IMCE: Energy-Efficient Bit-Wise In-Memory Convolution Engine for Deep Neural Network

Deliang Fan

Assistant Professor

dfan@ucf.edu

http://www.eecs.ucf.edu/~dfan/

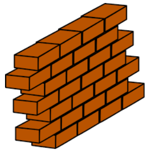
Department of Electrical and Computer Engineering,
University of Central Florida, Orlando, FL

OUTLINE

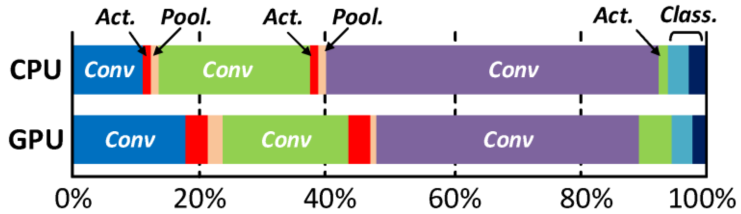
- Motivation
- STT-MRAM and SOT-MRAM
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion

MOTIVATION (CNN WALL)

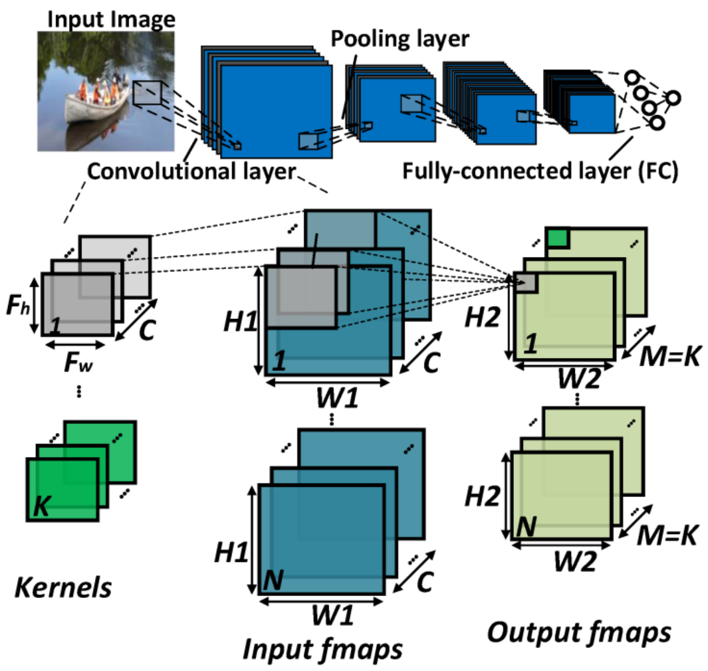
- Convolutional neural networks (CNNs) are reaching record-breaking accuracy in image recognition on small data sets like ***MNIST***, ***SVHN*** and ***CIFAR-10*** with accuracy rates of 99.79%, 98.31% and 96.53% [1].
- On the large data-sets like ***ImageNet*****,** ResNet shows a prominent recognition accuracy (96.43%) even higher than humans! (94.9%).
- Following the trend, when going deeper in CNNs (e.g. ResNet employs 18-1001 layers), memory/computational resources and their communication have faced inevitable limitations called "CNN power and memory wall") [1,2].
- Several methods have been proposed to break the wall:



A. Compressing pre-trained networks,
B. Quantizing parameters, and
C. Binarization
D. Prunning



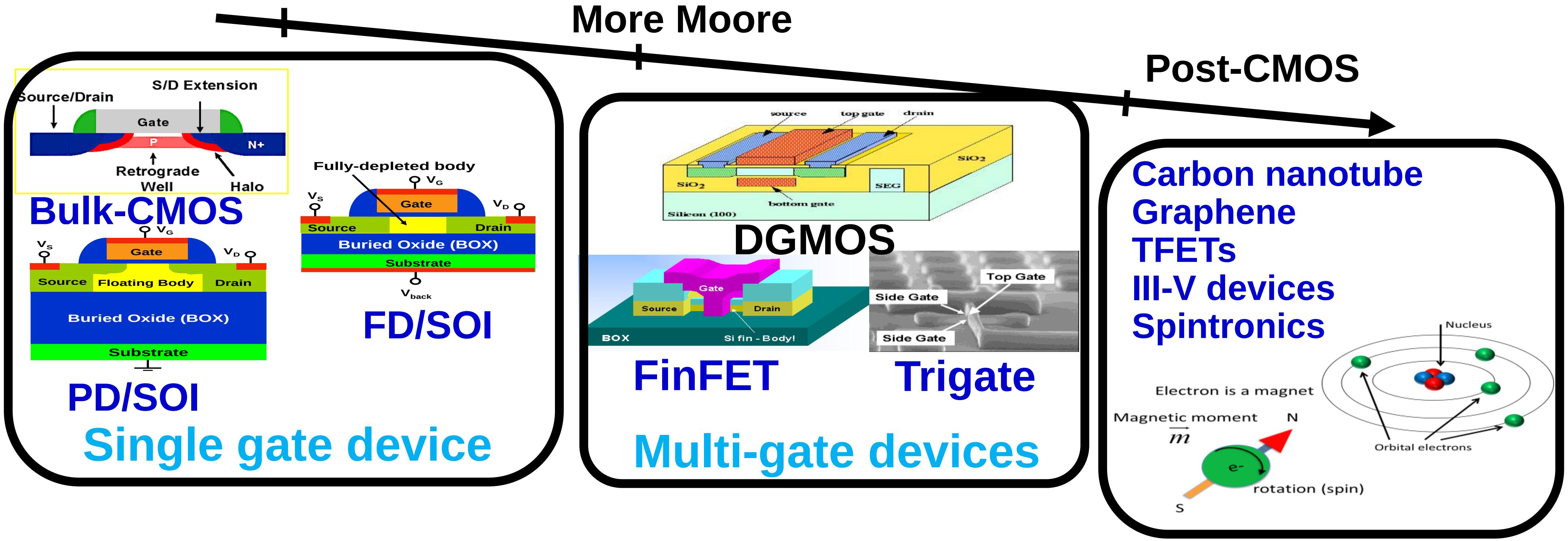
Execution time of a sample CNN for scene labeling on CPU and GPU [3]. Convolutional layer always takes most fraction of execute time and computational sources



Visualization of Inference in CNN

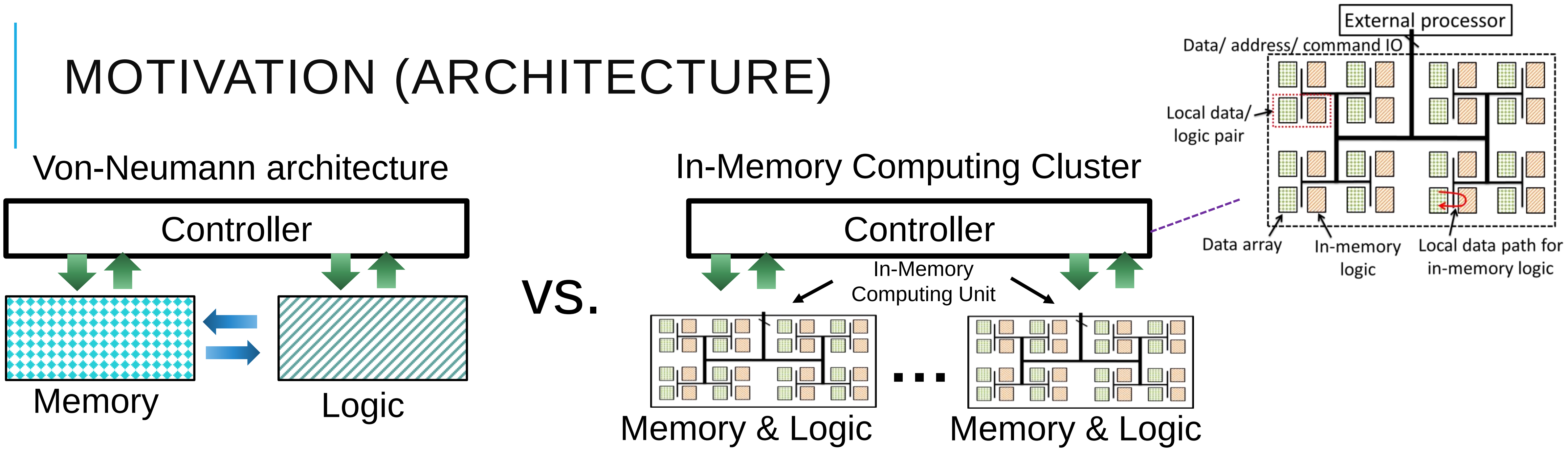
[1] R. Andriet al., "Yodann: An architecture for ultra-low power binary-weight cnn acceleration," IEEE TCAD, 2017
[2] Y.-H. Chen et al., "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," IEEE Journal of Solid-State Circuits, vol. 52, 2017.
[3] L. Cavigelli et al., "Accelerating real-time embedded scene labeling with convolutional networks," in DAC, 2015 52nd ACM/EDAC/IEEE. IEEE, 2015, pp. 1–6.

MOTIVATION (DEVICE)-TECHNOLOGY TREND



- Energy efficient and high performance computing hardware development is beginning to stall fundamentally due to limitations in both ***devices*** and ***architectures***.
- First, the current computing platforms primarily depend on Complementary Metal Oxide Semiconductor (CMOS) technology, which is reaching its power wall

MOTIVATION (ARCHITECTURE)



- ➤ Energy hungry data transfer
- ➤ Long memory access latency
- ➤ Limited memory bandwidth

- ✓ Parallel, local data processing
- ✓ Short memory access latency
- ✓ Ultra-low energy
- ✓ Programmable, Low cost/ area

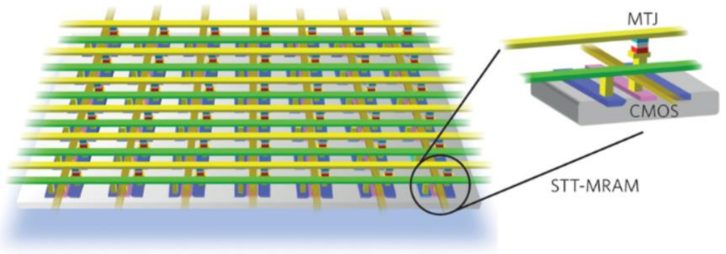
- There is an urgent need to investigate fundamentally different devices and architectures for information processing and data storage with the ability to continuously deliver energy efficient and high performance computing solutions.

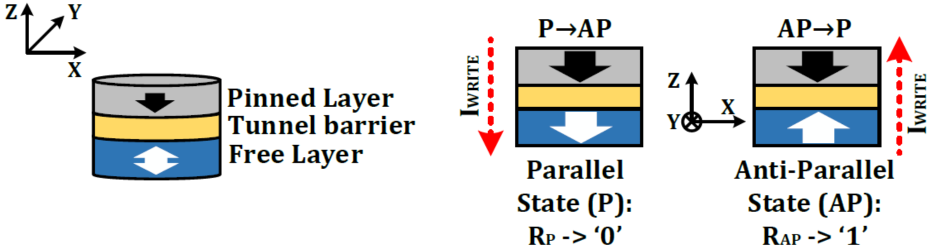
[1] P. Chi et al., “Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory,” in ISCA, vol. 43, 2016.
[2] S. Li et al., “Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories,” in 2016 53nd DAC. IEEE, 2016..

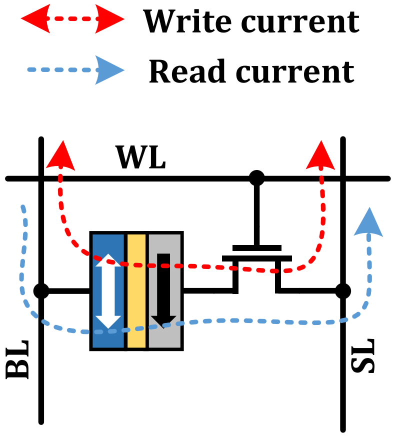
OUTLINE

- Motivation
- STT-MRAM and SOT-MRAM
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion

STT-MRAM



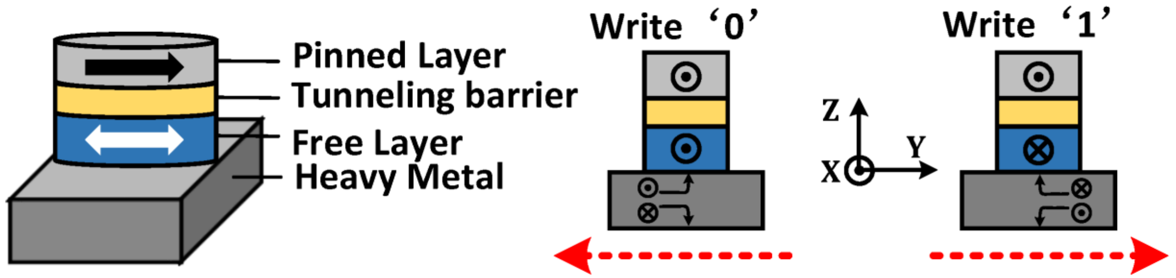




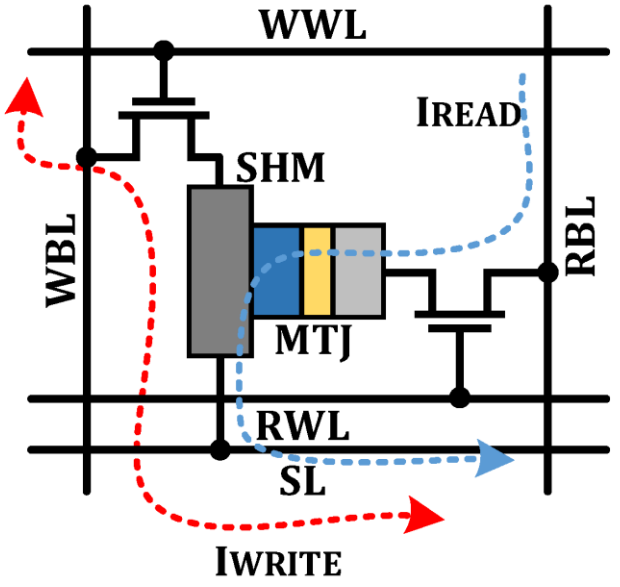
Operations	Write '1'('0')	Read
WL	V_{DD}	V_{DD}
BL	GND (V_{DD})	I_{sense}
SL	V_{DD} (GND)	GND

- Magnetic Tunnel Junction (MTJ) owns high/low resistance with respect to its free layer magnetization configuration.
- Current-induced Spin Transfer Torque (STT) MTJ switching scheme
- **Key Advantages**
  - Non-volatility
  - High density
  - No leakage power
- **Limitations**
  - Write asymmetry
  - Reliability-limited write speed
  - Read write optimization conflicts

SOT-MRAM



$$\vec{I}_S = \theta_{SH} \left(\hat{\sigma} \times \vec{I}_q \right) \left(\frac{A_{MTJ}}{A_{HM}} \right) \left(1 - \operatorname{sech} \left(\frac{t_{HM}}{\lambda_{sf}} \right) \right)$$



Key Advantages

- Energy-efficient write
- Decoupled R/W current paths
- Separate optimization for Read and for Write

Limitations

- Requires two access transistors
- Switching PMA MTJ requires FL engineering that involves fabrication challenge

Operations	Write '1'('0')	Read
WWL	V_{DD}	0
RWL	0	V_{DD}
RBL	0	I_{READ}
WBL	$V_{WP}(V_{WN})$	0
SL	0	0

The bit-cell structure of 2T1R SOT-MRAM and its biasing conditions

[1] G. Prenat et al., "Beyond stt-mram, spin orbit torque ram sot-mram for high speed and high reliability applications," in Spintronics-based Computing. Springer, 2015.
[2] C.-F. Pai et al., "Spin transfer torque devices utilizing the giant spin hall effect of tungsten," Applied Physics Letters, 2012.

OUTLINE

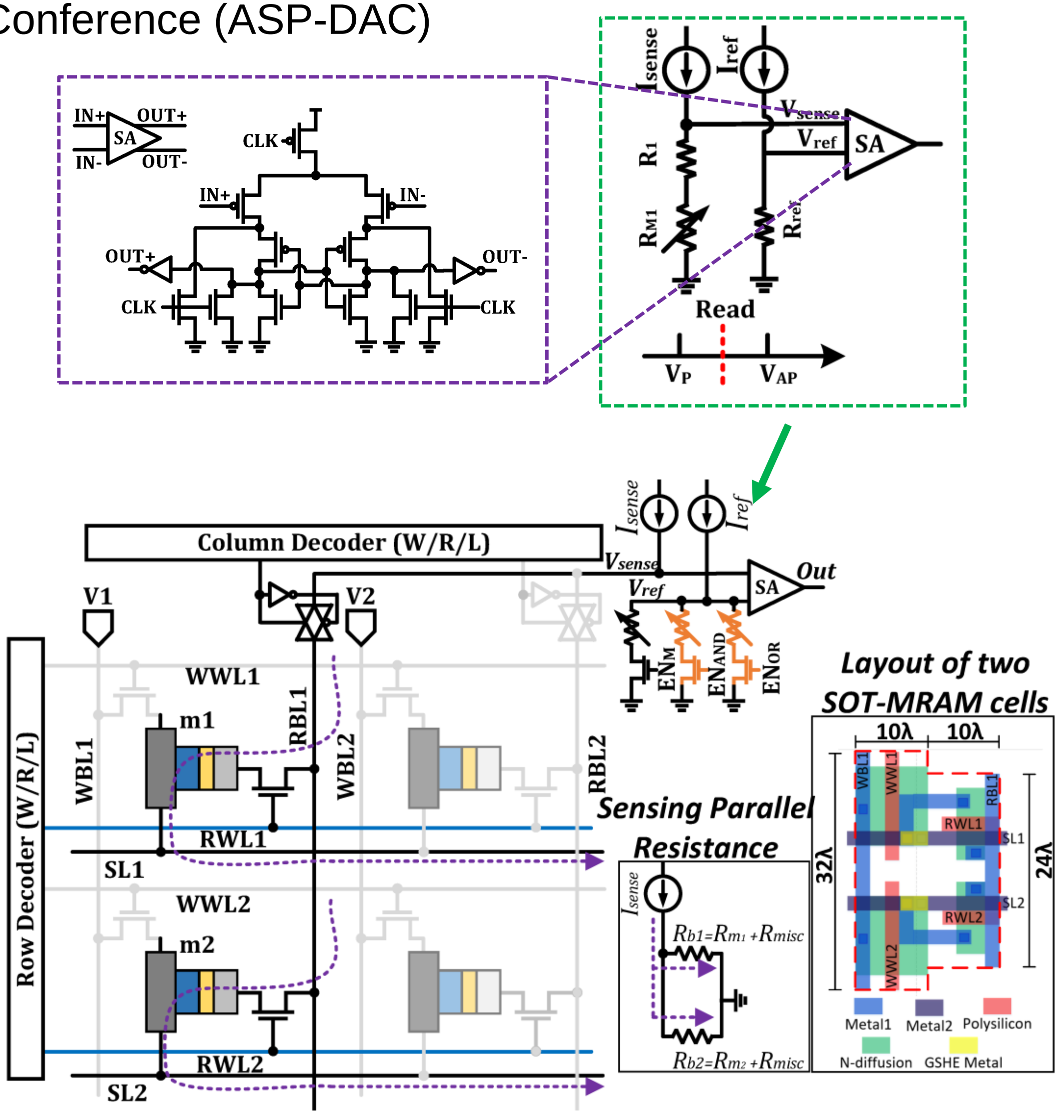
- Motivation
- STT-MRAM and SOT-MRAM
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion

IN-MEMORY PROCESSING PLATFORM

- Dual mode architecture that perform both memory read-write and AND/OR logic operations.
- ***Memory Write***: 1. WWL1 should be activated by the Row Decoder and SL1 is grounded. 2. To write '1' (/ '0'), the voltage driver (V1) connected with WBL1 is set to positive (/negative) write voltage.

charge current (~120 μA)
1ns switching speed

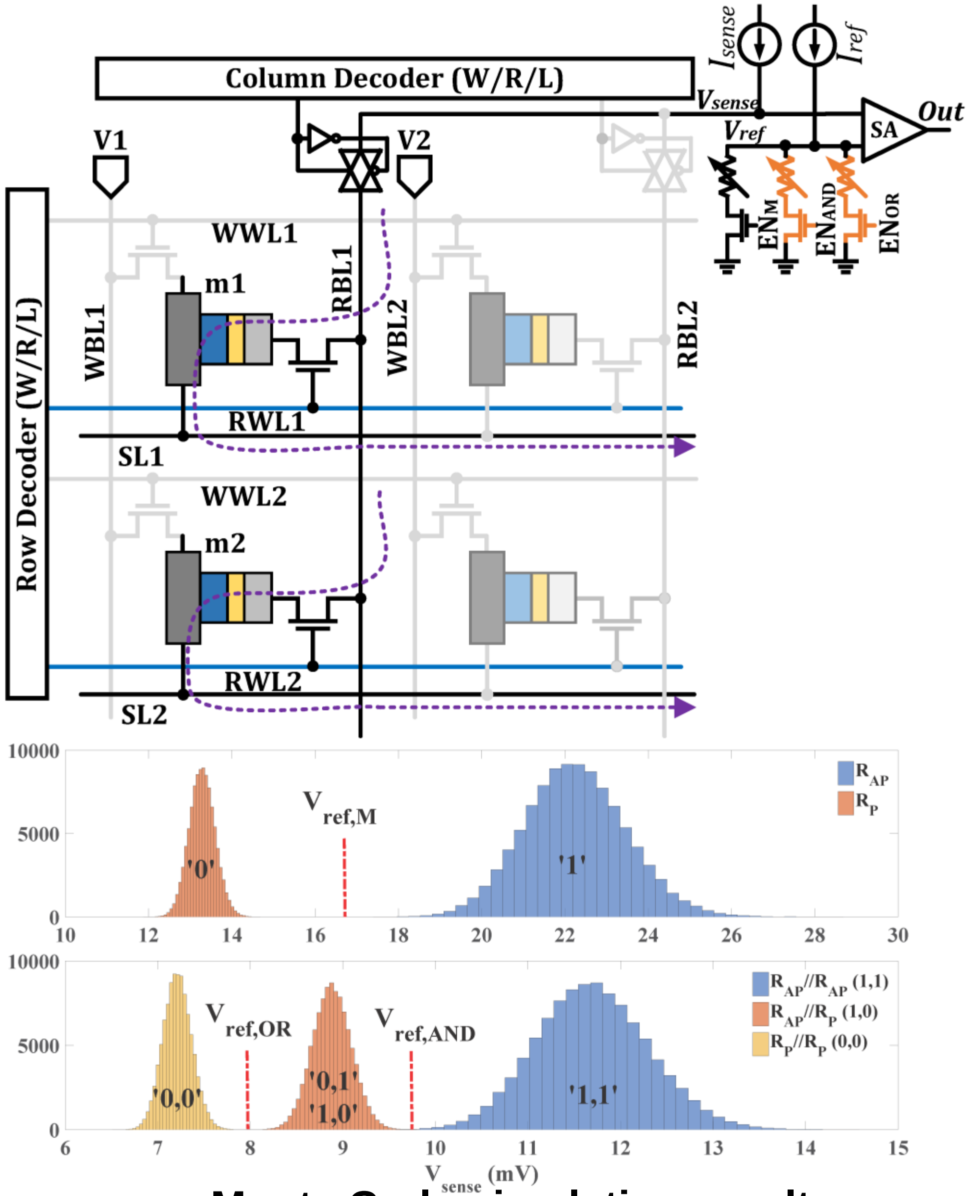
- ***Memory Read***: 1. RWL1 is activated while SL1 is grounded. 2. The Column Decoder activates the RBL1 line to be connected to the SA. 3. a read current flows from the SOT-MRAM cell to ground, generating a sense voltage, which is compared with V_{ref} : $V_{sense,P} < V_{ref} < V_{sense,AP}$ where: $(EN_M, EN_{AND}, EN_{OR}) = (1,0,0)$
- ***Computing Mode***: Every two bits stored in the identical column can be selected and sensed simultaneously. Through selecting different reference resistances $(EN_M, EN_{AND}, EN_{OR})$, the SA can perform basic in-memory Boolean functions (i.e. AND and OR).



Proposed SOT-MRAM based dual-mode in-memory processing architecture. The layout of two adjacent SOT-MRAM cells is also indicated

IN-MEMORY PROCESSING PLATFORM

- For AND operation, R_{ref} is set at the midpoint of $R_{AP} \| R_P = (1,0)$ and $R_{AP} \| R_{AP} = (1,1)$
- For OR operation, R_{ref} is set at the midpoint of $R_P \| R_P$ and $R_P \| R_{AP}$
- We have performed Monte-Carlo simulation with **100000 trials**. A $\sigma = 5\%$ variation is added on the Resistance-Area product (RA_P), and a $\sigma = 10\%$ process variation is added on the TMR.
- Sense Margin will be reduced by increasing the logic fan-in (i.e. number of parallel memory cells).
- To avoid read failure, only two fan-in in-memory logic is used in this work.



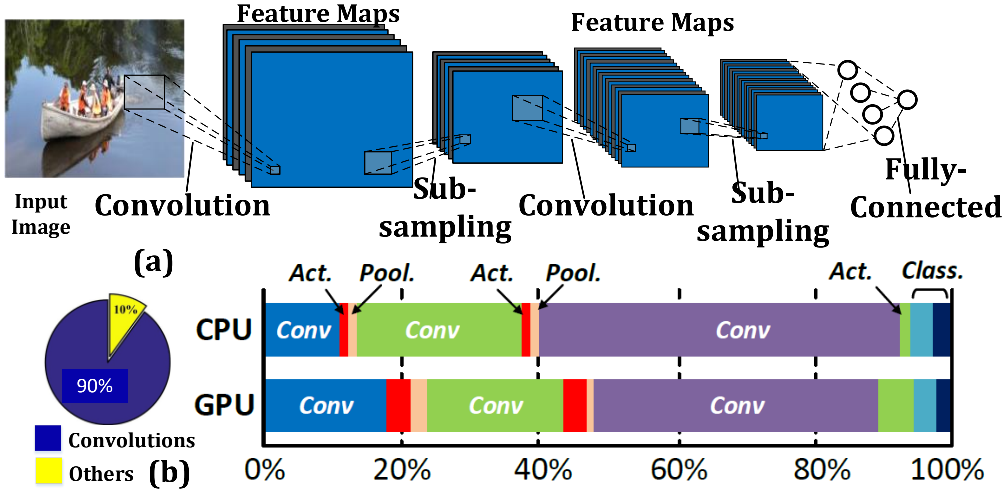
Monte Carlo simulation result

OUTLINE

- Motivation
- Spintronic Devices
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion

BIT-WISE CONVOLUTIONAL NEURAL NETWORK

- DoReFa-Net [1] proposes to employ low bitwise convolutions (weights and activations/gradients) during forward/backward passes to accelerate both training and inference.
- Quantize floating point convolution into bit-wise convolution with limited accuracy loss for large scale IMAGENET benchmark in different configurations.
- Such bit-wise convolution could be totally implemented within our proposed accelerator using IMCE.



$$O[n][k][x][y] =$$

$$ReLU(B[k] + \sum_{i=0}^{F_h-1} \sum_{j=0}^{F_w-1} \sum_{z=0}^{C-1} I[n][z][U_x + i][U_y + j] \times W[k][z][i][j]),$$

$$0 \le n < N, 0 \le k < K, 0 \le x < W2, 0 \le y < H2$$

Typical dot-product operation of convolutional layer

$$I * W = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} 2^{m+n} bitcount(and(C_n(W), C_m(I)))$$

Low bit-width fixed-point integers dot-product developed by DoReFa-NET

[1] S. Zhou *et al.*, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2016.

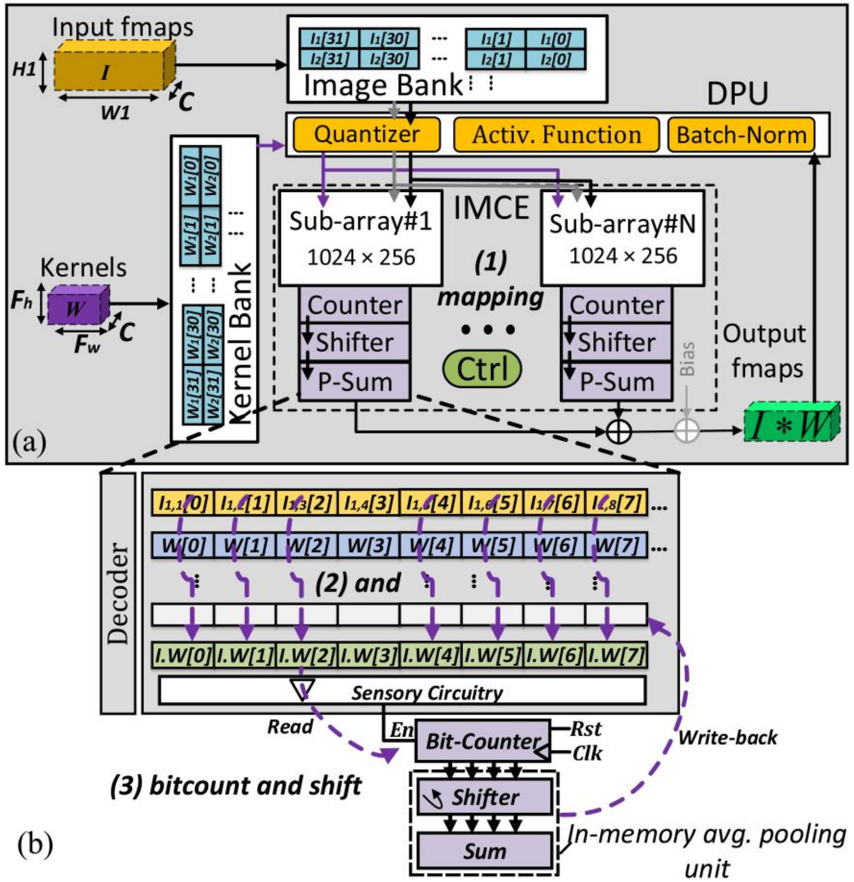
IN-MEMORY CONVOLUTION ENGINE CONCEPT AND IMPLEMENTATION

- A potential solution to better address storage, computation, and data transfer bottlenecks of CNNs.
- This architecture mainly consists of Image Bank, Kernel Bank, bit-wise In-Memory Convolution Engine (IMCE), and Digital Processing Unit (DPU).
- Preprocessing:

✓Assume Input fmaps (I) and Kernels (W) are stored in Image Banks and Kernel Banks of memory.

✓Inputs need to be constantly quantized before mapping into computational sub-arrays. This step is performed using DPU's Quantizer and then the results are mapped to IMCE's sub-arrays.

✓IMCE is realized through the proposed SOT-MRAM based computational sub-array.



(a) General overview of the proposed CNN accelerator with image bank, kernel bank, computational sub-arrays, and DPU, (b) Bit-wise IMCE's sub-array.

IN-MEMORY CONVOLUTION ENGINE CONCEPT AND IMPLEMENTATION

- The main idea is to exploit *logic AND*, *bitcount*, and *bitshift* as rapid and parallelizable operations to accelerate the MACs in convolutional layers.

- Operation:

➢ I is a sequence of *M*-bit input integers (3-bit, here) located in input fmap covered by sliding kernel of W, such that $I_i \in I$ is a 3-bit vector representing a fixed-point integer (e.g. 3 = "011")

➢ We index the bits of each I_i element from LSB to MSB with *m = [0,M − 1]*, such that *m = 0* and m = M − 1 are corresponding to LSB and MSB.

➢ A second sequence $C_m(I)$ including the combination of *mth* bit of all I_i elements (shown by colored elliptic) (e.g. $C_0(I)$ vector is "0110"). Thus I can be written as:

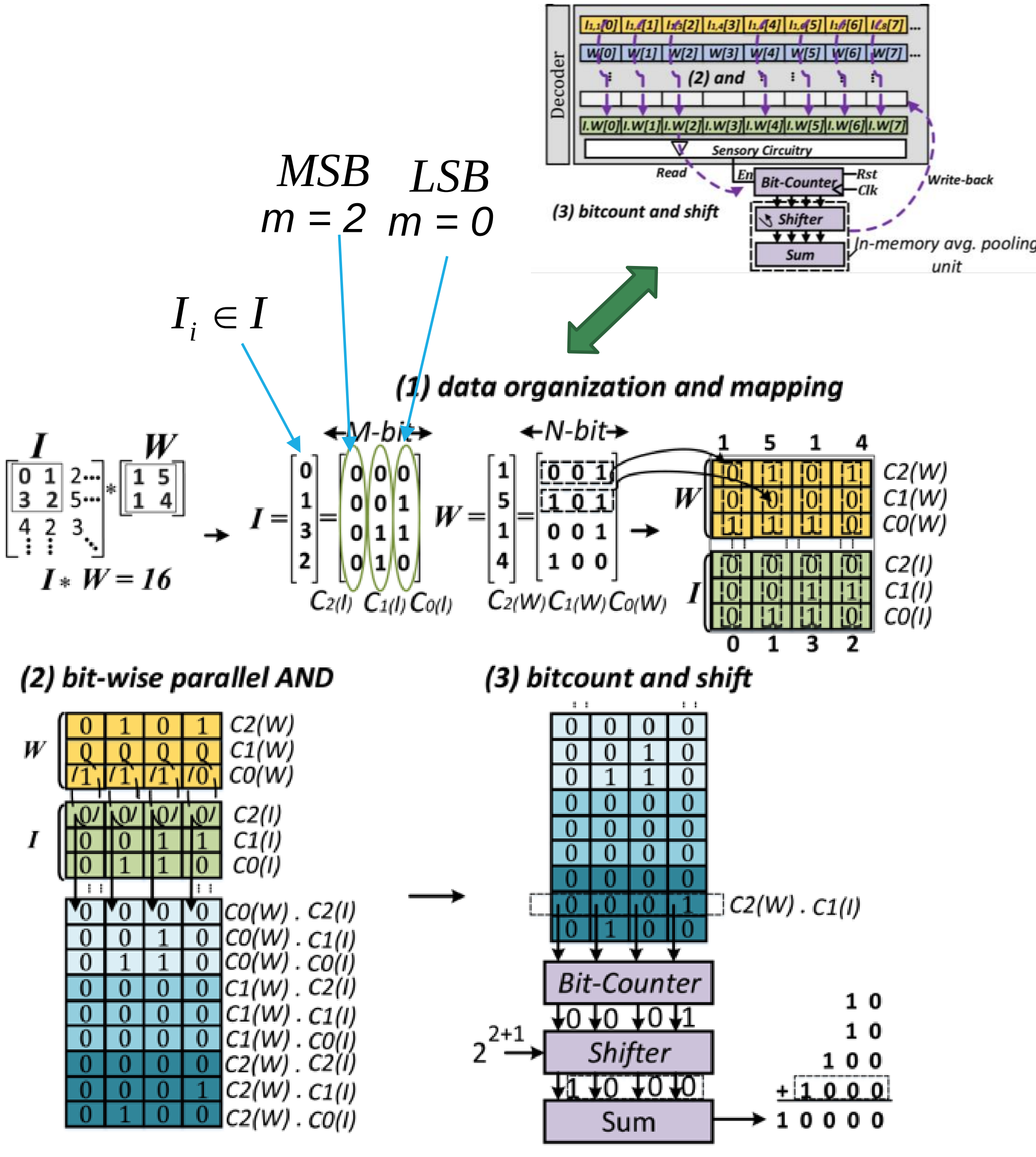
$$I = \sum_{m=0}^{M-1} 2^m C_m(I)$$

➢ Same procedure is applied to *W* as a sequence of N-bit weight integers (3-bit, herein) to make:

$$W = \sum_{n=0}^{N-1} 2^n C_n(W)$$

➢ In this way, the convolution between I and W can be defined as:

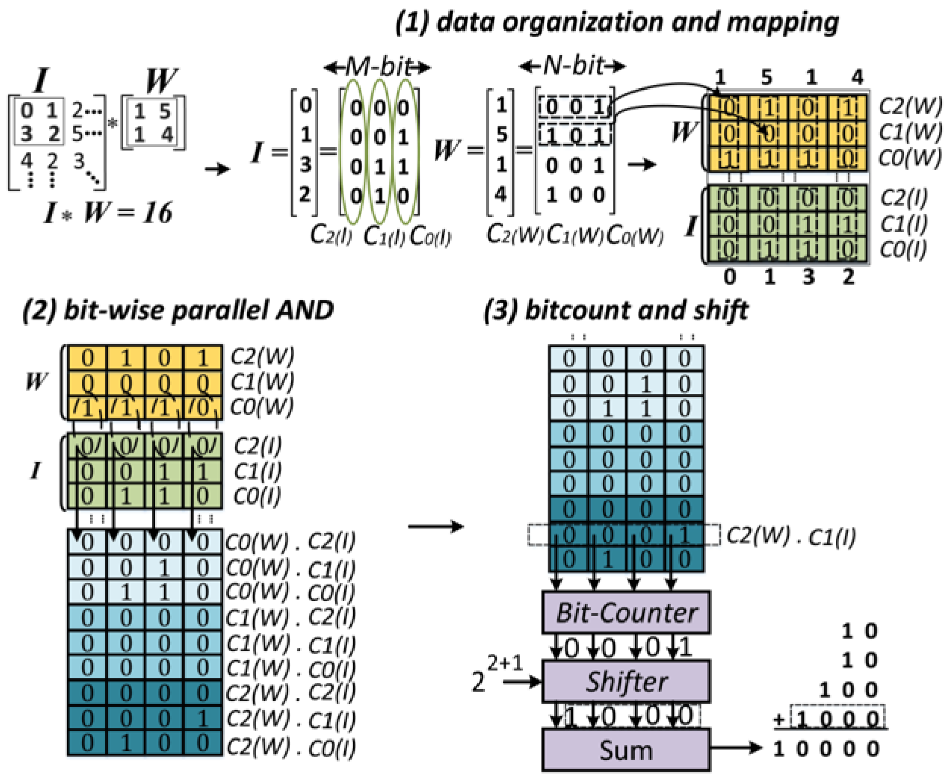
$$I * W = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} 2^{m+n} bitcount(and(C_n(W), C_m(I)))$$



Mapping method and operations of bit-wise IMCE.

IN-MEMORY CONVOLUTION ENGINE CONCEPT AND IMPLEMENTATION

1) Mapping $C2(W)$-$C0(W)$ to the designated sub-array rows,
2) Mapping $C2(I) - C0(I)$ in the following memory rows in the same way,
3) Performing bit-wise parallel AND operation of $C_n(W)$ and $C_m(I)$ using the proposed SOT-MRAM computational sub-array,
4) The Bit-Counter readily counts the number of "1"s and passes it to the Shifter unit,
5) The Shifter left-shifts input data by specific number of bits, here "0001", as result of Bit-Counter is left-shifted by 3-bit ($\times 2^{2+1}$) to "1000",
6) Eventually, Sum unit adds the Shifter unit's outputs to produce the output fmaps,
7) The output fmaps coming from convolutional layer can be later processed for down-sampling using average pooling performed using Sum and Shifter units.



Mapping method and operations of bit-wise IMCE.

DIGITAL PROCESSING UNIT (DPU)

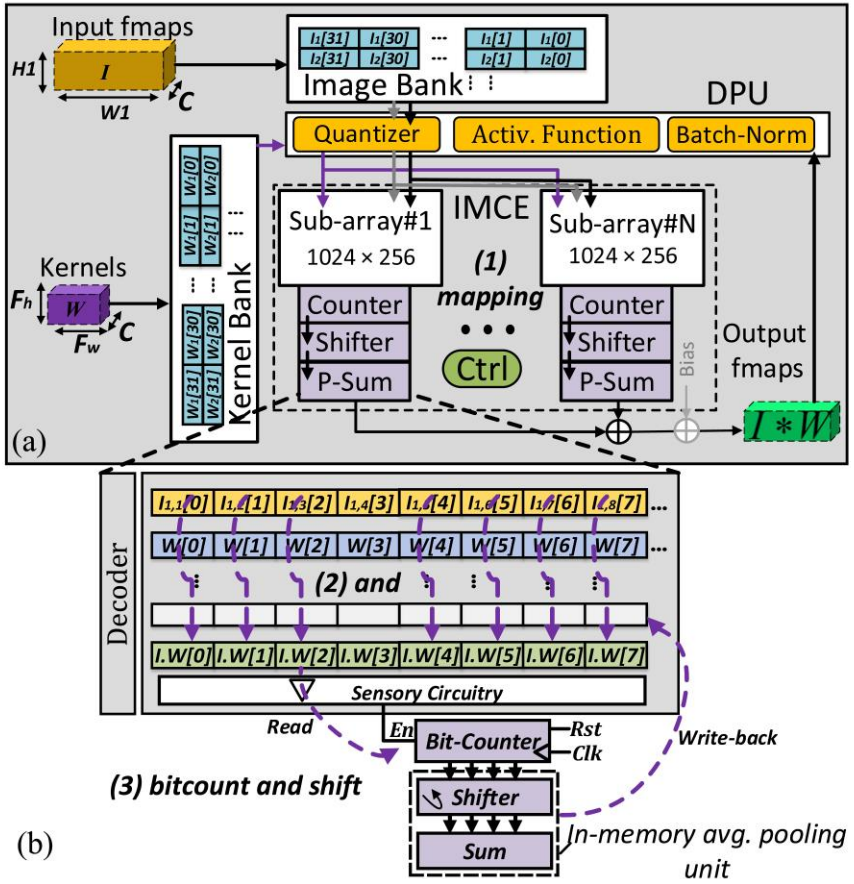
- **Quantizer:** This unit quantizes a real number input r_i [0, 1] to a k-bit number output r_o [0, 1] using quantization Function:

$$r_o = \frac{1}{2^k - 1} round((2^k - 1) r_i)$$

- **Batch-Norm.:** Batch Normalization layer alleviates the in formation loss during quantization by normalizing the input batch to have zero mean and unit variance.

$$I_o(R) = \frac{I_i(R) - \mu}{\sqrt{\sigma^2 + \varepsilon}} \gamma + \beta$$

- **Activ. Function:** The proper selection of activation function has a profound impact on network prediction accuracy specially in lower bit-width CNN. This unit can be reconfigured to perform two distinct activation functions ($\frac{tanh(x)+1}{2}$, $sign(x)$)



OUTLINE

- Motivation
- STT-MRAM and SOT-MRAM
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion

PERFORMANCE EVALUATION

Device to System Level Simulations:

Device Level:

Verilog-A model of 3T SHE-MTJ device was developed to co-simulate with the interface CMOS circuits in SPICE to validate the functionality and evaluate performance of the proposed design.

Circuit Level:

45nm North Carolina State University (NCSU) Product Development Kit (PDK) [1] library is used in SPICE to verify the proposed design and evaluate the performance.

System Level:

We employ the modified self-consistent NVSim [2] along with an in-house developed C++ code to verify the performance of memory.

[1] www:eda.ncsu.edu/wiki/FreePDK45
[2] X. Dong et al., "NVSim: A circuit-level performance, energy, and area model for emerging non-volatile memory," Springer, 2014, pp. 15-50.

PERFORMANCE EVALUATION

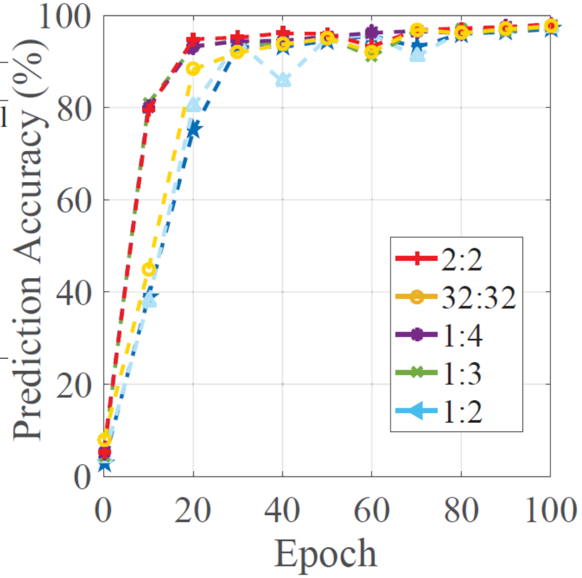
Accuracy:

- **Bit-width configuration:** Six bit-width configurations of W:I (32:32, 1:1, 1:2, 1:3, 1:4, and 2:2) are considered. The 8-bit gradient is applied to all configurations except 32:32.
- **Model:** A CNN with 6 (bit-wise) convolutional layers, 2 (average) pooling layers and 2 FC layers that cost about 80 FLOPs for a 40×40 image. FC layers are equivalently implemented by convolutions.
- **Data-set:** The SVHN. The cropped format of colored images (32×32) centered around each single digit is selected.
- **Training:** Modification on open source algorithm by DoReFa-Net, we adopt batch normalization and different dropout techniques to accelerate and avoid over-fitting. The model is trained on TensorFlow [1].

Test error of CNN model for processing SVHN in different bit-width configuration and Evolution of prediction accuracy vs. epoch.

Config.		Complexity		Test Error
W	I	Inference	Training	CNN Model
32	32	-(†)	-	2.4%
1	1	1	9	3.1%
1	2	2	10	2.6%
1	3	3	11	2.4%
1	4	4	12	2.4%
2	2	4	20	1.8%

(†) The computation complexity of 32:32 is not shown, since it is not computationally efficient to perform bit-wise convolution of 32:32 configuration [3] and it is already reported in previous works.



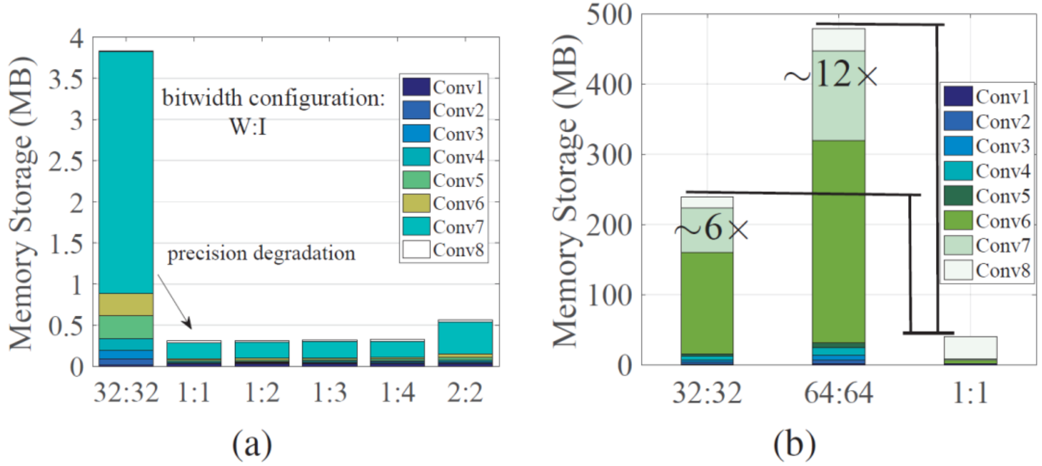
- Complexity of inference and training are achieved using W × I and W × I + W × G, respectively.
- Experiments show that inputs and gradients are progressively more sensitive to bit-width changes.
- The accuracy in different configurations after modifications is almost matched with reported data in DoReFA-NET.

[1] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

PERFORMANCE EVALUATION

Main Memory Storage:

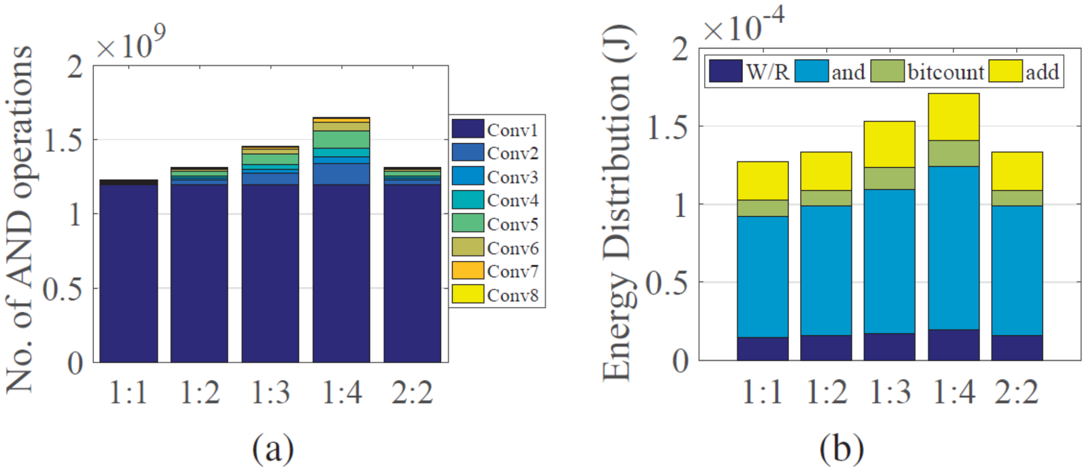
- As shown, lower bit-width the CNN model is, less memory storage is required.
- AlexNet-BCNN mapped to our proposed accelerator requires 39.7MB memory storage which is 12× and 6× less compared to double precision (DP) and single precision (SP) CNNs.



Memory storage (inputs/weights) required by:
(a) CNN model for processing SVHN,
(b) AlexNet architecture for processing ImageNet.

Energy Consumption Estimation:

- The intensity of AND operations processed by IMCE makes the most fraction of energy consumption (up to 65%).
- 1st un-quantized convolutional layer is the most computationally intensive one and the size of computations is diminished for next layers after quantization.
- According to system and application constraints, the designer can choose different configurations considering the accuracy and energy consumption trade-offs.

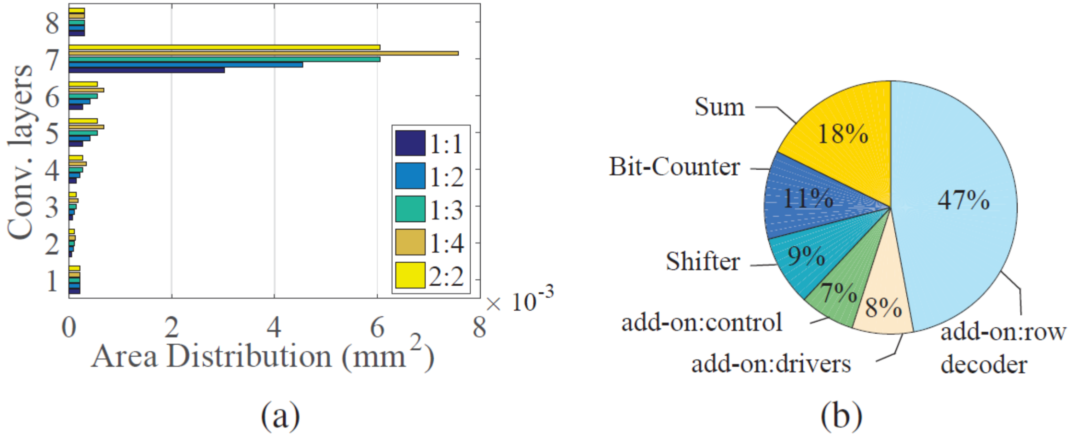


(a) Break-down of no. AND operations in bit-wise IMCE for processing SVHN data-set,
(b) Energy distribution under varying configurations.

PERFORMANCE EVALUATION

Area Estimation:

- The experimental results show that the last two conv layers (converted from fully-connected layers) take up the most part of the area due the high number of weight parameter,
- IMCE imposes 15.4% area overhead to original memory chip. It can be seen that modified row decoder and Sum unit contribute more than 60% of area overhead.



(a) Area distribution of conv. layers mapped to IMCE for processing a single image of SVHN, (b) Breakdown of area overhead of IMCE.

PERFORMANCE EVALUATION

Hardware Mapping Comparison:

- Comparison of two promising resistive memories (i.e. RRAM [8] and SOT-MRAM herein) over three different data-sets in terms of energy and area under 45nm technology node.
- The proposed accelerator exploiting bit-wise IMCE can process Binary CNN (BCNN) over different datasets very efficiently.
- It processes binary-weight AlexNet [1] for ImageNet favorably with 785.25μJ/img where ~ 3× and ~ 4× lower energy and area are achieved, respectively, compared to RRAM-based design.

Performance Estimation of CNN and BCNN accelerators.

	ImageNet		SVHN		MNIST	
Designs	Energy (μJ/img)	Area (mm^2)	Energy (μJ/img)	Area (mm^2)	Energy (μJ/img)	Area (mm^2)
CNN-RRAM [8]	5444.85	21.25	850.42	0.09	18.39	0.054
BCNN-RRAM [8]	2275.34	9.19	425.21	0.085	13.55	0.060
BCNN-SOT-MRAM	785.25	2.12	135.26	0.01	0.92	0.009

[1] M. Rastegari *et al.*, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 525–542.
[8] T. Tang *et al.*, "Binary convolutional neural network on rram," in *22nd ASP-DAC*. IEEE, 2017, pp. 782–787.

OUTLINE

- Motivation
- Spintronic Devices
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion

CONCLUSION

- In this work, we develop a new in-memory processing architecture based on SOT-MRAM, which could be used as both non-volatile memory and reconfigurable in-memory logic.
- The new concept of bit-wise In-Memory Convolution Engine (IMCE) is then developed based on the proposed in-memory processing architecture achieving 3 goals:

(1) All bitwise computation can be implemented within the proposed in-memory accelerator by eliminating massive energy consumption of data communication in traditional architecture between memory and computing units (i.e. CPU/GPU);

(2) Reducing the energy consumption of convolutional layers through utilizing energy efficient intrinsic in-memory computation;

(3) Accelerating the inference task by employing in-memory parallelism.

- Our accelerator can process low bit-width AlexNet on ImageNet data-set favorably with 785.25μJ/img, which consumes ~3× less energy than that of recent RRAM based counterpart. Besides, the chip area is ~ 4× smaller.

THANKS