CS / EE 320
Computer Organization and Assembly Language

Spring 2025

Lecture 21

Shahid Masud

Topics: Memory Hierarchy, Memory Cell Technology, Memory Array Architecture

Topics

- Levels of memory hierarchy w.r.t. different technology, capacity, performance
- Examples of memory organization in memory array chip architecture using row and column decoders
- Concept of Lines in Cache and Blocks in main memory to capitalize on temporal and spatial locality
- Memory Hierarchy and CPU Connection

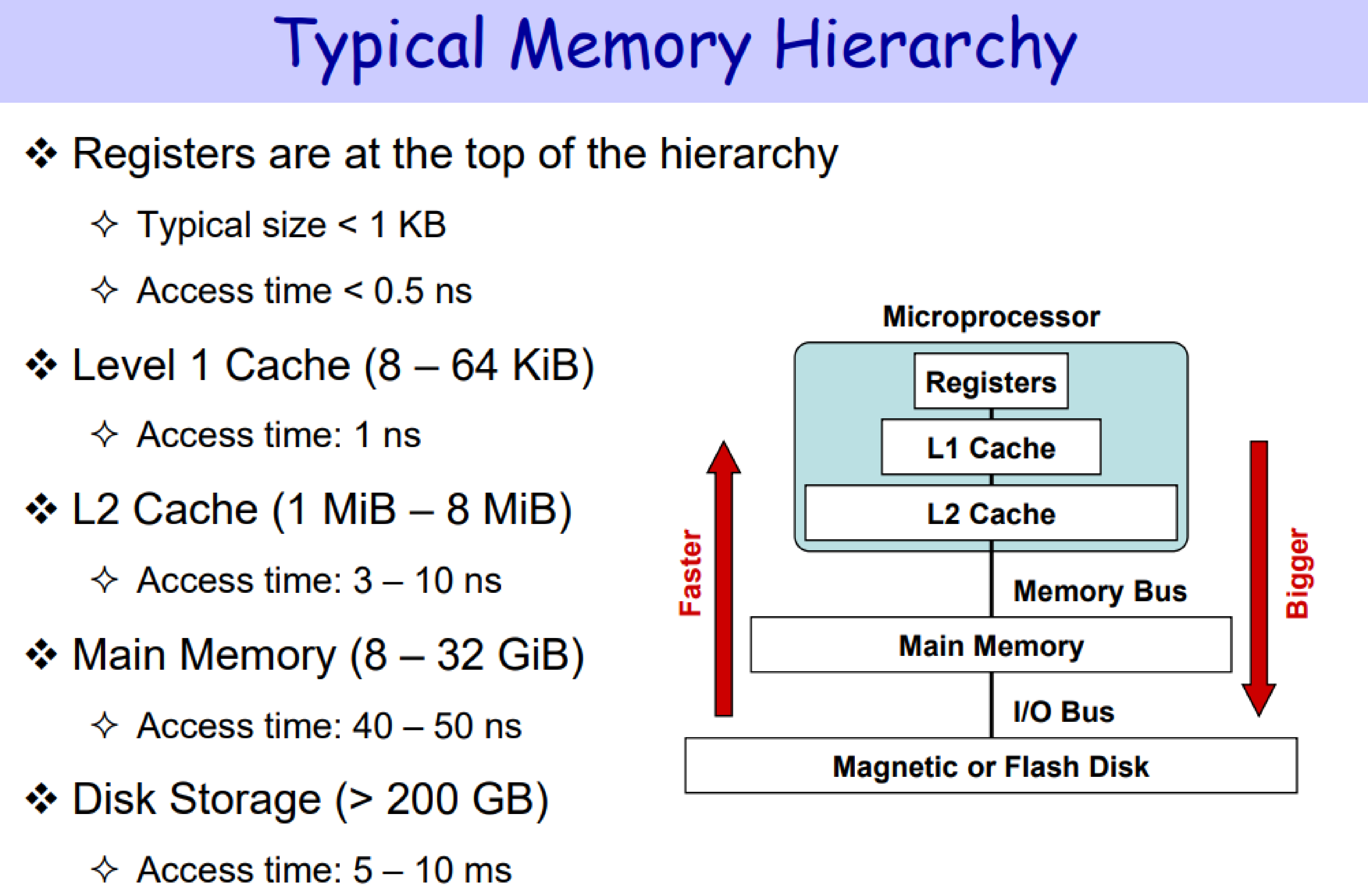
- **QUIZ 4 NEXT LECTURE**

Memory Hierarchy

Computer Memory Combination

- Registers
  - In CPU
- Internal or Main memory
  - May include one or more levels of Cache
  - “RAM”
- External memory
  - Backup store
  - Network Storage

Typical Memory Hierarchy

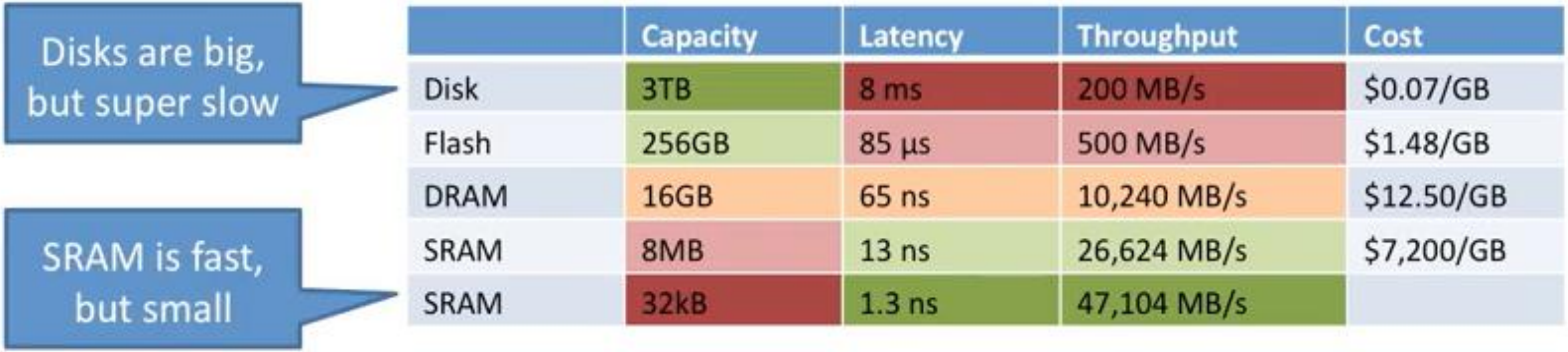


Typical Memory Hierarchy

- Registers are at the top of the hierarchy
  - Typical size < 1 KB
  - Access time < 0.5 ns
- Level 1 Cache (8 – 64 KiB)
  - Access time: 1 ns
- L2 Cache (1 MiB – 8 MiB)
  - Access time: 3 – 10 ns
- Main Memory (8 – 32 GiB)
  - Access time: 40 – 50 ns
- Disk Storage (> 200 GB)
  - Access time: 5 – 10 ms

Ideal Memory Requirement

- We want our memory to be big **and** fast
  - ISA promises big: 2^{32} memory address (4GB)
  - Want it to be fast because 33% of instructions are loads/stores and 100% of instructions load instructions
- But what do we have to work with?
  - Nothing that is **both big** and **fast**!



Disks are big, but super slow

SRAM is fast, but small

	Capacity	Latency	Throughput	Cost
Disk	3TB	8 ms	200 MB/s	$0.07/GB
Flash	256GB	85 µs	500 MB/s	$1.48/GB
DRAM	16GB	65 ns	10,240 MB/s	$12.50/GB
SRAM	8MB	13 ns	26,624 MB/s	$7,200/GB
SRAM	32kB	1.3 ns	47,104 MB/s	

Why do we need different types of Memory?

- What do we have?

– Hard disk:	**Huge** (1000 GB)	**Super slow** (1M cycles)
– Flash:	**Big** (100 GB)	**Very slow** (1k cycles)
– DRAM:	**Medium** (10 GB)	**Slow** (100 cycles)
– SRAM:	**Small** (10 MB)	**Fast** (1-10 cycles)

- Need **fast and big**
  - Can’t use **just SRAM** (too **small**)
  - Can’t use **just DRAM** (too **slow** and **small**)
  - Can’t use **just Flash/Hard disk** (way too **slow**)
- But we can **combine** them to get:
  - **Speed** from (small) **SRAMs**
  - **Size** from (big) **DRAM** and **Hard disk**

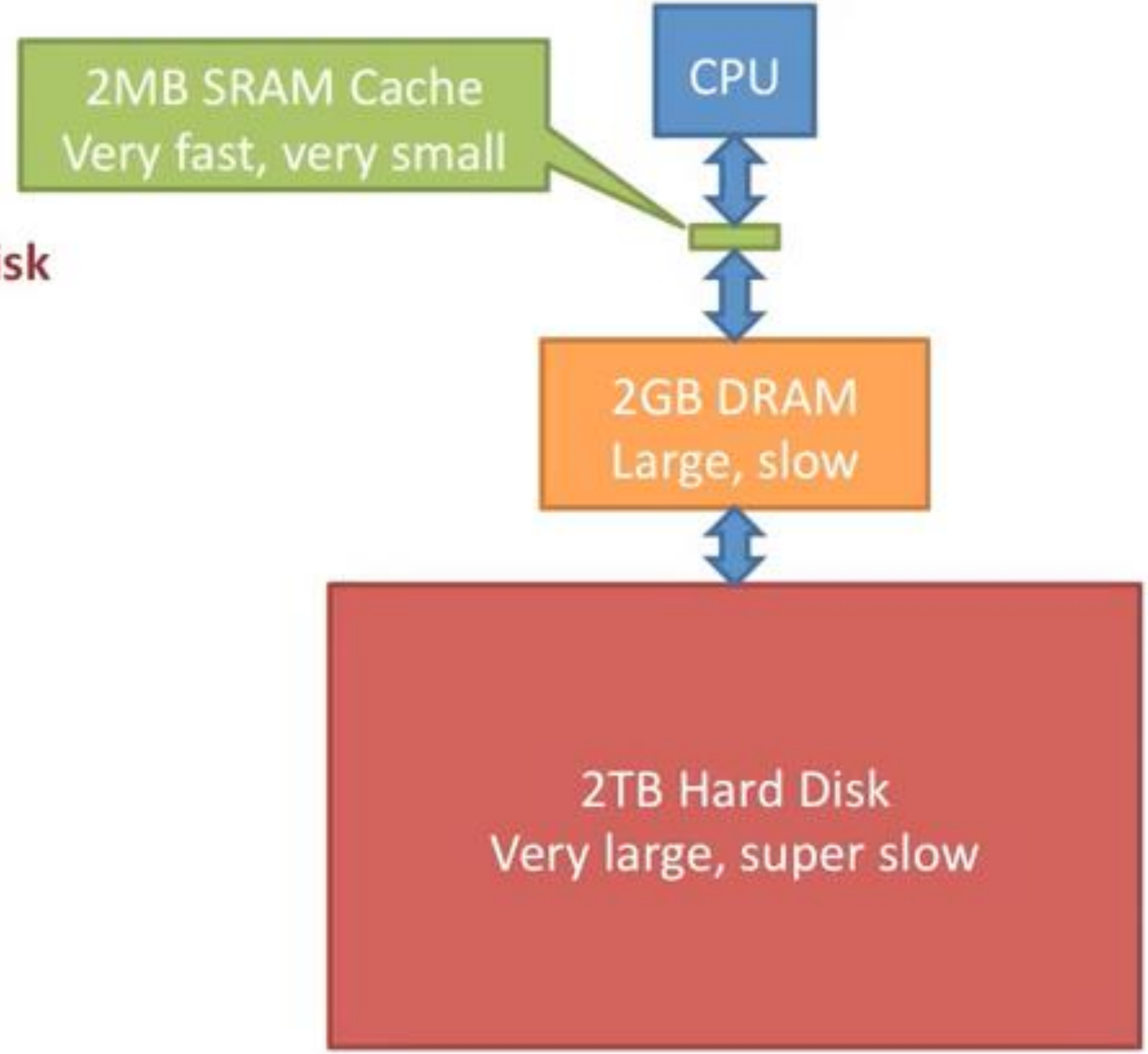
We’ll build a hierarchy using different technologies to get the best of all of them.

The Memory Hierarchy

- **Use:**
  - Small amounts of fast SRAM
  - Lots of slow DRAM
  - Huge amounts of super slow hard disk

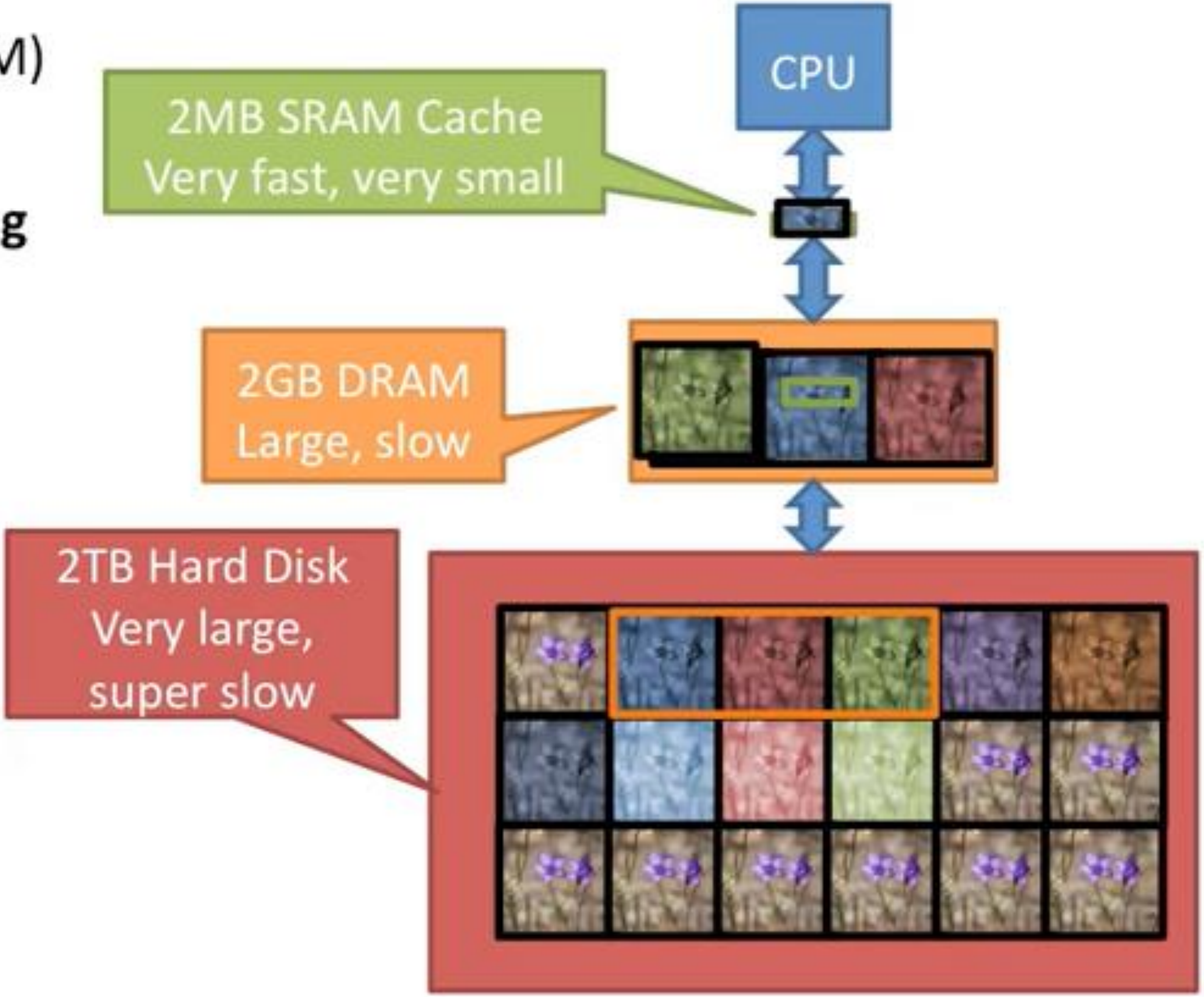
- **To create the illusion of:**
  - Very **large**
  - Very **fast** (on average)

- **How do we do this?**
  - Try to keep the important data in the fast memory
  - Move the unimportant data to the slow memory

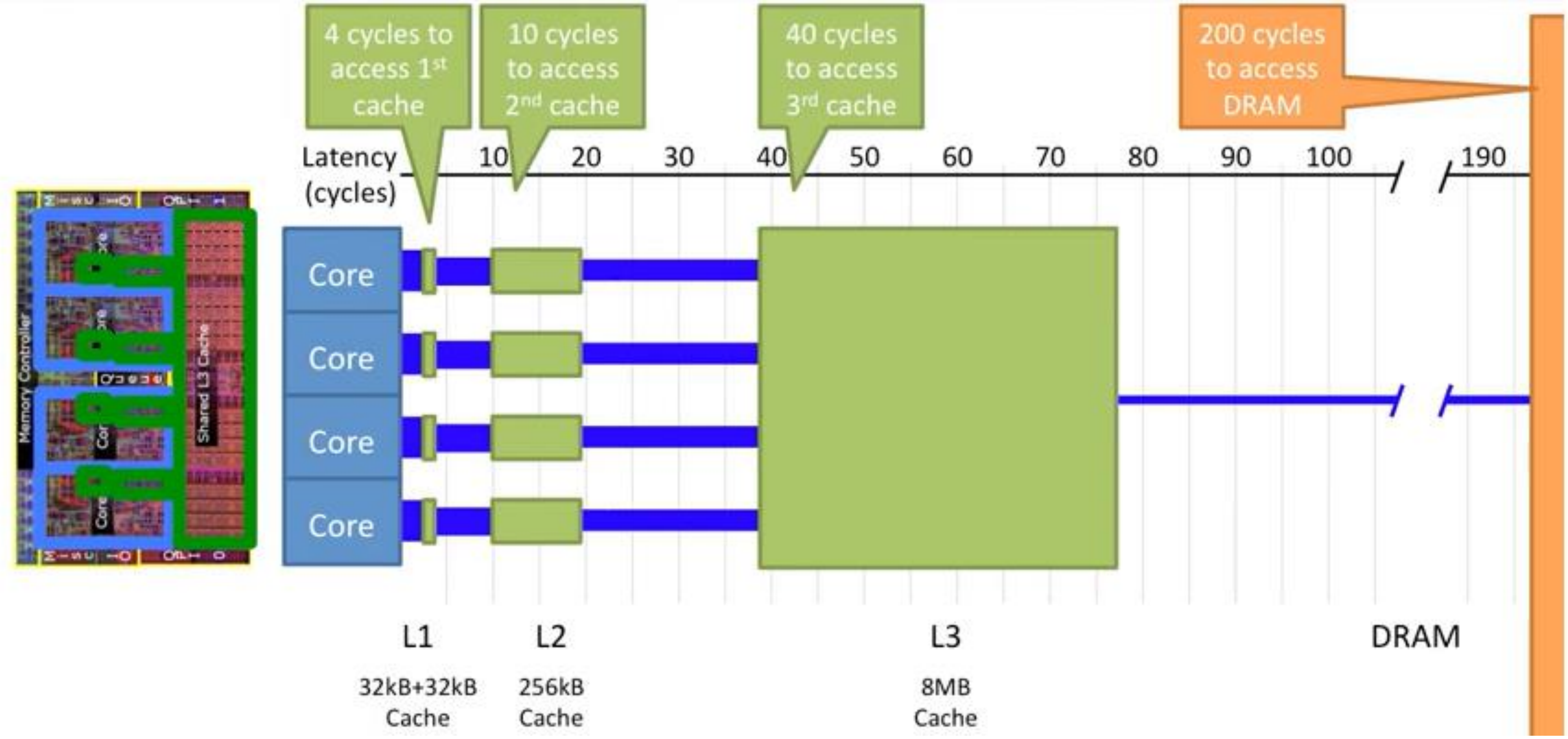


Example of Memory Hierarchy – Video Processing

- Video is large (bigger than DRAM)
- **Store on hard disk**
- **Load** just **the part we are editing** into **DRAM**
- The **CPU loads the data it is processing** into the **cache**
- Move new data into DRAM and cache as we process the video
- **Remember:**
  - Try to keep the **important data** in the **fast memory**
  - Move the **unimportant data** to the **slow memory**



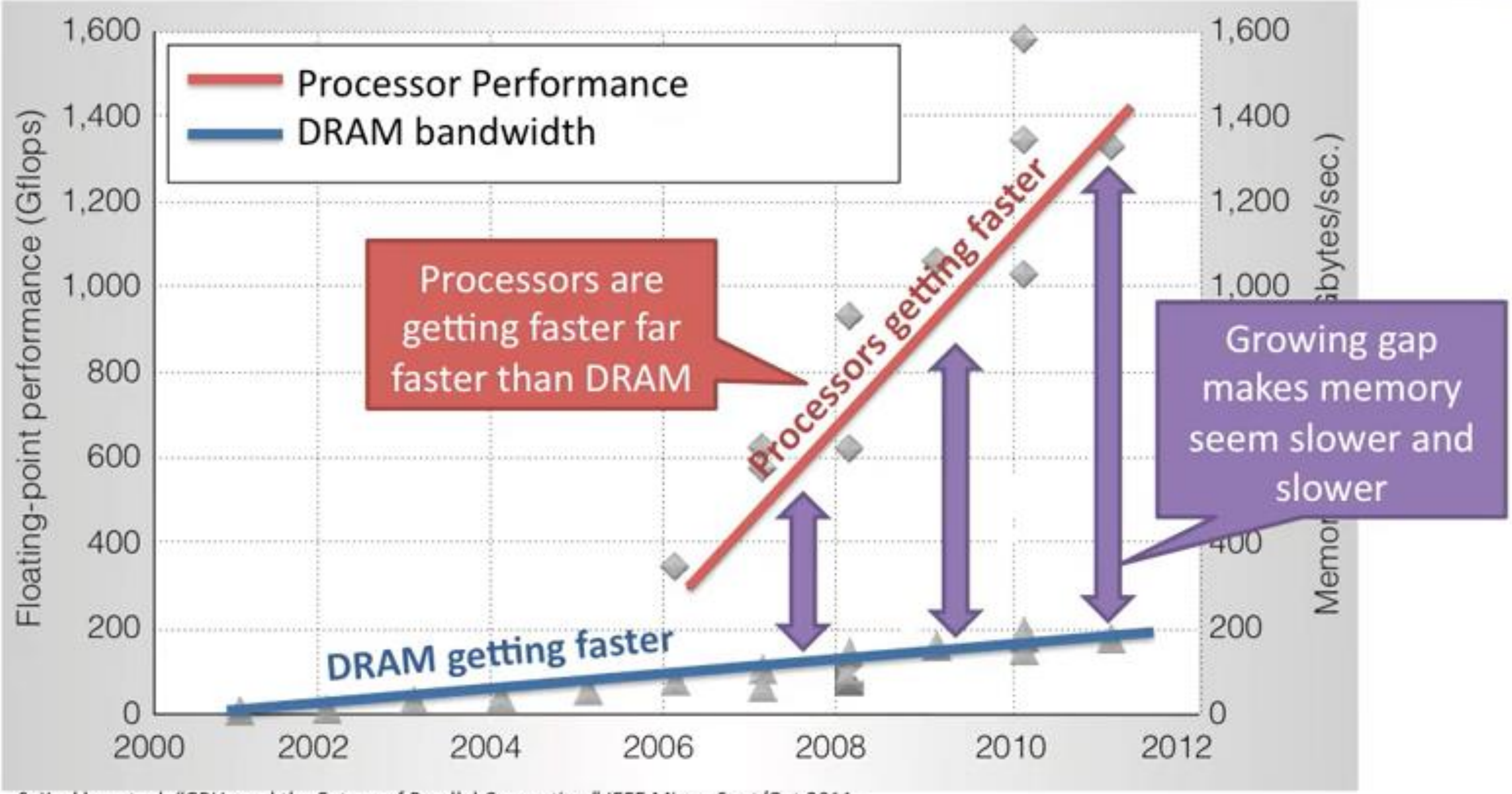
Intel Memory Hierarchy



Fast but Expensive?

- It is possible to build a computer which uses only static RAM (see later)
- This would be very **fast**
- This would need no cache
  - How will you replace a Cache?
- This would **cost** a very large amount

Problem with Memory and Moore’s Law



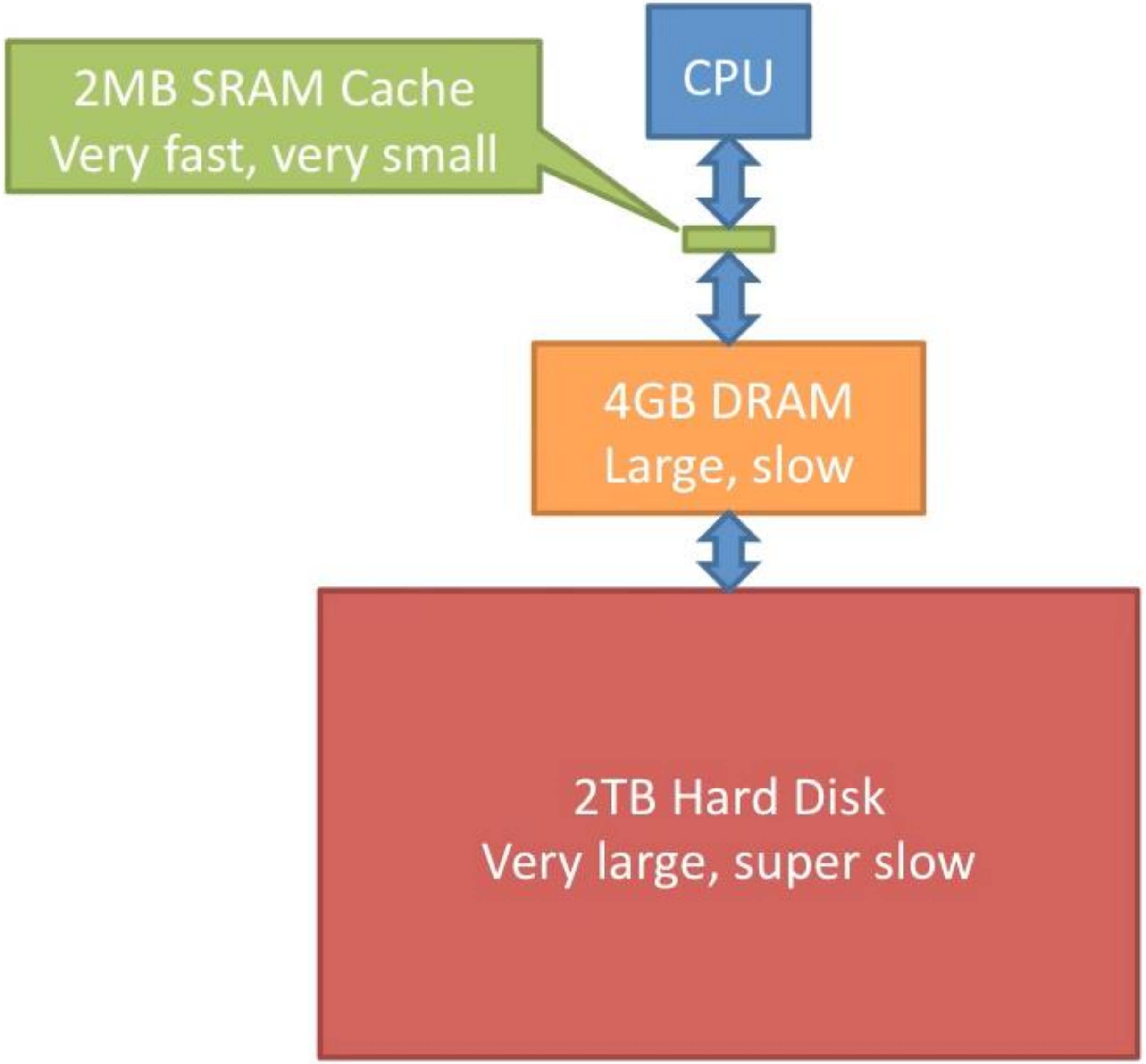
Processors are getting faster far faster than DRAM

Growing gap makes memory seem slower and slower

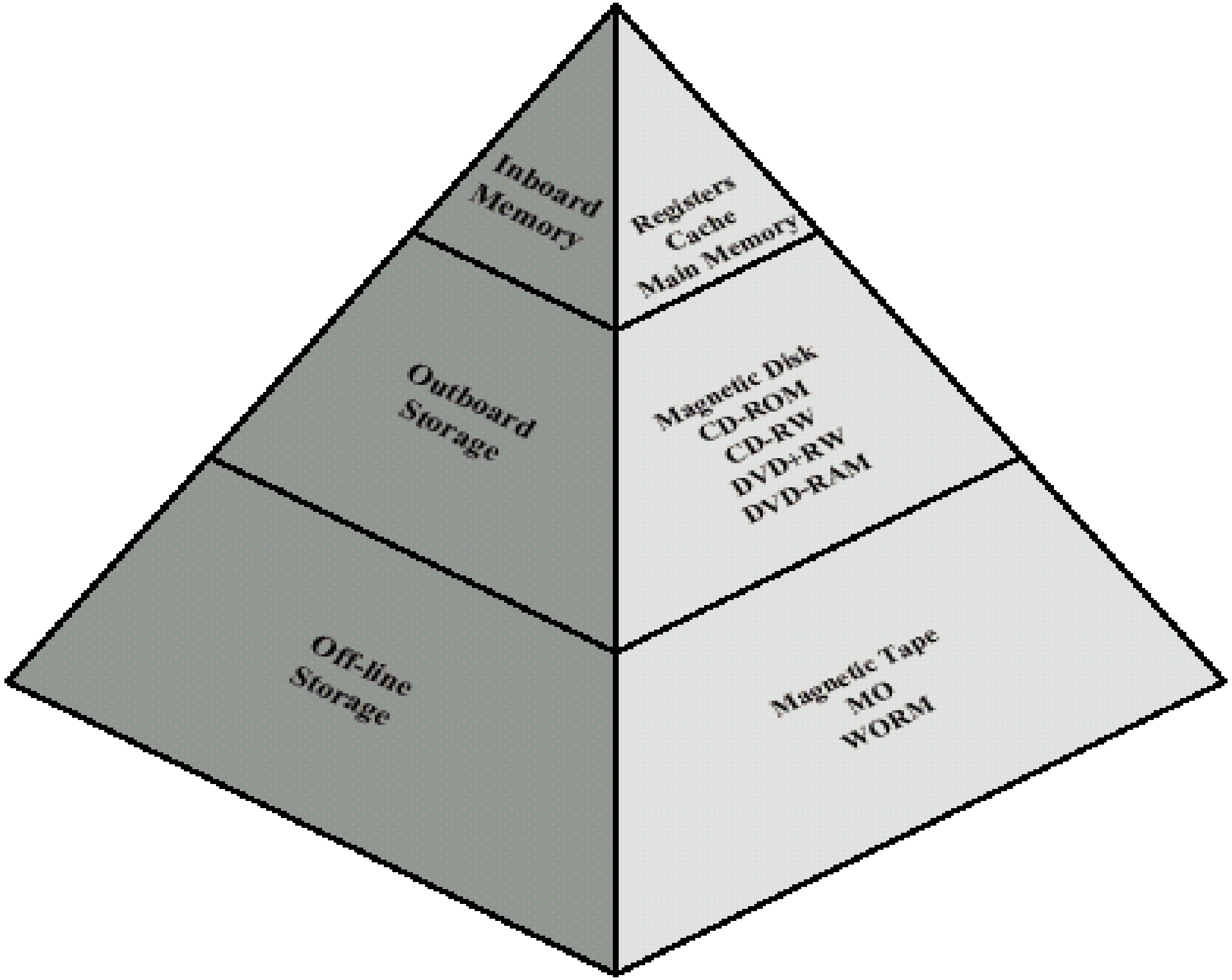
S. Keckler, et. al. “GPUs and the Future of Parallel Computing.” IEEE Micro, Sept/Oct 2011.

Advantages of Memory Hierarchy

- **Very fast**
  - If we have the right data in the right place
- **Very large**
  - But possibly very slow
- **Reasonably cheap**
  - Lots of the **cheap stuff**
  - A little of the **expensive stuff**

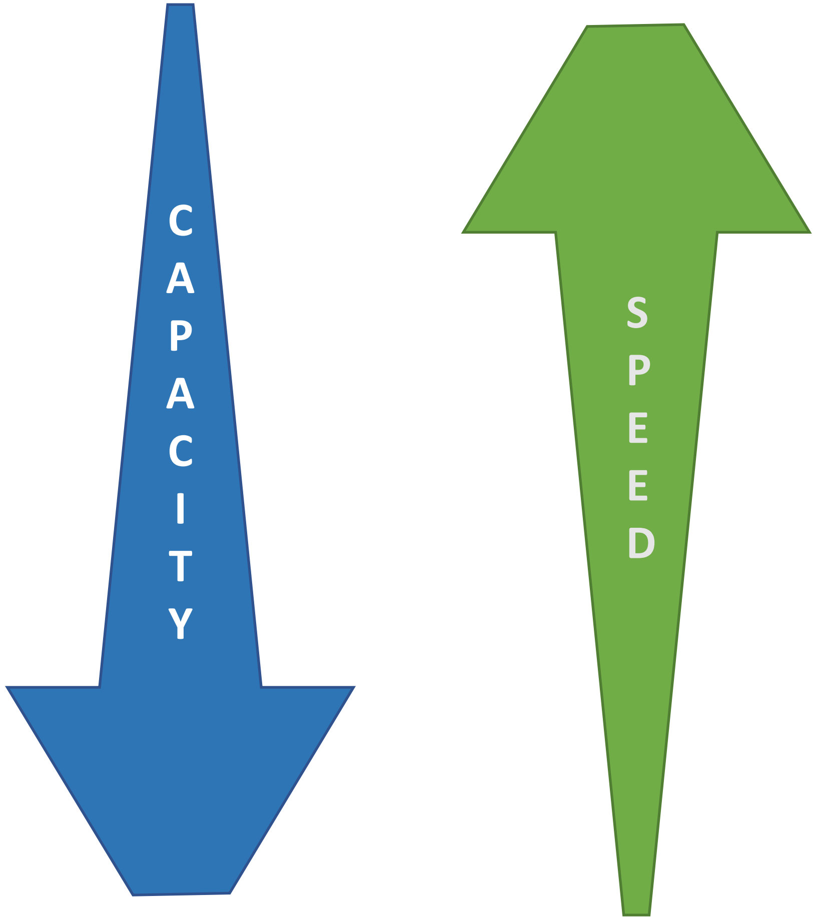


Memory Hierarchy in a Computer System



Hierarchy List

- Registers
- L1 Cache
- L2 Cache
- Main memory
- Disk cache
- Disk
- Optical
- Tape



Memory Types and Performance

Characteristics of Memory Types

- Location – CPU, Internal, External
- Capacity
- Unit of transfer
- Access method
- Performance
- Physical type
- Physical characteristics
- Organisation

Unit of Memory Transfer

- Internal
  - Usually governed by data bus width
- External
  - Usually as a block which is much larger than a word
- Addressable unit
  - Smallest location which can be uniquely addressed
  - Word internally
  - Cluster on disks

Performance

- Access time
  - Time between presenting the address and getting the valid data on port
- Memory Cycle time
  - Time may be required for the memory to “recover” before next access
  - Cycle time includes **access + recovery**
- Transfer Rate
  - Rate at which data can be moved in and out of memory

Memory Performance - Mathematically

- Access Time
  - Time between address appearing on address lines and data coming out from memory cells to data lines for RAM and vice versa.
- Memory Cycle Time
  - (Access Time + Extra time) before a second read / write can take place.
- Transfer Rate
  - For RAM $T_R = \frac{1}{Cycle\ Time}$
  - For non-RAM $T_N = T_A + \frac{N}{R}$
    - Where T_N = Avg time to read or write N bits
    - T_A = Avg Access Time
    - N = number of bits
    - R = Transfer Rate in bits / second

Physical Types of Memory Technology

- Semiconductor
  - RAM
- Magnetic
  - Disk & Tape
- Optical
  - CD & DVD
- Physical Properties
  - Volatility
  - Erasable
  - Power and Access

Semiconductor Memory, RAM and ROM

- RAM
  - Misnamed as all semiconductor memory is random location access
  - Read/Write
  - Volatile
  - Temporary storage
  - Static or dynamic
- ROM
  - Permanent storage
  - Microprogramming (see later)
  - Library subroutines
  - Systems programs (BIOS)
  - Function tables

Static RAM

- Bits stored as on/off switches
- No charges to leak
- No refreshing needed when powered
- More complex construction
- Larger per bit
- More expensive
- Does not need refresh circuits
- Faster
- Cache

Dynamic RAM

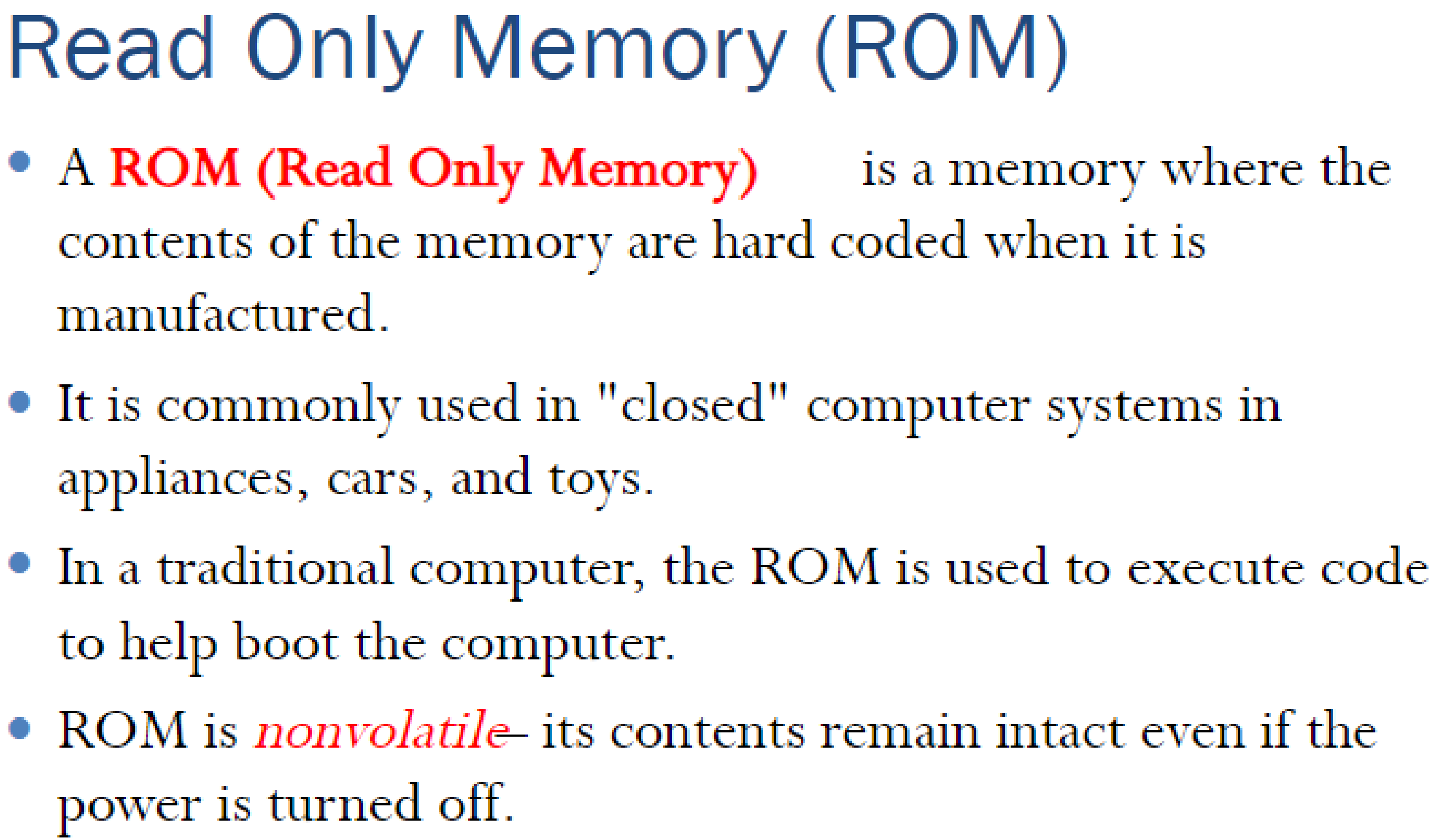
- Bits stored as charge in capacitors
- Charges leak
- Need refreshing even when powered on
- Simpler construction
- Smaller per bit area
- Less expensive
- Need refresh circuits that are always working
- Slower
- Used as Main memory

RAM Memory

Random Access Memory (RAM)

- Main memory is stored in **RAM (Random Access Memory)** .
- *Static RAM*:Implemented using a circuit similar to the D flip-flop circuit.
  - Uses 6 transistors
  - Very fast
- *Dynamic RAM*Implemented using a transistor and a capacitor.
  - Capacitors must be refreshed periodically
  - Very high density
- RAM is *volatile*– memory cells retain their values as long as the power is on.
  - However, if the power goes off – the values disappear.
  - Registers and caches are also volatile.

ROM Memory



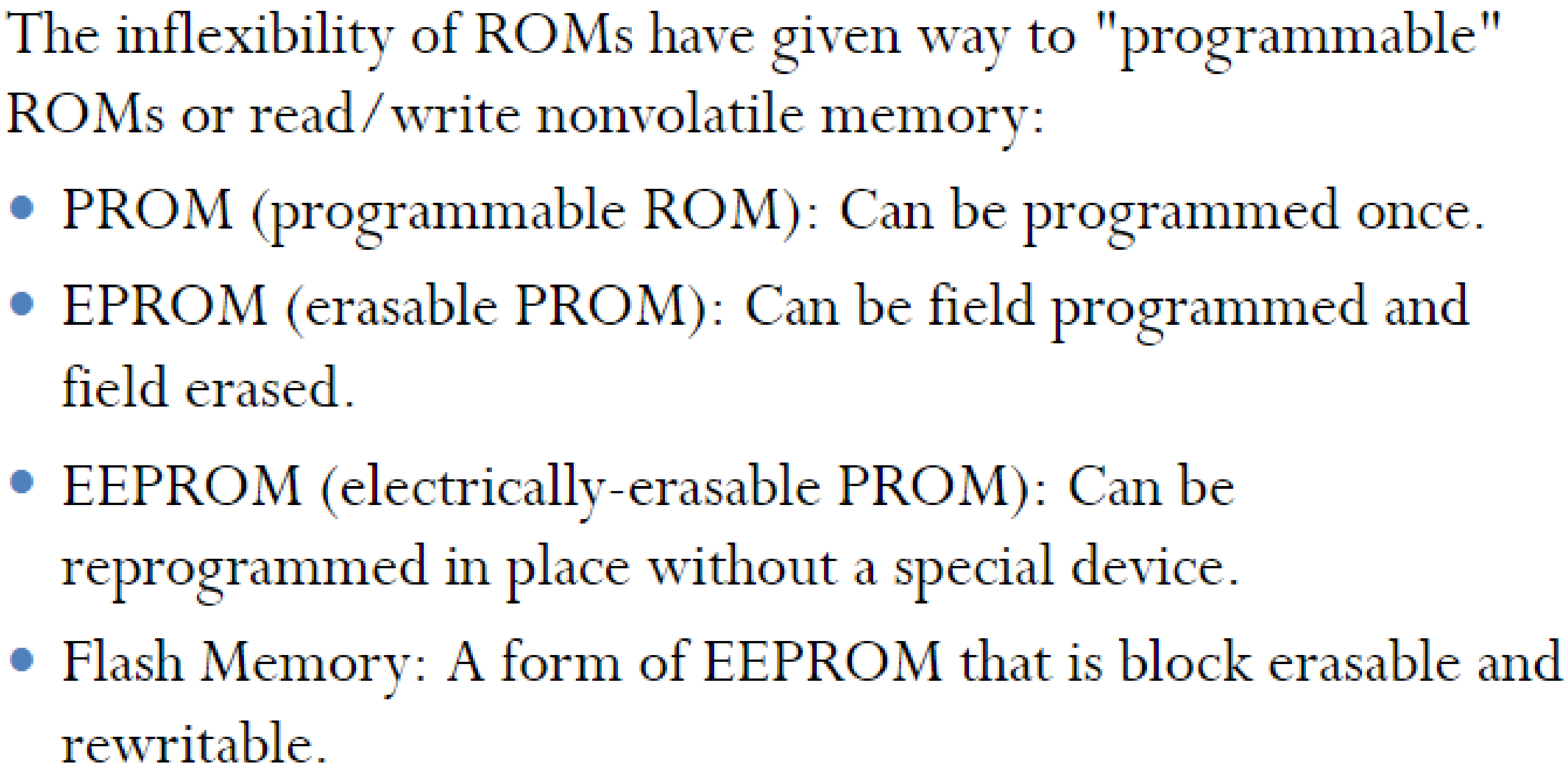
Read Only Memory (ROM)

- A **ROM (Read Only Memory)** is a memory where the contents of the memory are hard coded when it is manufactured.
- It is commonly used in "closed" computer systems in appliances, cars, and toys.
- In a traditional computer, the ROM is used to execute code to help boot the computer.
- ROM is *nonvolatile*– its contents remain intact even if the power is turned off.

Types of ROM

- Written during manufacture and special equipment
  - Very expensive for small runs
- Programmable (once)
  - PROM
  - Needs special equipment to program
- Read “mostly”
  - Erasable Programmable (EPROM)
    - Erased by UV
  - Electrically Erasable (EEPROM)
    - Takes much longer to write than read
  - Flash memory
    - Erase whole memory electrically

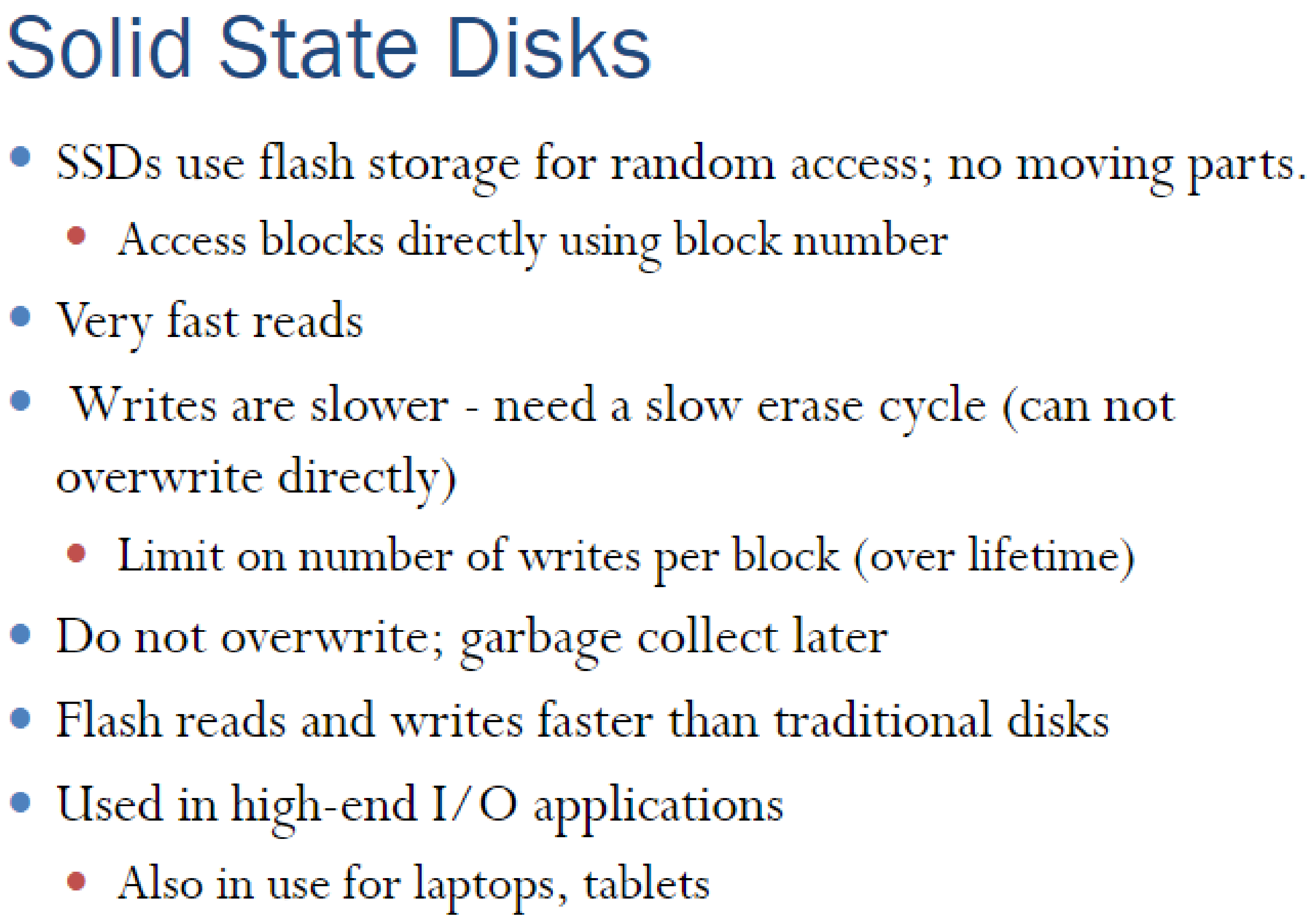
PROM / FLASH Memory



The inflexibility of ROMs have given way to "programmable" ROMs or read/write nonvolatile memory:

- PROM (programmable ROM): Can be programmed once.
- EPROM (erasable PROM): Can be field programmed and field erased.
- EEPROM (electrically-erasable PROM): Can be reprogrammed in place without a special device.
- Flash Memory: A form of EEPROM that is block erasable and rewritable.

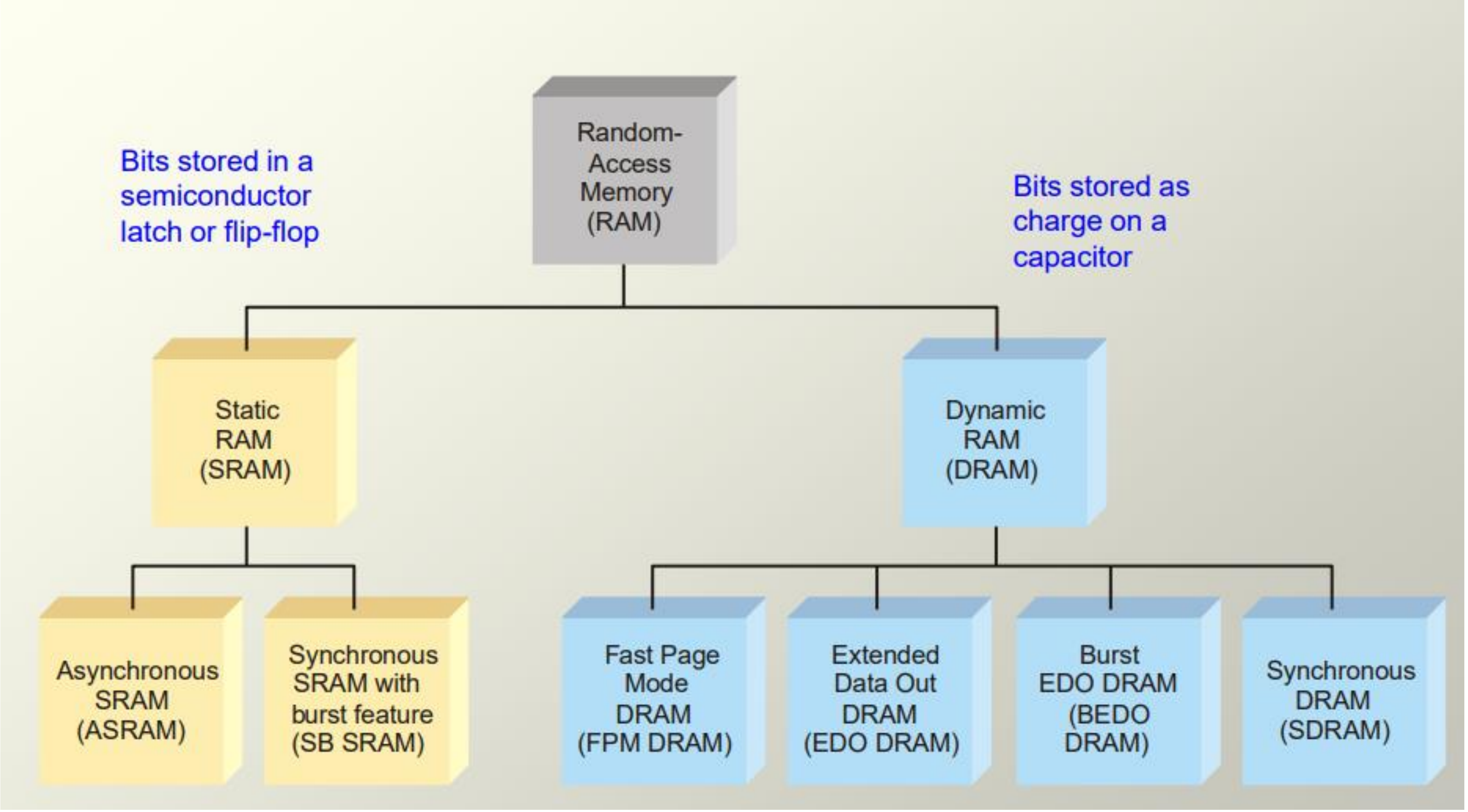
SSD Memory



Solid State Disks

- SSDs use flash storage for random access; no moving parts.
  - Access blocks directly using block number
- Very fast reads
- Writes are slower - need a slow erase cycle (can not overwrite directly)
  - Limit on number of writes per block (over lifetime)
- Do not overwrite; garbage collect later
- Flash reads and writes faster than traditional disks
- Used in high-end I/O applications
  - Also in use for laptops, tablets

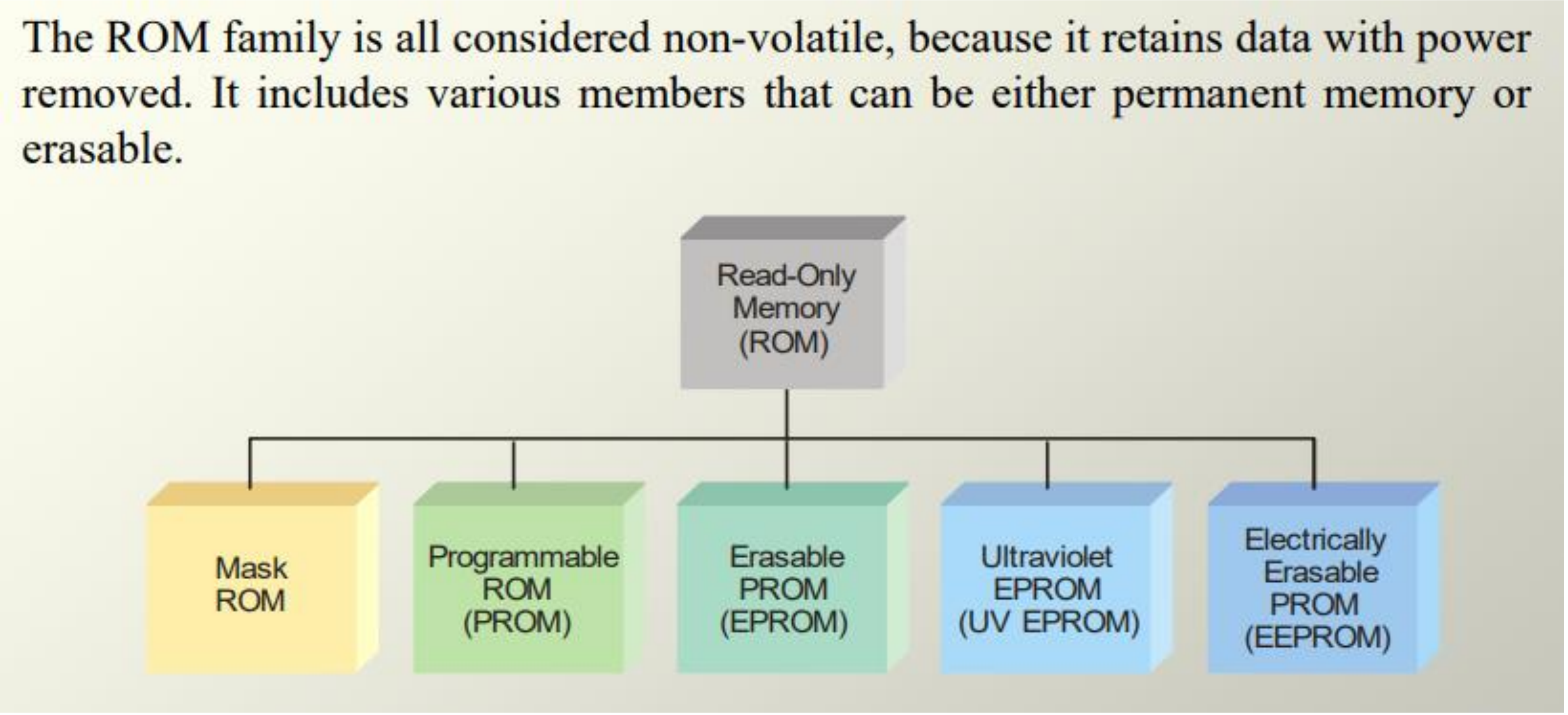
Types of RAM Memory


Random-Access Memory (RAM)
Static RAM (SRAM)
Dynamic RAM (DRAM)
Asynchronous SRAM (ASRAM)
Synchronous SRAM with burst feature (SB SRAM)
Fast Page Mode DRAM (FPM DRAM)
Extended Data Out DRAM (EDO DRAM)
Burst EDO DRAM (BEDO DRAM)
Synchronous DRAM (SDRAM)

Bits stored in a semiconductor latch or flip-flop

Bits stored as charge on a capacitor

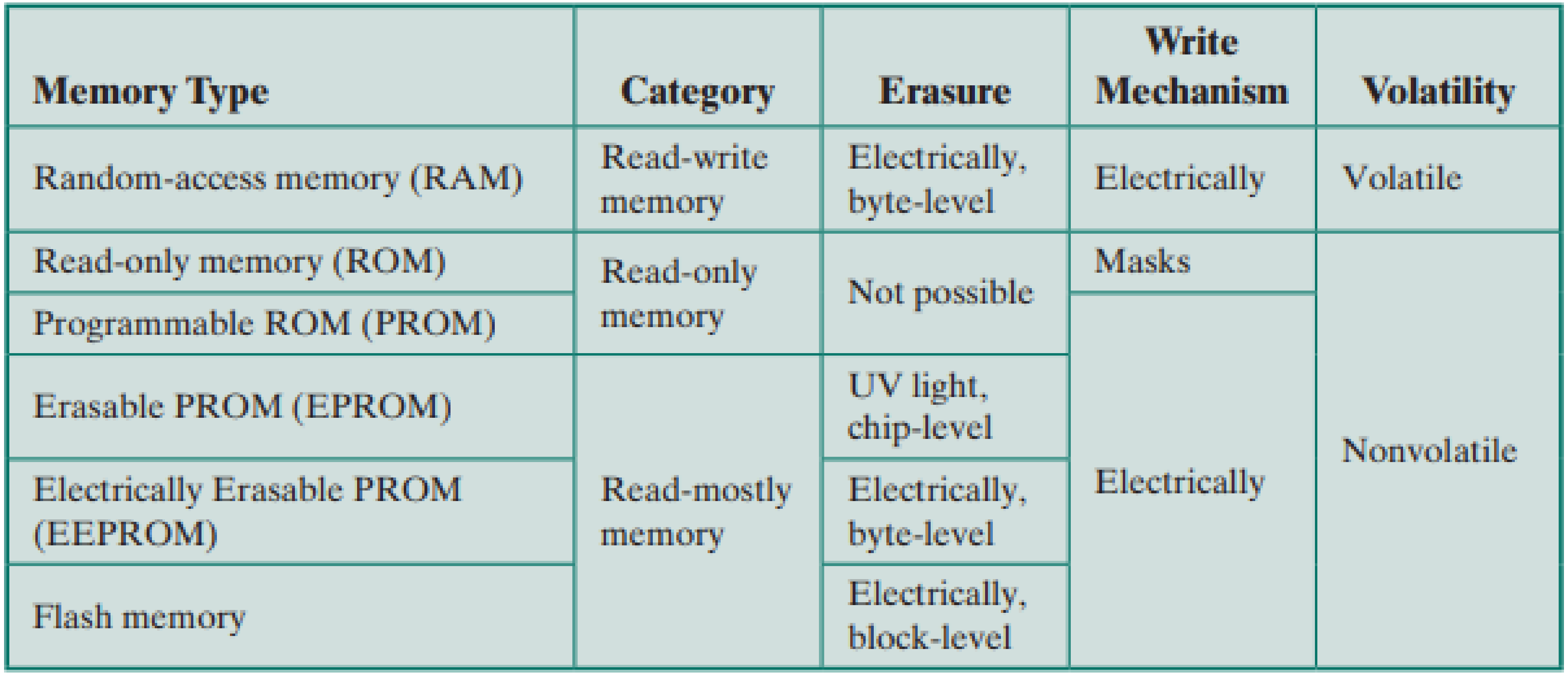
Types of ROM Memory



The ROM family is all considered non-volatile, because it retains data with power removed. It includes various members that can be either permanent memory or erasable.

Semiconductor Memory Types - Summary

Table 5.1 Semiconductor Memory Types



Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Electrically	Volatile
Read-only memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile
Programmable ROM (PROM)			Electrically	
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level		
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		

Memory Construction

Memory Cell Operation

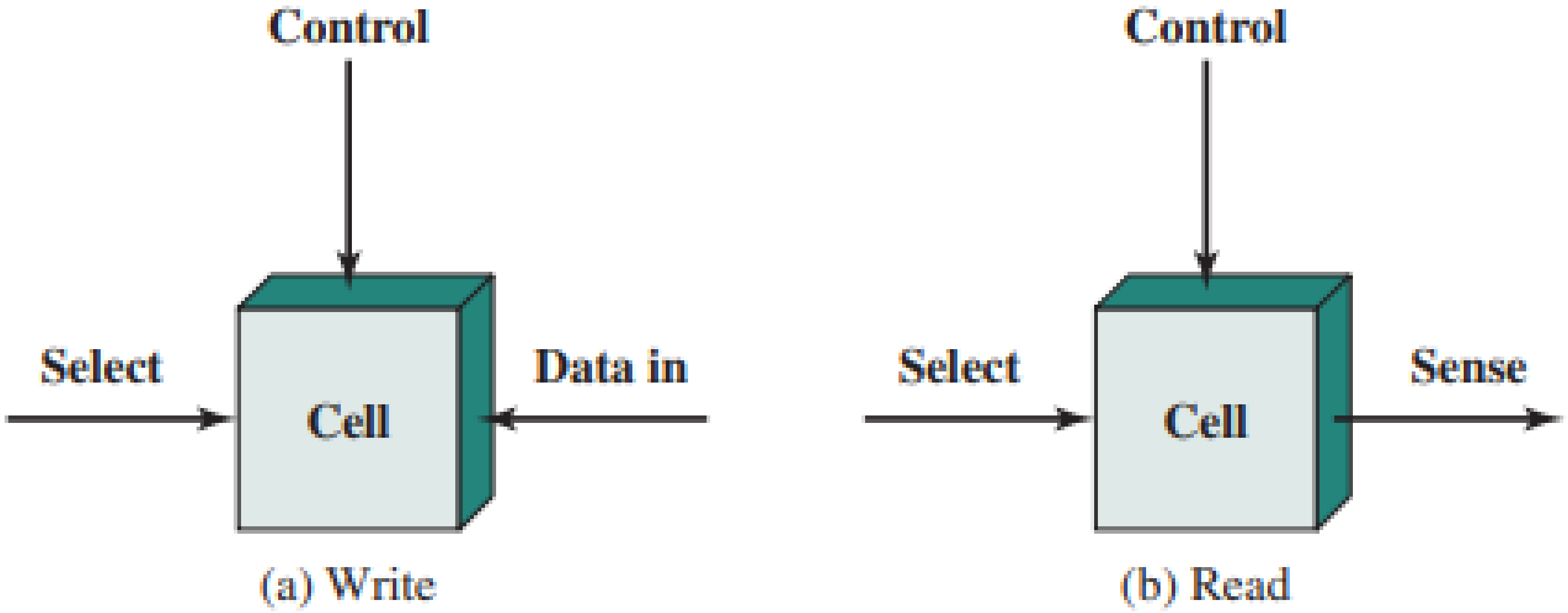
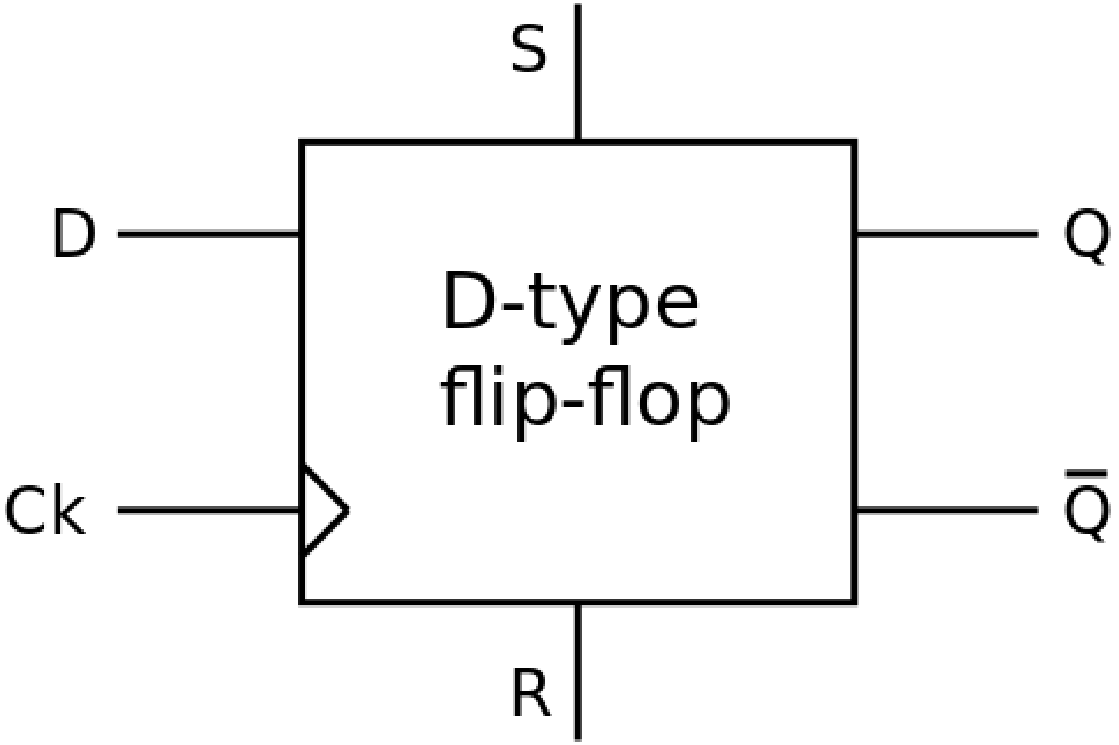


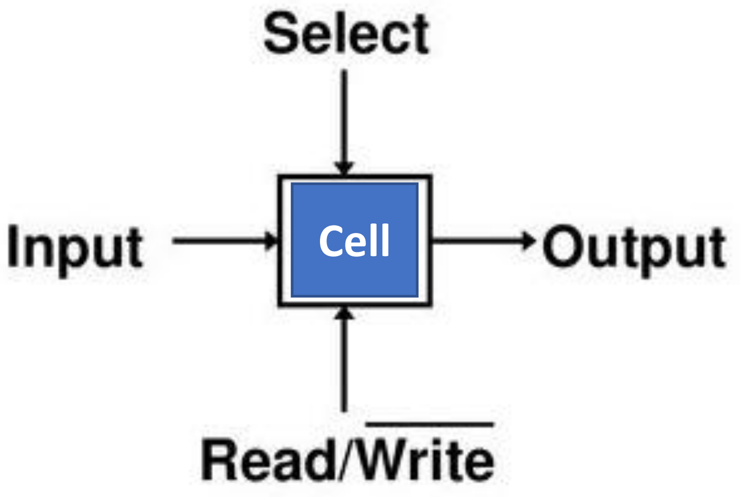
Figure 5.1 Memory Cell Operation

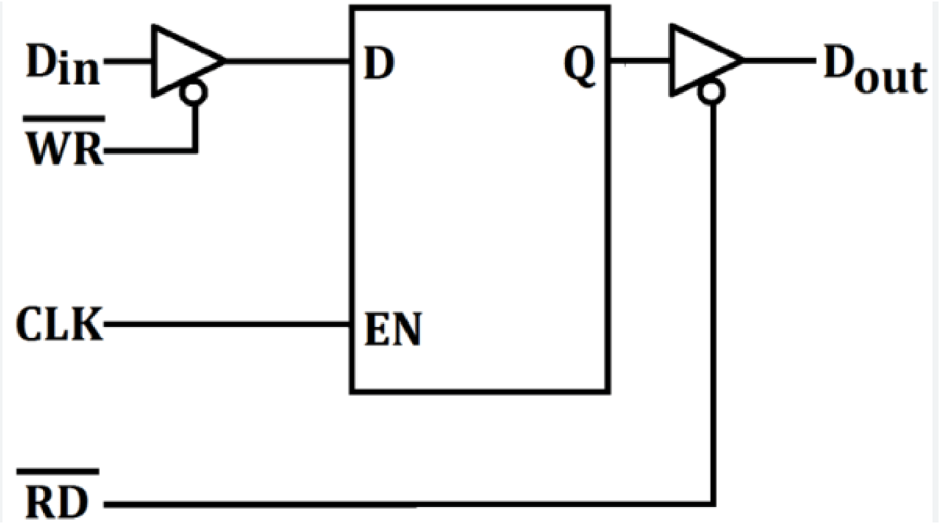
Simplest Digital Storage Element – D Flipflop



“D Latch” is a type of storage without a ‘Clock’. Instead it has an ‘Enable’ signal.

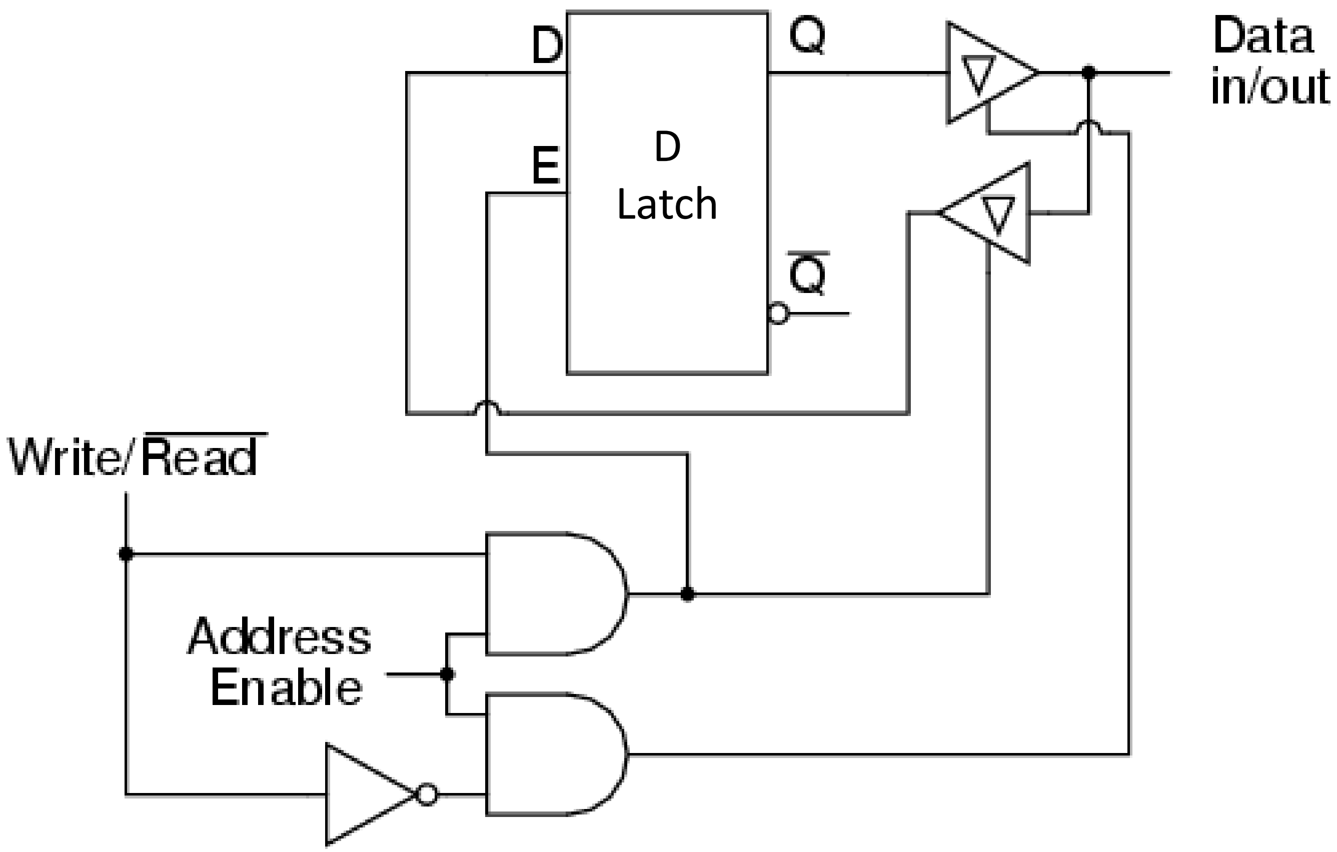
Memory Cell Design





Asynchronous Memory Cell using D Latch

Memory cell circuit

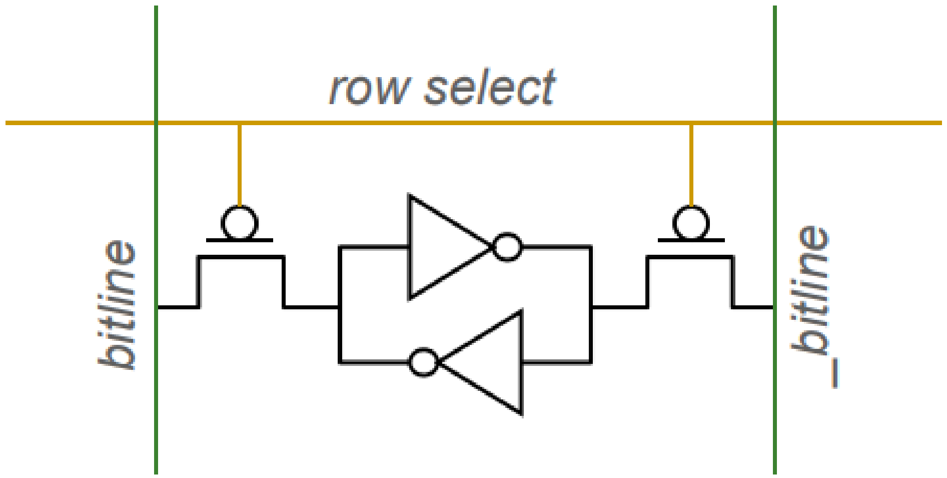


No Clock

Only Enable Signal

SRAM Cell Technology

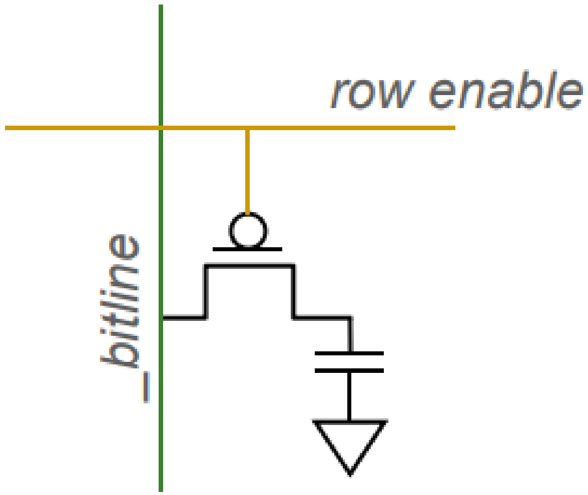
- Static random access memory
- Two cross coupled inverters store a single bit
  - Feedback path enables the stored value to persist in the "cell"
  - 4 transistors for storage
  - 2 transistors for access



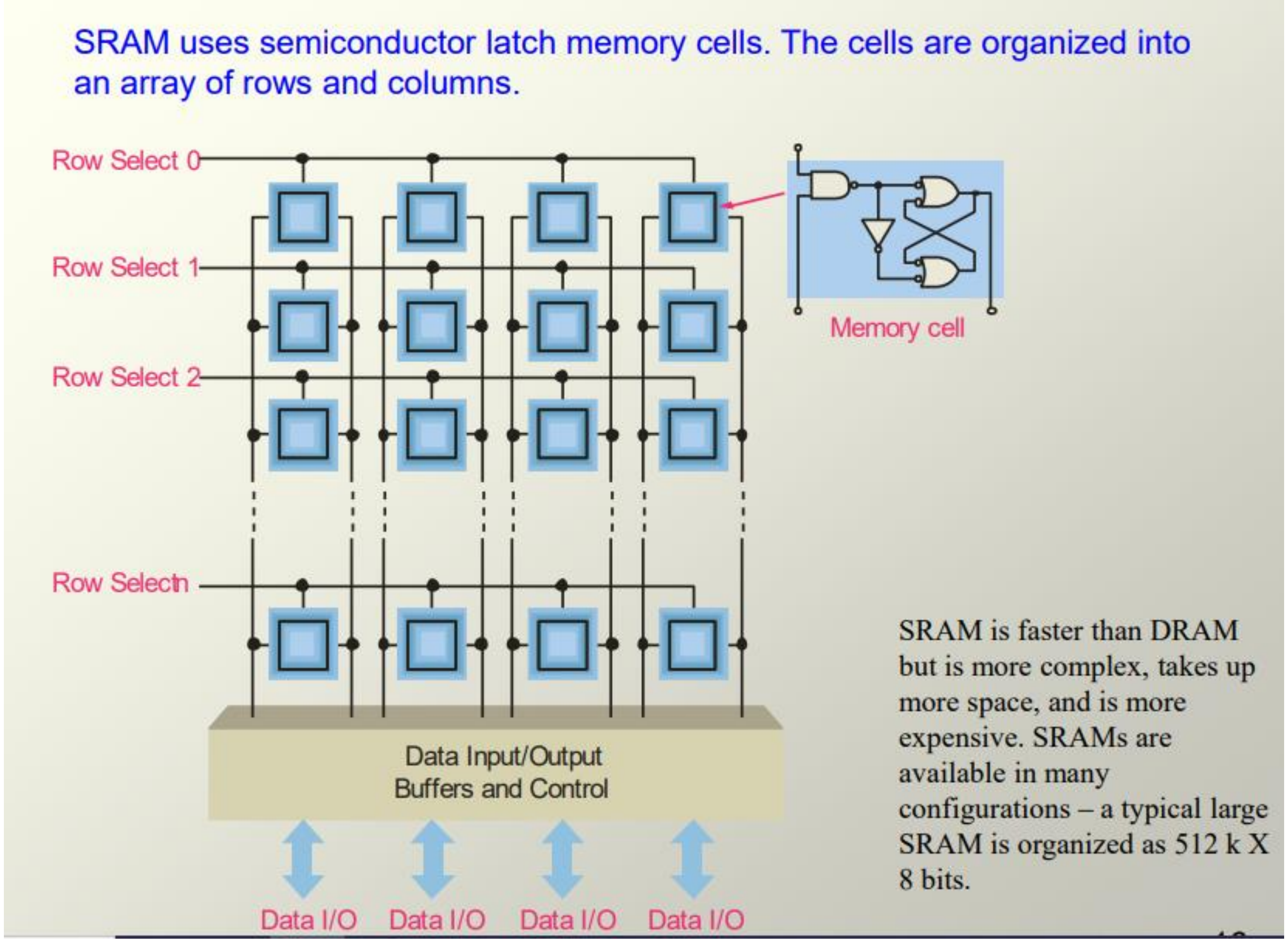
DRAM Cell Technology

- Dynamic random access memory
- Capacitor charge state indicates stored value
  - Whether the capacitor is charged or discharged indicates storage of 1 or 0
  - 1 capacitor
  - 1 access transistor

- Capacitor leaks through the RC path
  - DRAM cell loses charge over time
  - DRAM cell needs to be refreshed



Memory Cell Organization



SRAM uses semiconductor latch memory cells. The cells are organized into an array of rows and columns.

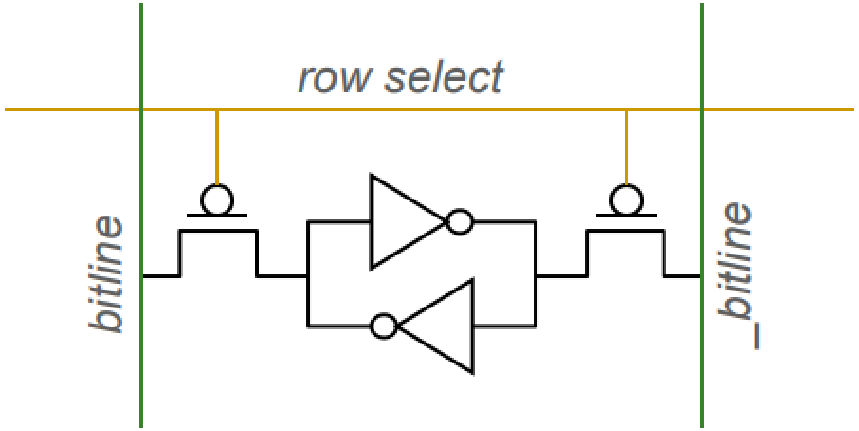
SRAM is faster than DRAM but is more complex, takes up more space, and is more expensive. SRAMs are available in many configurations – a typical large SRAM is organized as 512 k X 8 bits.

Memory Technology

- Static RAM (SRAM)
  - 0.5ns – 2.5ns, $500 – $1000 per GB
- Dynamic RAM (DRAM)
  - 50ns – 70ns, $3 – $6 per GB
- Magnetic disk
  - 5ms – 20ms, $0.01 – $0.02 per GB
- Ideal memory
  - Access time of SRAM
  - Capacity and cost/GB of disk

SRAM Cell Technology

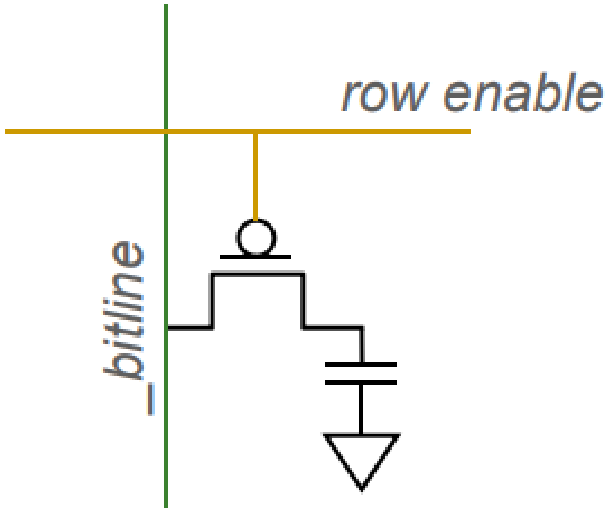
- Static random access memory
- Two cross coupled inverters store a single bit
  - Feedback path enables the stored value to persist in the "cell"
  - 4 transistors for storage
  - 2 transistors for access



DRAM Cell Technology

- Dynamic random access memory
- Capacitor charge state indicates stored value
  - Whether the capacitor is charged or discharged indicates storage of 1 or 0
  - 1 capacitor
  - 1 access transistor

- Capacitor leaks through the RC path
  - DRAM cell loses charge over time
  - DRAM cell needs to be refreshed

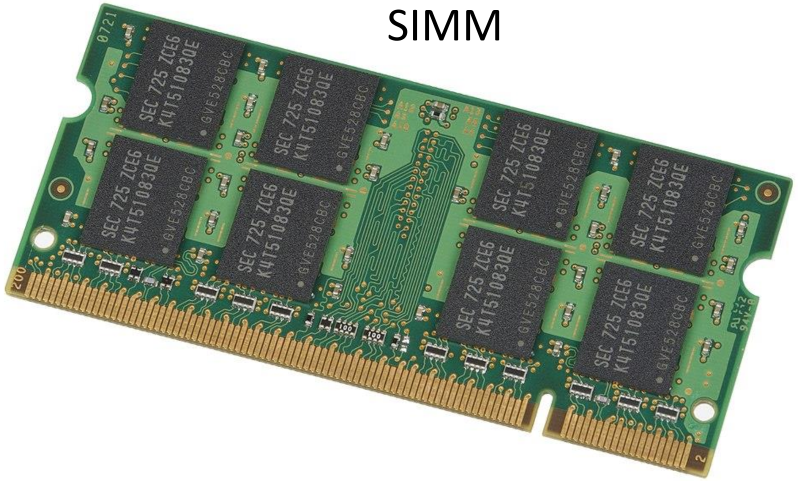


Refreshing in DRAM

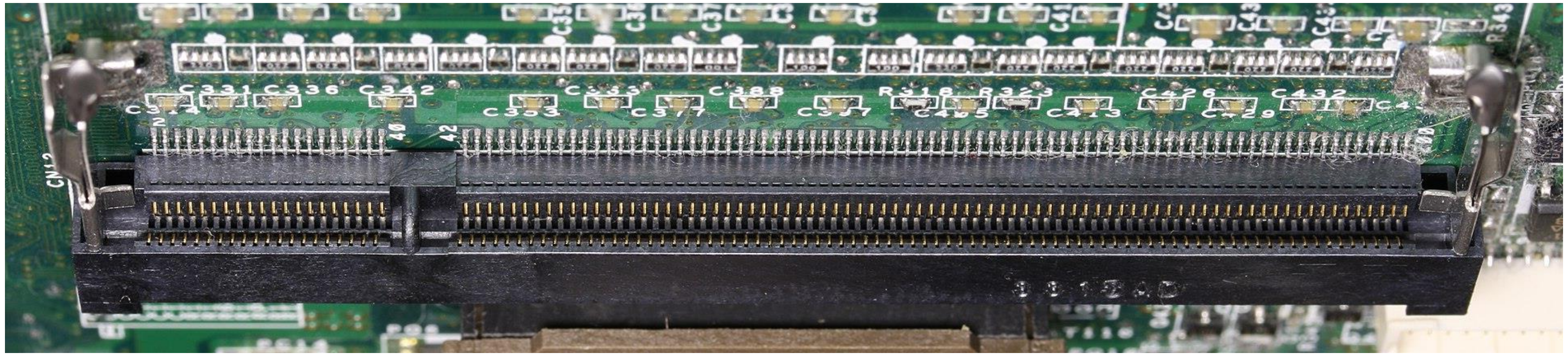
- Refresh circuit included on chip
- Disable chip
- Count through rows
- Read & Write back
- Takes time
- Slows down apparent performance

Computer Memory Modules

SIMM

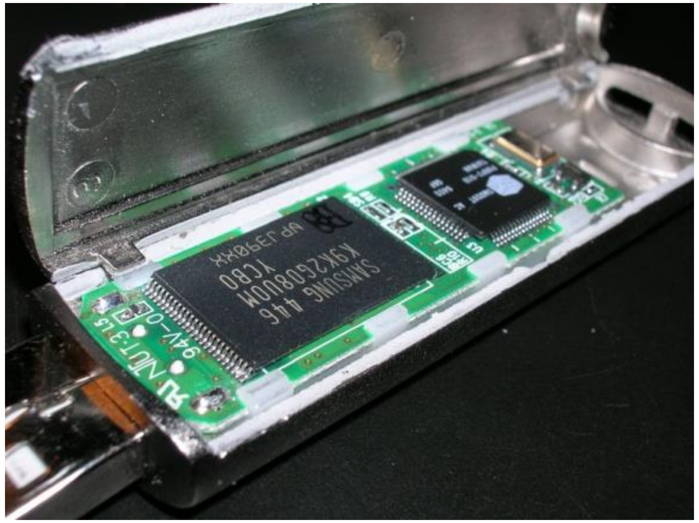


SO-DIMM SOCKET



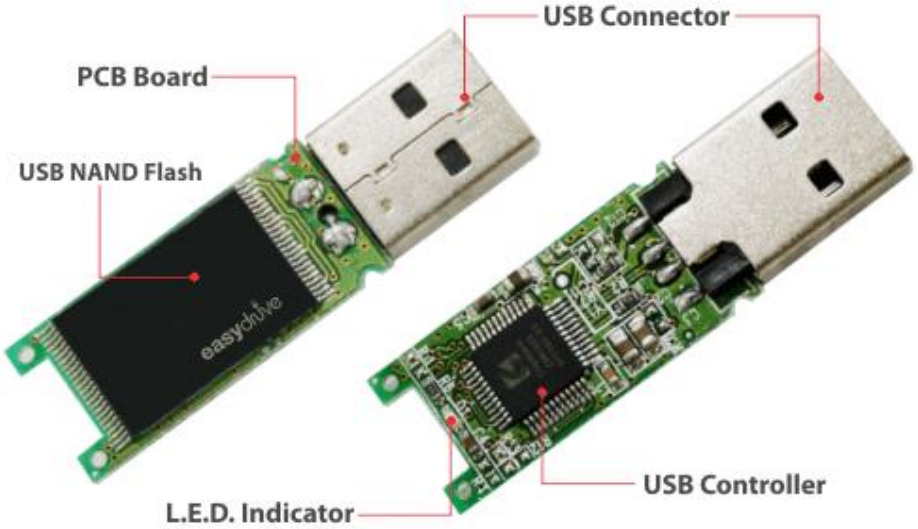
Flash Storage – SD Card

- Nonvolatile semiconductor storage
  - 100× – 1000× faster than disk
  - Smaller, lower power, more robust
  - But more $/GB (between disk and DRAM)

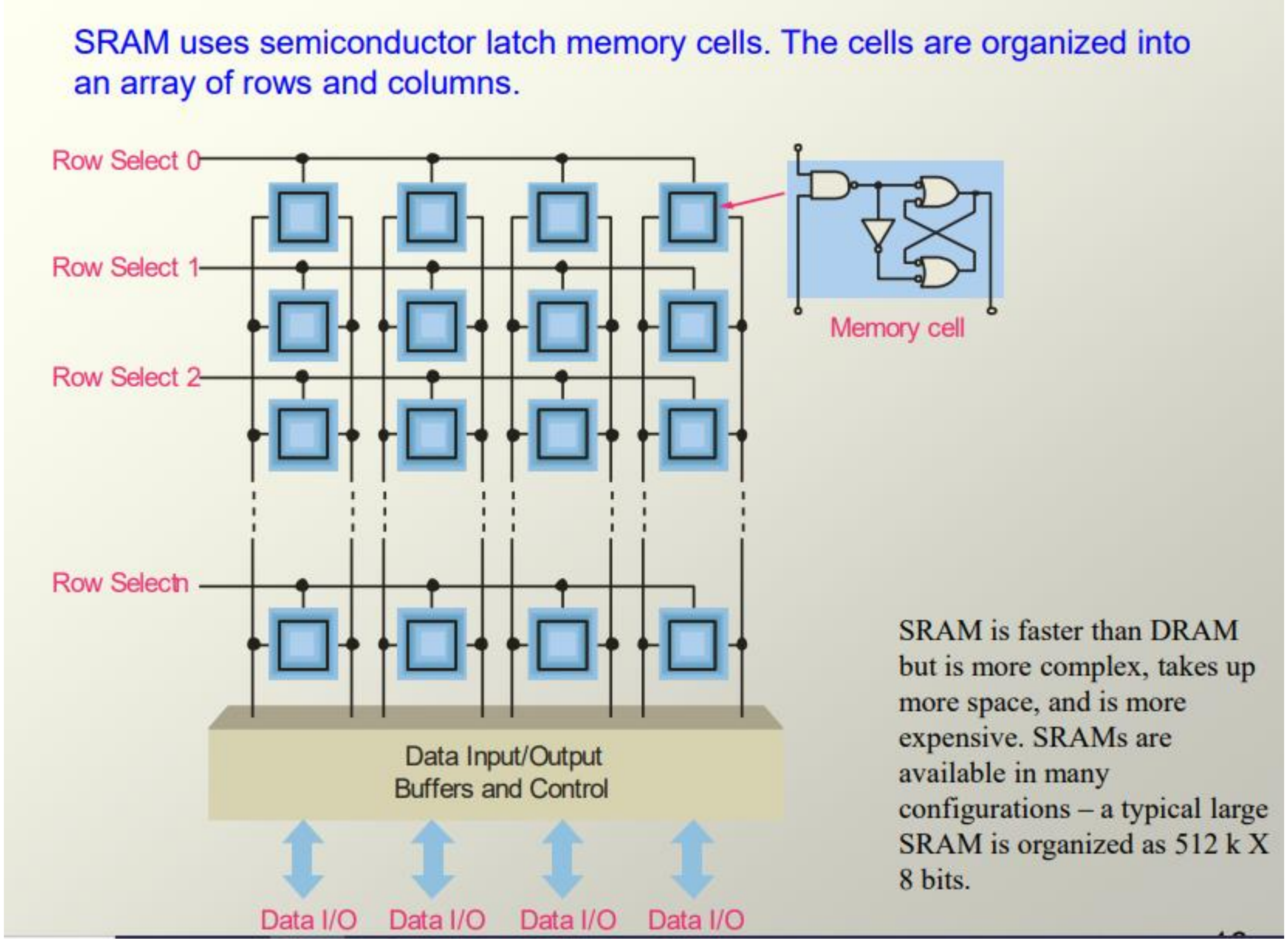


Flash Types – USB Disk

- NOR flash: bit cell like a NOR gate
  - Random read/write access
  - Used for instruction memory in embedded systems
- NAND flash: bit cell like a NAND gate
  - Denser (bits/area), but block-at-a-time access
  - Cheaper per GB
  - Used for USB keys, media storage, …
- Flash bits wears out after 1000’s of accesses
  - Not suitable for direct RAM or disk replacement
  - Wear leveling: remap data to less used blocks



Memory Cell Organization as Array

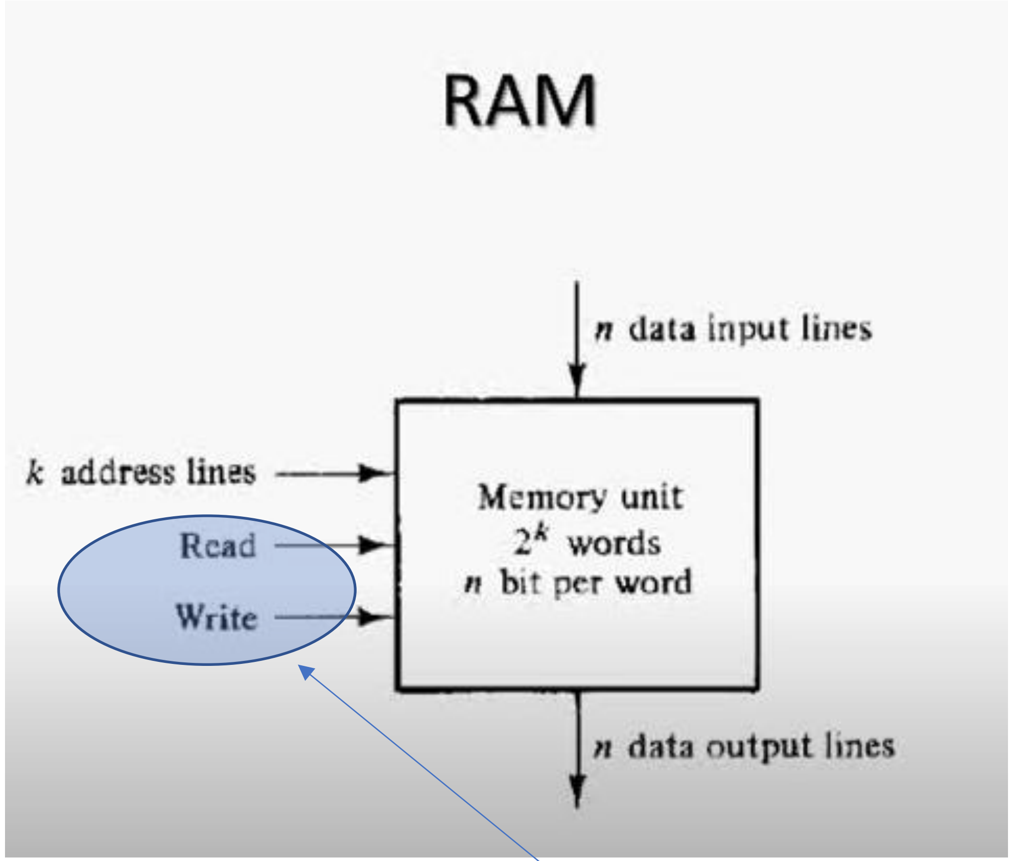


SRAM uses semiconductor latch memory cells. The cells are organized into an array of rows and columns.

SRAM is faster than DRAM but is more complex, takes up more space, and is more expensive. SRAMs are available in many configurations – a typical large SRAM is organized as 512 k X 8 bits.

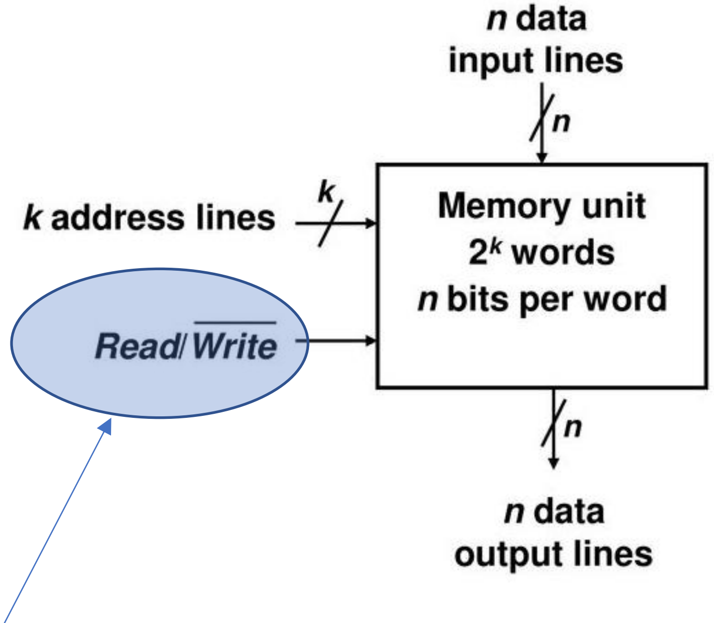
Memory Ports and Connection to CPU

Memory at Block Level with Read / Write



MEMORY UNIT

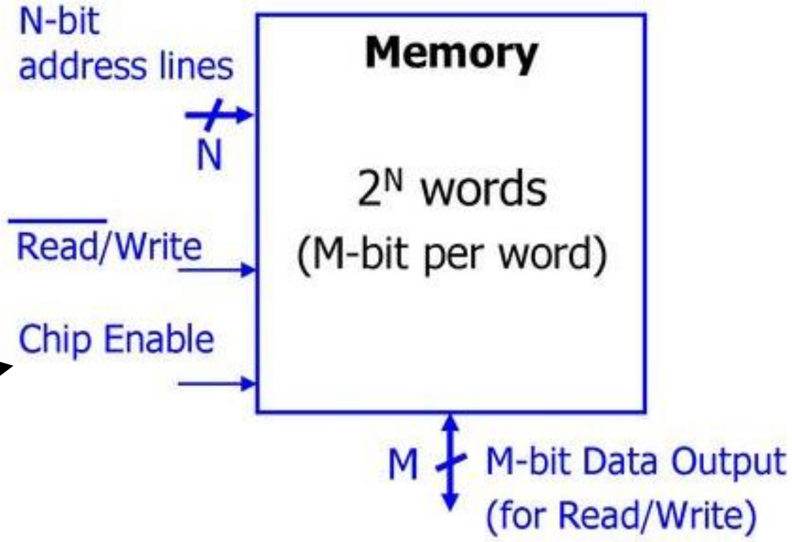
- Block diagram of a memory unit:



Two Ways of Read / Write signals

Block Diagram of Memory

- An M-bit data value can be read or written at each unique N-bit address



Chip Enable

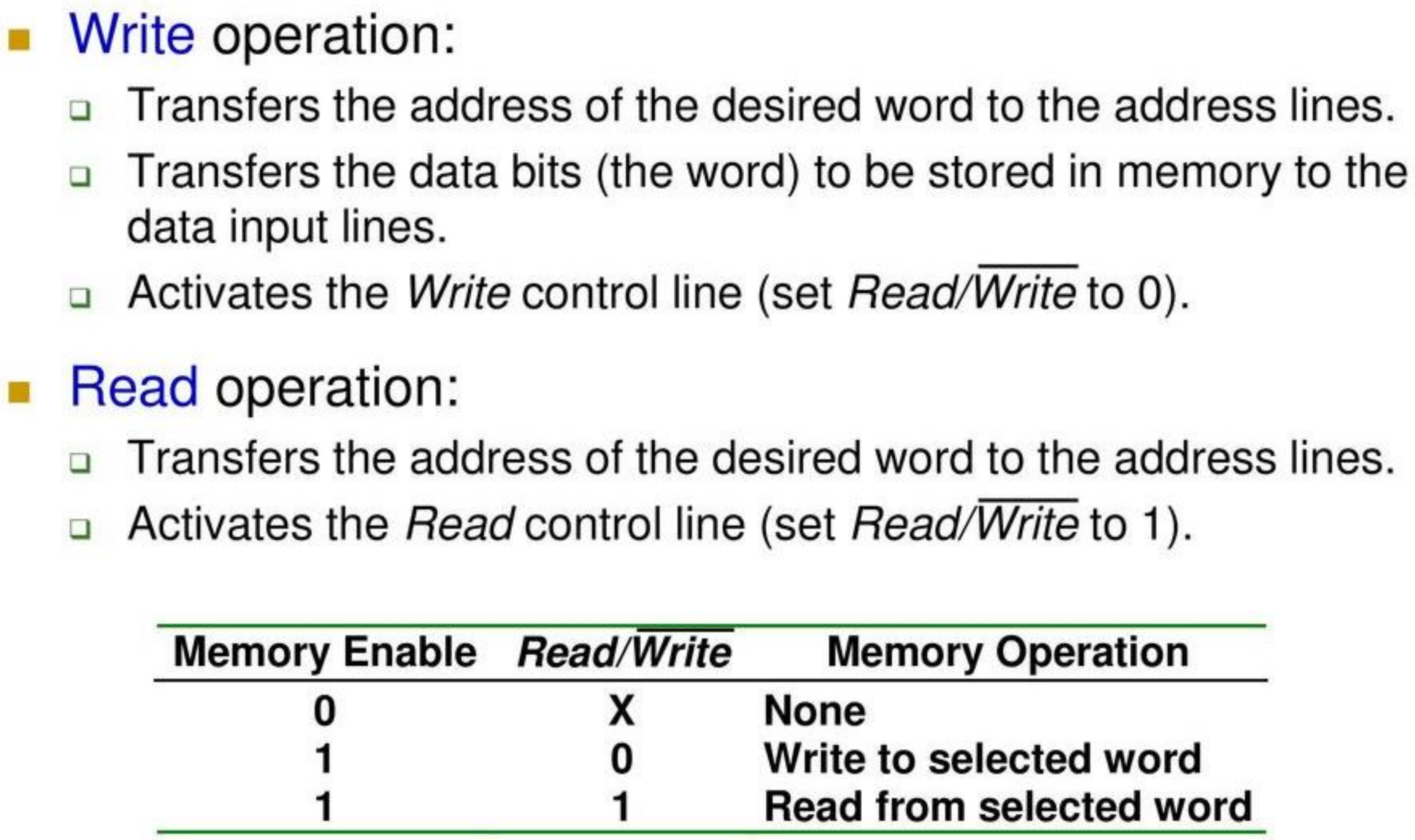
- Example: Byte-addressable 2MB memory
  - M = 8 (because of byte-addressability)
  - N = 21 (1 word = 8-bit)

RAM/ROM naming convention:

32 X 8, "32 by 8" => 32 8-bit words

1M X 1, "1 meg by 1" => 1M 1-bit words

Read / Write Memory Chip Operations



- Write operation:
  - Transfers the address of the desired word to the address lines.
  - Transfers the data bits (the word) to be stored in memory to the data input lines.
  - Activates the *Write* control line (set $Read/\overline{Write}$ to 0).
- Read operation:
  - Transfers the address of the desired word to the address lines.
  - Activates the *Read* control line (set $Read/\overline{Write}$ to 1).

Memory Enable	$Read/\overline{Write}$	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

Synchronous Memory: All Memory Read / Write Operations are Synchronized to Clock Signals

Monolithic View of Computer Memory

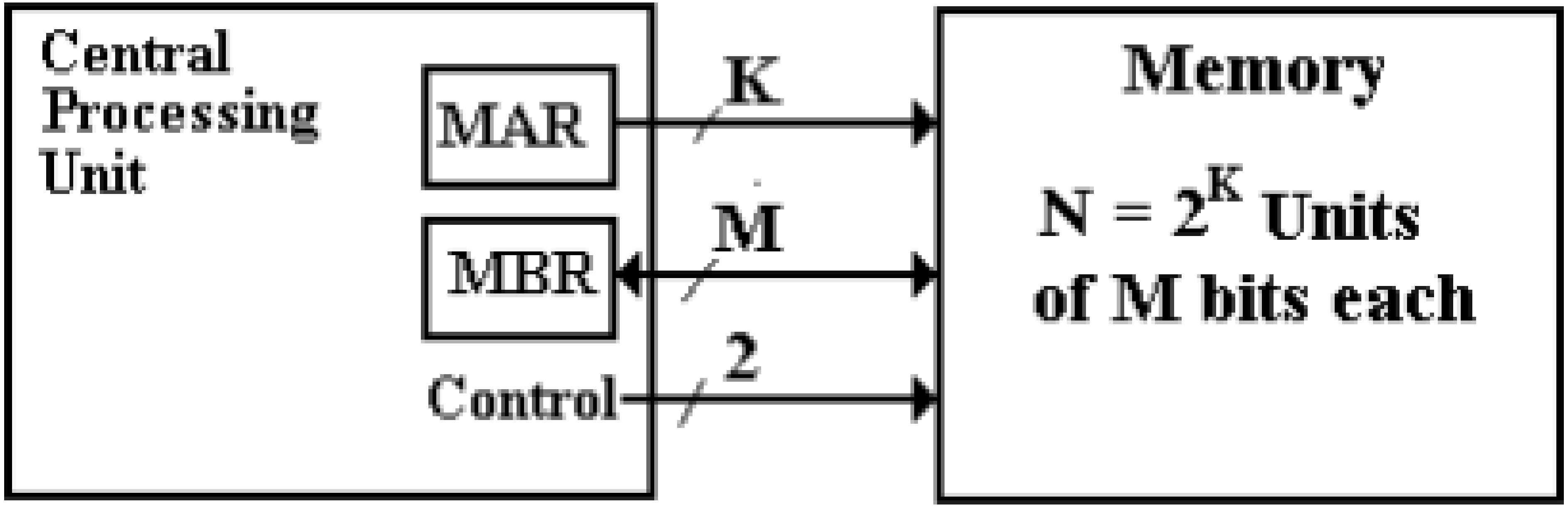
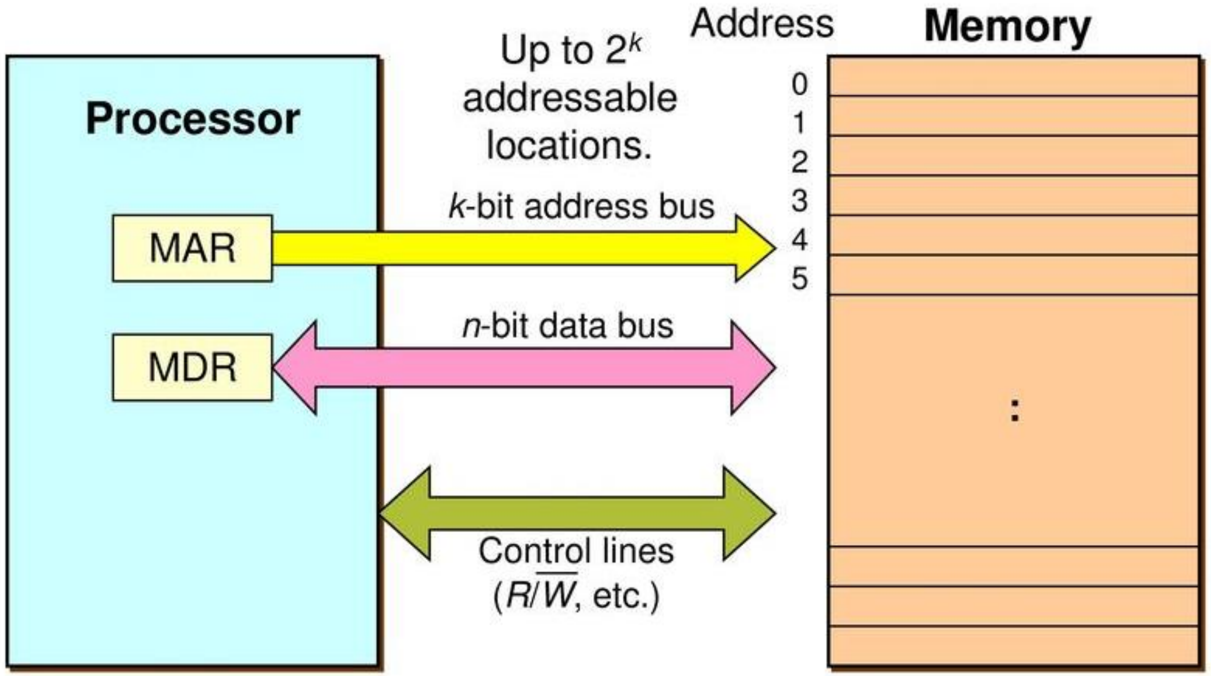


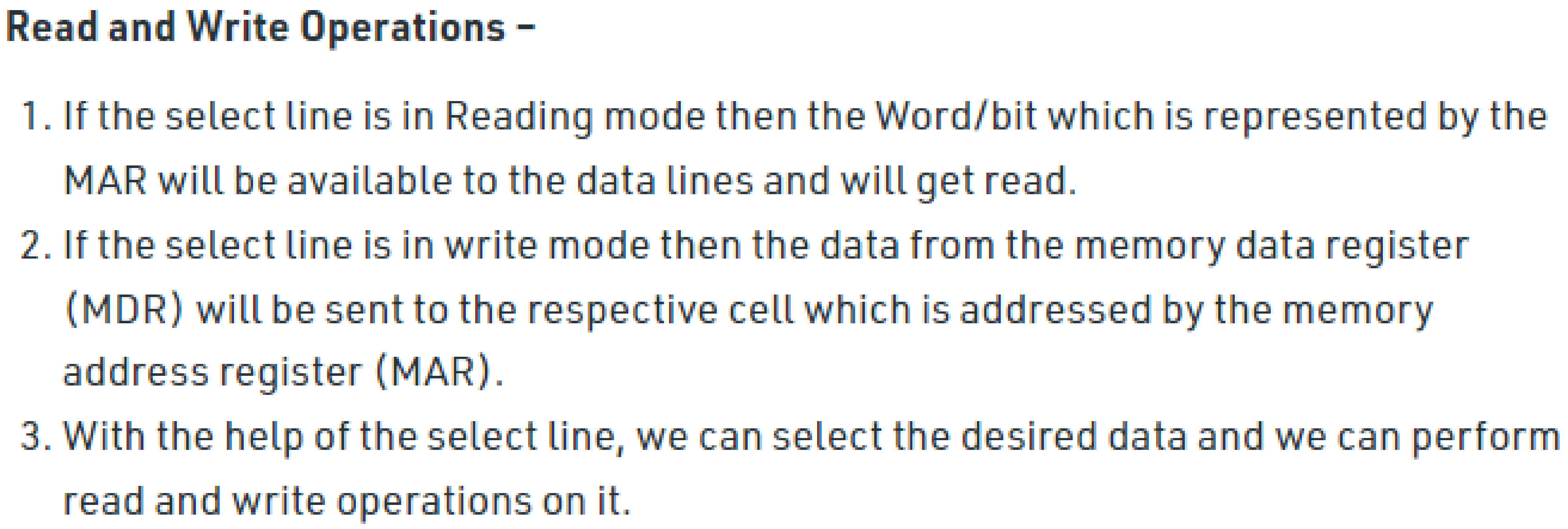
Figure: Monolithic View of Computer Memory

Memory Connections with CPU

Von Neumann Architecture


Processor
MAR
MDR
Up to 2^k addressable locations.
k-bit address bus
n-bit data bus
Control lines ($R/\overline{W}$, etc.)
Address
Memory
0
1
2
3
4
5
:

Read and Write Operations from CPU

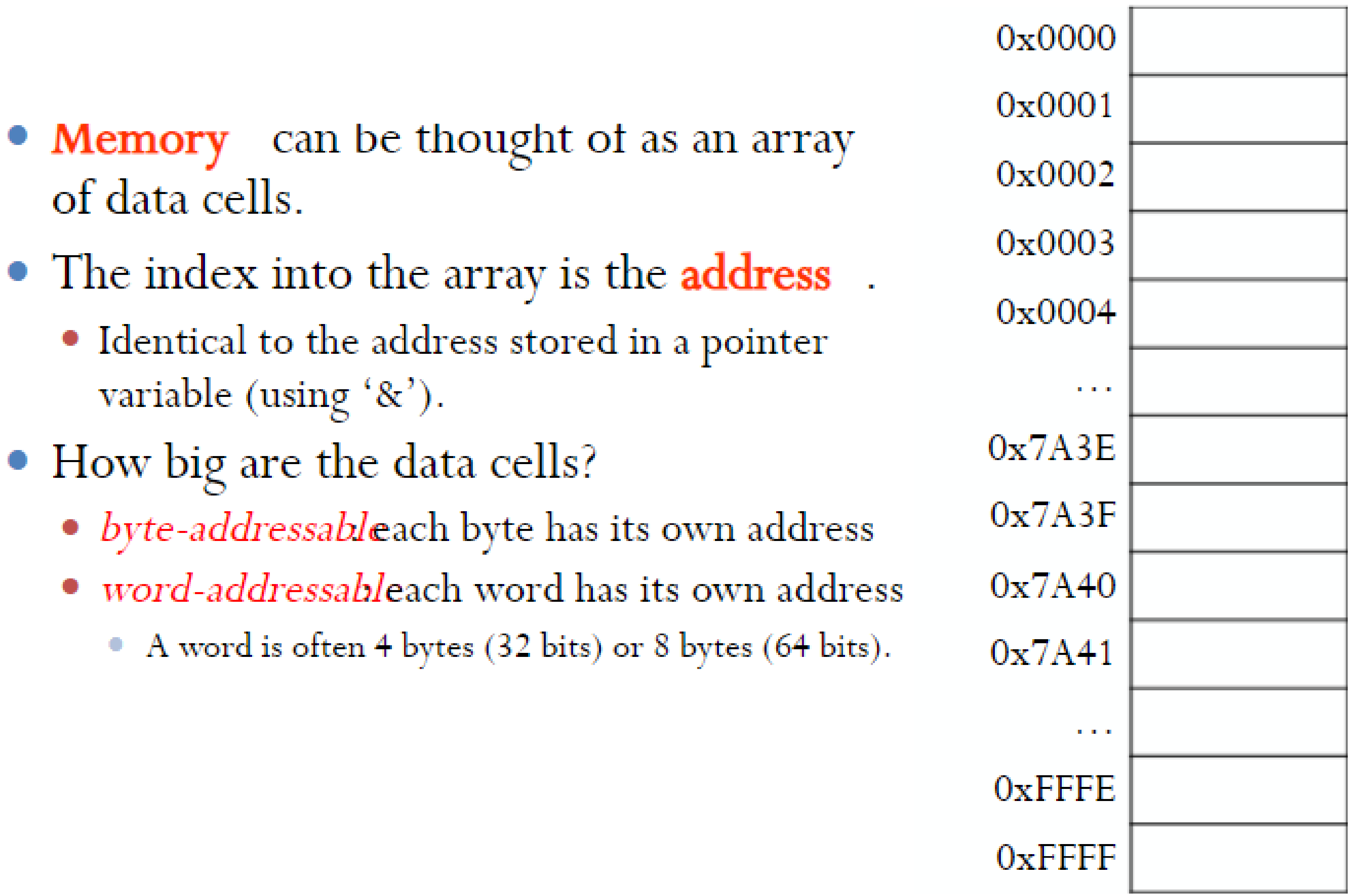


Read and Write Operations –

1. If the select line is in Reading mode then the Word/bit which is represented by the MAR will be available to the data lines and will get read.
2. If the select line is in write mode then the data from the memory data register (MDR) will be sent to the respective cell which is addressed by the memory address register (MAR).
3. With the help of the select line, we can select the desired data and we can perform read and write operations on it.

Memory Address Mechanism

Memory Addressing



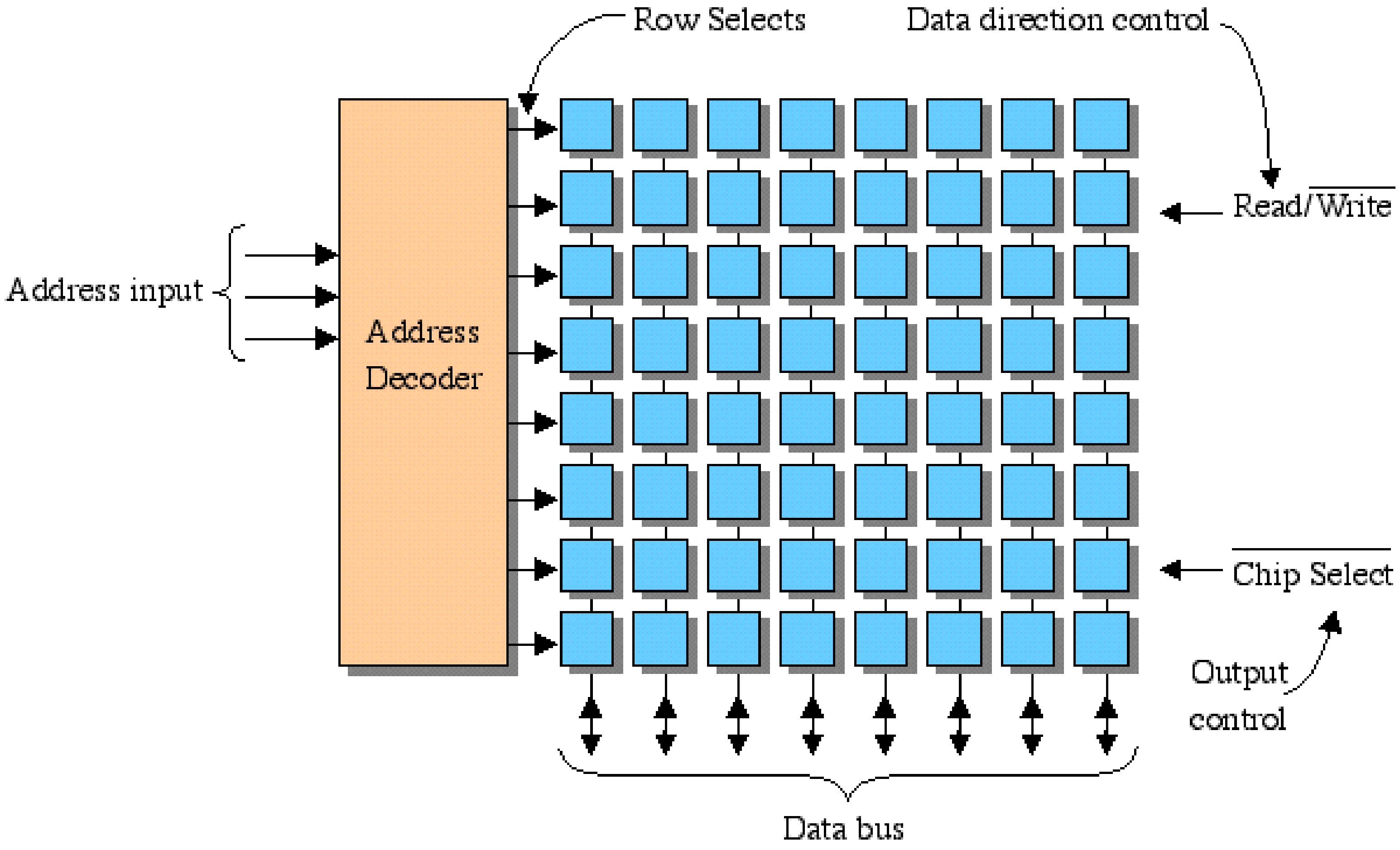
- **Memory** can be thought of as an array of data cells.
- The index into the array is the **address**.
  - Identical to the address stored in a pointer variable (using ‘&’).
- How big are the data cells?
  - *byte-addressable*: each byte has its own address
  - *word-addressable*: each word has its own address
    - A word is often 4 bytes (32 bits) or 8 bytes (64 bits).

0x0000	
0x0001	
0x0002	
0x0003	
0x0004	
. . .	
0x7A3E	
0x7A3F	
0x7A40	
0x7A41	
. . .	
0xFFFE	
0xFFFF	

Decoding Memory Addresses

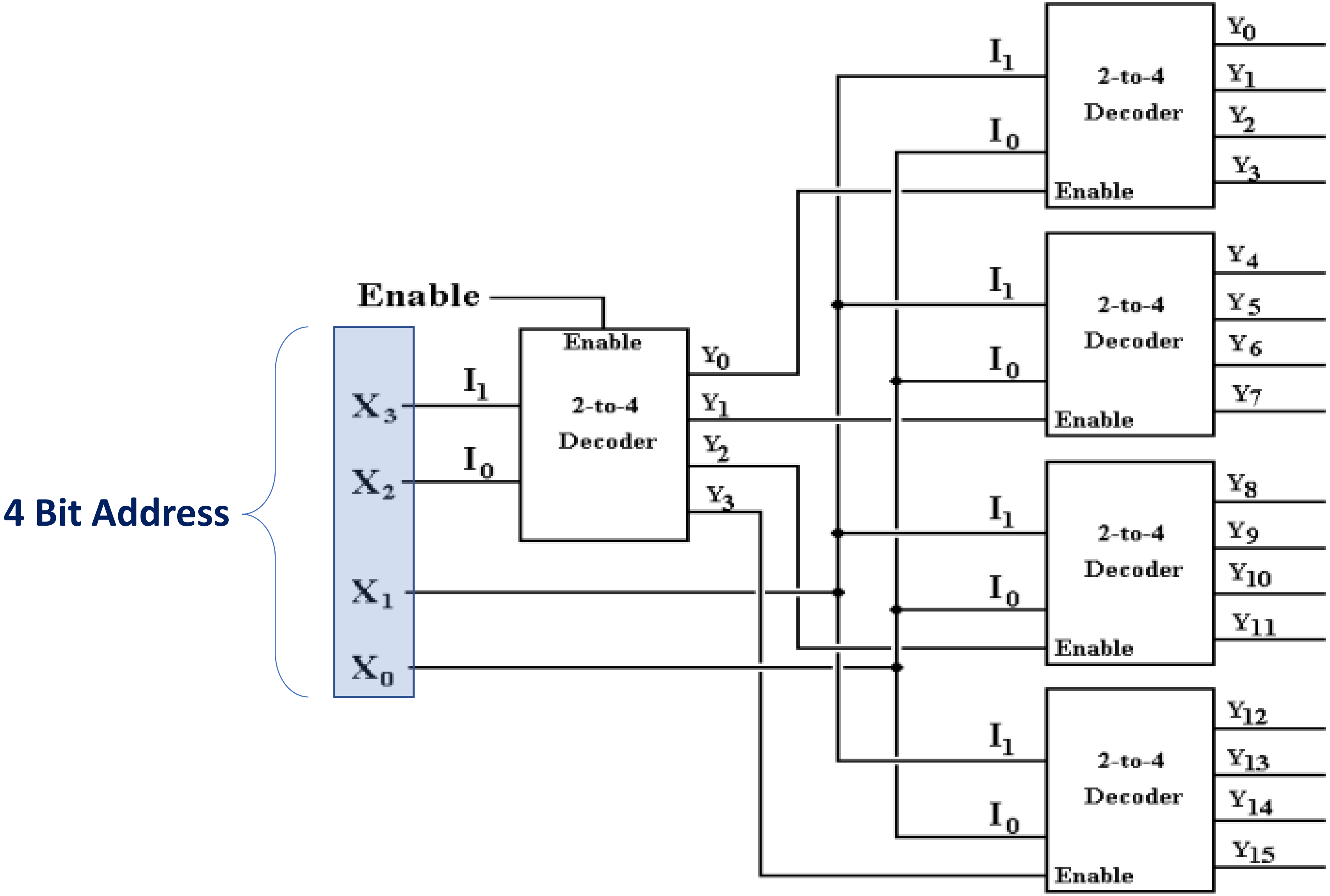
- A decoder is used to determine which cell is accessed:
  - Converts an address into the enable lines for the memory cell.
  - Makes sure that only one memory cell is enabled at a time.
- Useful for the decoder to have an enable input that can enable and disable the entire memory.
  - When a memory access occurs, memory enable is set to 1.
    - Decoder behaves normally.
  - When memory is not used, memory enable is set to 0.
    - All outputs of the decoder are 0.
    - No memory cell is enabled.

Row Select Decoders – 8 x 8 Memory Array

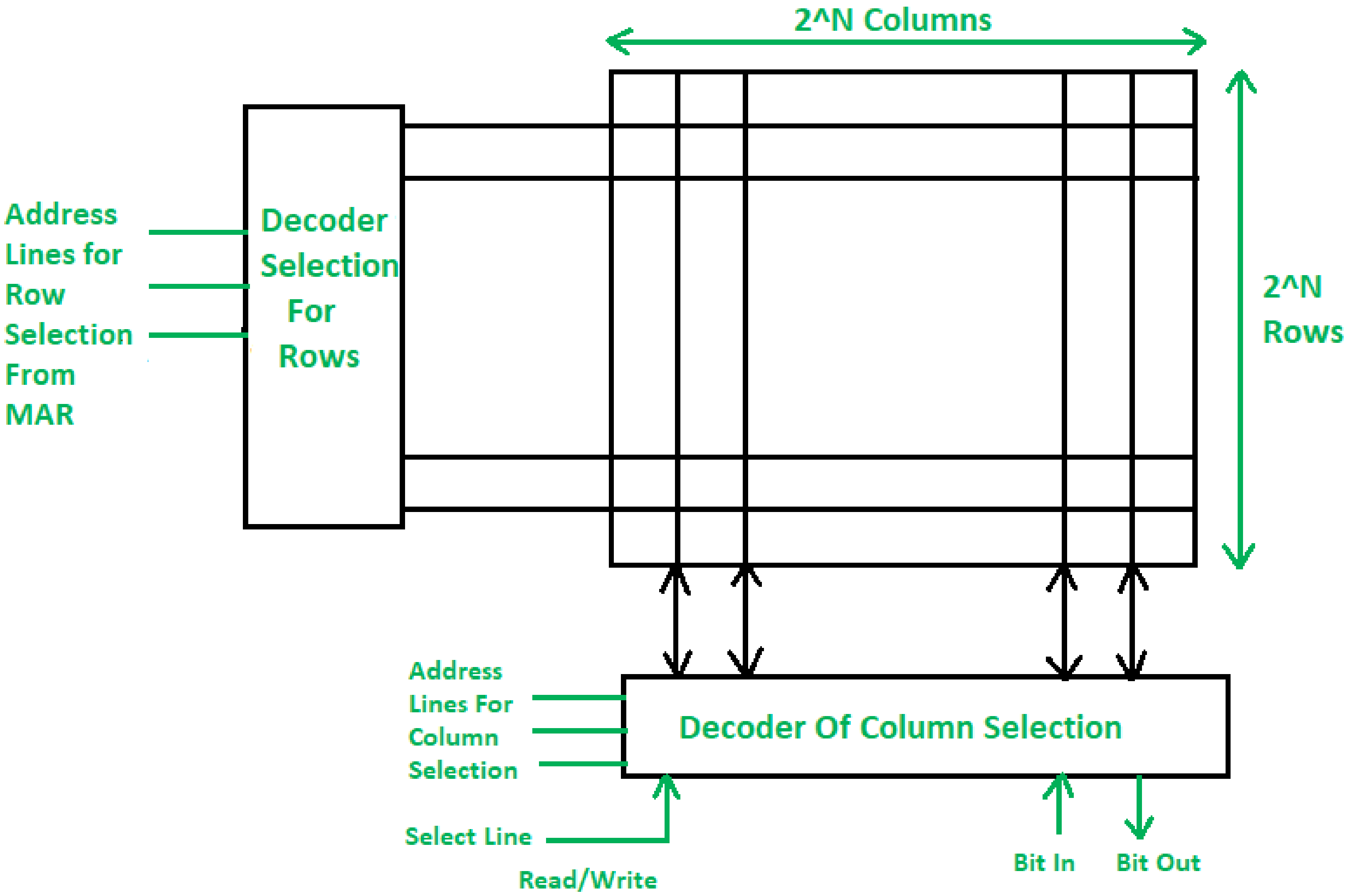


2D Array only selects entire ROW

Address Row Decoders - Example

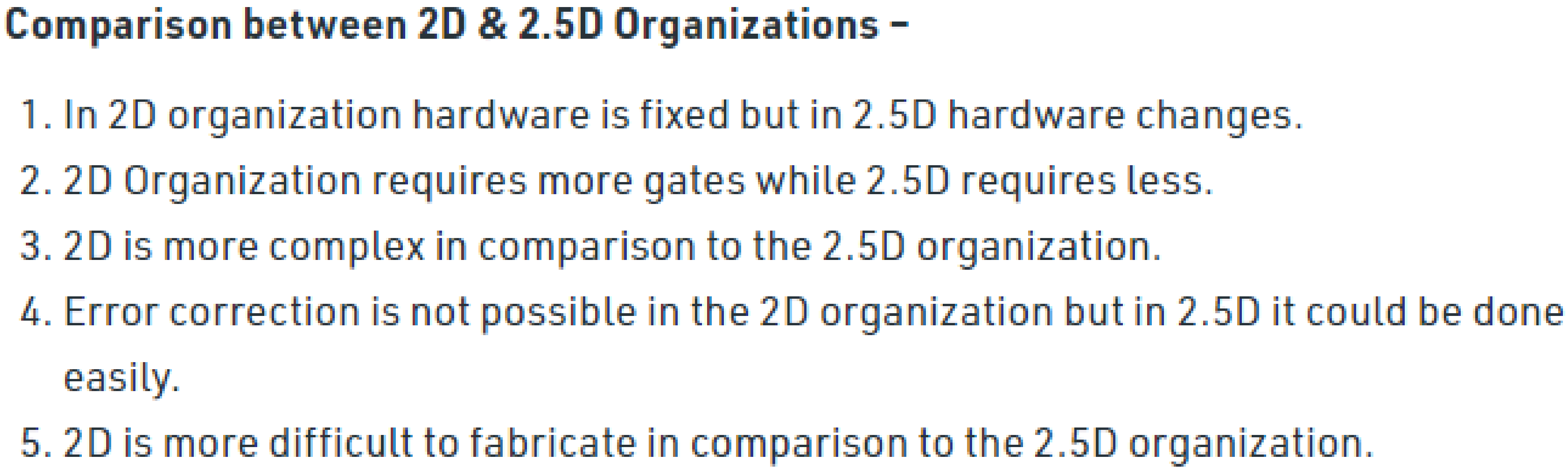


2.5 D Memory Organization


2^N Columns
Address Lines for Row Selection From MAR
Decoder Selection For Rows
2^N Rows
Address Lines For Column Selection
Decoder Of Column Selection
Select Line
Read/Write
Bit In
Bit Out

2.5D Memory Organization can Select ROWs as well as COLUMNs

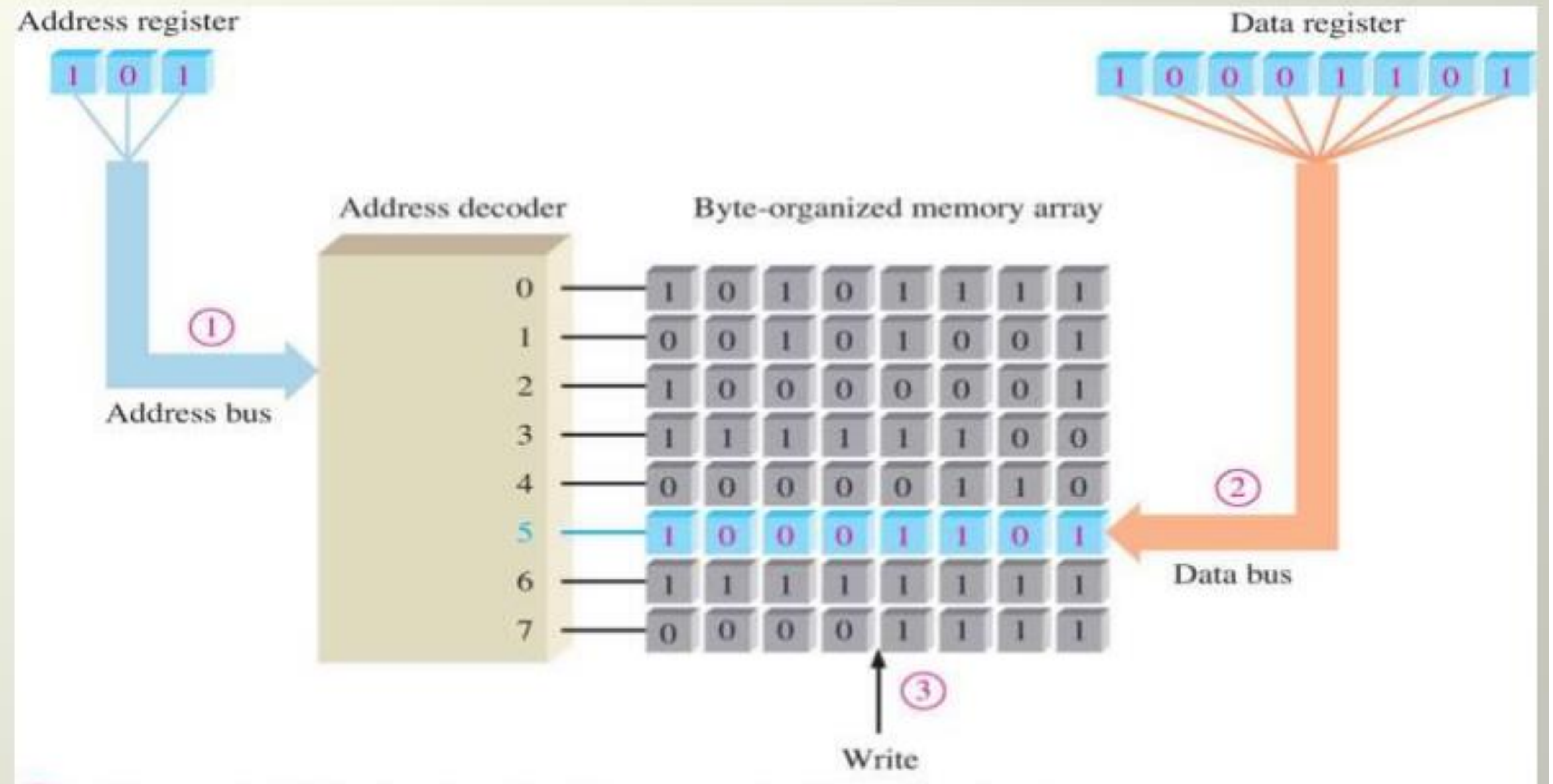
Comparison between 2 D and 2.5 D Organization



Comparison between 2D & 2.5D Organizations –

1. In 2D organization hardware is fixed but in 2.5D hardware changes.
2. 2D Organization requires more gates while 2.5D requires less.
3. 2D is more complex in comparison to the 2.5D organization.
4. Error correction is not possible in the 2D organization but in 2.5D it could be done easily.
5. 2D is more difficult to fabricate in comparison to the 2.5D organization.

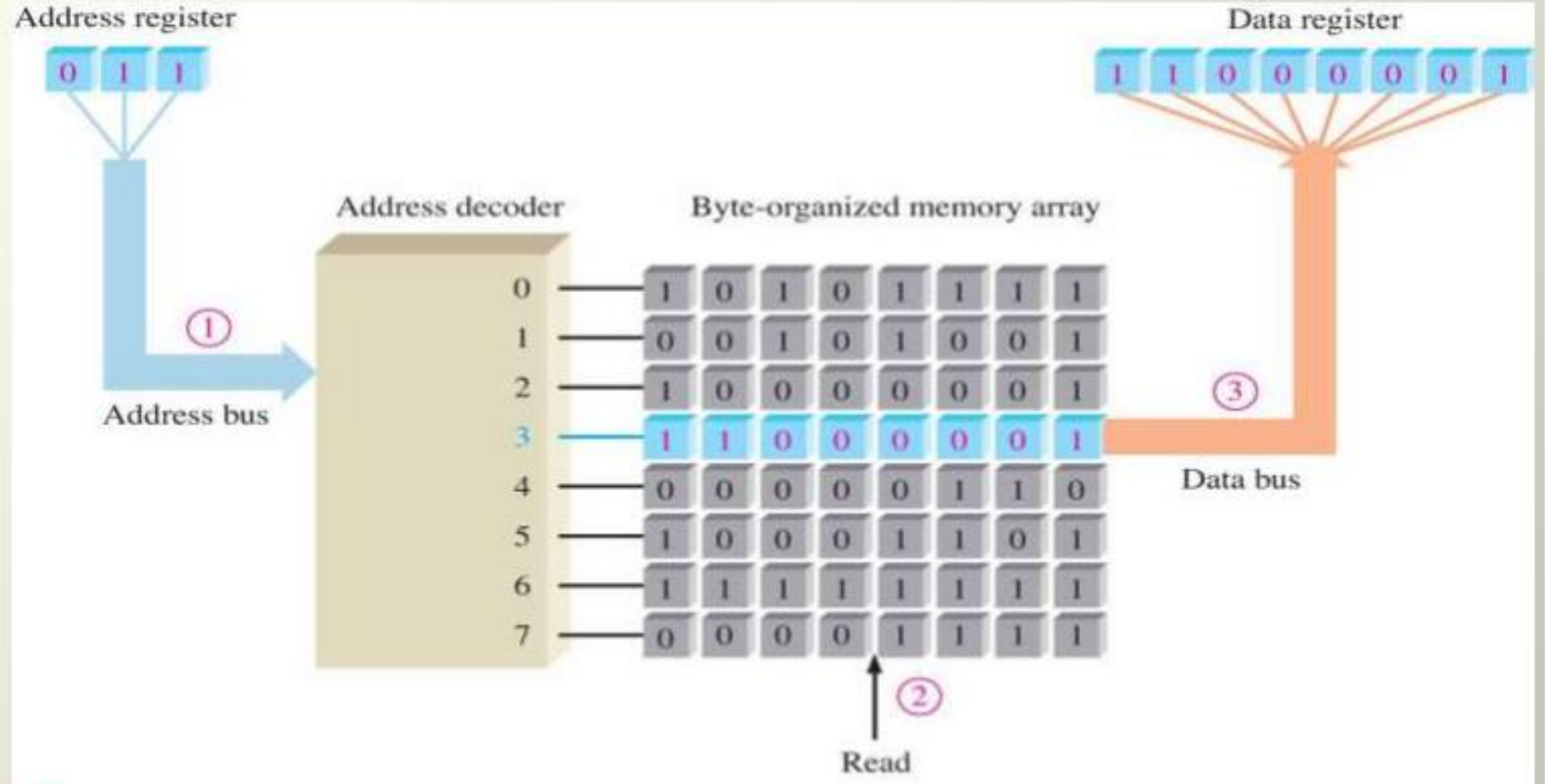
Memory Write Operation



0	1	0	1	0	1	1	1	1
1	0	0	1	0	1	0	0	1
2	1	0	0	0	0	0	0	1
3	1	1	1	1	1	1	0	0
4	0	0	0	0	0	1	1	0
5	1	0	0	0	1	1	0	1
6	1	1	1	1	1	1	1	1
7	0	0	0	0	1	1	1	1

1. Address code 101 is placed on the address bus and address 5 is selected.
2. Data byte is placed on the data bus.
3. Write command causes the data byte to be stored in address 5, replacing previous data.

Memory Read Operation



0	1	0	1	0	1	1	1	1
1	0	0	1	0	1	0	0	1
2	1	0	0	0	0	0	0	1
3	1	1	0	0	0	0	0	1
4	0	0	0	0	0	1	1	0
5	1	0	0	0	1	1	0	1
6	1	1	1	1	1	1	1	1
7	0	0	0	0	1	1	1	1

① Address code 011 is placed on the address bus and address 3 is selected.

② Read command is applied.

③ The contents of address 3 is placed on the data bus and shifted into data register.
The contents of address 3 is not erased by the read operation.

Readings

- Chap 5 of P&H Textbook

Acknowledge: Youtube channel David Black-Schaffer for some diagrams