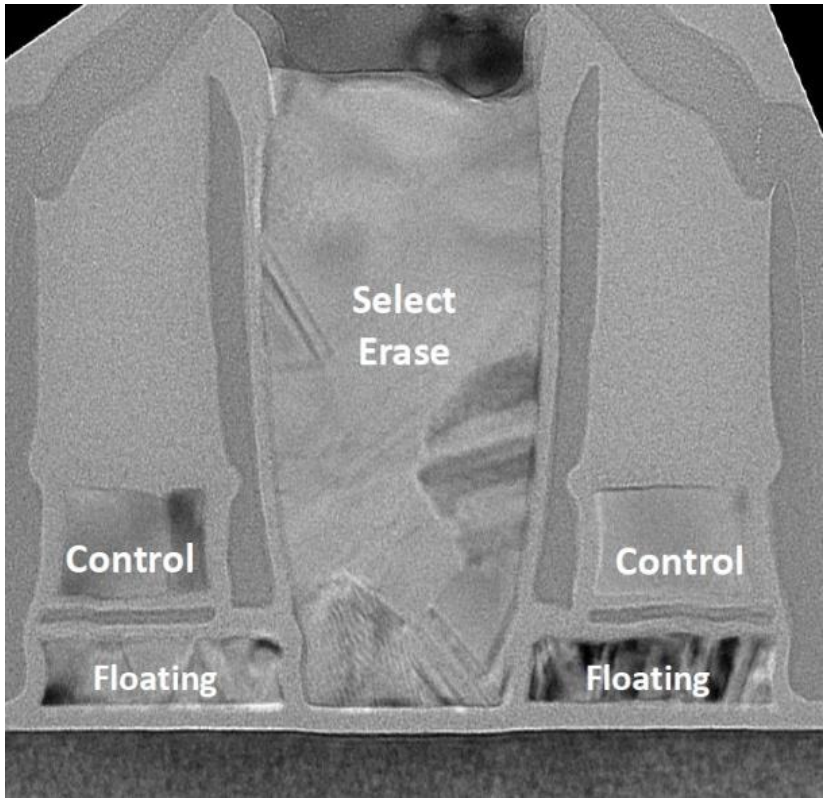


DRAM-based logic [F. Gao et al, MICRO, 2018]

A Nor-Flash based CIM chip

- NOR-Flash array as multiply–accumulate (MAC) engine
  - Each cell stores 8-bit weight

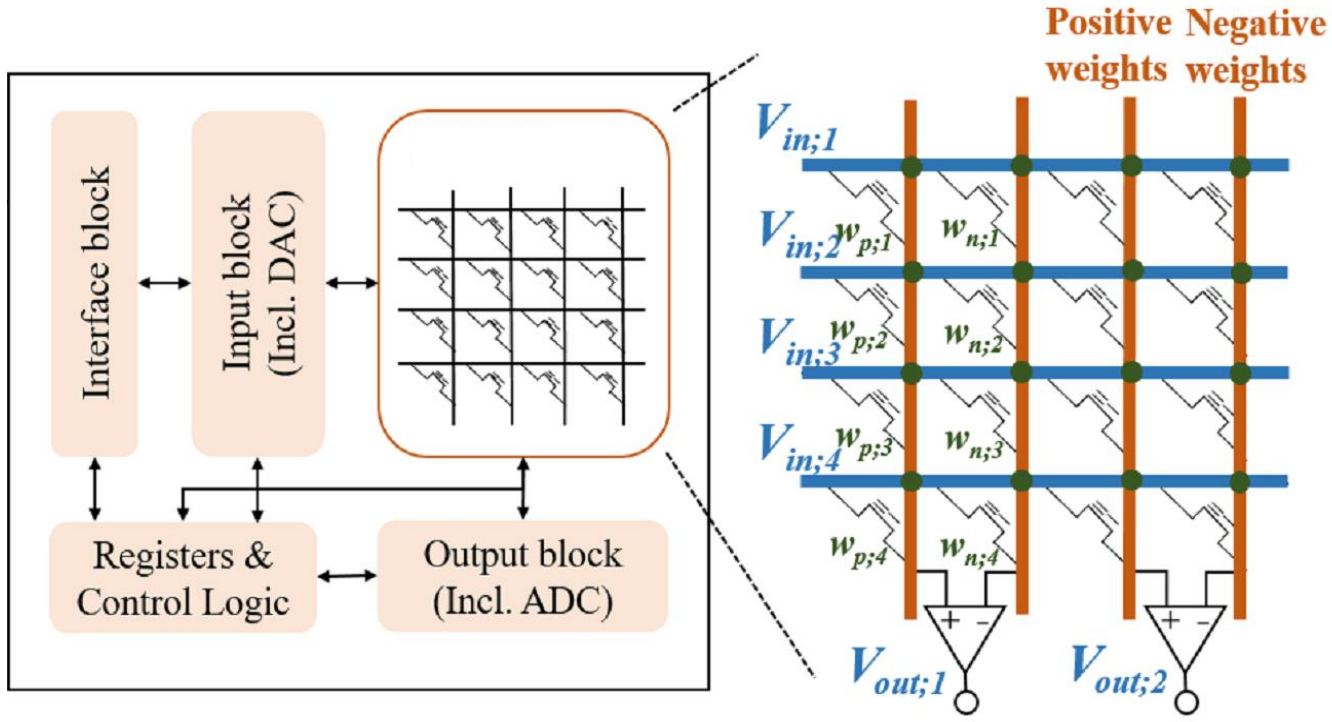




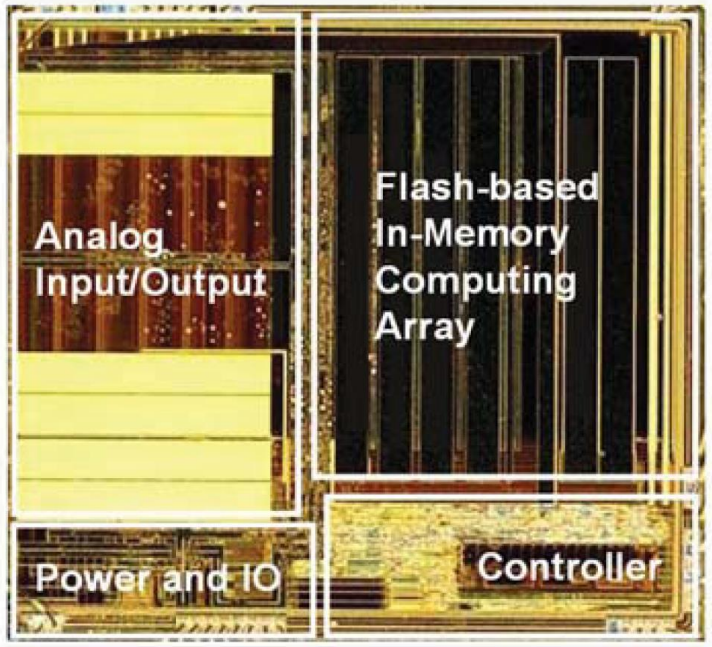
S. Gao *et al.*, *IEEE IEDM*, 2019

A Nor-Flash based CIM chip

- Parallel MAC based on Kirchhoff's law



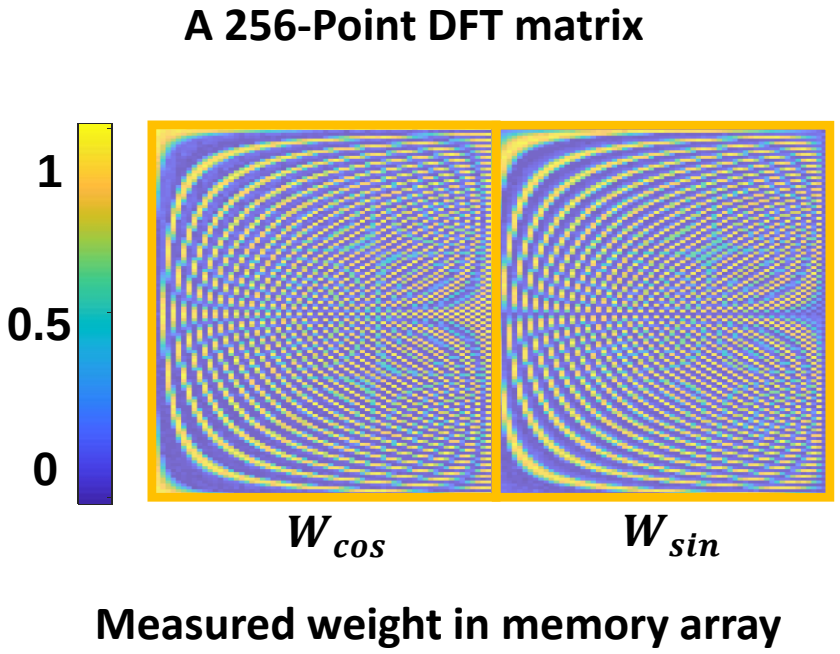
$$\sum(Vin_i - Vout)*G_i=0 \Rightarrow Vout = \frac{\sum Vin_i*G_i}{\sum G_i}$$

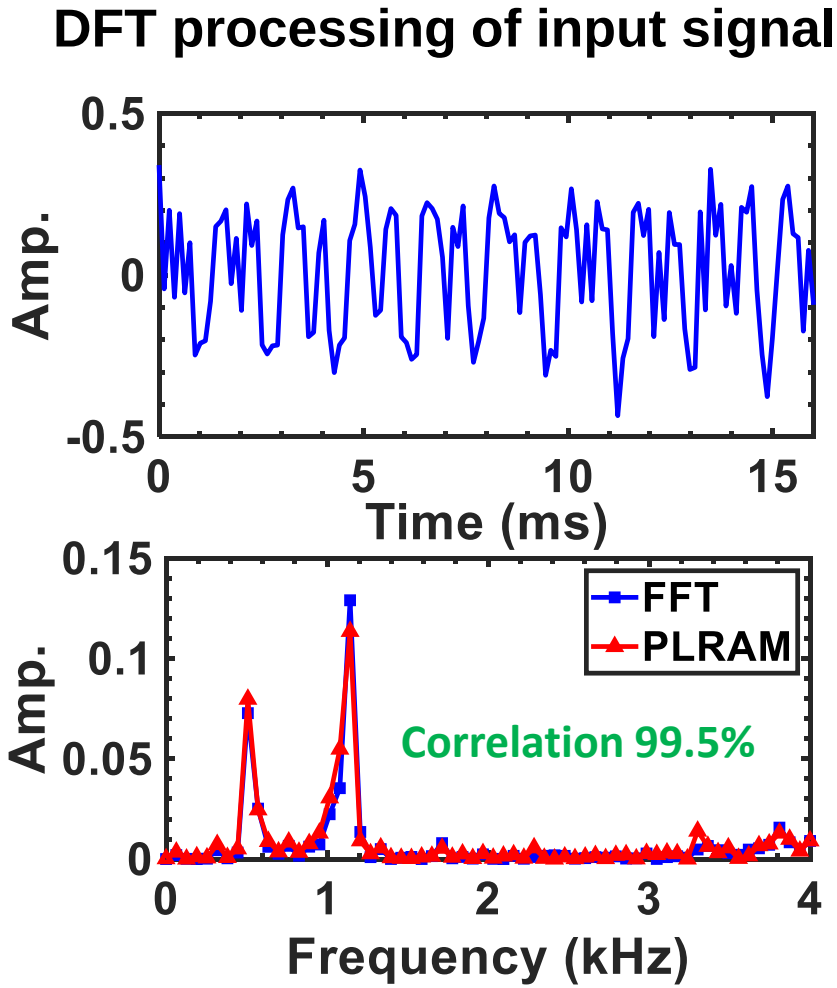


- 90 nm node embedded Flash
- Weight capacity: 6 MB
- Peak speed: 30 GOPS
- Power: 2mW (full chip)
- Used for voice recognition
- In volume production
- Shipped 1M chips

Accuracy of MAC operation

- Analog computing suffers noises

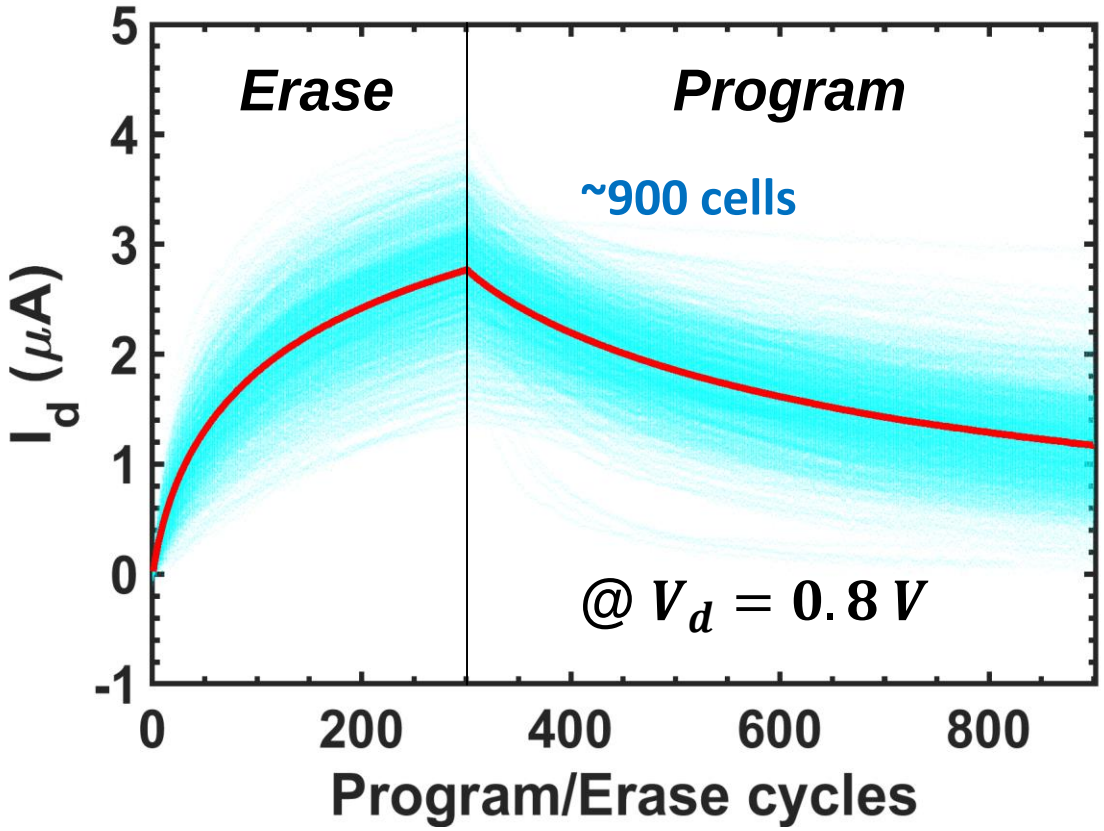


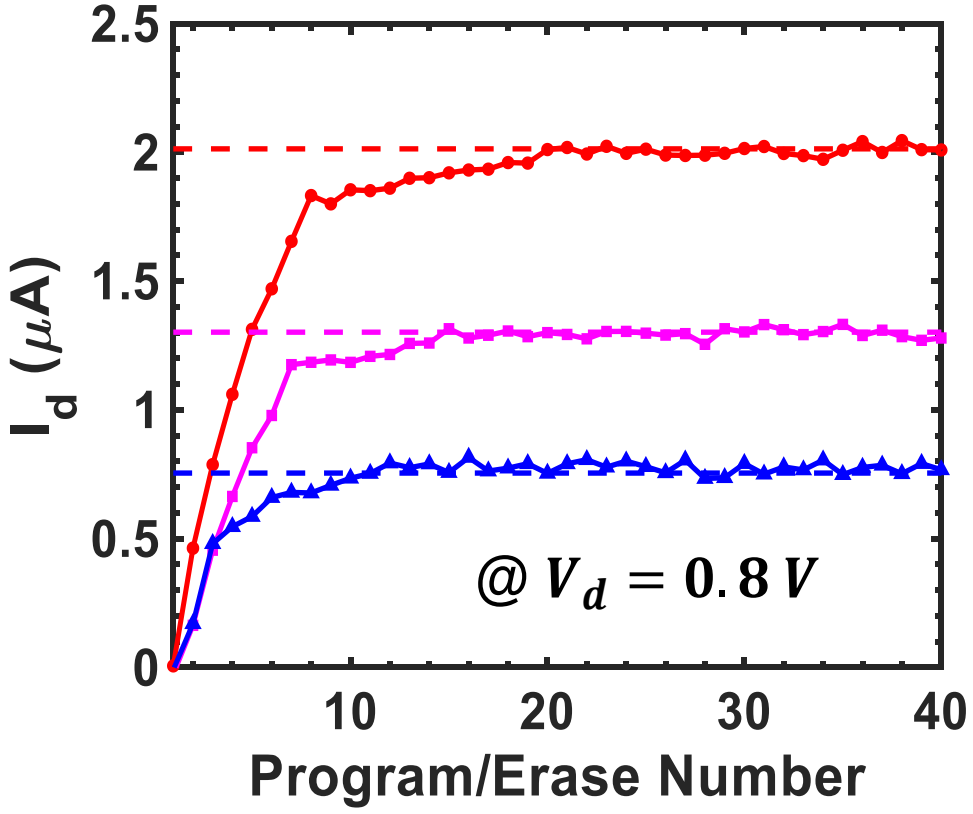


S. Gao *et al.*, *IEEE IEDM*, 2019

Memory Array Weight Program Error

- Trade-off between accuracy and programming time(cost)

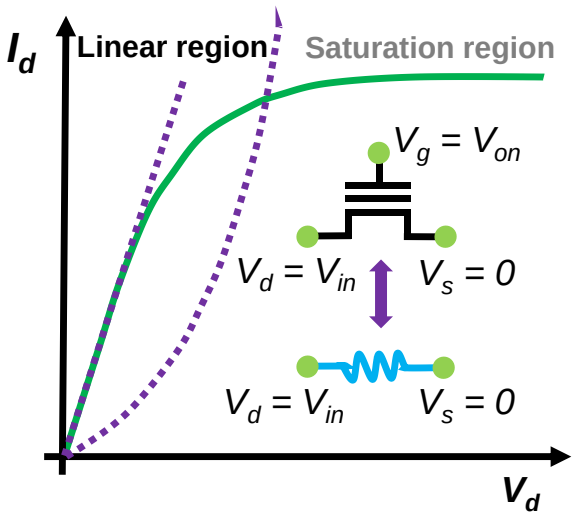


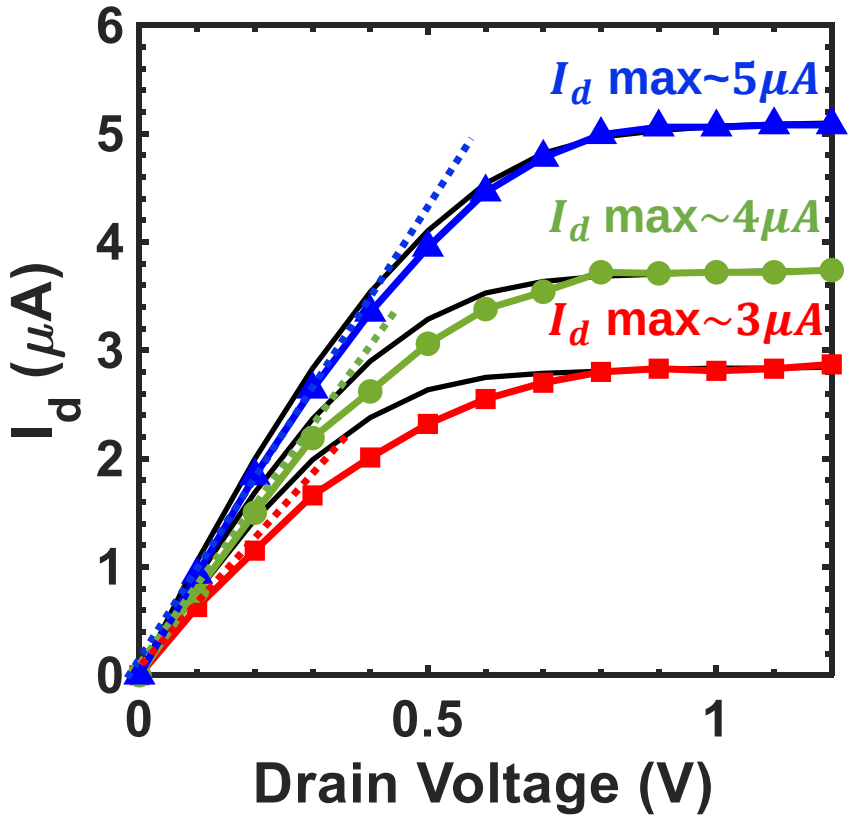


S. Gao *et al.*, *IEEE IEDM*, 2019

Memory cell non-linearity

- Memory cell non-linearity

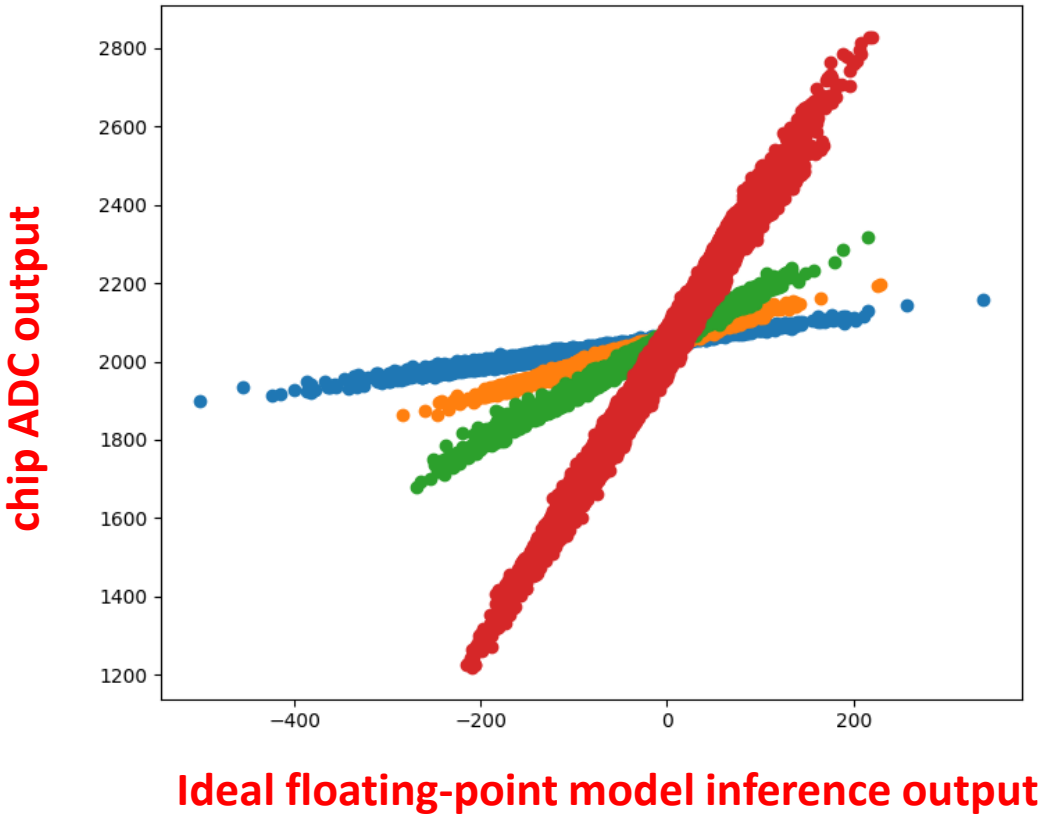




When large input meets small weight

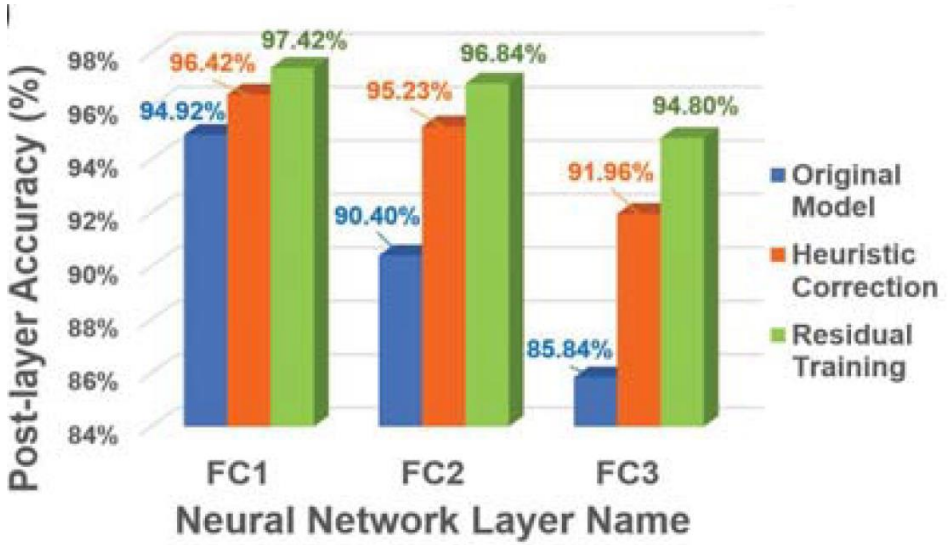
Other errors

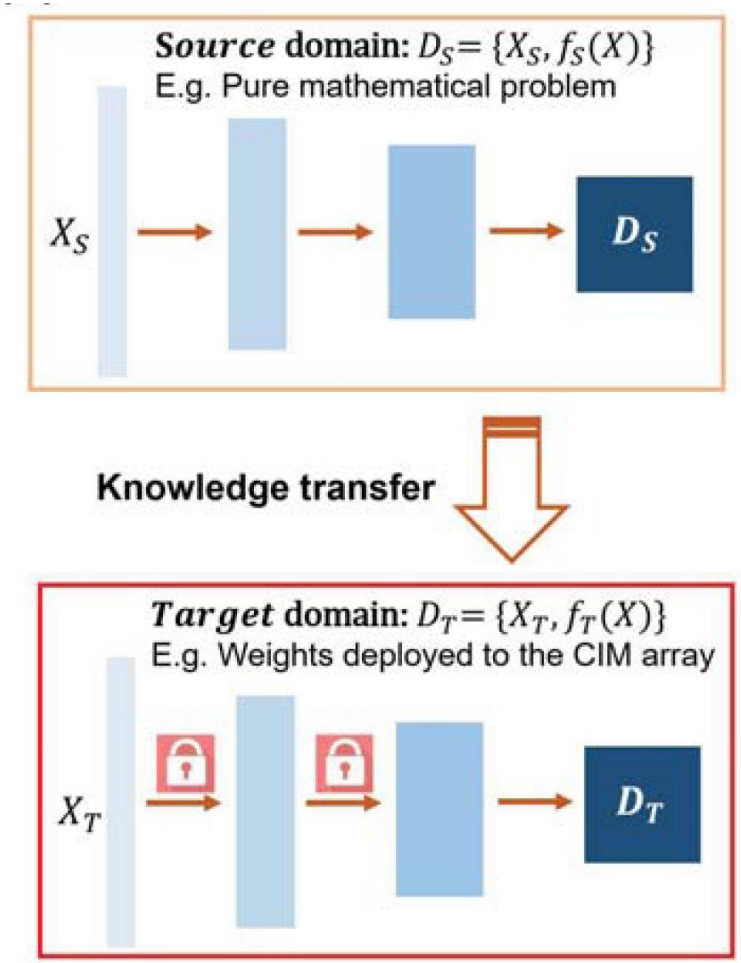
- Input DAC non-linearity
- Output ADC non-linearity + mismatch
- Interconnect(BL/SL) IR-drop
- Model quantization loss
- Temperature drift
- Process variation
- Layer by layer error accumulation
- …



Opportunities to conquer analog computing errors

- Transfer learning
- Noise aware neural network training
- …

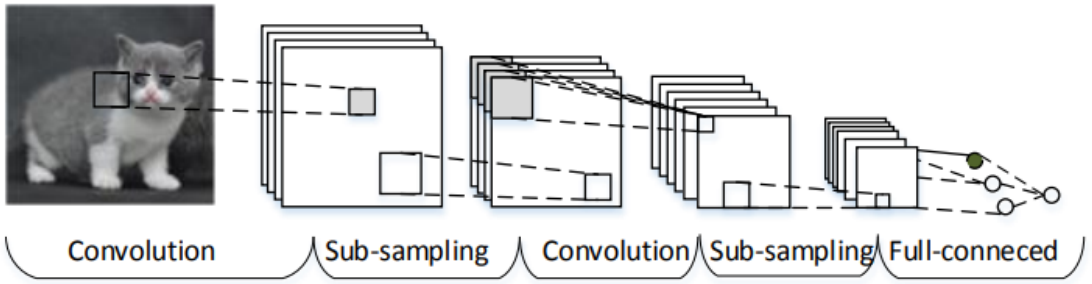


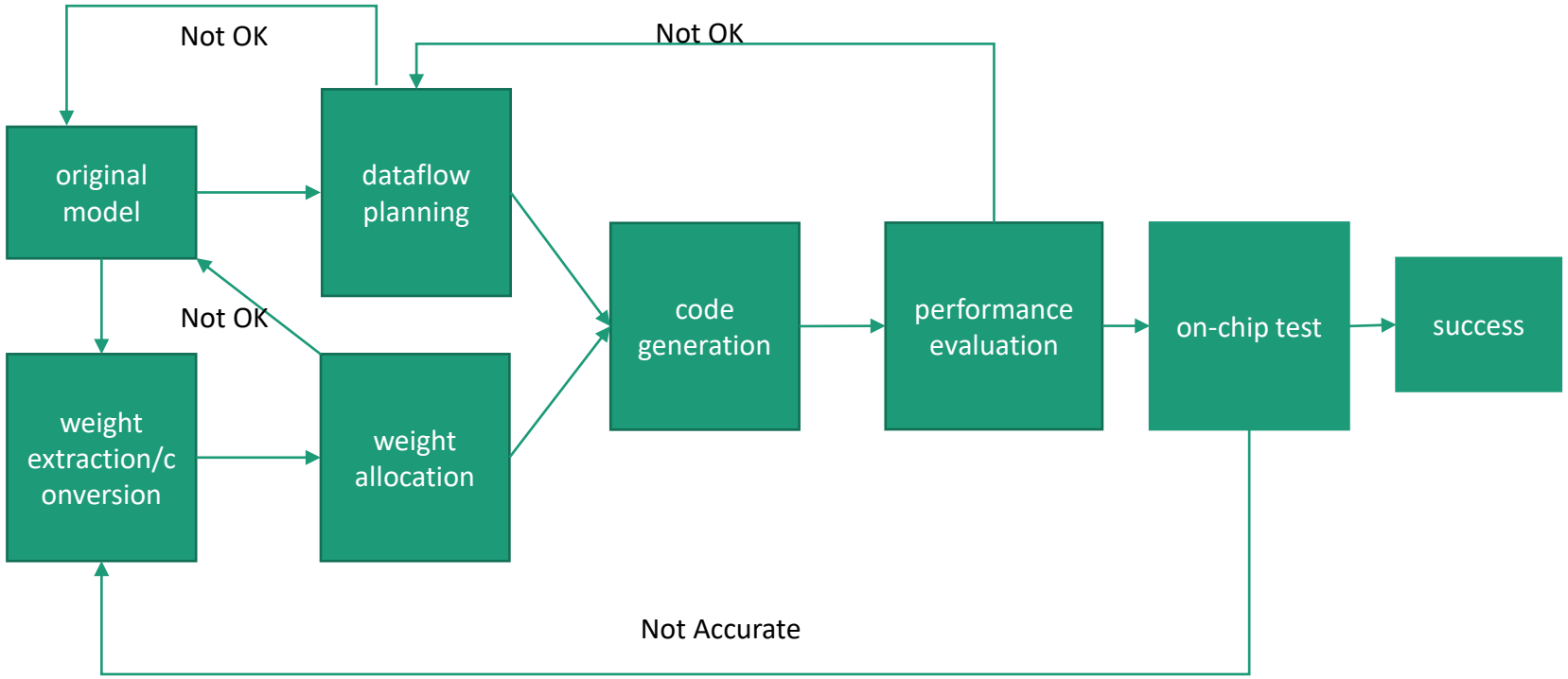


L. Zhao *et al.,* AICIS, 2021

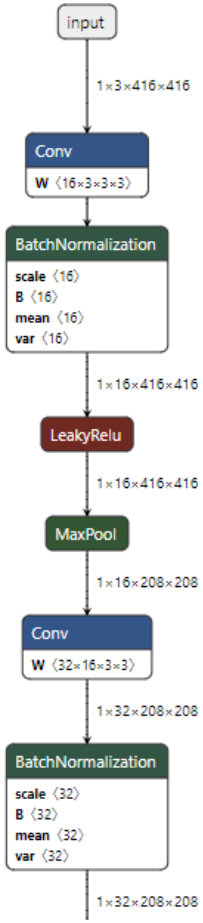
NN model deployment

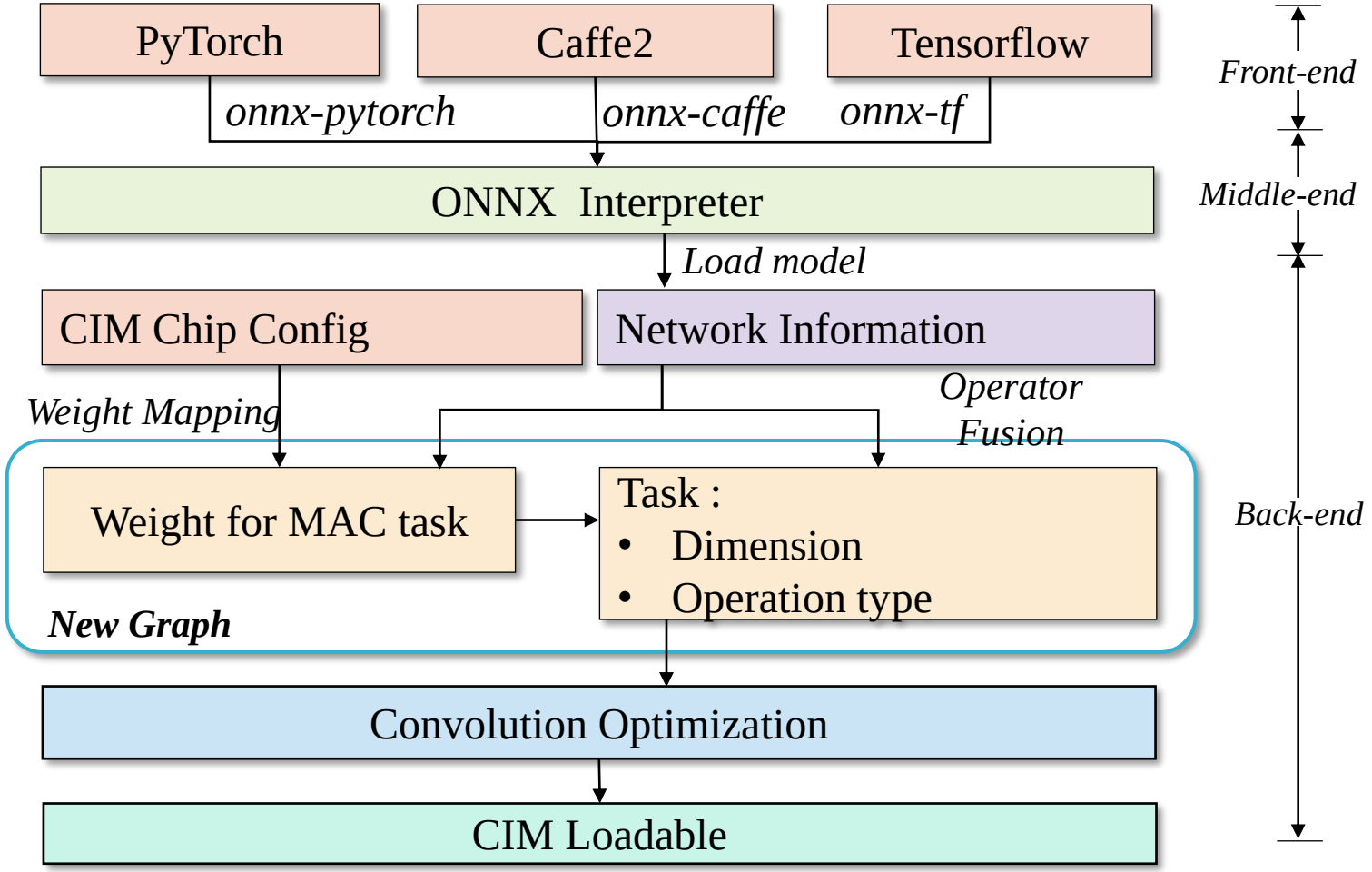
- Trained on GPUs
- Inferenced by CIM hardware
- In-house CIM model deployment flow





A CIM-aware NN compiler

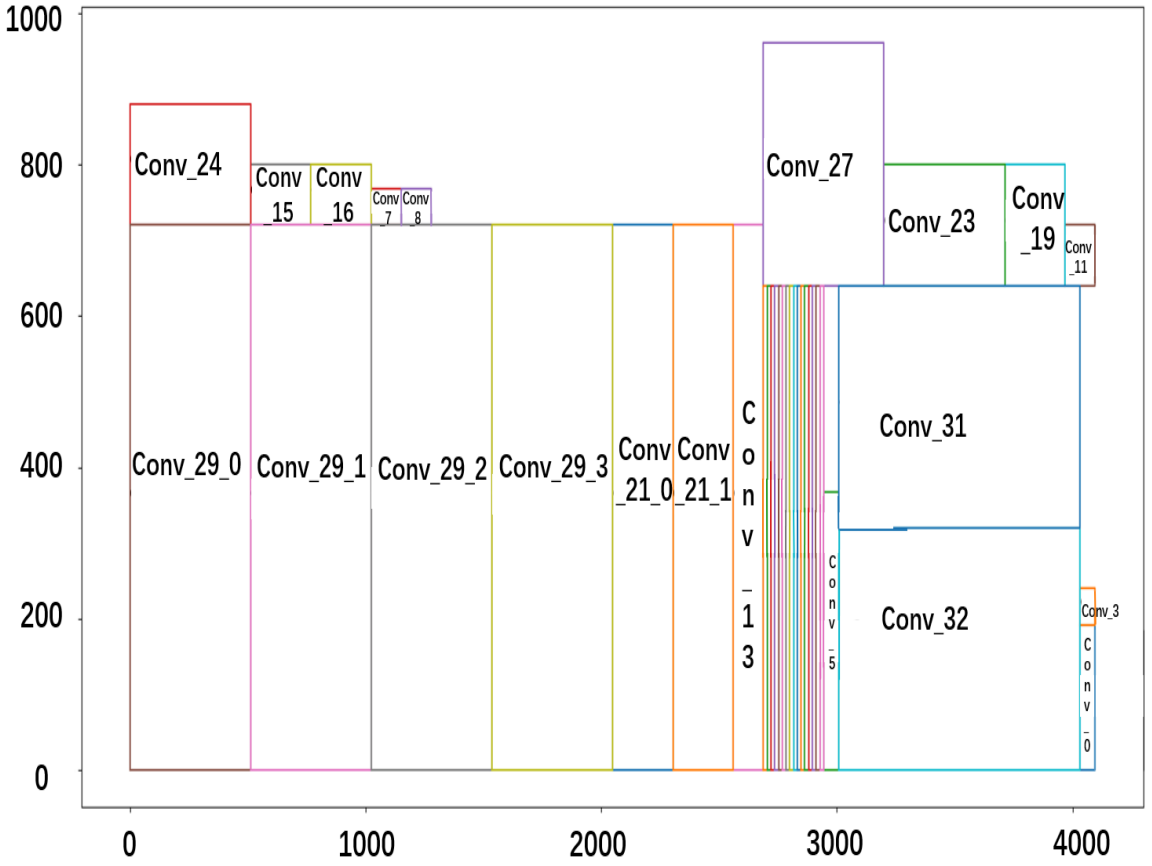




C. Yang *et al.*, WCCCT, 2023

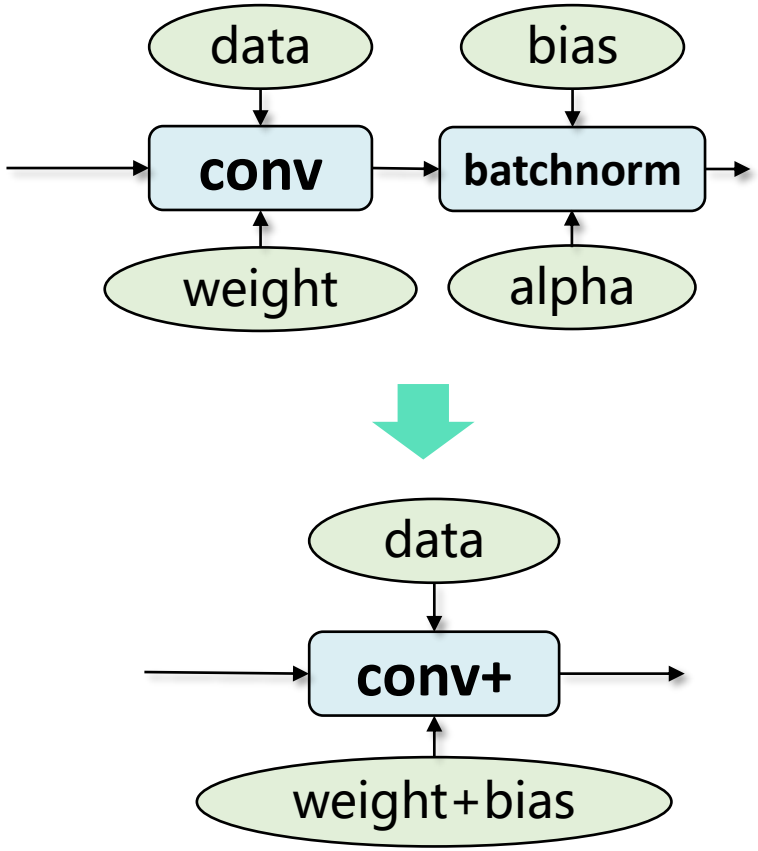
Weight Mapping

- Weights need to be mapped to memory arrays
- A much simpler floorplan problem
  - No performance/WL optimization
  - Small amount of blocks
- A greedy algorithm is applied.
- New constraints for future work
  - Multi-core mapping
    - Performance driven
    - data-flow related
  - Adding redundancy
  - IR-drop aware



Operator fusion

- Operations can be merged without cost



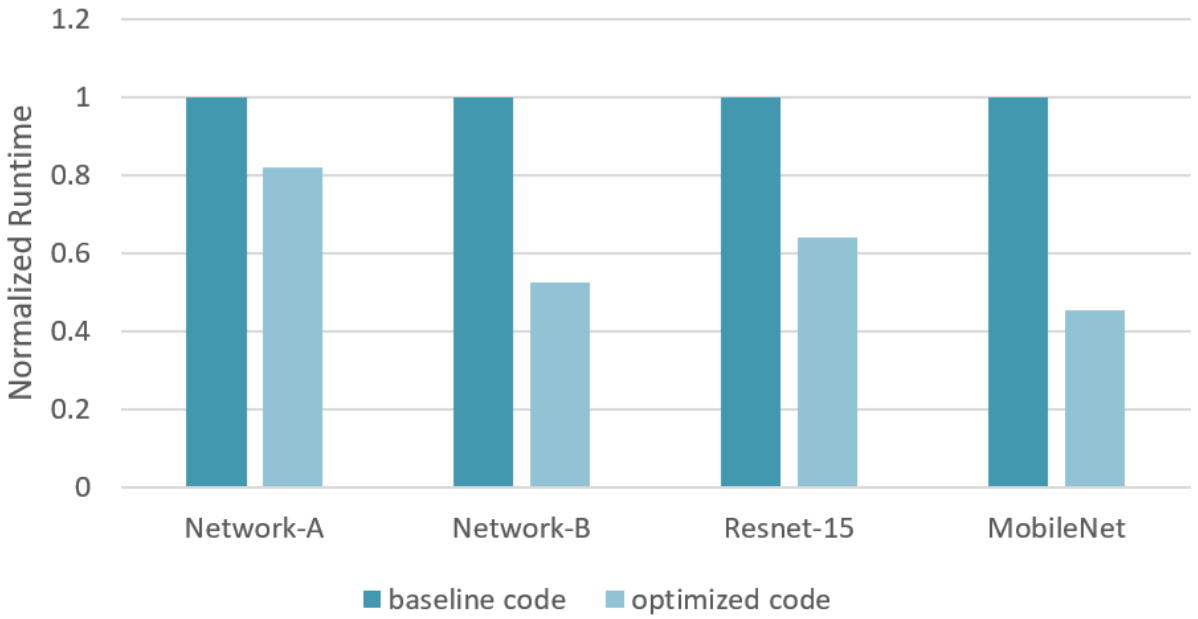
$$Y = alpha * (W * X + convbias) + batchbias$$
$$= (alpha * W) * X + (alpha * convbias + batchbias)$$

Merge bias into matrix

$$\begin{bmatrix} W_{11} & \cdots & W_{1N} \\ \vdots & \ddots & \vdots \\ W_{M1} & \cdots & W_{MN} \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_N \end{bmatrix} \Rightarrow \begin{bmatrix} W_{11} & \cdots & W_{1N} & B_1 \\ \vdots & \ddots & \vdots & \vdots \\ W_{M1} & \cdots & W_{MN} & B_M \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_N \\ 1 \end{bmatrix}$$

Compiler effectiveness

- ***Overall inference runtime comparison :***



1.2 x to 2.3 x acceleration

Performance comparison between baseline code and optimized code for four networks.

significantly reduces model deployment time and cost.

C. YANG *et al.*, WCCCT, 2023

Conclusion

- Computing-in-memory becomes real
- Still facing a lot of challenges
  - Analog Computing accuracy
  - Easy to use tool chain
  - Technology readiness
  - Application ecosystem
  - …
- Look forward to that CIM chips become widely used in our daily life.

Thanks!