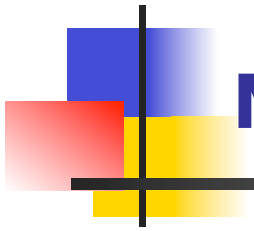
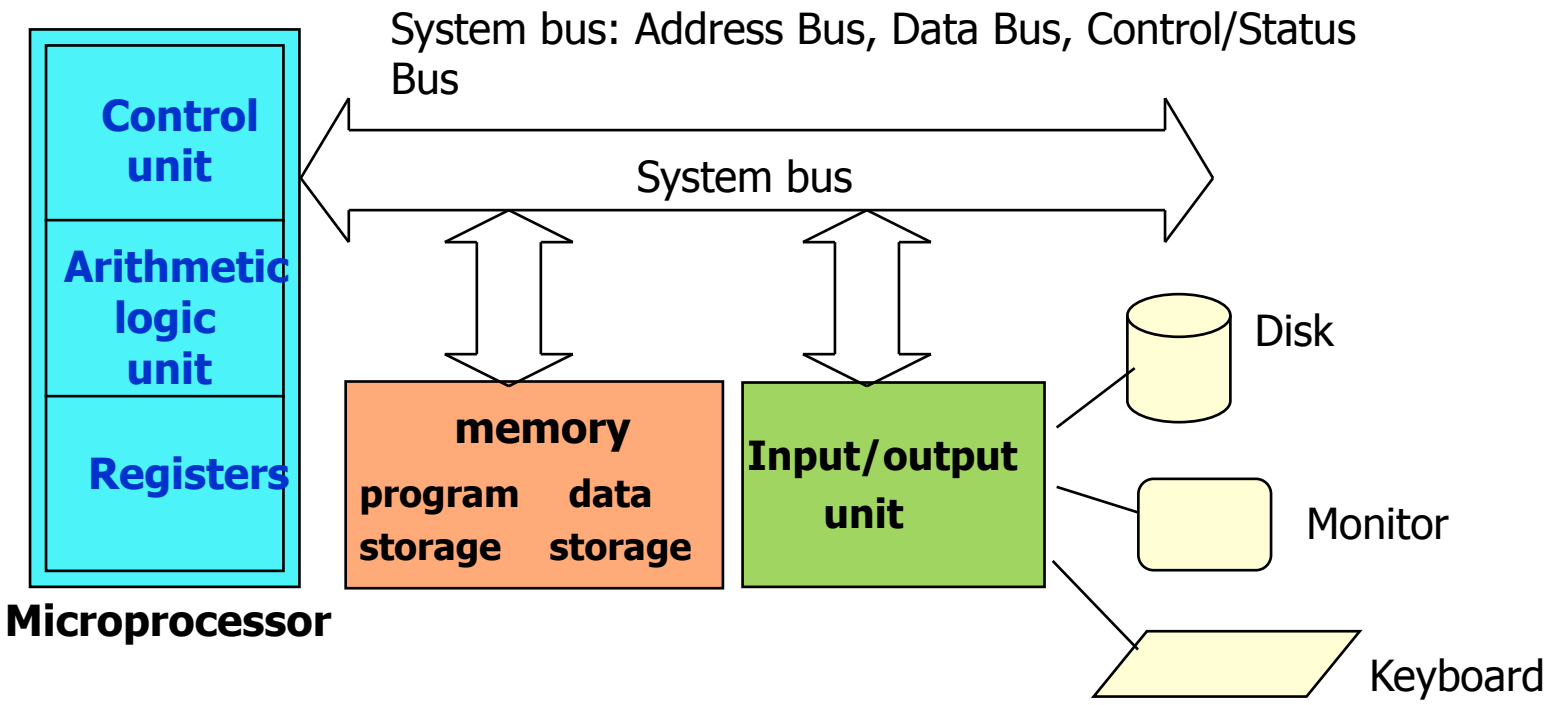


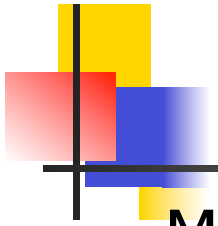
Microprocessor Systems

Fall 2024

MEMORY ORGANIZATION

Computer Organization





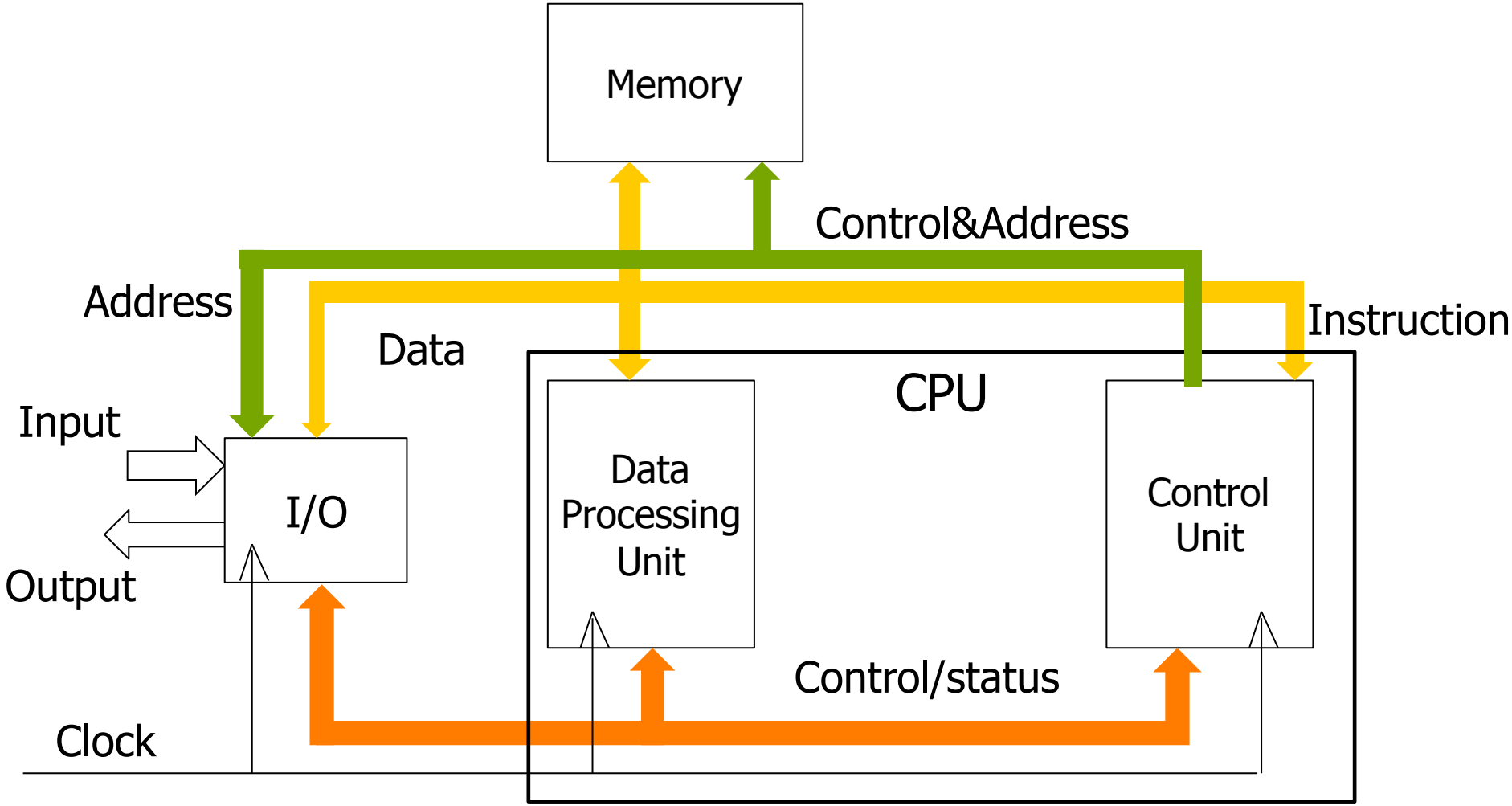
Memory sub-system

- Memory stores information such as instructions and data in binary format (0's and 1's). It provides this information to the microprocessor whenever it is needed.

- Usually, there is a memory "sub-system" in a microprocessor-based system. This sub-system includes:
  - The registers inside the microprocessor
  - Read Only Memory (ROM)
    - used to store information that does not change.
  - Random Access Memory (RAM) (also known as Read/ Write Memory)
    - used to store information supplied by the user such as programs and data.

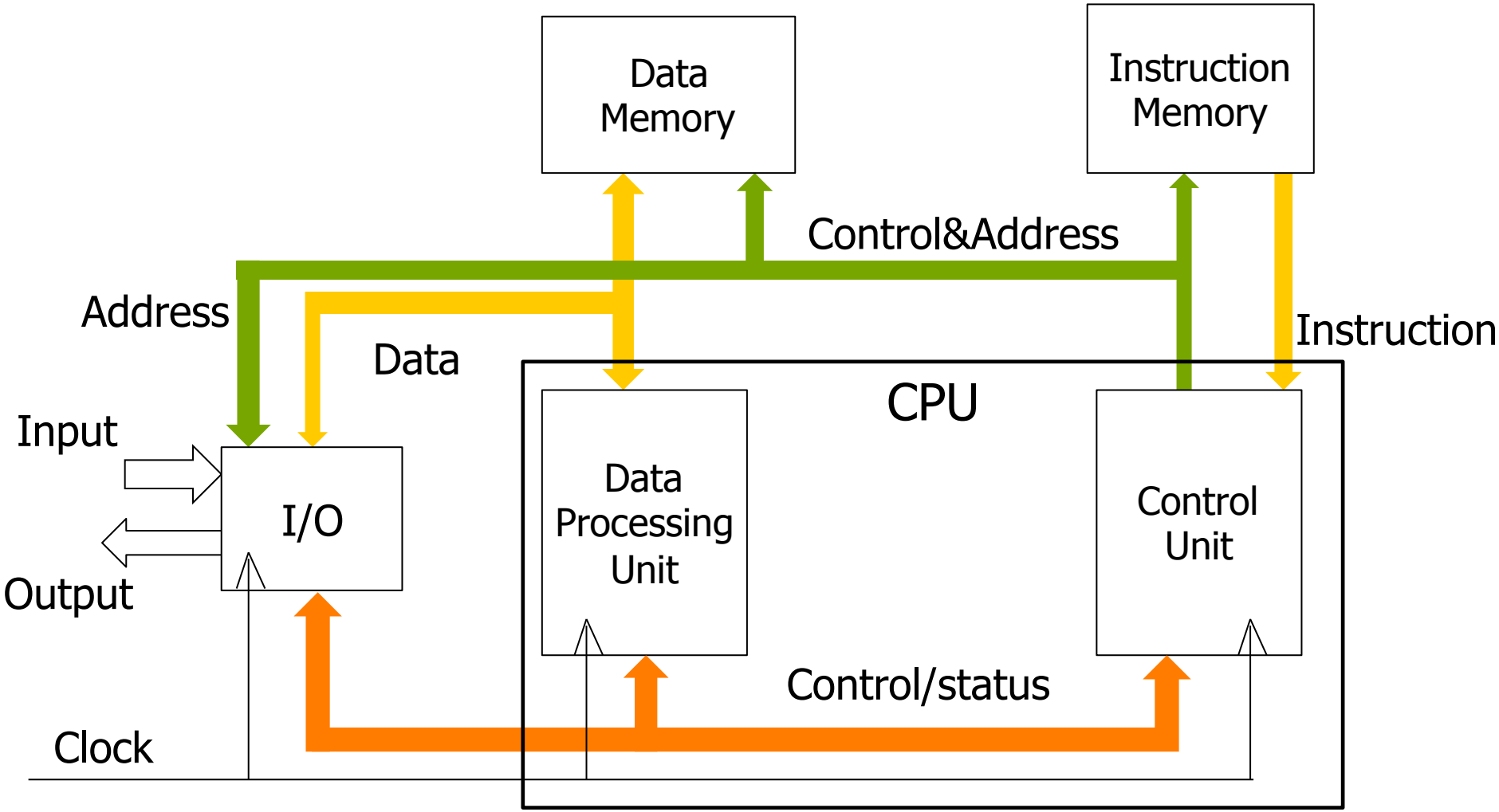
Von Neumann Architecture

- Instruction and data are in the same memory



Harvard Architecture

- There are two memories: Instruction and data



Memory

- The memory holds instruction code numbers and data numbers
  - Non-volatile memory
    - Read only memory (ROM)
    - Programmable read-only memory (PROM)
    - Erasable programmable ROM (EPROM)
  - Volatile memory
    - Static random-access memory (SRAM)
    - Dynamic random-access memory (DRAM)

Non-volatile Memory

- Read-only memory (ROM) can only be read but not written by the processor
  - Mask-programmed read-only memory (MROM): programmed when being manufactured
  - Programmable read-only memory (PROM): the memory chip can be programmed by the end user

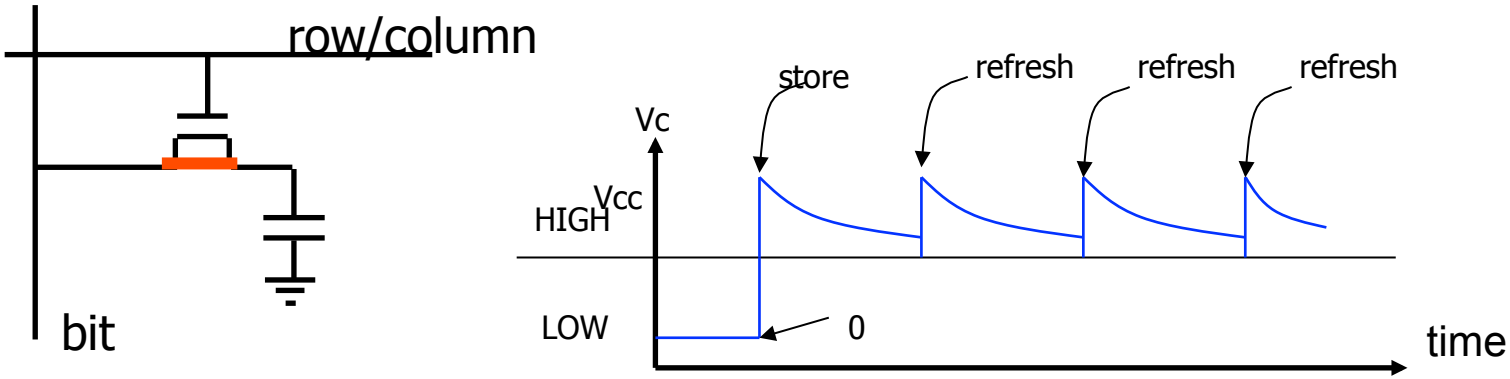
Non-volatile Memory

- Erasable programmable ROM (EPROM)
  - electrically programmable many times
  - erased by ultraviolet light (through a window)
  - erasable in bulk (whole chip in one erasure operation)
- Electrically erasable programmable ROM (EEPROM)
  - electrically programmable many times
  - electrically erasable many times
  - can be erased one location, one row, or whole chip in one operation
- Flash memory
  - electrically programmable many times
  - electrically erasable many times
  - can only be erased in bulk

Volatile Memory

DRAM: Dynamic Random Access Memory

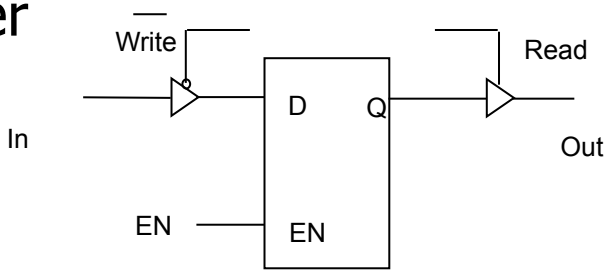
Fast, dense (only 1 transistor/bit), works like charge storage, refreshed periodically, read/write, cheap



SRAM: Static Random Access Memory

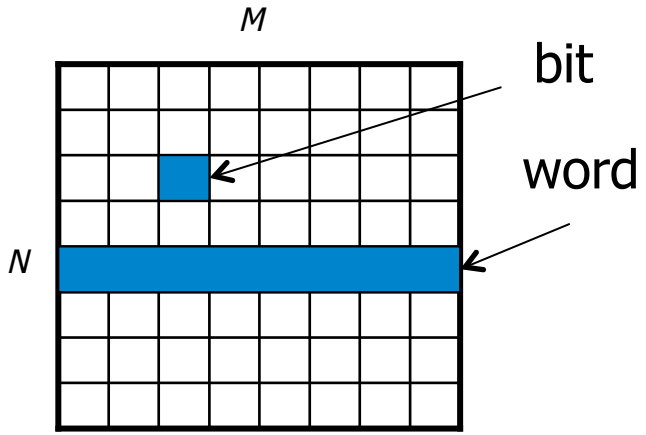
More power consumption than DRAM, each bit is stored in a flip-flop, more expensive, read/write, faster

Small amounts are often used in cache memory for high-speed memory access



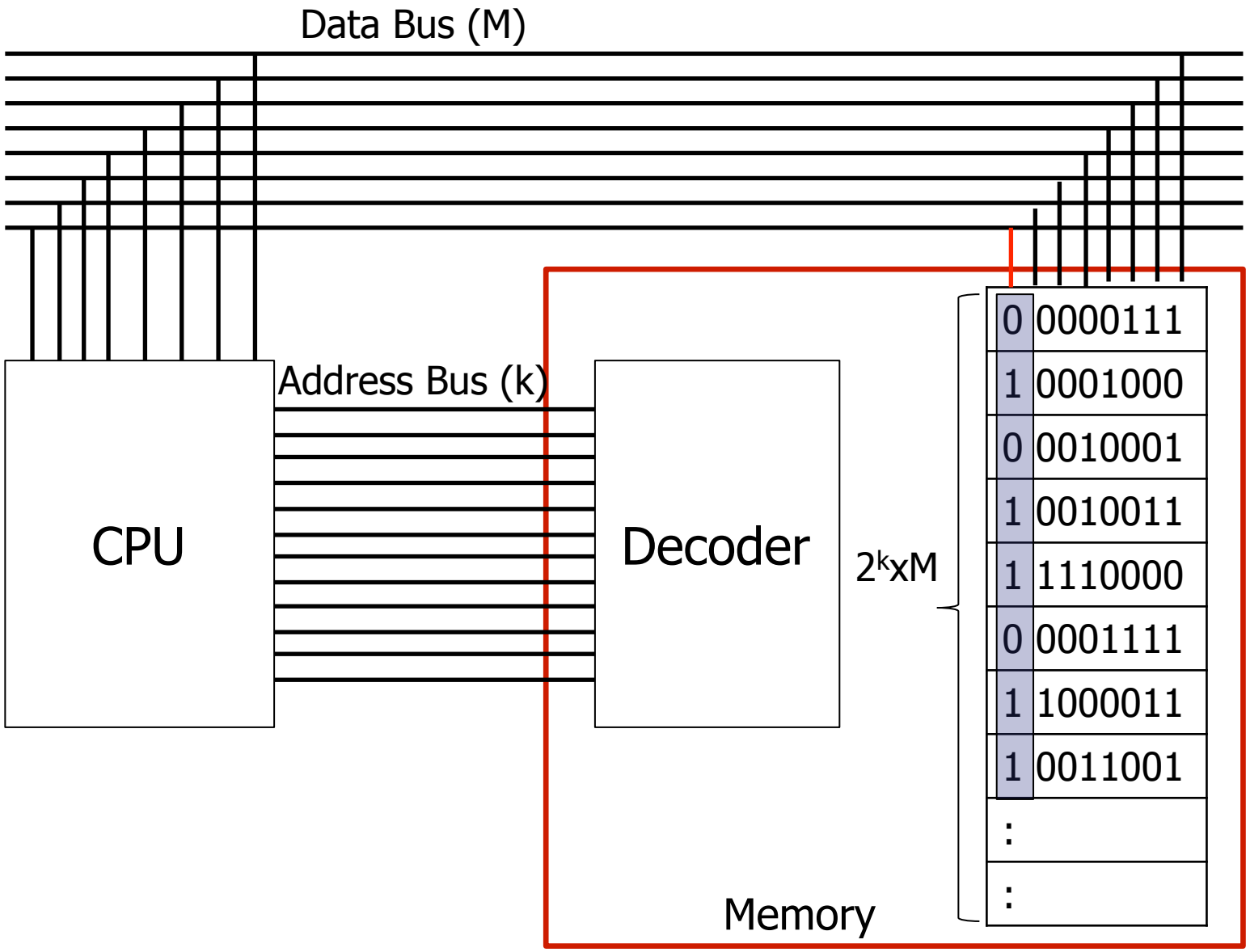
Memory Units

- Each entry in a memory unit is called a *word*
- Each word is composed of *M* bits (width)
- Size of a RAM is the number of words $N=2^k$
- A matrix of size *N* x*M*
  - *N*: number of rows (number of words)
  - *M*: number of columns (number of bits)

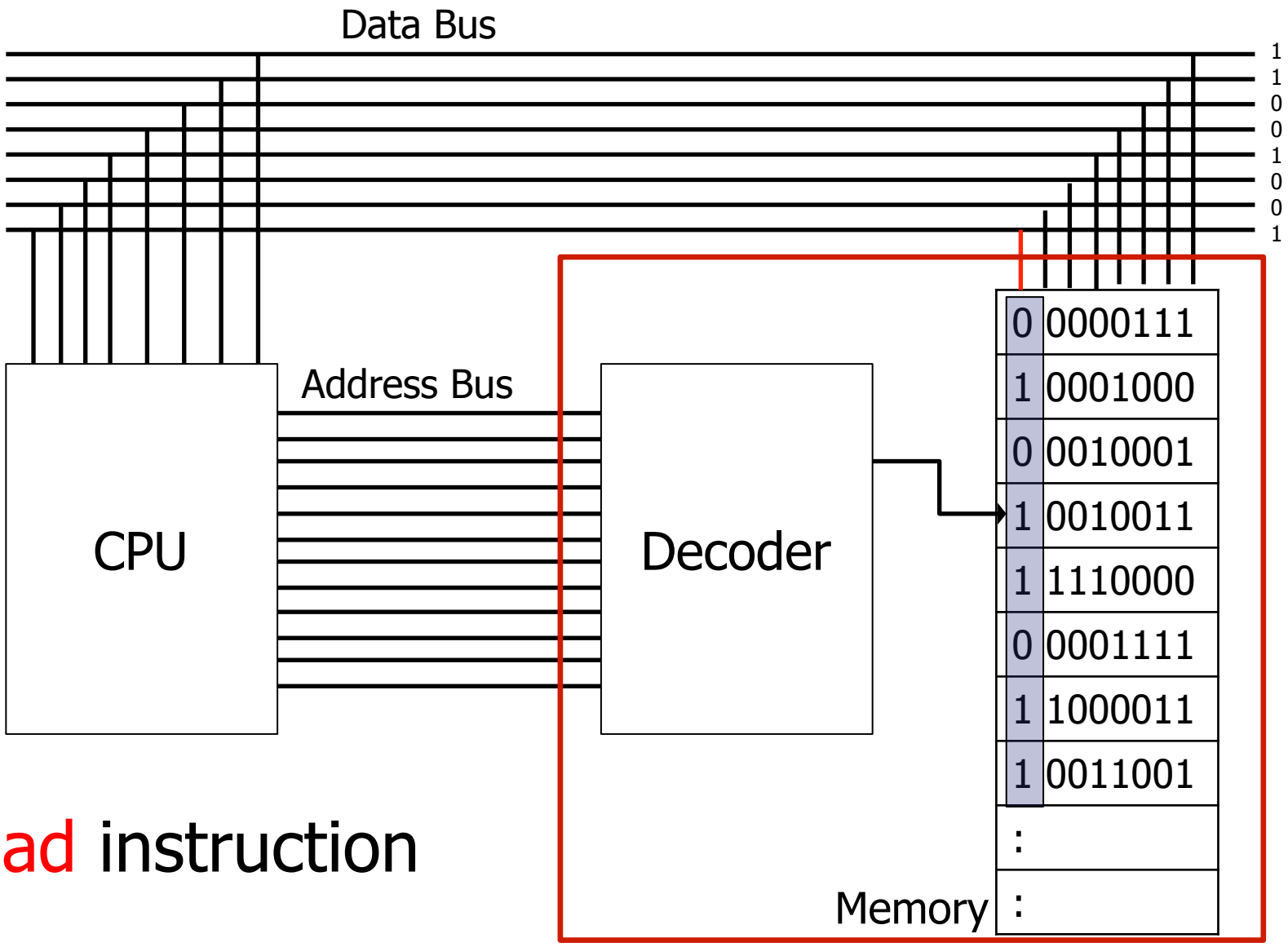


- Common widths: Byte (8 bits), Short (16 bits), Int (32 bits)

CPU-Memory Connection



CPU-Memory Connection



Read instruction

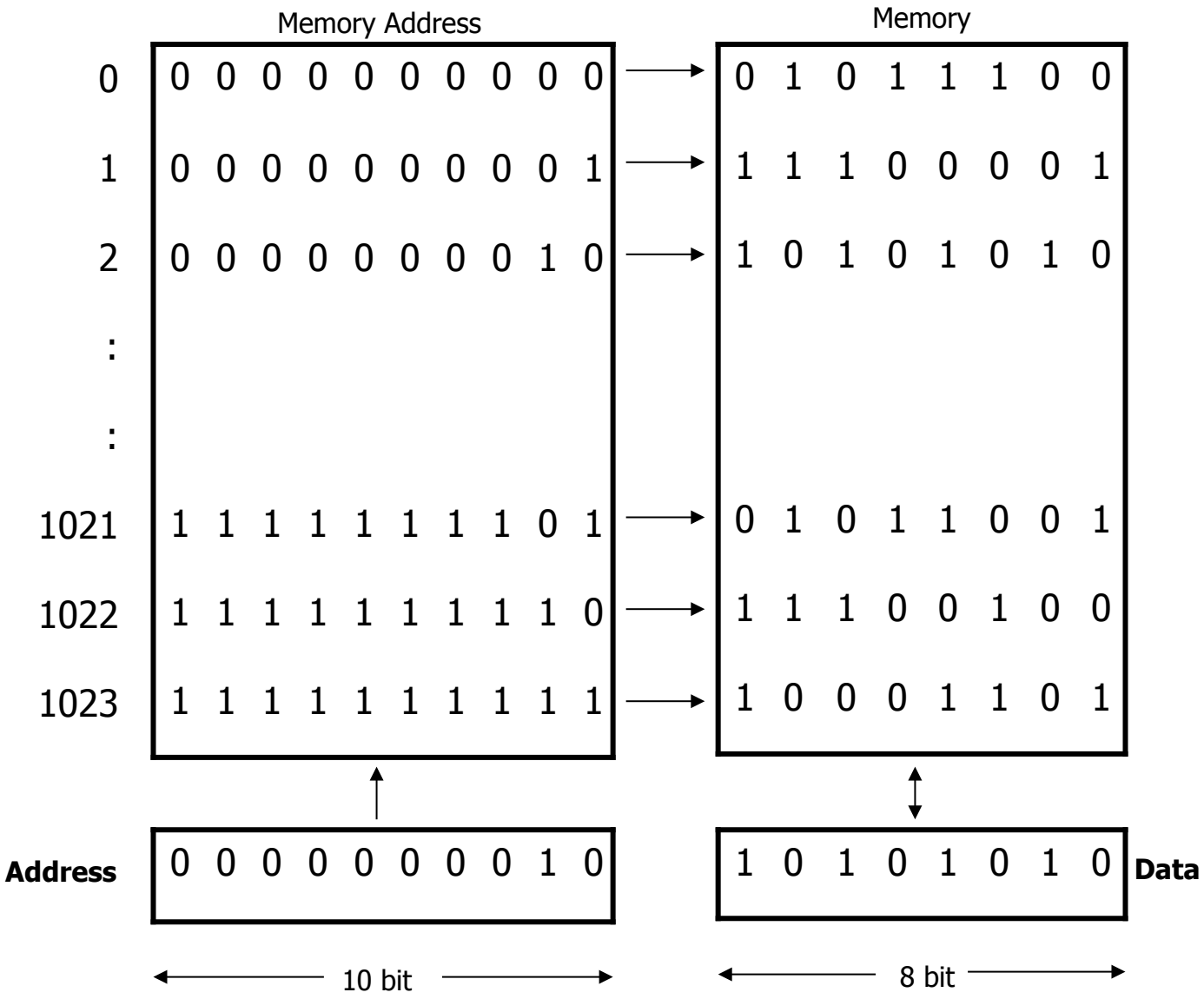
Example

Each word is identified with the address

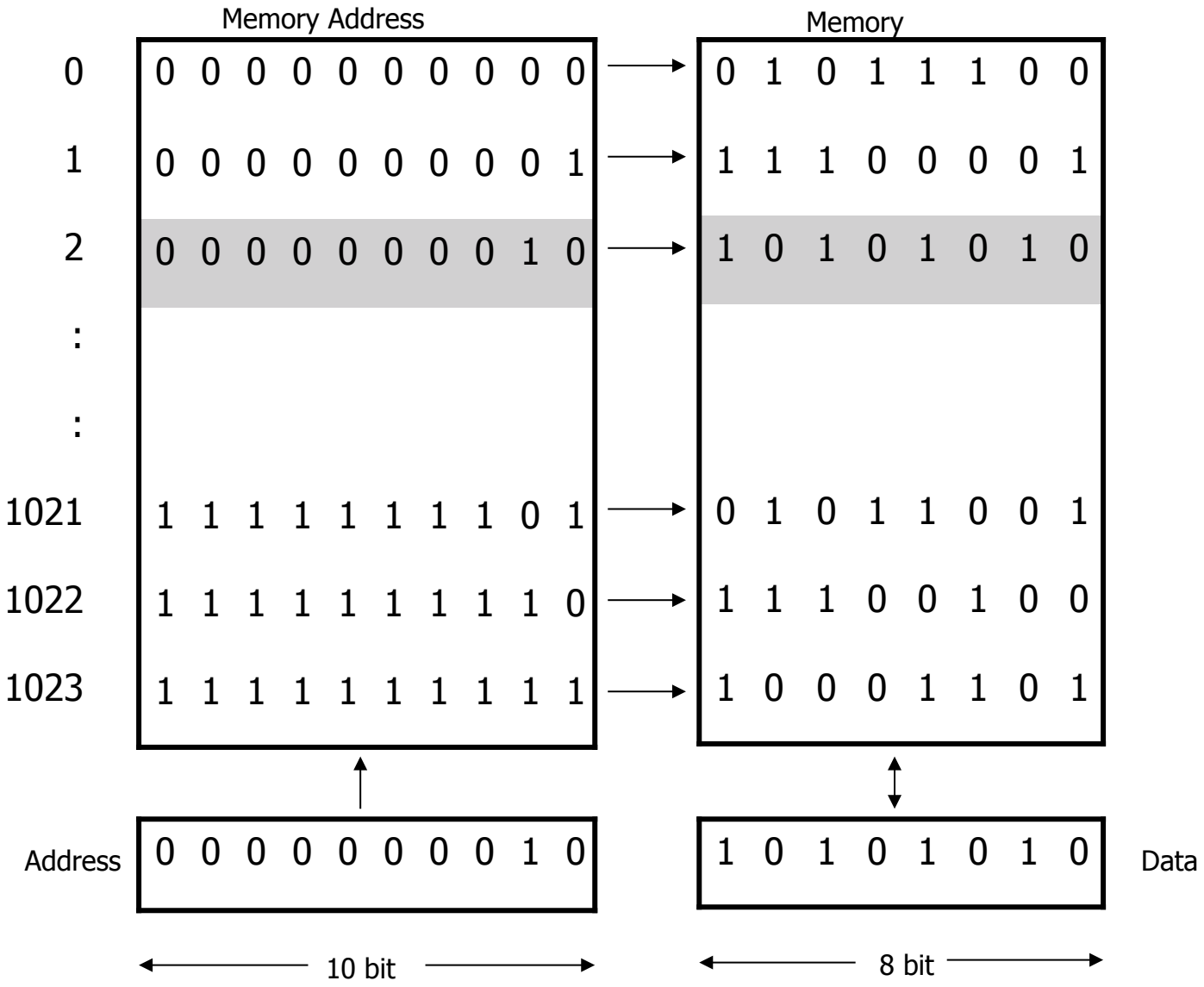
2^{10}=1024 (1K) words

Each word is 8-bit wide

1KByte memory

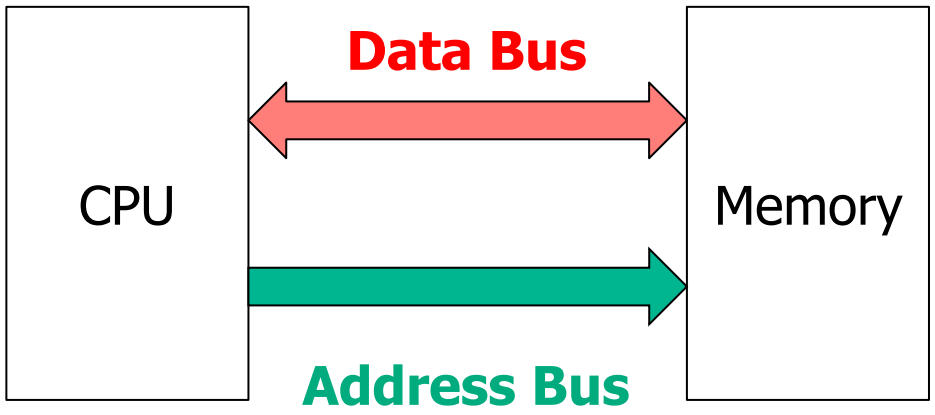


Example


Memory Address
Memory
0
1
2
:
:
1021
1022
1023
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 1 0
1 1 1 1 1 1 1 1 0 1
1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1
0 1 0 1 1 1 0 0
1 1 1 0 0 0 0 1
1 0 1 0 1 0 1 0
0 1 0 1 1 0 0 1
1 1 1 0 0 1 0 0
1 0 0 0 1 1 0 1
Address
0 0 0 0 0 0 0 0 1 0
1 0 1 0 1 0 1 0
Data
10 bit
8 bit

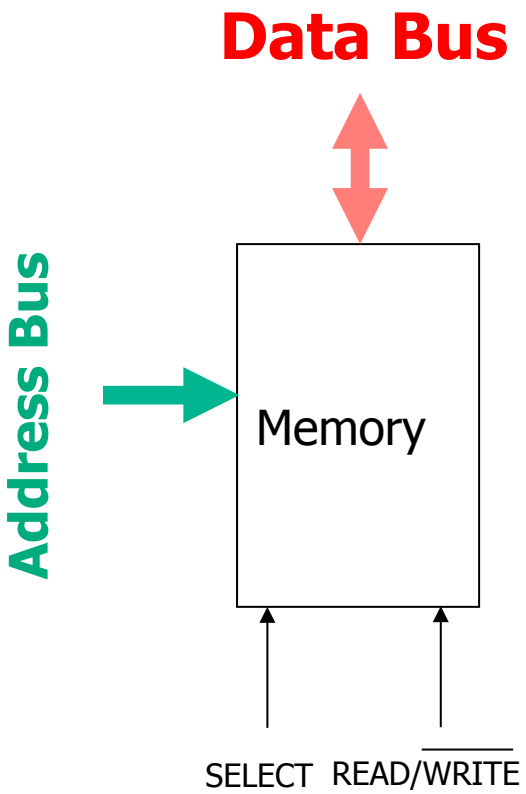
Memory Access

- Each memory location has a unique address
- Address from an instruction is set on address bus which finds the location in memory
- CPU determines if it is a store or retrieval
- Transfer takes place between the registers and memory

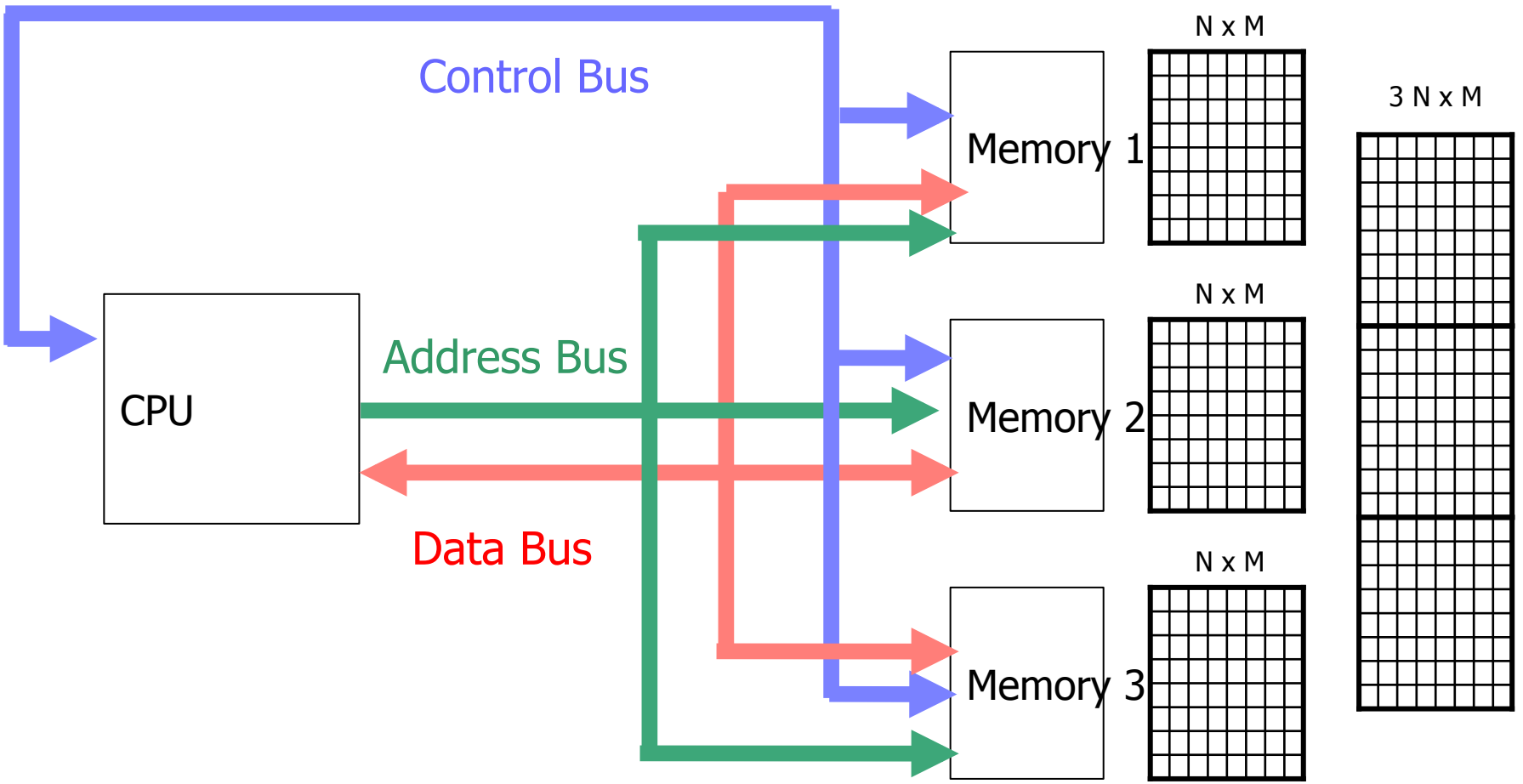


Memory Access

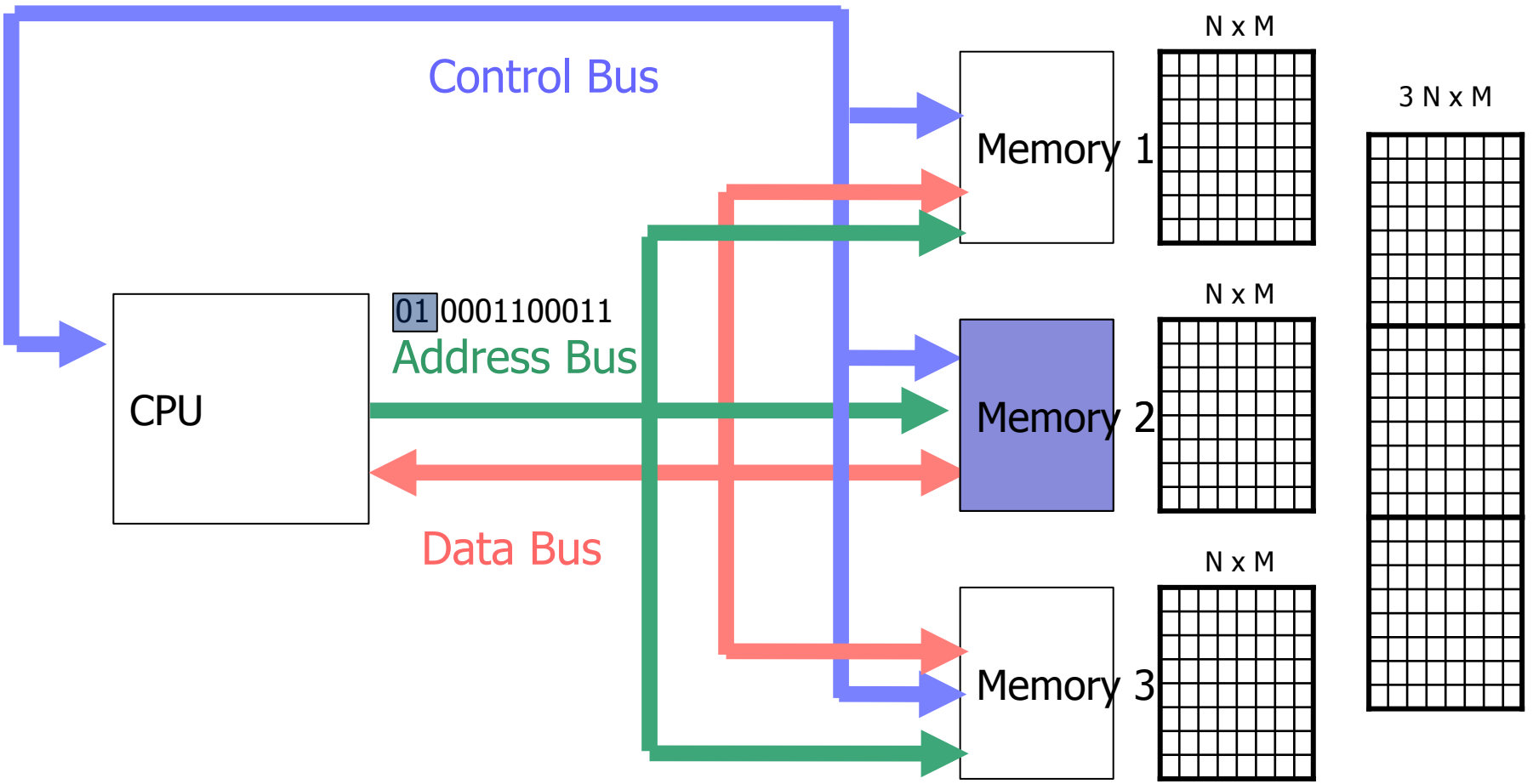
- For the microprocessor to access (Read or Write) information in memory (RAM or ROM), it needs to do the following:
  - Select the right memory chip (using part of the address bus).
  - Identify the memory location (using the rest of the address bus).
  - Access the data (using the data bus).



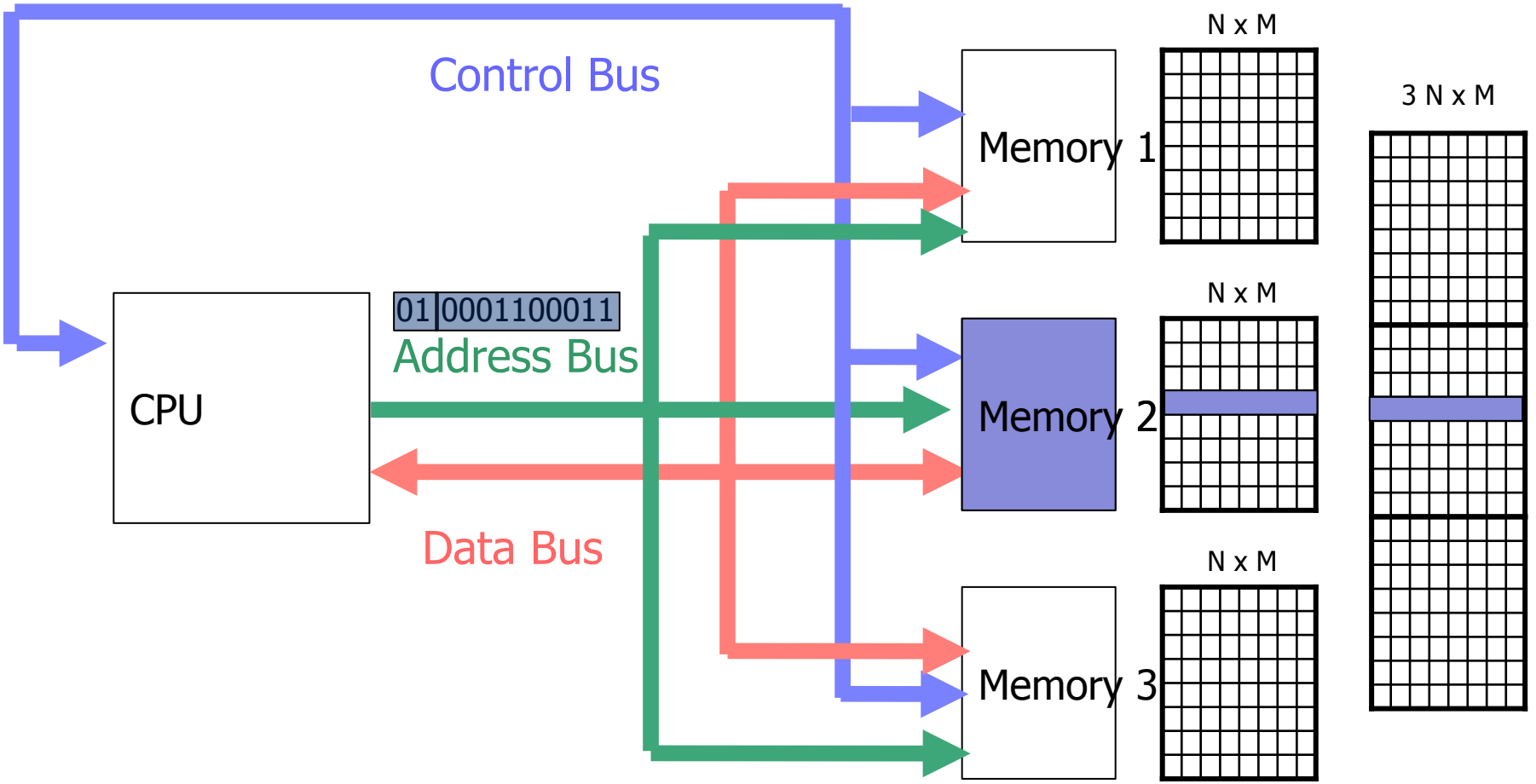
Memory Access



Memory Access

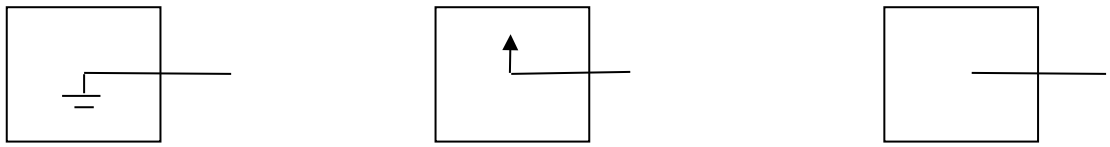


Memory Access



The Tri-State Buffer

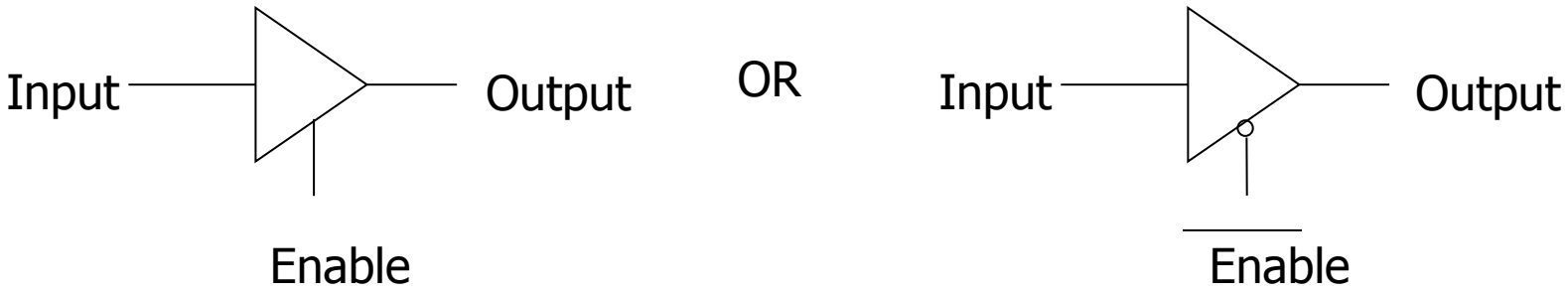
- An important circuit element that is used extensively in memory.
- This buffer is a logic circuit that has three states:
  - Logic 0, Logic 1, and high impedance.
  - When this circuit is in high impedance mode, it looks as if it is disconnected from the output completely.



The Output is Low The Output is High High Impedance

The Tri-State Buffer

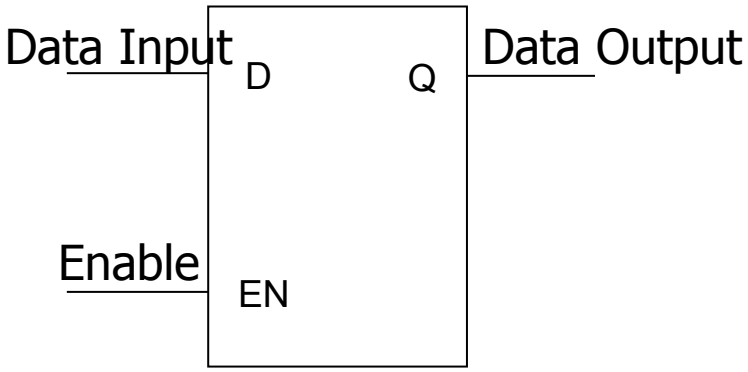
- This circuit has two inputs and one output.
  - The first input behaves like the normal input for the circuit.
  - The second input is an "enable".
    - If it is set high, the output follows the proper circuit behavior.
    - If it is set low, the output looks like a wire connected to nothing.



The Basic Memory Element

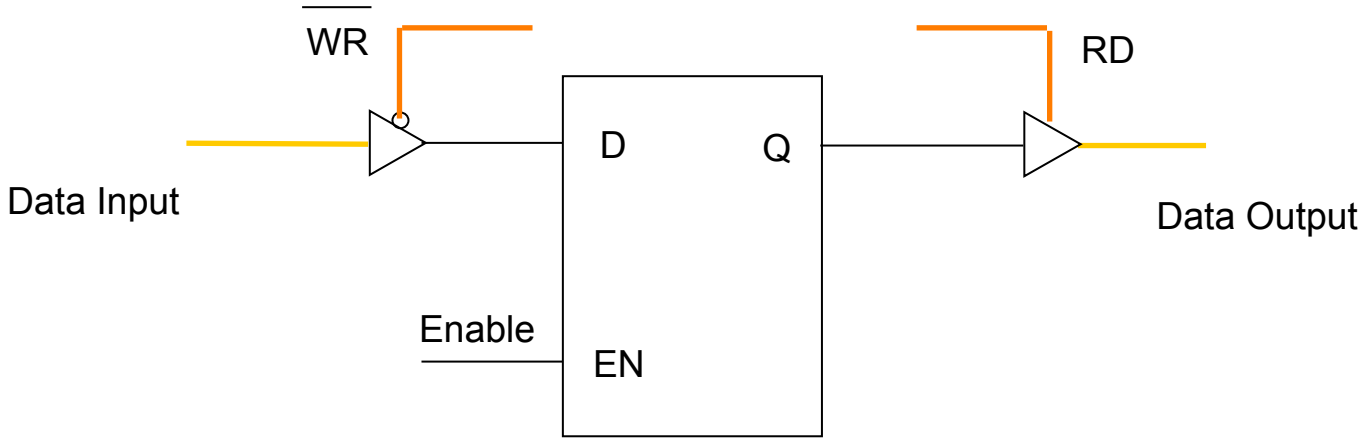
- The basic memory element is similar to a D latch.
  - Input
  - Output
  - Enable

D	Q(t+1)
0	0
1	1



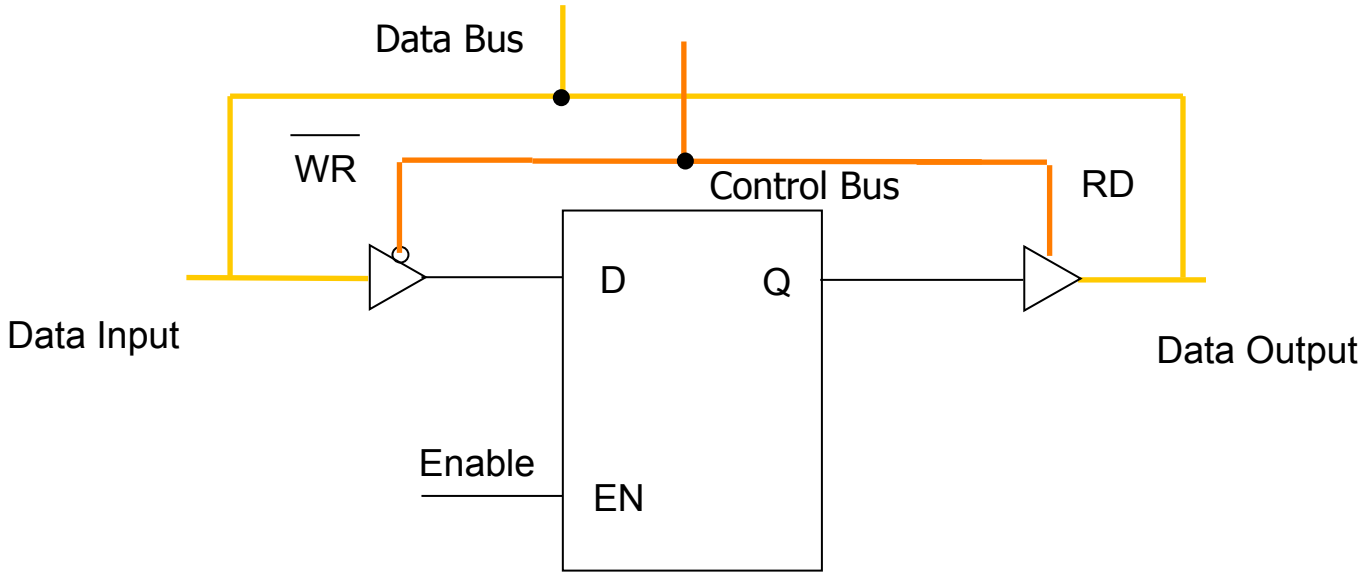
The Basic Memory Element

- Data is always present on the input and the output is always set to the content of the latch.
- Tri-state buffers are added at the input and output of the latch.



The Basic Memory Element

- Data is always present on the input and the output is always set to the content of the latch.
- Tri-state buffers are added at the input and output of the latch.

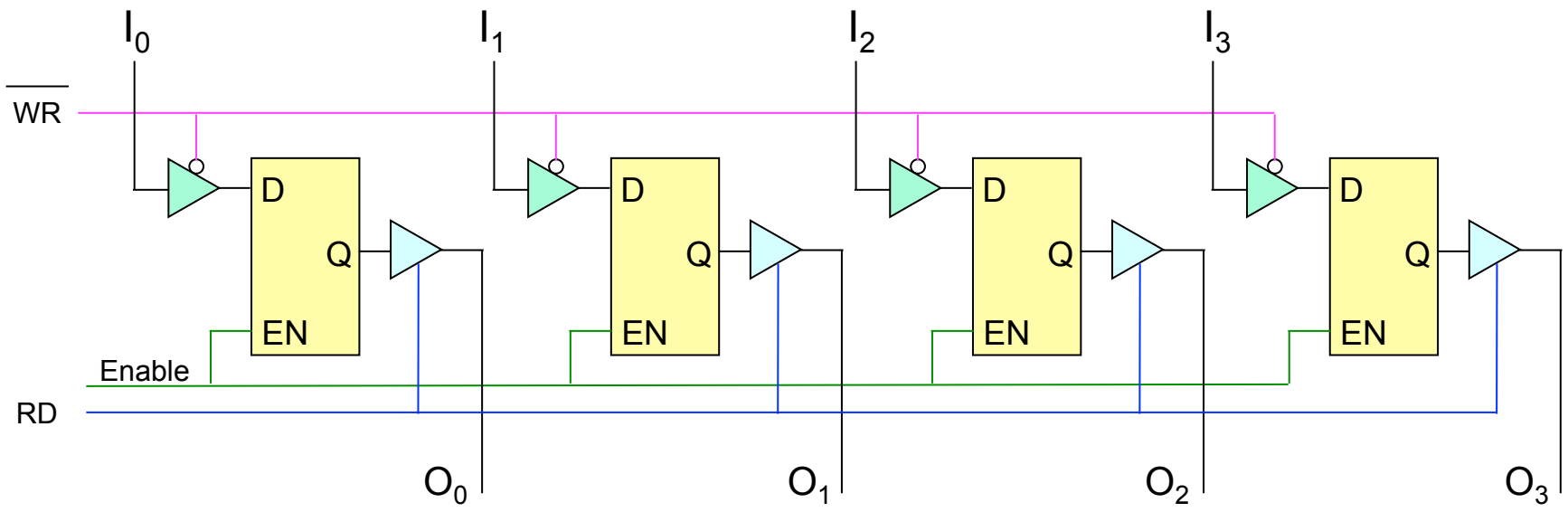


The Basic Memory Element

- The WR signal controls the input buffer.
- The bar over WR means that this is an active low signal.
- If WR is 0 the input data reaches the latch input.
- If WR is 1 the input of the latch looks like a wire connected to nothing.
- The RD signal controls the output in a similar manner.

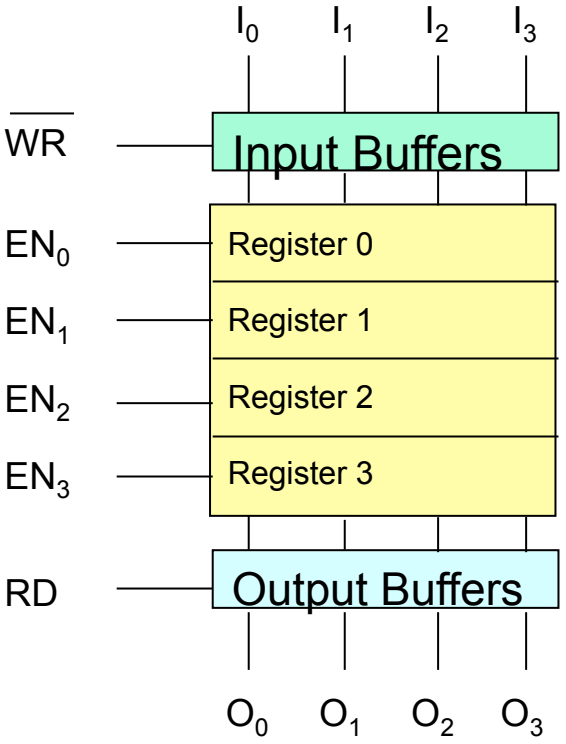
How to construct a "Register"

- If four latches are connected together, a 4-bit register is obtained

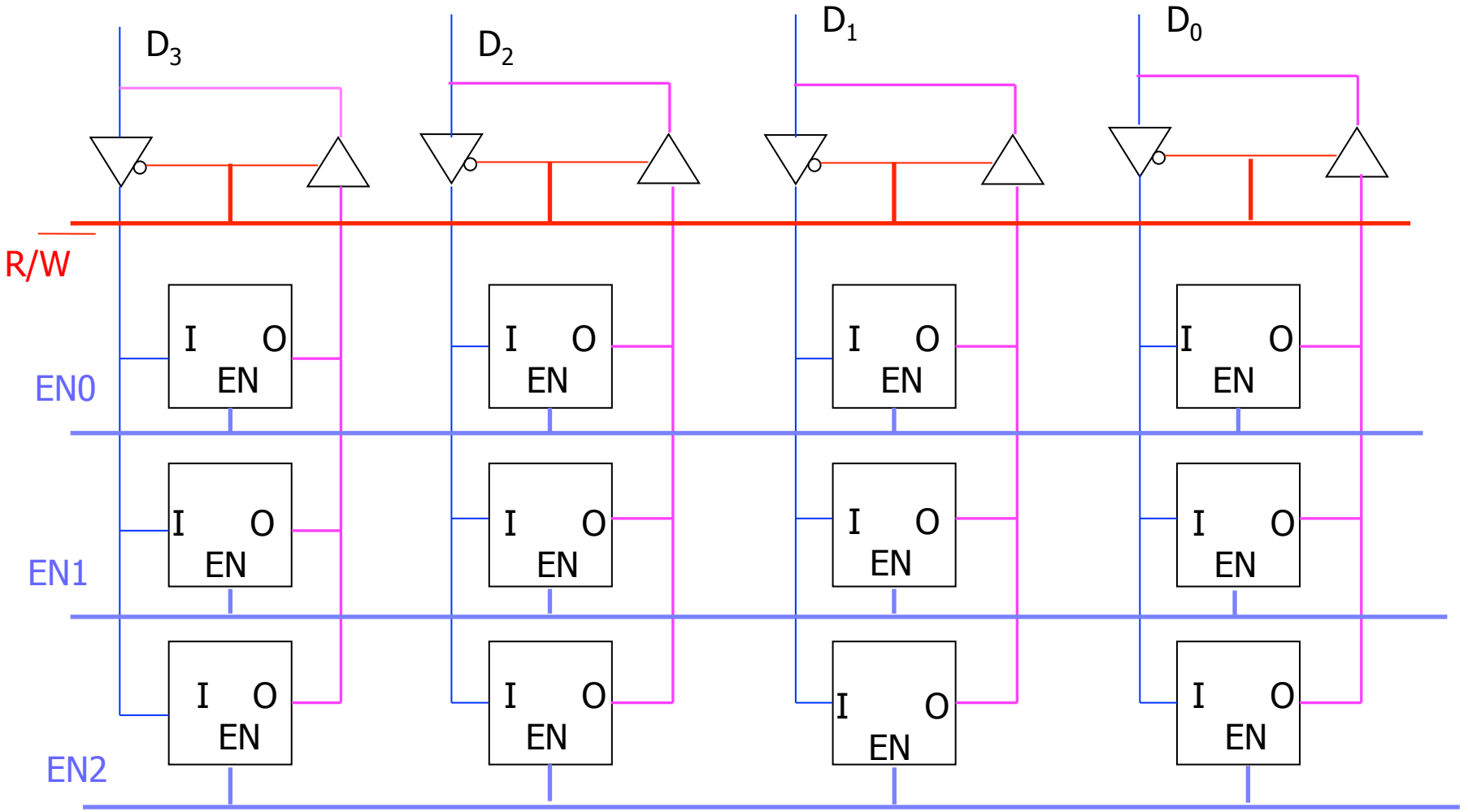


Memory Addressing

- Using the RD and WR controls we can determine the **direction of flow** either **into** or **out** of memory.
- Using the appropriate Enable input we enable an individual register.
  - Since we can **never** have more than one of these enables active at the same time, we can have them encoded to reduce the number of lines coming into the chip.
  - These encoded lines are the address lines for memory.

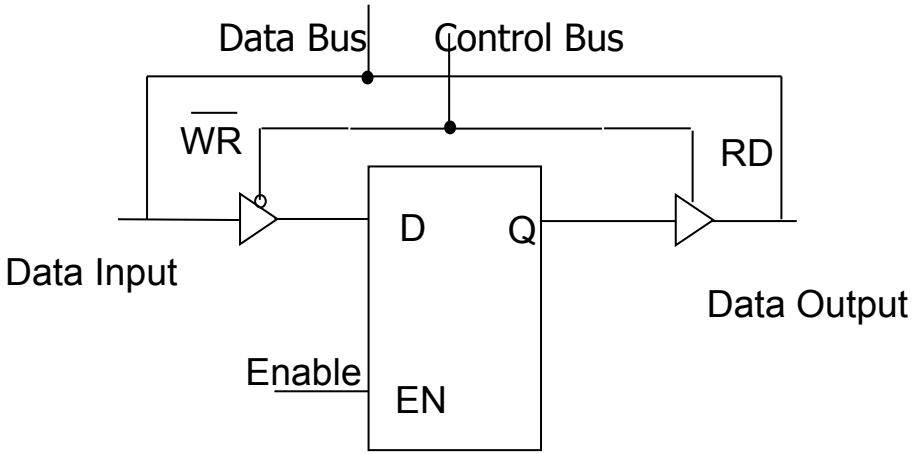


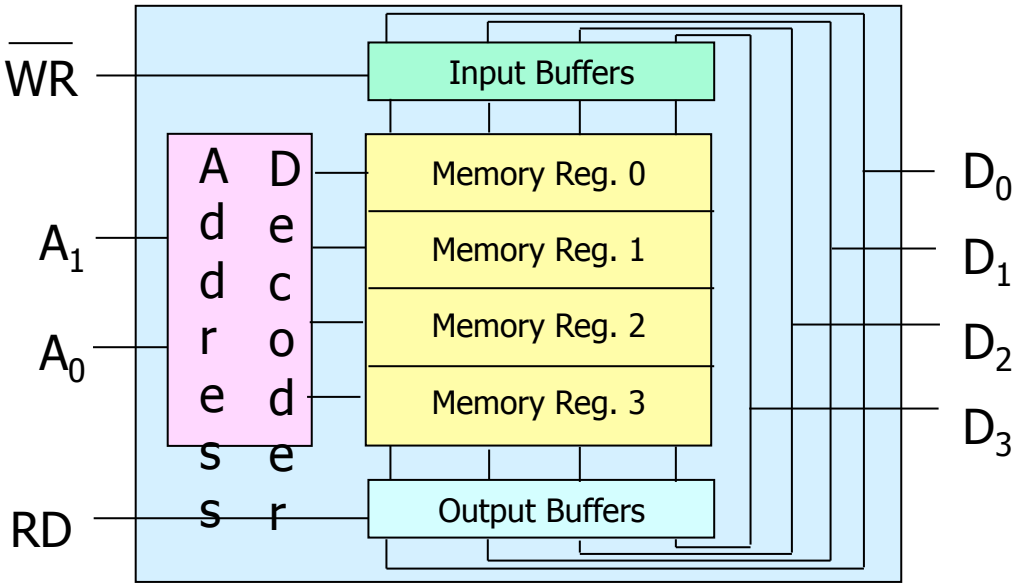
Memory Organization



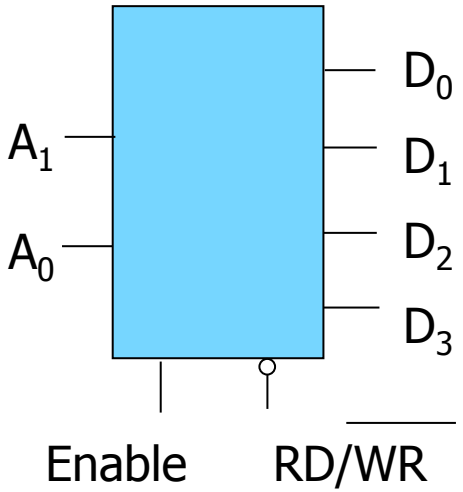
The Design of a Memory Chip

- Since we have tri-state buffers on both the inputs and outputs of the flip flops, we can actually use one set of pins only.



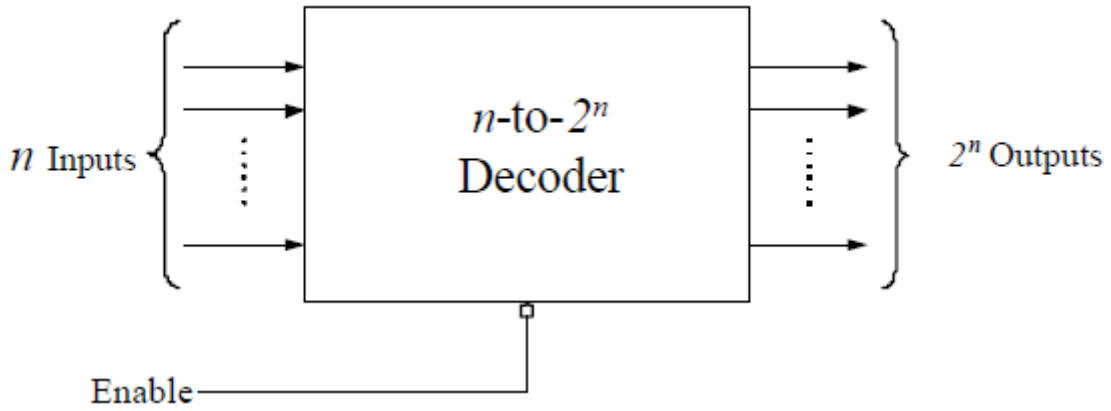


4x4 Memory



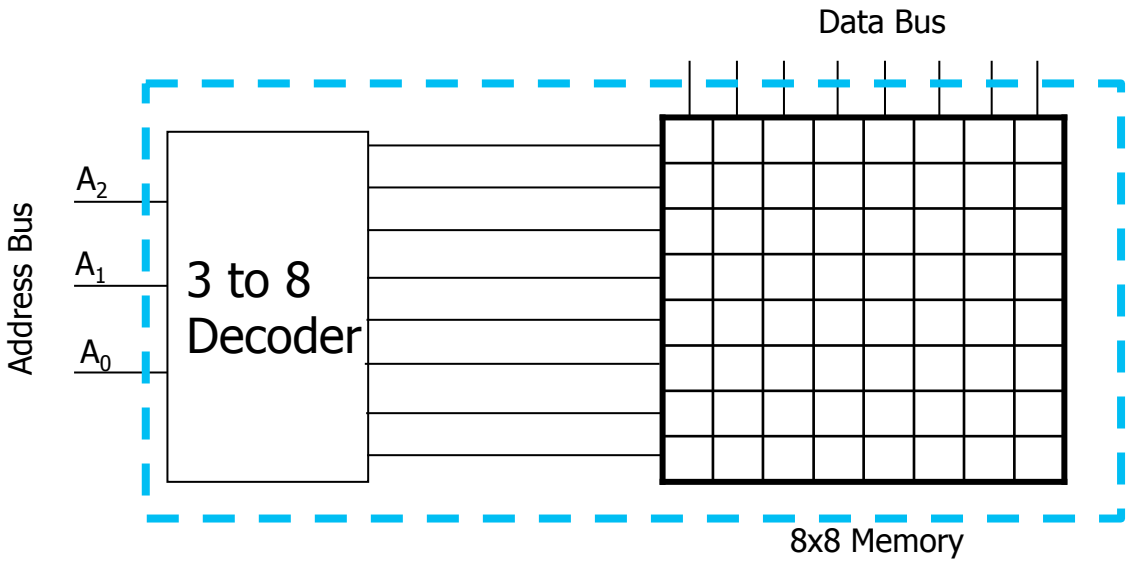
Memory Components

- Decoder is a combinational circuit that converts binary information from n-coded inputs to maximum of 2^n outputs

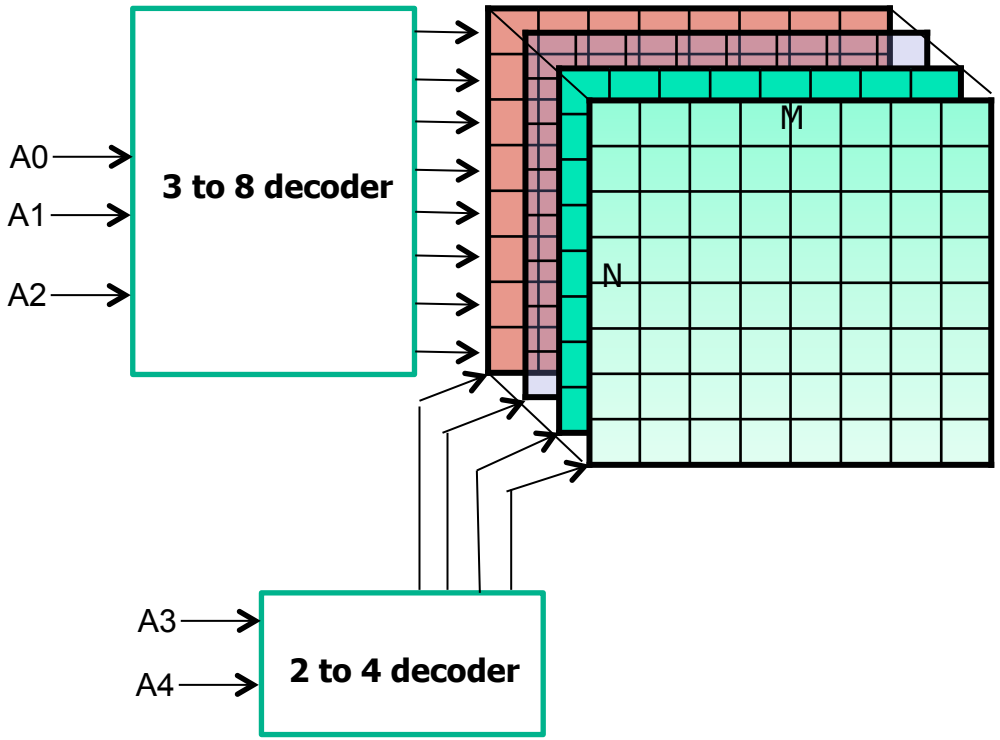


- n coded inputs to 2^n outputs

One Dimensional Addressing



Two Dimensional Addressing

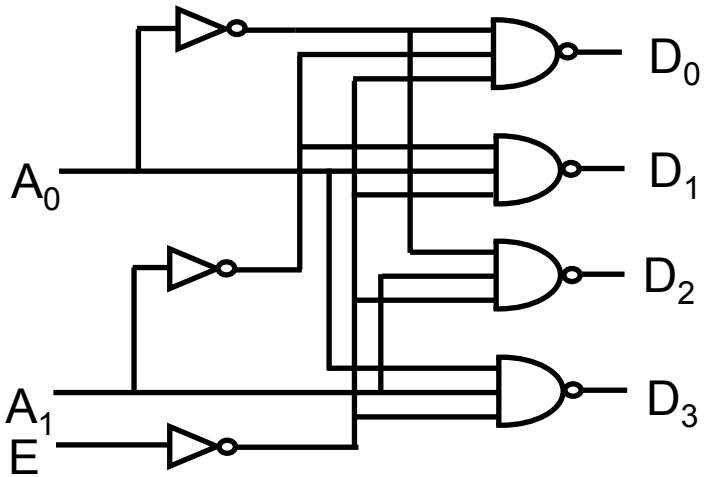


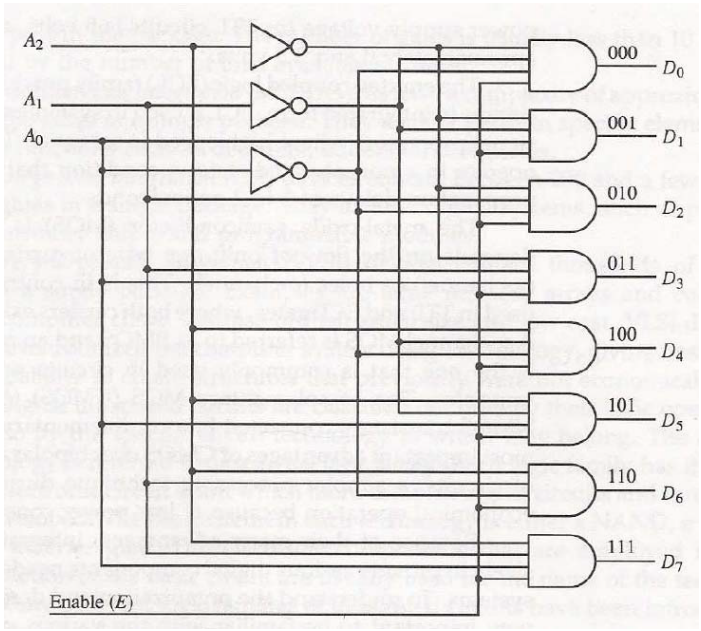
32X8 memory arranged in 2D configuration

Why?

Two Dimensional Addressing II

- This arrangement is more economical than 5-to-32 Decoder
- Check the cost of 2-4 decoder and 3-8 decoder

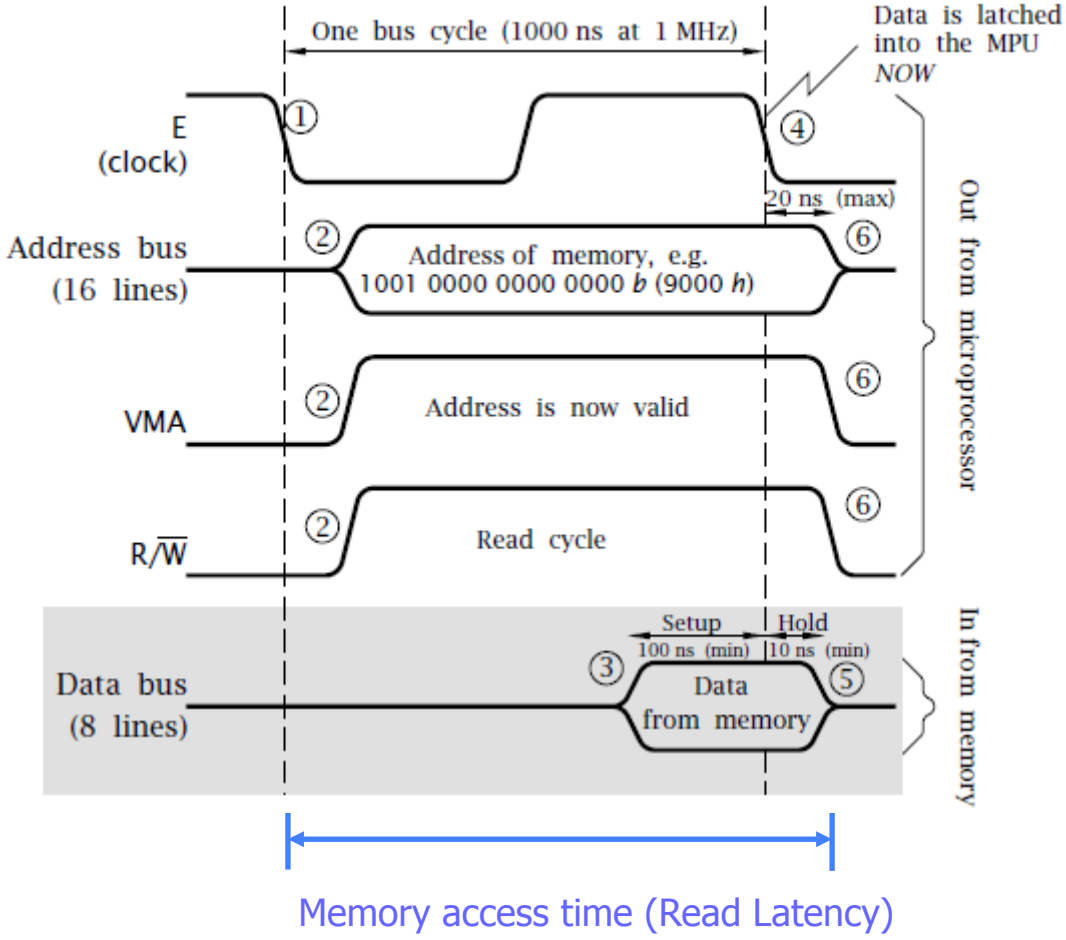




Memory Read

Motorola 6802 example:

- CPU applies the address on the address bus at the falling edge of the clock ①
- Proper select input for the memory is selected ②
- VMA signal indicates valid memory address ②
- MPU applies R/W signal HIGH to designate read cycle ②
- Memory places the data on the data bus ③
- CPU latches Data on the falling edge of the clock ④

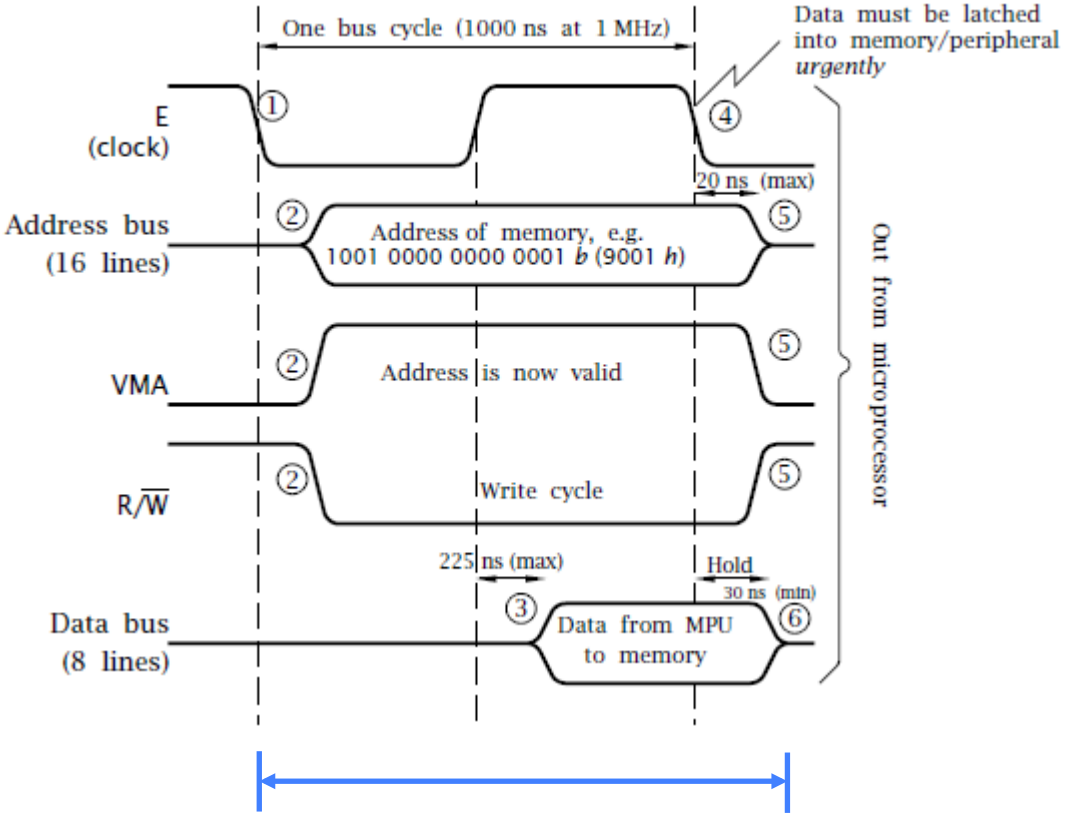


* Microprocessing Unit (MPU)
* Valid Memory Address (VMA)

Memory Write

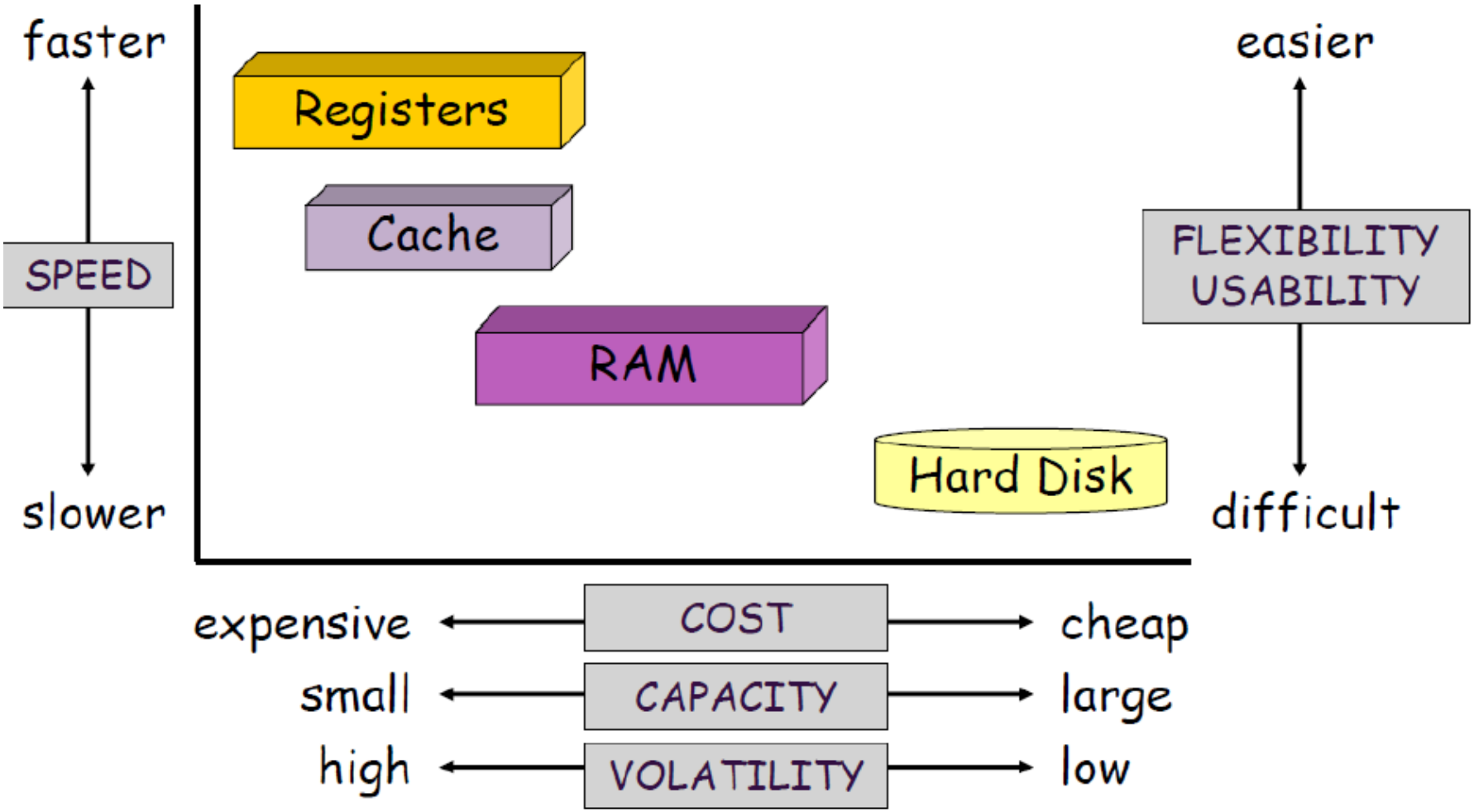
Motorola 6802 example:

- CPU applies the address on the address bus at the falling edge of the clock ①
- Proper select input for the memory is selected ②
- VMA signal indicates valid memory address ②
- CPU applies R/W signal LOW to designate write cycle ②
- MPU places the data on the data bus ③
- Memory latches Data on the falling edge of the clock ④



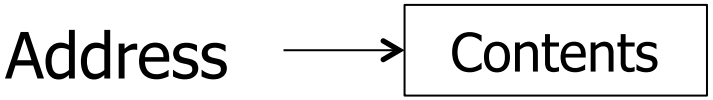
Memory access time (Write Latency)

Comparison of Memory Modules



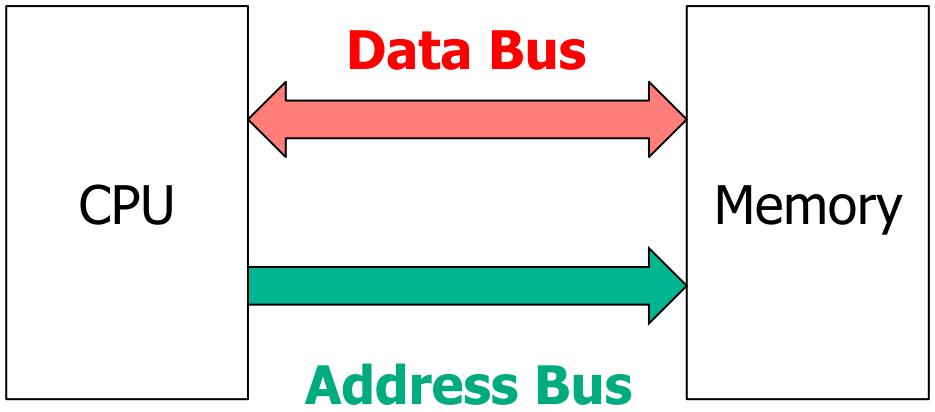
Memory Addressing

- Memory consists of a sequence of directly addressable "locations".
- A memory location is referred to as an information unit.
- An information unit has two components:
  - address
  - contents



Memory Addressing

- Each location in memory has an address that must be supplied before its contents can be accessed.
- The CPU communicates with memory by first identifying the location's address and then passing this address on the address bus.
- The data are transferred between memory and the CPU along the data bus.
- The number of bits that can be transferred on the data bus at once is called the data bus width of the processor.



Dimensions of Memory

- Memory is usually measured by two numbers: its length and its width (length x width).
  - The length is the total number of locations.
  - The width is the number of bits in each location.
- The length (total number of locations) is a function of the number of address lines.

  # of memory locations = $2^{(\#\text{ of address lines})}$
  - A memory chip with 10 address lines would have

    2^{10} = 1024 locations (1K)
  - A memory chip with 4K locations would need

    $\log_2 4096 = 12$ address lines

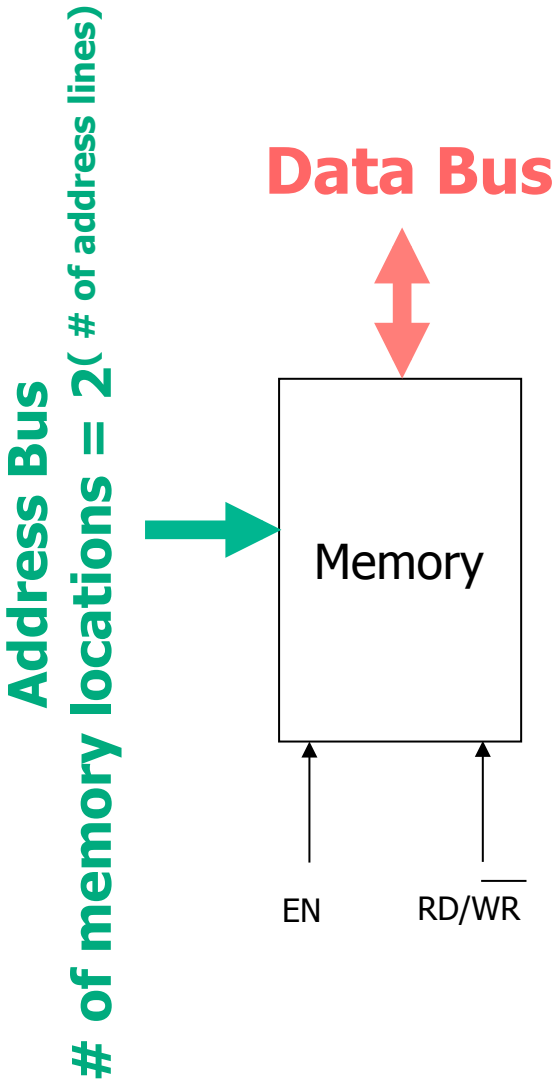
An Example CPU and Memory

- Consider an example CPU that has 16 address lines
  - It can address

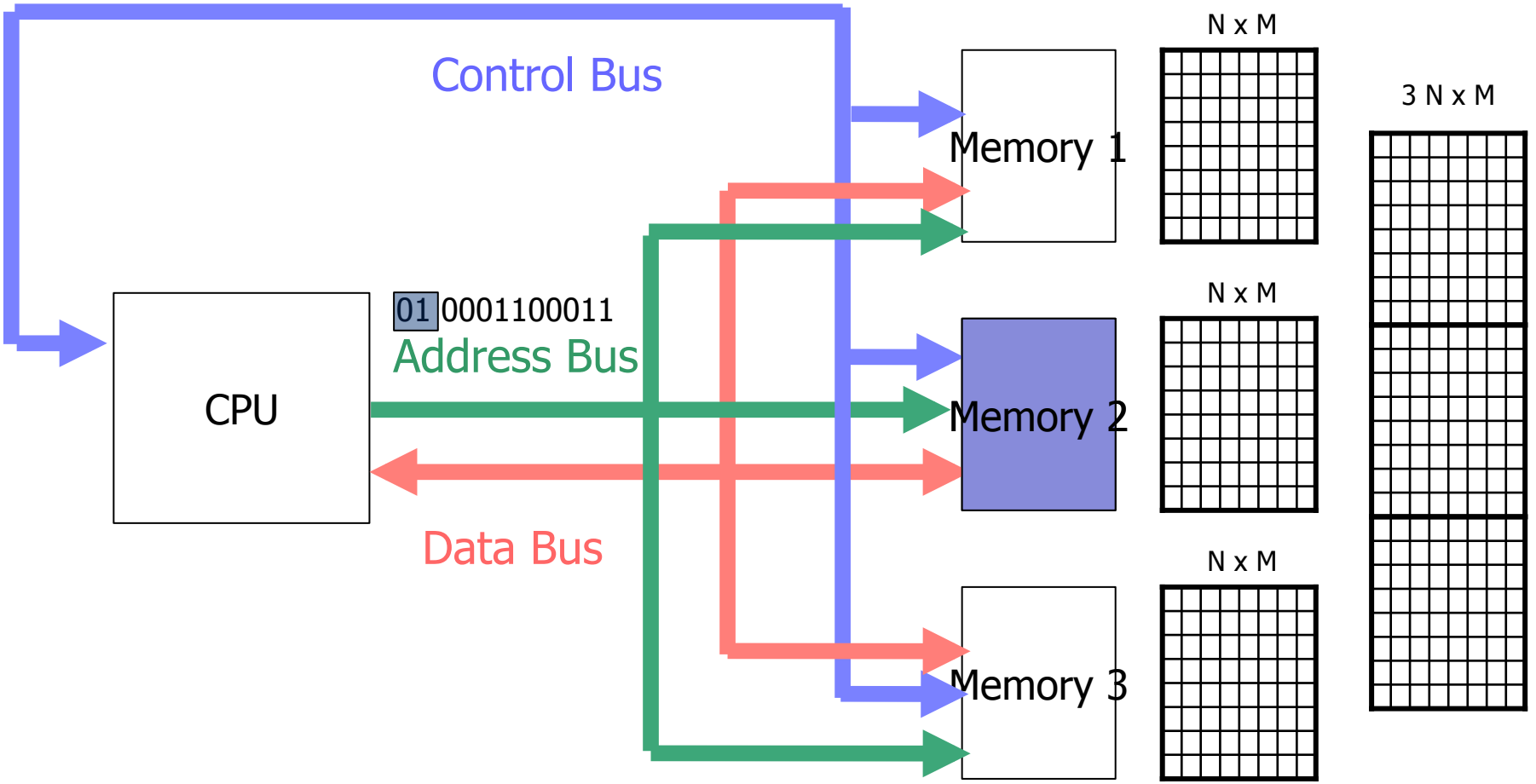
    2^{16} = 64K memory locations
- Then it will need 1 memory chip with 64 K locations, or 2 chips with 32 K in each, or 4 with 16 K each or 16 of the 4 K chips, etc.
- How would we use these address lines to control the multiple chips?

Chip Select

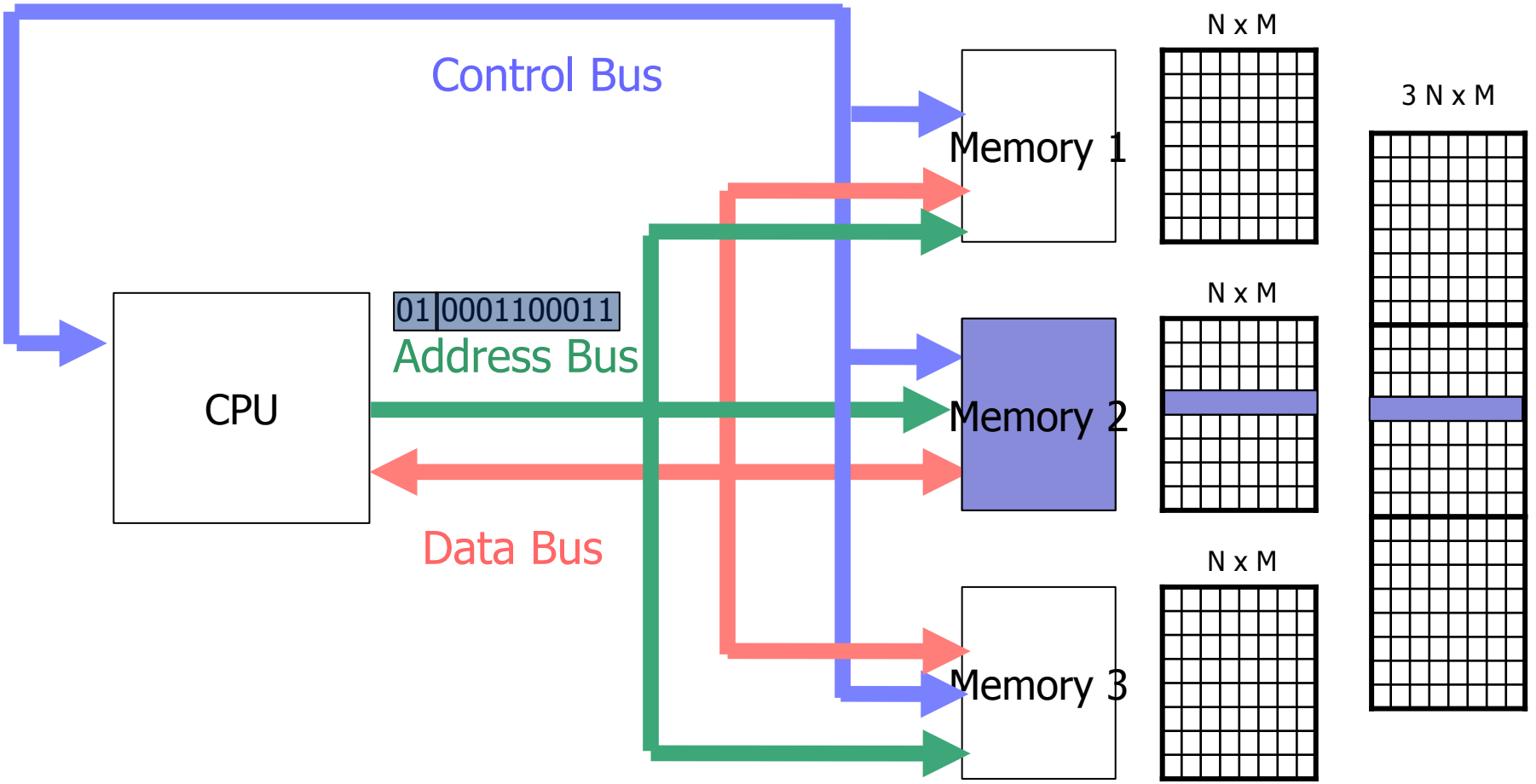
- Usually, each memory chip has a Chip Select (CS) input. The chip will only work if an active signal is applied on that input.
- To allow the use of multiple chips in the make up of memory, we need to use a number of address lines for the purpose of chip selection.
- These address lines are decoded to generate the 2^n necessary CS inputs for the memory chips to be used.



Memory Access

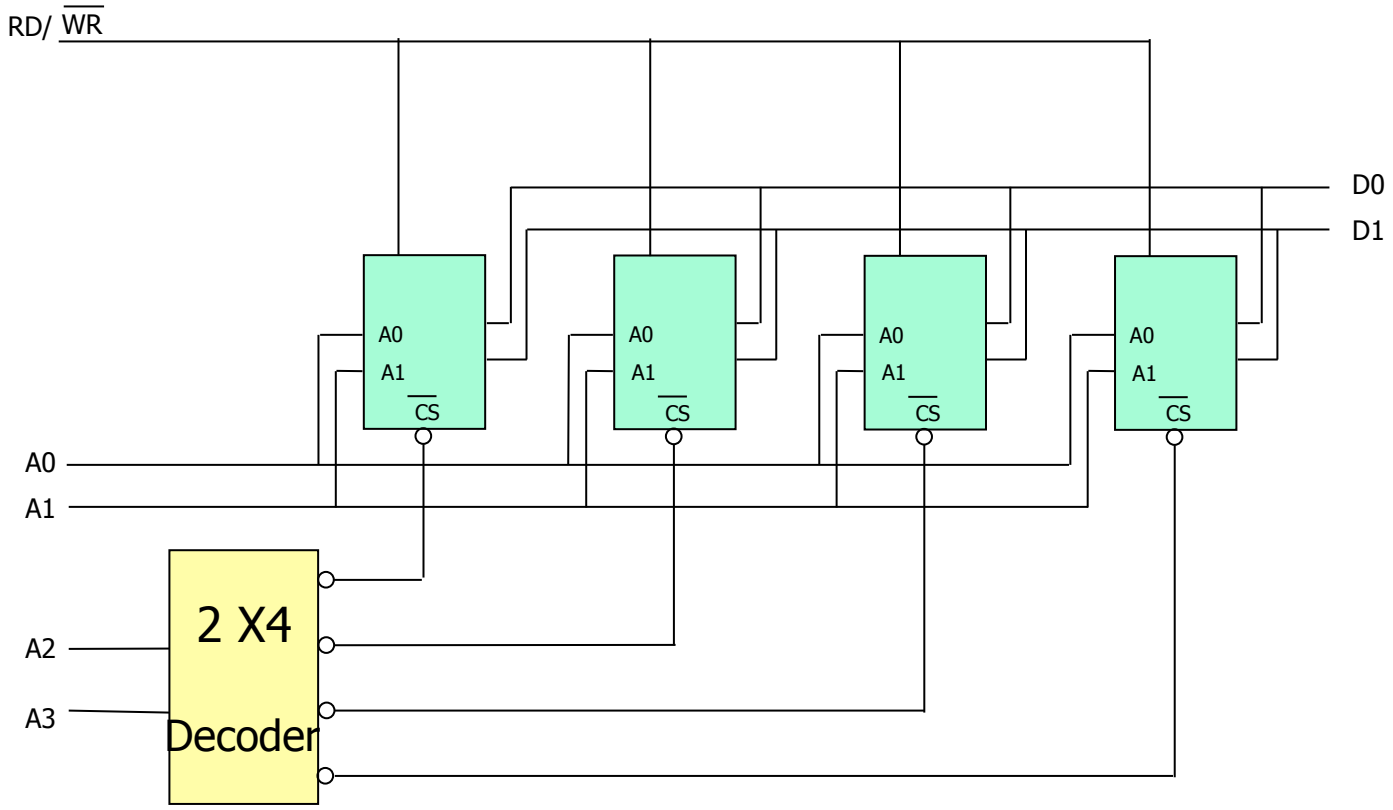


Memory Access



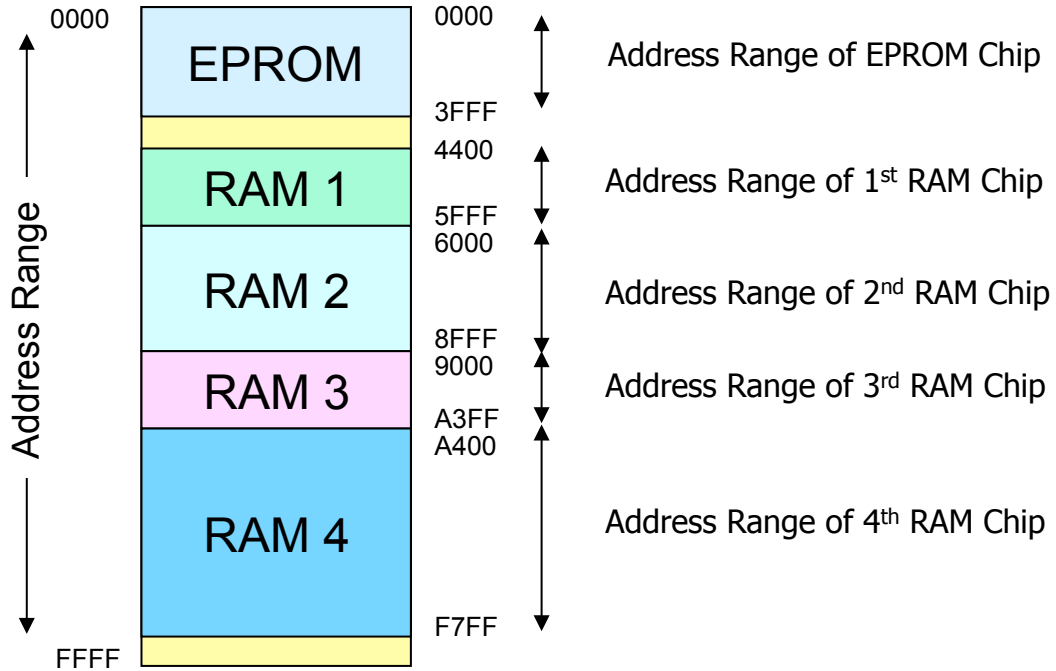
Chip Selection Example

- A memory system made up of 4 of the 4x2 memory chips



Memory Map

- Designates the address space for each memory chip

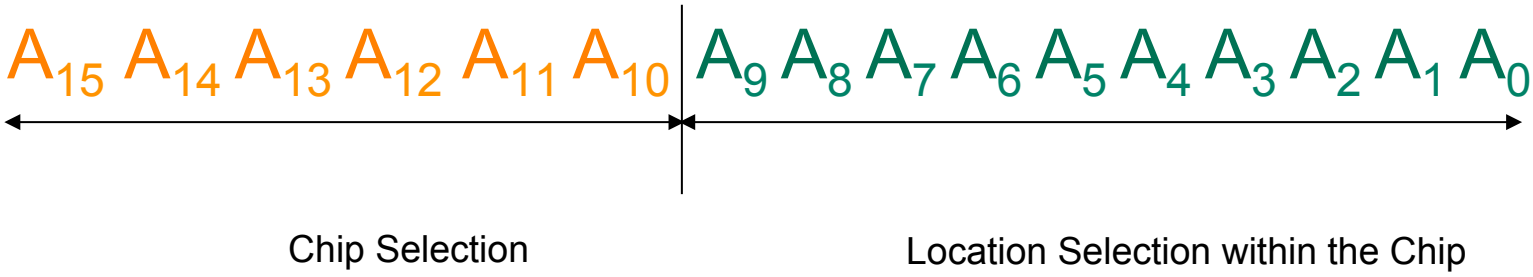


Address Range of a Memory Chip

- The address range of a particular chip is the list of all addresses that are mapped to the chip.
- An 8-bit CPU with 16 address bits can address a total of 64K memory locations.
  - If we use memory chips with 1K locations each, then we will need 64 such chips.
  - The 1K memory chip needs 10 address lines to uniquely identify the 1K locations. ($\log_2 1024 = 10$)
  - That leaves 6 address lines which is the exact number needed for selecting between the 64 different chips ($\log_2 64 = 6$).

Address Range of a Memory Chip

- 16 bit address lines can be separated into two pieces



  - Depending on the combination on the address lines A_{15}-A_{10}, the address range of the specified chip is determined.

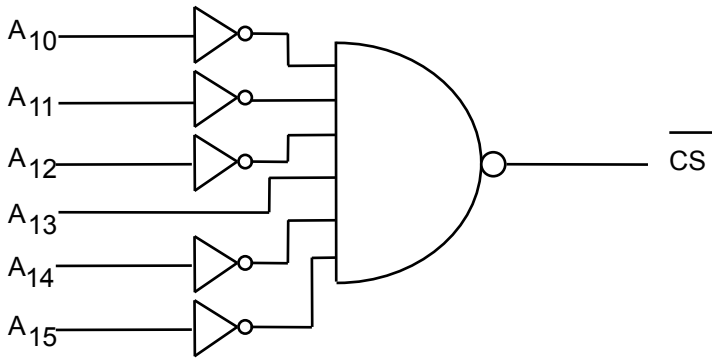
Chip Select Example

- A chip that uses the combination [A_{15}-A_{10}]= 001000 would have addresses that range from $2000 to $23FF.
  - The 10 address lines on the chip gives a range of

xxxx xx00 0000 0000 to xxxx xx11 1111 1111 or

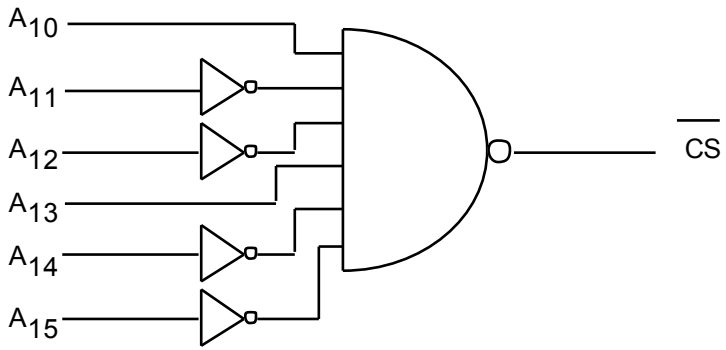
$x000 to $x3FF for each of the chips.

  - The memory chip in this example would require the following NAND circuit on its chip select input:



Chip Select Example

- If we change the above combination to the following:

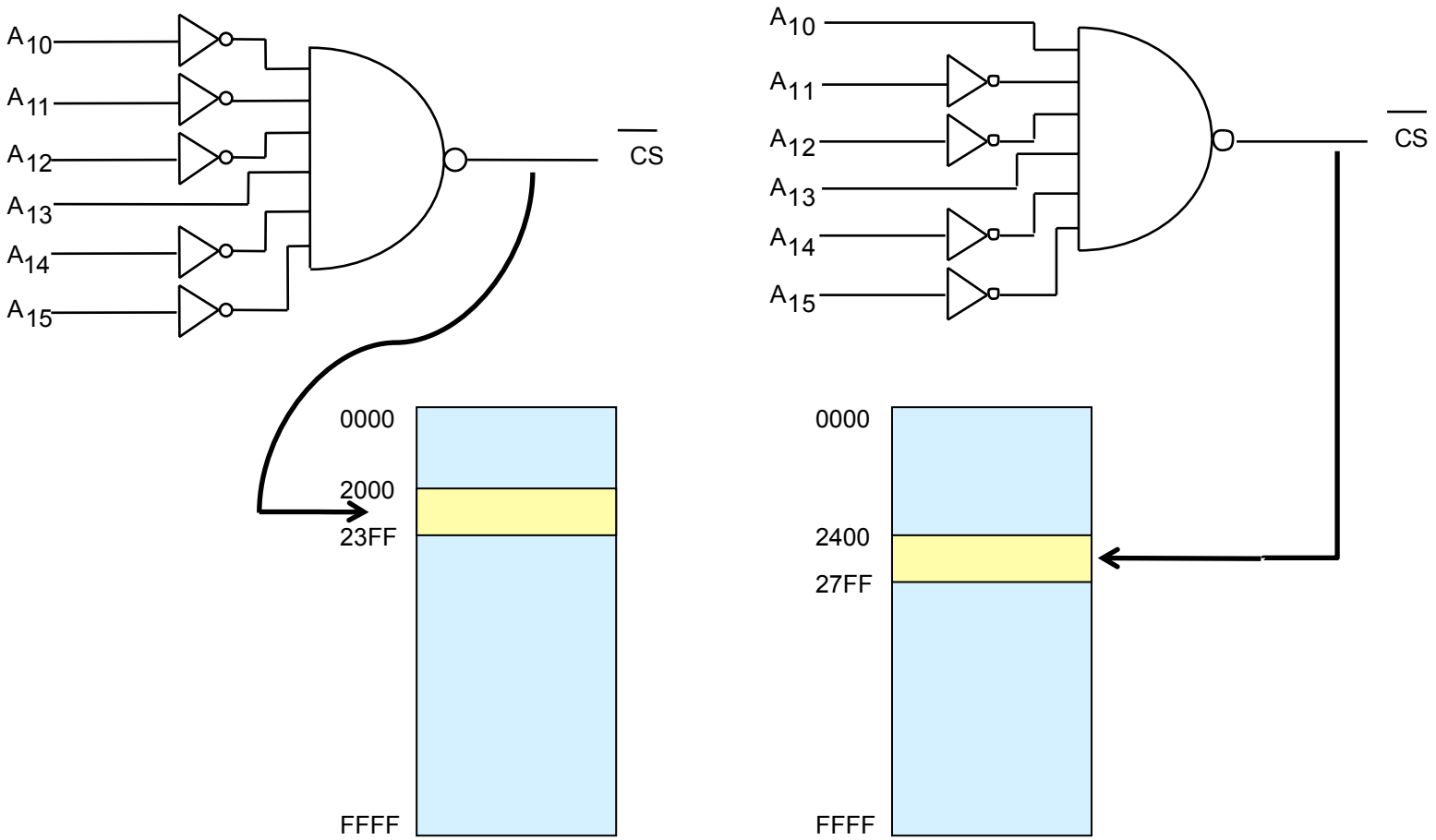


- Now the chip would have addresses ranging from: 2400 to 27FF.

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	=>$2400
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	=>$27FF

- Changing the combination of the address bits connected to the chip select changes the address range for the memory chip.

Chip Select Example



Memory Organization

- Example: For a CPU with 8-bit data bus and 16-bit address bus, build the memory that spans between $0000 and $1FFF with 2Kx8 memory chips.

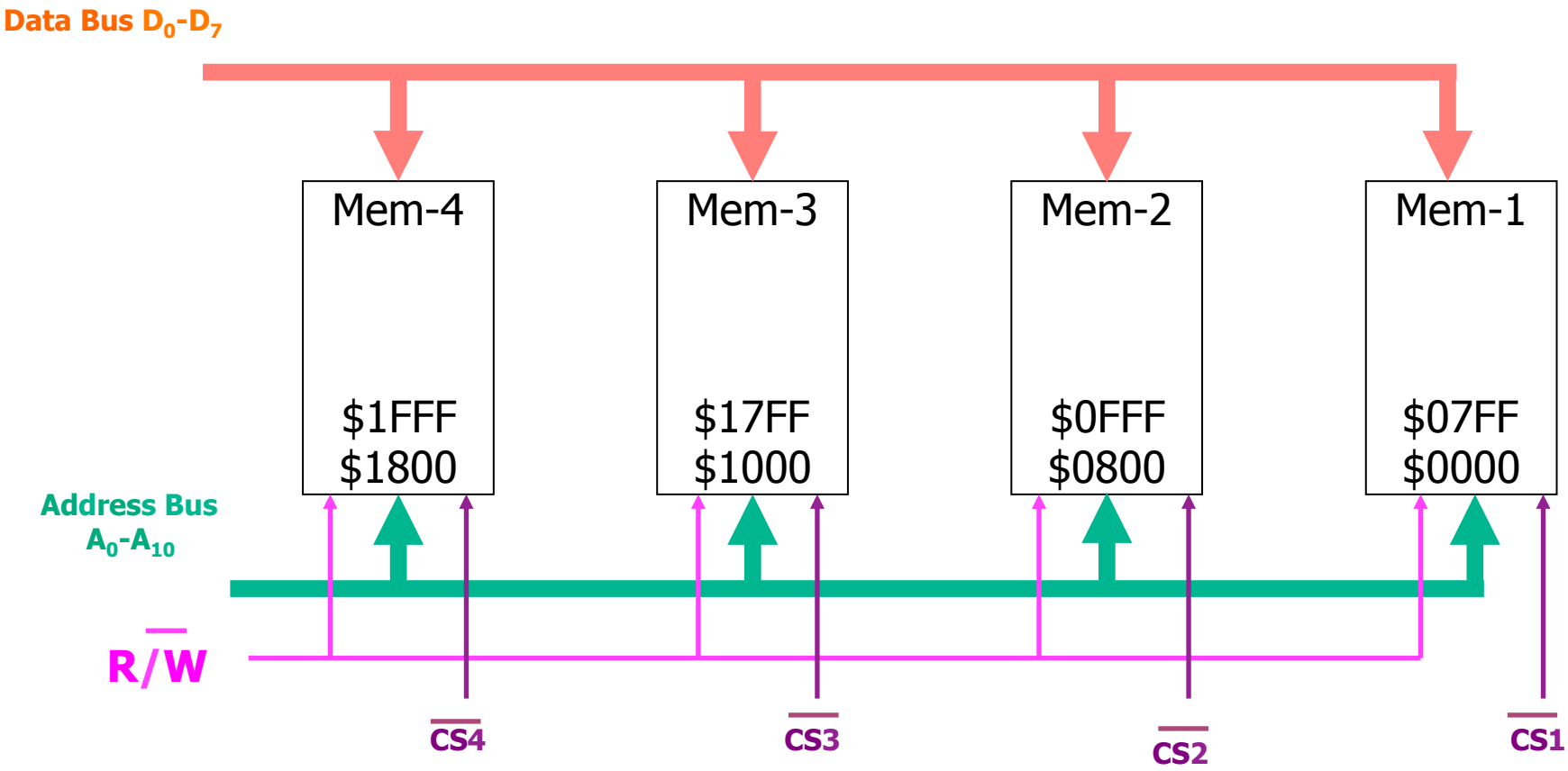
- What is the required memory space?
- How many 2K chips are needed?

Memory Organization

$A_{15}A_{14}A_{13}A_{12}$	$A_{11}A_{10}A_9A_8$	$A_7A_6A_5A_4$	$A_3A_2A_1A_0$	
0000	0000	0000	0000	$0000
0000	0111	1111	1111	$07FF
0000	1000	0000	0000	$0800
0000	1111	1111	1111	$0FFF
0001	0000	0000	0000	$1000
0001	0111	1111	1111	$17FF
0001	1000	0000	0000	$1800
0001	1111	1111	1111	$1FFF

Memory Organization

- Connect the DATA BUS, ADDRESS BUS, R/W together
- CS is determined using A12 and A11



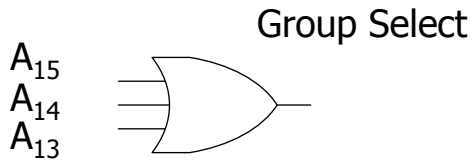
Memory Organization

- Chip select is done with address bits that are not used within the memory chip.

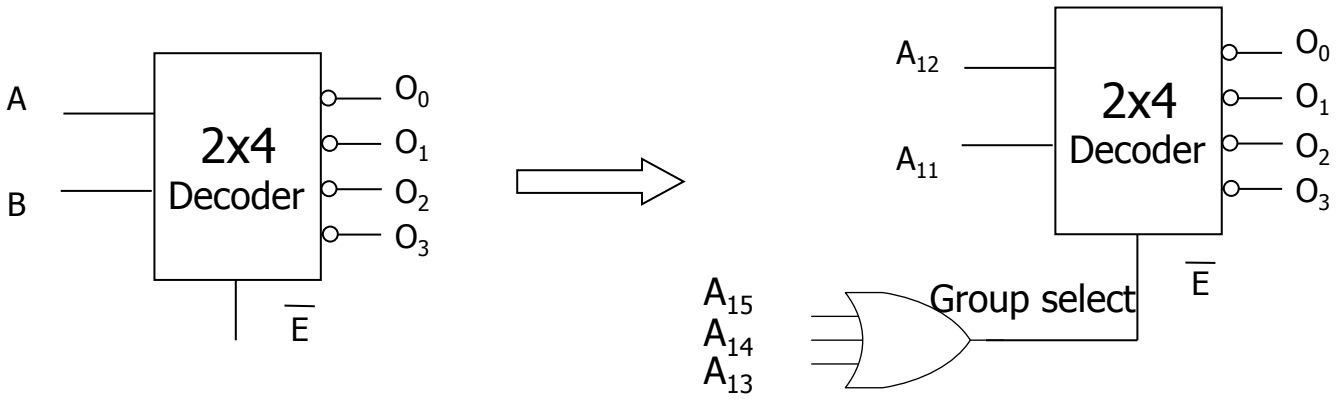
	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}.....
Memory 1	0	0	0	0	0	Used to address locations within a memory chip
Memory 2	0	0	0	0	1	
Memory 3	0	0	0	1	0	
Memory 4	0	0	0	1	1	

Memory Organization

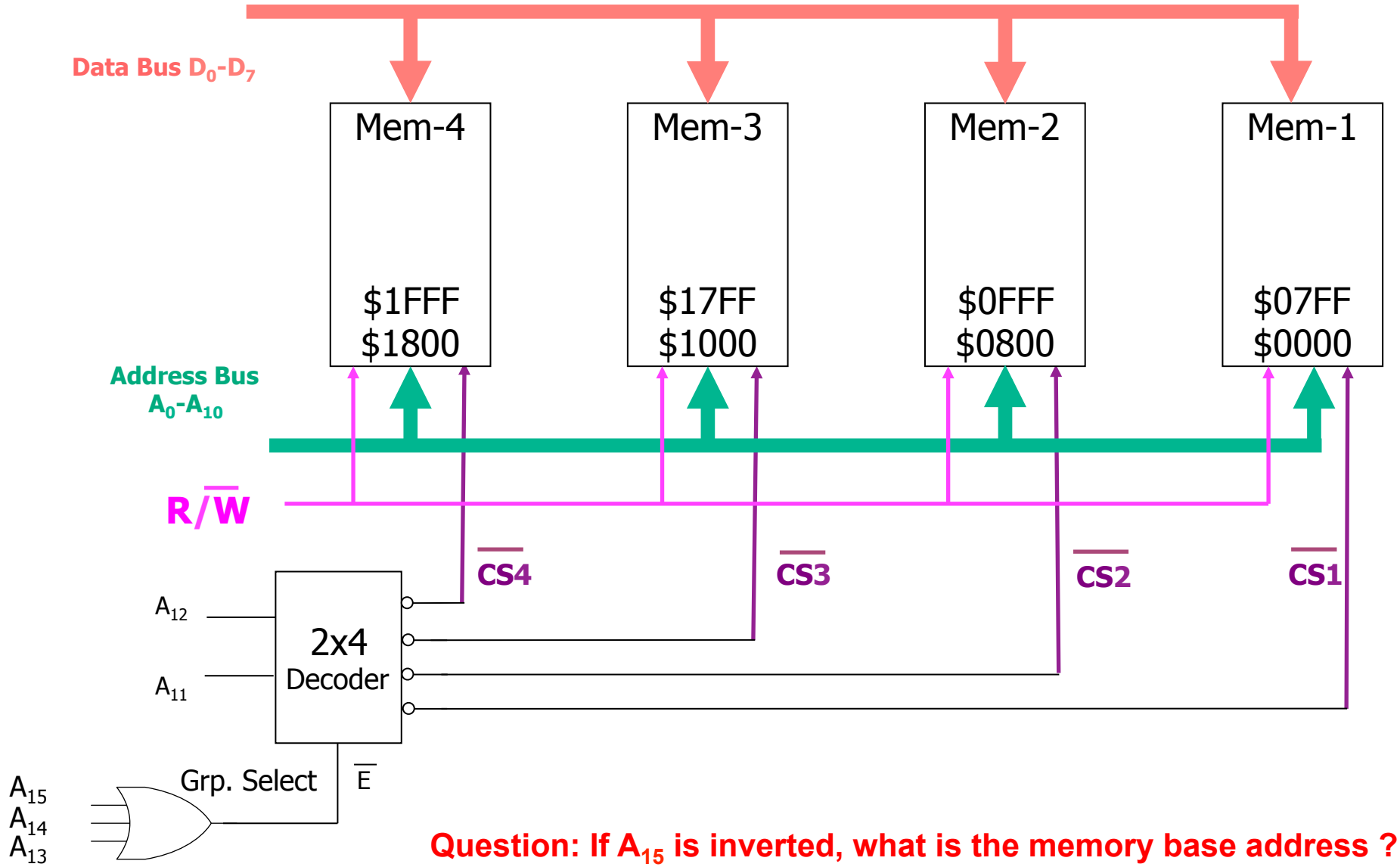
- A_{15}, A_{14}, A_{13} remain at low at all times. They can be used to form another Chip Select (Group Select)



- A_{12} and A_{11} can be used to select memory chips with 2x4 decoder

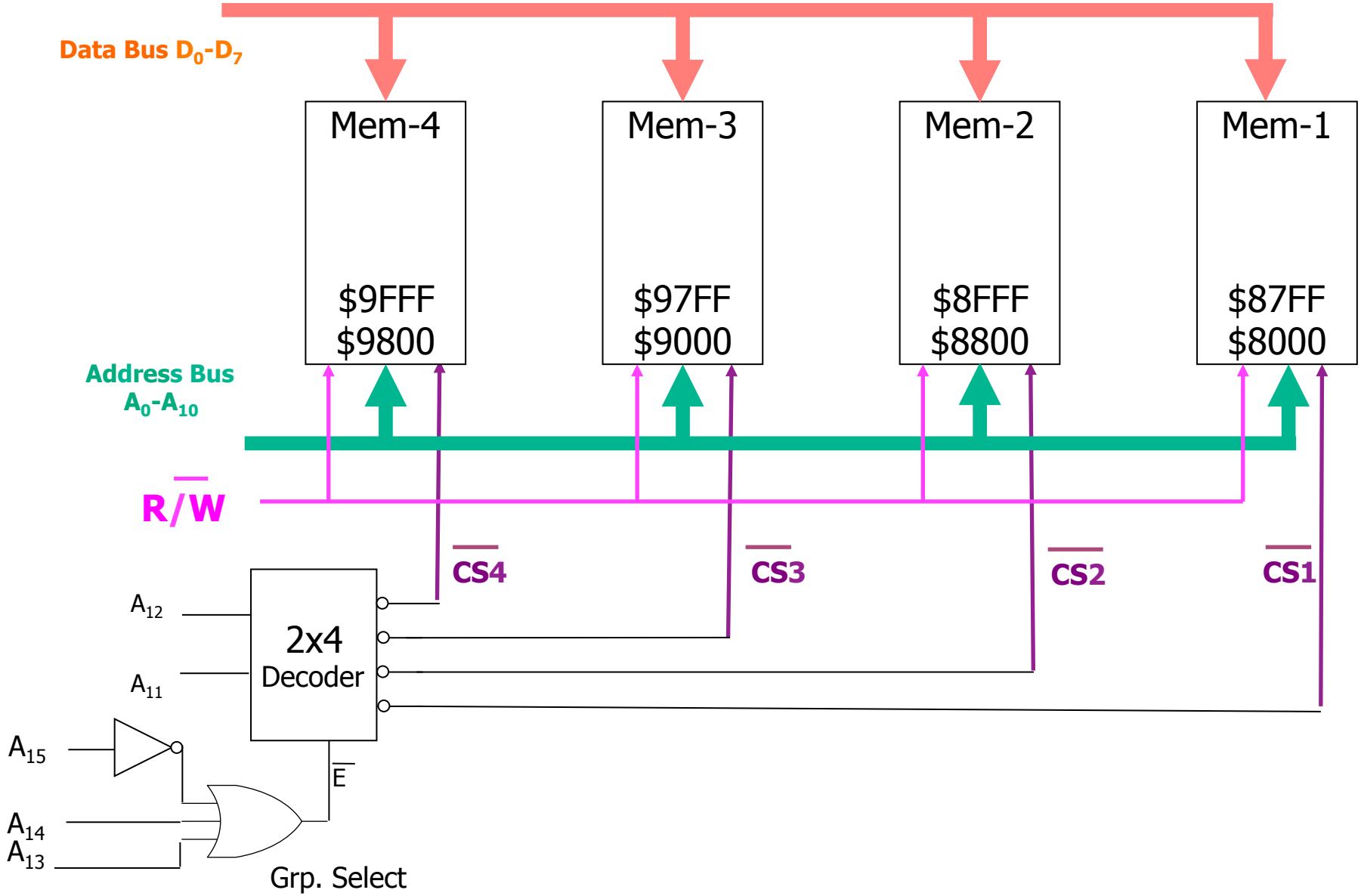


Memory Organization



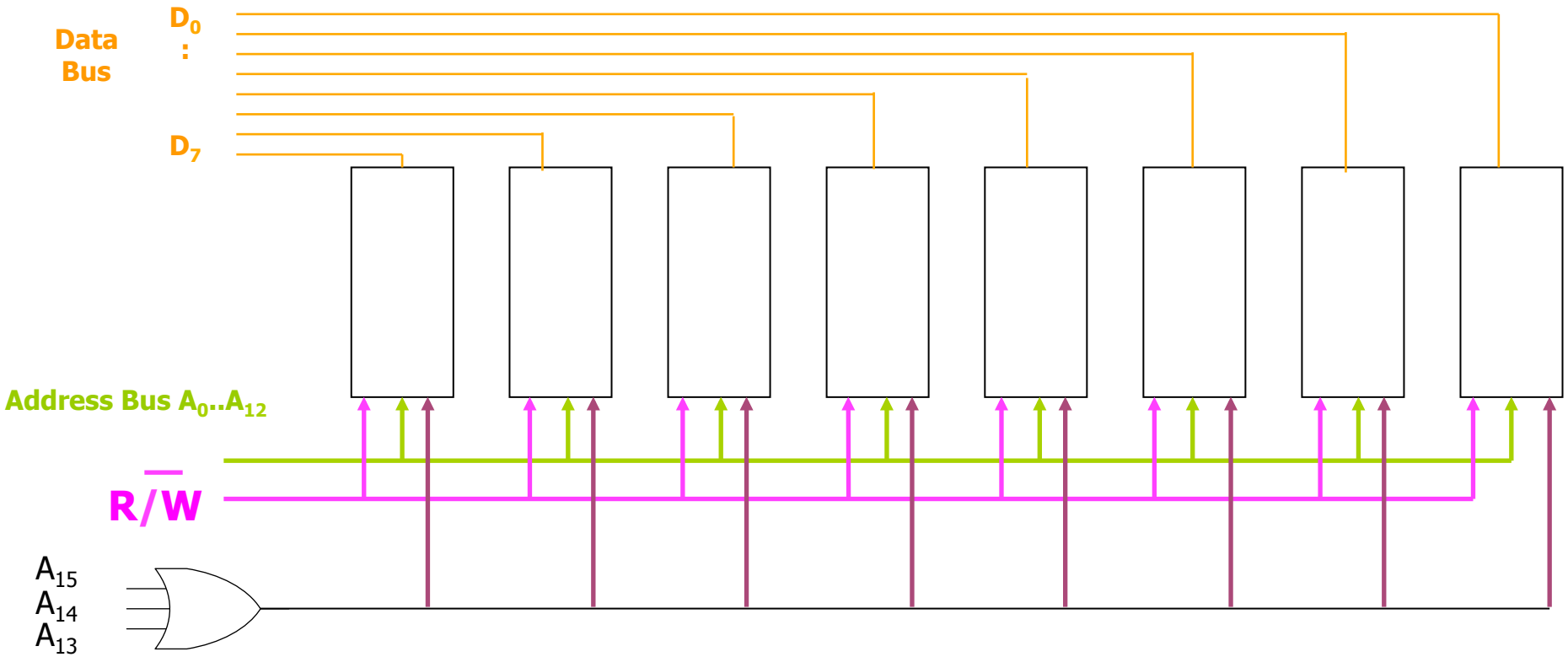
Question: If A_{15} is inverted, what is the memory base address ?

Memory Organization



Memory Organization

Example: Organize 8Kx1 memory chips to obtain 8Kx8 memory



References

- Memory Organization sldies prepared by
  - Dr. Şule Gündüz Öğüdücü
  - Dr. Erdem Matoğlu
  - Dr. Feza Buzluca
  - Dr. Bassel Soudan
  - Dr. Gökhan İnce