

# CS / EE 320 Computer Organization and Assembly Language Spring 2025 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
- QUIZ 4 TODAY

# Ideal Memory Requirement



- We want our memory to be big and fast
  - ISA promises big: 2<sup>32</sup> 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,		Capacity		Throughput	Cost		
	Disk	3ТВ	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		
but small	SRAM	32kB	1.3 ns	47,104 MB/s			

# Advantages of Memory Hierarchy



#### Very fast

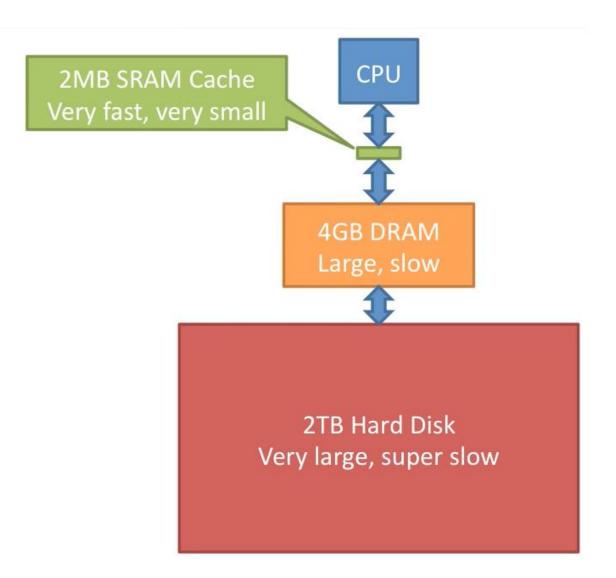
 If we have the right data in the right place

#### Very large

But possibly very slow

#### Reasonably cheap

- Lots of the cheap stuff
- A little of the expensive stuff



# 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	Not possible	Masks			
Programmable ROM (PROM)	memory	Not possible				
Erasable PROM (EPROM)		UV light, chip-level		Nonvolatile		
Electrically Erasable PROM (EEPROM)	Read-mostly memory	Electrically, byte-level	Electrically			
Flash memory		Electrically, block-level				

# Programmable ROM EPROM







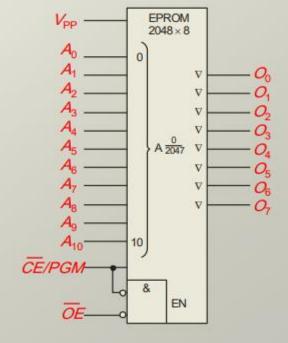


PROMs are programmable ROM, in which a fused link is burned open during the programming process. Once the PROM is programmed, it cannot be reversed.



An EPROM is an erasable PROM and can be erased by exposure to UV light through a window. To program it, a high voltage is applied to  $V_{PP}$  and OE is brought LOW.

Another type of erasable PROM is the EEPROM, which can be erased and programmed with electrical pulses.

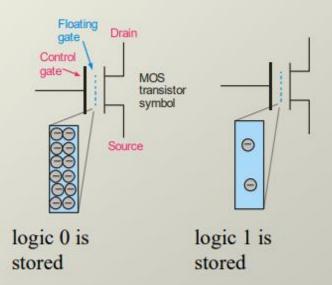


# Flash Memory

#### **5. The Flash Memory**

Flash memories are high density read/write memories that are nonvolatile. They have the ability to retain charge for years with no applied power.

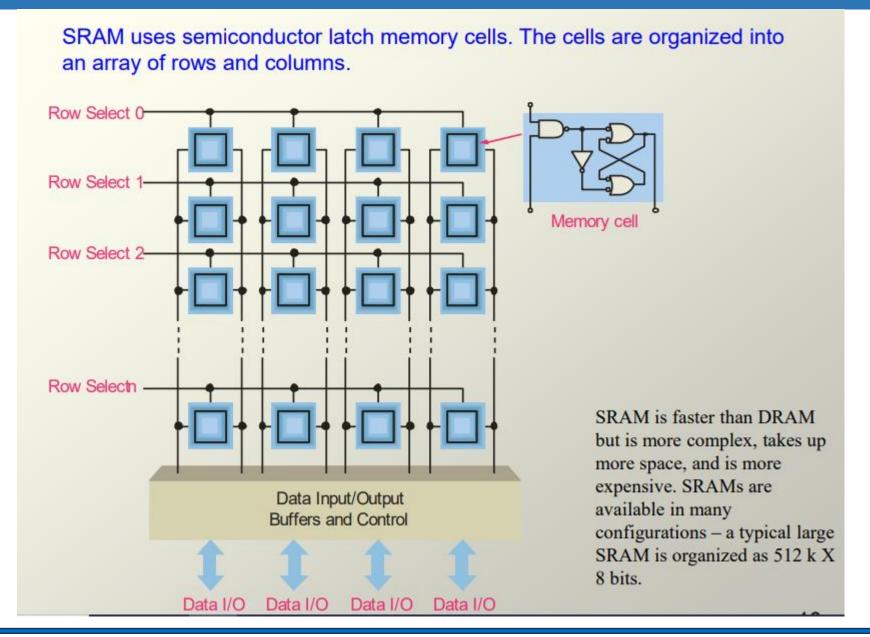
Flash memory uses a MOS transistor with a floating gate as the basic storage cell. The floating gate can store charge (logic 0) when a positive voltage is applied to the control gate. With little or no charge, the cell stores a logic 1.



The flash memory cell can be read by applying a positive voltage to the control gate. If the cell is storing a 1, the positive voltage is sufficient to turn on the transistor; if it is storing a 0, the transistor is off.

# Memory Cell Organization as Array







# Memory Ports and Connection to CPU

# Monolithic View of Computer Memory



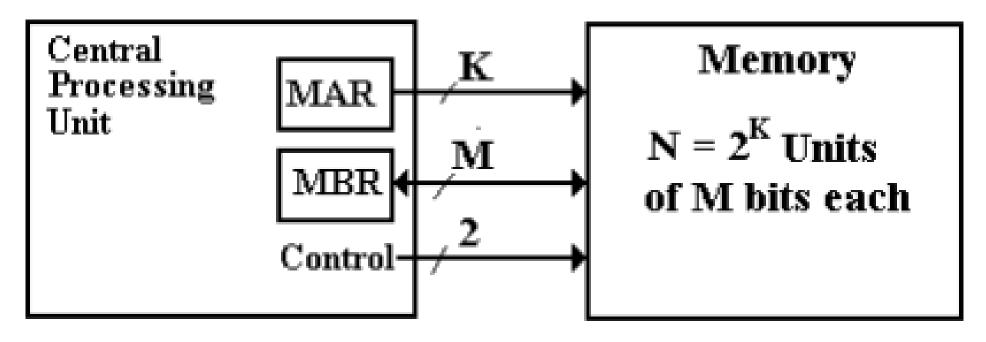


Figure: Monolithic View of Computer Memory

#### Read and Write Operations from CPU

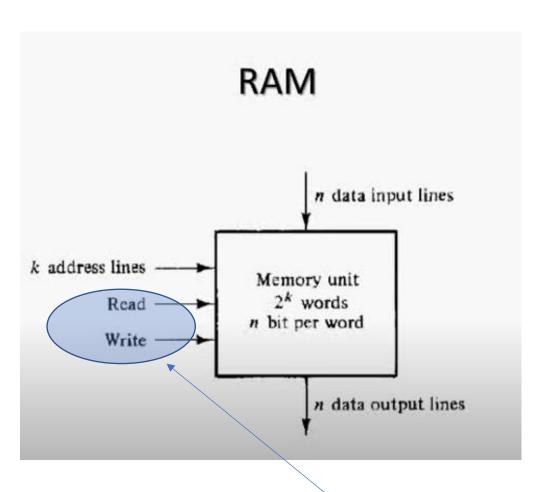


#### Read and Write Operations -

- 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.
- 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).
- With the help of the select line, we can select the desired data and we can perform read and write operations on it.

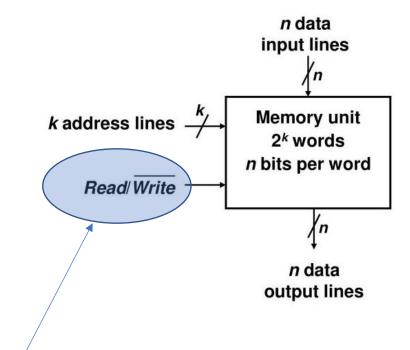
# Memory at Block Level with Read / Write





#### MEMORY UNIT

Block diagram of a memory unit:

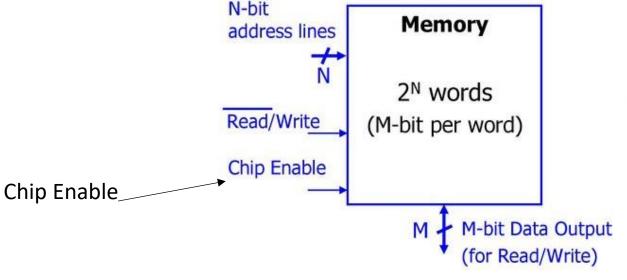


Two Ways of Read / Write signals

#### Block Diagram of Memory



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



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

#### RAM/ROM naming convention:

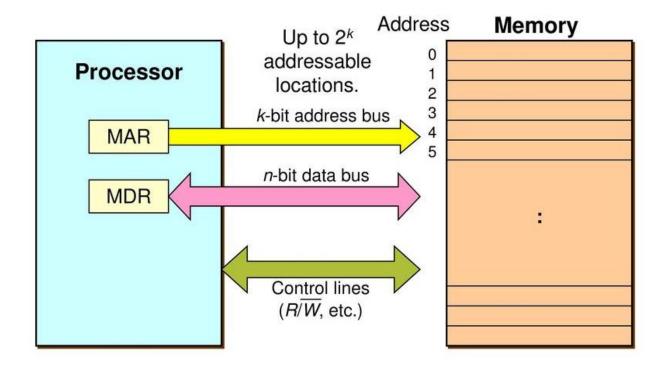
32 X 8, "32 by 8" => 32 8-bit words

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

# Memory Connections with CPU



#### **Von Neumann Architecture**





# 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-addressableach byte has its own address
  - word-addressableach 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	

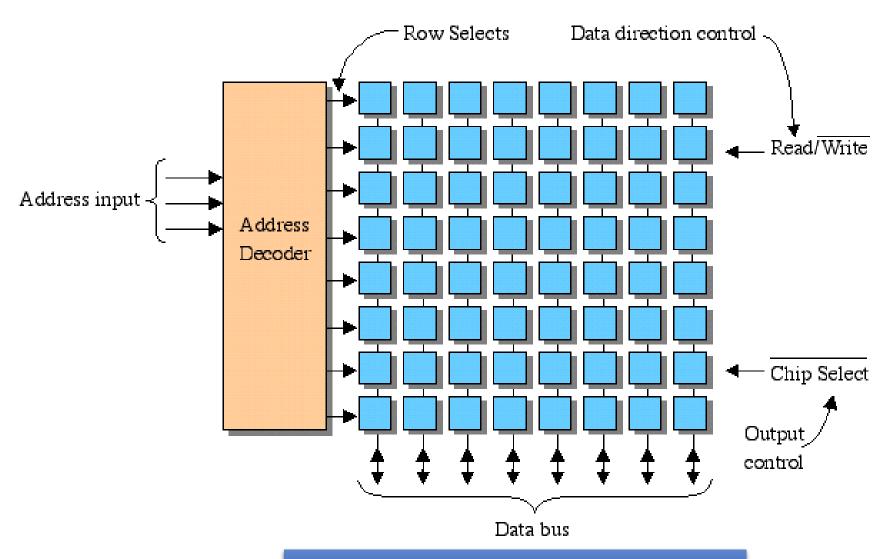
# **Decoding Memory Addresses**



- 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

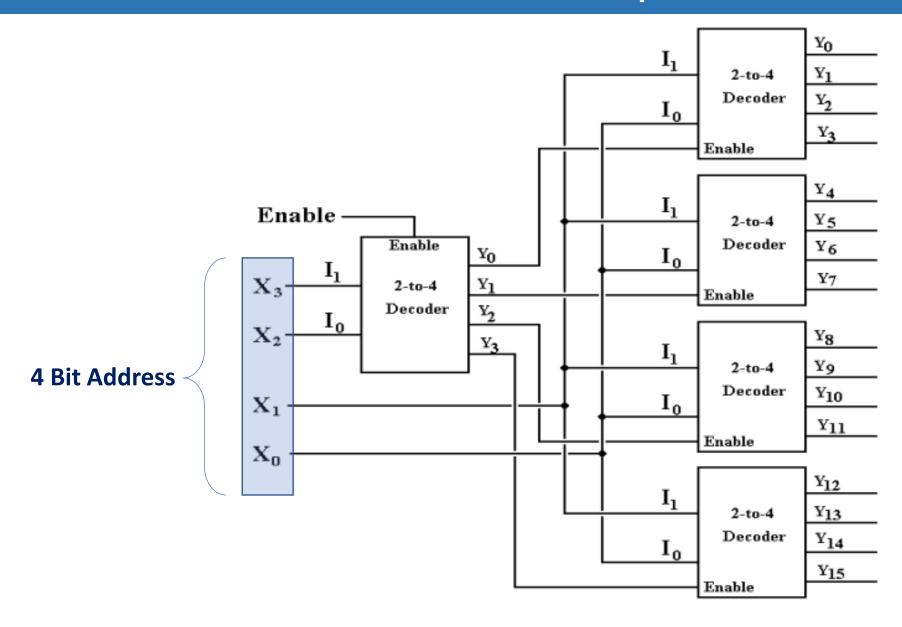




**2D Array only selects entire ROW** 

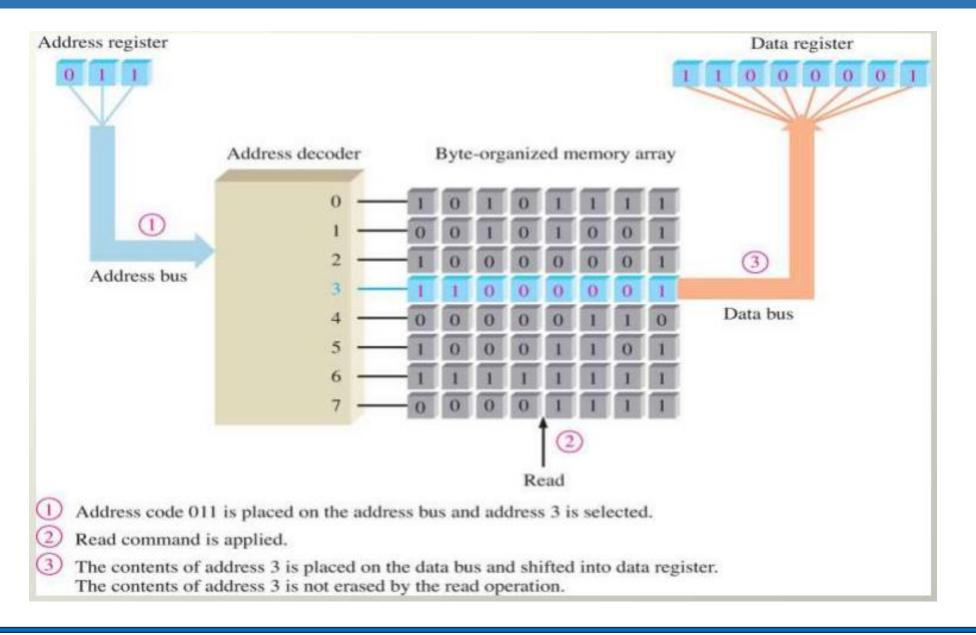
#### Address Row Decoders - Example





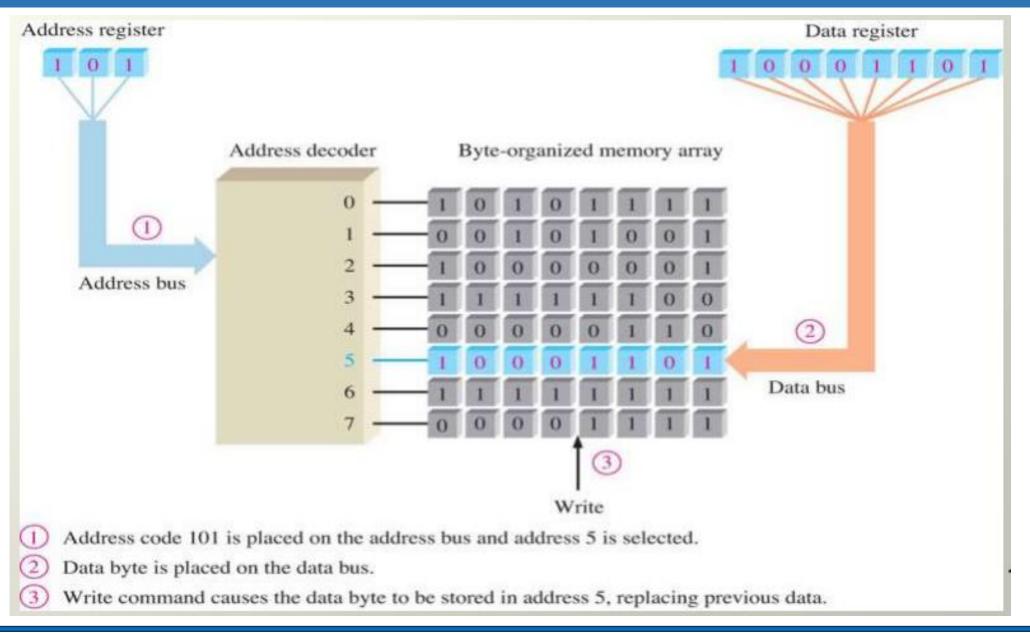
# Memory Read Operation





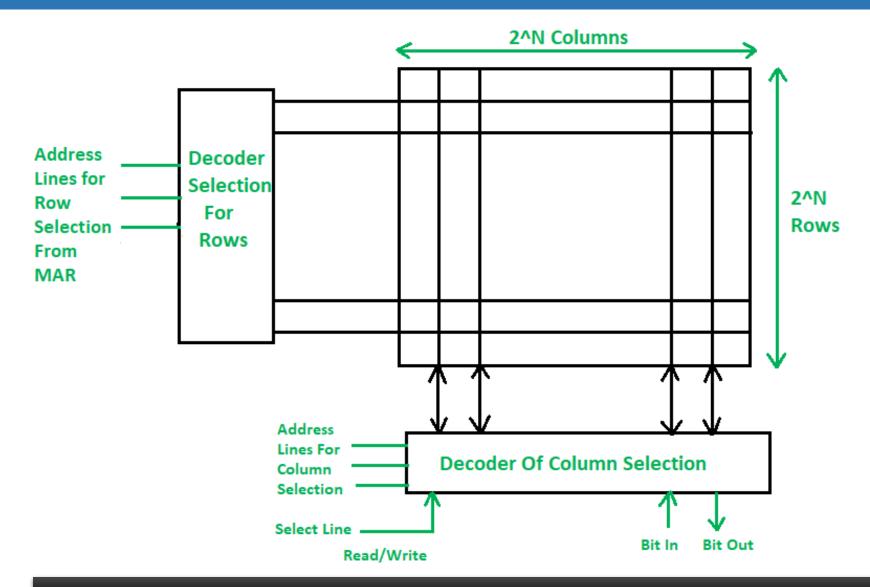
#### Memory Write Operation





# 2.5D 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 -

- 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.



# Expanding Memory Configuration through ADDRESS DECODING

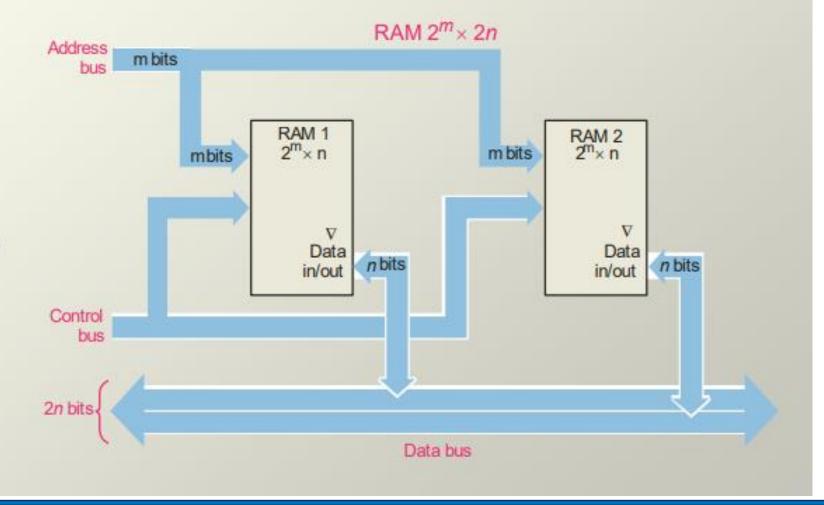
# **Expanding Memory Configuration**



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.



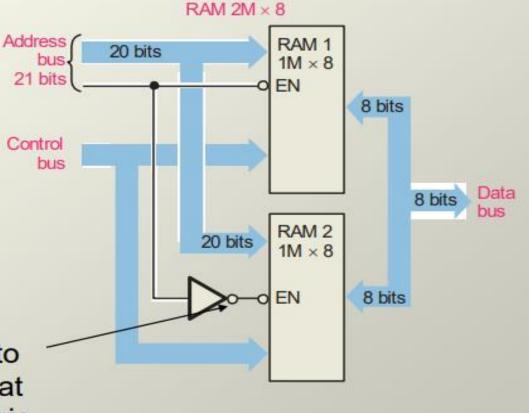
#### **Expand Word Capacity**



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.

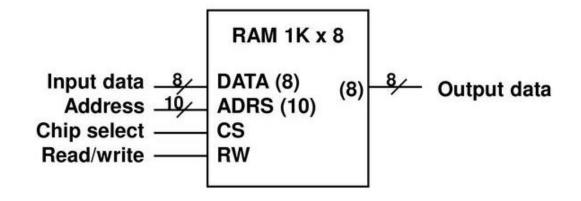
the purpose of the inverter is to make one of the ICs enabled at any time depending on the logic on the added address line.



#### A Memory Array



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

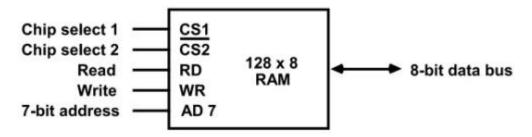


Block diagram of a 1K x 8 RAM chip

#### MAIN MEMORY



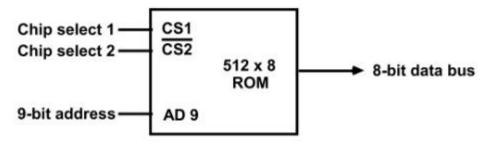
# RAM and ROM Chips Typical RAM chip



#### Given types of RAM and ROM

CS1	CS2	x x Inhibit High-in x x Inhibit High-in 0 0 Inhibit High-in 0 1 Write Input d	State of data bus		
0	0	х	х	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

15-3	Hexa	Address bus									
Component	address	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**

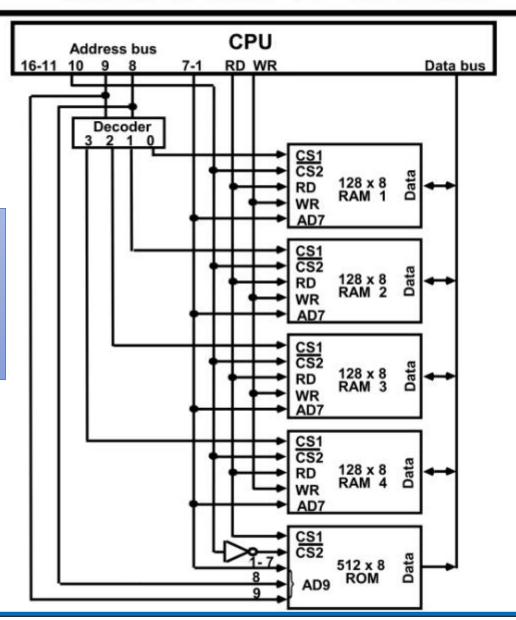
- RAM and ROM chips are connected to a CPU through the data and address buses
- The low-order lines in the address bus select the byte within the chips and other lines in the address bus select a particular chip through its chip select inputs

#### Example 3c



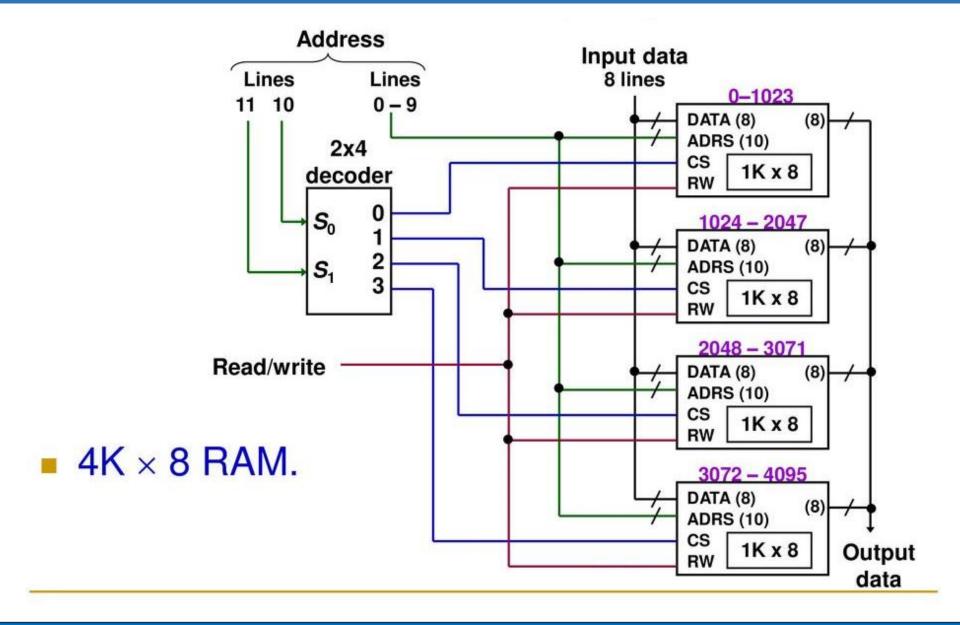
#### CONNECTION OF MEMORY TO CPU

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



# Memory Array Expansion with Decoder

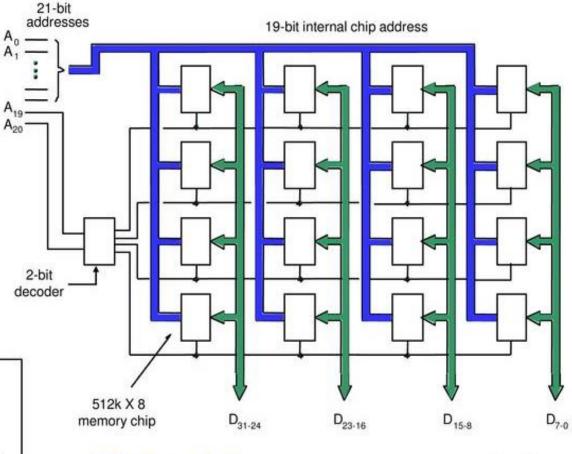




#### Complex Memory Arrays Expansion in 2 D



See how Address Lines are
Used to Select Relevant Module



- 512K x 8
  memory chip

  19-bit address

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