CS / EE 320
Computer Organization and Assembly Language
Spring 2023
Lecture 22

Shahid Masud

Topics: Memory Connections, Memory Arrays, Memory Organization, Memory Capacity and Configuration

Topics

- Some review from previous lecture
- Building Arrays from Memory Cells
- Modular Approach to Building Complex Memory
- CPU to Memory Connections
- Address Decoding Schemes
- 2D and 2.5D Memory Organization
- Examples of Address Decoding and Modular Memory Construction

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**!

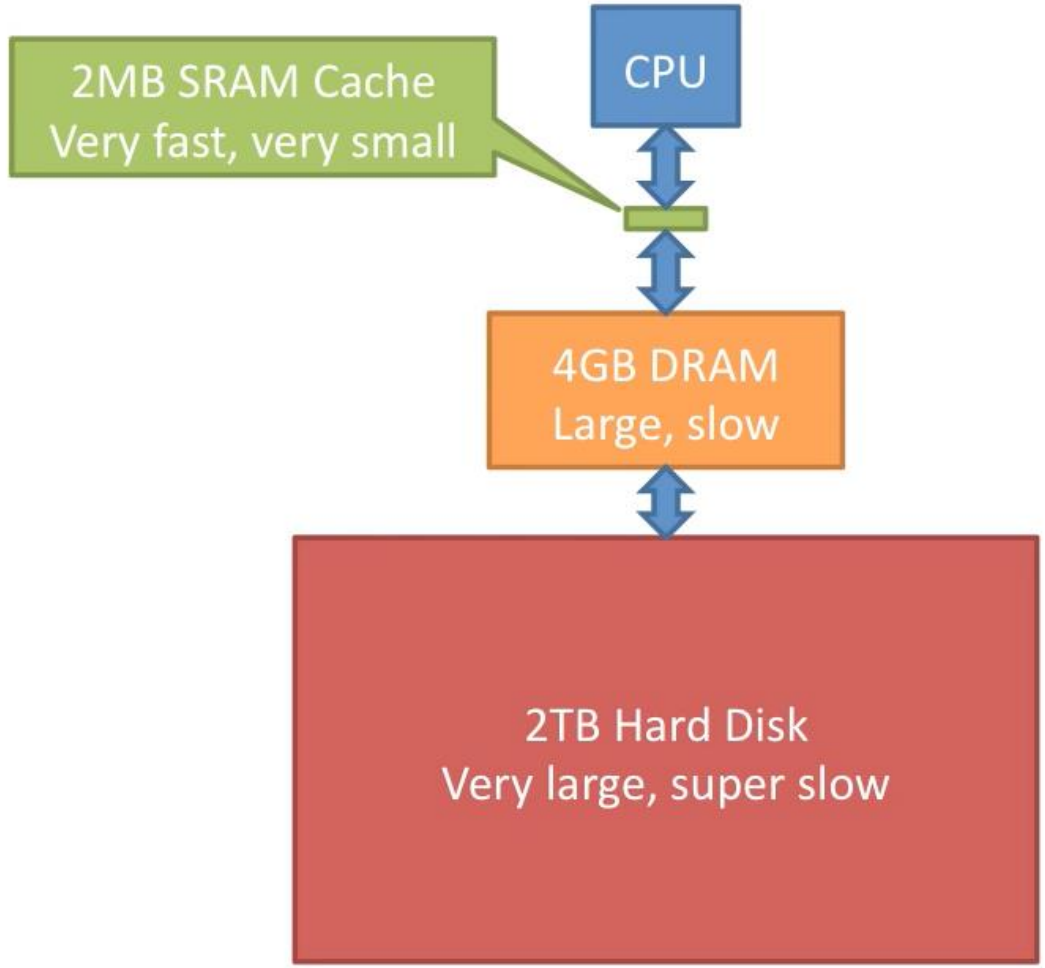
Disks are big, but super slow

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	

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 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

Semiconductor Memory, RAM and ROM

- RAM
  - Misnamed as all semiconductor memory is random 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
- Simpler construction
- Smaller per bit
- Less expensive
- Need refresh circuits
- Slower
- Main memory

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.

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		

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

Memory Construction

Memory Cell Operation

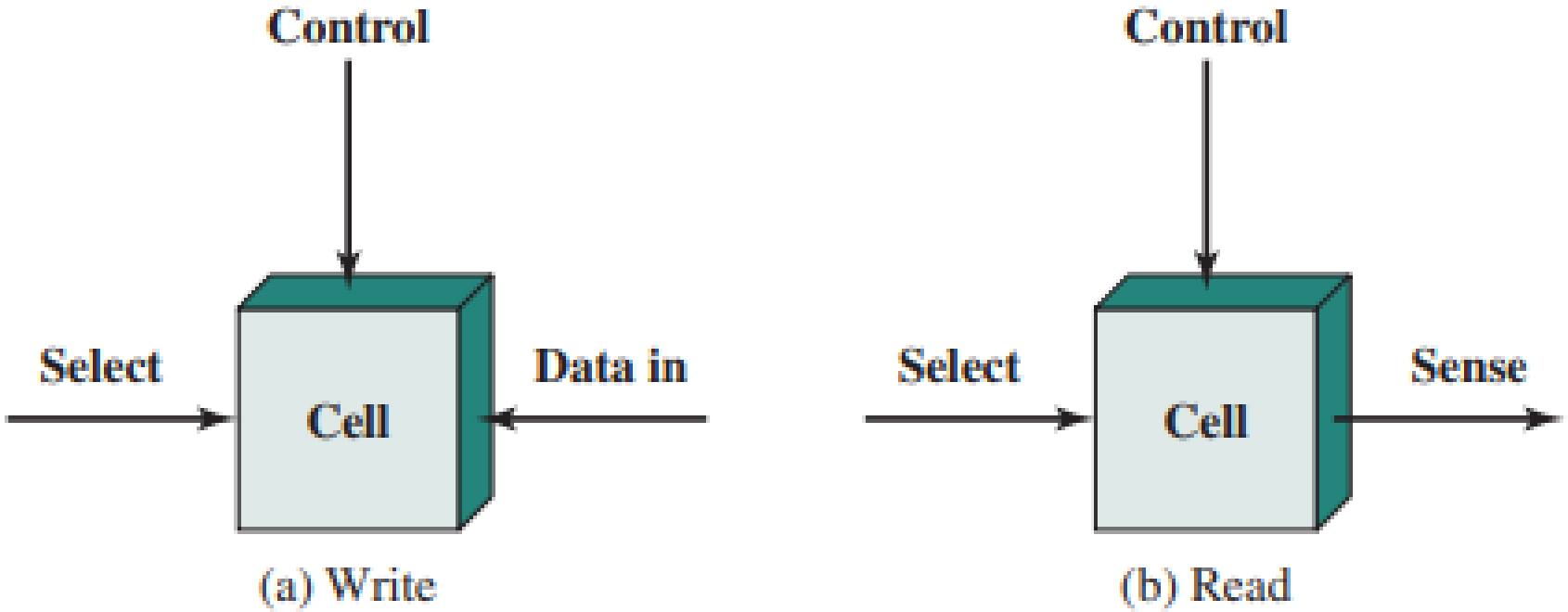
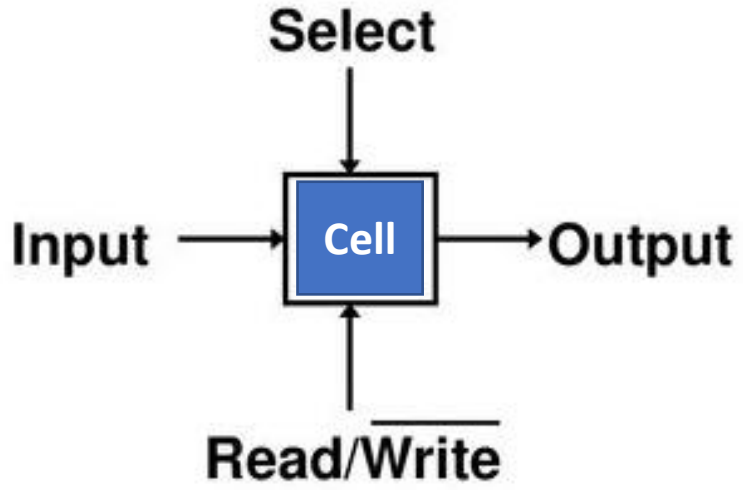
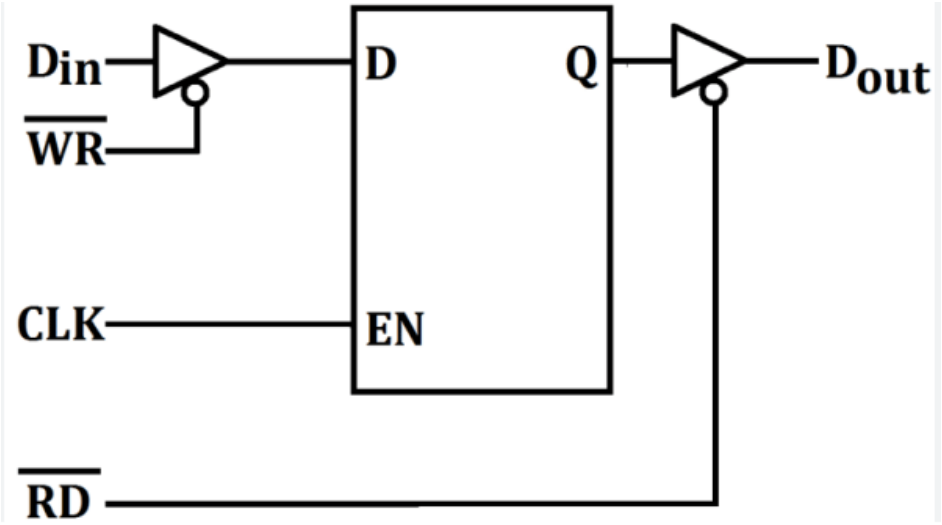


Figure 5.1 Memory Cell Operation

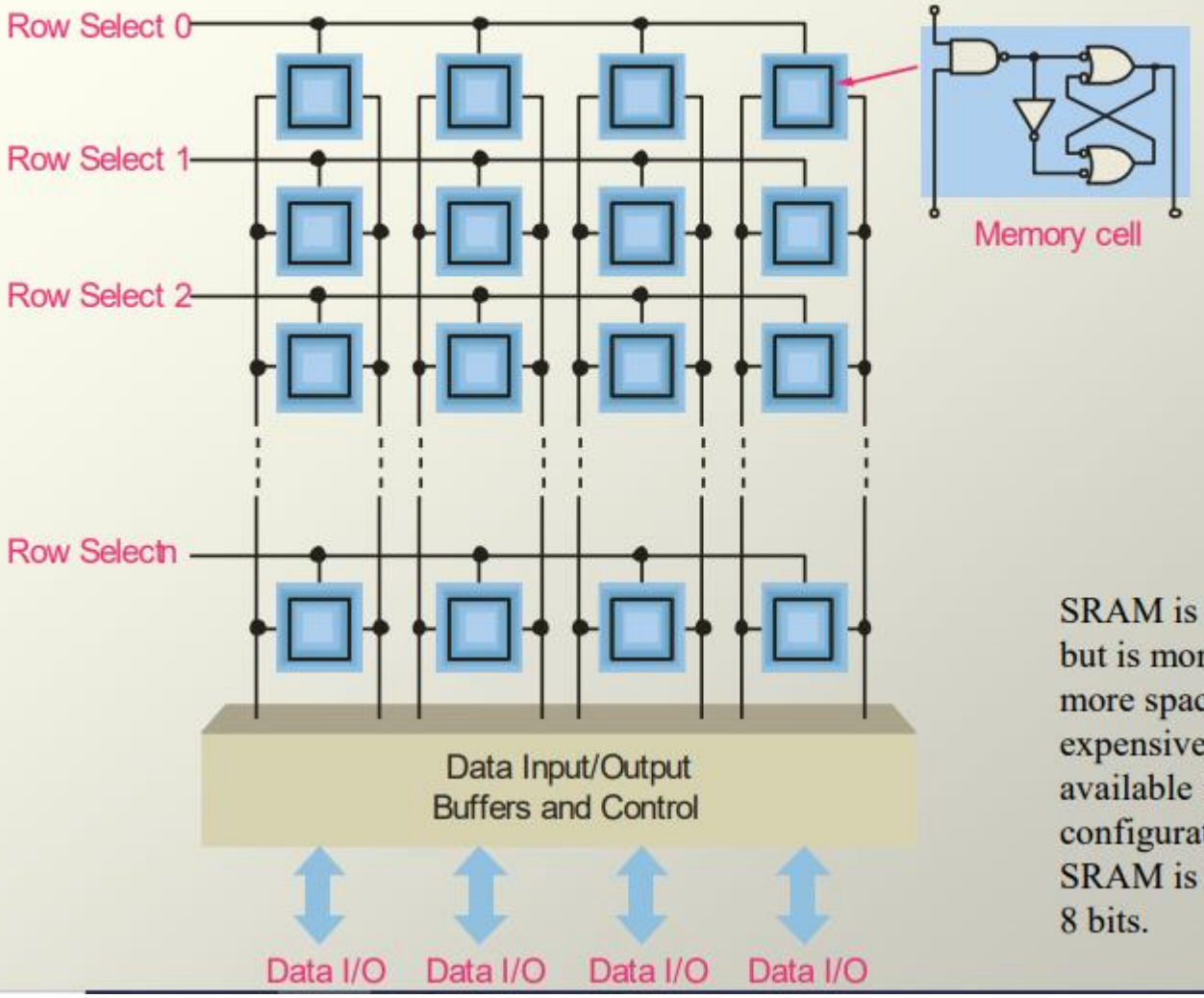
Memory Cell Design





Memory Cell Organization

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



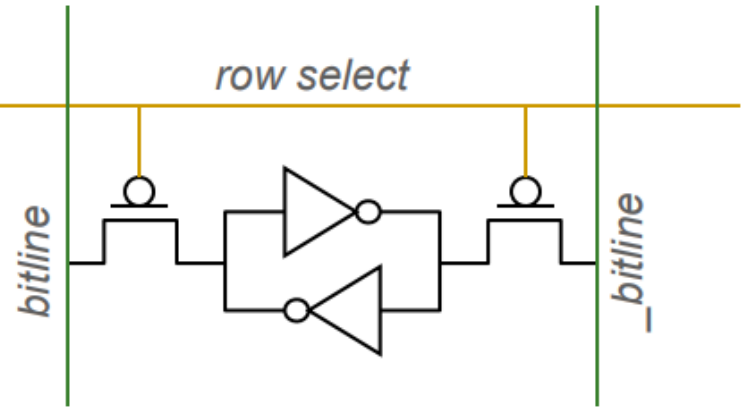
SRAM is faster than DRAM but is more complex, takes up more space, and is more expensive. SRAMs are available in many configurations – a typical large SRAM is organized as 512 k X 8 bits.

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

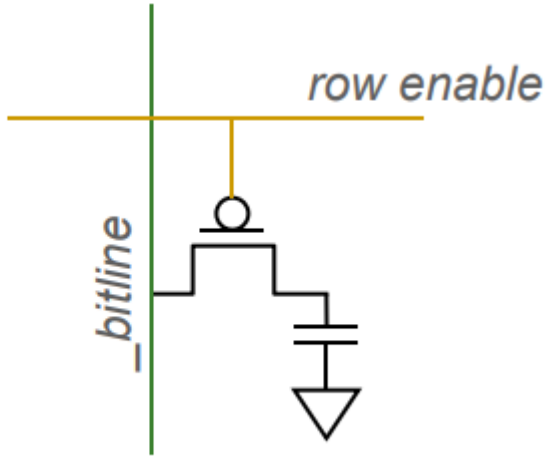
SRAM Cell Technology

- **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

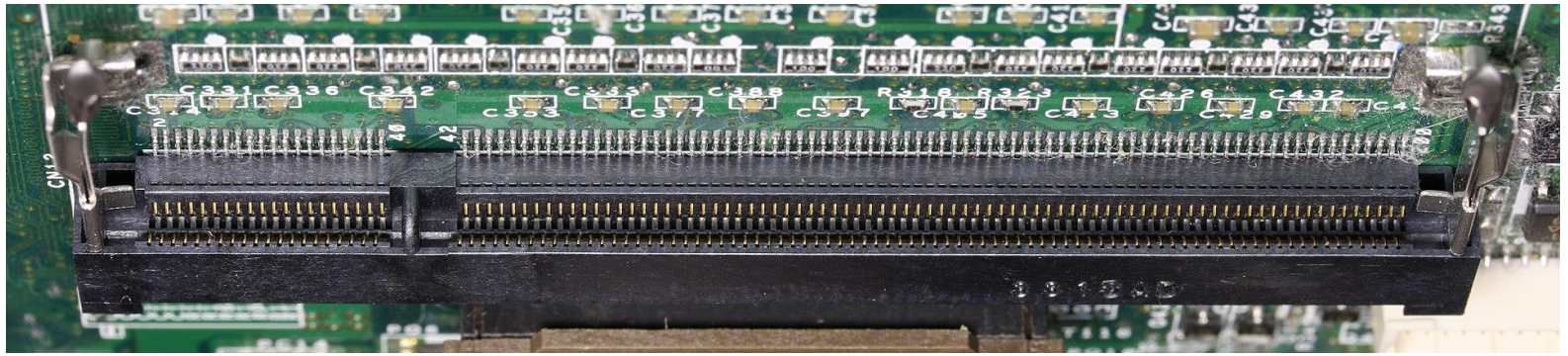
- Refresh circuit included on chip
- Disable chip
- Count through rows
- Read & Write back
- Takes time
- Slows down apparent performance

Computer Memory Modules

Crucial DDR DIMM

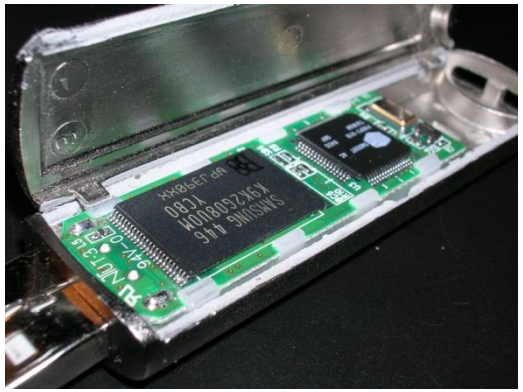
SIMM

SO-DIMM SOCKET



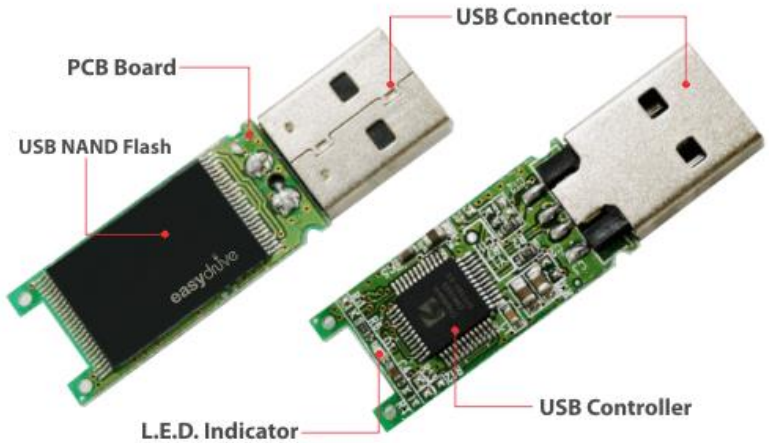
Flash Storage – SD Card

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



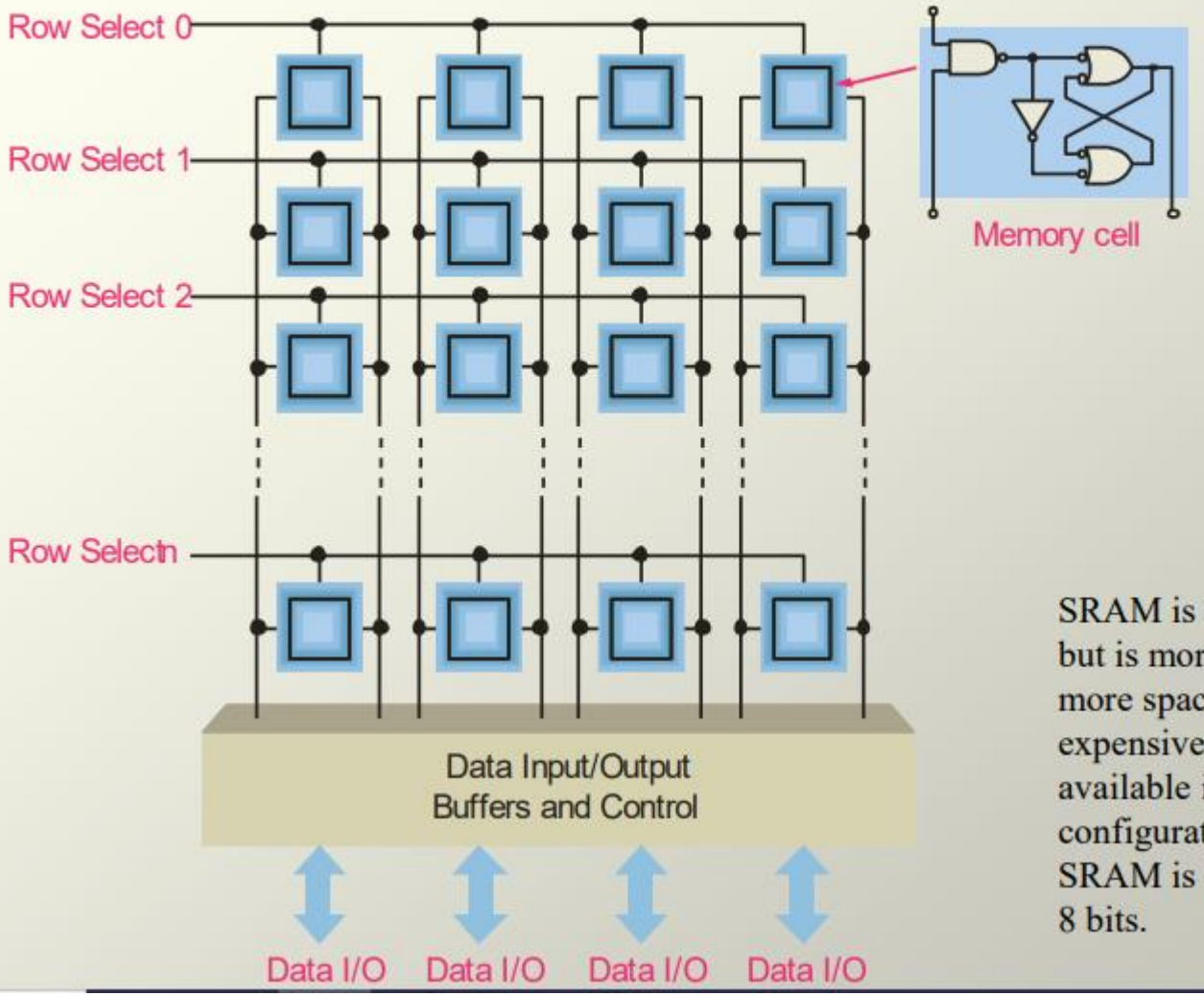
Flash Types – USB Disk

- 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

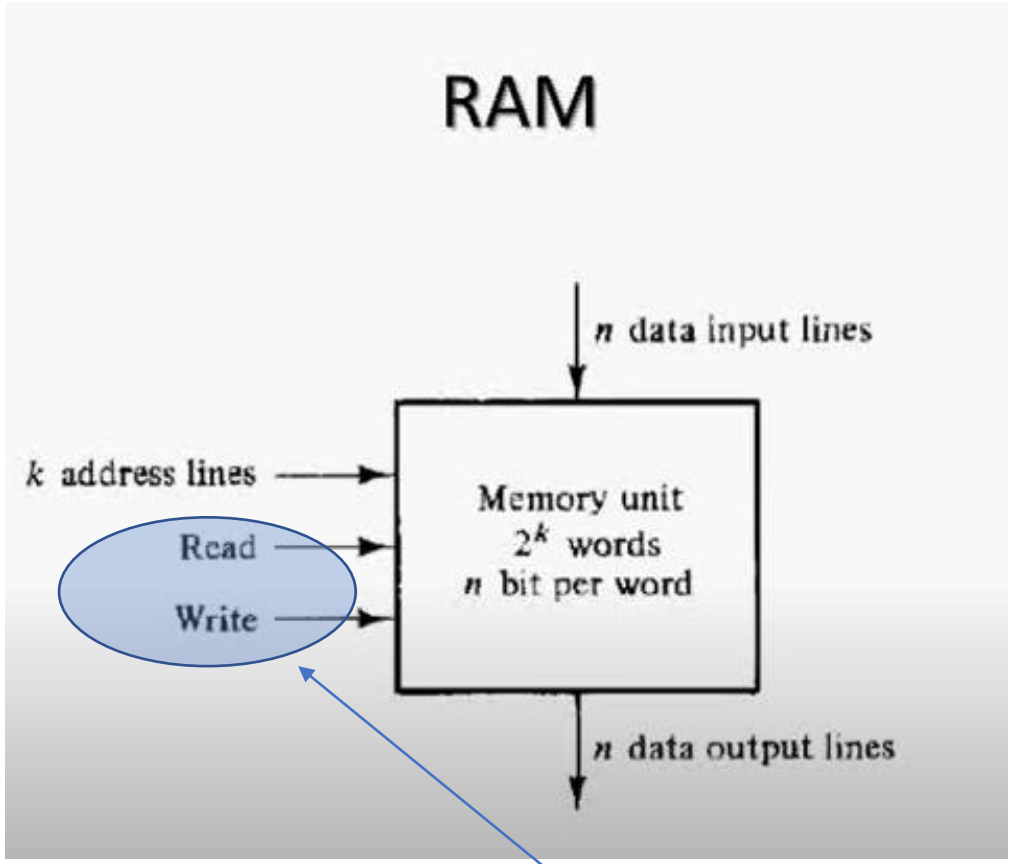
SRAM uses semiconductor latch memory cells. The cells are organized into an array of rows and columns.



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.

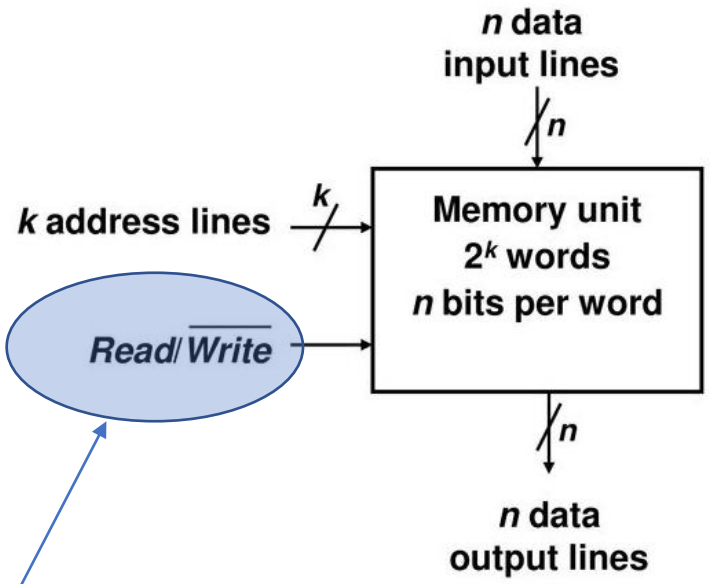
Memory Ports and Connection to CPU

Memory at Block Level with Read / Write



MEMORY UNIT

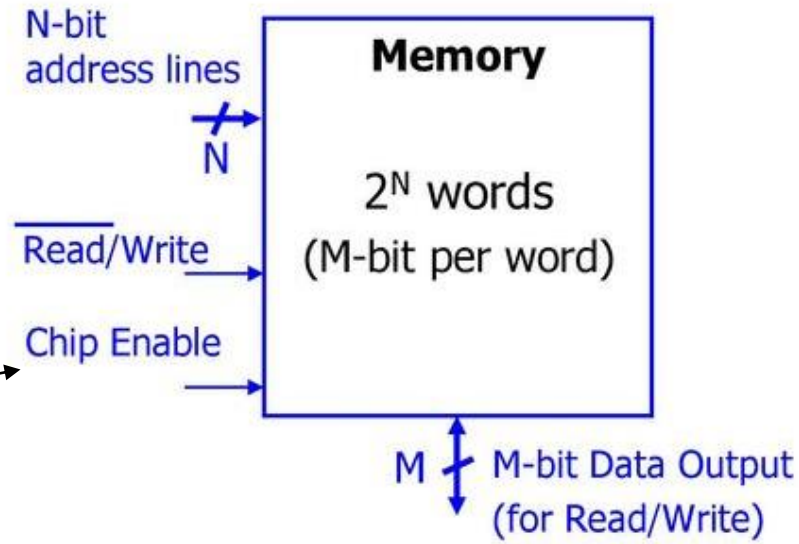
- Block diagram of a memory unit:



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

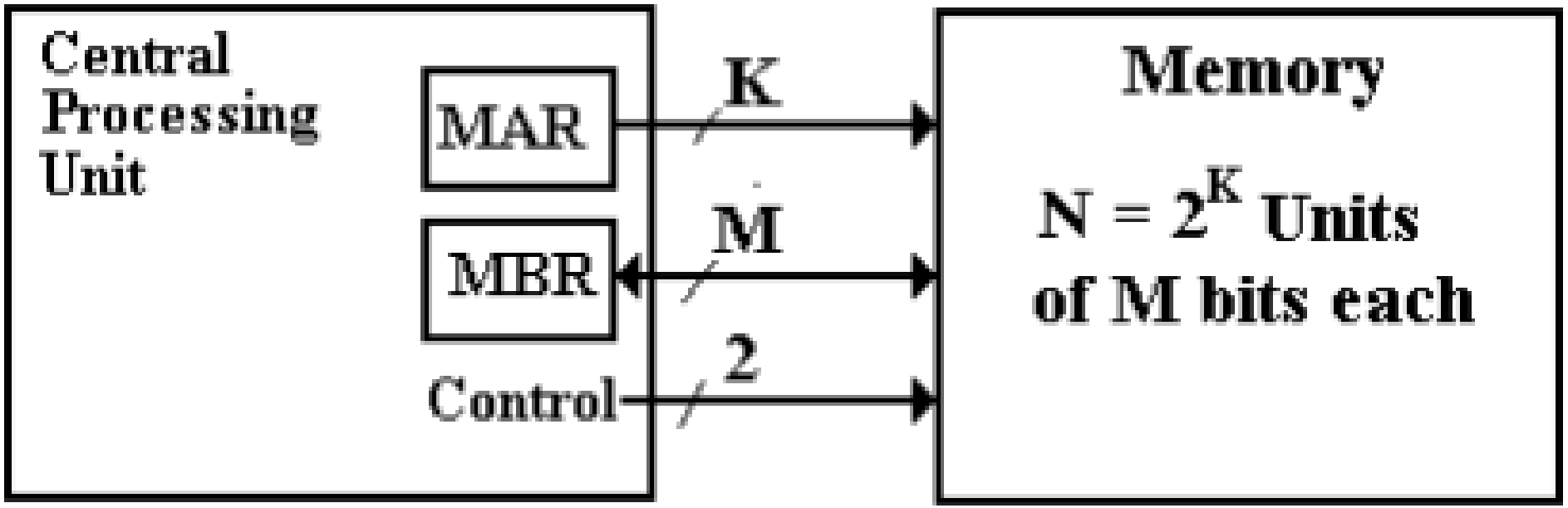
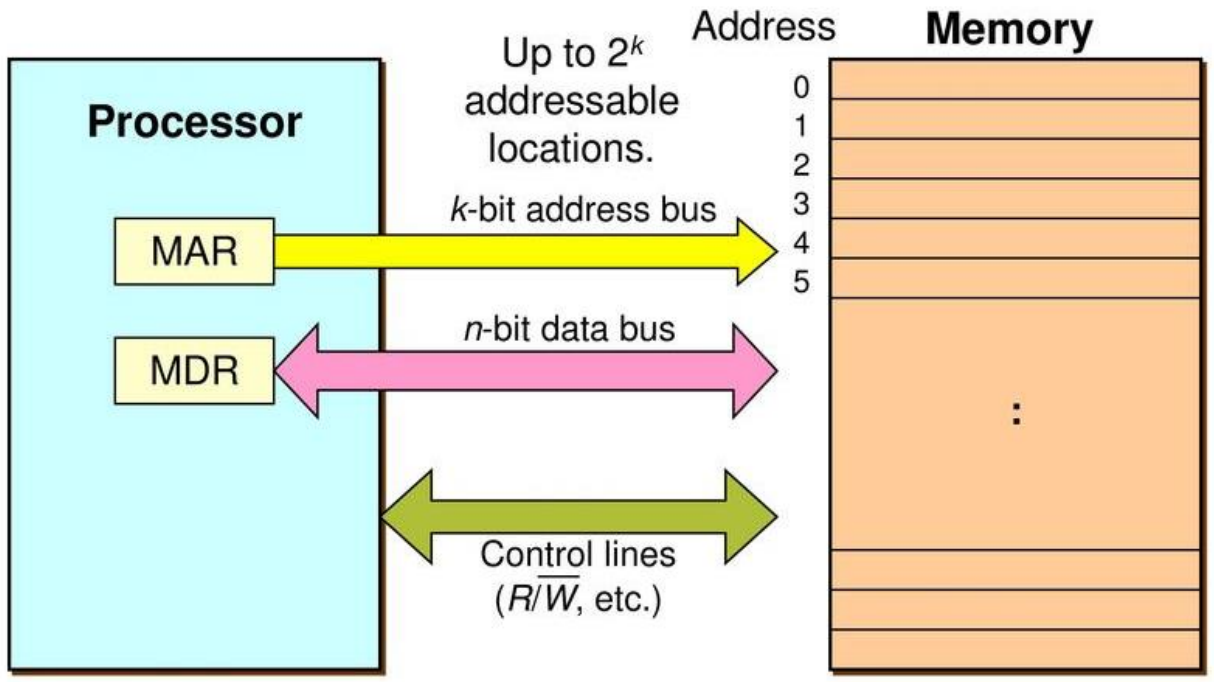


Figure: Monolithic View of Computer Memory

Memory Connections with CPU

Von Neumann Architecture



Read and Write Operations from CPU

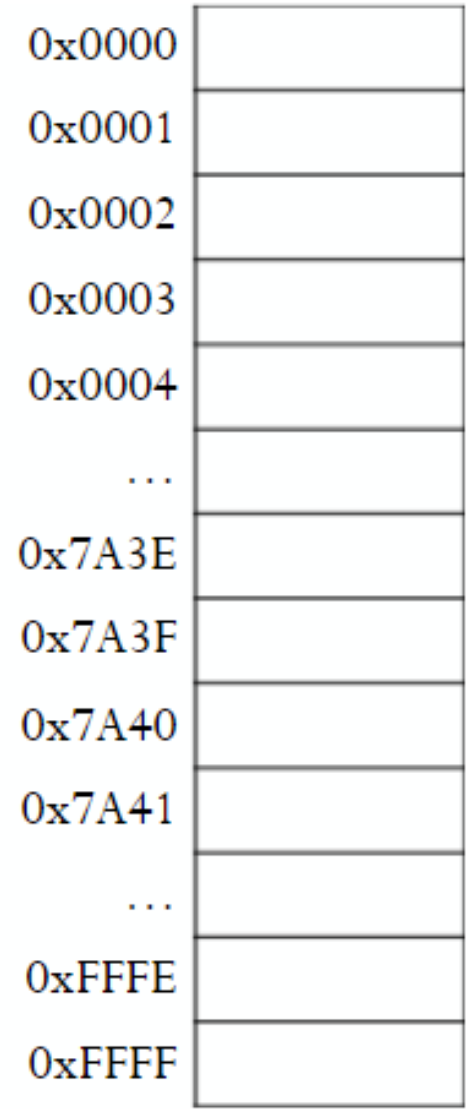
Read and Write Operations –

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

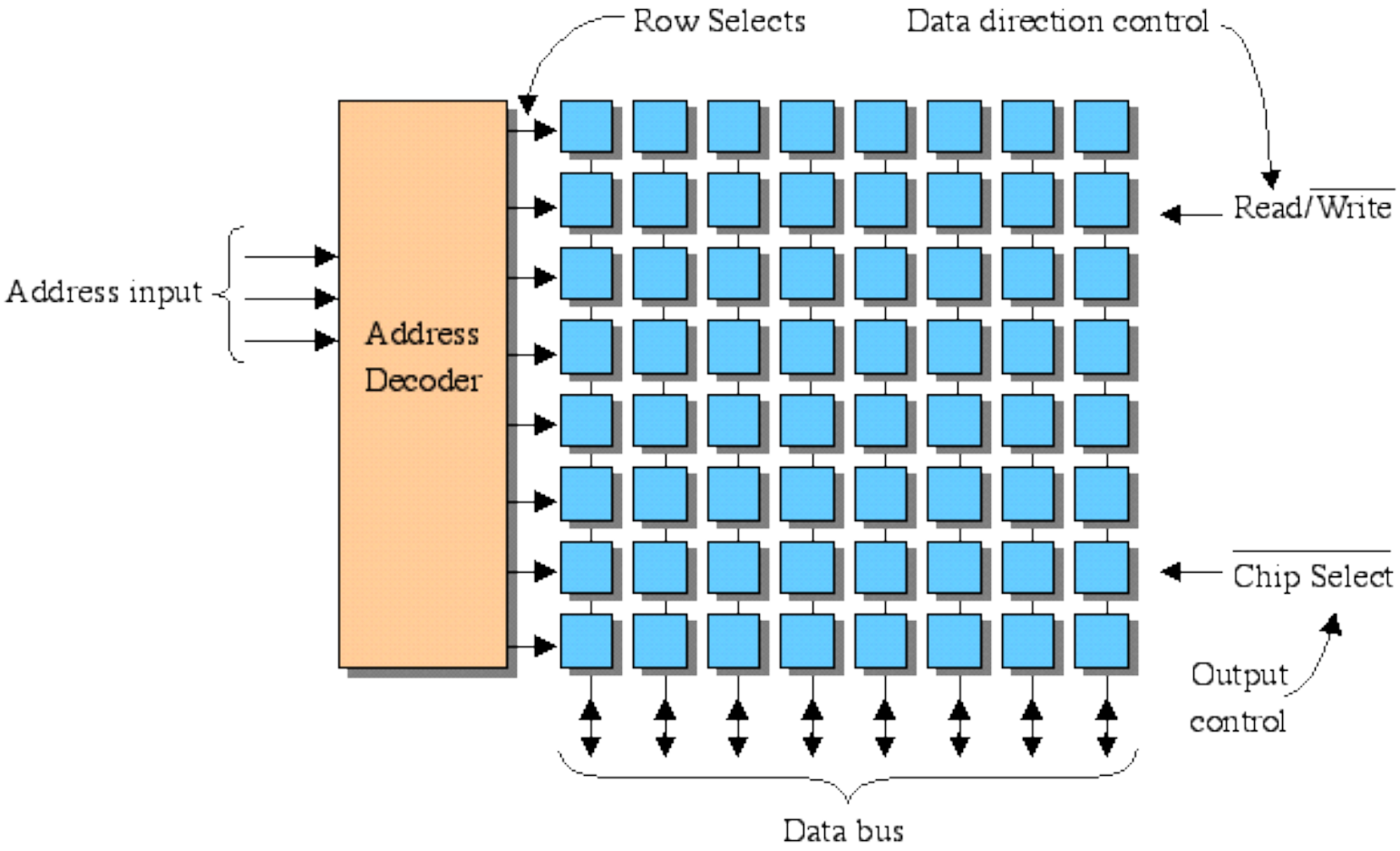
- **Memory** can be thought of as an array of data cells.
- The index into the array is the **address** .
  - Identical to the address stored in a pointer variable (using '&').
- How big are the data cells?
  - *byte-addressable* each byte has its own address
  - *word-addressable* each word has its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



Decoding Memory Addresses

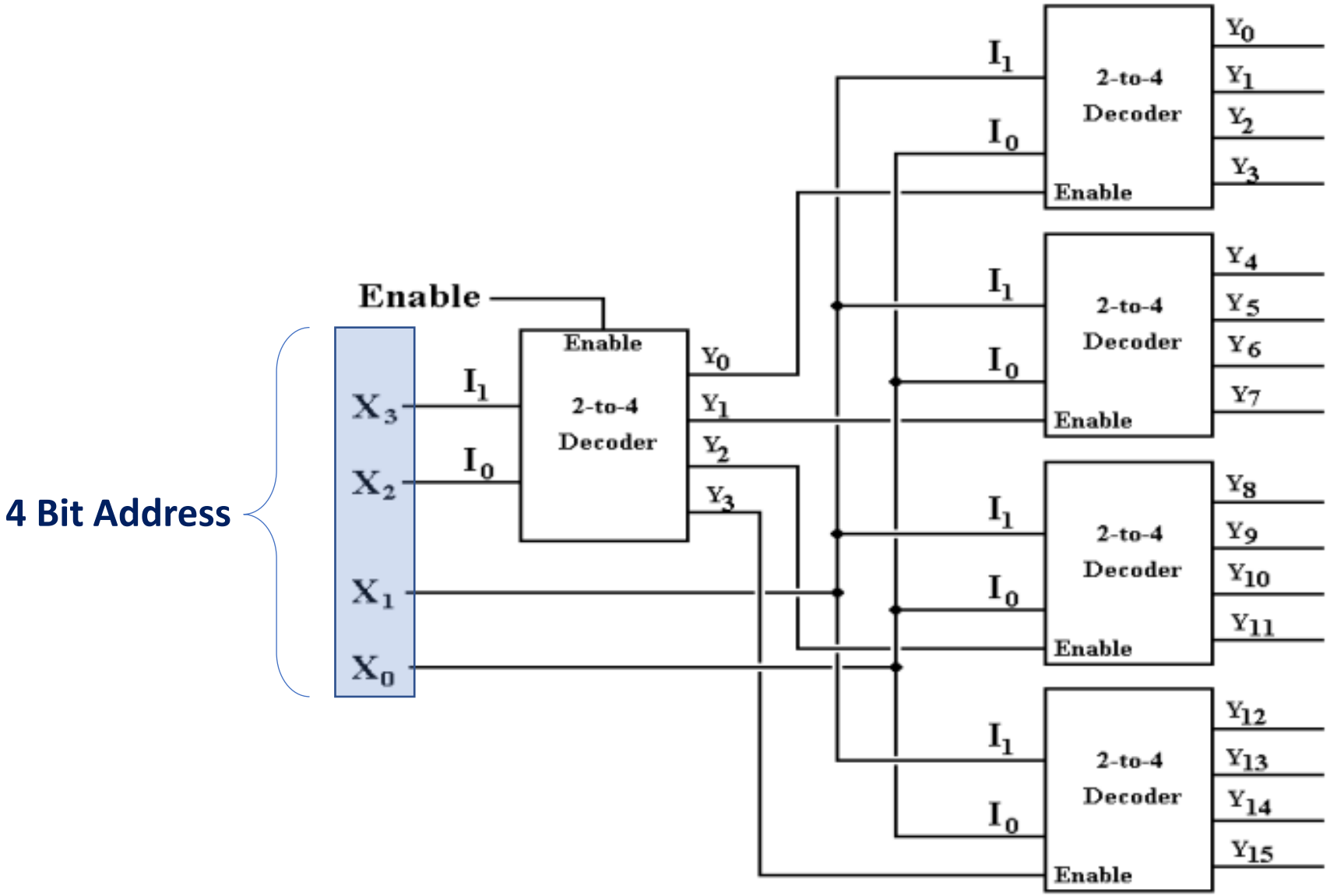
- A decoder is used to determine which cell is accessed:
  - Converts an address into the enable lines for the memory cell.
  - Makes sure that only one memory cell is enabled at a time.
- Useful for the decoder to have an enable input that can enable and disable the entire memory.
  - When a memory 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

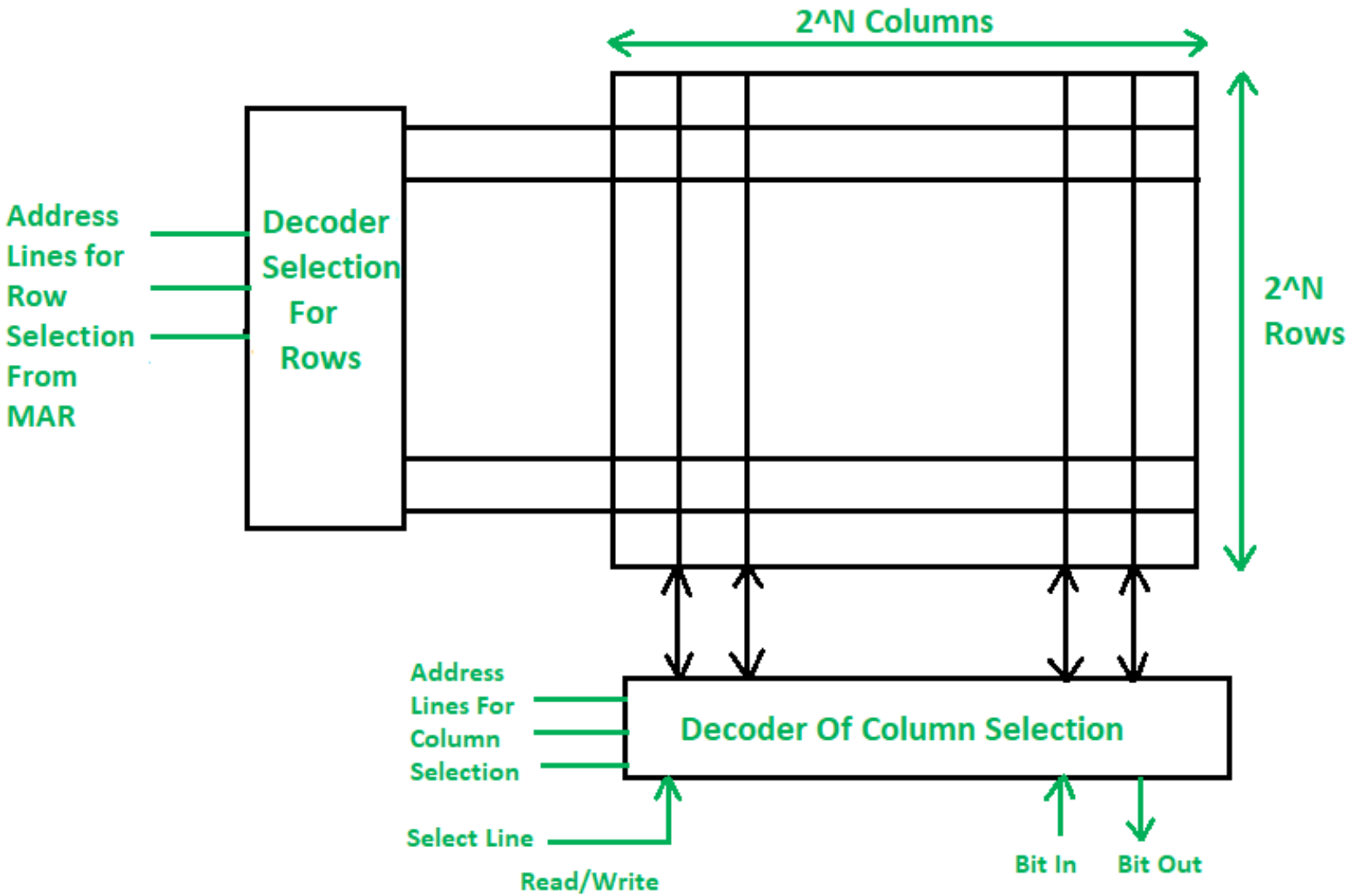


2D Array only selects entire ROW

Address Row Decoders - Example



2.5 D Memory Organization



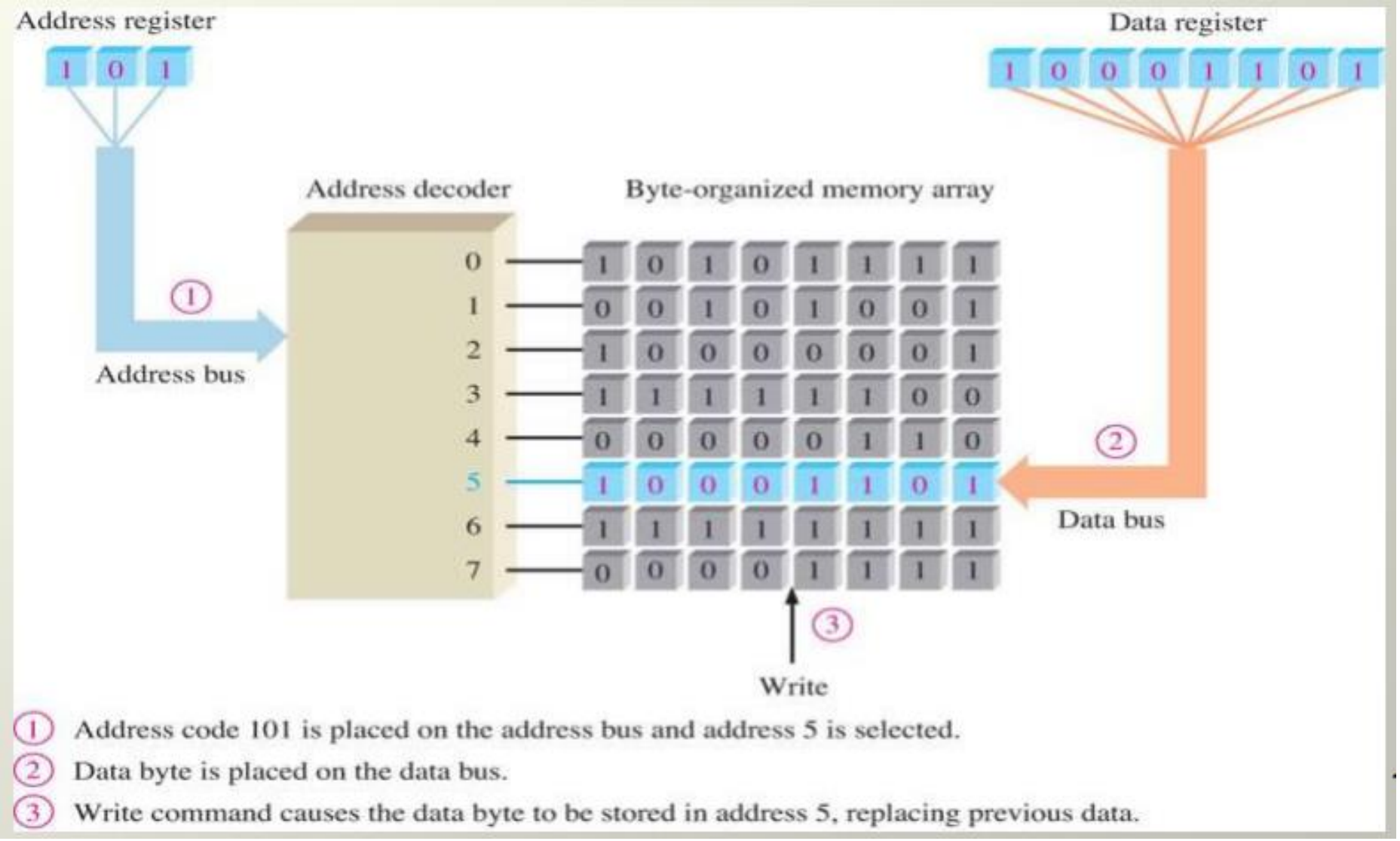
2.5D Memory Organization can Select ROWs as well as COLUMNs

Comparison between 2 D and 2.5 D Organization

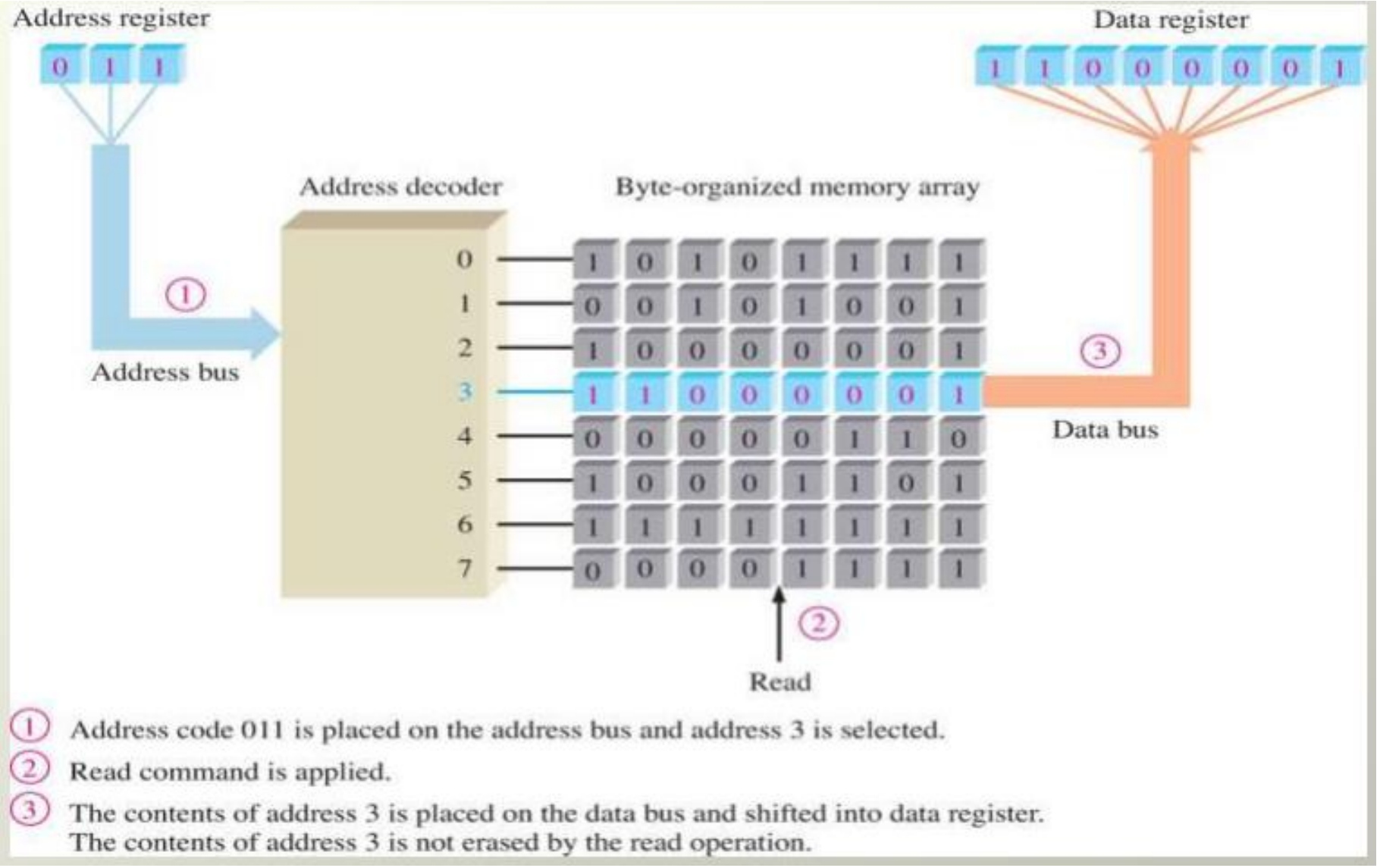
Comparison between 2D & 2.5D Organizations –

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



Memory Read Operation



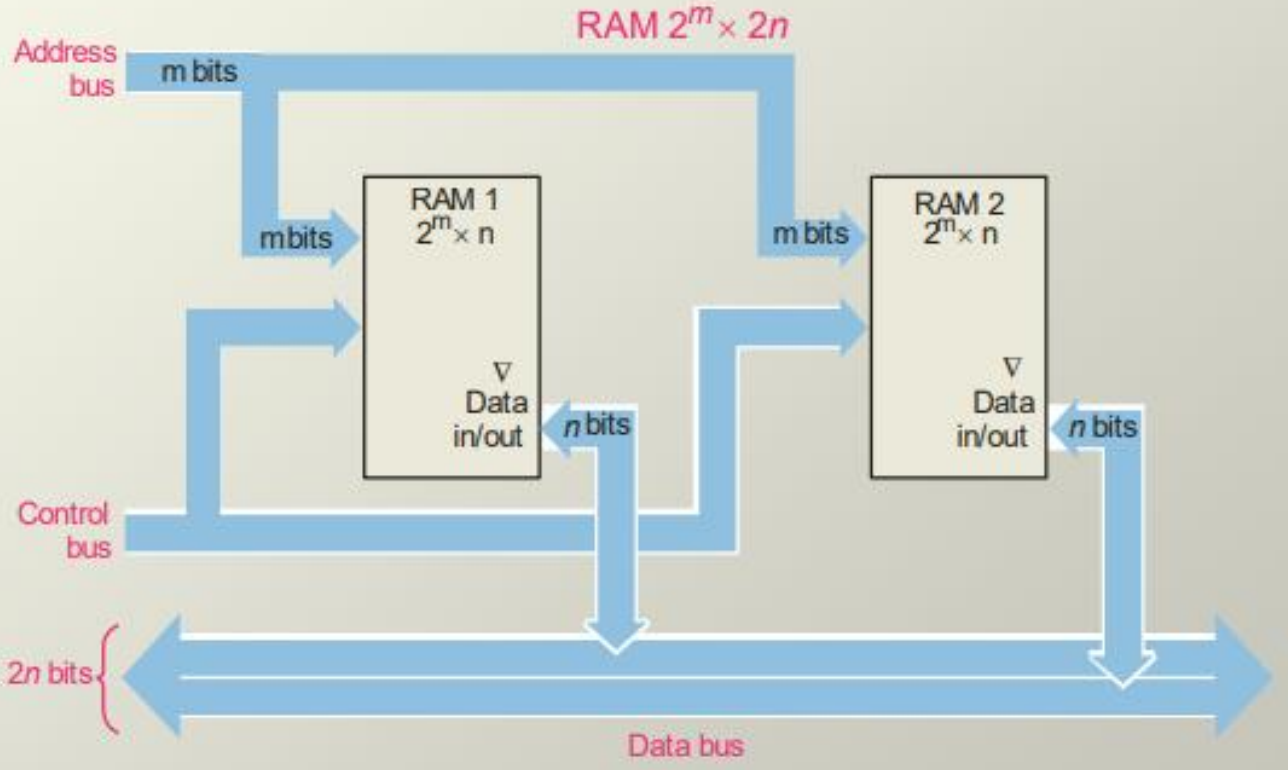
Expanding Memory Configuration through ADDRESS DECODING

Expanding Memory Configuration

Memory can be expanded in either word size or word capacity or both.

To expand word size:

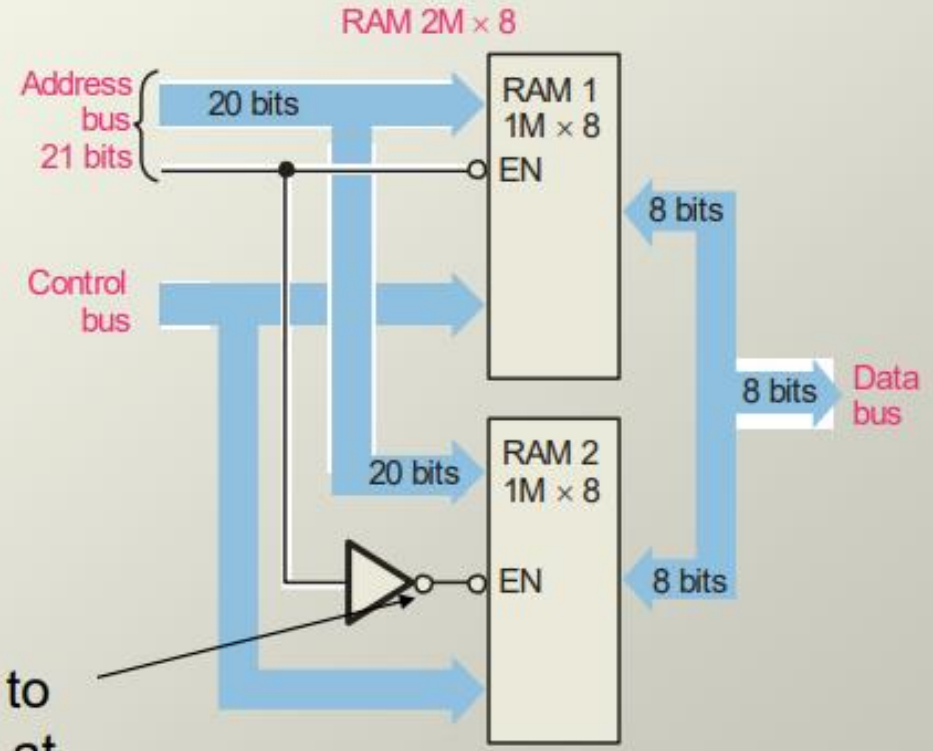
Notice that the data bus size is larger, but the number of address is the same.



Expand Word Capacity

To expand word capacity, you need to add an address line as shown in this example

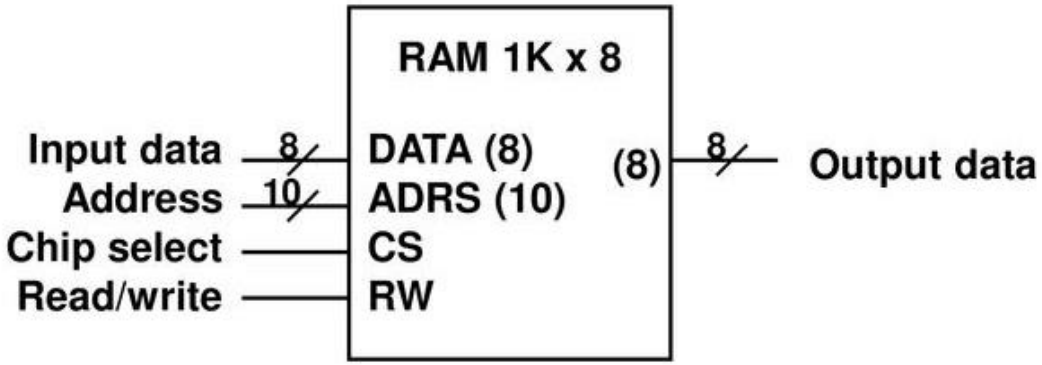
Notice that the data bus size does not change.



the purpose of the inverter is to make one of the ICs enabled at any time depending on the logic on the added address line.

A Memory Array

- An array of RAM chips: memory chips are combined to form larger memory.
- A 1K × 8-bit RAM chip:

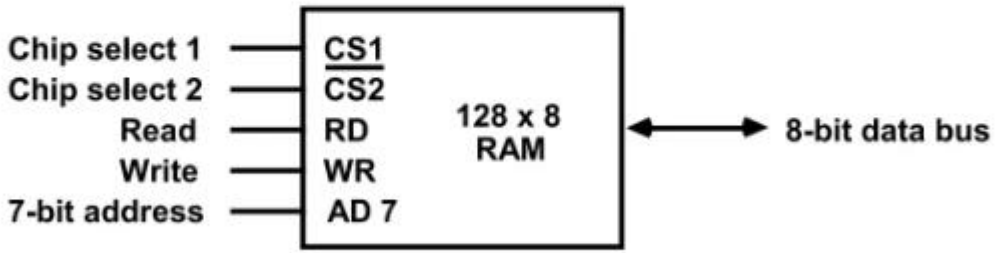


Block diagram of a 1K x 8 RAM chip

Example 3a

RAM and ROM Chips

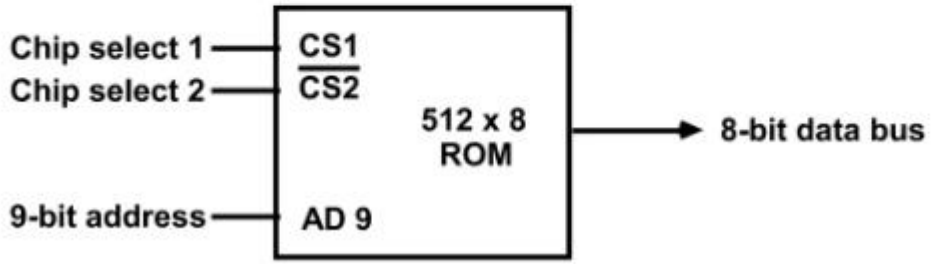
Typical RAM chip



Given types of RAM and ROM

CS1	$\overline{CS2}$	RD	WR	Memory function	State of data bus
0	0	x	x	Inhibit	High-impedence
0	1	x	x	Inhibit	High-impedence
1	0	0	0	Inhibit	High-impedence
1	0	0	1	Write	Input data to RAM
1	0	1	x	Read	Output data from RAM
1	1	x	x	Inhibit	High-impedence

Typical ROM chip



Example 3b

MEMORY ADDRESS MAP

Address space assignment to each memory chip

Example: 512 bytes RAM and 512 bytes ROM

Component	Hexa address	Address bus									
		10	9	8	7	6	5	4	3	2	1
RAM 1	0000 - 007F	0	0	0	x	x	x	x	x	x	x
RAM 2	0080 - 00FF	0	0	1	x	x	x	x	x	x	x
RAM 3	0100 - 017F	0	1	0	x	x	x	x	x	x	x
RAM 4	0180 - 01FF	0	1	1	x	x	x	x	x	x	x
ROM	0200 - 03FF	1	x	x	x	x	x	x	x	x	x

In Memory Map Space:

512 x 8 bits RAM is required

512 x 8 bits ROM is required

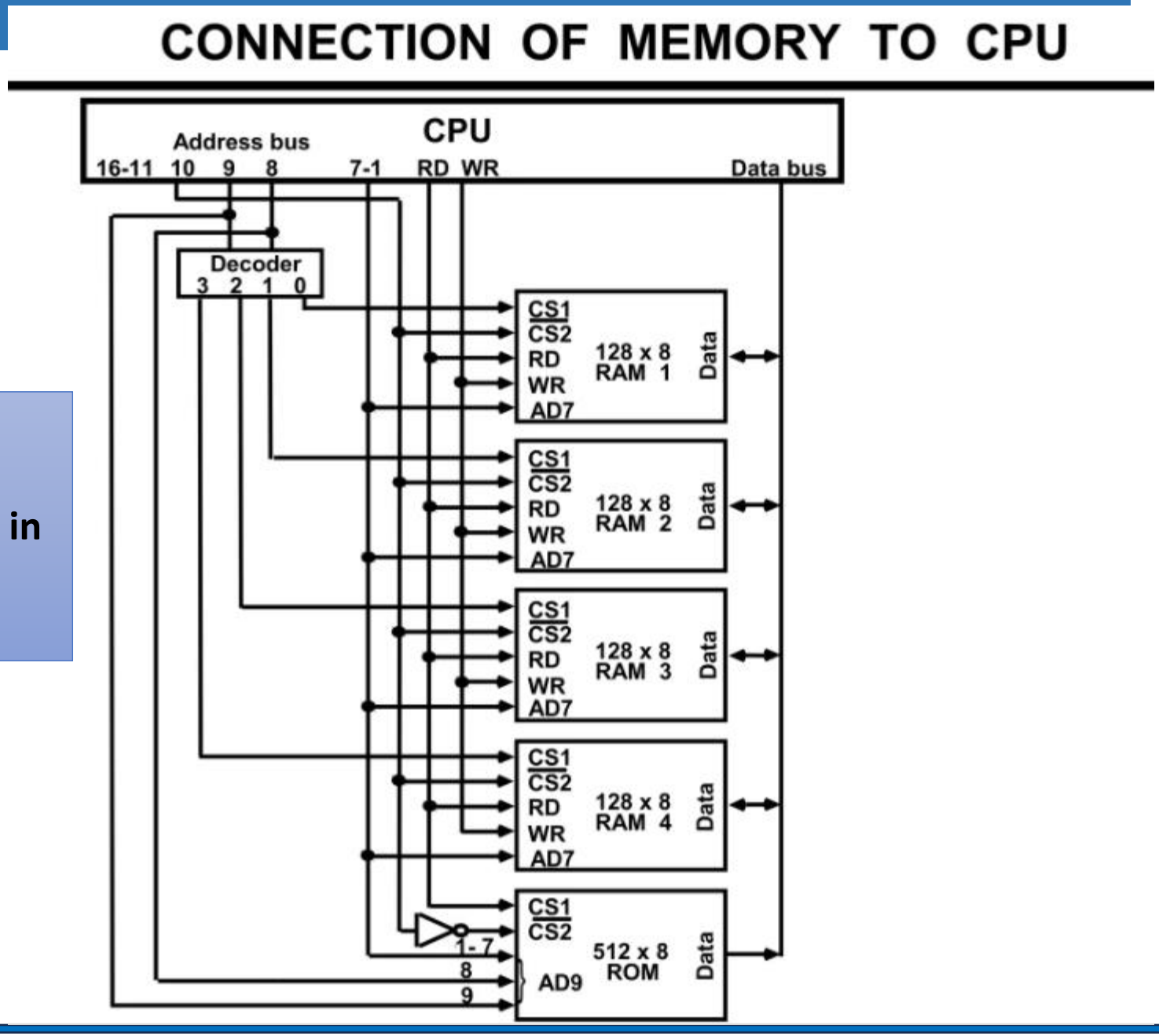
Memory Connection to CPU

- RAM and ROM chips are connected to a CPU through the data and address buses

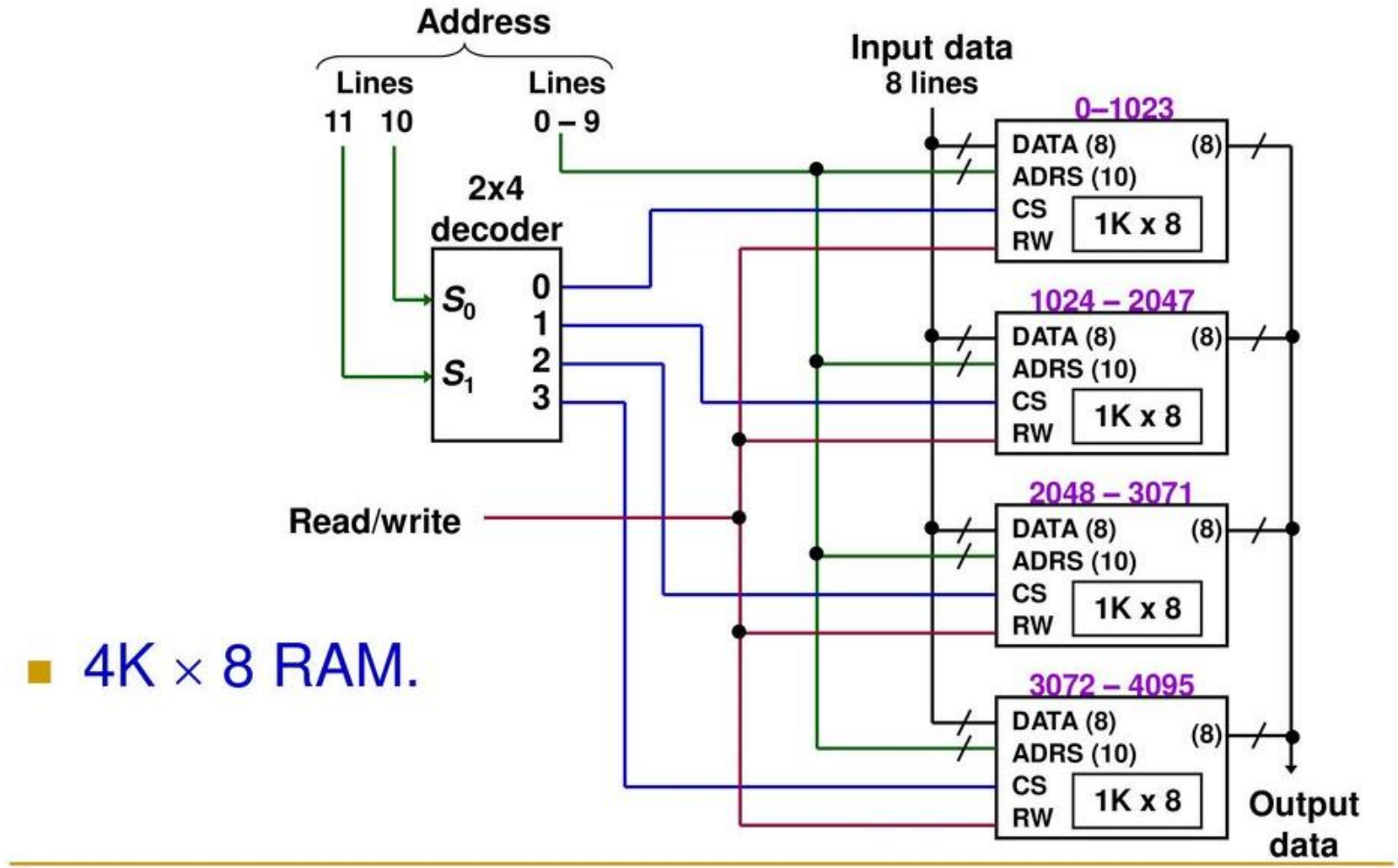
- The low-order lines in the address bus select the byte within the chips and other lines in the address bus select a particular chip through its chip select inputs

Example 3c

Look at How Address Bits are connected to select the right memory module for any address in the Range.



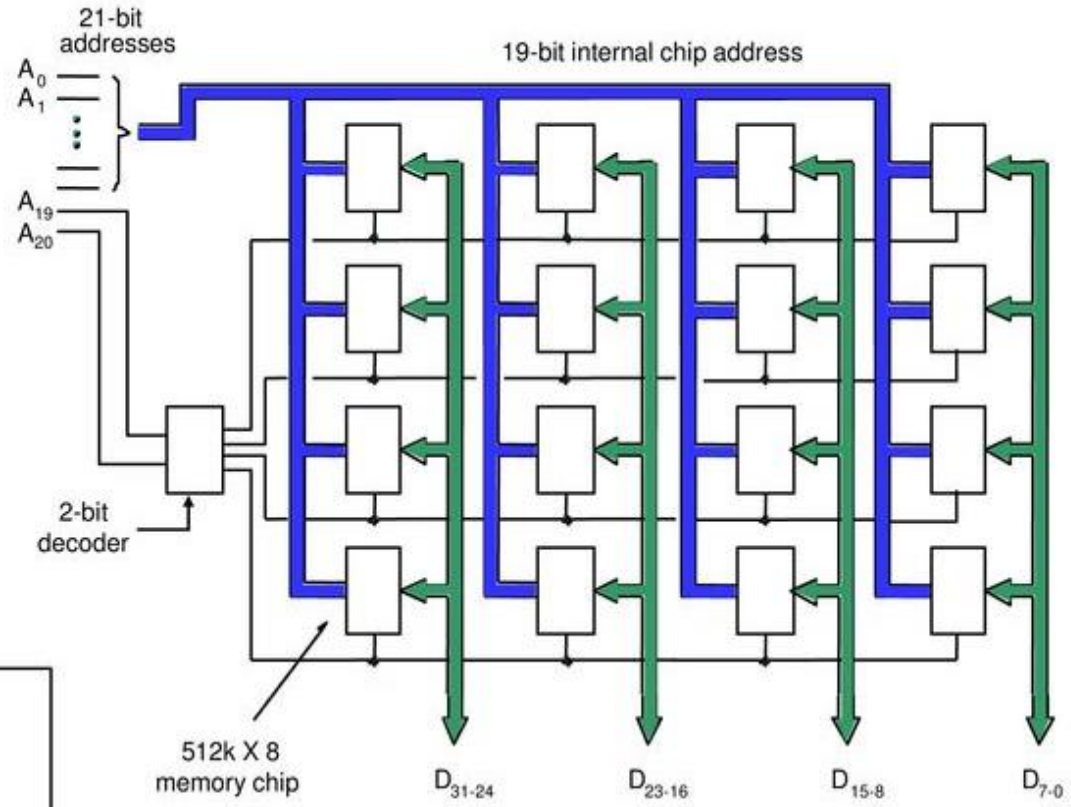
Memory Array Expansion with Decoder

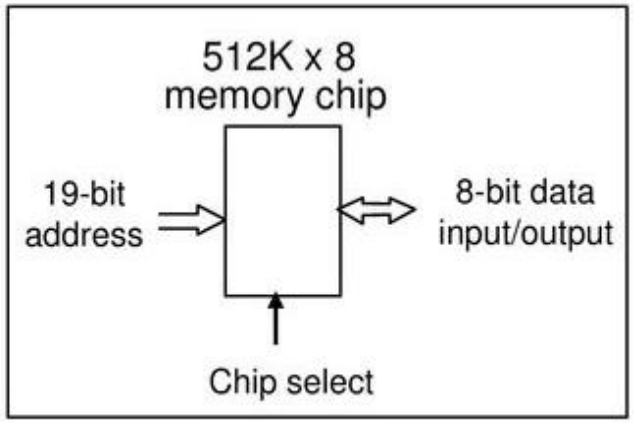


- 4K × 8 RAM.

Complex Memory Arrays Expansion in 2 D

See how Address Lines are Used to Select Relevant Module





- 2M × 32 memory module
  - Using 512K × 8 memory chips.