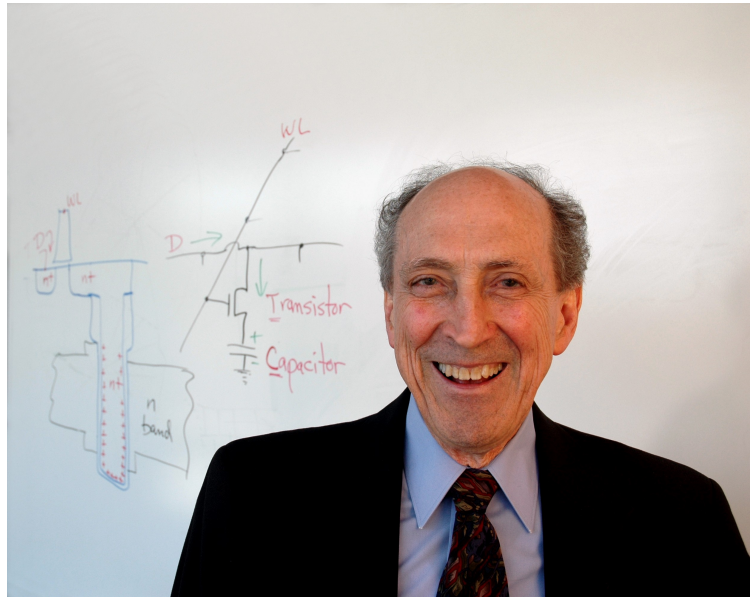
EECS151 : Introduction to Digital Design and ICs

Lecture 21 – Flash & Parallelism

Bora Nikolić and Sophia Shao

Robert Dennard

- Invented DRAM in 1968 at IBM
- Formulate Dennard's scaling: Maintain constant power density with improved frequency/performance
- The end of Dennard's scaling leads to the inability to further increase clock frequencies and multicore processors as an alternative way to improve processor performance



Review

- Memory arrays:
  - SRAM:
    - Unique combination of density, speed, power
    - SRAM cell: stability and writeability
  - Caches
    - Direct mapped and set-associative
  - DRAM
- Clocks
  - Clock generation and distribution is a major part of digital system design.

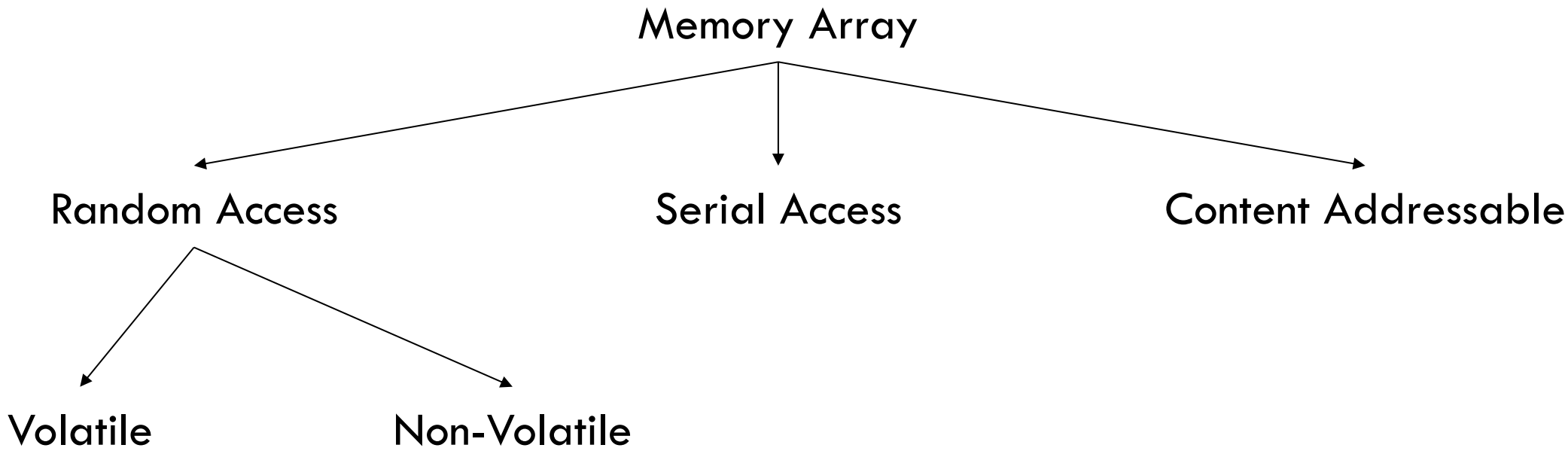
Flash

- Overview
- Read & Write Cell
- NAND/NOR Flash

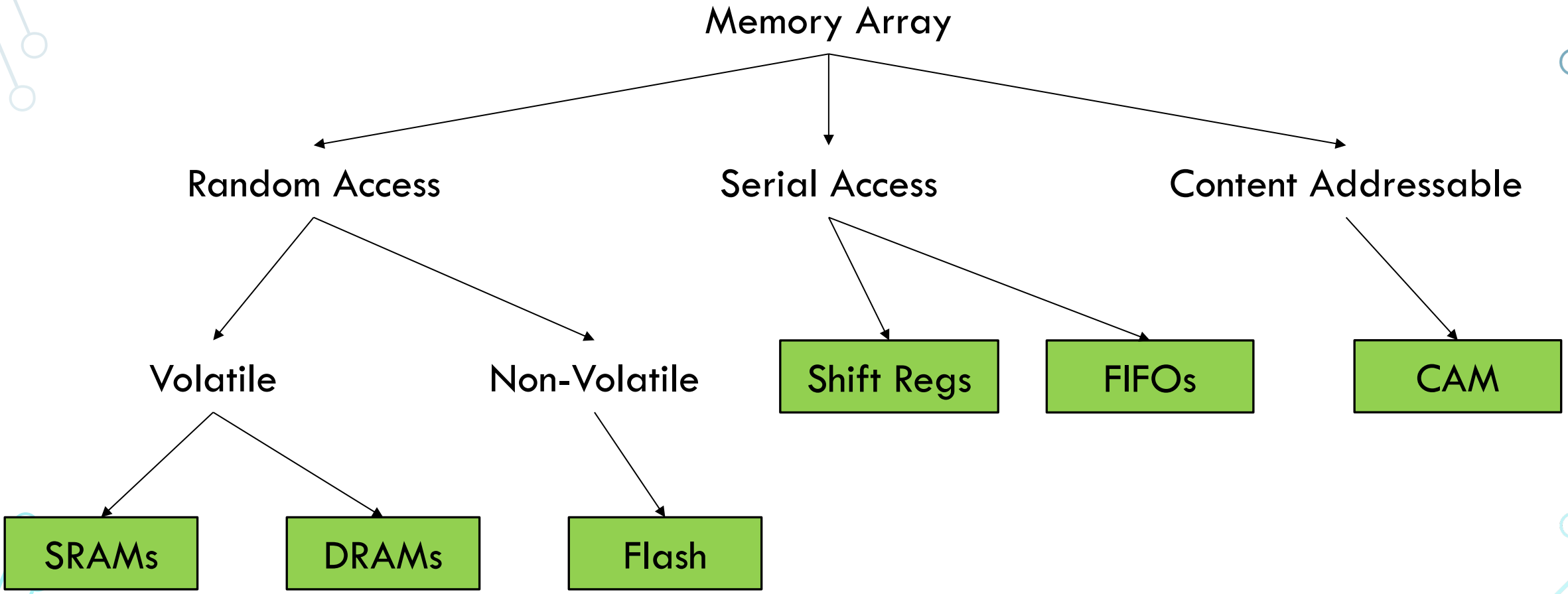
Parallelism

- Limits of Pipelining
- SIMD Processing
- Multithreading

Memory Overview

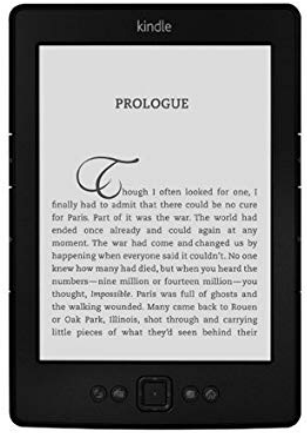


Memory Overview



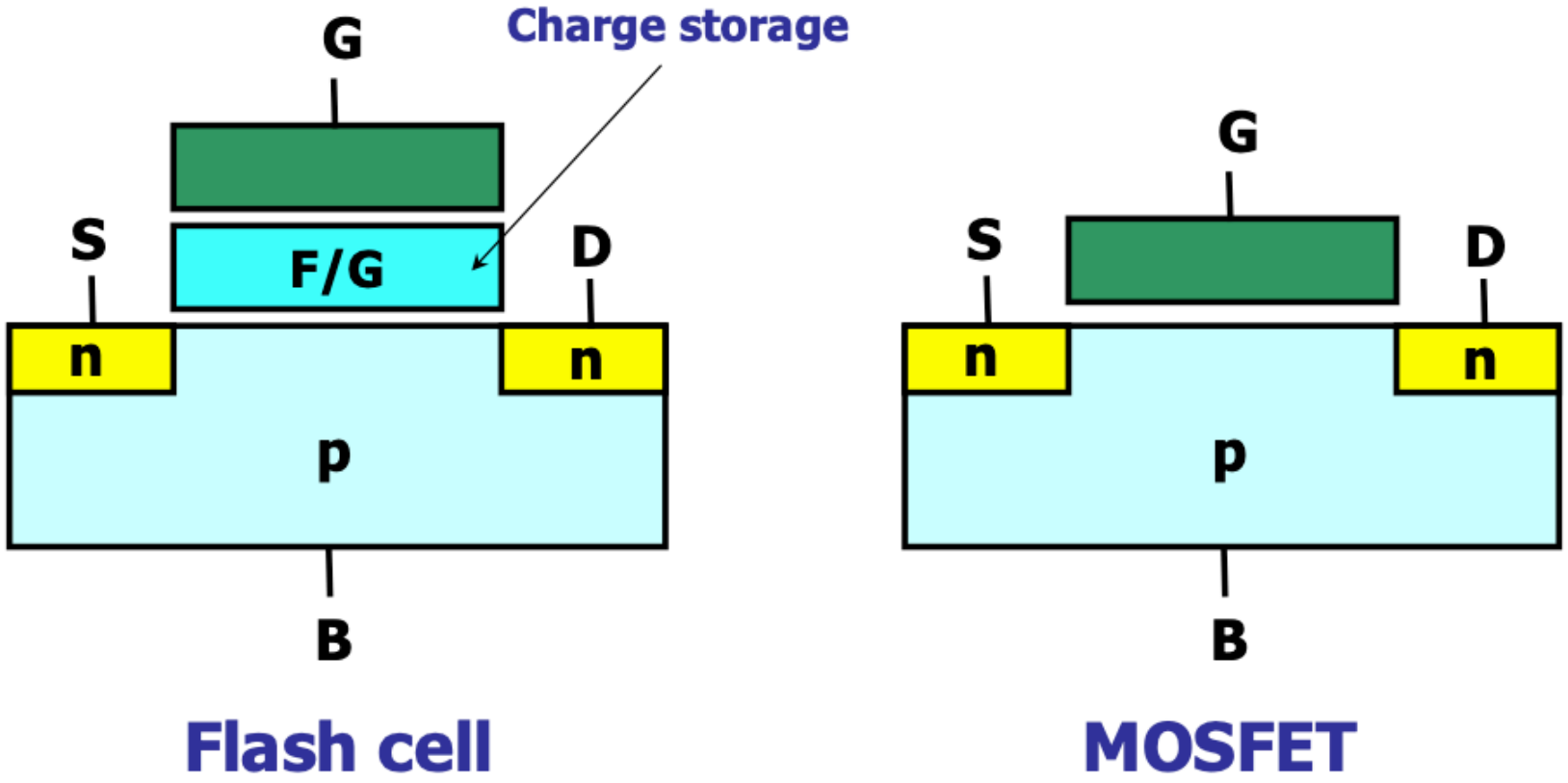
Flash Memory

- Non-volatile memory

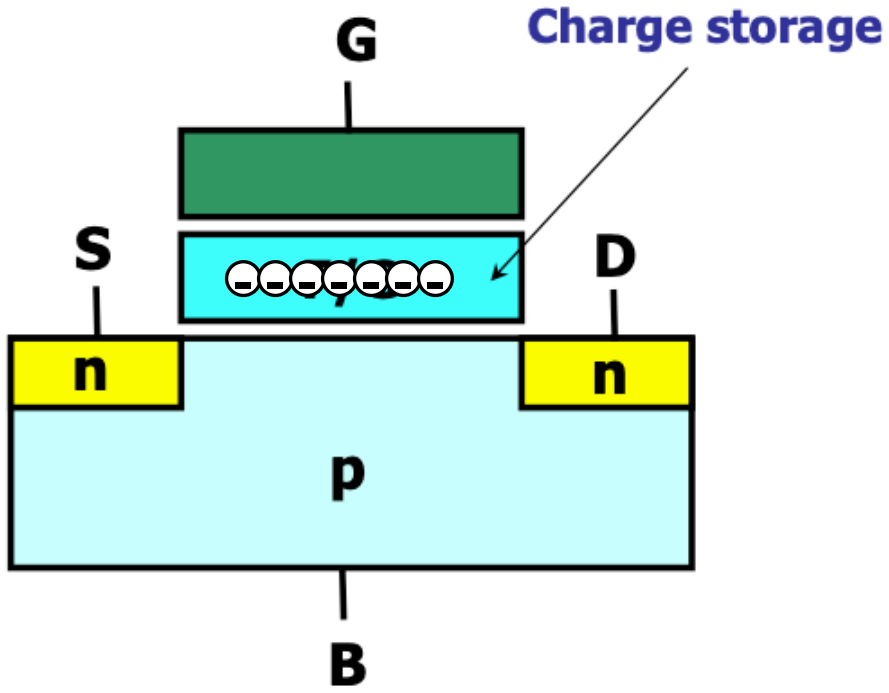


Key concept: Floating Gate

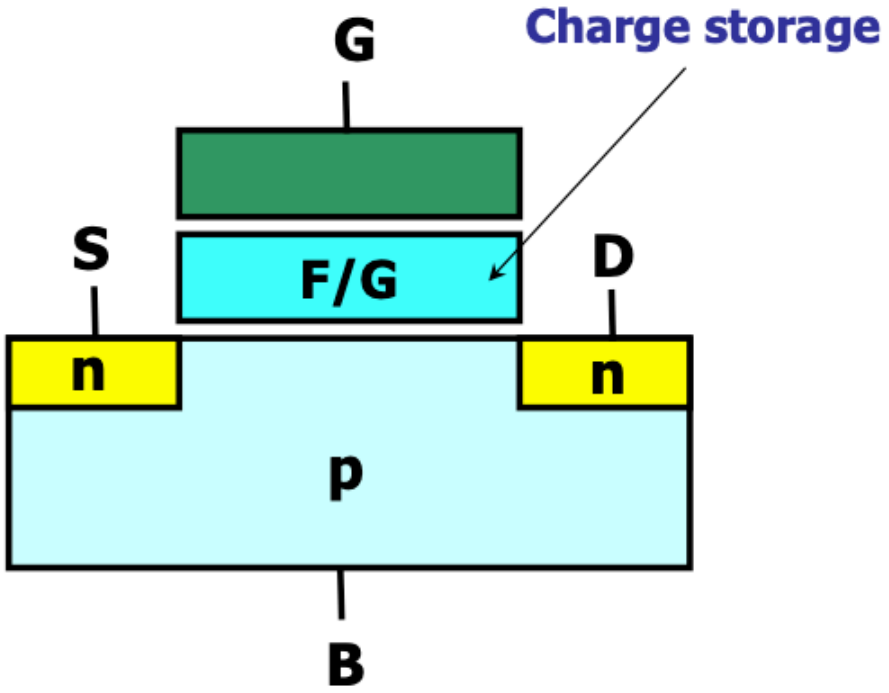
- Floating Gate: A charge storage layer -> memorize information
- A “Programmable-Threshold” Transistor



Single-Level Cell: 0 and 1 in Flash

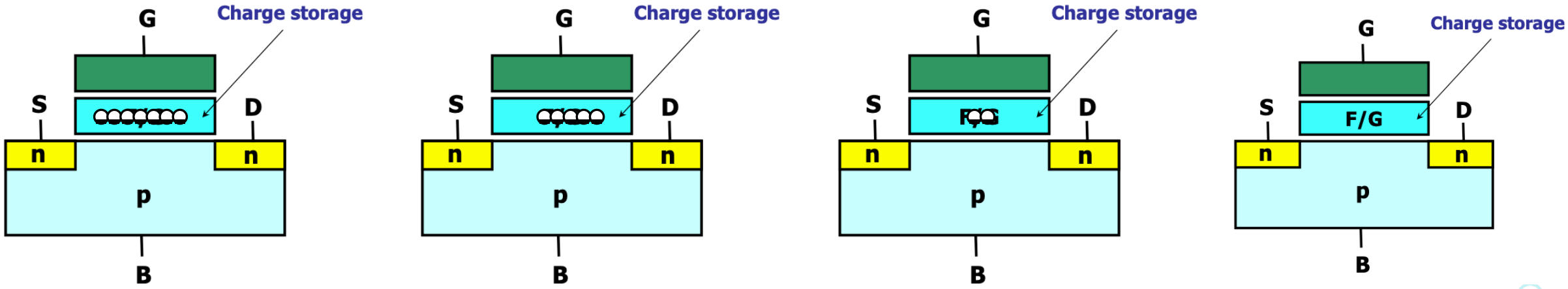


- Storing 0
- Negative charge in floating gate



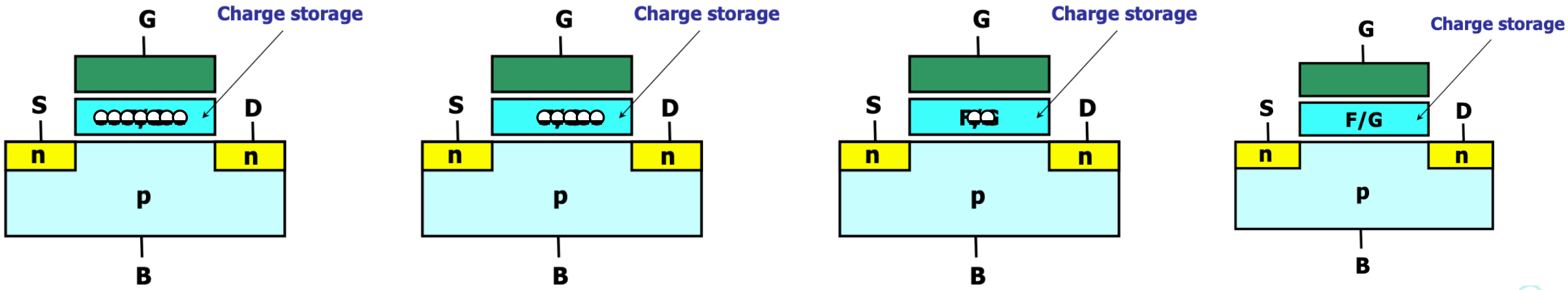
- Storing 1
- No charge in floating gate

Multi-Level Cell



Multi-Level Cell

- Higher density
- More errors



Flash

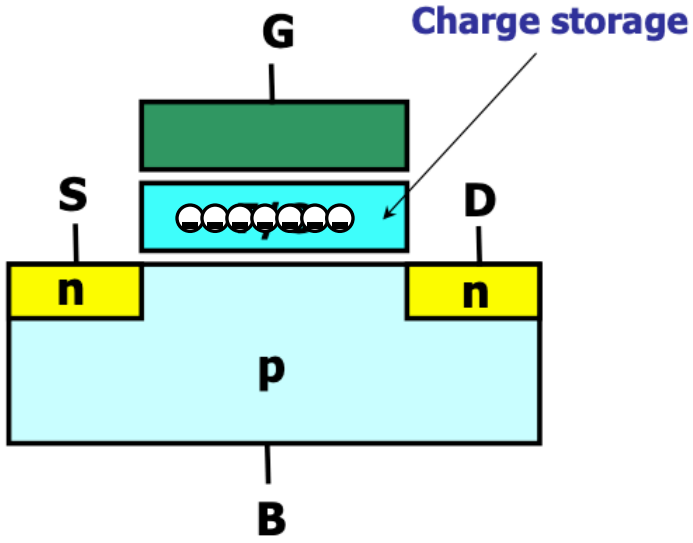
- Overview
- Read & Write Cell
- NAND/NOR Flash

Parallelism

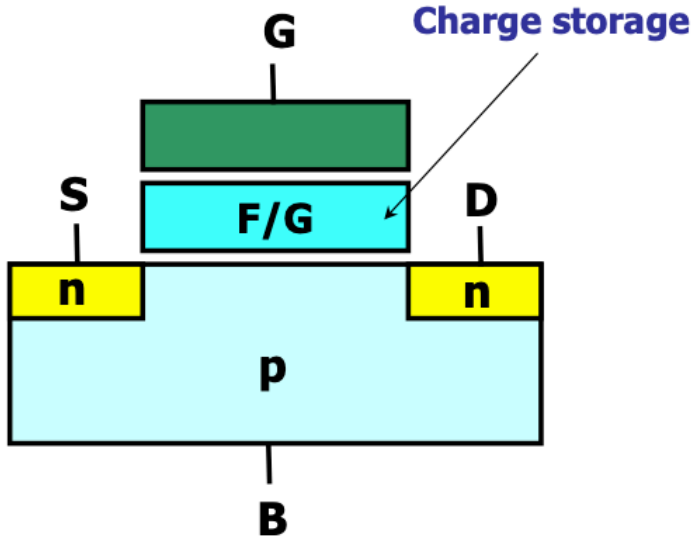
- Limits of Pipelining
- SIMD Processing
- Multithreading

Read a Flash cell

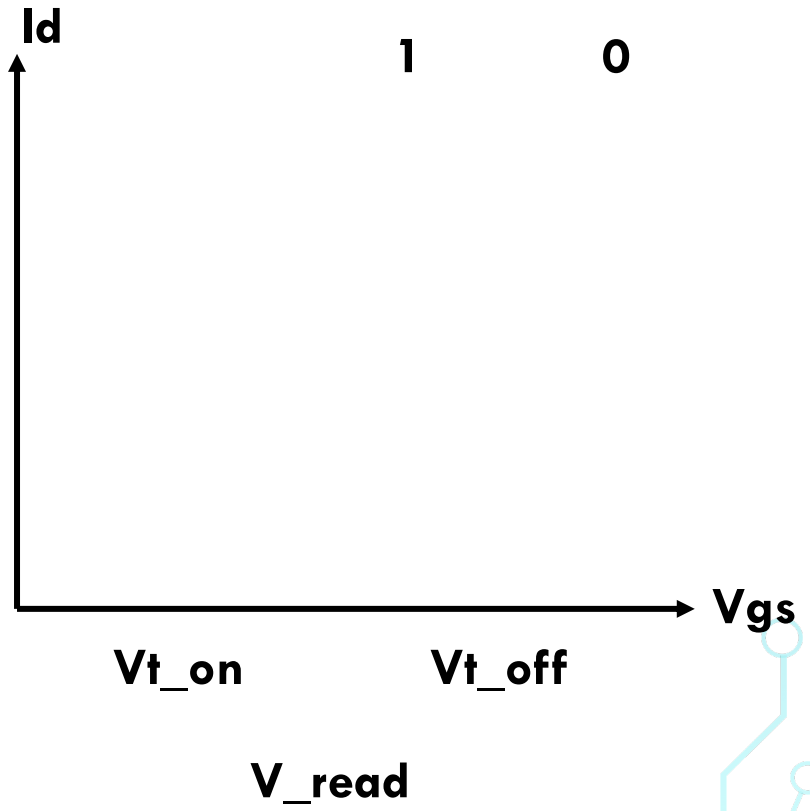
- Floating gate change the threshold voltage of a cell
- Read the cell value by sensing the current



Storing 0
OFF State

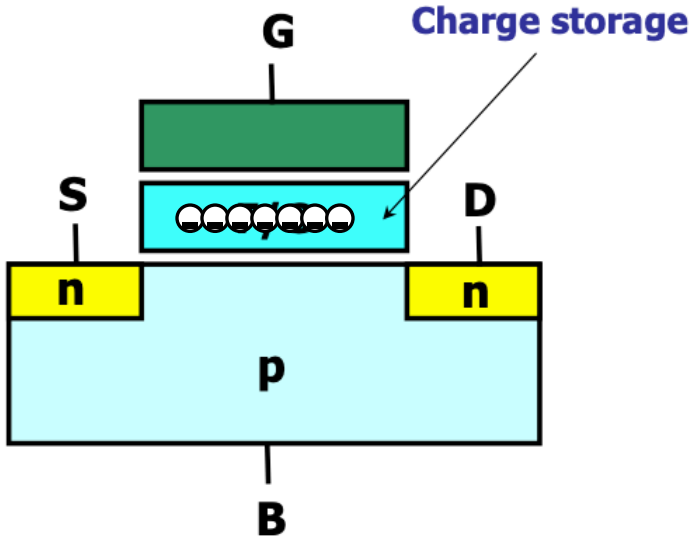


Storing 1
ON State

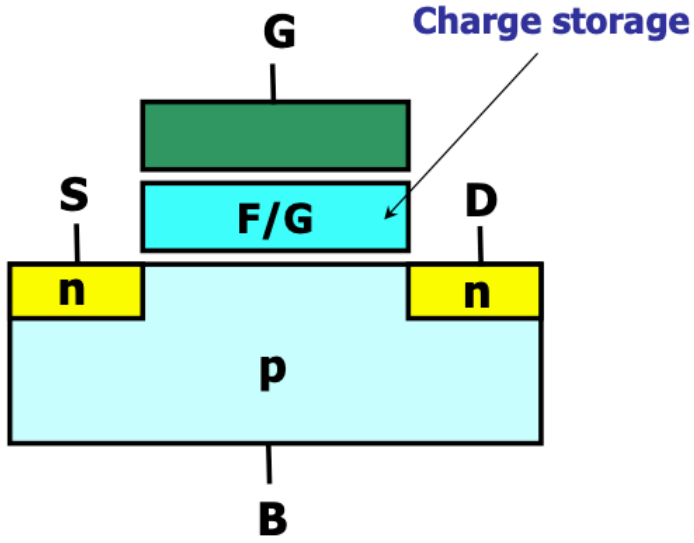


Read a Flash cell

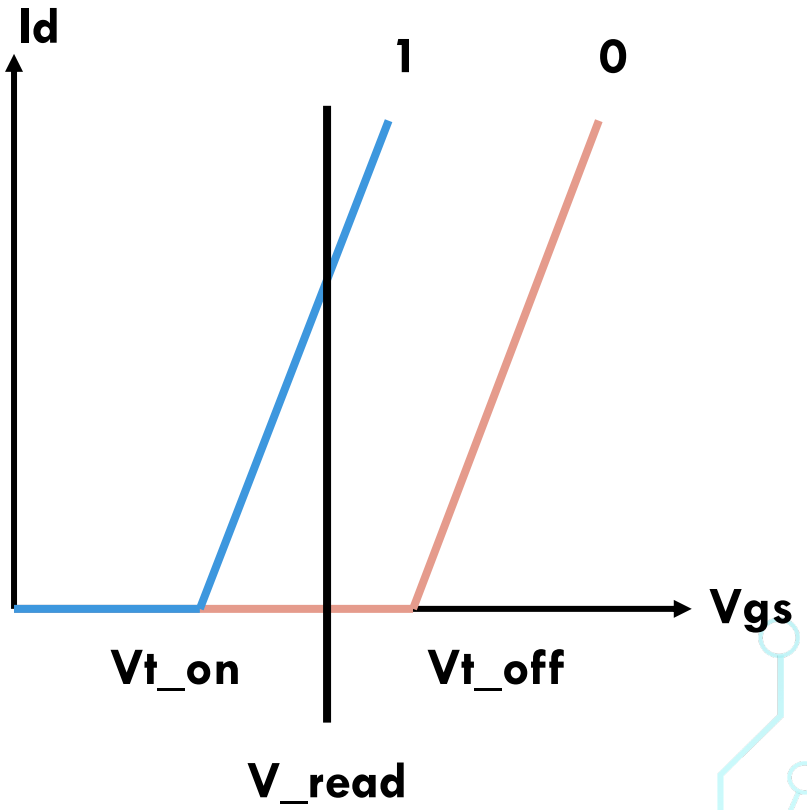
- Floating gate change the threshold voltage of a cell
- Read the cell value by sensing the current



Storing 0
OFF State

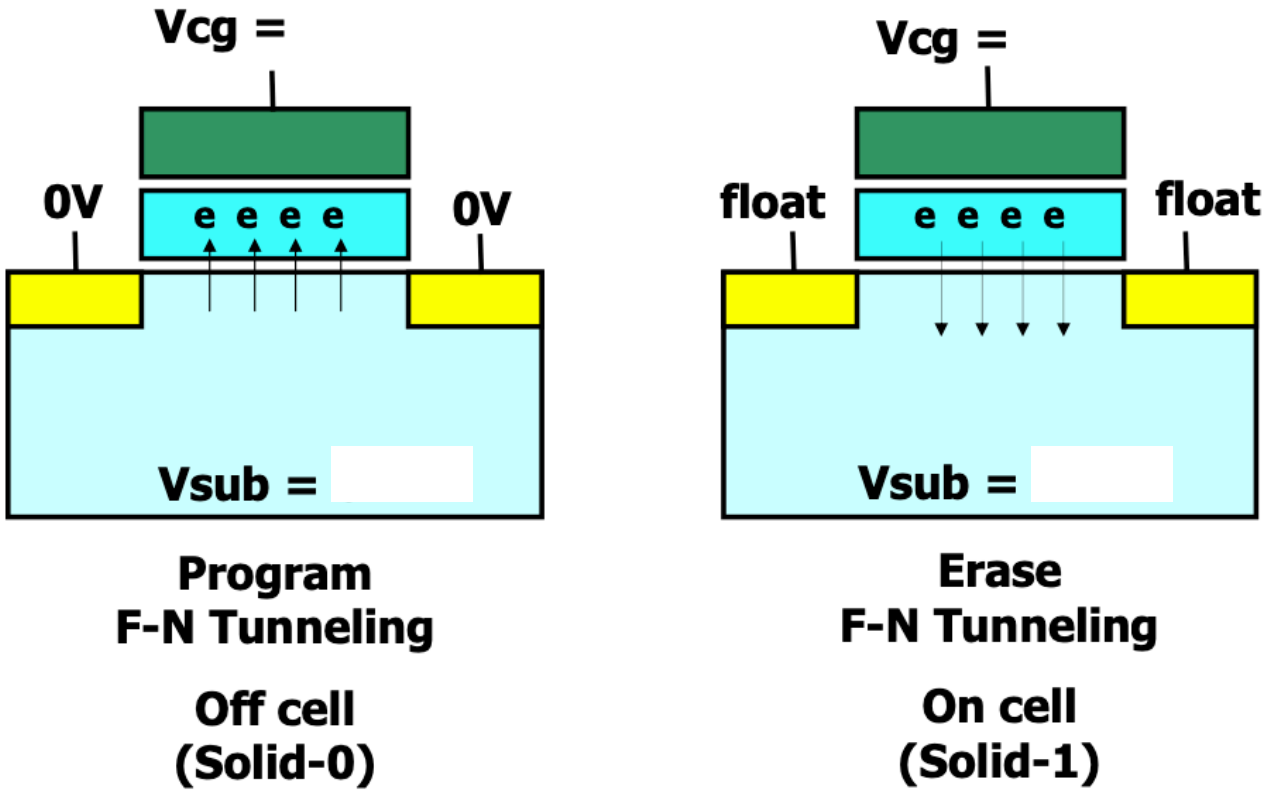


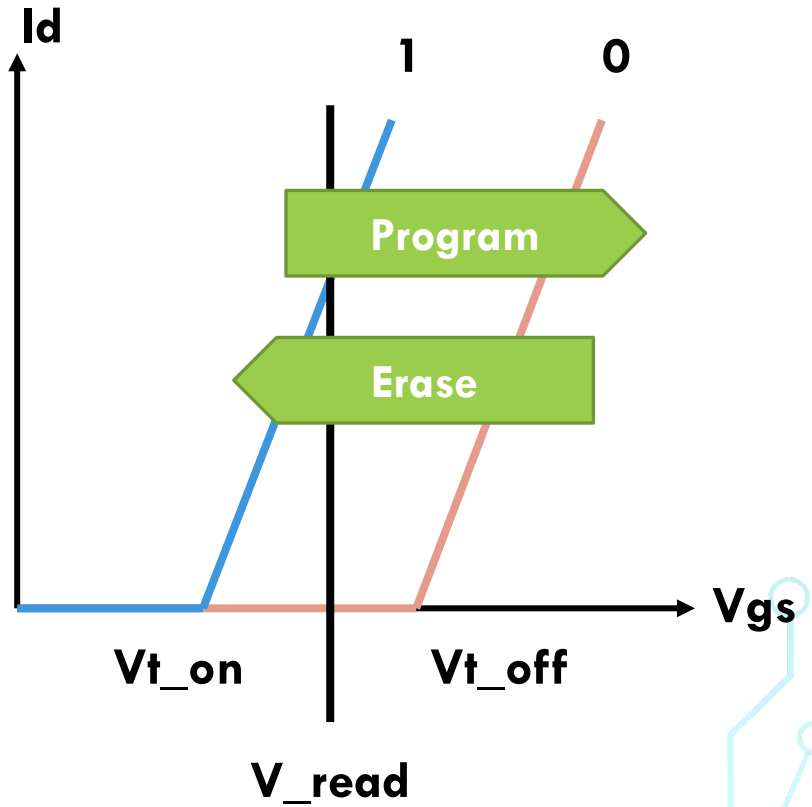
Storing 1
ON State



Write a Flash cell

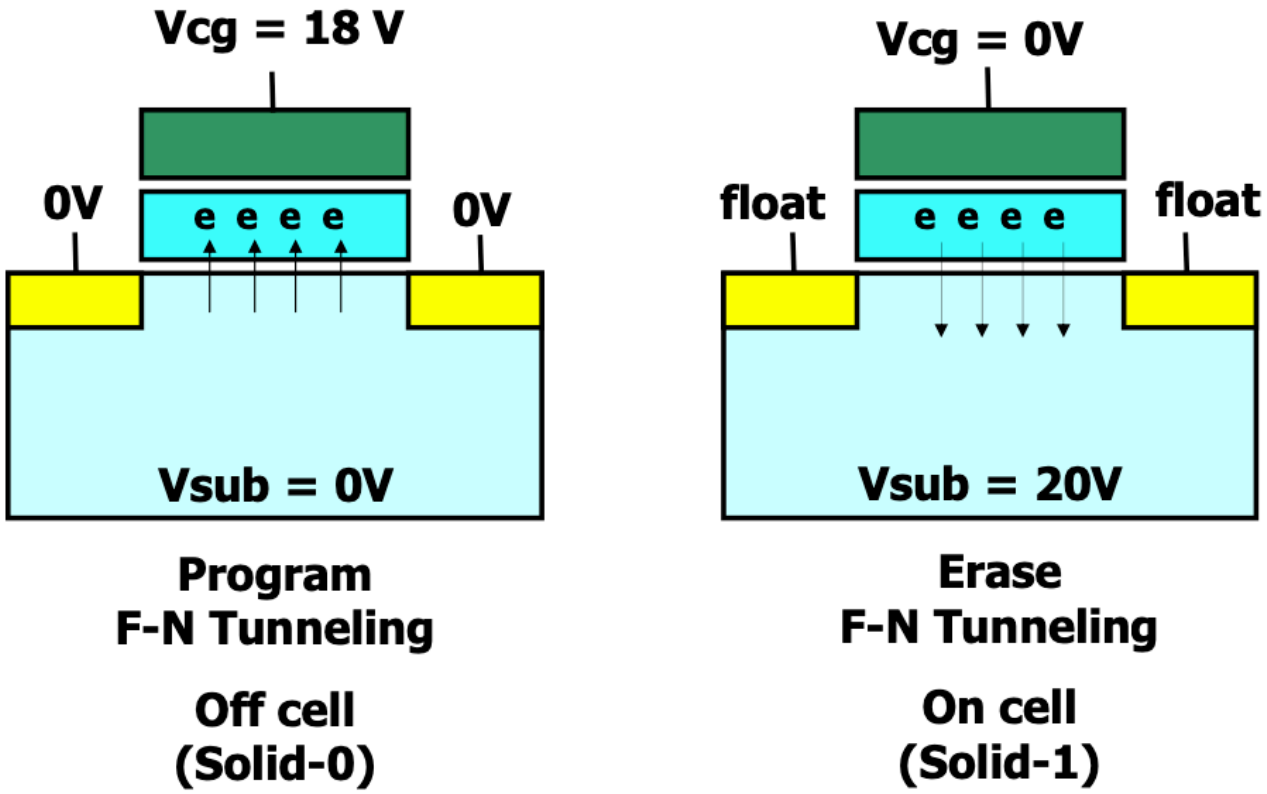
- Write 0: program; Write 1: erase
- Must be erased (storing 1) before reprogrammed.
- Endurance: ~100K erase-program cycles

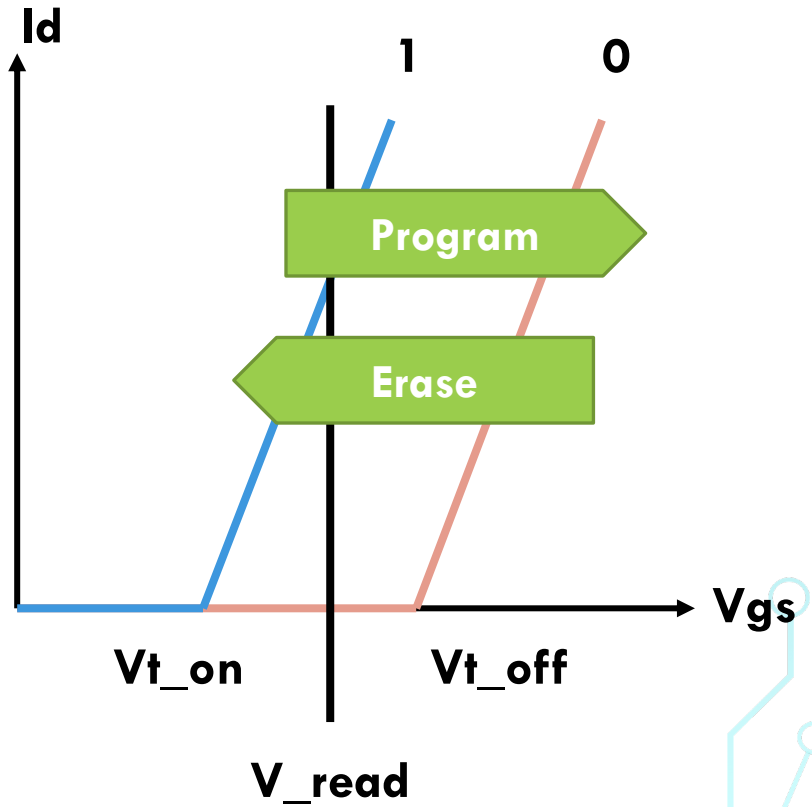




Write a Flash cell

- Write 0: program; Write 1: erase
- Must be erased (storing 1) before reprogrammed.
- Endurance: ~100K erase-program cycles





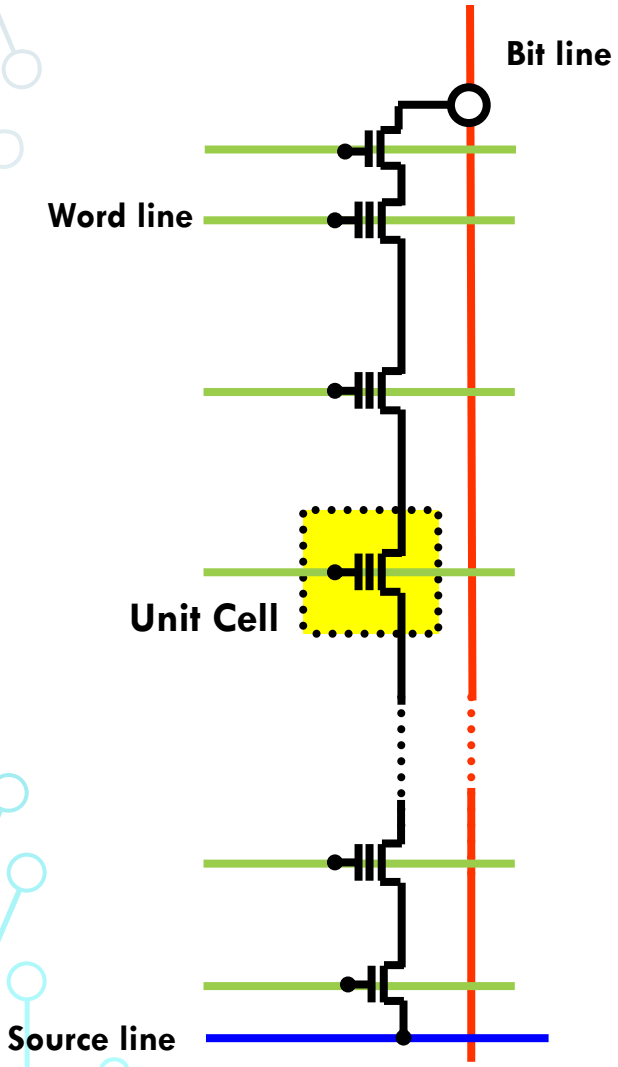
Flash

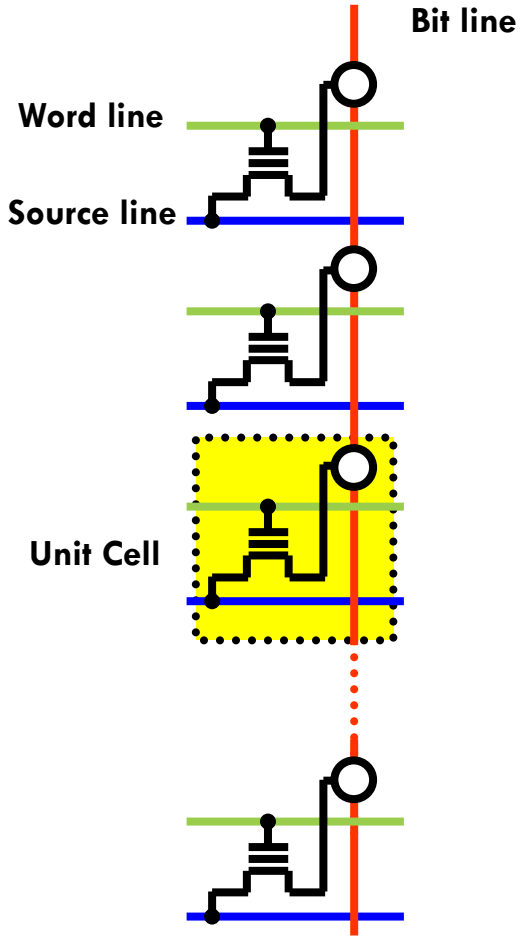
- Overview
- Read & Write Cell
- NAND/NOR Flash

Parallelism

- Limits of Pipelining
- SIMD Processing
- Multithreading

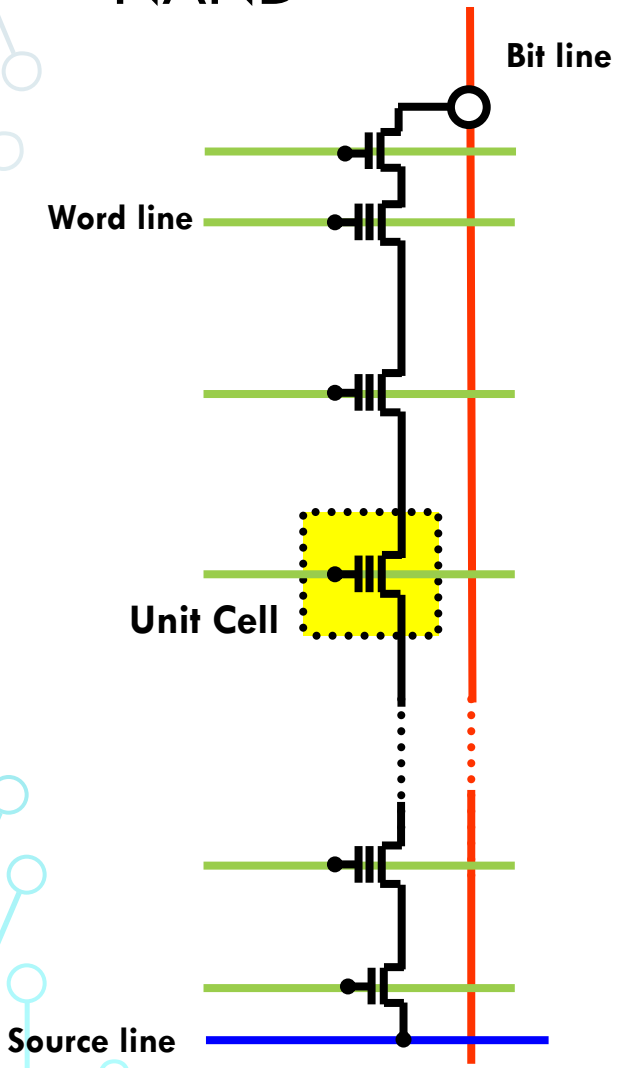
NAND vs NOR Flash





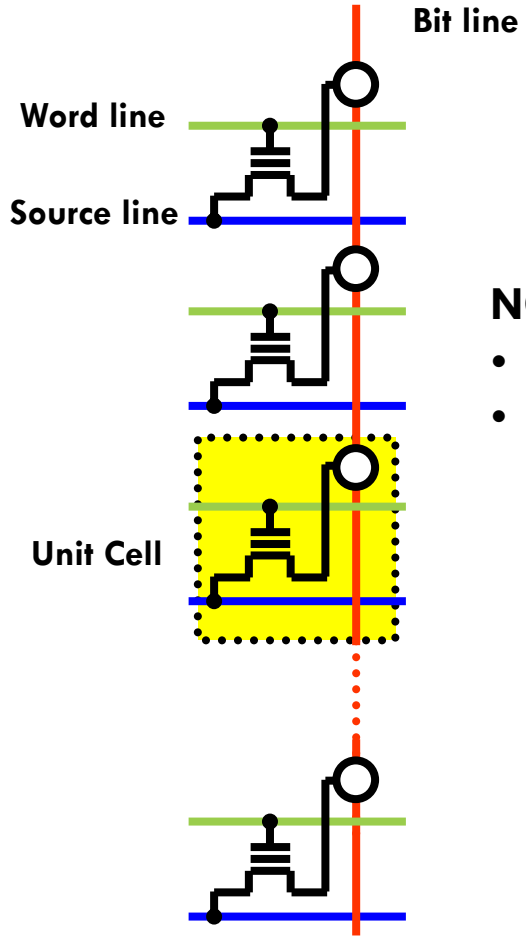
NAND vs NOR Flash

- NAND



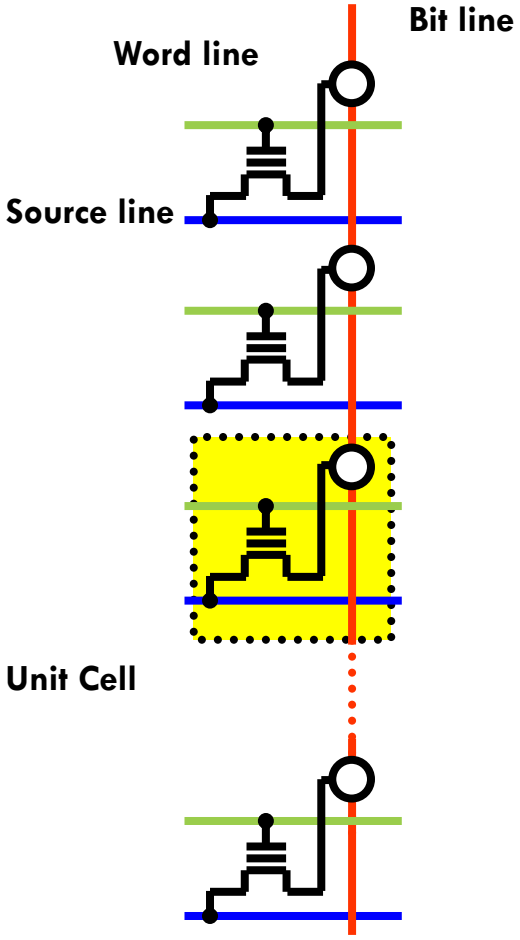
NAND:
- High Density
- Used for data storage
  - USB drives
  - Memory cards
  - SSD

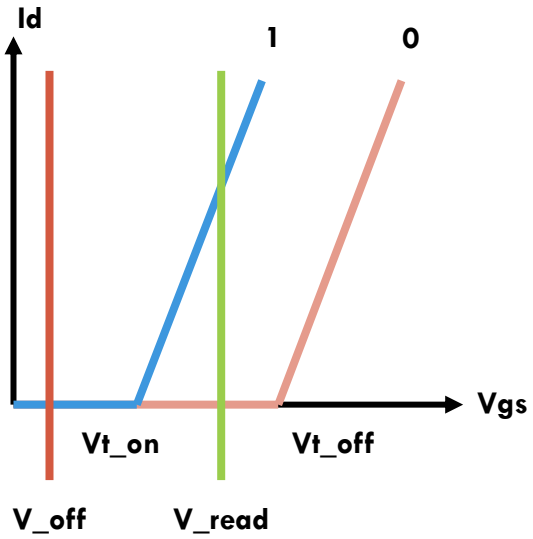
- NOR



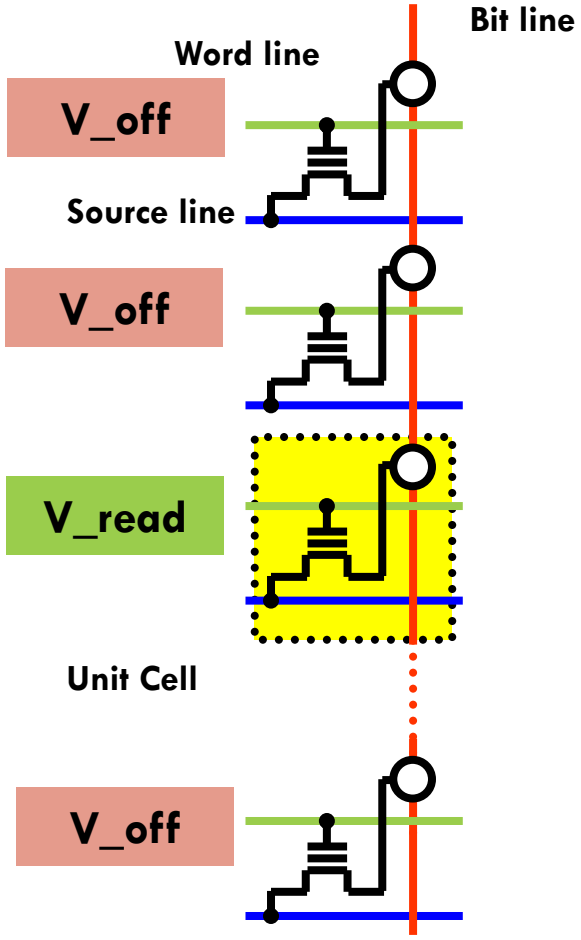
NOR:
- Lower Latency
- Used for code storage
  - Embedded systems

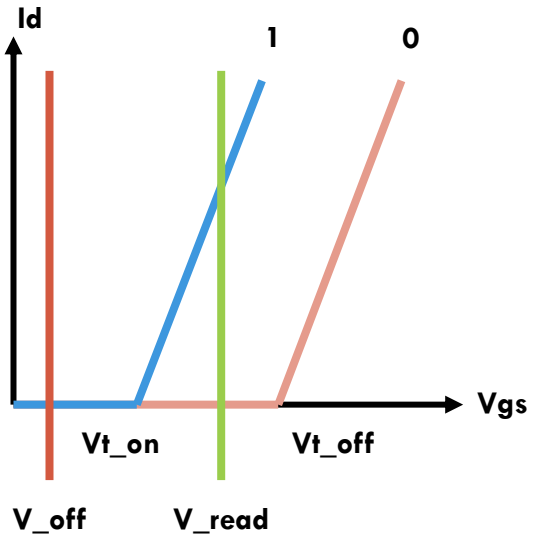
NOR Flash Read



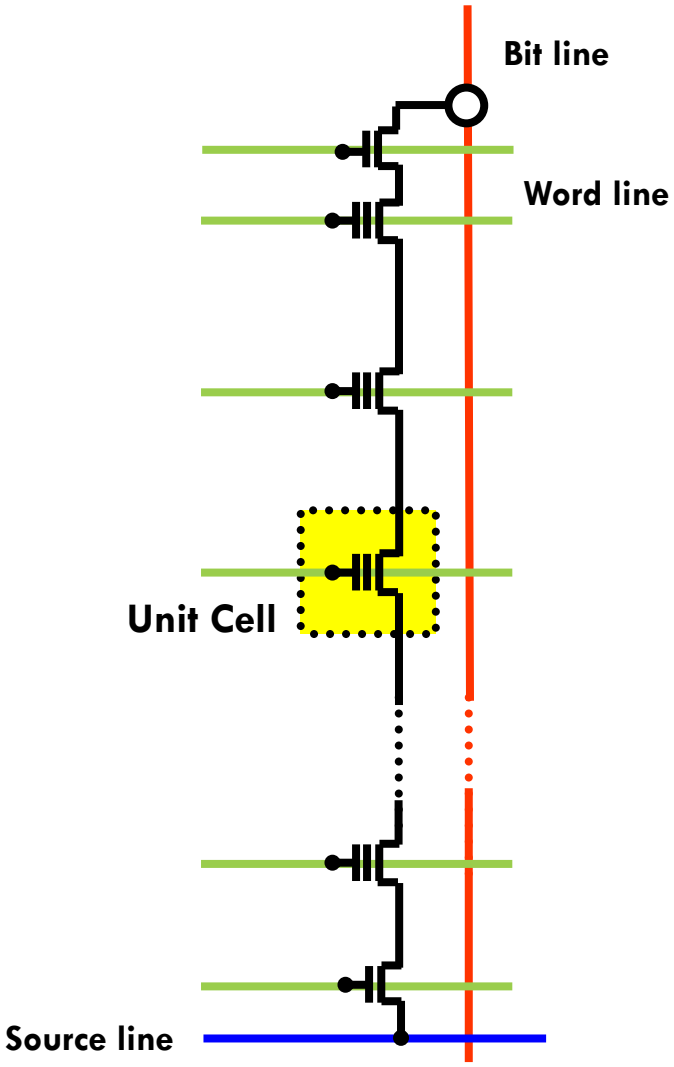


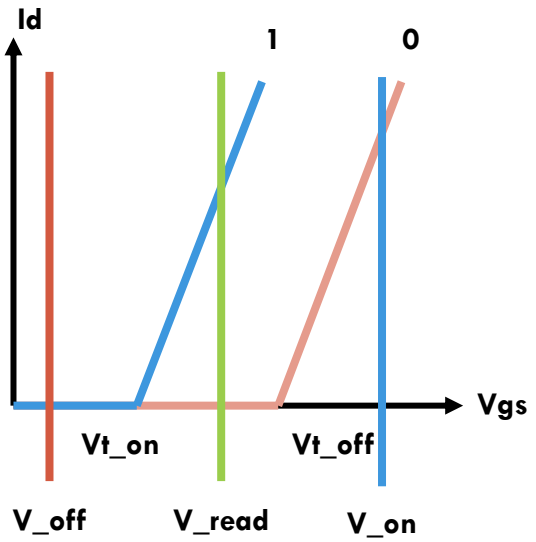
NOR Flash Read



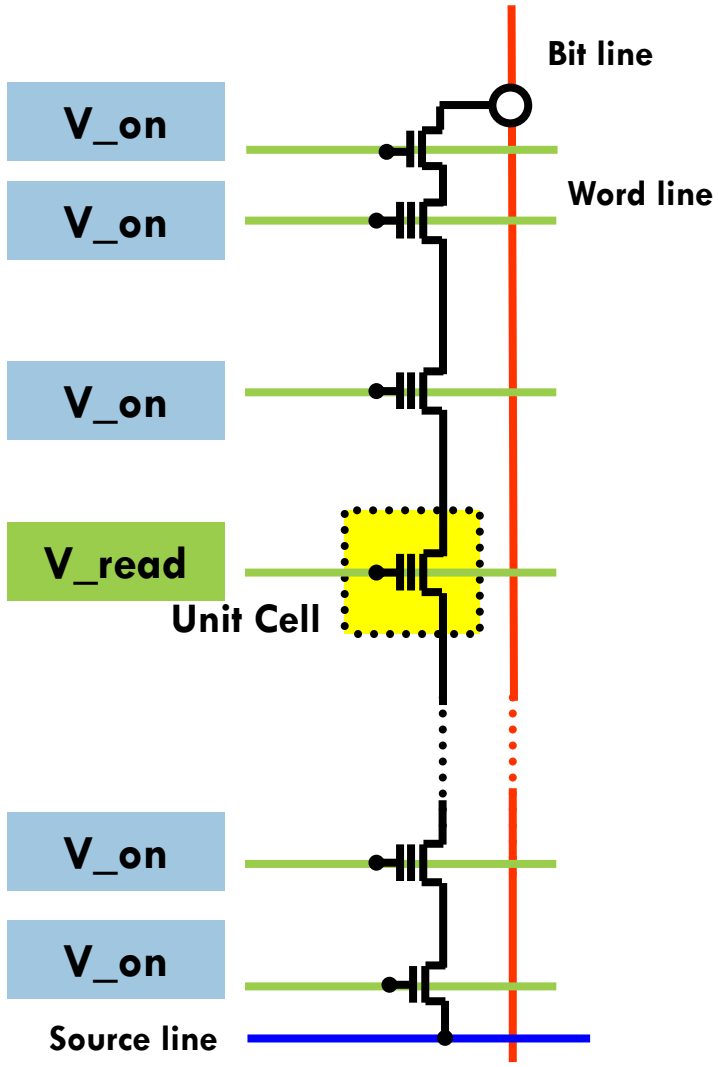


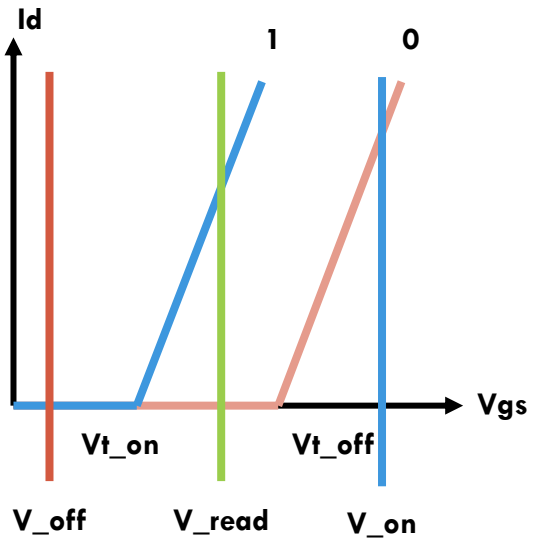
NAND Flash Read





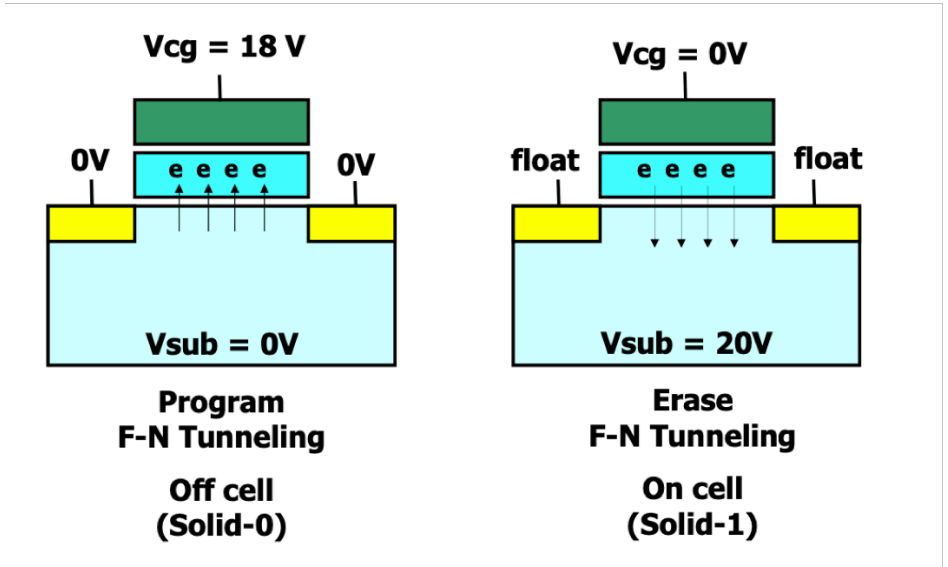
NAND Flash Read

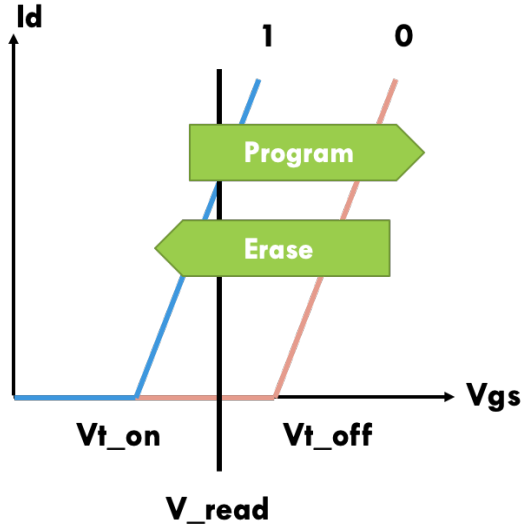




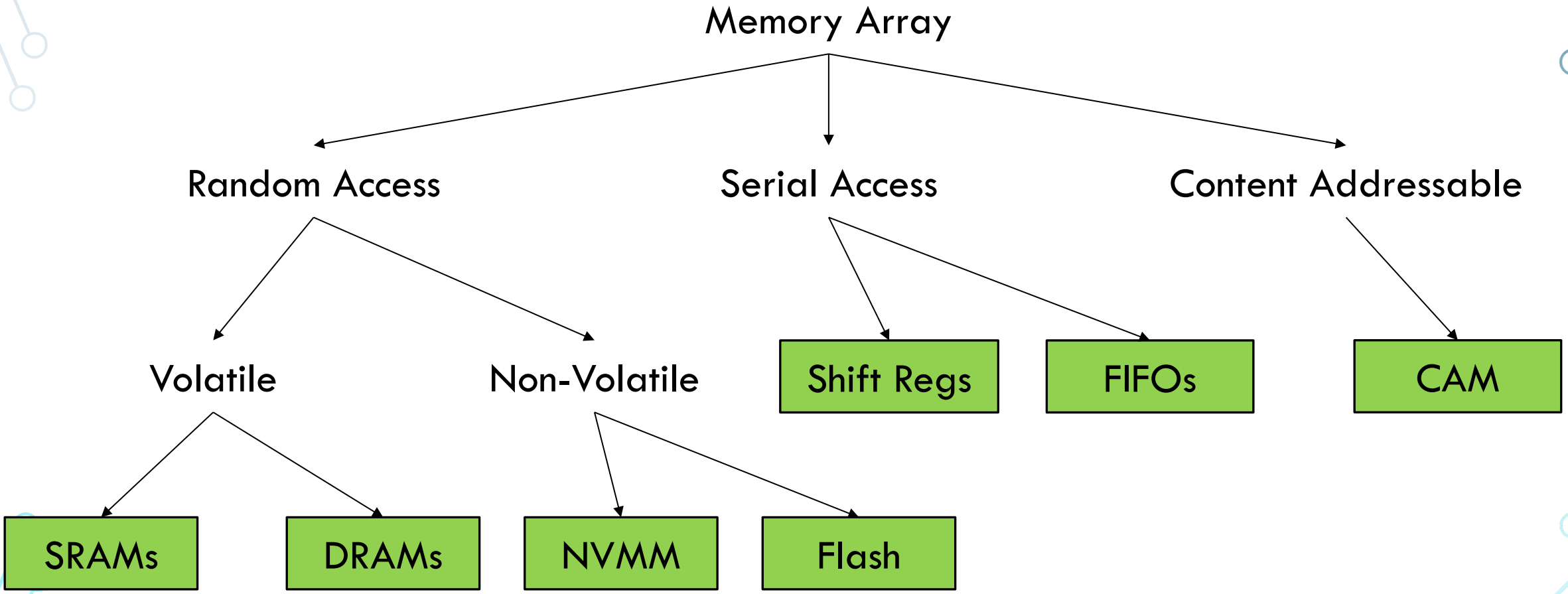
Flash Write

- Step 1: Erasing.
  - Erase all the FG transistors to set them to 1
  - Apply a negative voltage to the gate -> Electrons flow from the floating gate to the substrate.
- Step 2: Programming
  - Reprogram the appropriate FG transistors to set them to 0
  - Apply a high voltage to the gate -> Electrons are tunneled onto the floating gate.



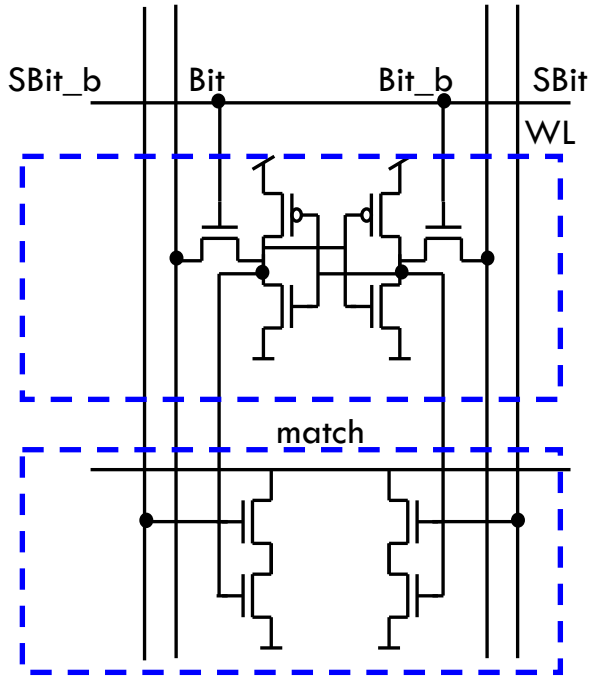


Memory Overview



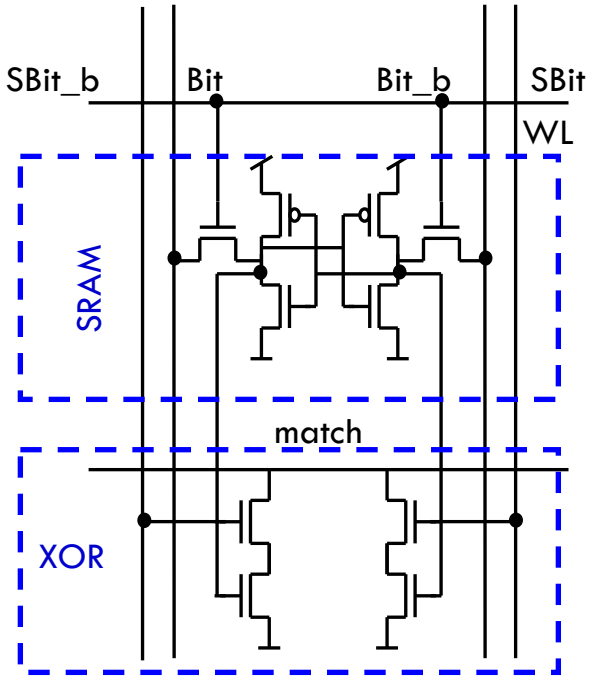
Content Addressable Memory

- Commonly used in translation lookaside buffers (TLBs).
- Matching asserts a matchline output for each world that contains a specified key

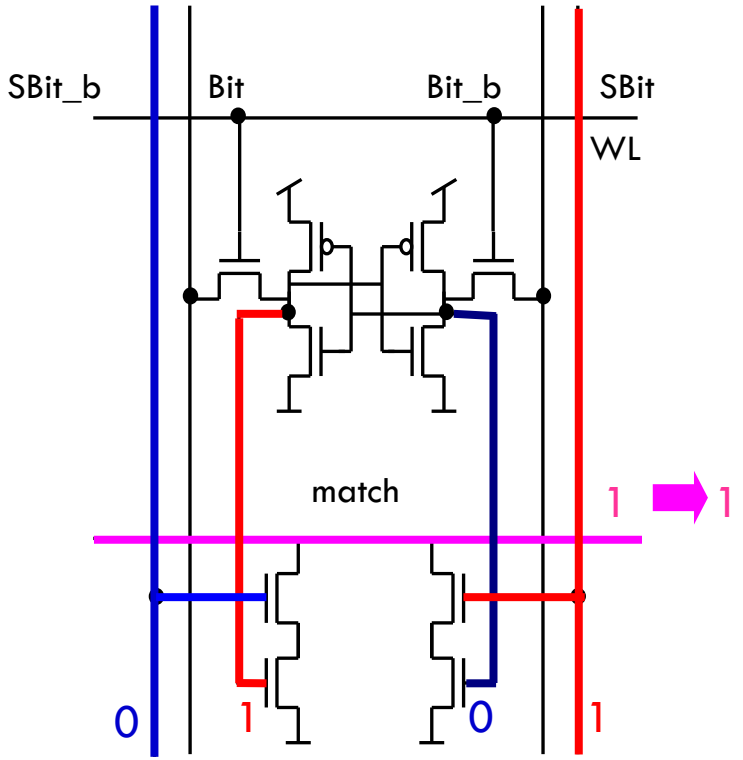


Content Addressable Memory

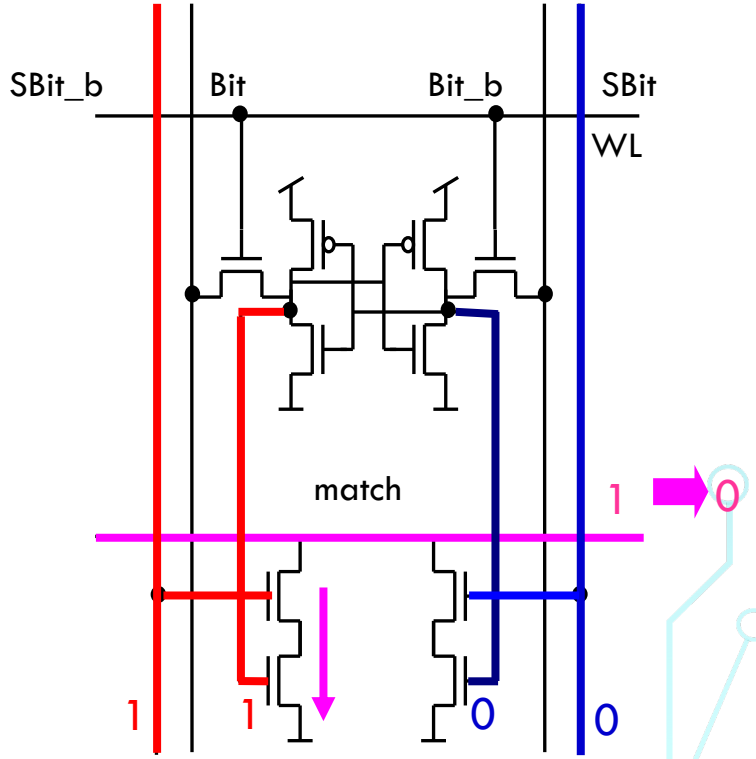
- Commonly used in translation lookaside buffers (TLBs).
- Matching asserts a matchline output for each world that contains a specified key



Match

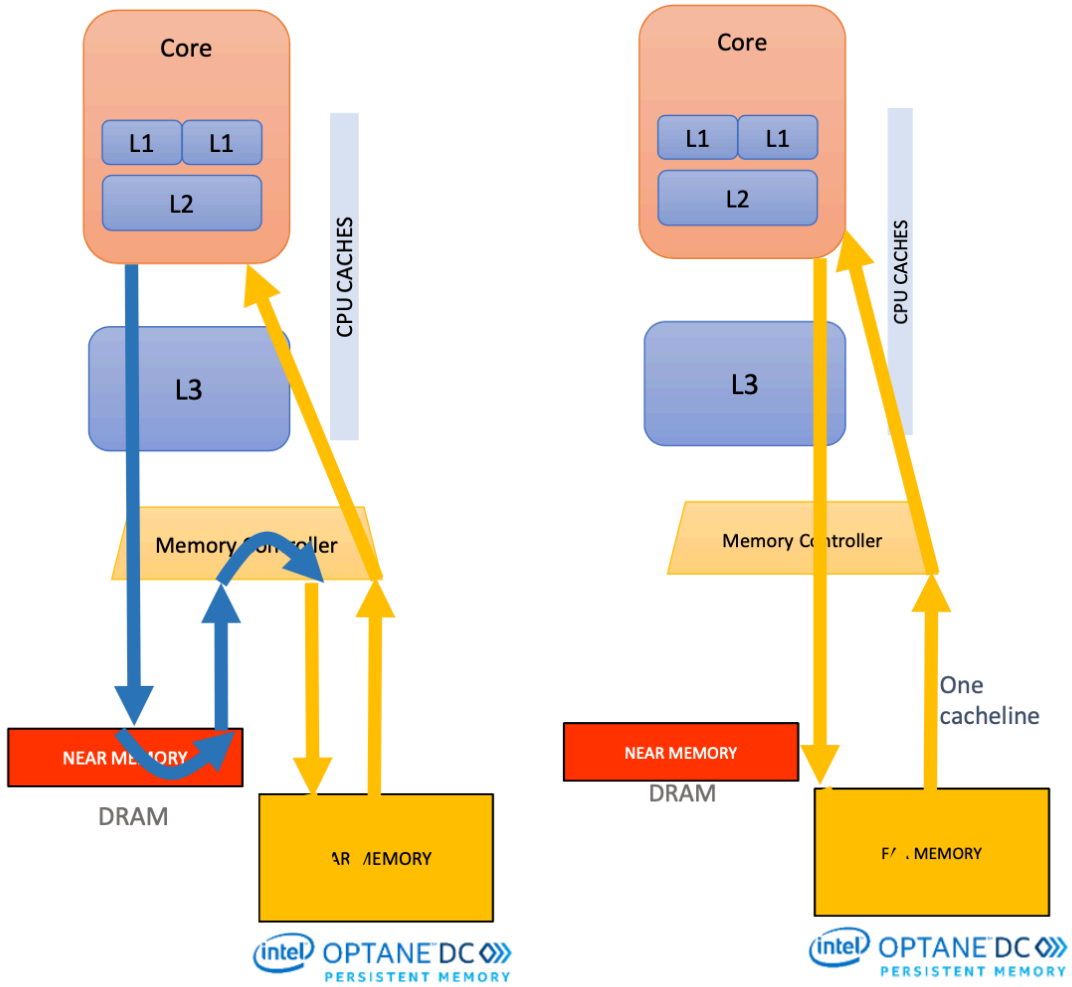


Mismatch

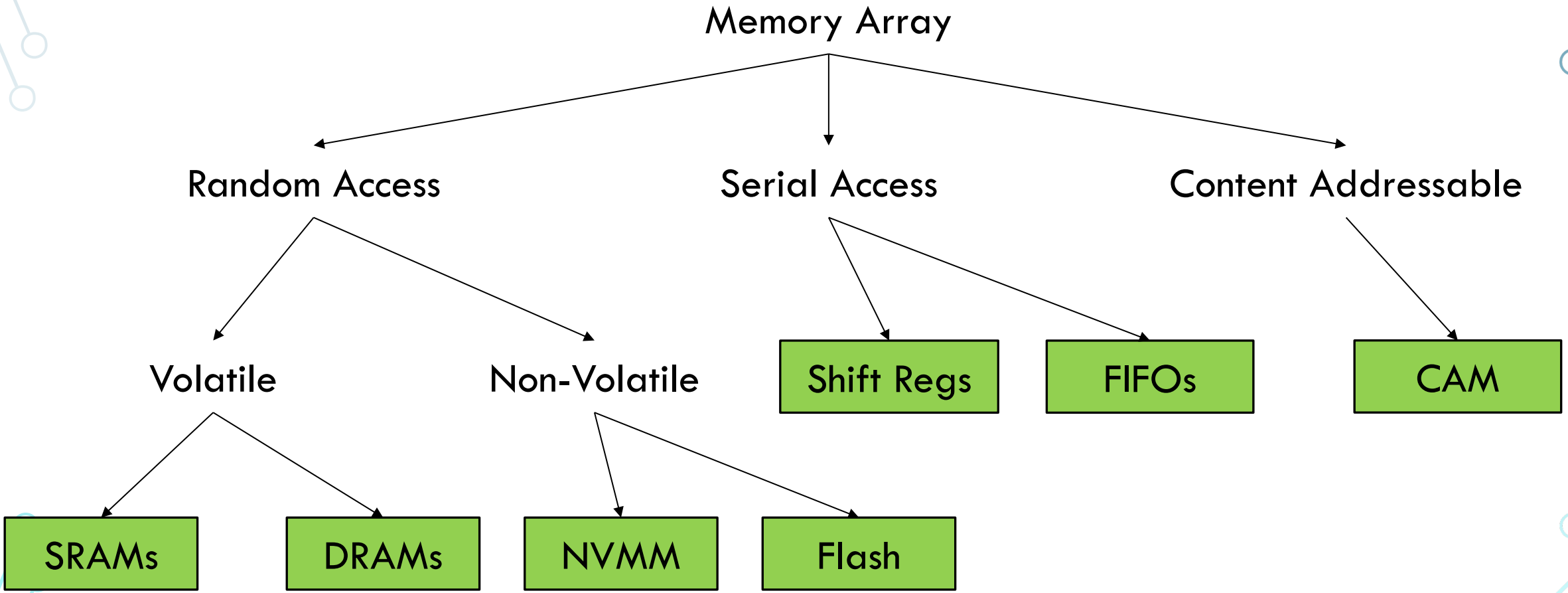


Non-Volatile Main Memory

- Intel Optane DC Persistent Memory
- Non-Volatile
- Storage based on resistance:
  - High resistivity : 0
  - Low Resistivity: 1
- High capacity:
  - 128, 256, 512 GB
- Modes:
  - Memory Mode
  - App-directed Mode



Memory Overview



Administrivia

- No class this Wednesday.
  - Happy Thanksgiving!
- Project, project, project ☺
- Homework 9 due today.
- Homework 10 will be out today and due Dec 6.
- Last day of classes is Dec. 6.

Flash

- Overview
- Read & Write Cell
- NAND/NOR Flash

Parallelism

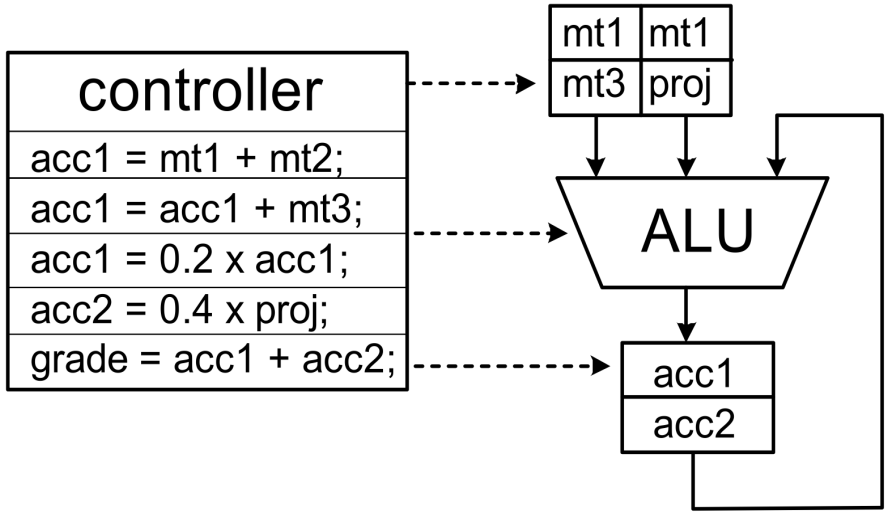
- Limits of Pipelining
- SIMD Processing
- Multithreading

Parallelism

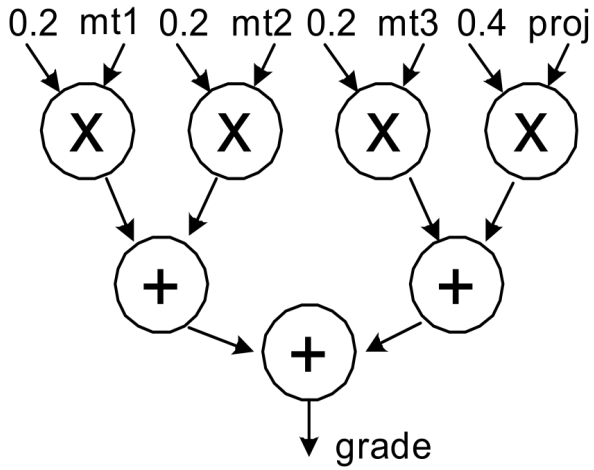
- The act of doing more than one thing at a time.
- Optimization in hardware design often involves using parallelism to trade between cost and performance.
- Can also be used to improve energy efficiency.

```
float mt1, mt2, mt3, project;

grade = 0.2 × (mt1 + mt2 + mt3)
        + 0.4 × project;
```



Sequential



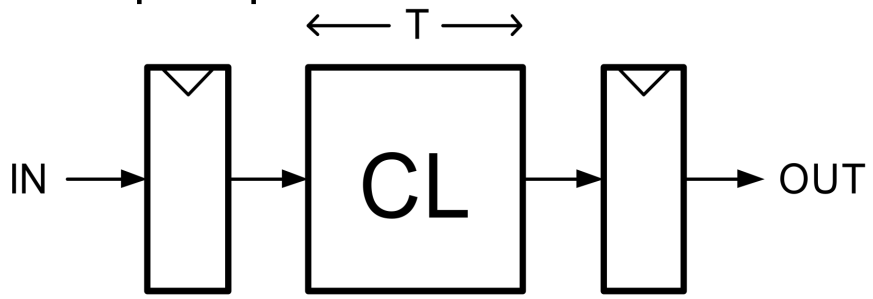
Parallel

Types of Parallelism in Hardware

- Parallelism in Hardware
  - Pipelining
  - SIMD Processing
  - Multithreading

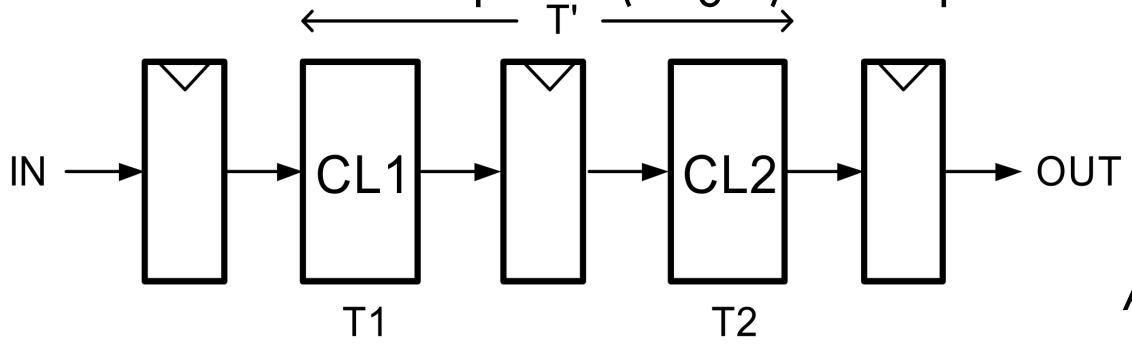
Pipelining

- General principle:



Assume T=8ns
T_{FF}(setup +clk→q)=1ns
F = 1/9ns = 111MHz

- Cut the CL block into pieces (stages) and separate with registers:



Assume T1 = T2 = 4ns

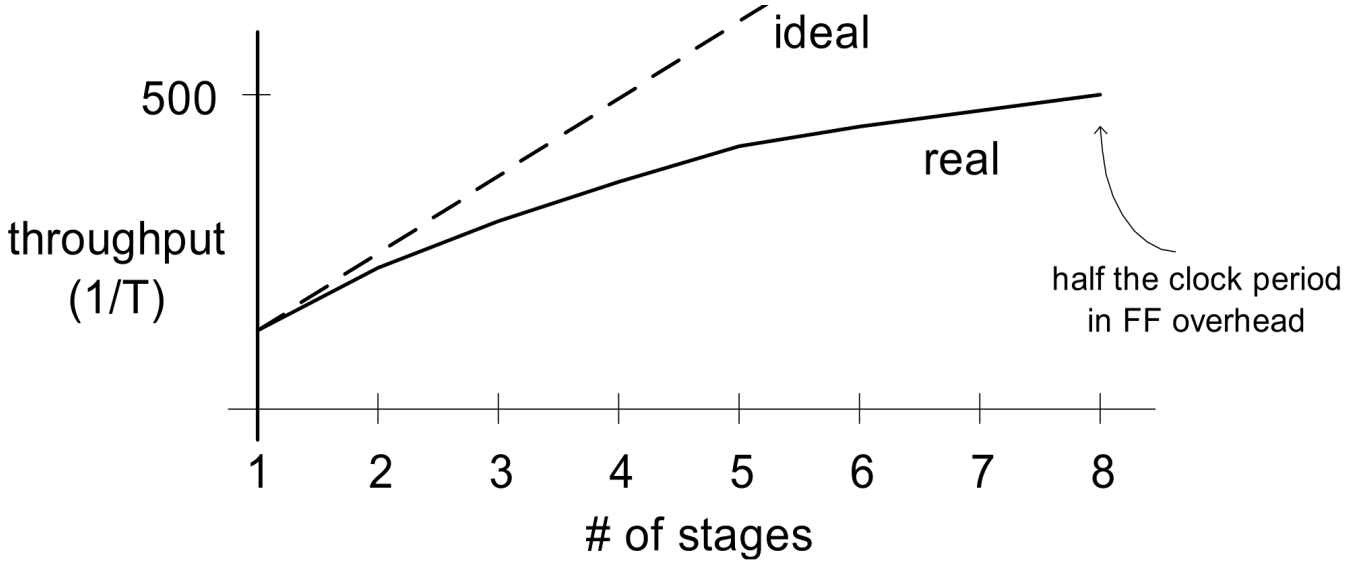
T' = 4ns + 1ns + 4ns +1ns = 10ns

F = 1/(4ns +1ns) = 200MHz

- CL block produces a new result every 5ns instead of every 9ns.

Limits of Pipelining

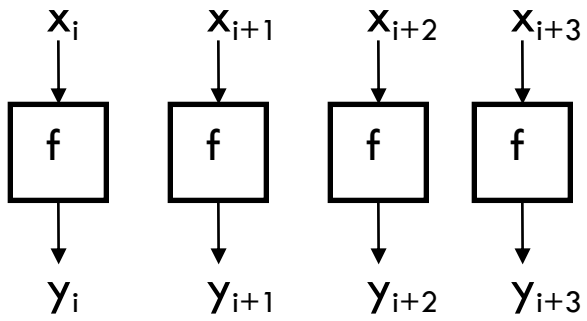
- Without FF overhead, throughput improvement α # of stages.
- After many stages are added FF overhead begins to dominate:



- Other limiters to effective pipelining:
  - clock skew contributes to clock overhead
  - unequal stages
  - FFs dominate *cost*
  - clock distribution power consumption

SIMD Parallelism

- Make multiple instances of the loop execution data-path and run them in parallel, sharing the some controller.
- Example: $y_i = f(x_i)$

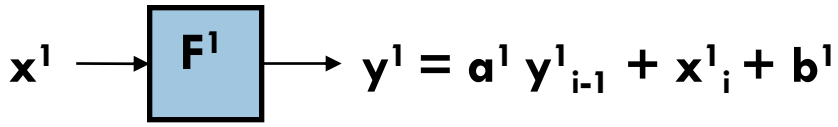


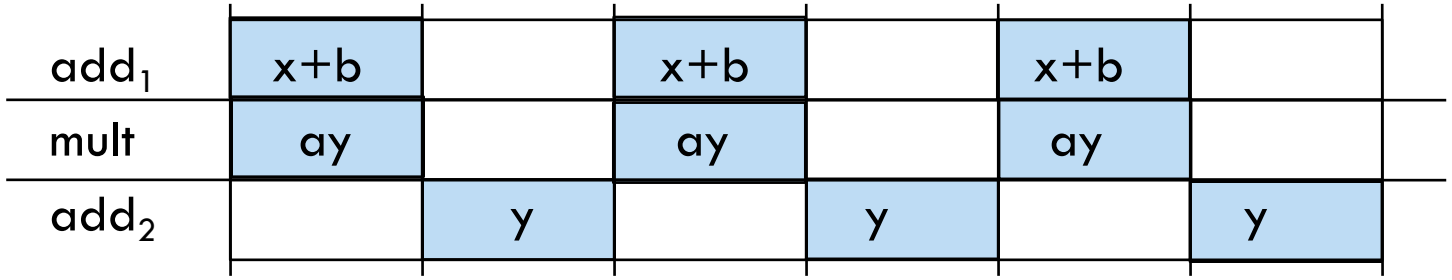
Usually called SIMD parallelism. Single Instruction Multiple Data

- Example: vector processors.

Multithreading

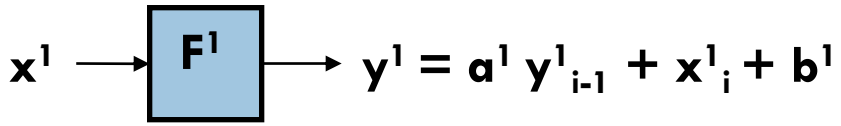
- Fill in "holes" in the pipeline with another (independent) computation

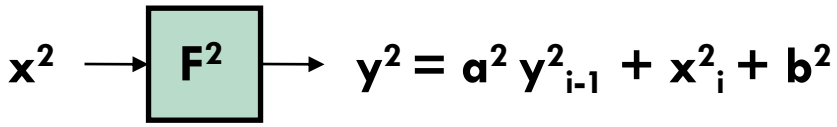




Multithreading

- Fill in "holes" in the pipeline with another (independent) computation.





add_1	x+b	x+b	x+b	x+b	x+b	
mult	ay	ay	ay	ay	ay	
add_2		y	y	y	y	y

- Example: multi-threading in microprocessors

Review

- Flash
  - Single-level vs multi-level
  - Read and Write Flash Cell
  - NAND vs NOR
- Parallelism in Hardware
  - Pipelining
  - SIMD Processing
  - Multithreading