CEND 3420 Computer Organization & Design Lecture 12: Memory

Textbook: Chapter 5.1-5.2, A.8-A.9

Zhengrong Wang

CSE Department, CUHK

zhengrongwang@cuhk.edu.hk

Introduction

Analogy of Memory Hierarchy

- Suppose you are working on an essay about the history of computer.
  - You search for some related topic from internet using Google.
  - You download some most related materials on your laptop to read in detail.
  - You opened one document and start to read it.

- This forms a memory hierarchy!
  - Internet → Large (almost infinite), but slow secondary memory.
  - Laptop → Medium main memory.
  - Screen → Small, but fast access cache.

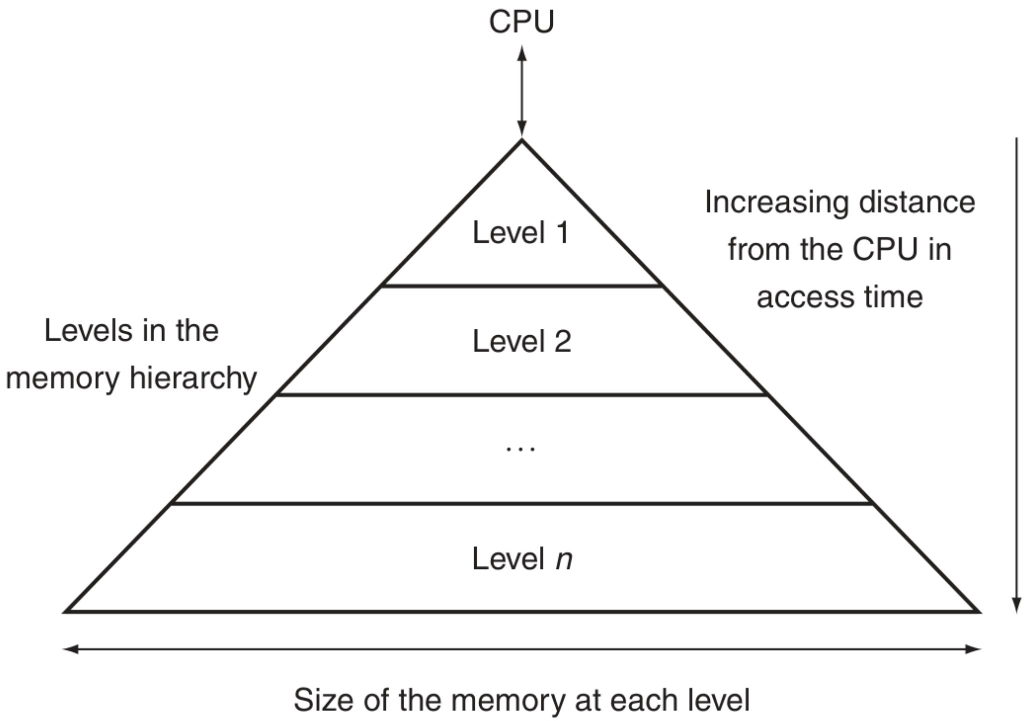
- Modern computers also integrate such memory hierarchy.

Computer Memory Hierarchy

- Memory hierarchy in your laptop with typical capacity and latency:
  - Register file (1-2KiB, 1 cycle).
  - L1 cache (64-128KiB, 1-4 cycles).
  - L2 cache (256-512KiB, 10 cycles).
  - L3 cache (1-2MiB, 50 cycles).
  - Main memory (16-32GiB, 100 cycles).
  - SSD (256GiB-1TiB, 10^5 cycles).

- Notation:

	Value	Decimal		Value	Decimal
1K	10^3	1000	1Ki	2^10	1024
1M	10^6	1000000	1Mi	2^20	1048579
1G	10^9	1000000000	1Gi	2^30	1,073,741,824


CPU
Level 1
Level 2
...
Level *n*
Increasing distance from the CPU in access time
Levels in the memory hierarchy
Size of the memory at each level

Why We Need Memory Hierarchy?

- Processor usually runs much faster than main memory:
  - Small memories are fast, large memories are slow.
  - Use a cache to store data in the processor that is likely to be used.
- Main memory is limited:
  - Use virtual memory to increase the apparent size of physical memory by moving unused sections of memory to disk (automatically).
  - A translation between virtual and physical addresses is done by a memory management unit (MMU).
  - To be discussed in later lectures.
- How computer memory works: YouTube

Why Memory Hierarchy Works?

- Temporal Locality (locality in time):
  - If a memory location is referenced, it will tend to be referenced again soon.
  - Keep most recently accessed data items closer to the processor.

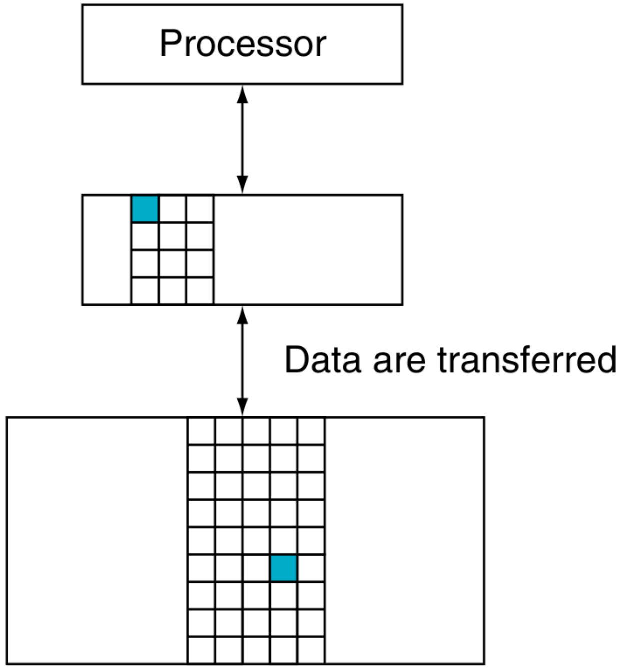
- Spatial Locality (locality in space):
  - If a memory location is referenced, the locations with nearby addresses will tend to be referenced soon.
  - Move blocks consisting of contiguous words closer to the processor.

- Example:
  - sum follows temporal locality.
  - array[i] follows spatial locality.

```
for (int i = 0; i < 100; i++)
{
    sum += array[i];
}
```

Terminology

- Random access memory (RAM)
    - Comparable access time for any memory locations.
- Block (line): The minimum unit of data that can be either present or not in the cache.
- Hit rate: The fraction of accesses found in a memory hierarchy level.
- Miss rate: 1 – Hit rate.
- Hit time: The time required to access a memory hierarchy level, including the time needed to determine whether the access is a hit or a miss.
- Miss penalty: The time required to fetch a block into a memory hierarchy level from the lower level, including the time to access the block, transmit it from one level to the other, insert it in the level that experienced the miss, and then pass the block to the requestor.



Recap Throughput vs. Latency

- Latency
  - The time between the start and the completion of a memory access.
  - May stall the pipeline until data is fetched from memory hierarchy.
- Throughput (bandwidth)
  - The total amount of data transferred in a given time, in unit of GB/s, TB/s etc.
  - What is the bandwidth of the main memory/GPU VRAM on your laptop/desktop?
- Example: My laptop.
  - Windows task manager reports 6400MT/s (6400×10^6 transactions per second).
  - CPU-Z reports DDR5 with 4 32-bit channels.
  - Total bandwidth: #channels x channel width x transactions per second

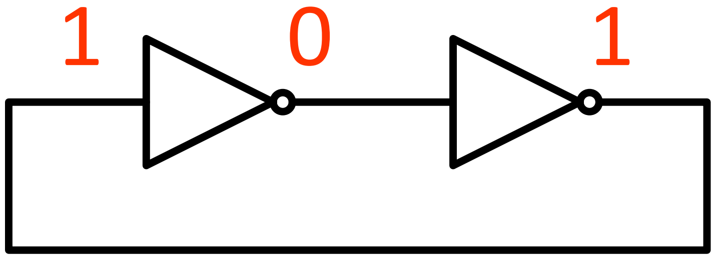
$$4 \times 4\text{Byte} \times 6400 \times 10^6 = 102.4\text{GB/s}$$

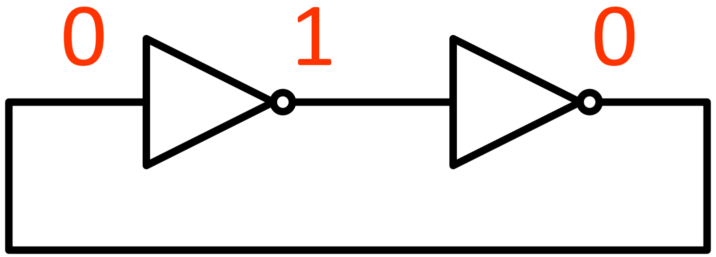
- GPU usually has much higher bandwidth, e.g., RTX5090 with 1.8TB/s

Latch and Flip-Flop

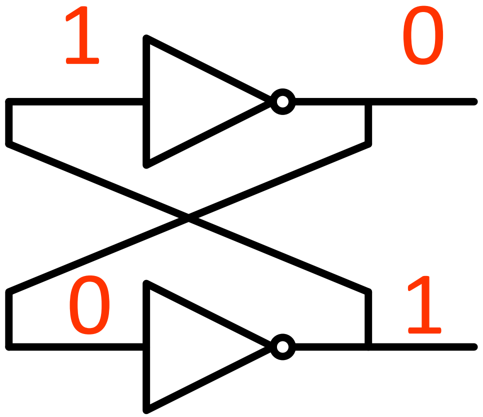
Store 1 Bit with A Pair of Inverters

- Add feedback to a pair of inverters.



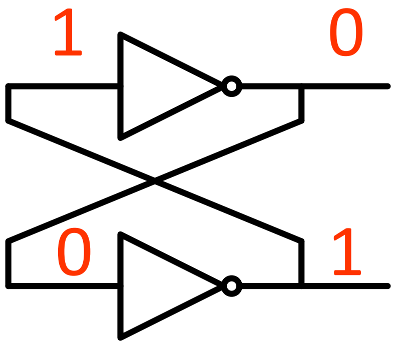


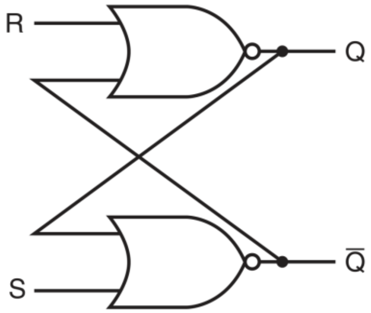
- Usually drawn as cross-coupled inverters.
- Stores 1-bit information.



How to Change the Stored Value?

- Replace inverter with NOR gate.
- SR-Latch (Set and Reset).





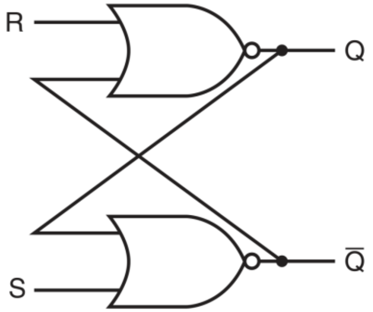
- Given the NOR gate truth table:

R ($Reset$)	S (Set)	Q	$\bar{Q}$
0	0		
1	0		
0	1		
1	1		

A	B	NOR(A, B)
0	0	1
0	1	0
1	0	0
1	1	0

SR-Latch

- Replace inverter with NOR gate.
- SR-Latch (Set and Reset).



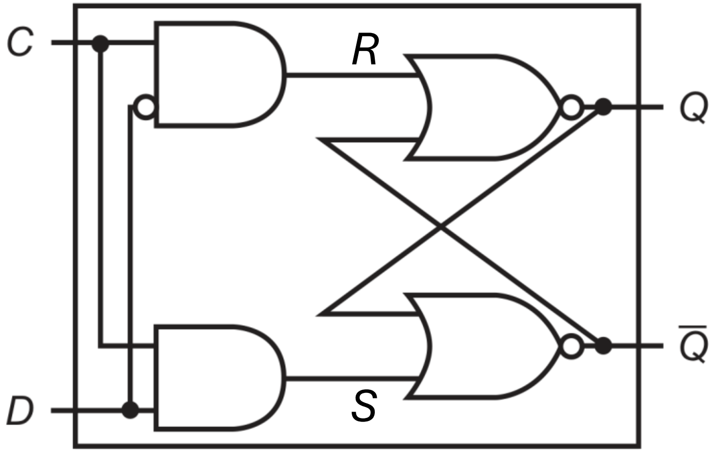
- Given the NOR gate truth table:

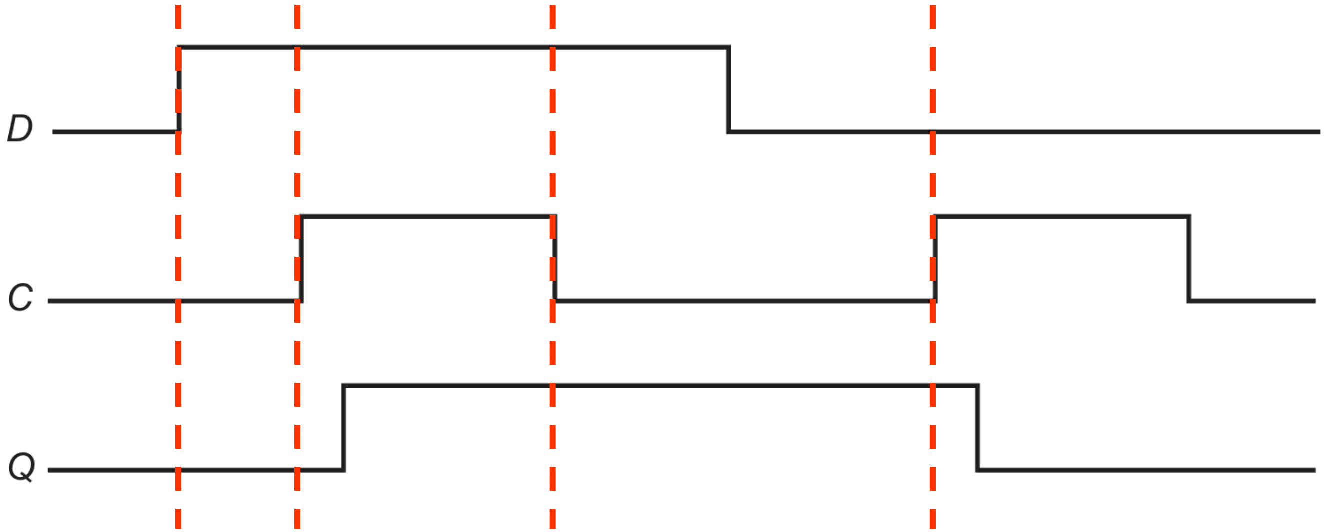
$\boldsymbol{R}$ (***Reset***)	$\boldsymbol{S}$ (***Set***)	$\boldsymbol{Q}$	$\boldsymbol{\bar{Q}}$
0	0	Q (unchanged)	$\bar{Q}$ (unchanged)
1	0	0	1
0	1	1	0
1	1	0 (invalid)	0 (invalid)

A	**B**	**NOR(A, B)**
0	0	1
0	1	0
1	0	0
1	1	0

D Latch

- Add two more AND gates to construct D Latch (D for data).
  - Two inputs: C (clock) and D (data).
  - Set Q = D when C = 1.
  - When C = 0, the AND gate ensures that $R = S = 0$, Q does not change.
  - When C = 1, $R = \overline{D}, S = D$, $Q = D$ (Verify with the previous page).





C=0,
Q unchanged

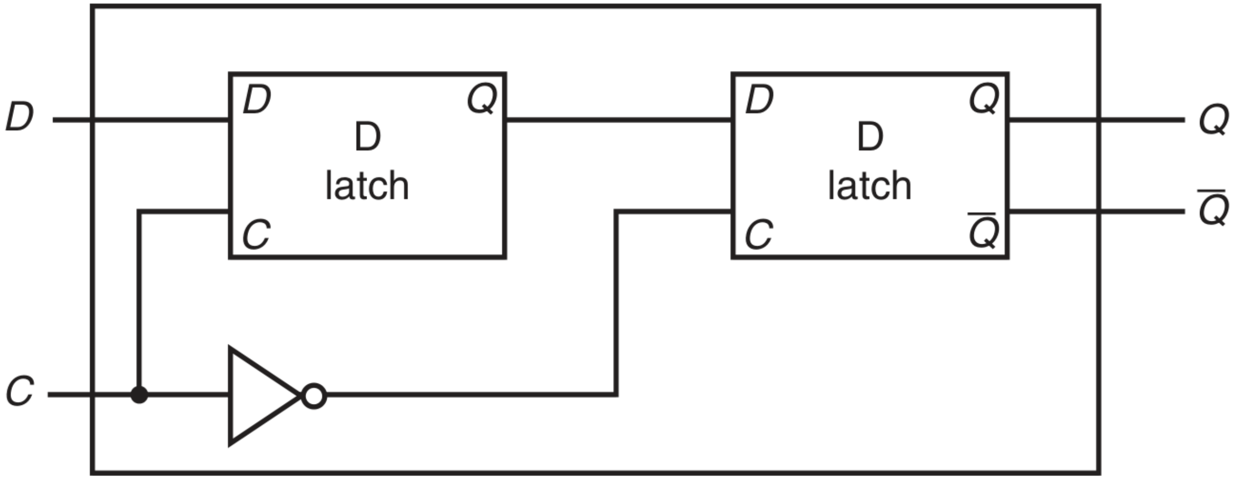
C=1,
Q =D

C=0,
Q unchanged

C=1,
Q =D

D Flip-Flop

- D Flip-Flop: Only set Q=D (data) at edges of C (clock).
  - In stable state (C=0 or C=1), one of the D latch is blocked, Q unchanged.
- Example: Negative edge triggered D flip-flop.
  - At negative edge (C=1 → C=0), since the NOT gate on C has a short delay t.
    - At T=0, the first D latch sees C=0, and locks D value in its output Q.
    - At T=t, the second D latch sees C=1 (inverted), and locks D value in its output.
  - At positive edge (C=0 → C=1), Q is unchanged.



Latch vs. Flip-Flop

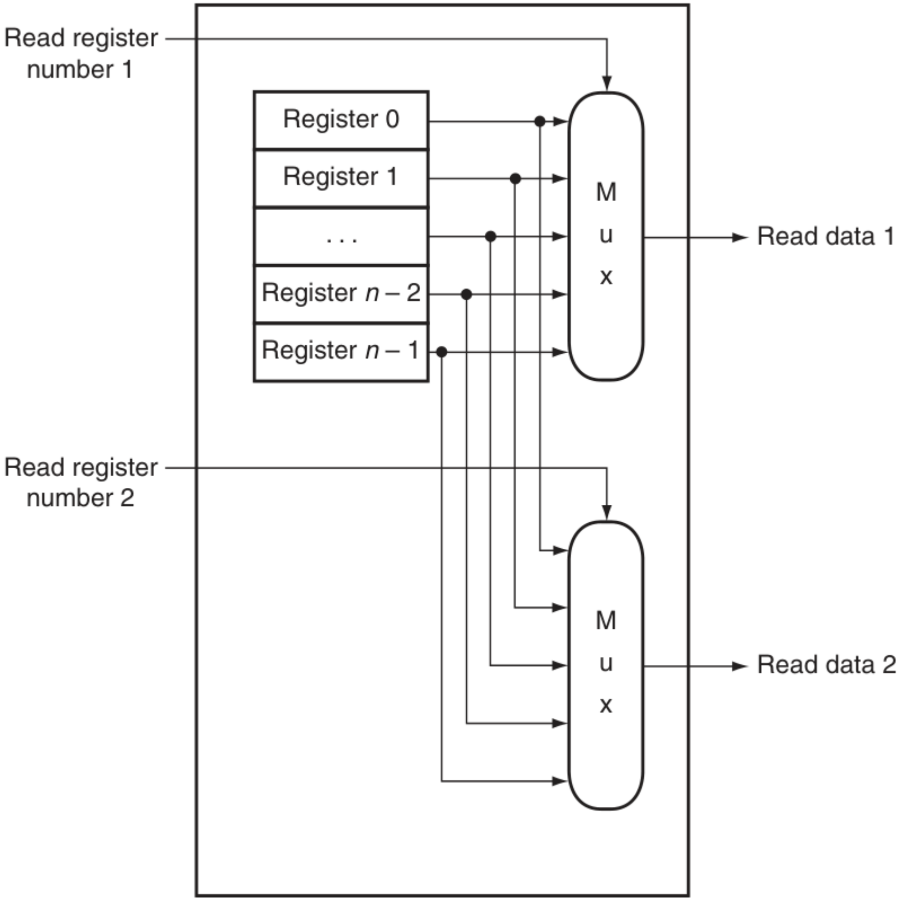
- Latch: Level-sensitive.
  - A memory element in which the output is equal to the value of the stored state inside the element and the state is changed whenever the appropriate inputs change and the clock is asserted.
- Flip-Flop: Edge-sensitive.
  - A memory element for which the output is equal to the value of the stored state inside the element and for which the internal state is changed only on a clock edge.

- For this course, we only use flip-flops as the basic building block.
  - Many other way to build latch and flip-flop.
  - If you are interested, check Wikipedia.

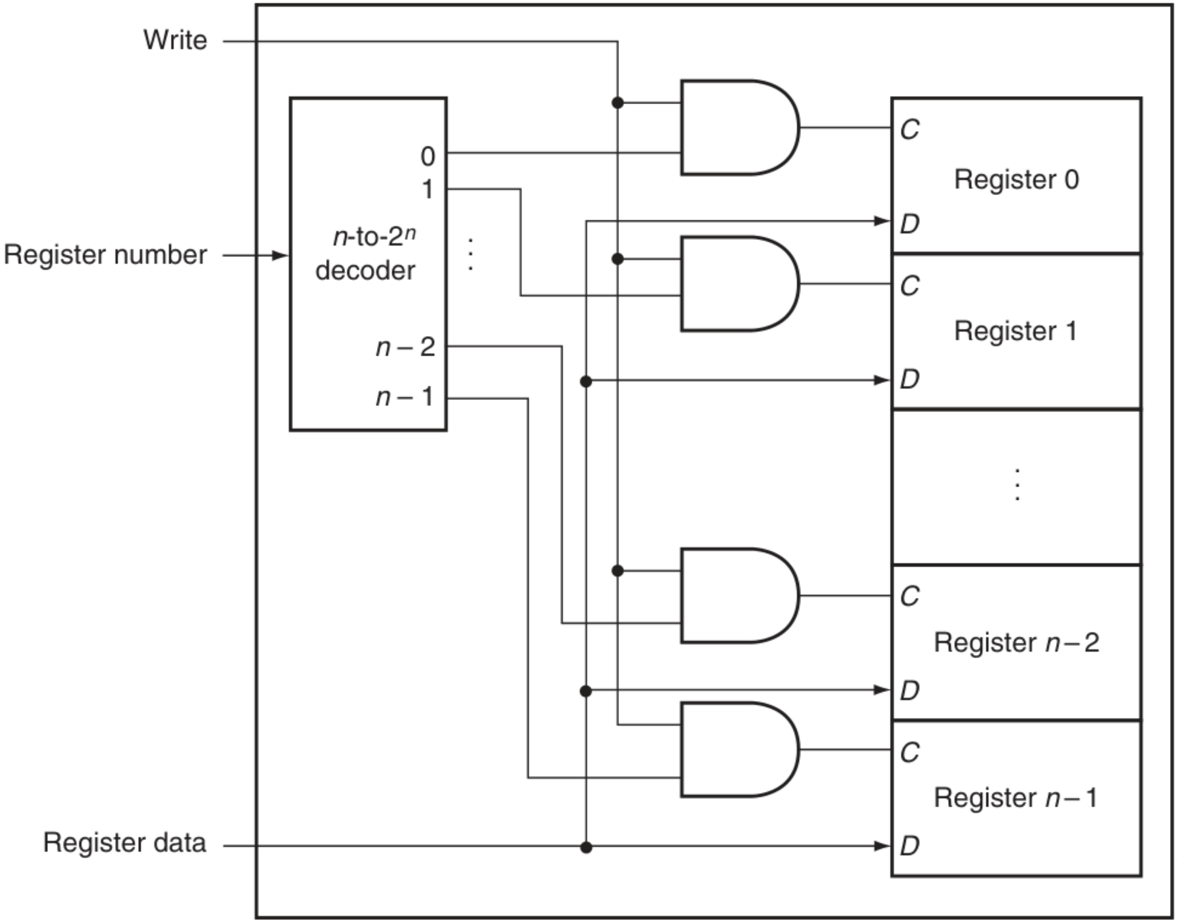
Register File

- Register file can be implemented with D flip-flop, mux, and write decoder.

Mux to select read value.



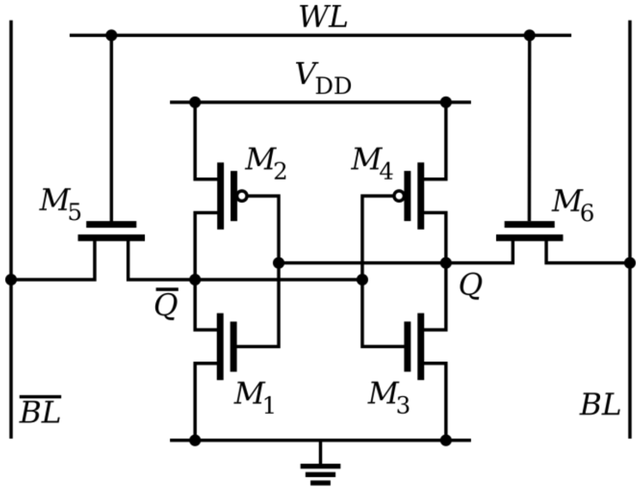
Decoder and Write signal to write a specific register.



Random Access Memory (RAM)

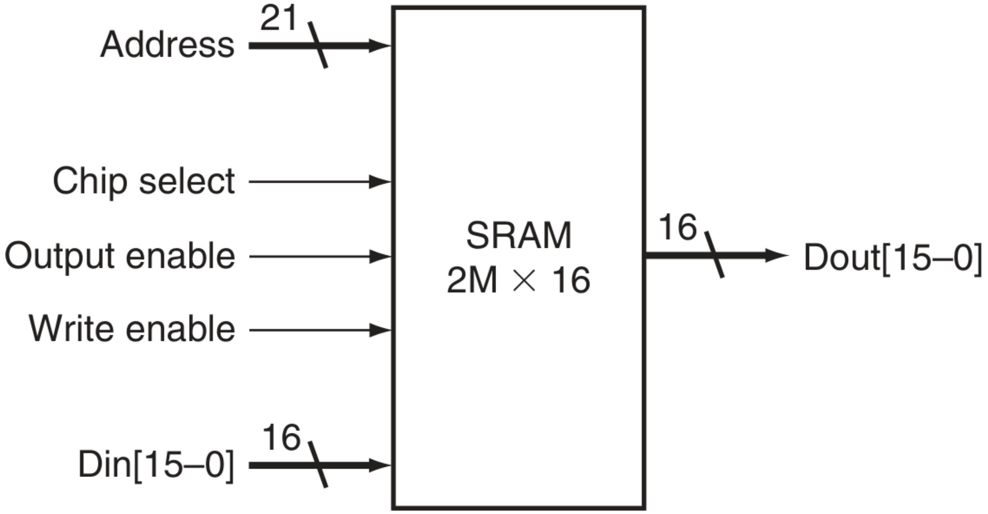
Static Random Access Memory (SRAM)

- Implemented with flip-flops.
  - Static: holds the data permanently unless losing power (volatile).
  - Fast but expensive in terms of silicon area and cost.
  - Usually used for cache and internal registers.
- Typical 6 transistor SRAM cell (1-bit).
  - Data stored in M1-M4, forming two cross-coupled inverters.
  - M5-M6 control access during read or write.

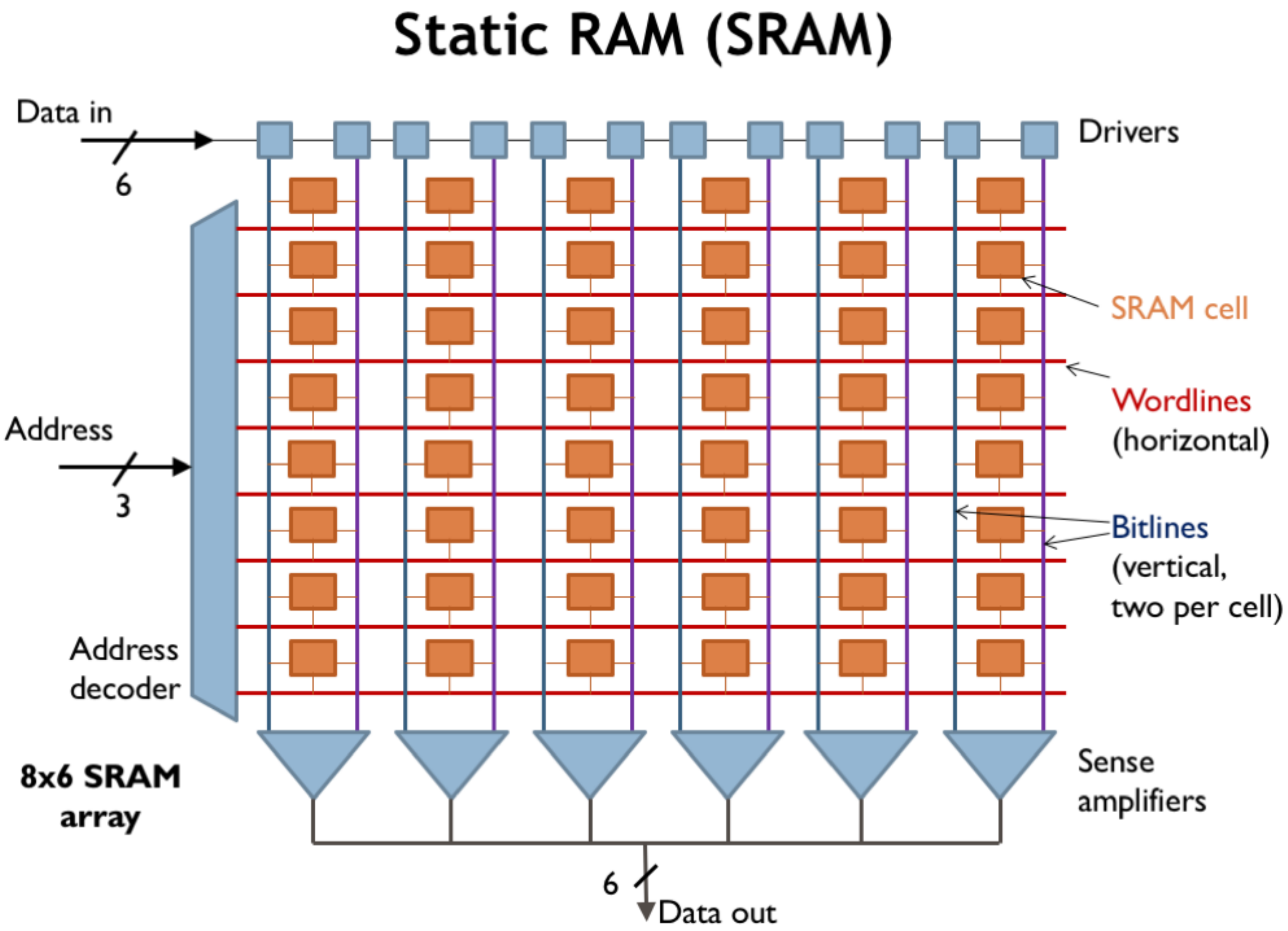


SRAM Organization

- SRAM is organized as array of $\mathrm{P} \times \mathrm{Q}$ entries.
  - P is number of entries; Q is entry width in bits.
  - E.g., 2M x 16 has 2×10^6 entries (16-bit), total 4MB.
  - Address requires $\log_2 P$ bits.



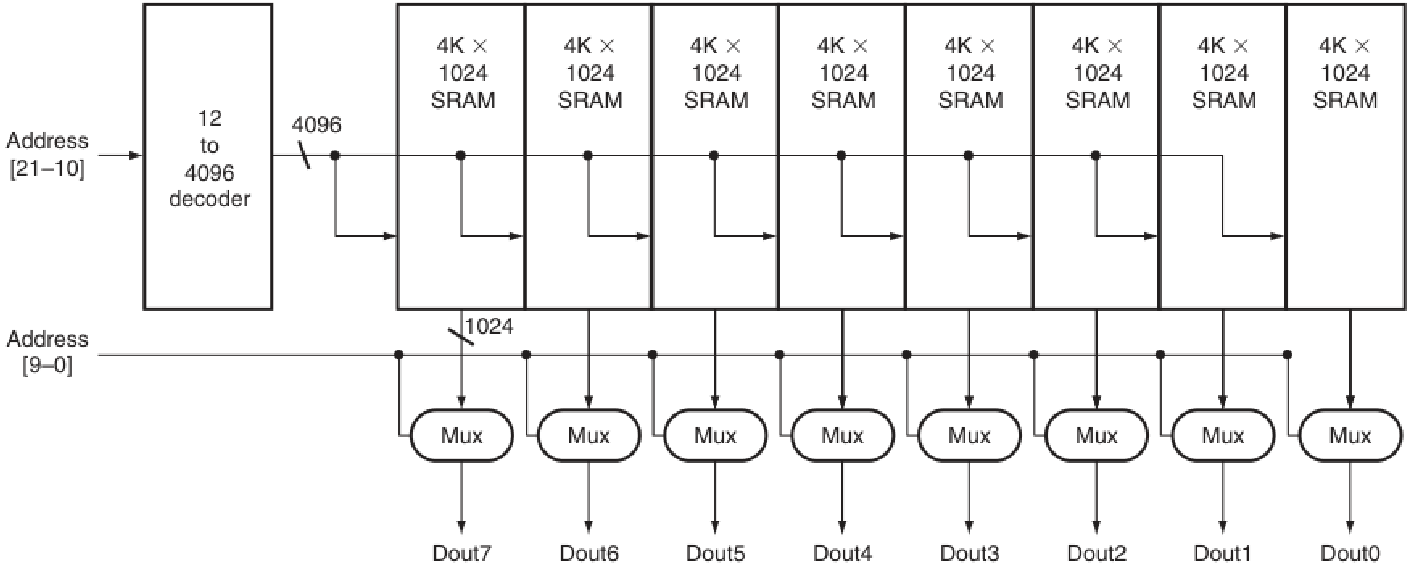
SRAM Array



- Each cell holds 1-bit.
- Each row is an entry (block).
- Address decoded to select the row.
- Data read out from bitlines.
  - 2 bitlines: Q and $\bar{Q}$.
  - Sense amplifier detects difference between Q and $\bar{Q}$.

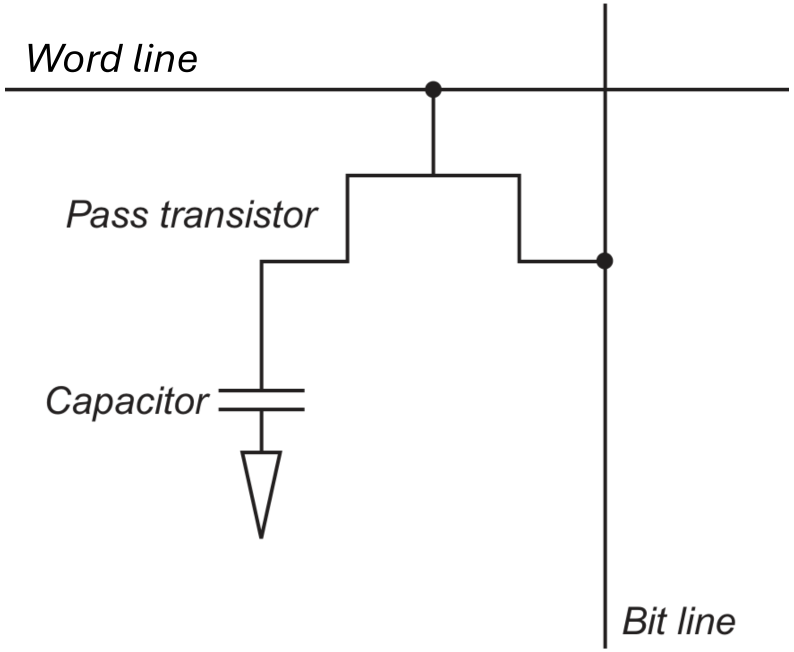
Two Level Decode

- Break a 4M×8 SRAM into 8 4K×1024 SRAMs.
  - 1st decode level: select one row (1024-bit) from each SRAM array.
  - 2nd decode level: select 1 bit from each row, total 8 bits.



Dynamic Random Access Memory (DRAM)

- SRAM: holds the data permanently unless losing power.
- DRAM: data will be lost even with power.
  - A DRAM cell is just a capacitor.
  - 1 transistor vs 6 transistors for SRAM.
  - Much cheaper and denser → Large capacity.
  - Read destructs the data.
  - Leakage also gradually destructs the data.
- Must periodically refresh the array.
  - Read it out and write it back.
- Similarly, DRAM is organized in array of cells.
  - With usually more than 2-level decoding.



Synchronous DRAM (SDRAM)

- The common type used today as it uses a clock to synchronize the operation.
- The refresh operation becomes transparent to the users.
- All control signals needed are generated inside the chip.
- The initial commercial SDRAM in the 1990s were designed for clock speed of up to 133MHz.
- Today's SDRAM chips operate with clock speeds exceeding 1 GHz.

Double Data Rate (DDR) SDRAM

- Normal SDRAMs only operate once per clock cycle.
- Double Data Rate (DDR) SDRAM transfers data on both clock edges.
- They offer increased storage capacity, lower power and faster clock speeds.
- Today, DDR4 runs at 1600MHz → 3200MT/s.
- DDR5 further pushes to 3200MHz → 6400MT/s.
- JEDEC (The Joint Electron Device Engineering Council Solid State Technology Association) defines the standard, implemented by manufacturers (Micron, Samsung, etc.)

Other Flavors of DRAM

Technology	Typical Form Factor	Key Characteristics	Typical Application
DDR (DDR4 / DDR5)	DIMMs, SO-DIMMs	General-purpose system memory; moderate latency; wide ecosystem and low cost per GB	Desktops, laptops, servers, general compute
LPDDR (LPDDR4 / LPDDR5 / LPDDR5X)	PoP, LP-SO-DIMM, soldered packages	Low power, optimized for mobile; reduced voltage and power states	Smartphones, tablets, ultra-thin laptops, embedded devices
GDDR (GDDR5 / GDDR6 / GDDR6X)	Discrete memory chips on PCB (graphics cards)	High bandwidth, optimized for streaming bandwidth; higher power than LPDDR	Discrete GPUs, gaming consoles, some AI accelerators
HBM (HBM2 / HBM2E / HBM3 / HBM3E)	3D-stacked dies on interposer or package	Extremely high bandwidth, low power per bit, expensive and complex packaging	High-end GPUs, AI accelerators, HPC, networking ASICs

SRAM vs. DRAM

- Static RAM (SRAM)
  - Capable of retaining the state as long as power is applied.
  - They are fast, low power (current flows only when accessing the cells) but costly (require several transistors), so the capacity is small.
  - They are the Level 1 cache and Level 2 cache inside a processor, of size 3 MB or more.
- Dynamic RAM (DRAM)
  - Store data as electric charge on a capacitor.
  - Charge leaks away with time, so DRAMs must be refreshed.
  - In return for this trouble, much higher density (simpler cells).

Mix-and-Match: Best of Both

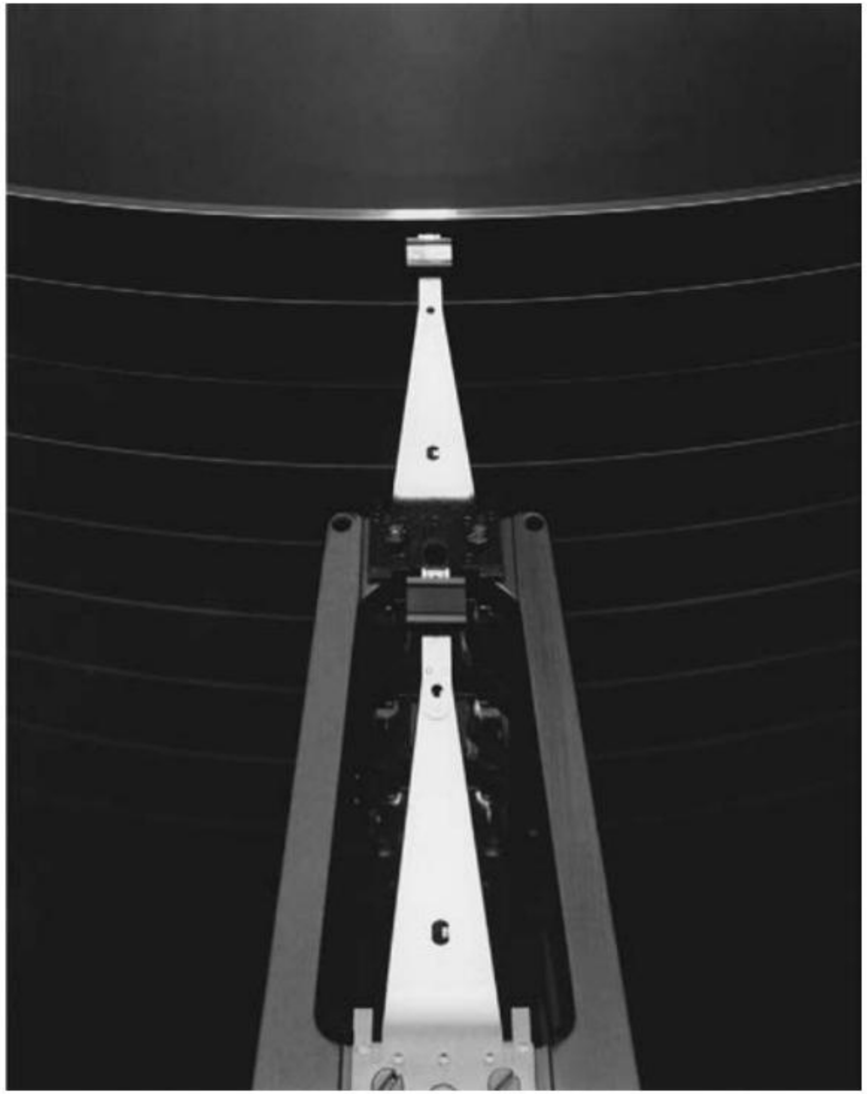
- By taking advantages of the principle of locality:
  - Present the user with as much memory as is available in the cheapest technology.
  - Provide access at the speed offered by the fastest technology.

- DRAM is slow but cheap and dense:
  - Good choice for presenting the user with a BIG memory system – main memory.

- SRAM is fast but expensive and not very dense:
  - Good choice for providing the user FAST access time – L1 and L2 cache

Disk Memory

- Hierarchy:
  - Many rotating platters with movable head.
  - Each surface divided into concentric tracks.
  - Each track divided into sectors.
  - Each sector contains typical 512-4096B.
- To access:
  - First seek the track: 3-13ms.
  - Then find track: $\frac{0.5Rotation}{RPM}$.
  - Final transfer time: bandwidth usually 250MB/s.



Flash Memory

- First credible challenger to disks.
- Nonvolatile, and 100 × −1000× faster than disks.
- Wear leveling to overcome wear out problem.

Summary

Summary

- Processor usually runs much faster than main memory.
- Common RAM types: SRAM, DRAM, SDRAM, DDR SDRAM.
- Principle of locality: Temporal and Spatial.
  - Present the user with as much memory as is available in the cheapest technology.
  - Provide access at the speed offered by the fastest technology.
- Memory hierarchy:
  - Register → Cache → Main Memory → Disk → Tape