http://inst.eecs.berkeley.edu/~ee290-2/sp21/

Hardware for Machine Learning

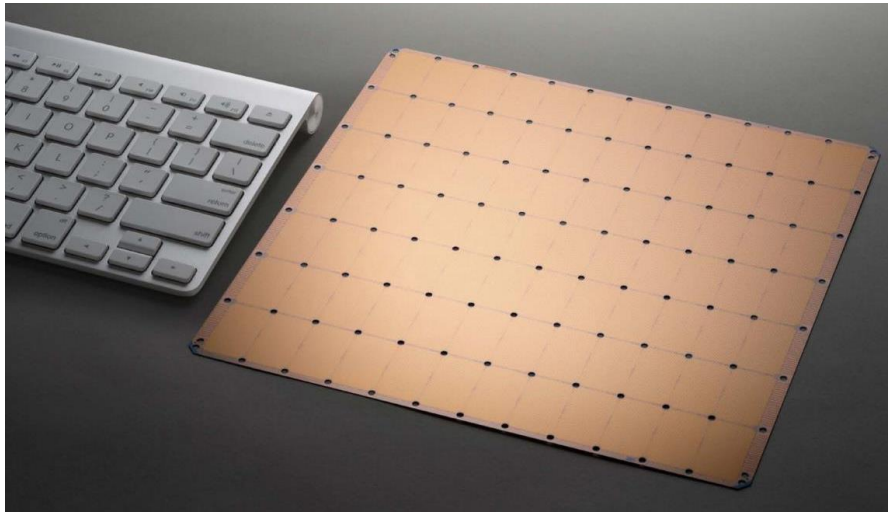
Lecture 22: Advanced Technology

Sophia Shao

Meet Cerebras WSE, The World's Largest Chip

"While our friendly chip giants are bickering over performance increases in the double digits, a startup called Cerebras Systems has gone ahead and shown off a prototype that offers an absolutely unbelievable transistor count increase of 5600 % over the current best available chip: the NVIDIA V100. Bumping transistor count from 21.1 Billion to 2.1 Trillion, the startup has managed to solve key technical challenges that no one else has been able to do and hence make the world's first wafer-scale processor."

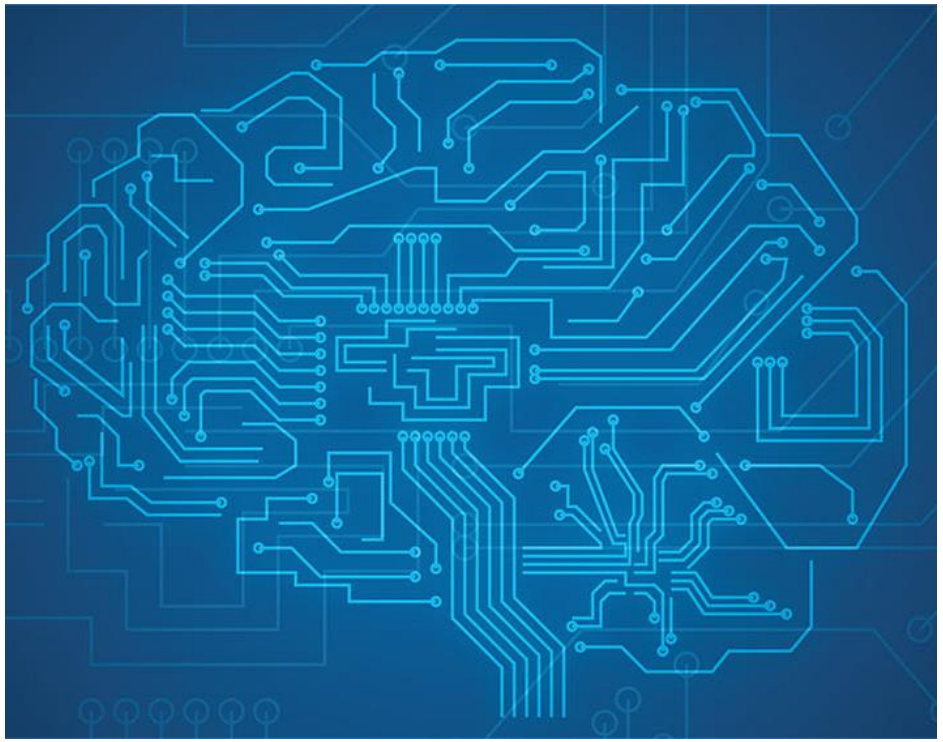
https://wccftech.com/meet-cerebras-wse-the-worlds-largest-chip-at-more-than-56-times-the-size-of-an-nvidia-v100/



Cerebras

Review

- Core computation in DNN
- Execution order of the core computation
- Hardware realization of the core computation
- Mapping DNNs to hardware
- Data transfer mechanisms across storage hierarchy
- Sparsity in DNNs
- Codesign example
- Other Operators and Near-Data Processing
- Training Kernels
- Accelerator-Level Parallelism

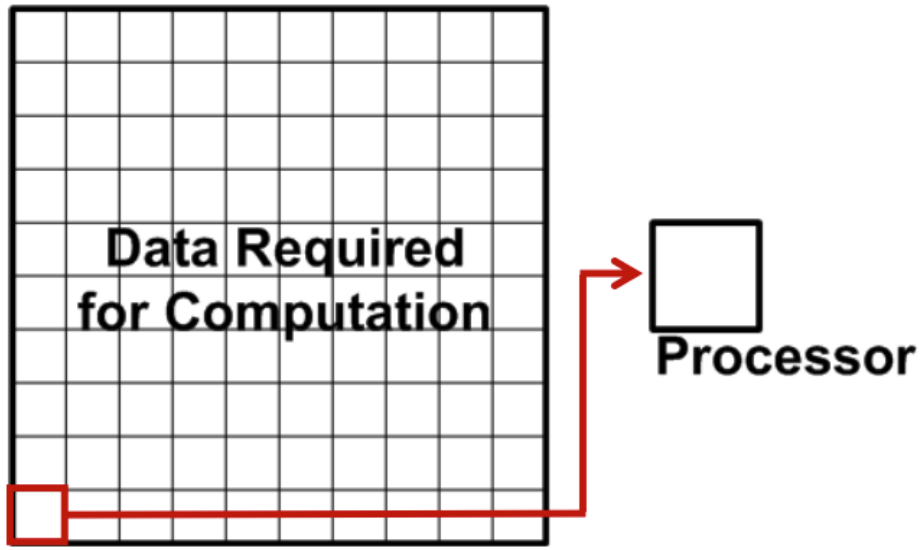


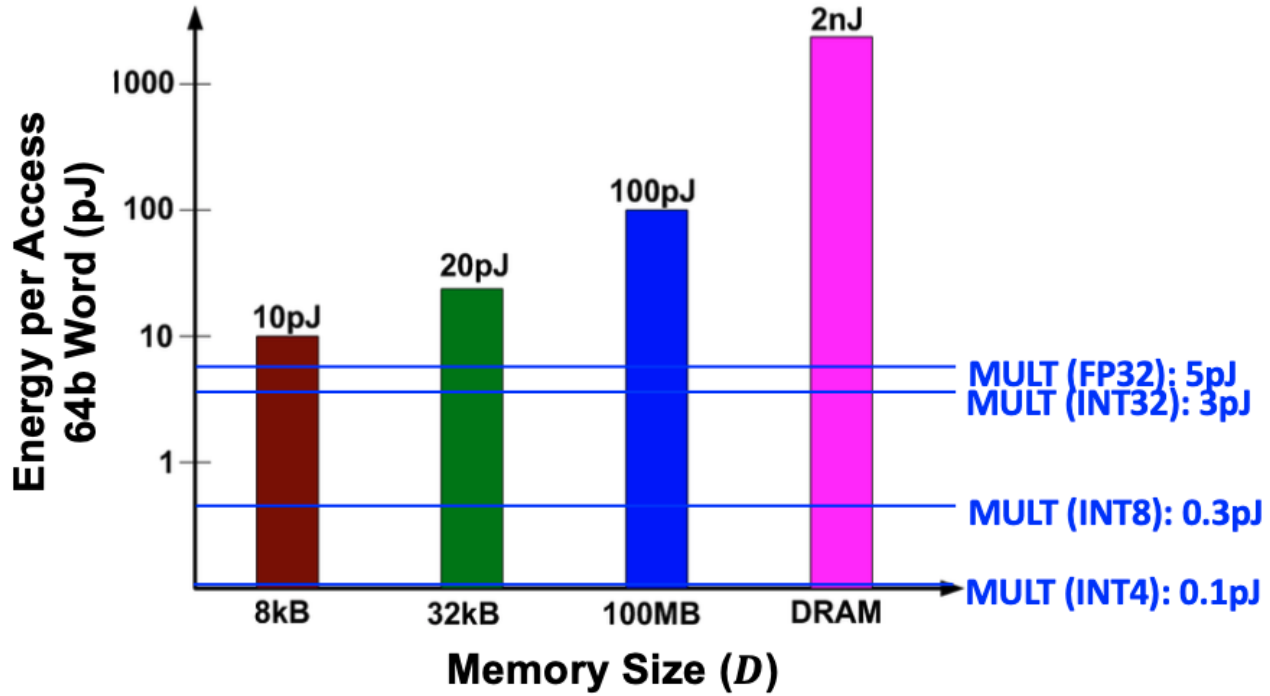
Advanced Technology

- Overview
- In-Memory Computing
  - Non-Volatile Memory
  - SRAM
  - DRAM

Fundamental Cost of Data Movement

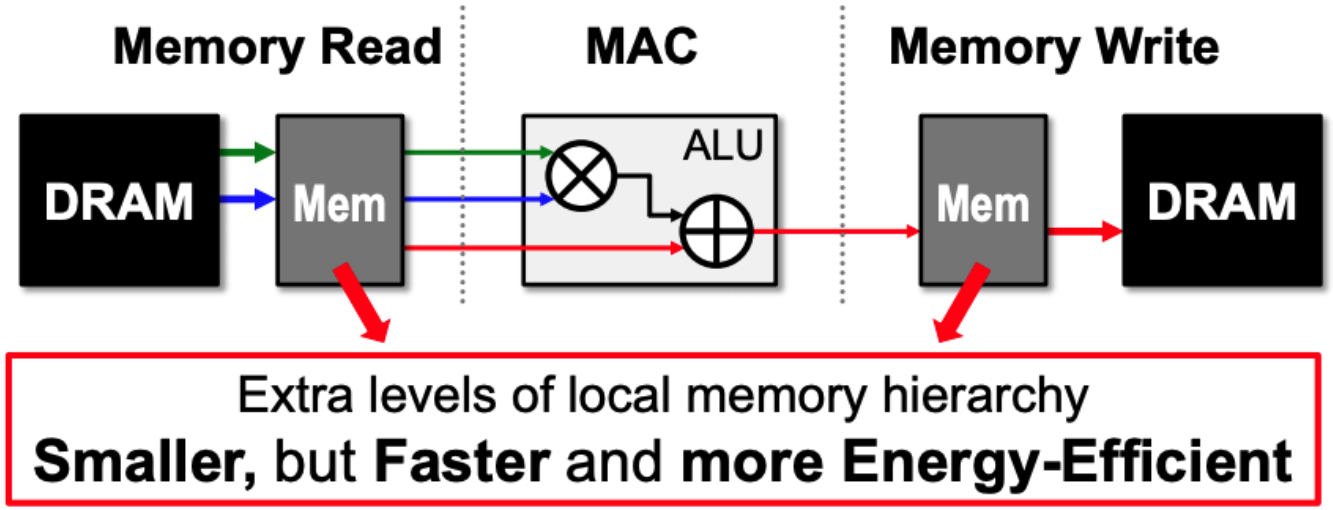
- Overall energy cost is dominated by data movement.





Naveen Verma, Advanced and Prospects for In-Memory Computing

Leveraging Locality in Memory Hierarchy

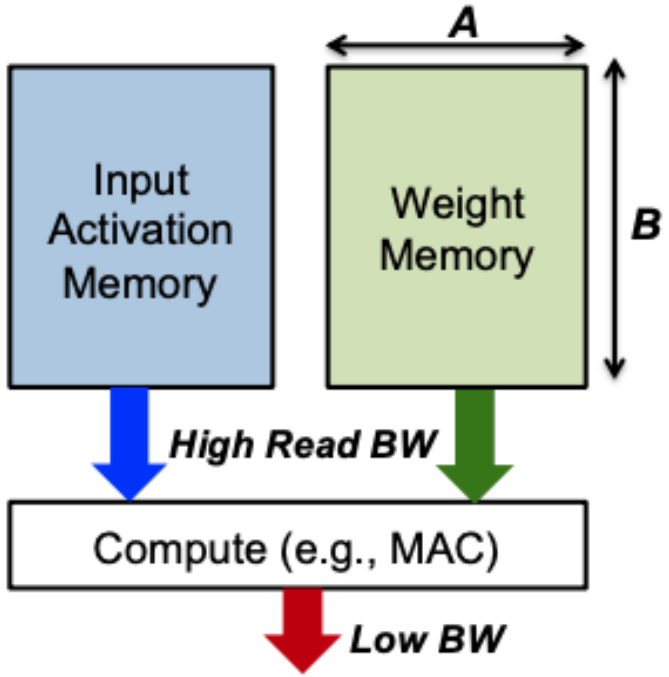


Eyeriss Tutorial

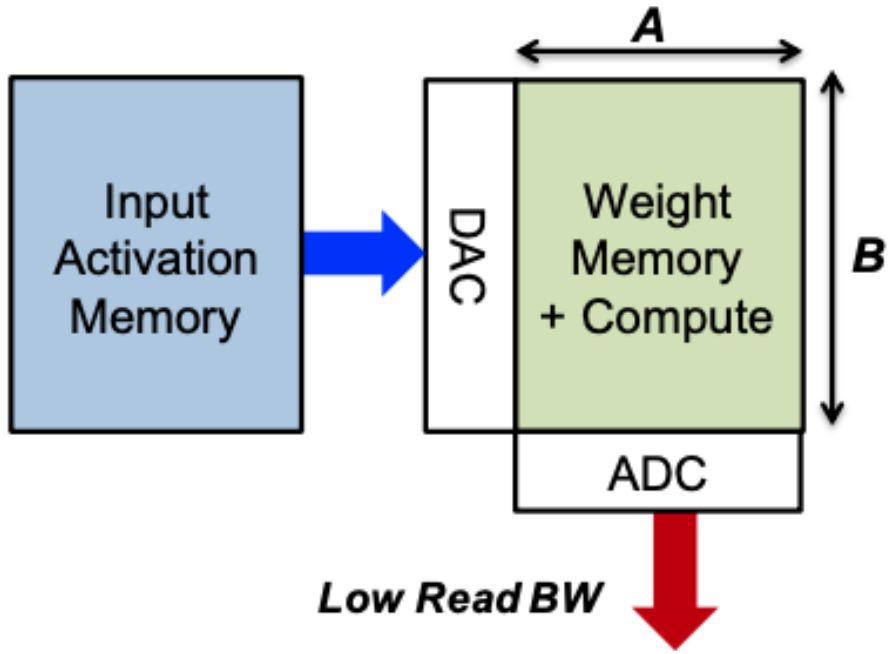
- **Temporal** reuse: the same data is used more than once over time by the same consumer.
- **Spatial** reuse: the same data is used by **more than one consumer** at **different spatial locations** of the hardware.

Processing-in-Memory (PIM) for DNNs

- Brings compute to where weight data is stored.
  - No need to move weights at all



(a) Conventional

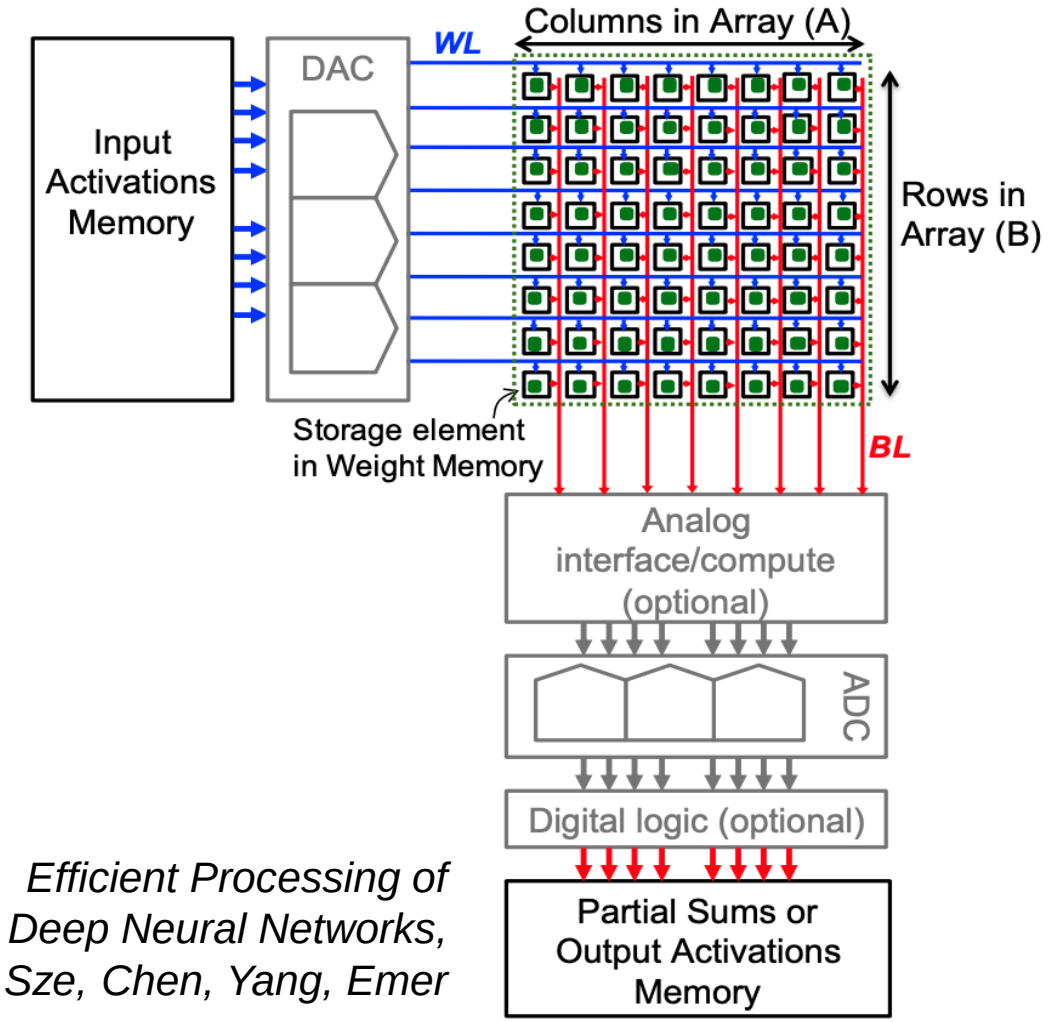


(b) Processing in memory

Efficient Processing of Deep Neural Networks, Sze, Chen, Yang, Emer

Processing-in-Memory for DNNs

- Brings compute to where weight data is stored.
  - Weight-stationary dataflow, i.e, no need to move weights at all
  - Needs to convert input activations and output activations with digital-to-analog converter (DAC) and analog-to-digital converter (ADC)
- In theory, the entire weight matrix (A x B) could be processed in parallel.

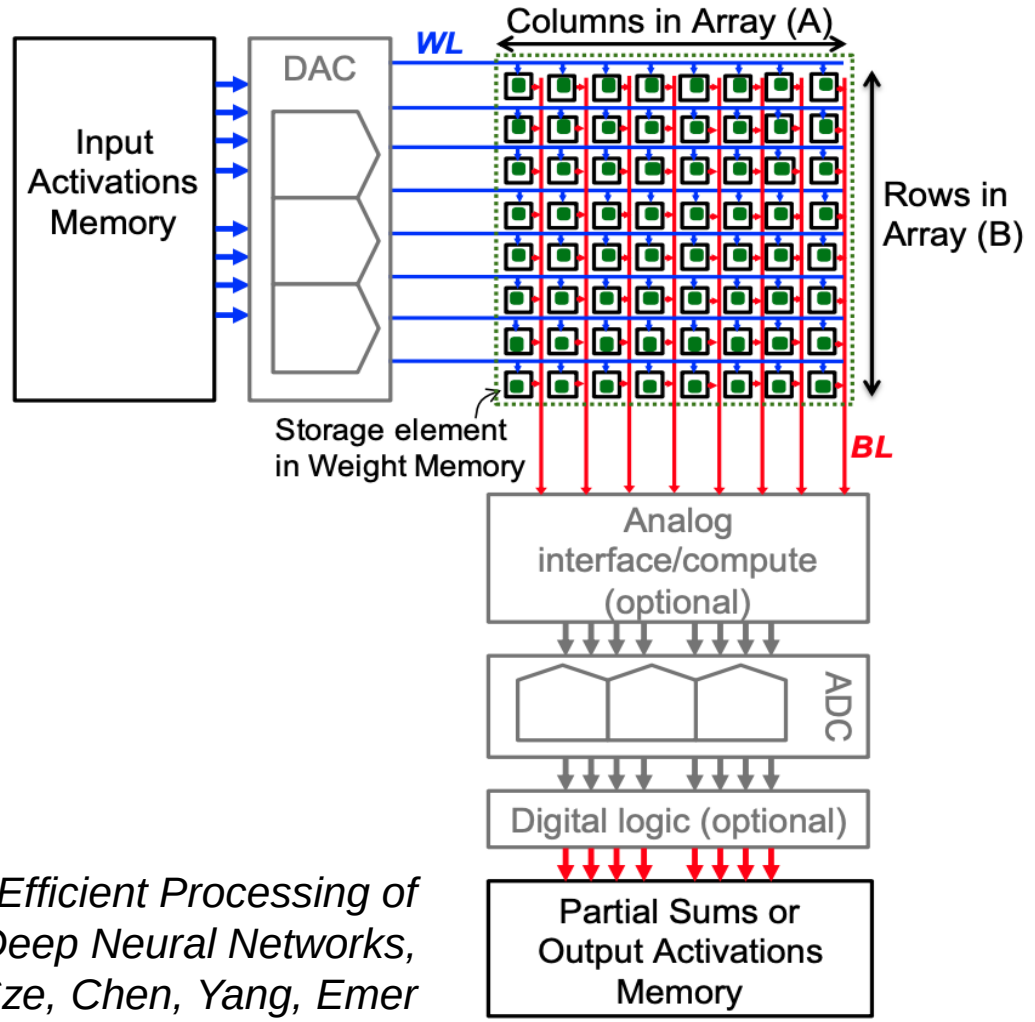


Efficient Processing of Deep Neural Networks, Sze, Chen, Yang, Emer

Processing-in-Memory for DNNs

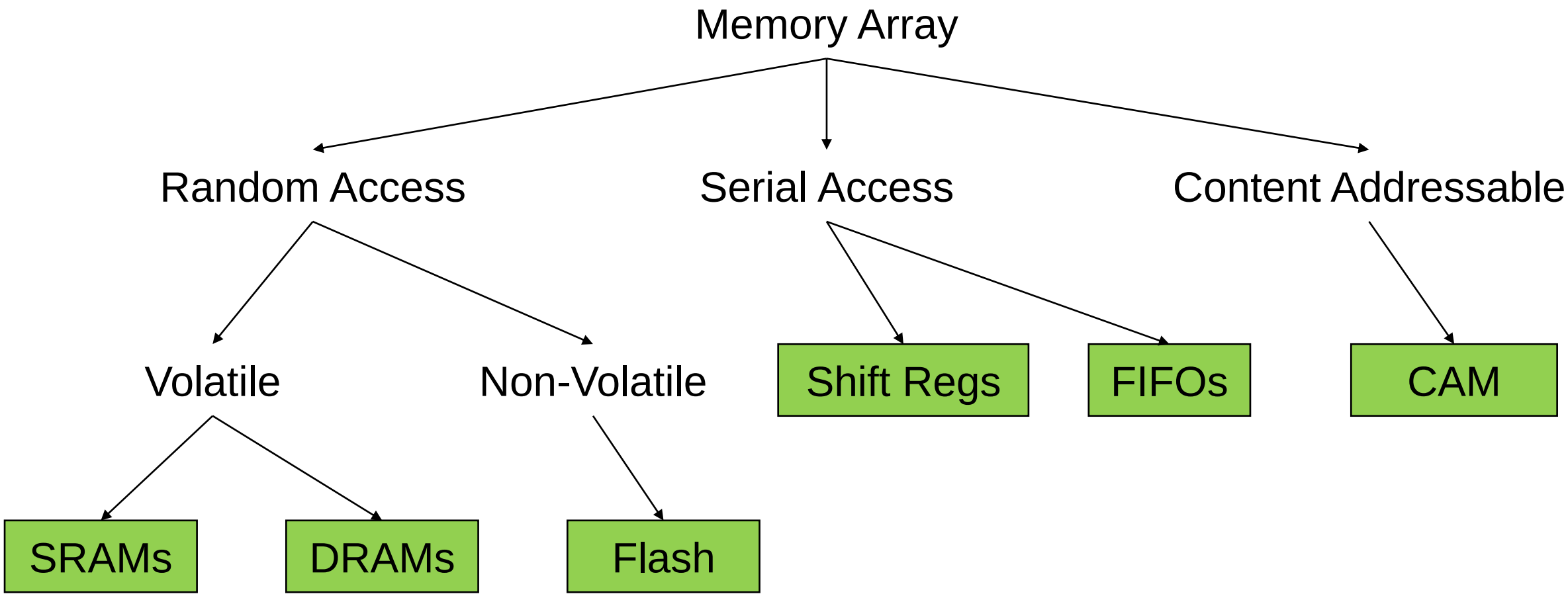
In a typical processing-in-memory DNN accelerator,

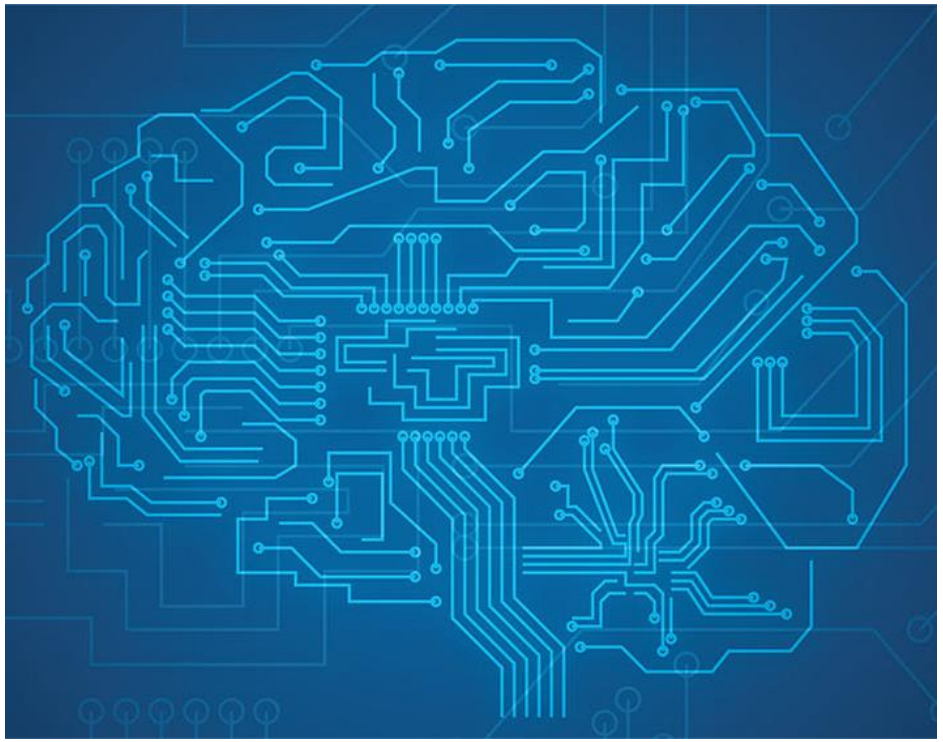
- Input activation:
  - Delivered by word lines.
- Weight:
  - Stored in the storage element, technology dependent.
- Output activation/Partial sum:
  - Accumulated with bit lines.



Efficient Processing of Deep Neural Networks, Sze, Chen, Yang, Emer

Memory Overview



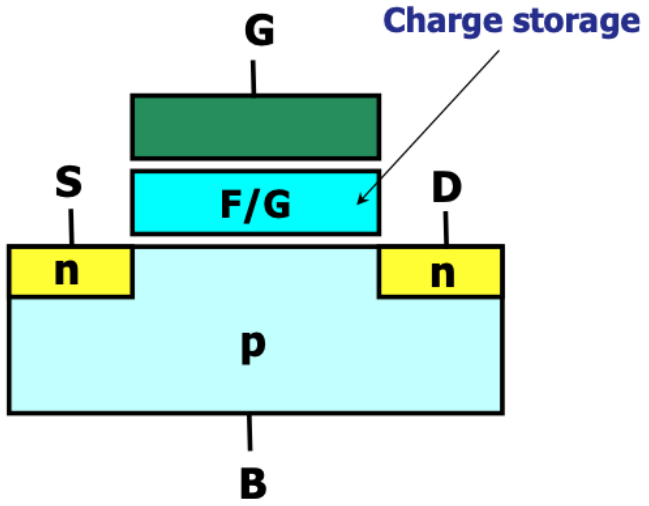


Advanced Technology

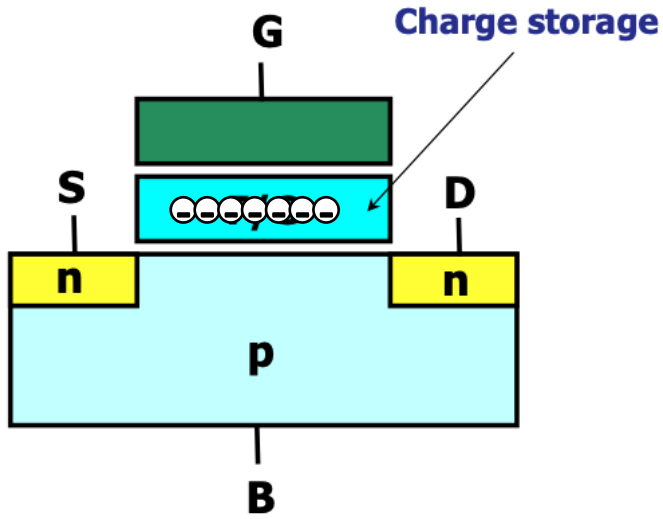
- Overview
- In-Memory Computing
  - Non-Volatile Memory
  - SRAM
  - DRAM

Flash Transistor

- Key concept:
  - Floating Gate: A charge storage layer -> memorize information
- A “Programmable-Threshold” Transistor



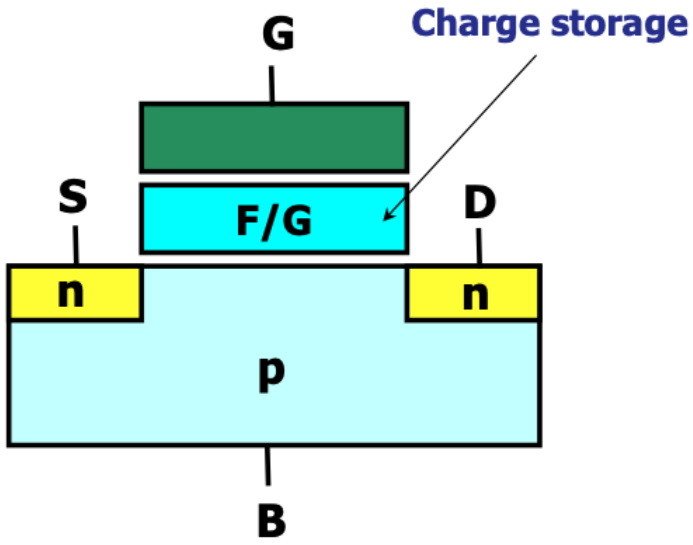
- Storing 1
- No charge in floating gate



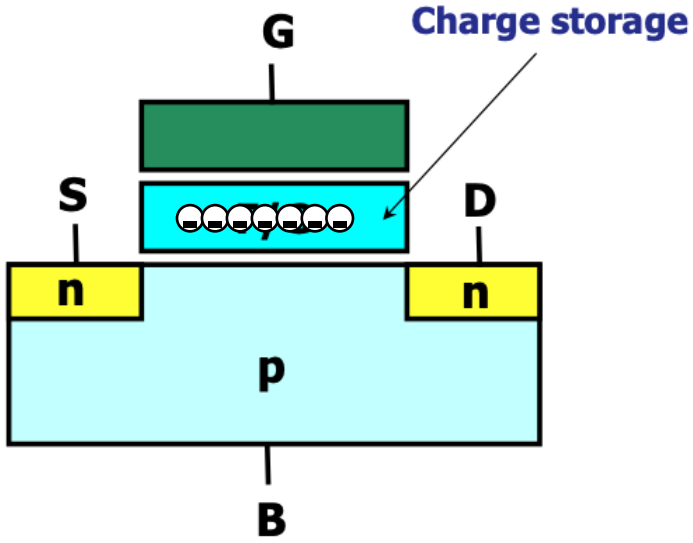
- Storing 0
- Negative charge in floating gate

Flash Transistor

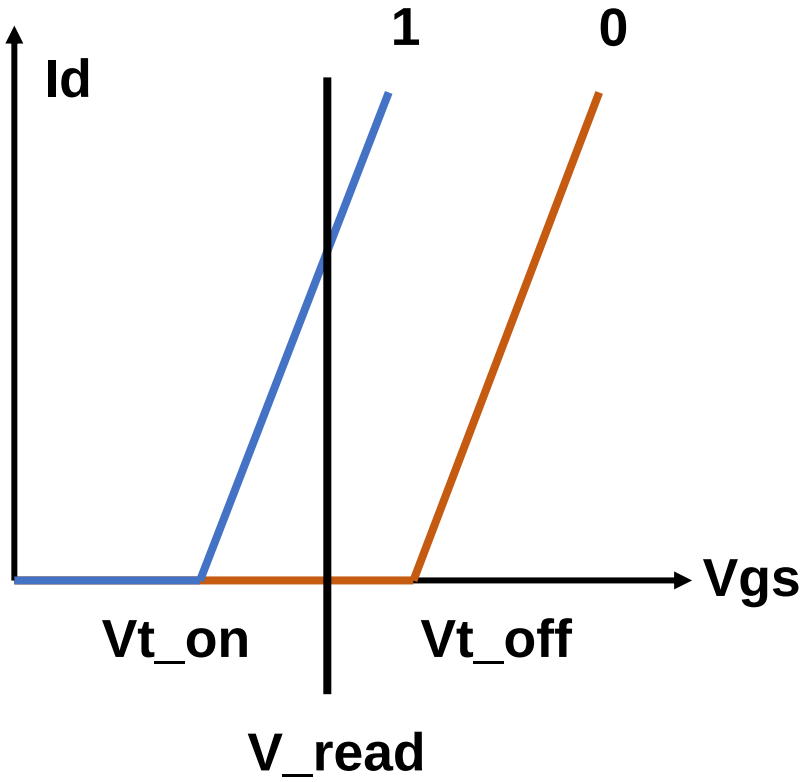
- Floating gate change the threshold voltage of a cell
- Read the cell value by sensing the current



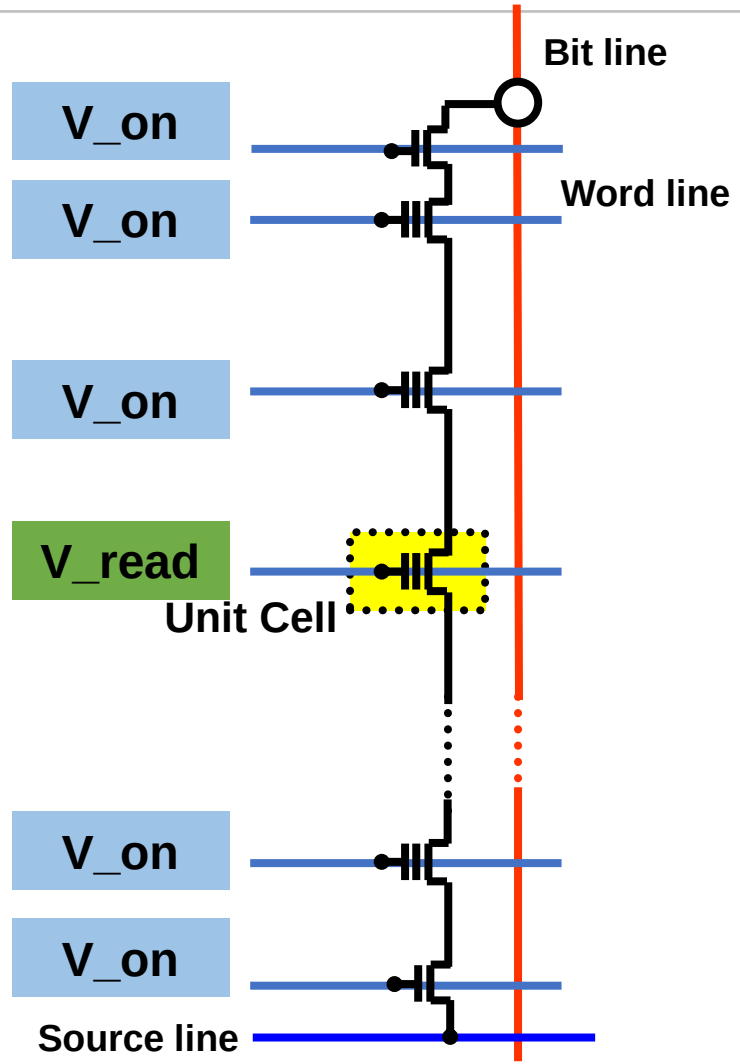
Storing 1
ON State

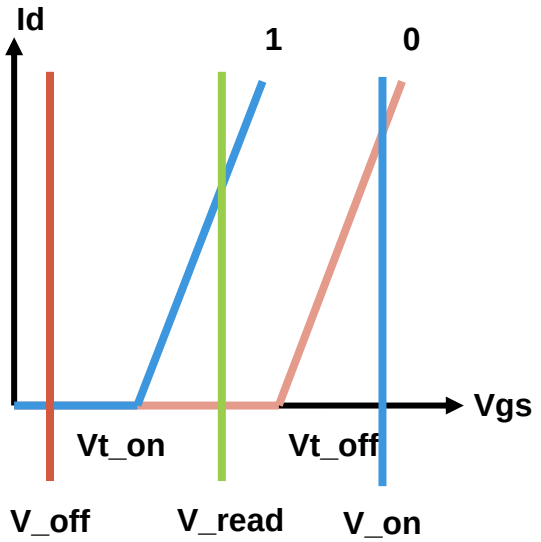


Storing 0
OFF State



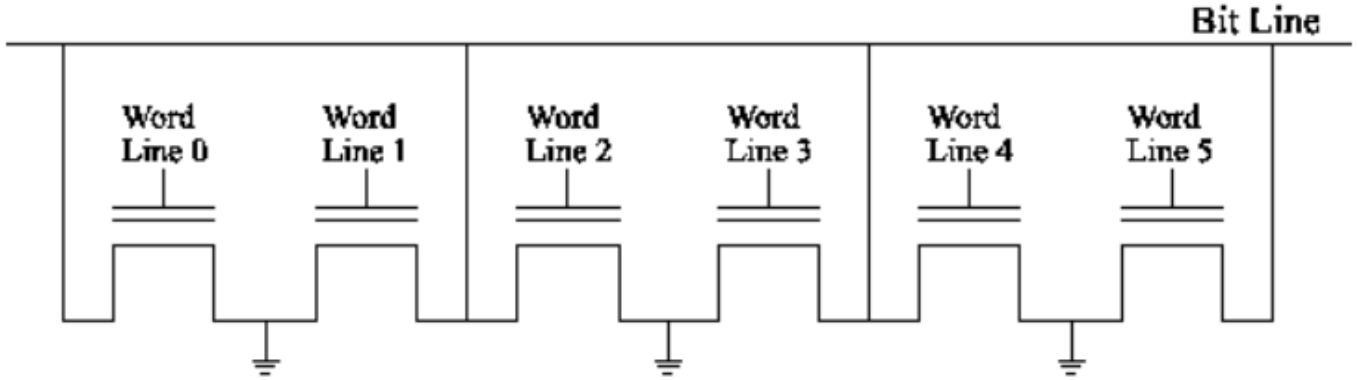
NAND Flash Read



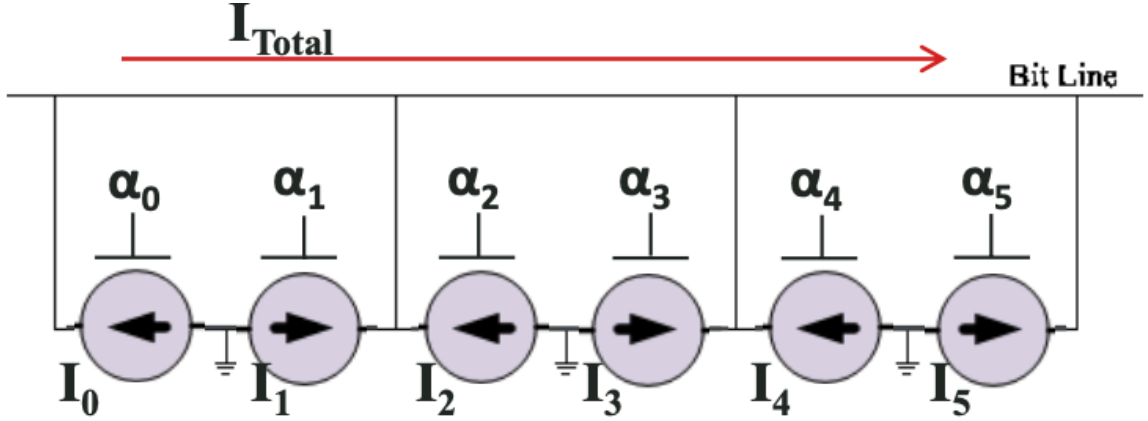


Flash Cell -> Multiply-accumulate function

$$I_n = \alpha_n \times I'_n = \alpha_n \times (V_{GS} - V_{th_n})$$
$$I_{Total} = I_0 + I_1 + I_2 + I_3 + I_4 + I_5$$



- Each flash cell acts as a gated current source (multiplication)
- Flash cells on the same bit line adds current (accumulation)



Mythic

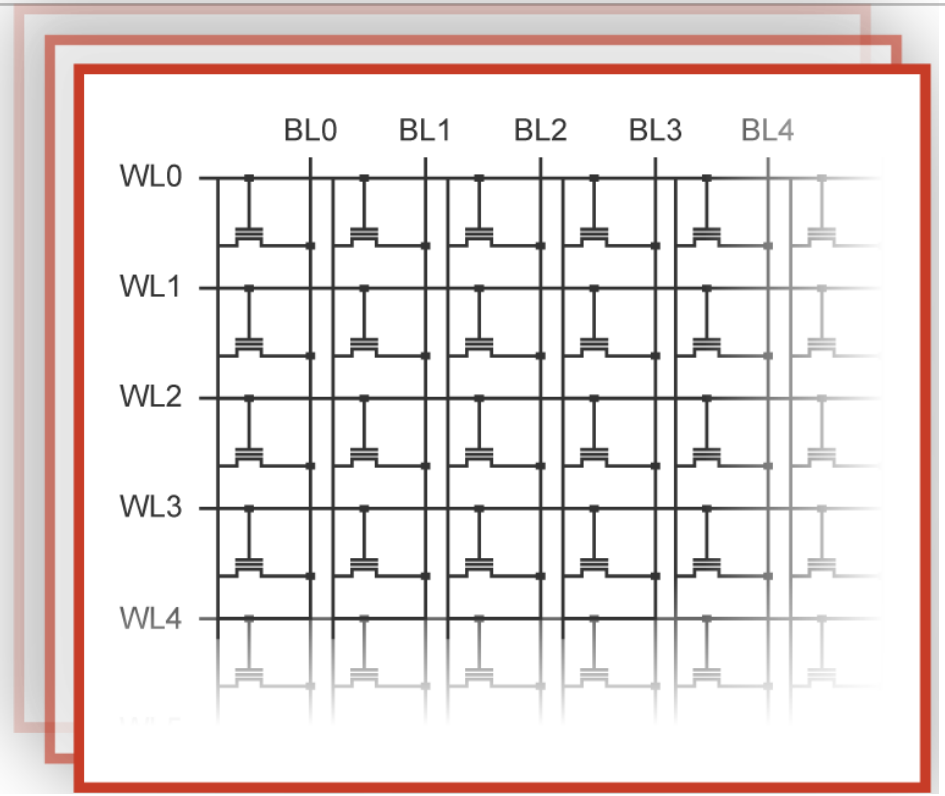
Example: Mythic

	Enterprise With DRAM	Enterprise No-DRAM	Edge With DRAM	Edge No-DRAM	Mythic NVM
SRAM	<50 MB	100+ MB	< 5 MB	< 5 MB	< 5 MB
DRAM	8+ GB	-	4-8 GB	-	-
Power	70+ W	70+ W	3-5 W	1-3 W	1-5 W
Sparsity	Light	Light	Moderate	Heavy	None
Precision	32f / 16f / 8i	32f / 16f / 8i	8i	1-8i	1-8i
Accuracy	Great	Great	Moderate	Poor	Great
Performance	High	High	Very Low	Very Low	High
Efficiency	25 pJ/MAC	2 pJ/MAC	10 pJ/MAC	5 pJ/MAC	0.5 pJ/MAC

Mythic

Example: Mythic

- Mythic introduces the ***Matrix Multiplying Memory***
  - Never read weights
- This effectively makes weight memory access ***energy-free*** (only pay for MAC)
- And eliminates the need for…
  - Batch > 1
  - CNN Focus
  - Sparsity or Compression
  - Nerfed DNN Models



Made possible with Mixed-Signal Computing on embedded flash *Mythic*

Challenges and Opportunities

- Challenges:
  - Programming weights, i.e., writing to NVM
    - Latency
    - Endurance (e.g. program once per day)
    - What if the size of weights is larger the the memory size?
  - Precision, i.e., # of bits that can be stored per device (< 8bit)
- Opportunities:
  - Emerging NVM devices
    - Phase Change RAM (PCRAM), Resistive RAM (RRAM), STT-MRAM
    - With different trade-offs in endurance, density, variations, and speed.

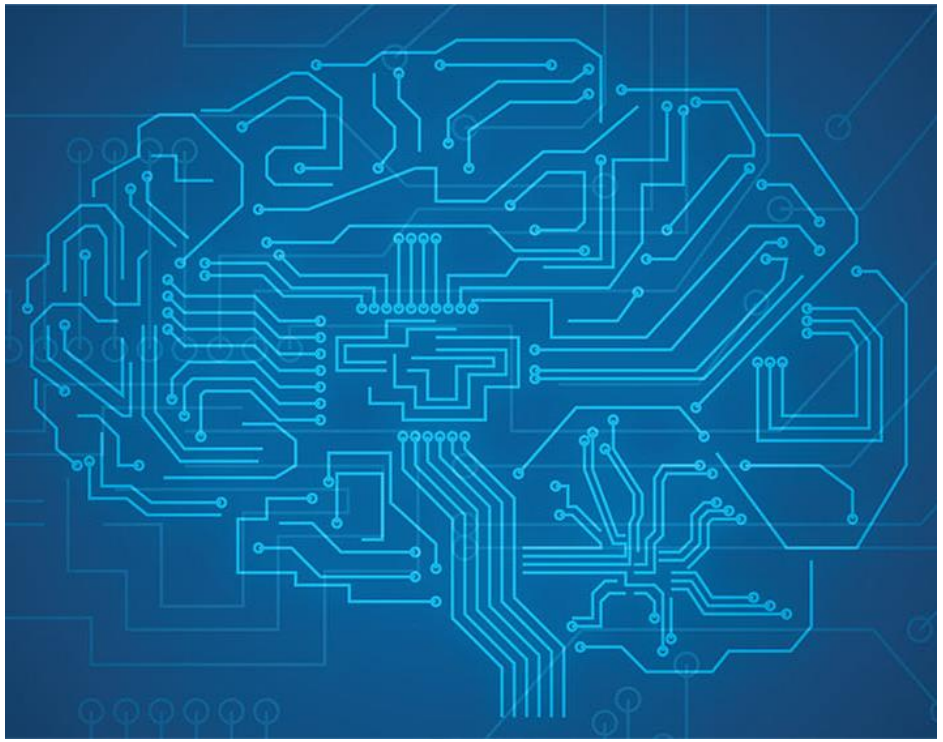
Administrivia

- Last guest lecture coming up:
  - 4/19, Brian Zimmer, NVIDIA
    - Problems Facing Analog and In-Memory Computing

- Project (50%):
  - Proposal (10%)
  - Checkpoint * 2 (5% * 2. Required.)
  - Presentation (10%)
  - Final report (15%)
  - Results (5%)

Administrivia

- Project Checkpoint 1 done.
  - Good job all.
- Project Checkpoint 2:
  - Sign up here:
    - https://docs.google.com/spreadsheets/d/1Mer3sGy5jVTP2KKykFJ_QABfyE9nEy44-fW1diAVCCU/edit?usp=sharing
    - Please prepare 2 or 3 slides showing your current progress.
  - The slides should include:
    - Initial characterization/performance/synthesis results

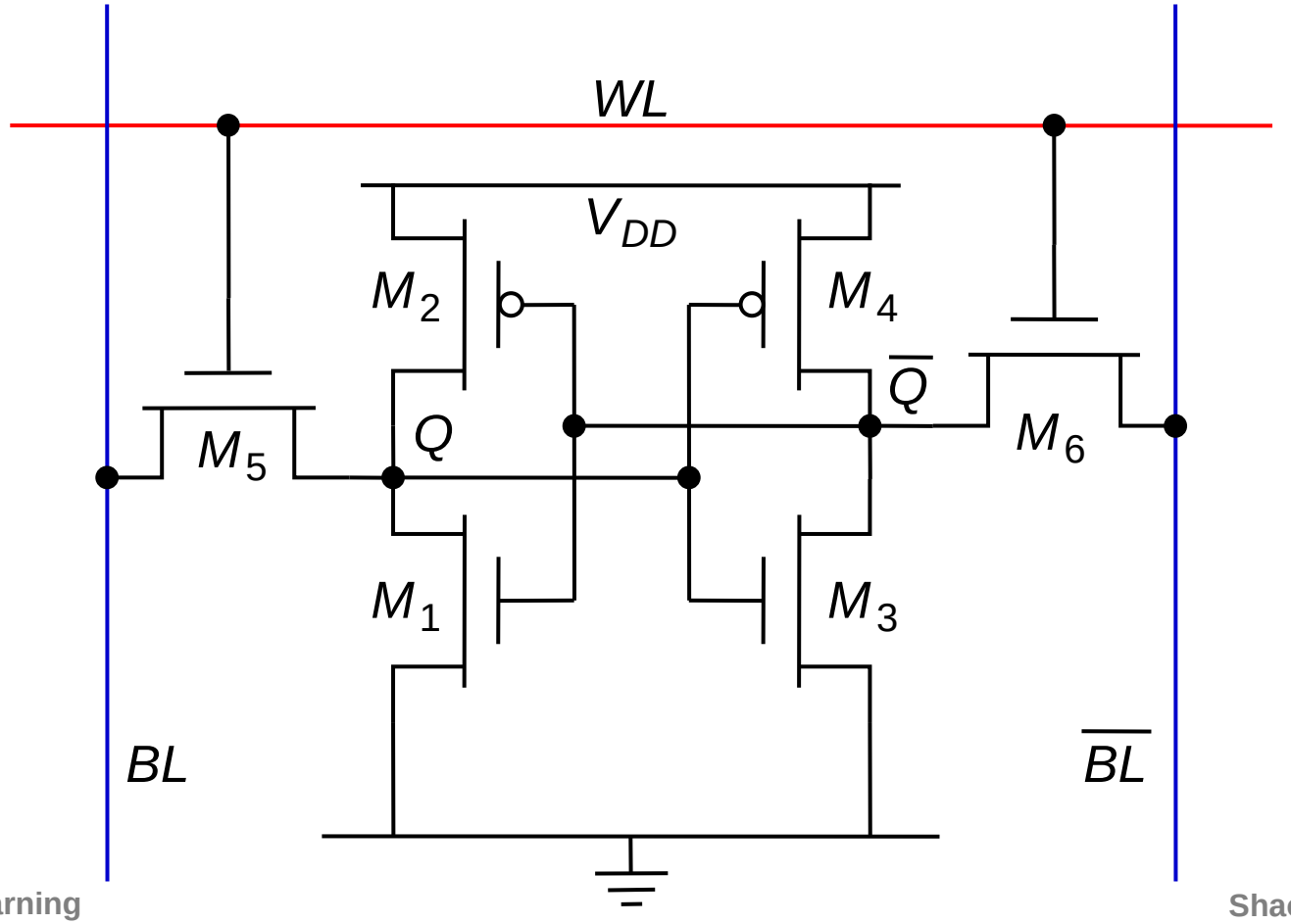


Advanced Technology

- Overview
- In-Memory Computing
  - Non-Volatile Memory
  - SRAM
  - DRAM

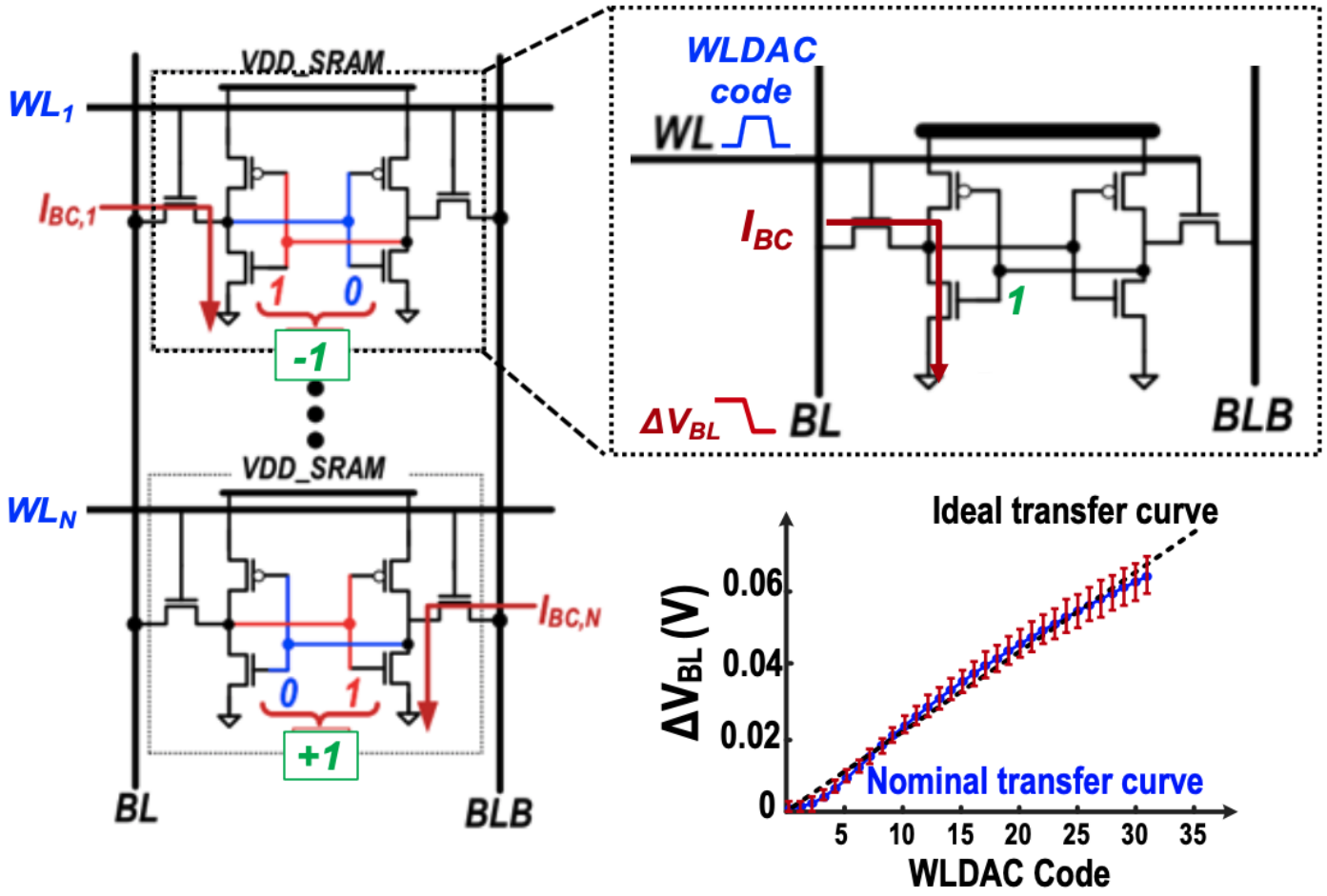
6T CMOS SRAM Cell

- Wordline (WL) enables read/write access for a row
- Data is written/read differentially through shared BL, ~BL



SRAM Cell -> Multiply-accumulate function

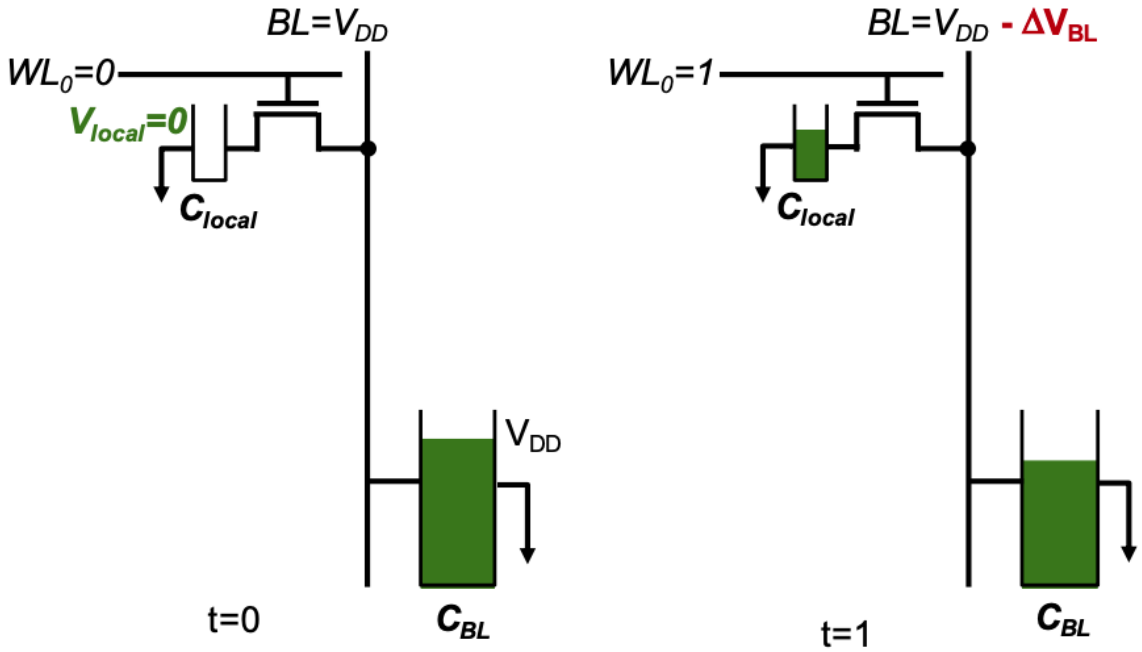
- Input activation is encoded as a voltage amplitude on the word line.
- Accumulation is done on the bit line.



A machine-learning classifier implemented in a standard 6T SRAM array, VLSI'2016

Challenges

- Array scalability
  - Ideally, a large array size is preferred for better reuse and area efficiency.
  - However, larger array -> longer bitline -> larger bit line capacitance

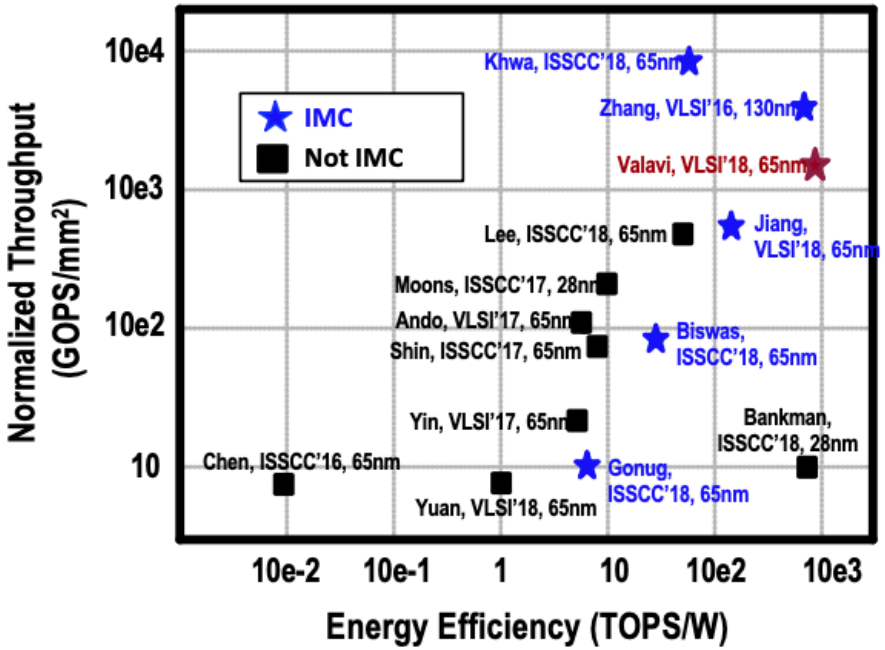


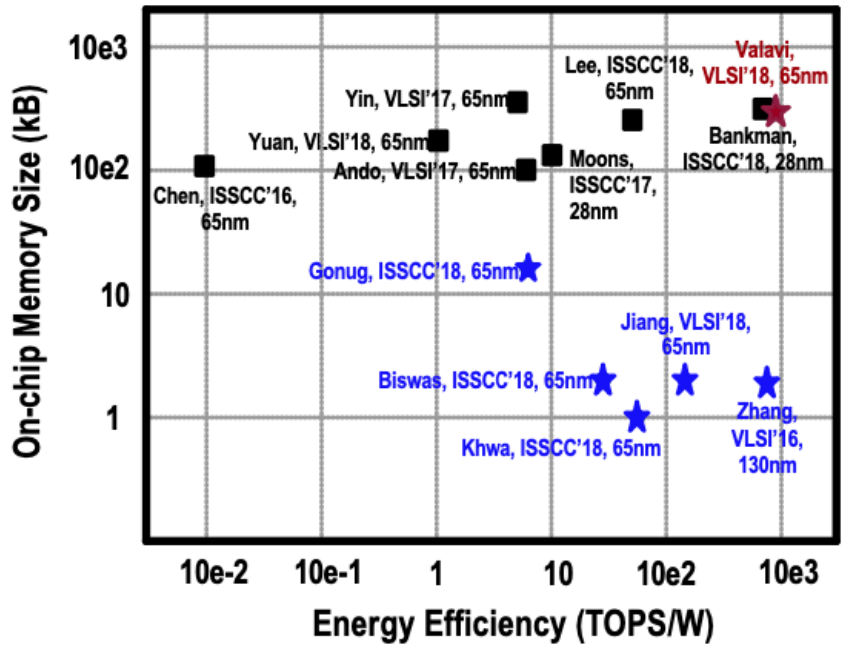
$$\Delta V_{BL} = (V_{DD} - V_{local}) \frac{C_{local}}{C_{local} + C_{BL}}$$

Efficient Processing of Deep Neural Networks, Sze, Chen, Yang, Emer

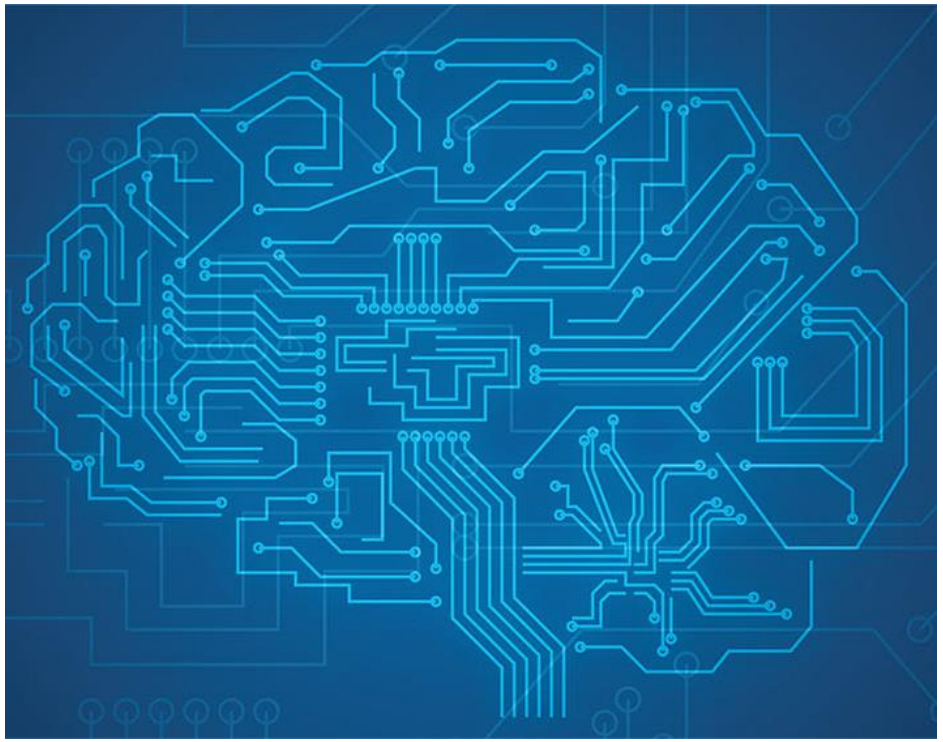
Challenges

- Array scalability
  - Ideally, a large array size is preferred for better reuse and area efficiency.
  - However, larger array -> longer bitline -> larger bit line capacitance





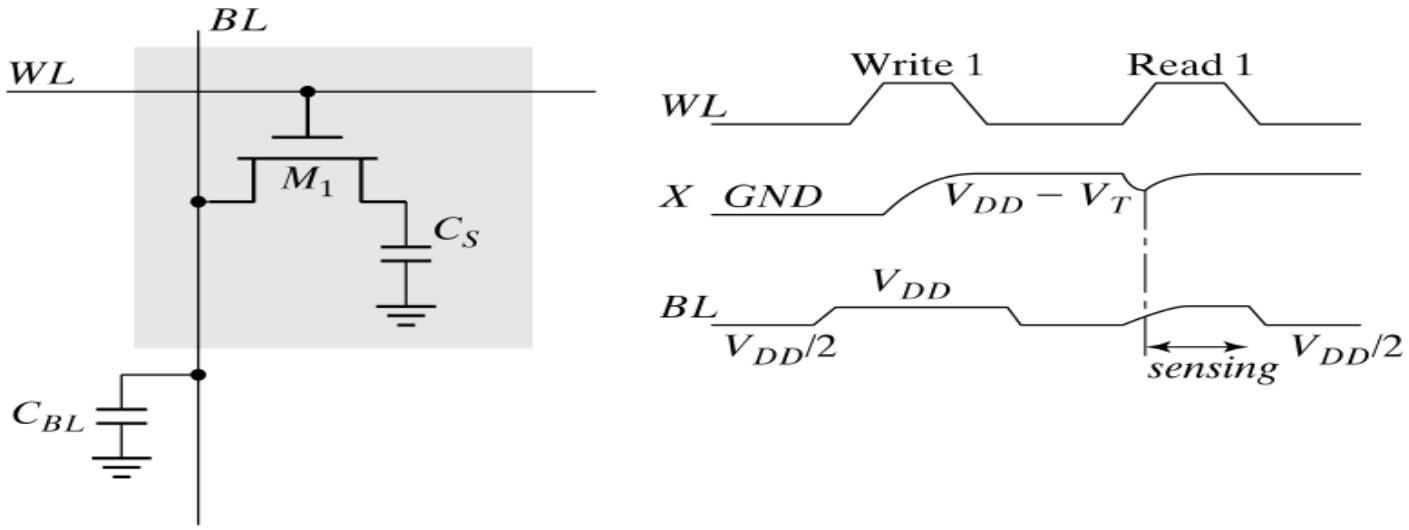
Naveen Verma, Advanced and Prospects for In-Memory Computing



Advanced Technology

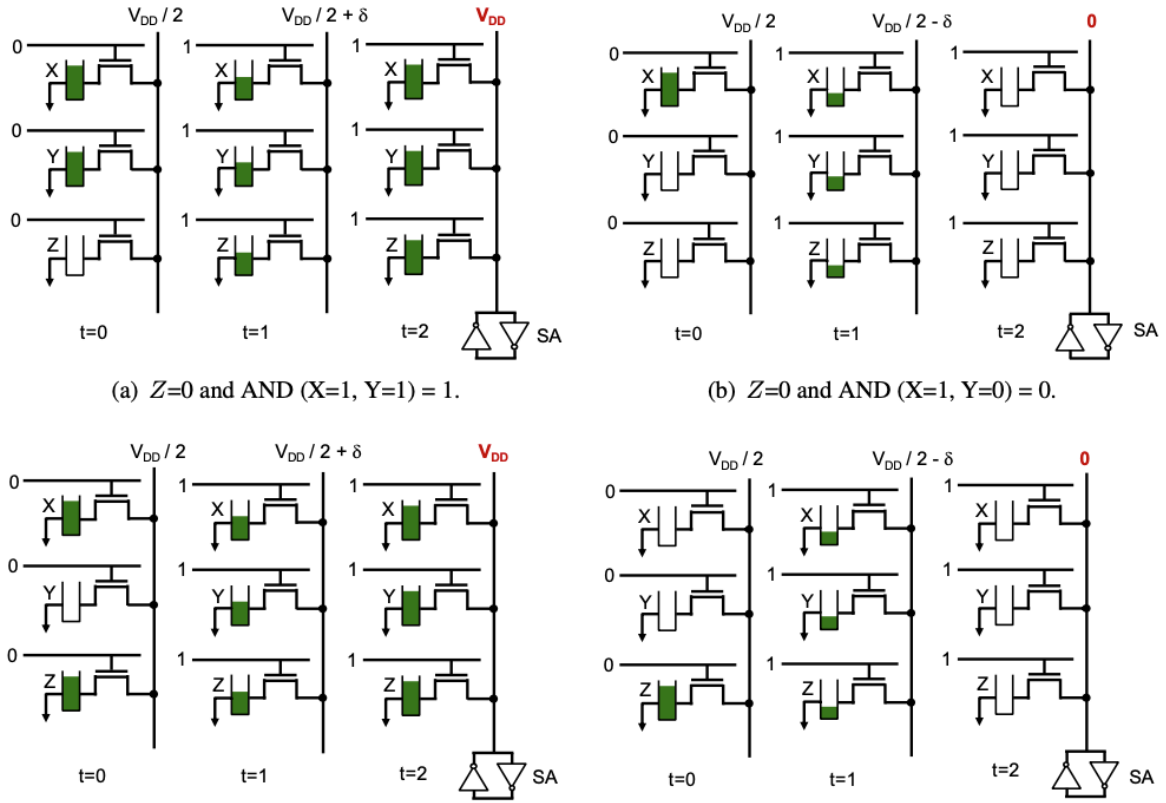
- Overview
- In-Memory Computing
  - Non-Volatile Memory
  - SRAM
  - DRAM

DRAM Cell


BL
WL
M_1
C_S
C_{BL}
Write 1
Read 1
WL
X
GND
$V_{DD} - V_T$
BL
V_{DD}
$V_{DD}/2$
sensing
$V_{DD}/2$

DRAM Cell -> AND/OR Operation

- Needs to access 3 rows in parallel.
- Use bit-wise operation to MAC across multiple cycles.



(a) Z=0 and AND (X=1, Y=1) = 1.

(b) Z=0 and AND (X=1, Y=0) = 0.

(c) Z=1 and OR (X=1, Y=0) = 1.

(d) Z=1 and OR (X=0, Y=0) = 0.

Efficient Processing of Deep Neural Networks, Sze, Chen, Yang, Emer

Review

- Core computation in DNN
- Execution order of the core computation
- Hardware realization of the core computation
- Mapping DNNs to hardware
- Data transfer mechanisms across storage hierarchy
- Sparsity in DNNs
- Codesign example
- Other Operators and Near-Data Processing
- Training Kernels
- Advanced Technology
  - In-Memory Computing
    - Flash-based
    - SRAM-based
    - DRAM-based