Memory Technologies

Process in Memory

Lecturer : Mehrdad Arghand

Memory Wall

Ways To Defeat

- Memory interleaving
- Memory partitioning
- Multithreading
- Memory partitioning
- Cache optimization
- Use of on-chip memory
- Memory bypassing
- Memory hierarchies
- Near Memory Processing

Near Memory Processing Vs Process in Memory

Computer systems: Trends and opportunity

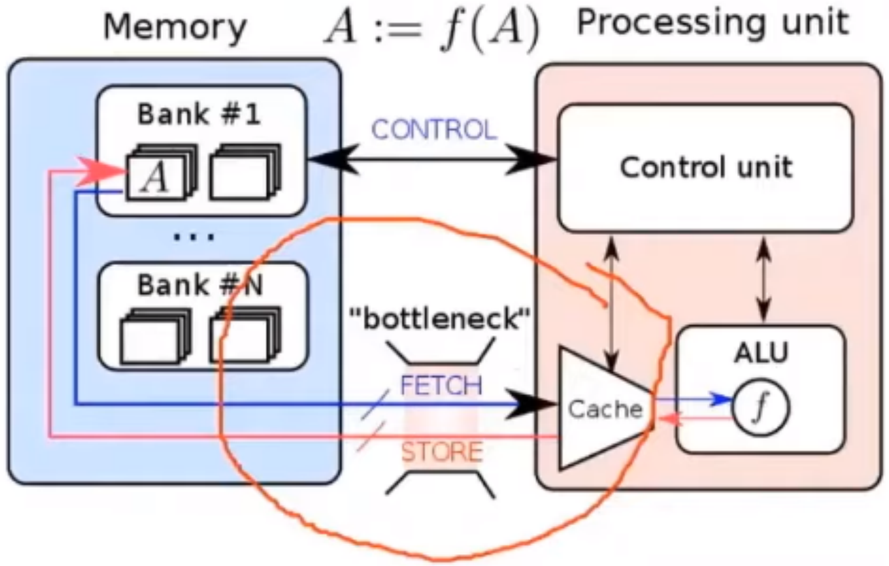
- Three key trends
  - ✓Data access is a major bottleneck
  - ✓Energy consumption is a key limiter
  - ✓Energy to move data dominates compute energy

- Opportunity
  - ✓Minimize data movement by performing computation directly (near) where the data resides
  - ✓Processing in memory (PIM)
    - In-memory computing
    - Near-memory computing/near data processing

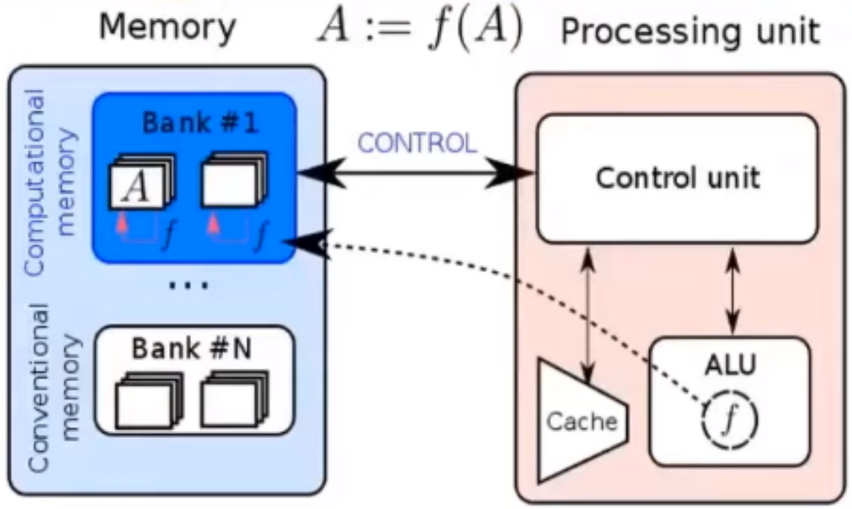
Mutlu et al., Microprocess. Microsyst. (2019)

In-memory computing

Processing unit & Conventional memory



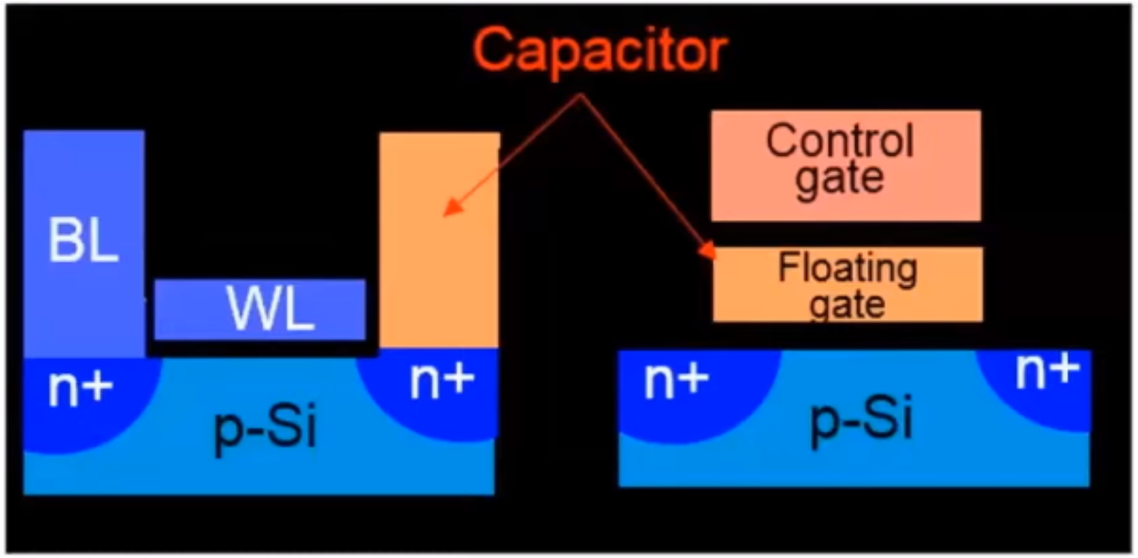
Processing unit & Computational memory



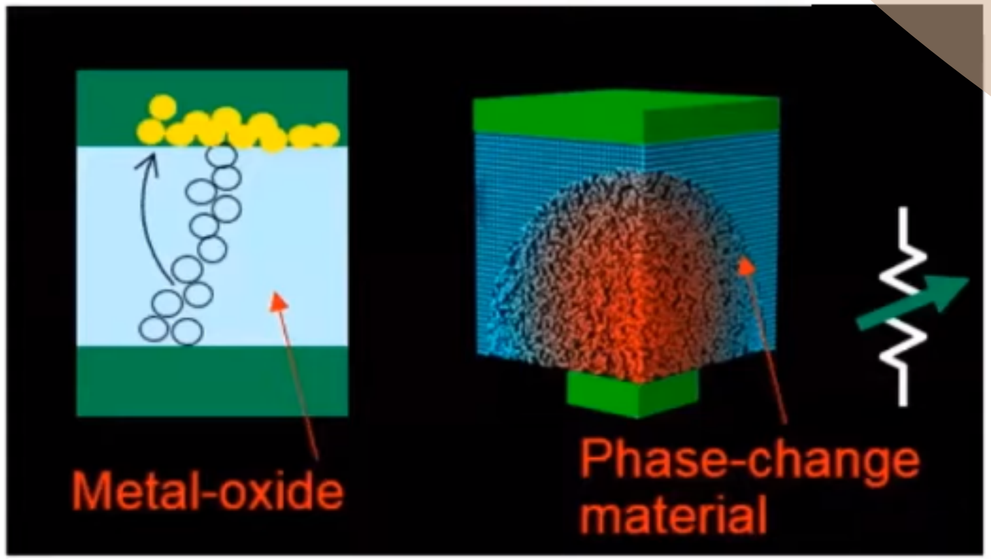
- Perform “certain” computational tasks **in place in memory**
- Achieved by exploiting **the physical attributes of the memory devices**, their **array level organization**, the **peripheral circuitry** as well as the **control logic**
- At **no point during computation**, the memory content is read back and **processed at the granularity of a single memory element**

Constituent elements

Charge-based memory

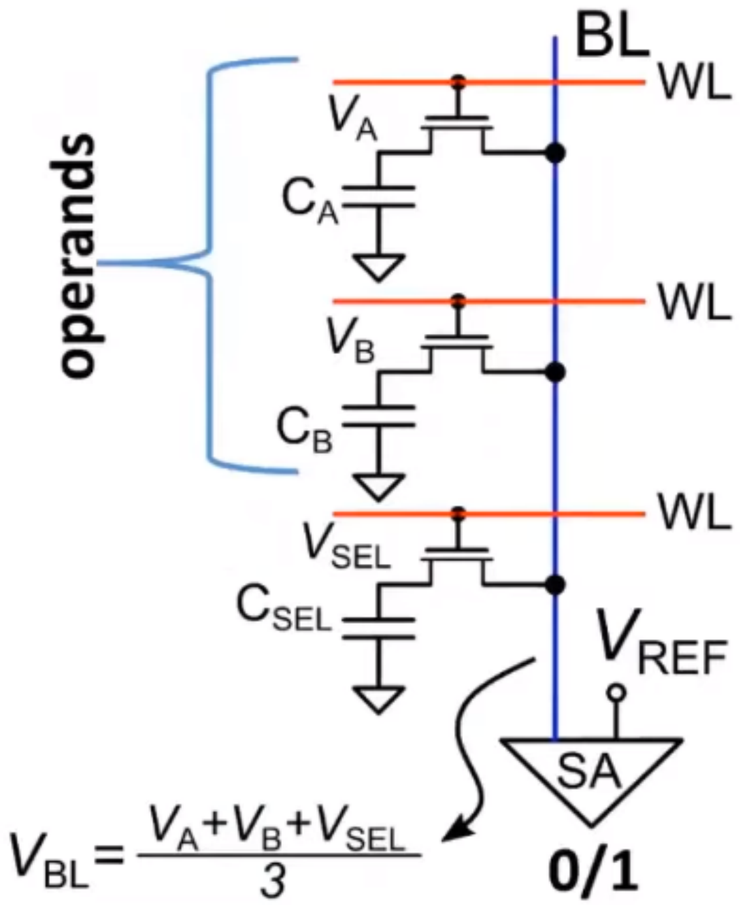


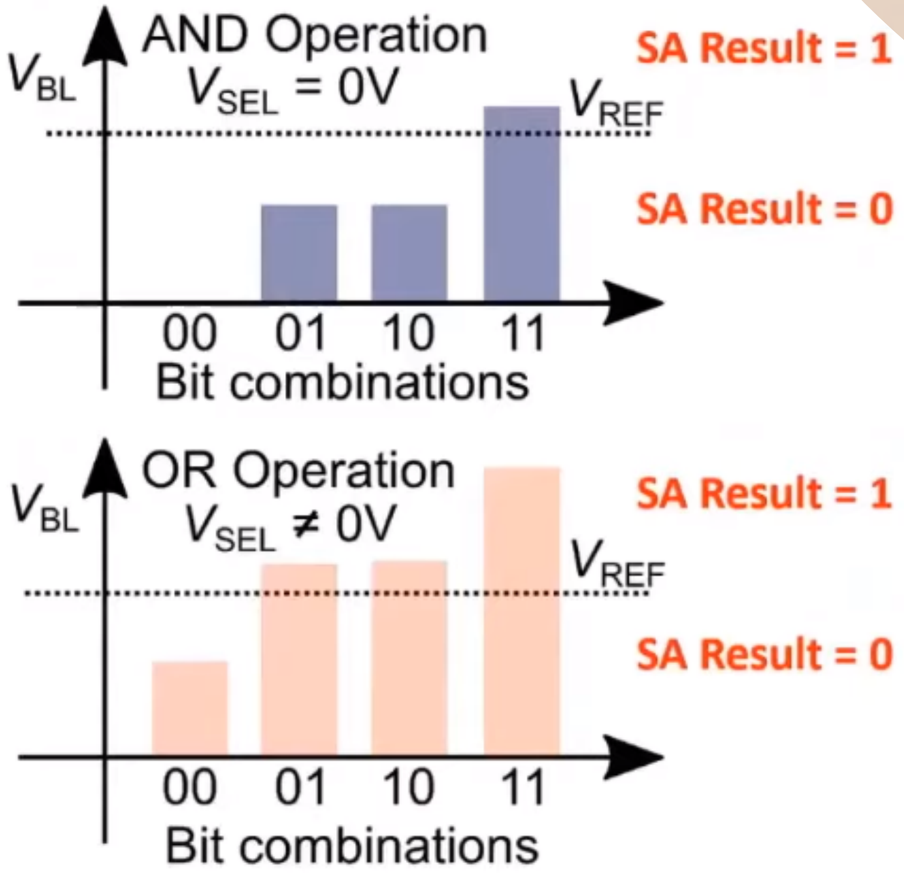
Resistance-based memory



- **Charge-based memory:** Presence or absence of charge (eg. DRAM, SRAM, Flash)
- **Resistance-based memory:** Differences in atomic arrangements or orientation of ferromagnetic metal layers (eg. PCM, metal-oxide RRAM, STT-MRAM)

Logical operations using DRAM

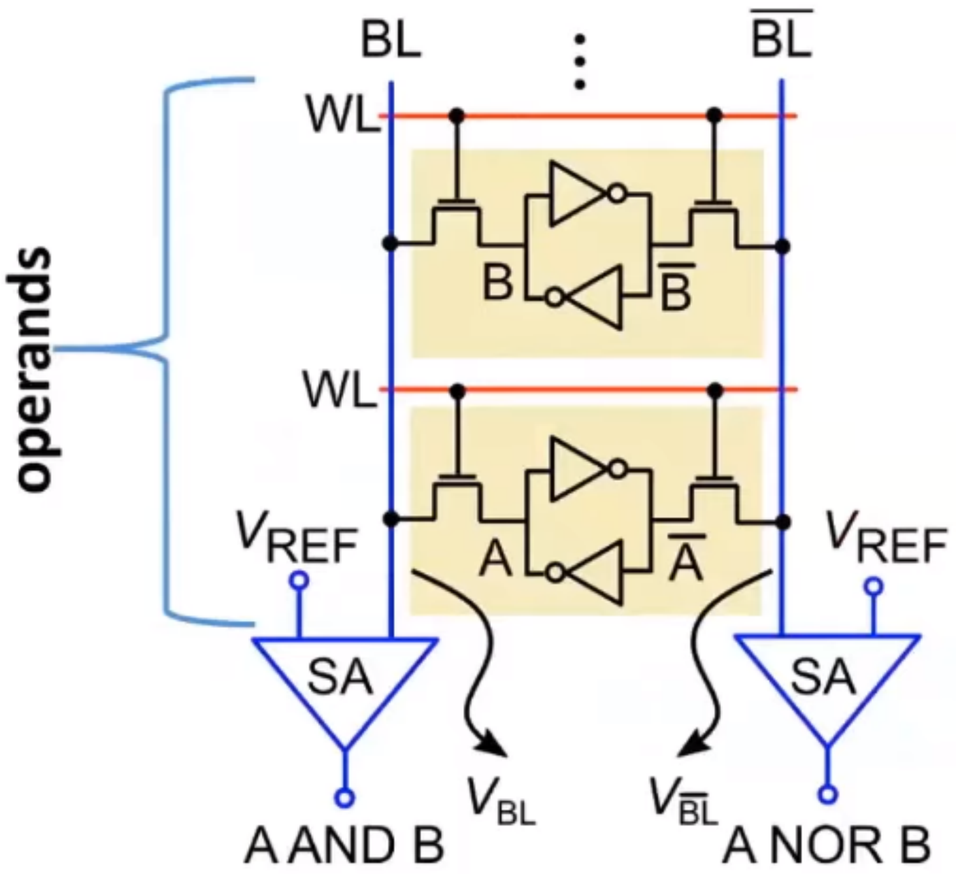


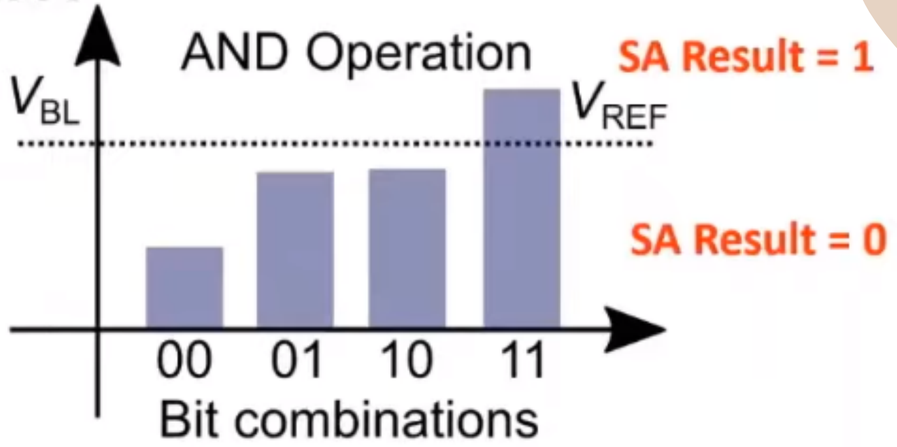


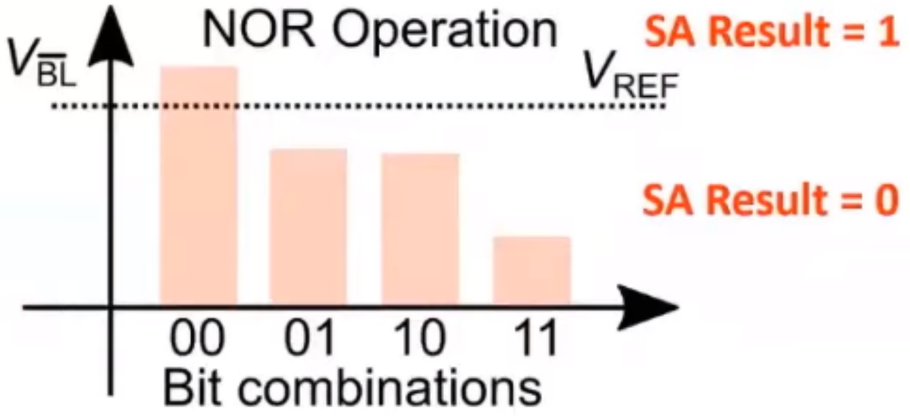
- Bitwise logical operations performed by **simultaneously activating WLs**
- Operands in cells A and B, **SEL is used to dictate whether AND or OR is realized**

Sheshadri et al., MICRO (2017), Li et al., MICRO (2017)

Logical operations using SRAM





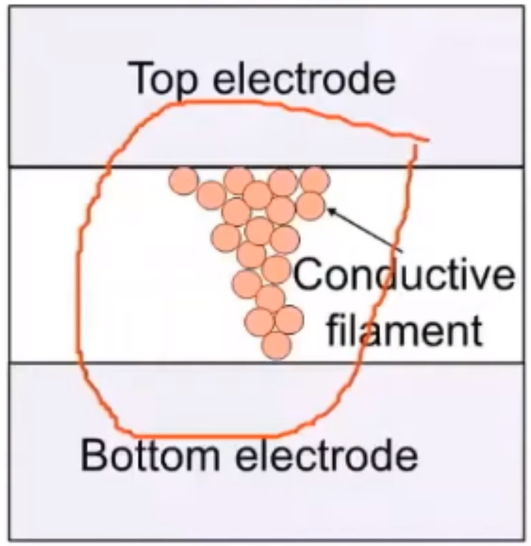


- BL and $\overline{BL}$ are **pre-charged to the supply voltage**
- **Both the WLs are activated** so that both BL and $\overline{BL}$ are discharged at different rates that depend on the data stored in the bit-cells

Aga et al., HPCA (2017), Jeloka et al., JSSC (2016)

Resistance-based memory devices

ReRAM

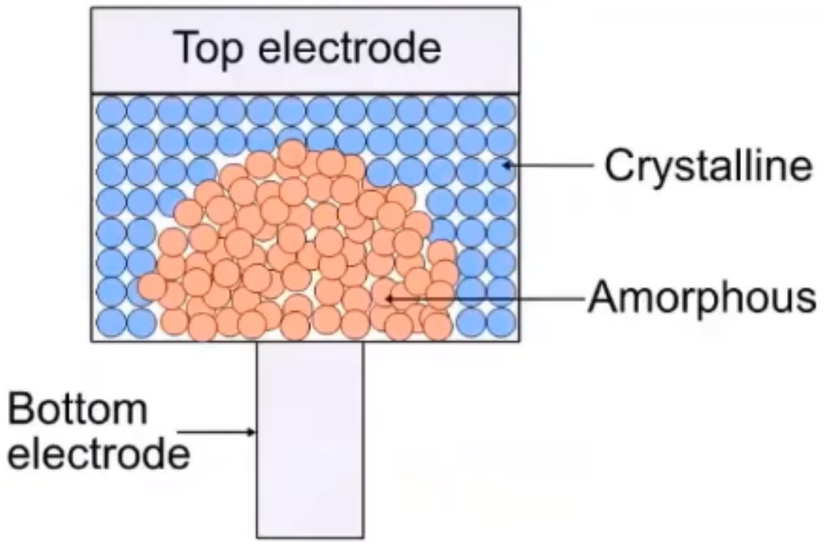


Resistance range = 10^3-10^7

Access time (write) = 10ns - 100ns

Endurance = 10^6-10^9

PCM

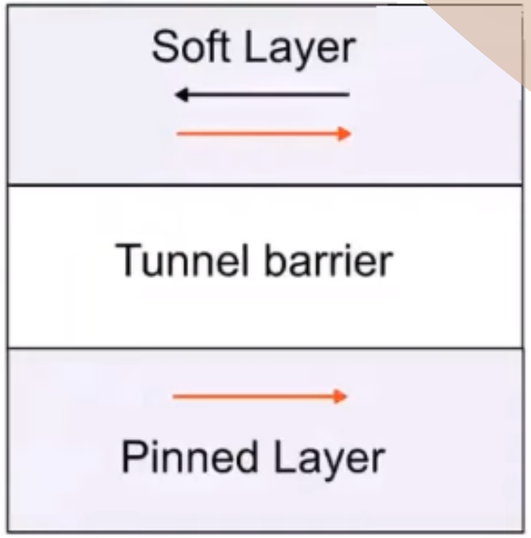


Resistance range = 10^4-10^7

Access time (write) ~ 100ns

Endurance = 10^6-10^9

STT-MRAM



Resistance range = 10^3-10^4

Access time (write) < 10ns

Endurance > 10^{14}

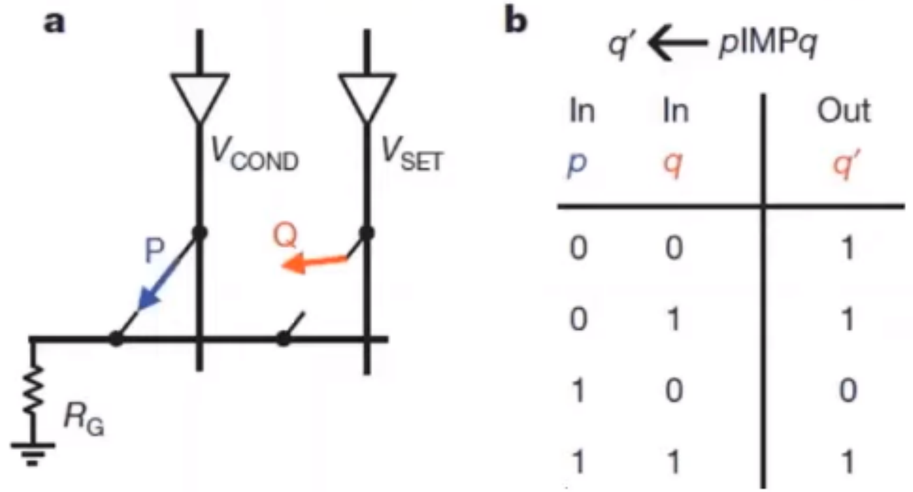
- **ReRAM:** Migration of defects such as oxygen vacancies or metallic ions
- **PCM:** Joule-heating induced reversible phase transition
- **STT-MRAM:** Magnetic polarization of a free layer with respect to a pinned layer

Wong and Salahuddin, Nature Nanotechnology (2015)

Stateful logic

'Memristive' switches enable 'stateful' logic operations via material implication

Julien Borghetti[1], Gregory S. Snider[1], Philip J. Kuekes[1], J. Joshua Yang[1], Duncan R. Stewart[1]† & R. Stanley Williams[1]



b

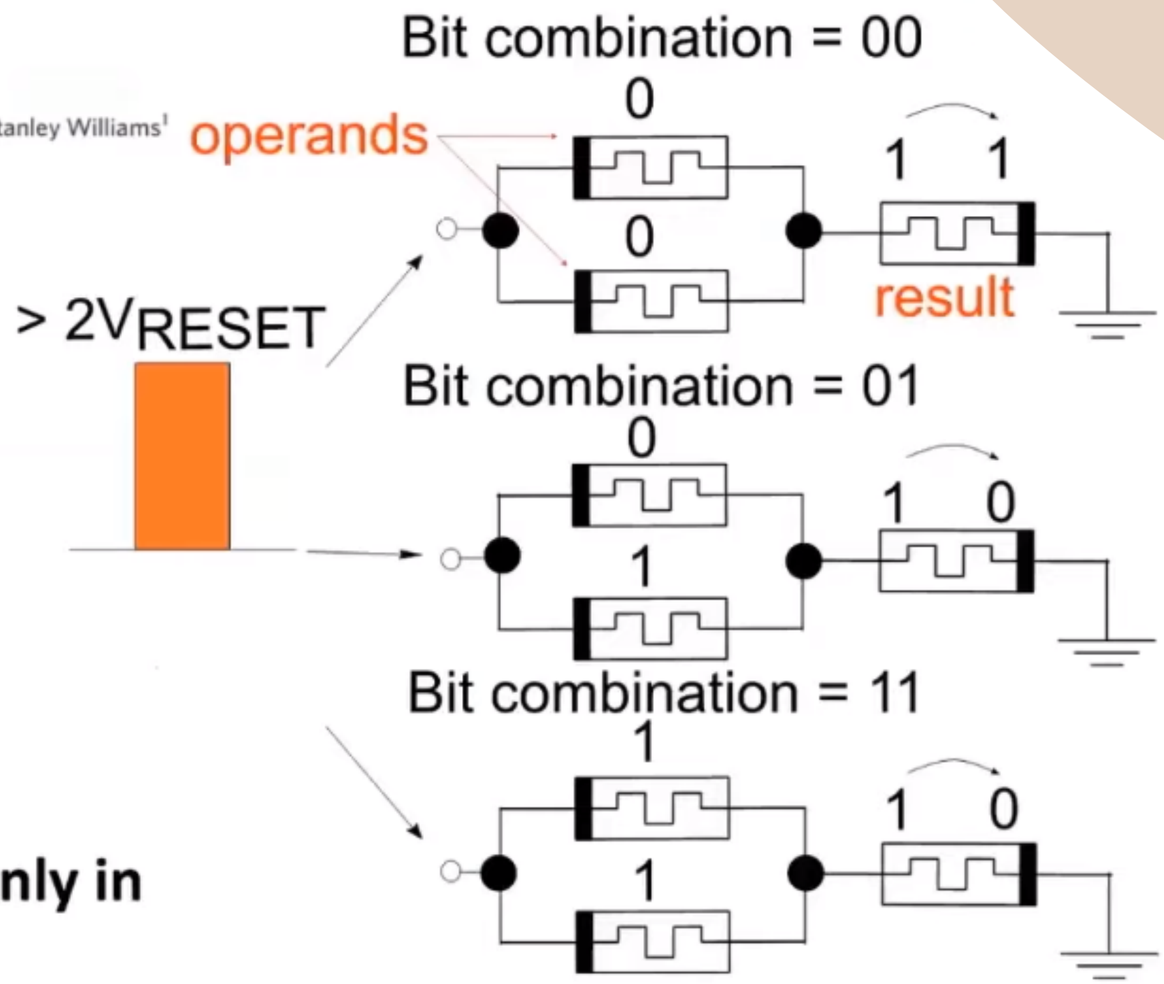
$q' \leftarrow p\text{IMP}q$

In p	In q	Out q'
0	0	1
0	1	1
1	0	0
1	1	1

Borghetti et al., Nature (2010)

- The Boolean variable is represented **only in terms of the resistance state**
- Both the operands and result are stored in terms of the resistance state variable

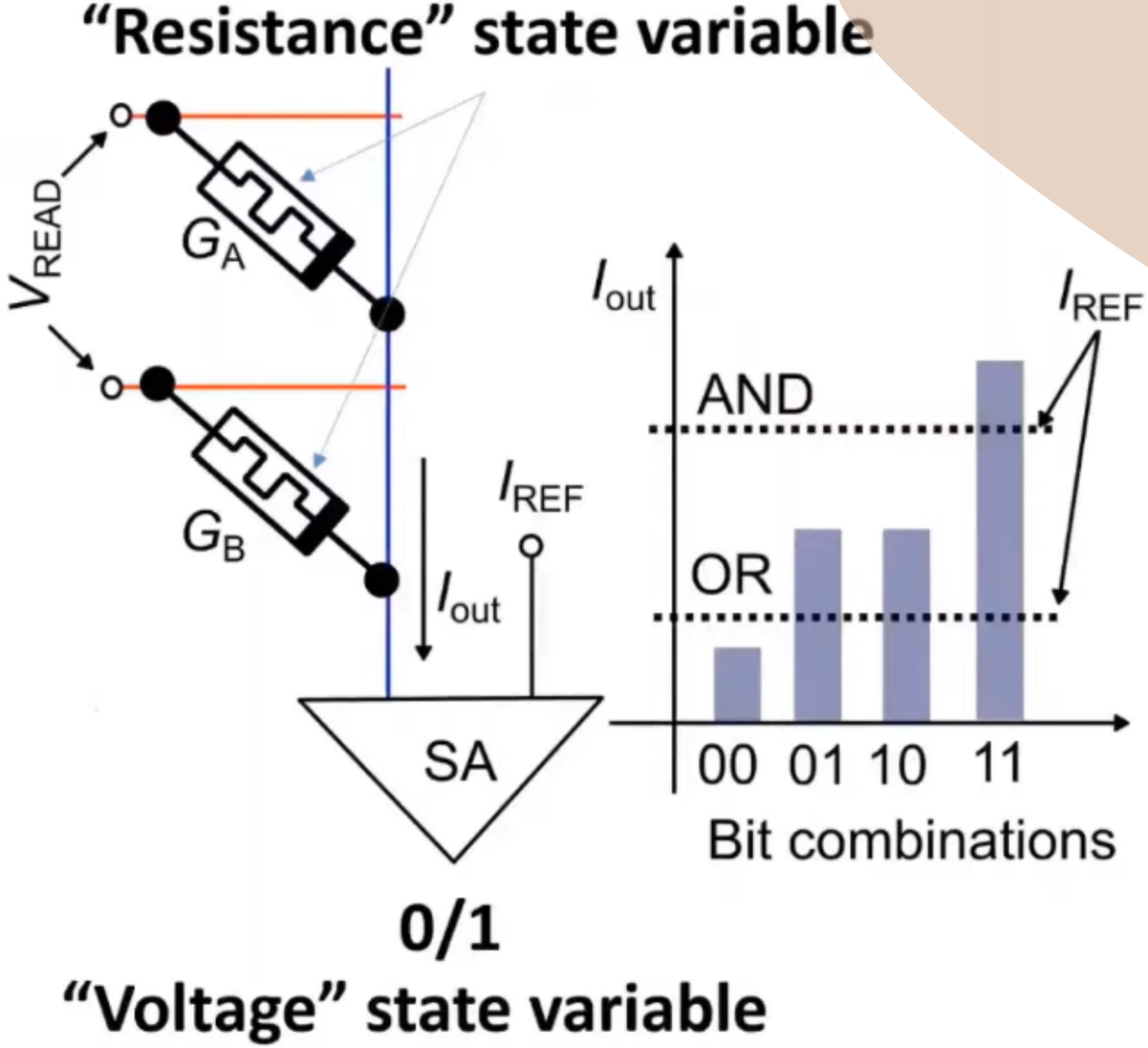
MAGIC: NOR Logic



Kvatinsky et al., IEEE TCAS (2014)

Non-stateful logic

- Both resistance and voltage state-variables co-exist
- Data is stored in terms of **resistance logic state-variables**; However, the logical operations are implemented in the periphery
- Eg. by **simultaneously sensing multiple memristive devices connected to the same sense amplifier**
- **Key advantage:** Memristive devices are programmed rather infrequently → limited cycling endurance is not a challenge



Li et al., Proc. DAC (2016), Xie et al., Proc. ISVLSI (2017), Hamdioui et al., DATE (2019)

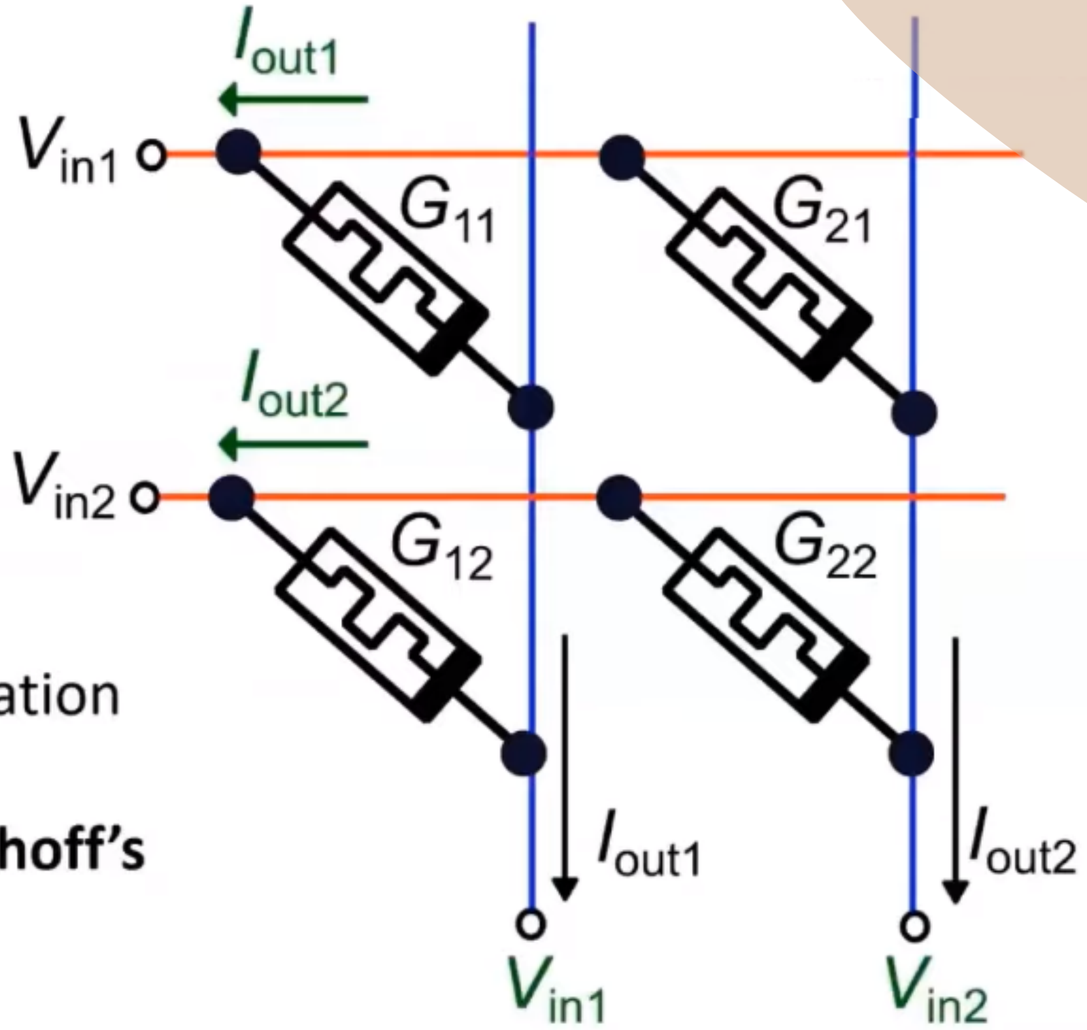
MVM using resistive memory

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

MAP to conductance values

MAP to read voltage

DECIPHER from the current



- In-place matrix-vector multiply (MVM) operation with O(1) time complexity
- Exploits **analog storage capability and Kirchhoff's circuits laws**
- Can also implement **MVM with the matrix transpose**

Burr et al., Adv. Phys. X (2017), Xia and Yang, Nature Materials (2019)

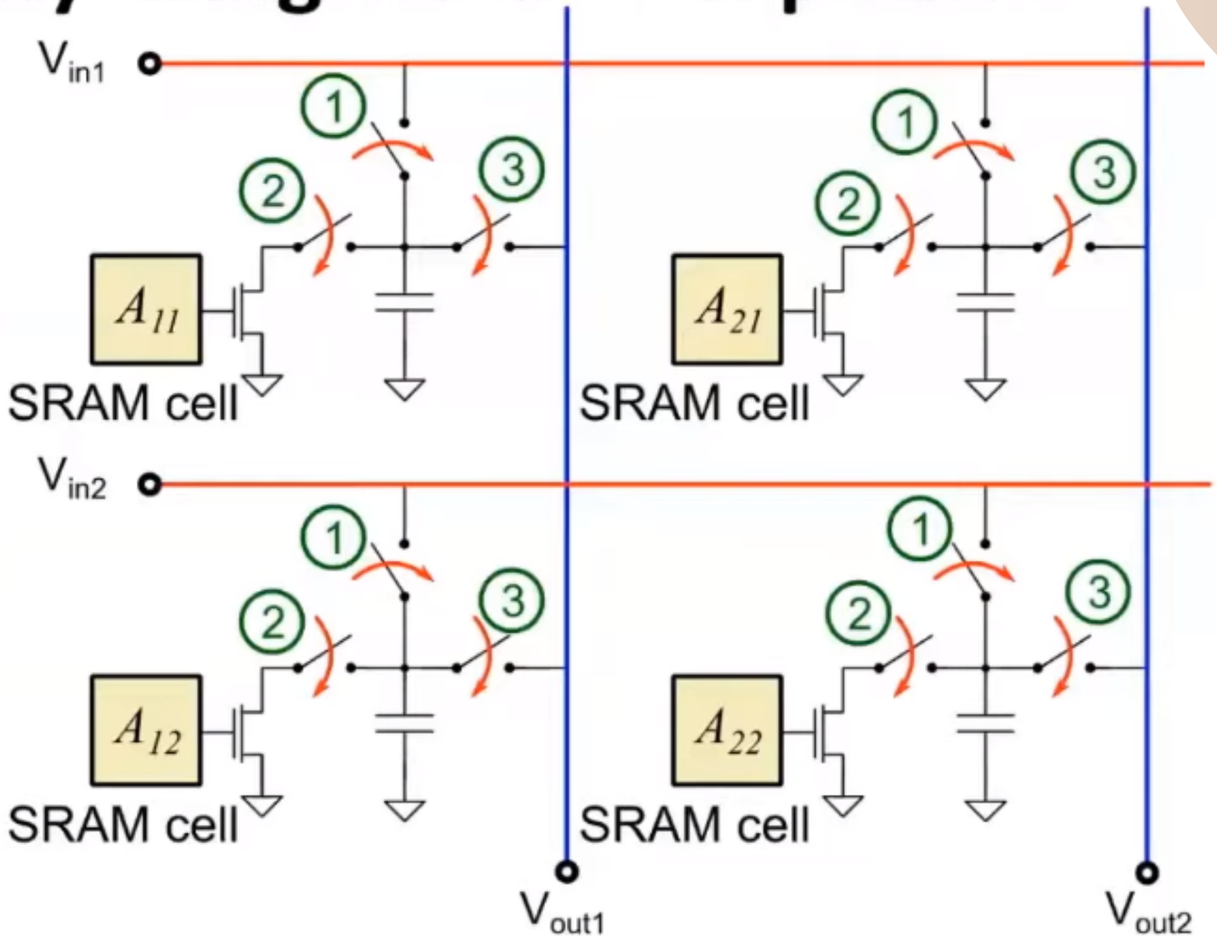
Matrix-Vector Multiply using SRAM + Capacitor

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

MAP to SRAM content

MAP to cap voltage

DECIPHER from voltage along the BL



- SRAM cells used to store **the elements of a binary matrix**
- **Step 1:** Capacitors charged to input values
- **Step 2:** Capacitors associated with value 0 are discharged
- **Step 3:** Capacitors shorted along the columns

Biswas et al., ISSCC (2018)
Valavi et al., JSSC (2019)
Khaddam-Aljameh, TVLSI (2020)

Thank You For Your Attention

We're ready to answer your questions.

Email

meh.arghand@webmail.ac.ir

Supervisor

Dr. Hamed Farbeh

Memory Technologies

Amir Kabir University Of Technology