

# **PRIME: Novel Processing-in-memory Architecture for Neural Network Computation in ReRAM-based Main Memory**

---

Presented by: Ravi Raju

March 23, 2018

QII Presentation Spring 2018

Problem Statement and Solution

Background

Architecture and System Design

Evaluations and Results

Discussion

# Problem Statement and Solution

---

# Problem Statement/Motivation

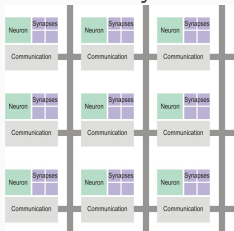
## 1. Neural Networks

- Popular for image/speech recognition application
- High Memory Bandwidth Requirement

## 2. Current Solutions

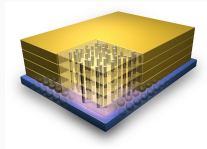
- DaDianNao - large on-chip eDRAM for high bandwidth and data locality
- TrueNorth - SRAM crossbar memory for synapses

## 3. Both solutions suffer from latency of data movement



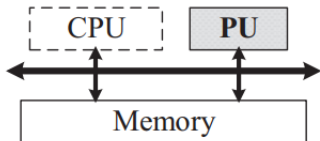
# Proposed Solution: PRIME

1. Processing in Memory is a natural solution
  - Inspired by HMC
  - Place compute units in memory to do NN computation
  - Latency of In-memory data communication vs. DRAM memory access
2. PRIME
  - ReRAM crossbar array solution
  - Dynamically reconfigure between NN accerelation and memory
    - 2.1 Architectural/circuit level support
    - 2.2 Software interface
3. Targets large-scale MLP and CNN applications

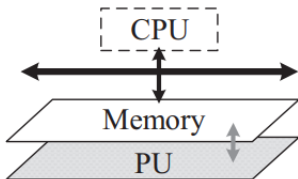


# Key Idea: PRIME

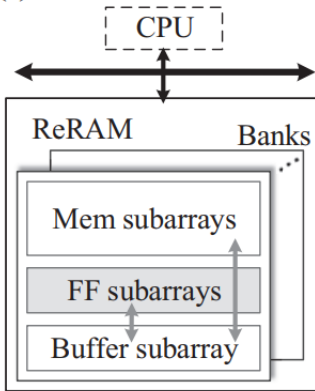
**(a) Processor-Coprocessor Arch.**



**(b) PIM with 3D integration**



**(c) PRIME**



# Background

---

# What is ReRAM

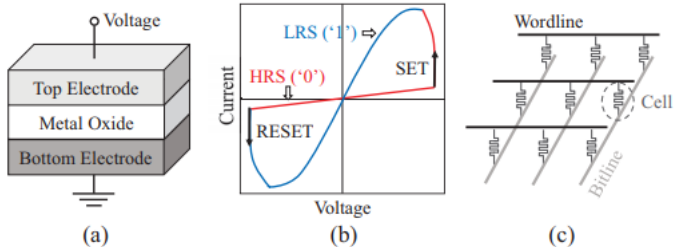
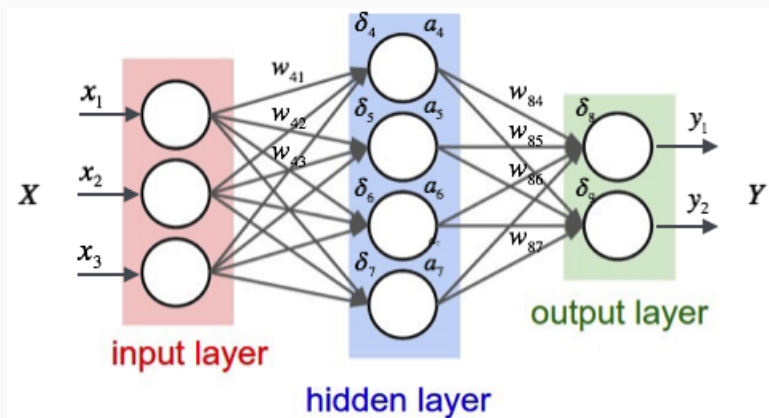


Figure 1. (a) Conceptual view of a ReRAM cell; (b) I-V curve of bipolar switching; (c) schematic view of a crossbar architecture.



# What is a Neural Network



# ReRAM in relation to Neural Nets

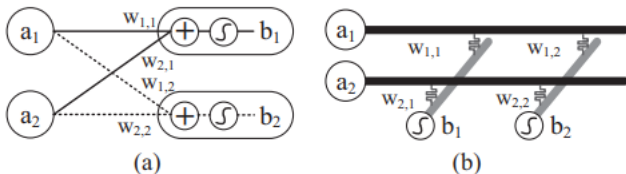
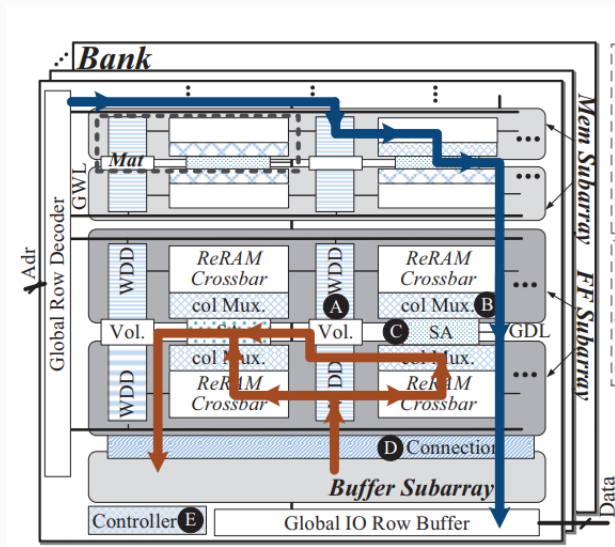


Figure 2. (a) An ANN with one input/output layer; (b) using a ReRAM crossbar array for neural computation.

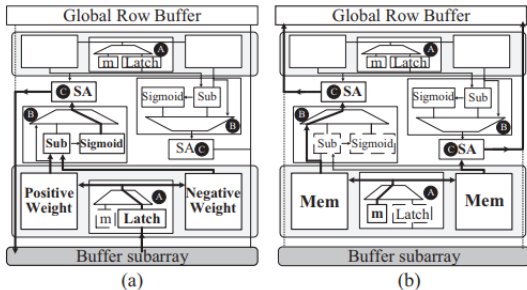
# Architecture and System Design

---

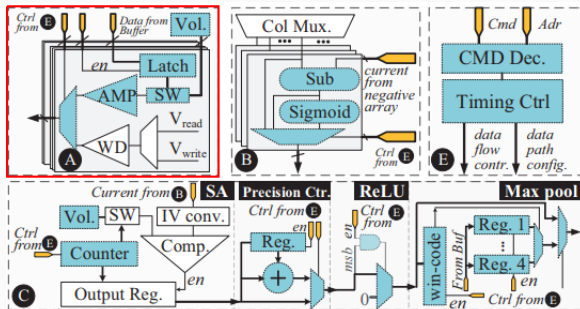
## High Level Overview of Architecture



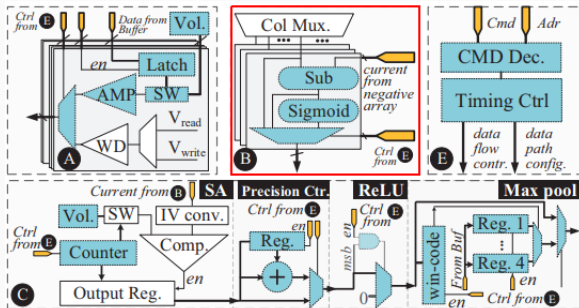
# Choosing between NN Computation and Memory Mode



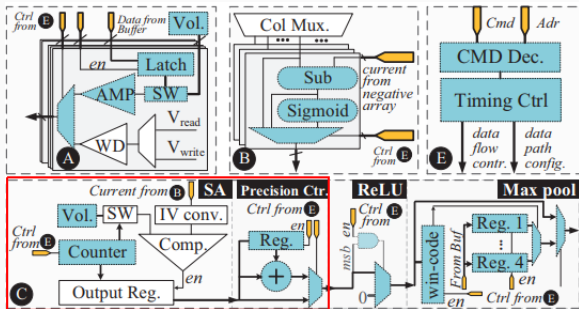
# MicroArchitecture of FF SubArray: Decoder



# MicroArchitecture of FF SubArray: Col Mux



# MicroArchitecture of FF SubArray: SA





# Precision Issues

- Input precision
  - Weight precision
  - Output precision
- 
- Multiple low-precision input signals
  - Multiple cells to make one high precision weight
  - Multiple phases for one computation

# System Level Design

- Small-Scale NN: Replication
- Medium-Scale NN: Split-Merge
- Large-Scale NN: Inter-Bank Communication

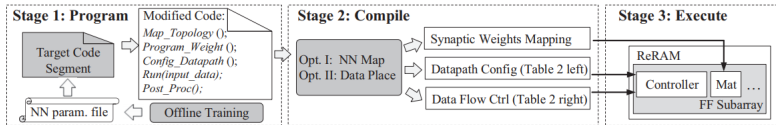


Figure 7. The software perspective of PRIME: from source code to execution.

## Evaluations and Results

---

# Experimental Setup

Table III  
THE BENCHMARKS AND TOPOLOGIES.

<i>MBench</i>		<i>MLP-S</i>	784-500-250-10
<i>CNN-1</i>	conv5x5-pool-720-70-10	<i>MLP-M</i>	784-1000-500-250-10
<i>CNN-2</i>	conv7x10-pool-1210-120-10	<i>MLP-L</i>	784-1500-1000-500-10
<i>VGG-D</i>	conv3x64-conv3x64-pool-conv3x128-conv3x128-pool		
	conv3x256-conv3x256-conv3x256-pool-conv3x512		
	conv3x512-conv3x512-pool-conv3x512-conv3x512		
	conv3x512-pool-25088-4096-4096-1000		

Table IV  
CONFIGURATIONS OF CPU AND MEMORY.

Processor	4 cores; 3GHz; Out-of-order
L1 I&D cache	Private; 32KB; 4-way; 2 cycles access;
L2 cache	Private; 2MB; 8-way; 10 cycles access;
ReRAM-based Main Memory	16GB ReRAM; 533MHz IO bus; 8 chips/rank; 8 banks/chip; tRCD-tCL-tRP-tWR 22.5-9.8-0.5-41.4 (ns)

Table V  
THE CONFIGURATIONS OF COMPARATIVES.

Description		Data path	Buffer
<b>pNPU-co</b>	Parallel NPU [17]	16×16 multiplier	2KB in/out
	as co-processor	256-1 adder tree	32KB weight
<b>pNPU-pim</b>	PIM version of parallel NPU, 3D stacked to each bank		

# Performance

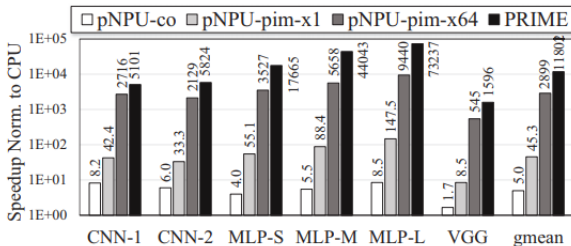


Figure 8. The performance speeds (vs. CPU).

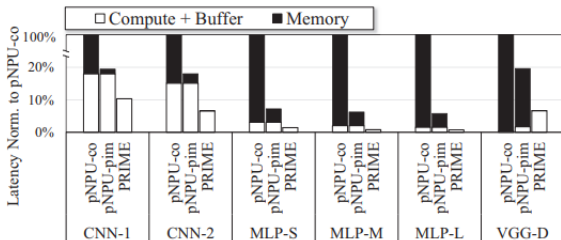


Figure 9. The execution time breakdown (vs. pNPU-co).

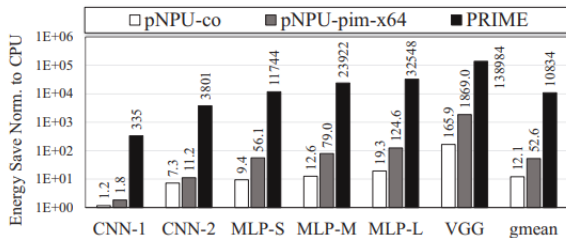


Figure 10. The energy saving results (vs. CPU).

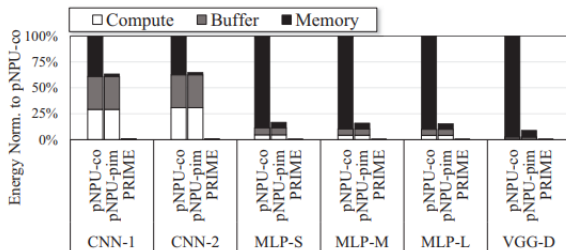


Figure 11. The energy breakdown (vs. pNPU-co).

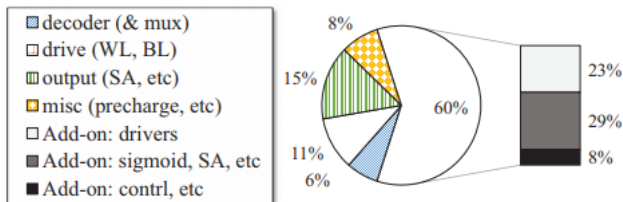


Figure 12. Area Overhead of PRIME.

## Discussion

---



1. PRIME only supports unsigned input vectors
2. Dot-product computations in PRIME are lossy
  - Precision of ADC does not always match precision of dot-product
3. Opportunity to exploit sparsity of NN for energy savings
  - Introduce some logic into pipeline to check how many non-zero weights in crossbar
  - If below some defined threshold, skip the computation

# Conclusion

- PRIME is the solution to the data movement and high memory bandwidth problem
- Using ReRAM crossbar accelerates NN computation
- Circuit/microarchitectural changes as well as software interface enables wide spectrum of NN applications
- Very little area overhead and extra processing elements

## References



P. Chi, et al. (2016, Sept. 4). *PRIME: A Novel Processing-in-Memory Architecture for Neural Network Computation in ReRAM-Based Main Memory* [Online]. Available: <http://ieeexplore.ieee.org/document/7551380/citations?tabFilter=papers>



A. Shafiee, et al. (2016, Aug. 25). *ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars* [Online]. Available: <http://ieeexplore.ieee.org/document/7551379/>

Thank you and Questions

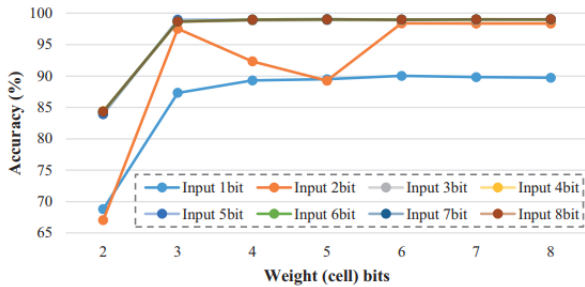


Figure 6. The precision result.