Spintronic Technology For Energy-Efficient In Memory Computing

Part 4 – Emerging Memory Structures: MRAM

Dr. Esteban Garzón

Department of Computer Engineering, Modeling, Electronics and Systems (DIMES), University of Calabria, Rende 87036, Italy

esteban.garzon@unical.it

Lecture Outline

Introduction

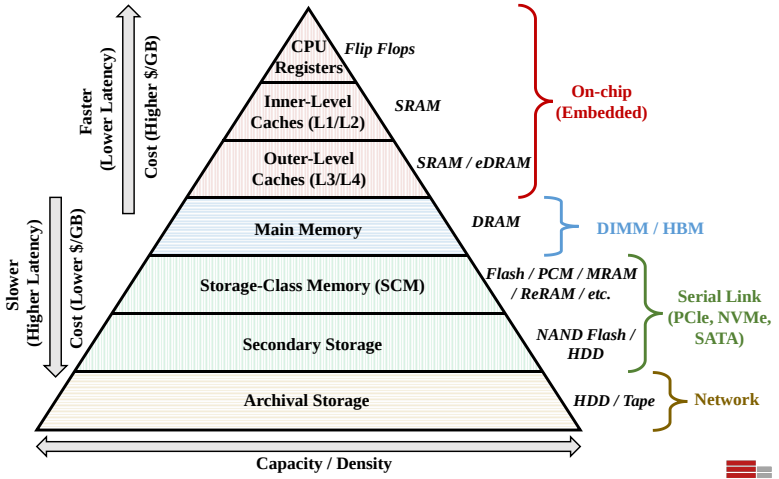
- Programmable electronic computers have been evolving for three-quarters of a century
  - Early days of computing: every logic gate was expensive
  - Today's computing systems: multibillion device integrated circuit (IC) chips
- **Large-scale integration** brought forward by Moore's Law has **shifted the weight** from **computational units** to **data storage and movement**

Bottleneck of modern computing systems is centered in the memory

- Current computing systems:
  - Memory bottleneck affects from high-performance processors to AI accelerators
  - Extra micron of silicon dedicated to memory (usually built with on-chip SRAM)
  - Not enough memory?... External dies/drives added for data storage.

The Computer Memory Hierarchy

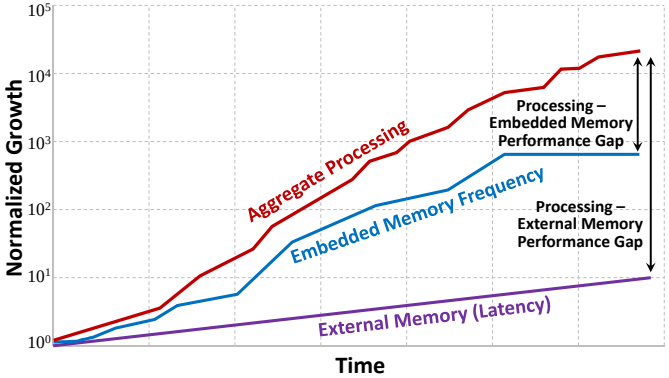
- Memory hierarchy defined over past 4 decades according to pyramid
- Top-down:
  - CPU registers at top: subcycle access latencies
  - On-chip memories located below CPU registers
  - System goes off-chip to DRAM
  - High-capacity secondary storage implemented with solid-state or hard-disk drives



Source: E. Garzón, et al., Wiley EEEE, 2022

Gap Between Processing and Memory

- While the memory pyramid has been sufficient to satisfy the extraordinary increase in computing power during the Moore's Law era, **the increase in memory performance has lagged behind that of computing**



Source: Based on Synopsys, reproduced by E. Garzón, et al., Wiley EEEE, 2022

Gap Between Processing and Memory

Real-life case facing Memory Bottleneck

- **GPT3 model is very large, and it is growing. hundred of GB of parameters** (Nvidia DL Architect, May 2023)
  - we need a lot of compute!
  - Nvidia DL architects are trading computations for more memory
- **chatGPT**
  - Its not a one-shot inference → If you want an output of 1K words, that means we are doing about 1K inferences
  - Computing hardware costs is about $ 700K per day. Per query is about 0.36 cents (source: Here)
  - It needs 100 servers with about 29K GPUs!
  - 108G parameters that have to be stored in **memory**... **What about the bandwidth?**
- **Industry does not have a good memory solution that address these challenges**

Closing the Gap?

- Commercially available products to close the gap and provide the required memory bandwidth for current and future products

1. Complex caching configurations have been applied for several generations (L4-cache)
   - **limited to several hundreds of MB and is usually well below 100 MB**
2. Embedded dynamic random access memory (eDRAM) has been used as a higher density alternative
   - **does not scale beyond 14 nm**
3. Scaling the capacity and bandwidth of DRAM (DDR5...)
   - **scaling problems and high-power consumption**
4. Flash memory improvements: NAND Flash, 3-D stacking
   - **large access times, higher error as we scale them. It is reaching its scaling limits**

The Utopian Universal Memory

A truly utopian universal memory would have the following features

High density

Similar to Flash and DRAM

Scalability

Similar to SRAM, it would be compatible with deeply nanoscaled process nodes

Retention

Unlike DRAM, it would not suffer from limited retention times, and like Flash and HDD, it would preferably be nonvolatile

Performance

low latency read and write operations, comparable to SRAM

The Utopian Universal Memory

Endurance

Similar to SRAM and DRAM, it would support a practically unlimited number of write/read accesses ($> 10^{15}$)

Integration

As with SRAM, eDRAM, and embedded Flash (eFlash), it would be embedded on the same die as the computational logic.

Power consumption

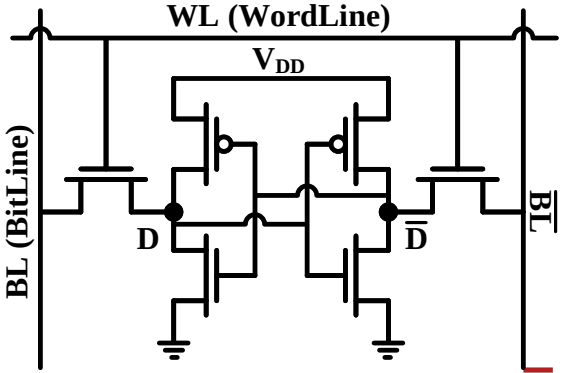
In addition to the low leakage obtained through non-volatility, it would also provide low-power read and write operations, achievable with low voltages.

Limitations of Current Memory Technologies

Static Random-Access Memory (SRAM)

- Primary embedded memory technology: SRAM
- Basic building block of SRAM: 6T bitcell
  - Single-port access
  - Static volatile data storage
  - Full-rail data levels
  - Fast access times with differential, nondestructive read
  - Support for access at bit granularity
- Logic process technologies are optimized for integration of 6T SRAM bitcells, allowing nonstandard (pushed-rule) design rules to achieve high density.

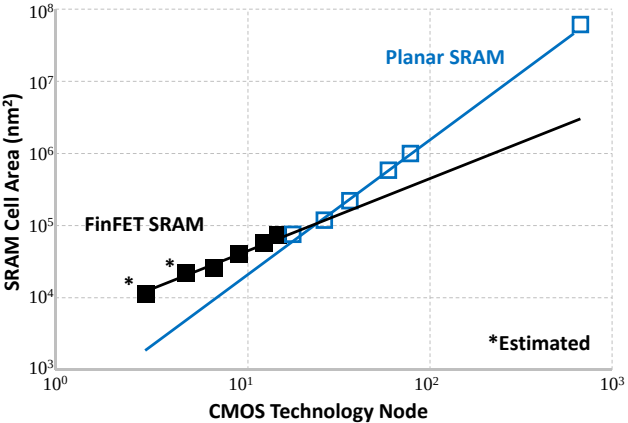
6T-SRAM Bitcell



Limitations of Current Memory Technologies

Static Random-Access Memory (SRAM)

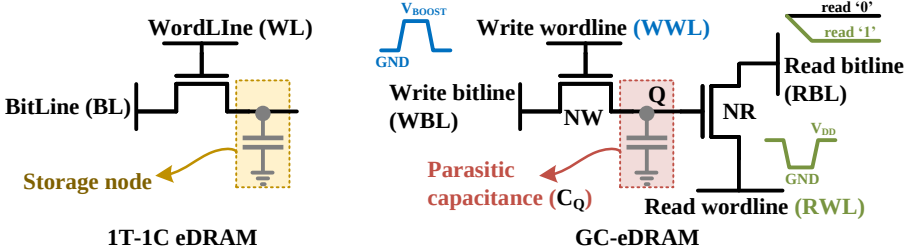
- With move to FinFET technologies at below 22 nm nodes, SRAM bitcell scaling has slowed down
- SRAM suffers from relatively high leakage and tends to fail under voltage scaling, reducing the opportunities for low-power operation



Source: Based on Walker EEtimes, reproduced by E. Garzón, et al., Wiley EEEE, 2022

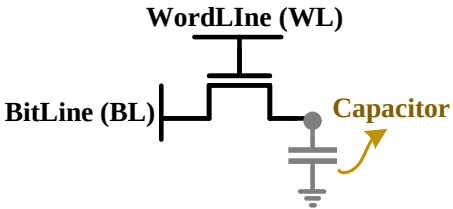
Embedded DRAM (eDRAM)

- eDRAM proposed to implement large embedded memory blocks
- The majority of **eDRAM** is implemented using the **1T–1C**
- **Embedding this technology along with the logic requires a large number of process cost adders!**
  - Implementation of 1T–1C eDRAM in sub-20 nm technologies: IBM z15 servers
- As a logic-compatible alternative: gain-cell-embedded DRAM (GC-eDRAM)
  - potential for scaling into FinFET technologies without bulky process cost adders
  - However, lower density gains than 1T–1C eDRAM



Dynamic Random Access Memory (DRAM)

- Gigabytes of main memory require high-capacity technology
  - Standalone 1T–1C DRAM is used due to limited SRAM density
- DRAM access time slower than embedded memories (10-100 cycles)
- **Single-ported and destructive read; write-back required**
- Periodic refresh required (every 64 ms), resulting in **high-power consumption**
- DRAM faces many challenges as technology scales
- Harder to fabricate capacitor, reducing retention times and sensing margins
- Small transistor sizes and dense capacitor integration make DRAM susceptible to soft errors and malicious tampering



Flash and Embedded Flash (eFlash)

- NAND **Flash** provides nonvolatility and relatively fast access times but has several **drawbacks**:
  - **Write operations lead to breakdown**, limiting endurance and requiring wear leveling controllers
  - Asymmetric access limits technology to block-access granularity
  - **Reducing the size** of the device **reduces the amount of charge**, limiting scalability

- **eFlash** is widely used in microcontrollers for nonvolatile microcode storage
  - Integrating NVM on-chip simplifies design and reduces overall cost.
  - eFlash is typically based on NOR technology, providing high-speed read access and simpler control and management than NAND Flash.
  - planar approach: compromise density for ease of integration.
  - eFlash below 28 nm is seen as extremely challenging and is not currently provided by leading foundries.

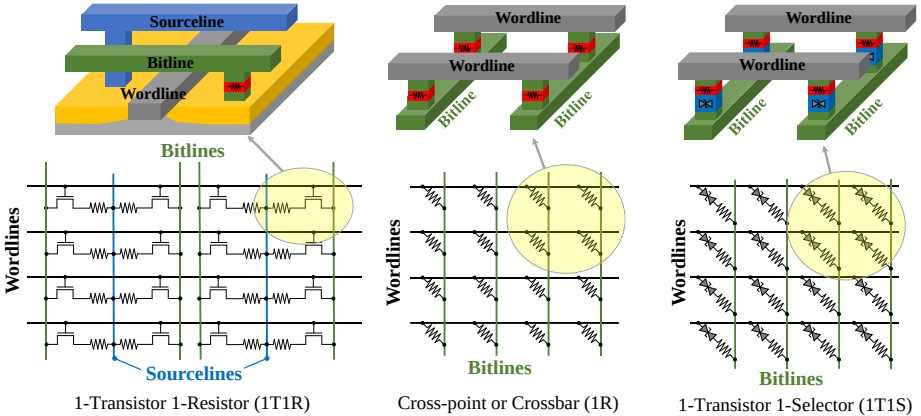
Alternative Approaches

- Search for alternative technologies with the desire to approach the capabilities of a universal memory.
  - Phase-Change Memory (PCM)
  - Resistive Random-Access Memory (RRAM)
  - **Magnetoresistive Random-Access Memory (MRAM)**
- All these programmed to change their resistance to represent the stored data levels.

Electrical operation is applied to the device causing it to change its properties resulting in device resistance change

Array architectures

- Resistive memories can be implemented in three typical array architectures
  - 1-transistor 1-resistor (1T1R) approach used for PCM and STT-MRAM
  - Crosspoint (1R) array: dense but present sneak paths)
  - 1-Transistor 1-Selector (1T1S): solves the sneak path problem, but blocks bipolar writes



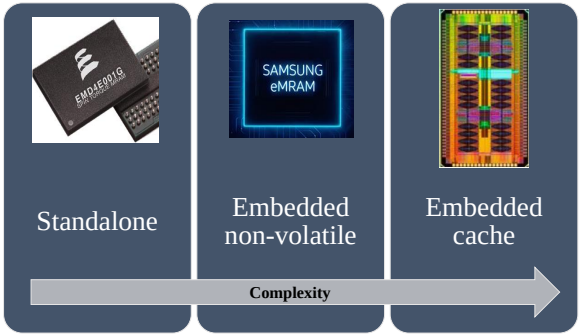
Comparison Between Memory Technologies

	SRAM	DRAM	NAND Flash	NOR Flash	PCM	STT-MRAM	RRAM
Cell Elements	6T	1T1C	1T	1T	1T1R or 1D1R	1T1MTJ	1T1R or 1D1R
Cell Size	60-500 F^2	6-10 F^2	4-6 F^2	6-8 F^2	4-12 F^2	6-50 F^2	4-10 F^2
Stackable	No	Yes	Yes	No	Yes	No	Yes
MLC (bits/cell)	No	No	Yes	Yes	Yes	No	Yes
Projected Process	< 3 nm	~10 nm	~14 nm	~ 45 nm	< 10 nm	5 nm	< 5 nm
Access Granularity	1 B	64 B	R/W: 4 KiB Erase: 256 KiB	R/W: 1 B Erase: 64 KiB	64 B	64 B	64 B
Read Latency	< 1 ns	< 10 ns	25 µs	25 ns	< 50 ns	10 ns	< 100 ns
Write Latency	< 1 ns	< 10 ns	500 µs	5 ms	< 150 ns	10 ns	< 100 ns
Endurance	> 10^{16}	> 10^{16}	10^4-10^5	10^5	10^6-10^8	10^{13}-10^{15}	10^6-10^{11}
Retention	Volatile	Volatile	> 10 yr	> 10 yr	> 10 yr	> 10 yr	> 10 yr

- STT-MRAM is a prime candidate to replace NOR Flash in embedded NVM applications
- Fast access times supported by STT-MRAM position it as a non-volatile SRAM alternative
- Stand-alone STT-MRAM is also considered a persistent DRAM alternative

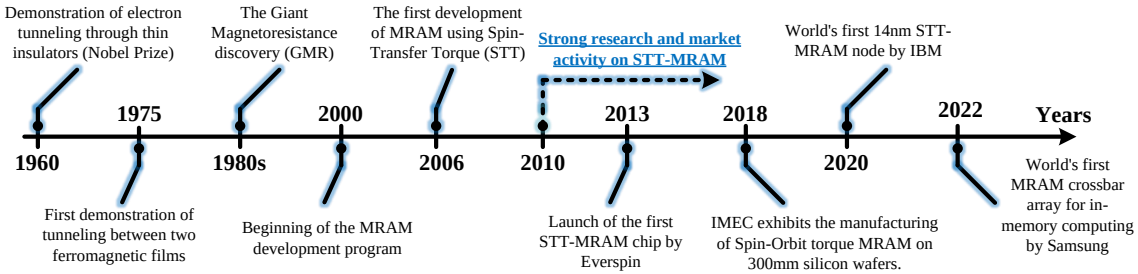
Magnetoresistive Random-Access Memory (MRAM)

- Magnetoresistive Random-Access Memory (MRAM) technology has grown in popularity, thanks to its promising features
  - reduced area footprint
  - inherent non-volatility
  - relatively large endurance
  - compatibility with CMOS processes
  - ability to operate at low voltages
- **MRAMs are promising candidates to replace: semiconductor-based memory technologies, from standalone memories (e.g., DRAM and Flash) to embedded nonvolatile memories (e.g., eFlash) and embedded caches (e.g., SRAM)**



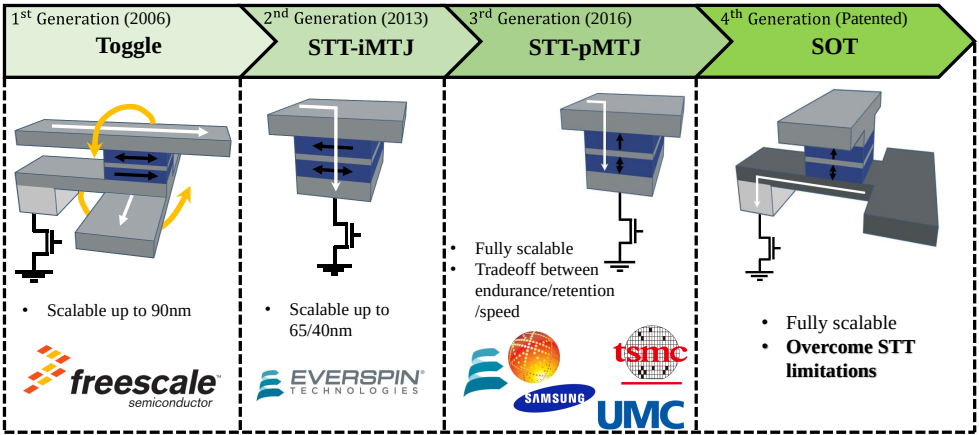
Brief Historical Overview

- Historical timeline of MRAM technology



MRAM Generations

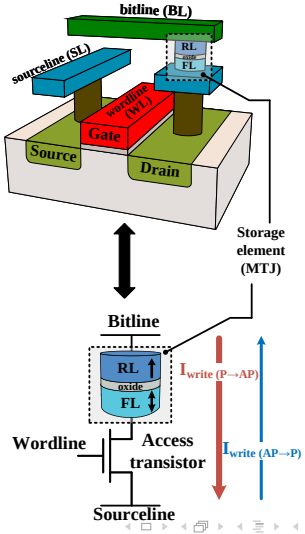
MRAM technology has been evolving throughout the past two decades, with four main generations driving these spintronics-based memories to the market.



*Note: Non-exhaustive list of companies

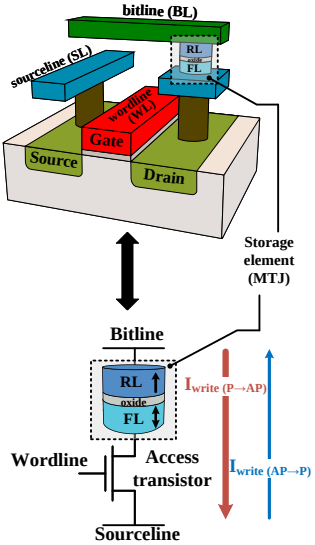
MRAM Memory Architecture

- Industry efforts to integrate spintronic technology with commercial CMOS process have been successful
- Examples:
  - Samsung: manufacturability of an 8Mb STT-MRAM embedded in 28 nm process.
  - imec: BEOL compatibility of MTJs based on dual barriers (an improved version of the single barrier) for the first time.
  - Samsung: A novel STT-MRAM integration approach for cost- and energy-efficient memory systems.
  - GlobalFoundries: manufactured a 40Mb embedded MRAM



MRAM Memory Architecture

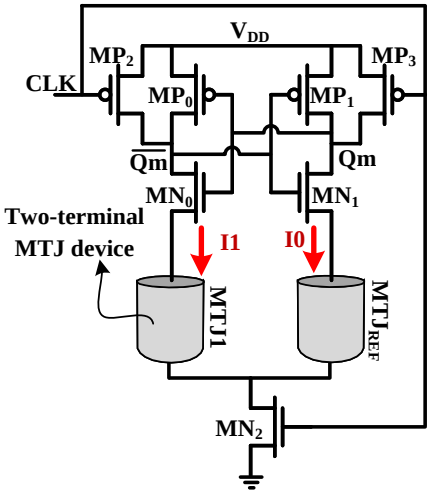
- An STT-MRAM cell can be constructed with different configurations, depending on whether the RL points to the access device or to the bitline
- To store or read the information, the access transistor is asserted, and electrical current is driven from the bitline to the sourceline or vice versa.
- Note that the read and write operations are executed through the same path



Read Operation – Sensing Circuitry

Pre-Charge Sensing Amplifier (PCSA)

- The typical method for reading the stored data is to drive a current below the critical switching current of the MTJ from the bitline.
- Depending on the resistance of the bitcell, a voltage drop is generated between the bitline and the sourceline, which can be read out with a sense amplifier.

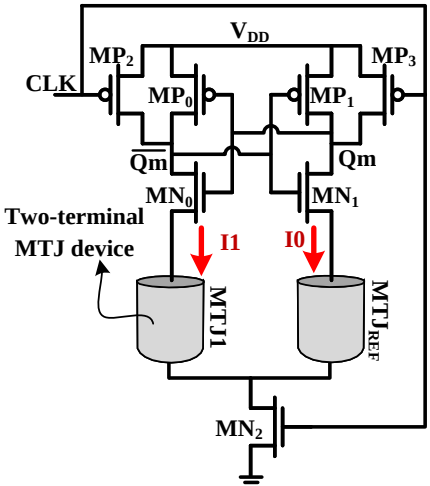


Read Operation – Sensing Circuitry

Pre-Charge Sensing Amplifier (PCSA)

PCSA building blocks

- Pre-Charge sub-circuit: MP2-3
- Discharge sub-circuit: MN2
- Amplifier sub-circuit: cross-coupled of inverters
- A reference MTJ (MTJ_{REF}) and the MTJ we are going to sense

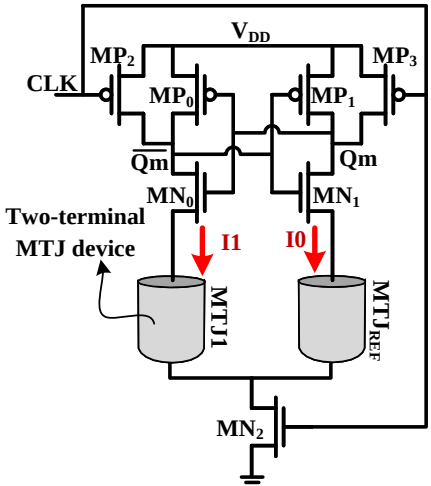


Read Operation – Sensing Circuitry

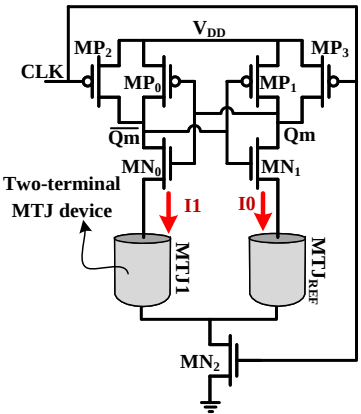
Pre-Charge Sensing Amplifier (PCSA)

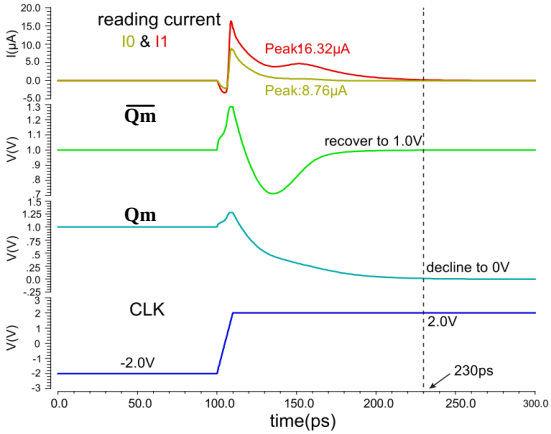
Principle of operation

1. Pre-charge phase: the gate of the MP0 and MP3 are asserted (CLK = 0), and the Qm nodes are charged to the supply voltage (V_{DD})
2. Evaluation: starts when CLK = 1, turning off the MP0 and MP3 transistors.
   - One of the two branches (left or right) will discharge faster down to ground. This depends on the resistance of the MTJs.



Example of waveforms of the PCSA circuit. Note: read operation is fast!

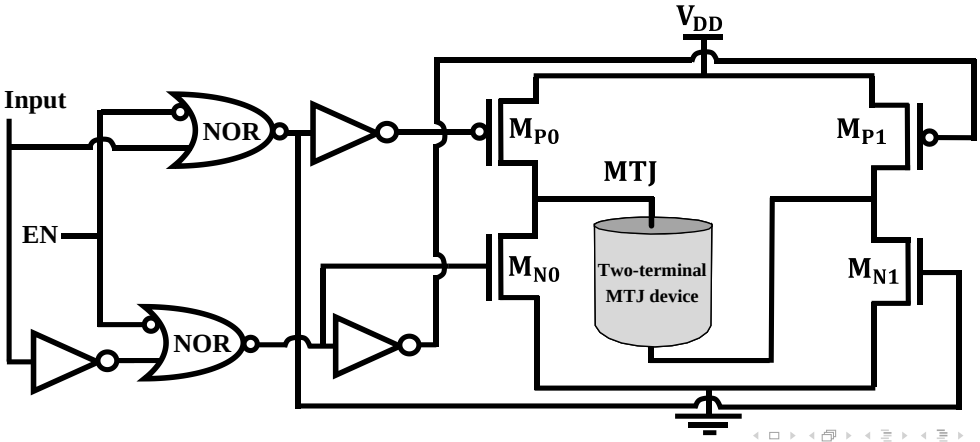




Source: ©H. Wang, et. al., IEEE Transactions on Electron Devices, vol.65, issue 12, 2018

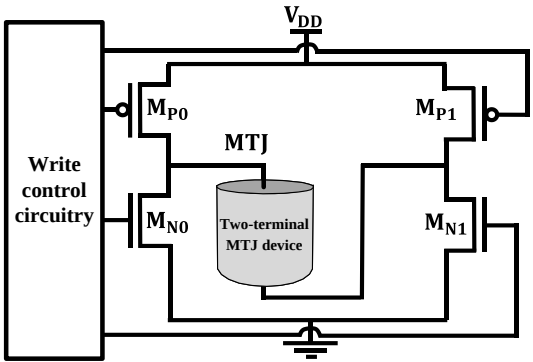
Write Operation - Writing Circuitry

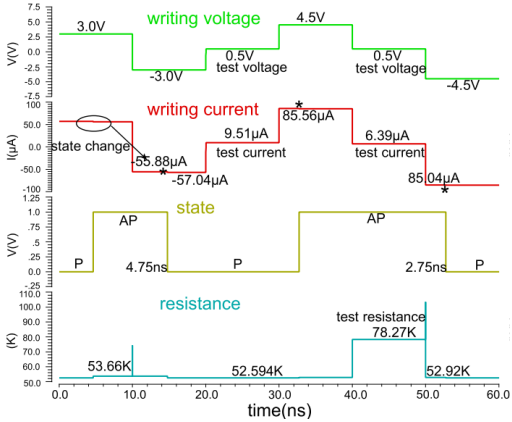
- Single MTJ write circuitry shown below.
- Two main sub-circuits: (1) the write transistors MP0-1, and MN0-1. (2) Control block composed of NOR gates and inverters.



Write Operation - Writing Circuitry

- Example: write operation waveforms





Source: ©H. Wang, et. al., IEEE Transactions on Electron Devices, vol.65, issue 12, 2018

Summary MRAM

- ☺ **MRAM as a universal memory as it exhibits non-volatility, compatibility with CMOS BEOL process and efficient scaling as well as low access latency and high endurance.**

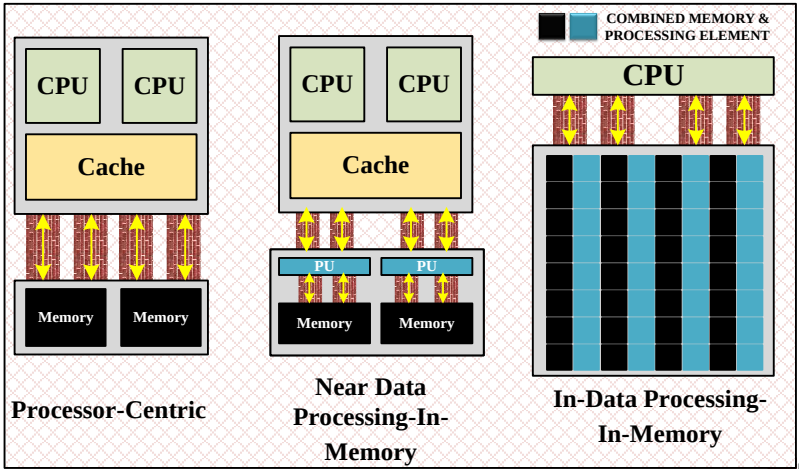
  However, some challenges have to be addressed:
- ☹ write latency
- ☹ density
- ☹ reliability

The main application of STT-MRAM in the short term is eFlash replacement

Processing In Emerging Memory

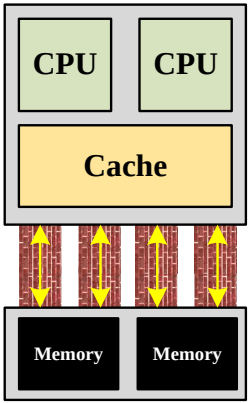
The big data era brings **two main "walls" for conventional computer architecture** (i.e. Von-Neumann – Processor-centric):

- **Memory Wall**: Limited data transfer bandwidth between processor and memory
- **Power Wall**: High energy consumption related to data transfer



Data-Centric Processing

- Limitations
  - Memory bandwidth
  - Speed
  - High-energy
- **Processor-centric model is sufficient when computing dominates data transfer**
  - Not sufficient with data-centric applications (big-data analytics, AI)
- Processing-in-memory (PIM): computing either near-data using explicit processing units or in-data using the memory itself.
  - **Past PIM research efforts were not adopted at large scale mainly because, until recently, computing used to dominate data movement and memory technologies did not scale sufficiently fast.**

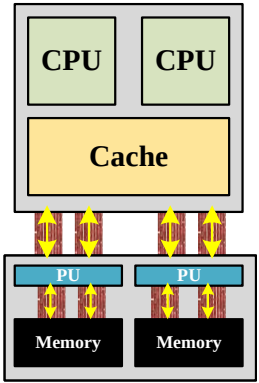


Processor-Centric (von-Neumann architecture)

Processing Near Memory

Near data-processing systems: processing cores are placed close to the data.

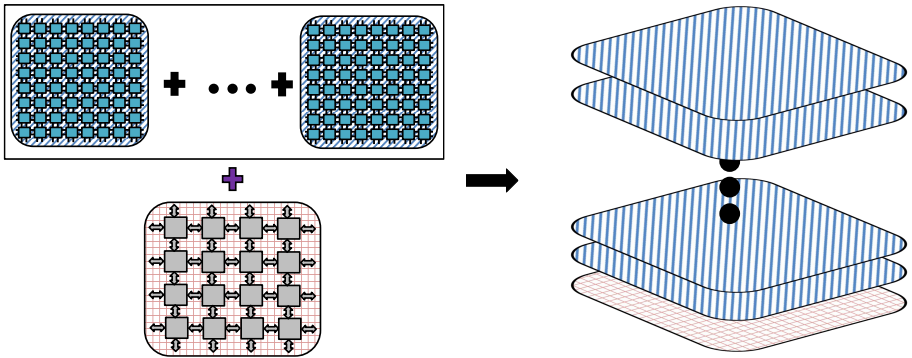
- The concept has been around since the 1960s
- Vast number of works have proposed different near memory architectures.
- **Near-data PIM alleviates the limitations of von Neumann architecture but does not remove them**
  - Data still needs to be transferred between memory and processing units
  - To alleviate this: 3D-based near-data PIM.



Near Data Processing-In-Memory

3D-Based Processing Near Memory

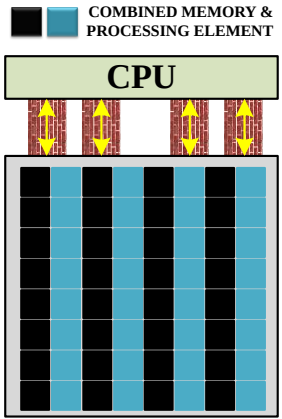
- 3D memory and logic stacking technology (e.g., 3D DRAM cube/logic stack-based PIM designs have been proposed)
  - Improves the bandwidth and the energy cost of data transfer. (partially alleviates the problem)



In-Data Processing

In-data processing: computing operations performed by memory cells.

- The key idea: bitwise amalgamation of memory and processing
- In an **in-data processing architecture**, each **processing element comprises** a **storage element** and hence requires much less data transfer.
  - The bulk of data never leaves the memory
- Several architectures have been proposed
  - DRAM-based
  - Associative Processor (AP) in Memory (e.g., replace last level cache with an AP)



In-Data Processing-In-Memory

In-Data Processing in Emerging Memory

- New nonvolatile memory technologies (e.g., PCM, RRAM, and STT-MRAM) provide significant advantages over conventional DRAM in terms of memory capacity, energy efficiency, and compute capability.
- Computing principles
  1. **Analog computing** in resistive crossbars
  2. **Reconfigurable computing** using the ability of resistive elements to implement logic functions
  3. **Associative computing**, utilizing resistive content addressable memory.
- Applications of Processing in Emerging Memories
  - Deep neural network (DNN) acceleration
  - Graph processing
  - Bioinformatics and genome analysis
  - Blockchain and physically unclonable functions (PUFs)

Summary

- Search for alternative technologies with the desire to approach the capabilities of a universal memory.
- Primary candidates for replacing current memory technologies are defined as resistive memories.
- Assuming that the write latency, density, and reliability issues are overcome, STT-MRAM may target applications such as higher level cache, main memory, and persistent memory
- MRAM is offered by leading foundries, such as TSMC, GlobalFoundries, and Samsung.
- Emerging memory technology (e.g., spintronics) is a potential alternative to build non-volatile logic-in-memory architectures