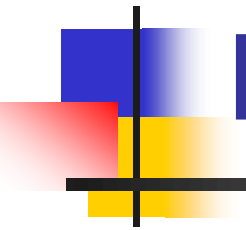
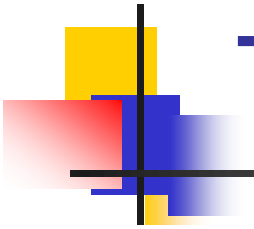


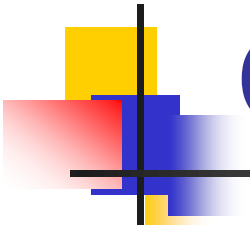
Microprocessor Systems

Dr. Gökhan İnce



Topics

- Computer Structure
- Memory Systems



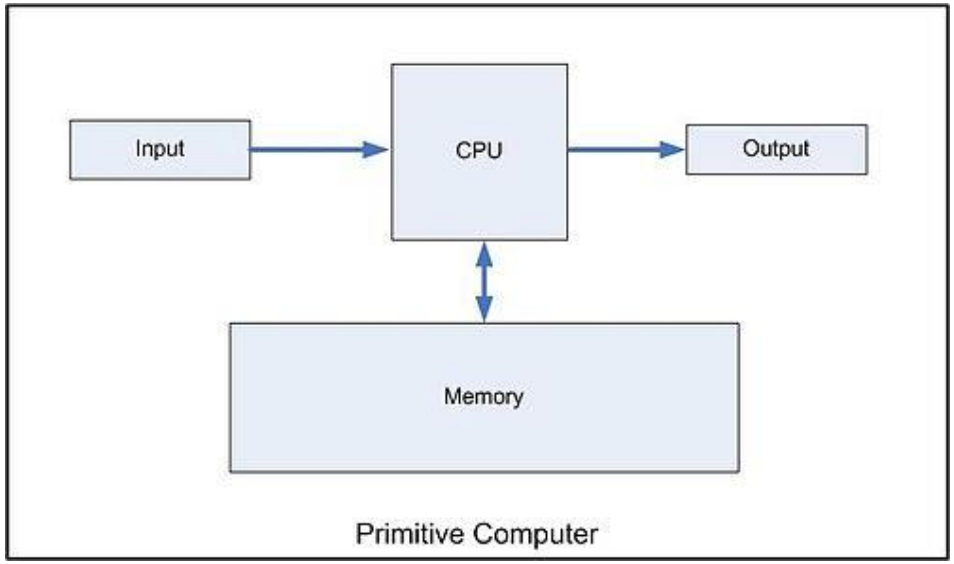
Computer Structure

- A computer is a programmable machine designed to automatically carry out a sequence of arithmetic or logical operations.

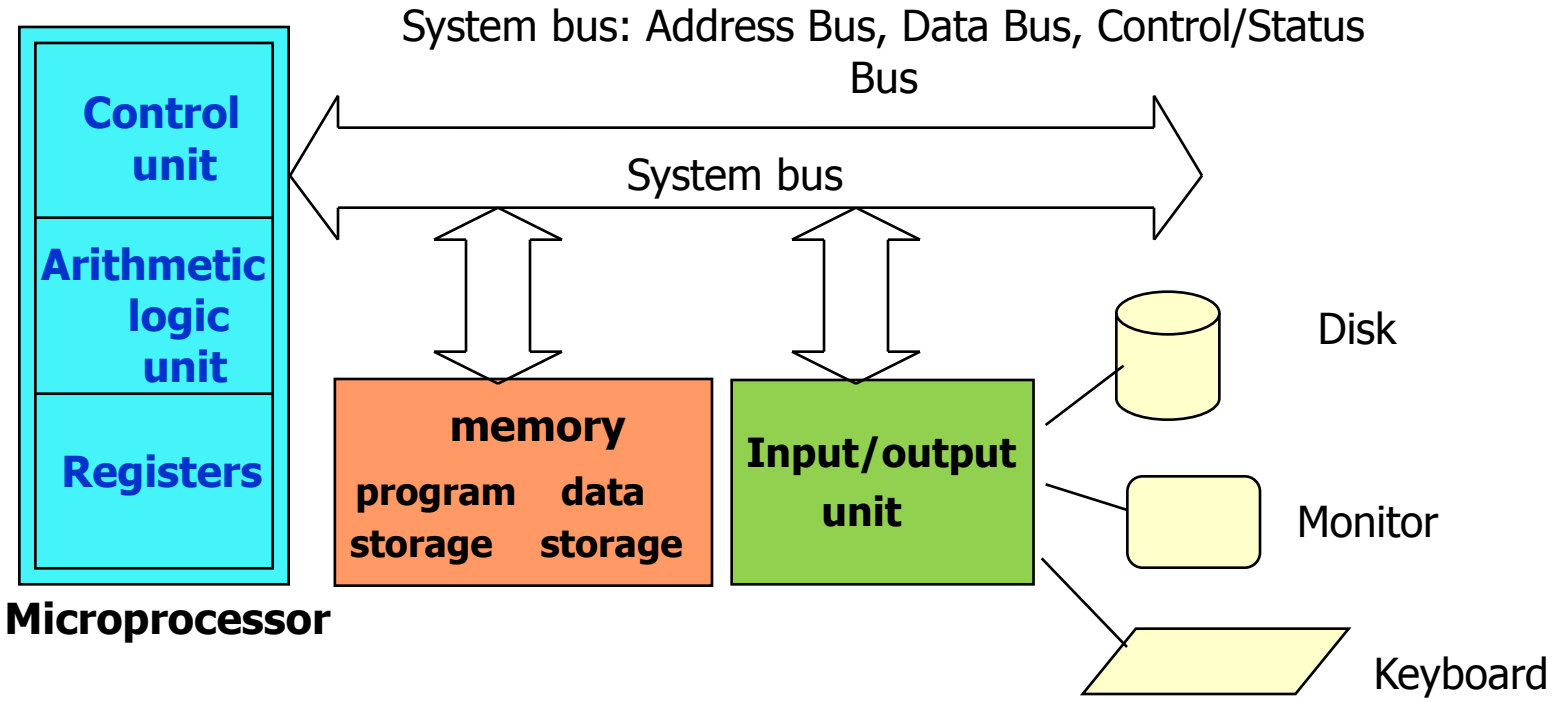
- Fundamental features of a computer
  - Memory capability
  - Compute capability
  - Decision capability
  - Input/Output capability

What is a computer ?

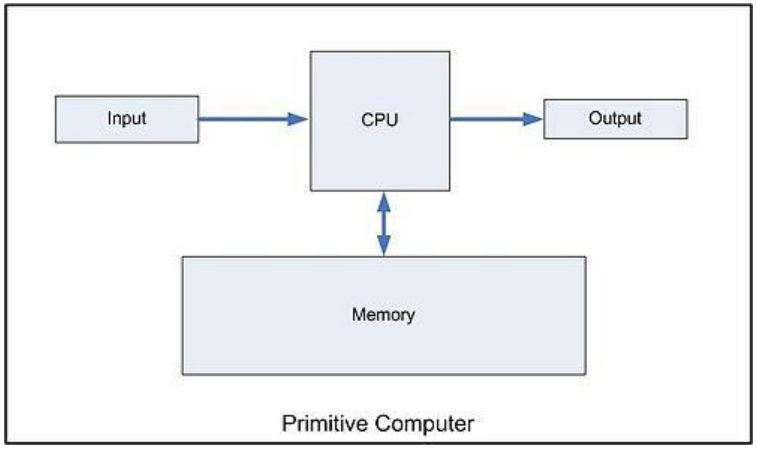
- It is programmable – where does the code reside ?
  - Memory
- Executes or processes the instructions of the program or code
  - Arithmetic and logic processor
- How does the code or data get in to the computer?
- How does it output the result?
  - Input/Output device(s)

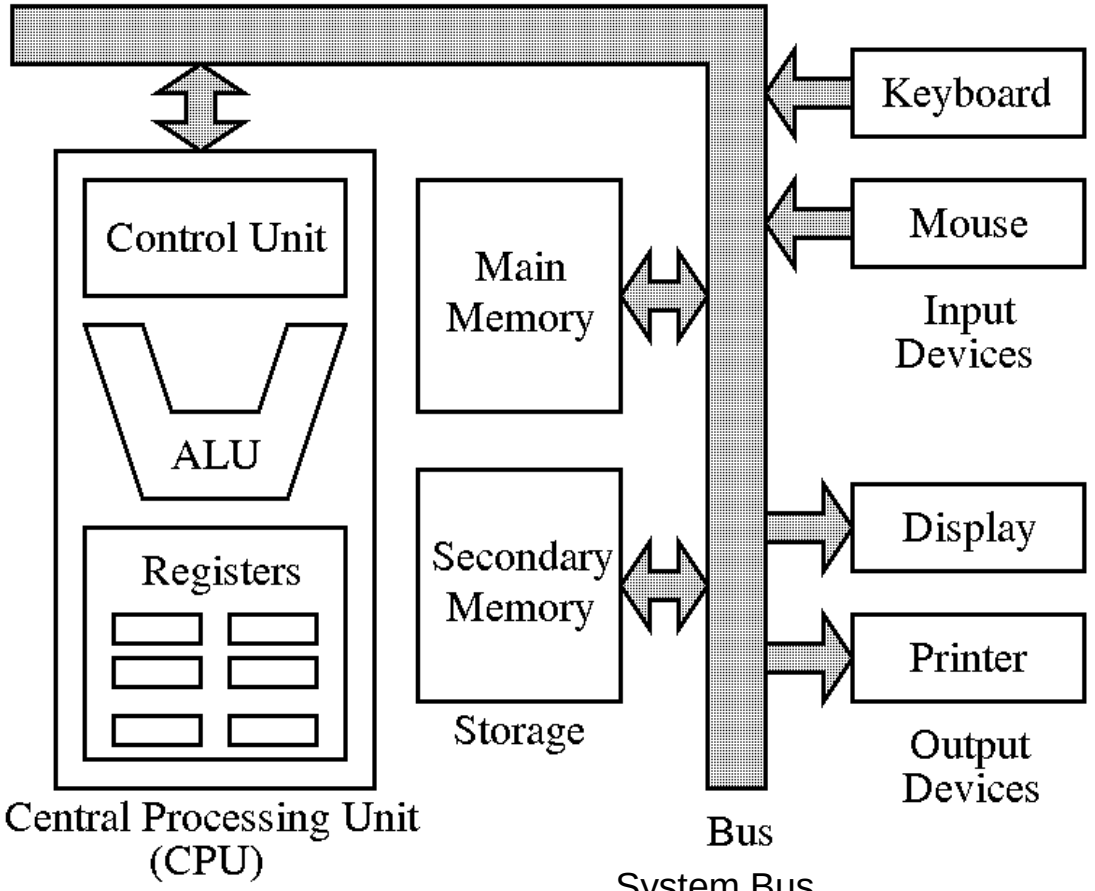


Computer Organization

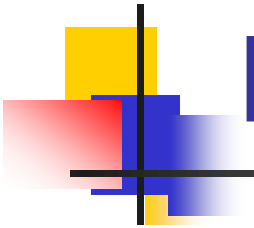


What is a computer ?





System Bus
(Common or Separate):
Address bus,
Data bus,
Control bus



Memory

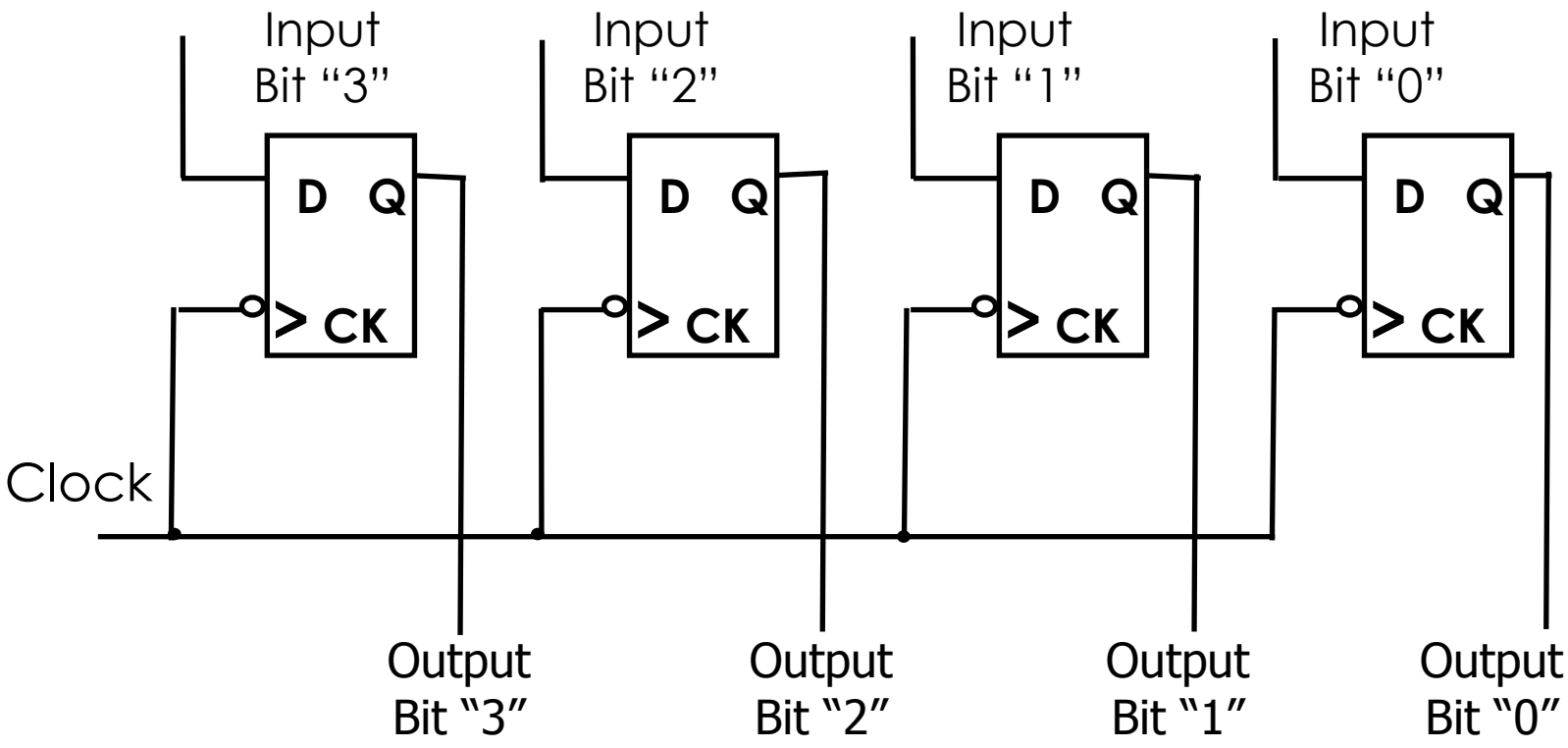
- Memory stores information such as **instructions** and **data** in binary format (0 and 1). It provides this information to the microprocessor whenever it is needed.

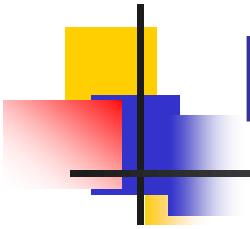
- Usually, there is a memory "sub-system" in a microprocessor-based system. This sub-system includes:
  - The registers inside the microprocessor
  - Read Only Memory (ROM)
    - used to store information that does not change.
  - Random Access Memory (RAM) (also known as Read/Write Memory)
    - used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

Registers

- A flip flop is a 1-bit memory device.
- A register is a collection of flip flops that are clocked as a unit.

D	Q(t+1)
0	0
1	1





Registers

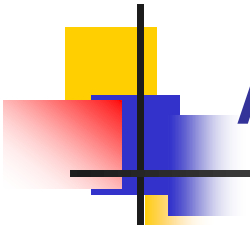
- Register is a group of n-flip-flops capable of storing n-bits of information
- Registers are accessed very fast within the CPU
- Different CPUs have different set and size of registers
- General-purpose registers are used to store data during the CPU execution process
- Special-purpose registers control or monitor various aspects of the microprocessor's function (MAR, IR...).

CPU Structure

- **Memory Address Register** (MAR) either stores the memory address from which data will be fetched to the CPU or the address to which data will be sent and stored.

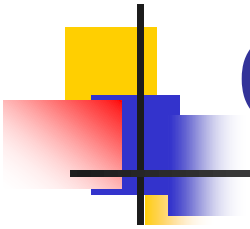
- **The Memory Data Register** (MDR) contains the data to be stored in the computer storage (e.g. RAM), or the data after a fetch from the computer storage.

- **Accumulator** (ACC) stores intermediate arithmetic and logic results.



Arithmetic and Logic Unit (ALU)

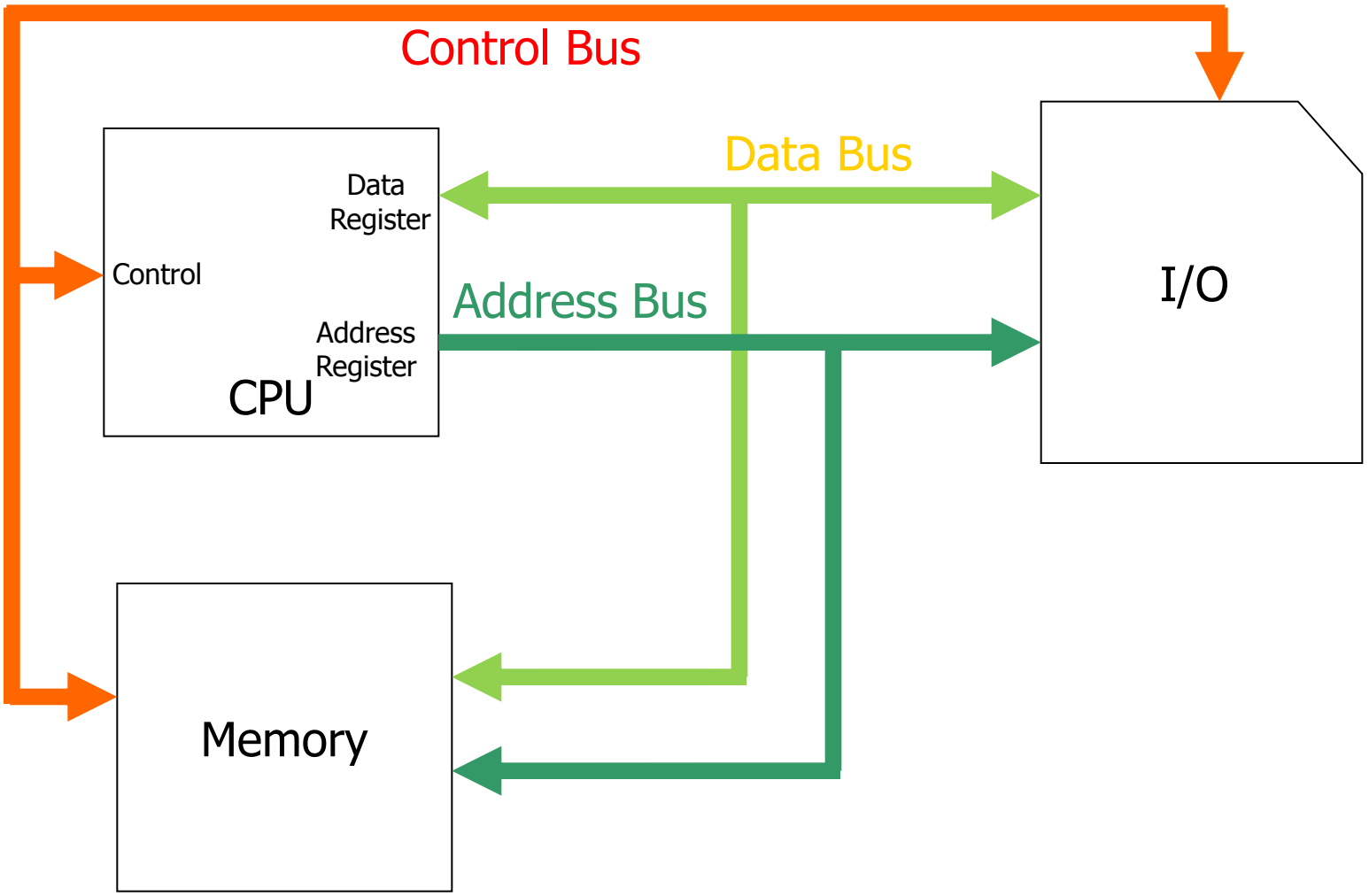
- **Arithmetic and Logic Unit** (ALU) performs arithmetic and logical operations
- The ALU receives data from main memory and/or the register file, performs a computation, and, if necessary, writes the result back to main memory or registers.



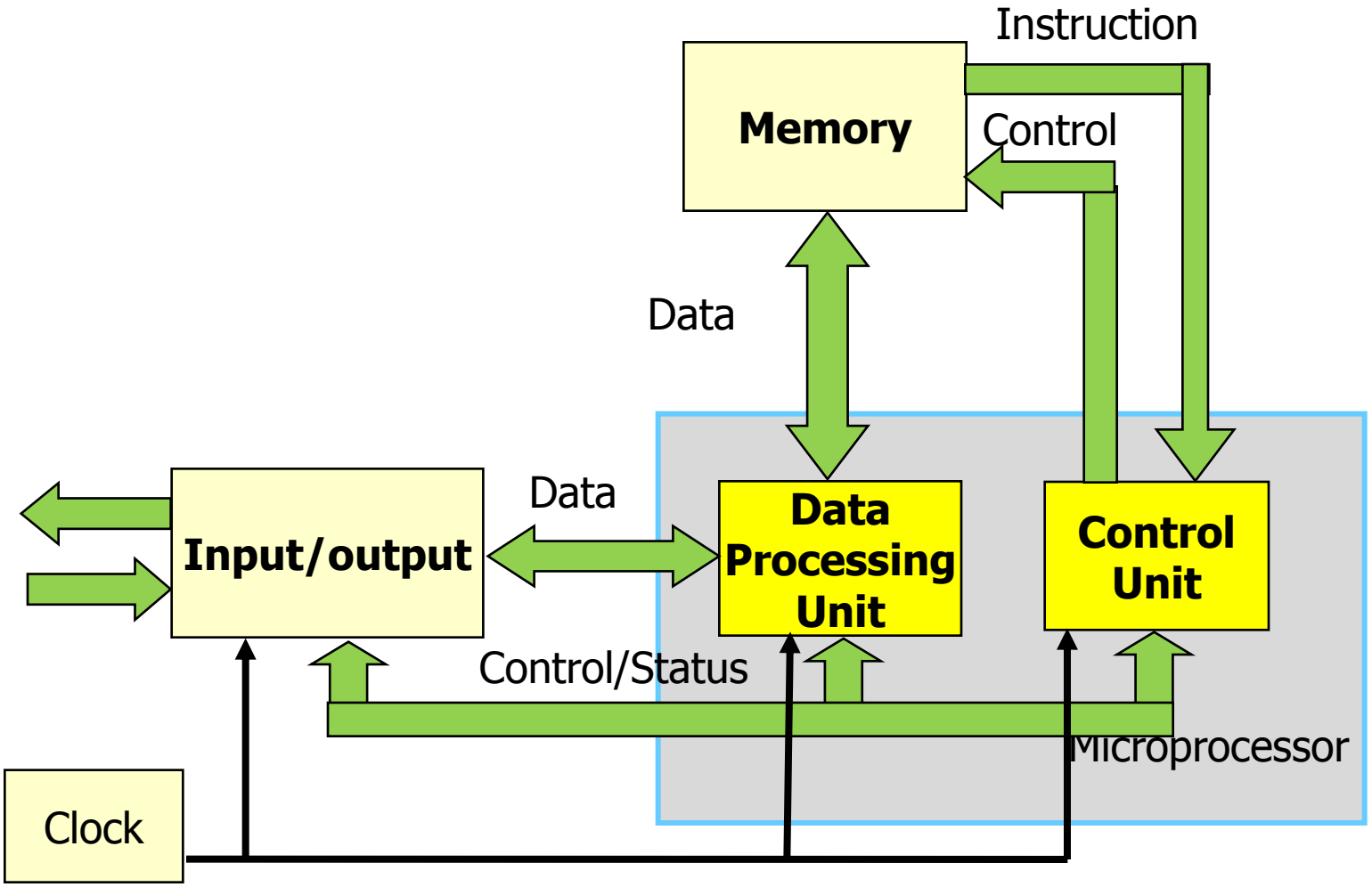
Control Unit

- **Control Unit** manages the computer operation within and outside the CPU
  - Sequencing and execution cycles of commands
  - Register operations
  - Bus regulation
  - Interrupt handling
  - System status
  - Memory management

Basic Computer Connections

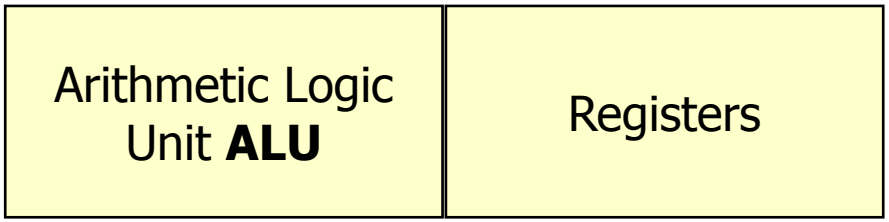


Stored Program Procedure



Stored Program Procedure

- The Clock: The stored program processor is a large synchronous sequential network (logic circuit) that requires a clock signal to synchronize all its elements.
  - The signal from the clock is a periodic pulse waveform.
- The Data processing Unit: The principal part of the data processing unit is a combinational logic circuit, called the ALU, that manipulates binary numbers.
  - Practical ALUs can perform dozens of different operations, but only one at time. Usually ALUs can add, subtract, complement bits, shift binary values and so on.
  - Registers hold the data numbers operated or produced.

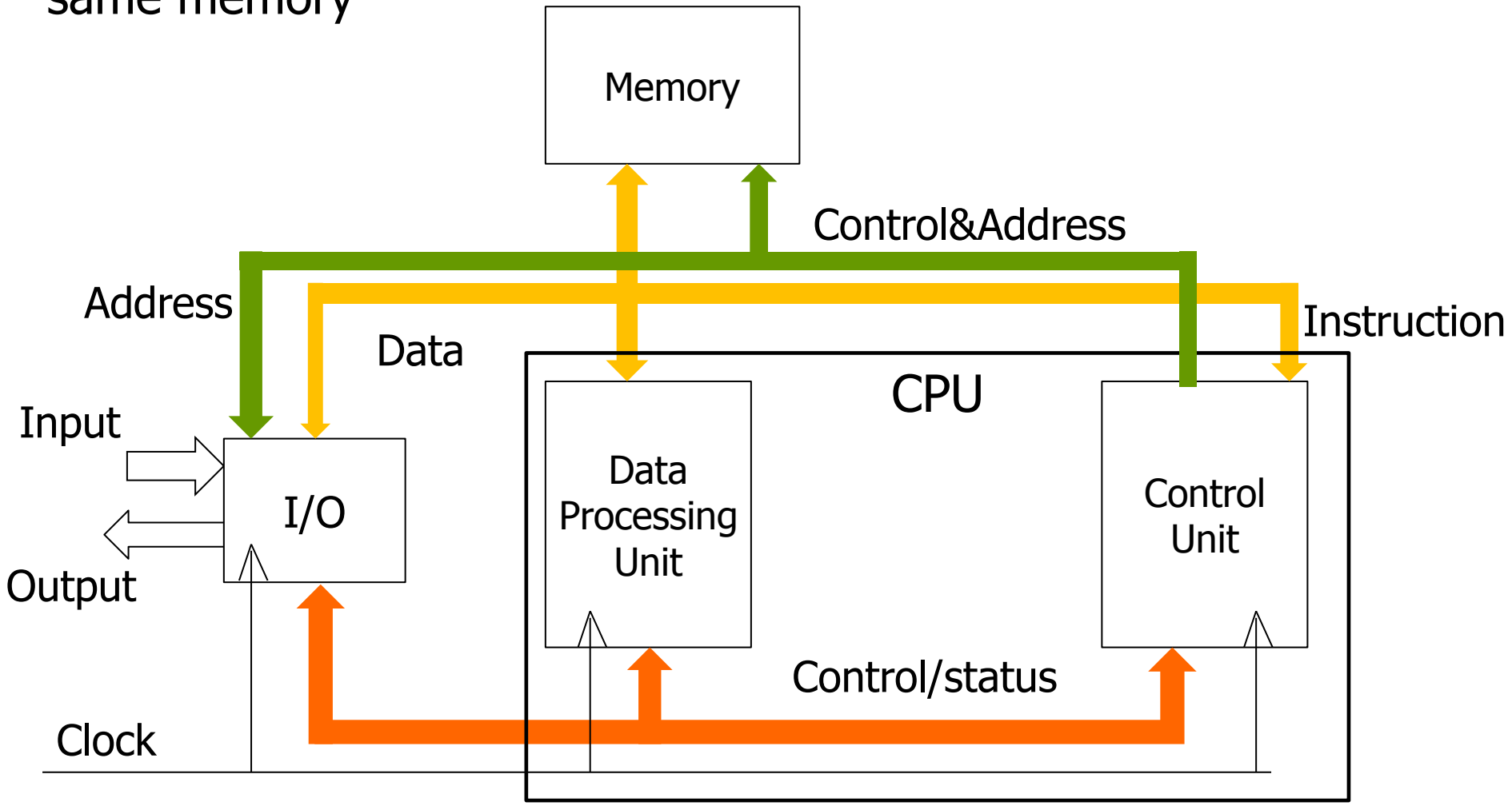


Stored Program Procedure

- The control unit is a synchronous sequential logic circuit that sends control signals to the data processing unit, memory and other parts of the system.
  - The signals from the control unit tells the data processing unit to manipulate data according to the algorithm built into the sequential logic circuit.
  - The control unit is instruction controlled; therefore it can do more than one algorithm based on its design. Typical control units recognize several hundred different instruction codes.
- The Memory: The memory holds instruction code numbers and data numbers
- The I/O Unit: The input/output unit includes any hardware that allows data transfer between the CPU and the real world.

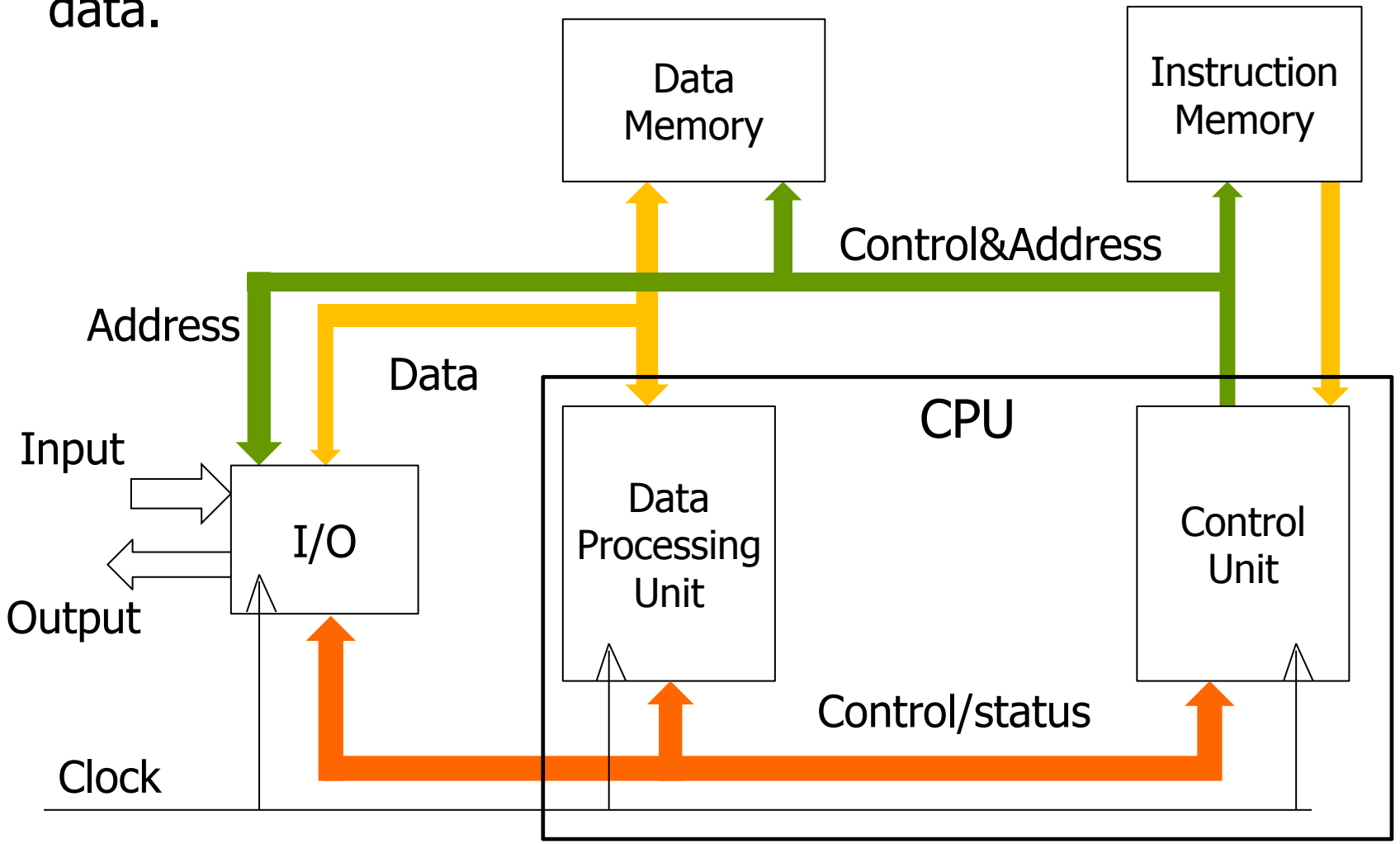
Von Neumann Architecture

Von Neumann Architecture: Instruction and data are in the same memory



Harvard Architecture

Harvard Architecture: There are two memories: Instruction and data.



Operation of Computer

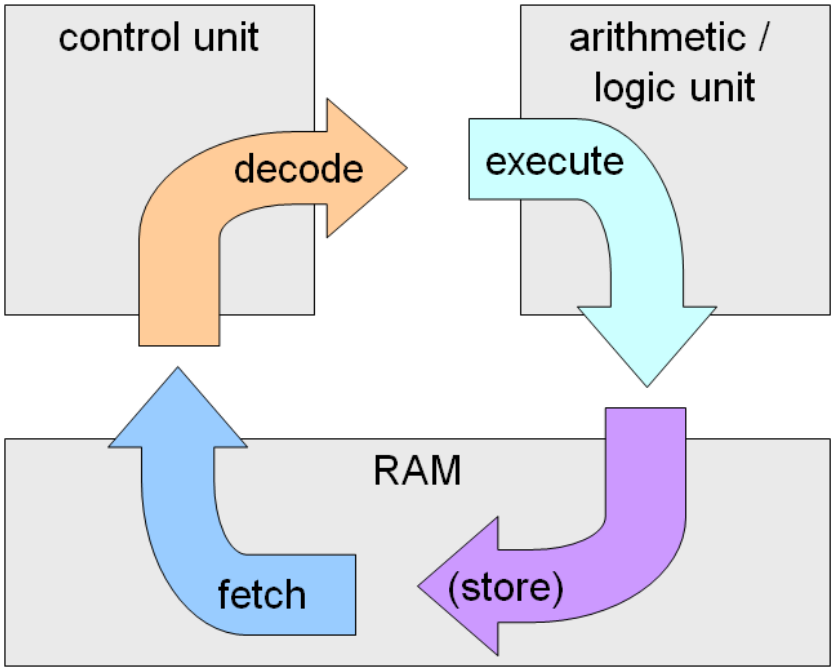
- Example: The X and Y numbers will be written on memory addresses $3000 and $3001. The sum of X and Y will be stored on address $3002

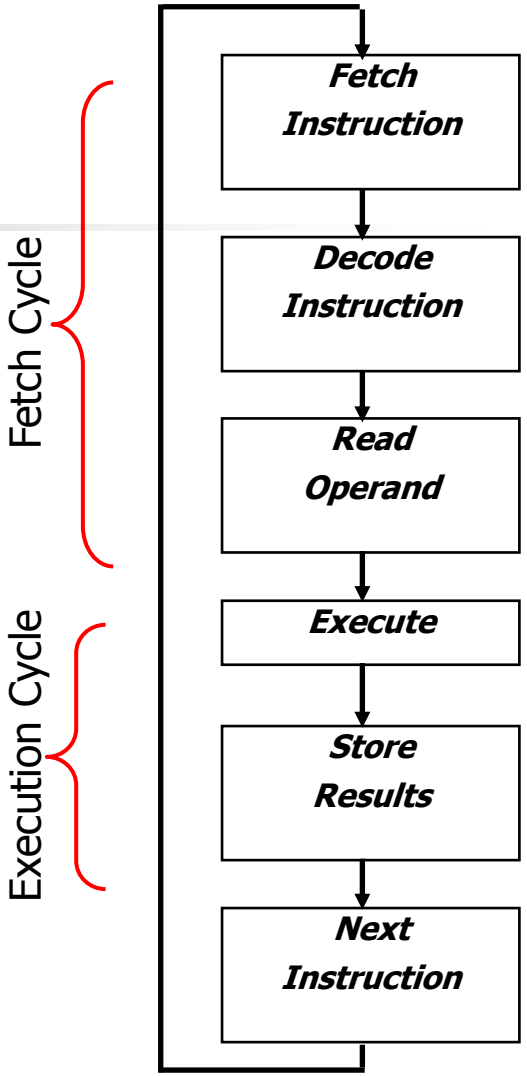
Memory Address	Instruction
1001	Load X to ACC (accumulator)
1002	Store contents of ACC to address $3000
1003	Load Y to ACC
1004	Store contents of ACC to address $3001
1005	Add contents of $3000 to ACC
1006	Store contents of ACC to address $3002

$3000	X	Results after execution
$3001	Y	
$3002	X+Y	

Instruction Cycle

Fetch-Decode-Execute

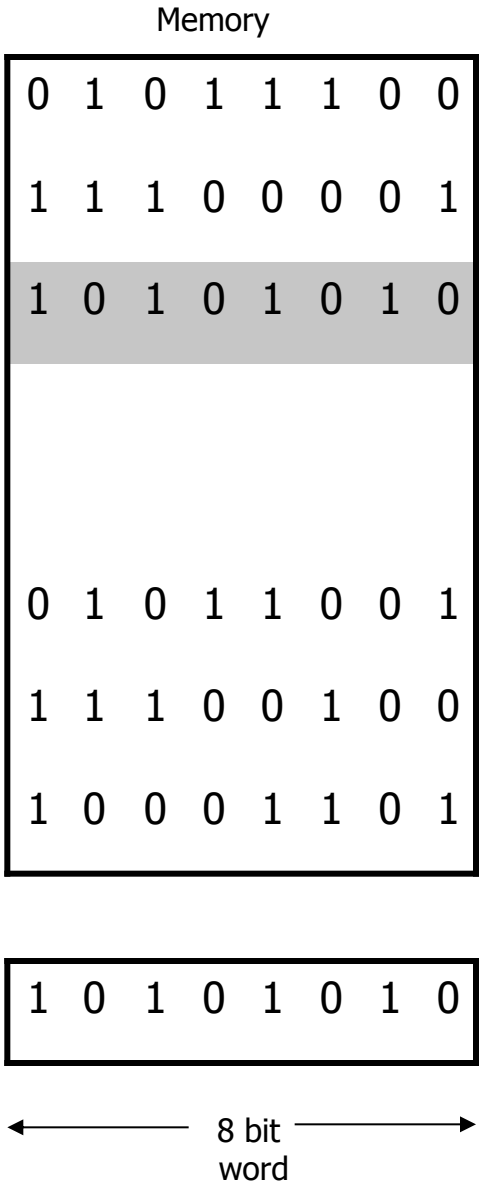




- Two-cycle process because both instructions and data are in memory
- Fetch
  - Decode or find instruction, load from memory into register and signal ALU
- Execute
  - Performs operation that instruction requires
  - Move/transform data

Instructions Formats

- Instructions are stored in the memory
- They are executed in a sequence
- They instruct the computer what to do
- The computer fetches the next instruction and decodes it (See Instruction Cycle).
- Each CPU has different set of instructions
- Each CPU has different template of instructions



Instruction Templates

- 3+1 Address instruction template (Example: ADD X,Y,Z,N)

Opcode	1.Operand address	2.Operand address	Result address	Address of next instruction

- 3 Address instruction template (Example: ADD X,Y,Z)

Opcode	1.Operand address	2.Operand address	Result address

- 2 Address instruction template (Example: ADD X,Y)

Opcode	1.Operand address	2.Operand address

- 1 Address instruction template (Example: ADD A,X)

Opcode	Register	Operand address

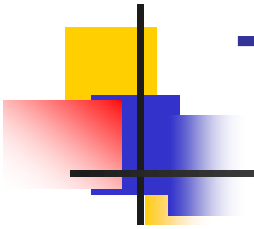
- 0 Address instruction template (Example: PSH A, PUL A)

Opcode	Register

Long instructions can take up multiple words in memory

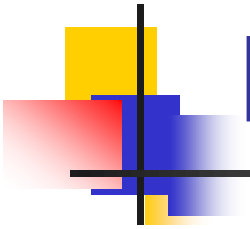
Example: 8-bit words in memory, instruction template can be 40bits, takes up 5 words in memory

0, 1, and 2 address instructions are commonly used in modern computers



Topics

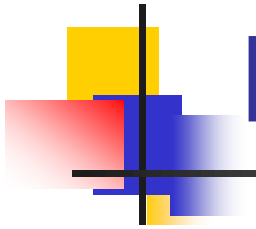
- Computer Structure
- Memory Systems



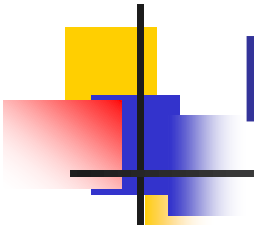
Memory

- 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



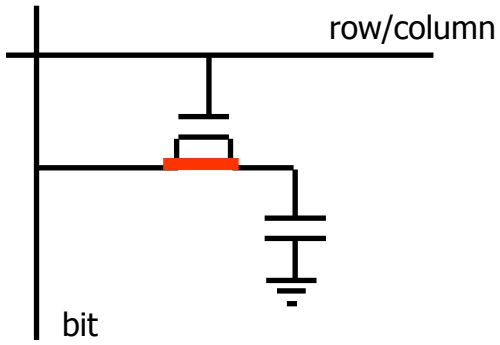
Non-volatile Memory

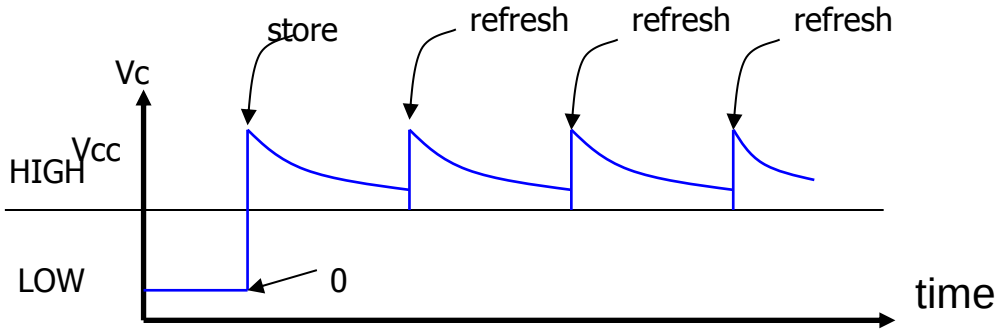
- Erasable programmable ROM (EPROM)
  - electrically programmable many times
  - erased by ultraviolet light (through a window)
  - erasable in bulk (whole chip in one erasure operation)
- Electrically erasable programmable ROM (EEPROM)
  - electrically programmable many times
  - electrically erasable 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 and write, cheap

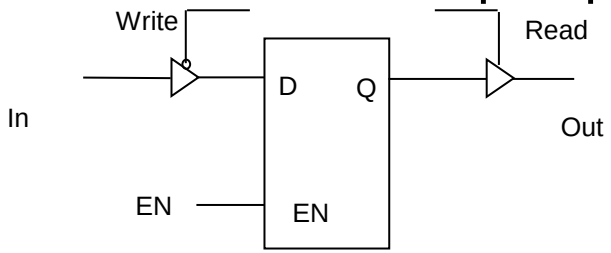




- SRAM : Static Random Access Memory

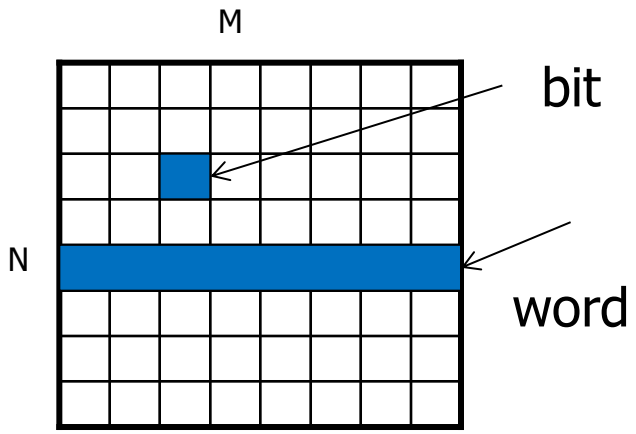
More power consumption than DRAM, each bit is stored in a flip-flop, more expensive, read/write

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



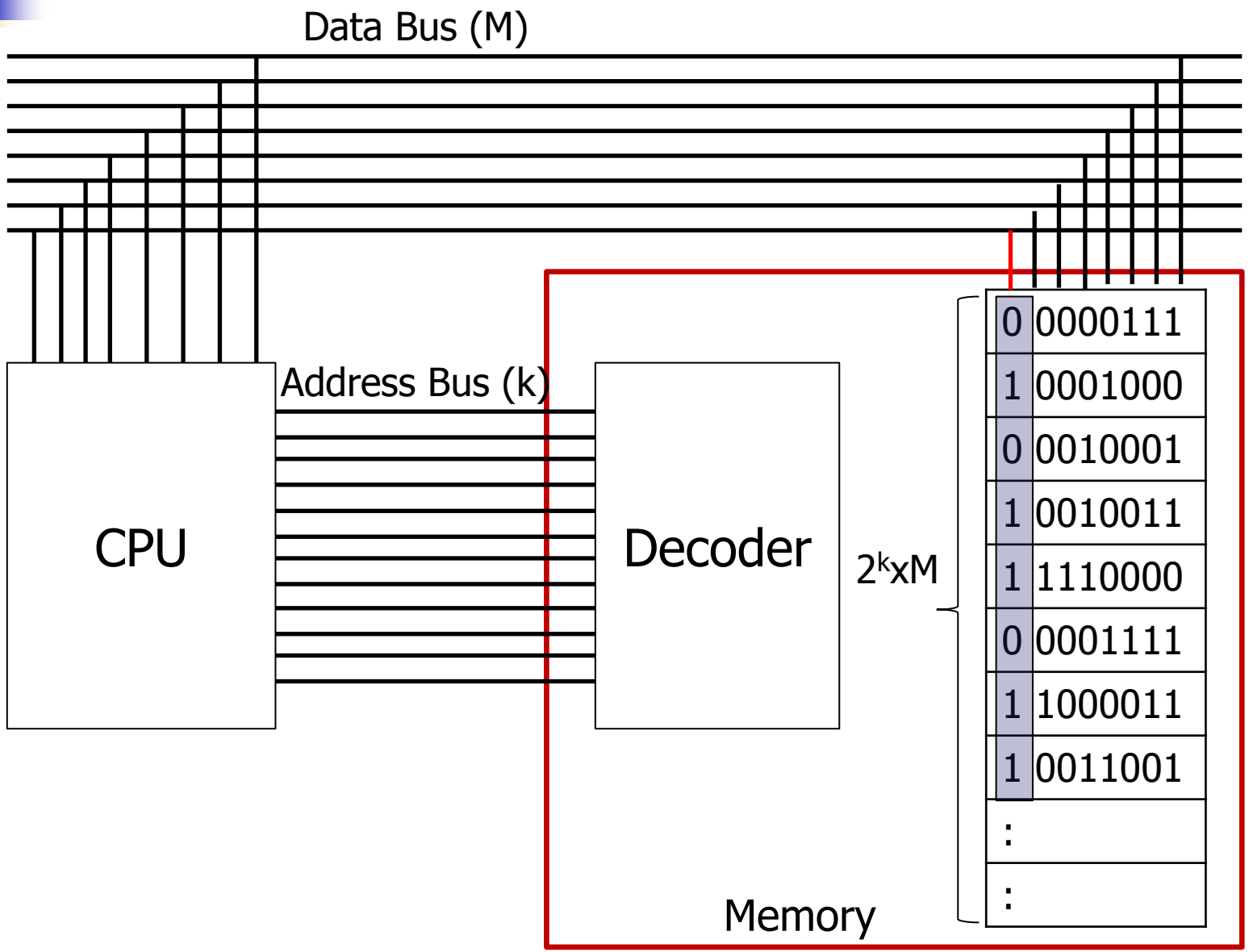
Memory Units

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

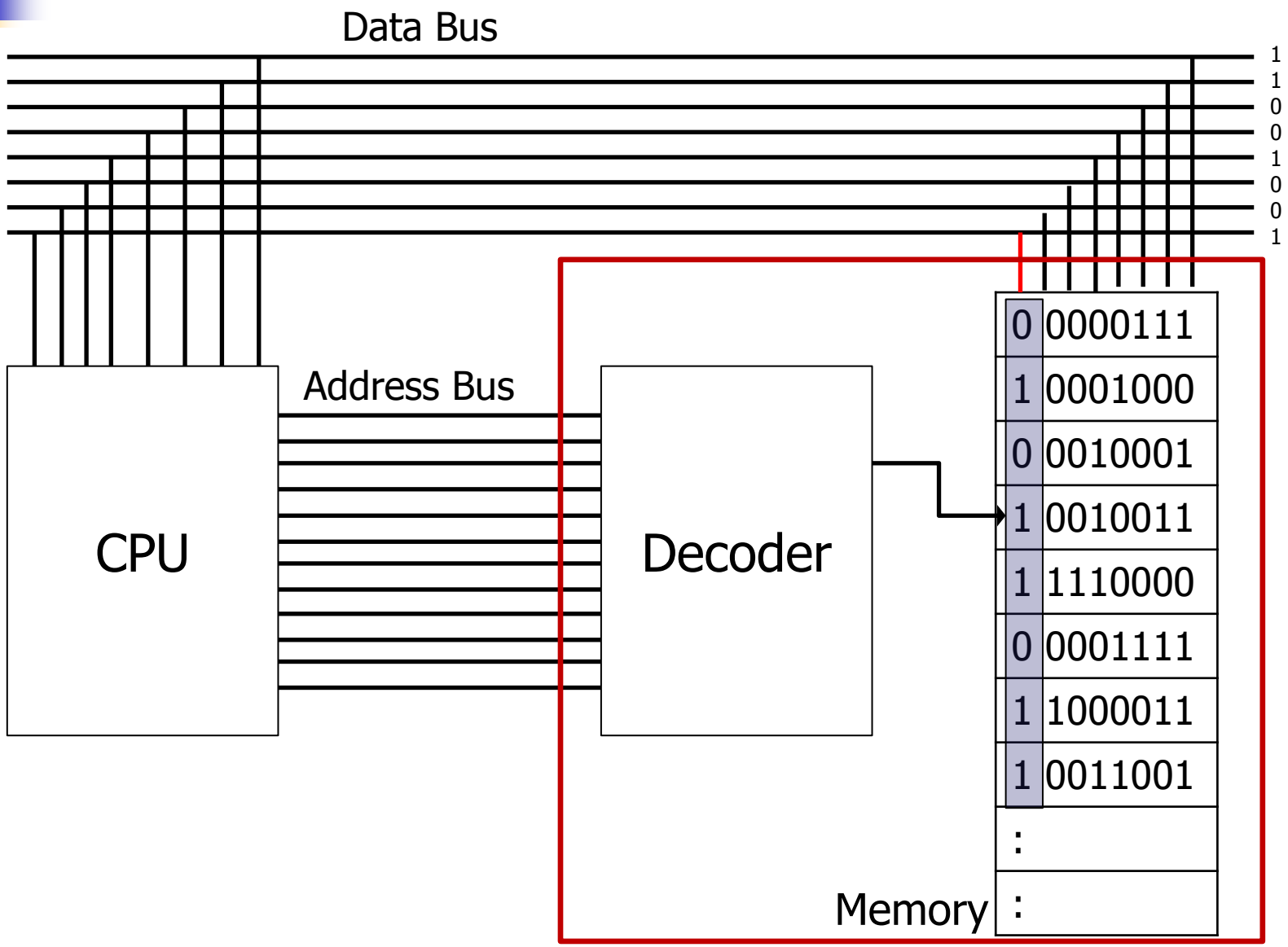


- Common widths: Byte (8 bits), Short (16 bits), Int (32 bits)

CPU-Memory Connection



CPU-Memory Connection



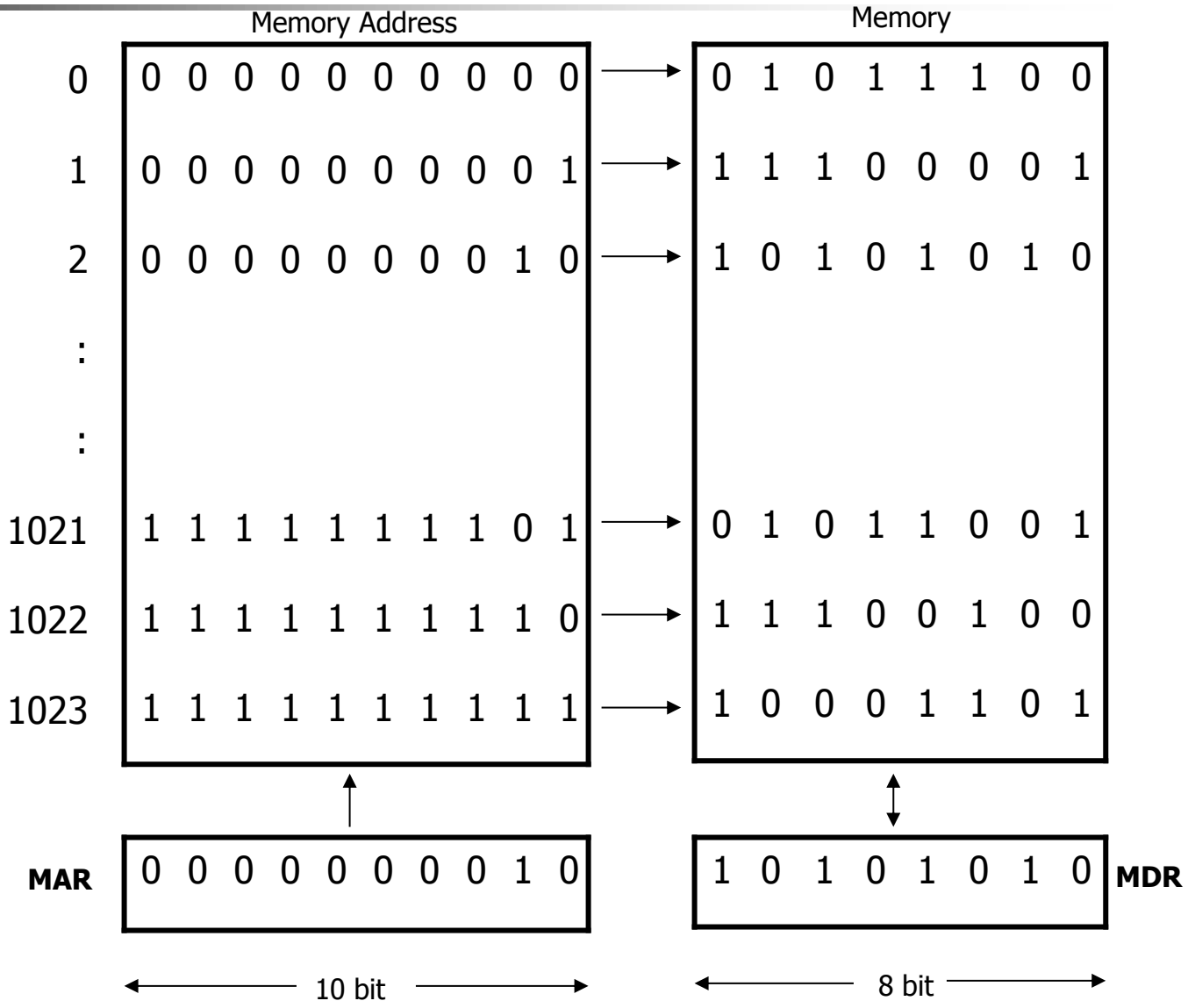
Example

Each word is identified with the address

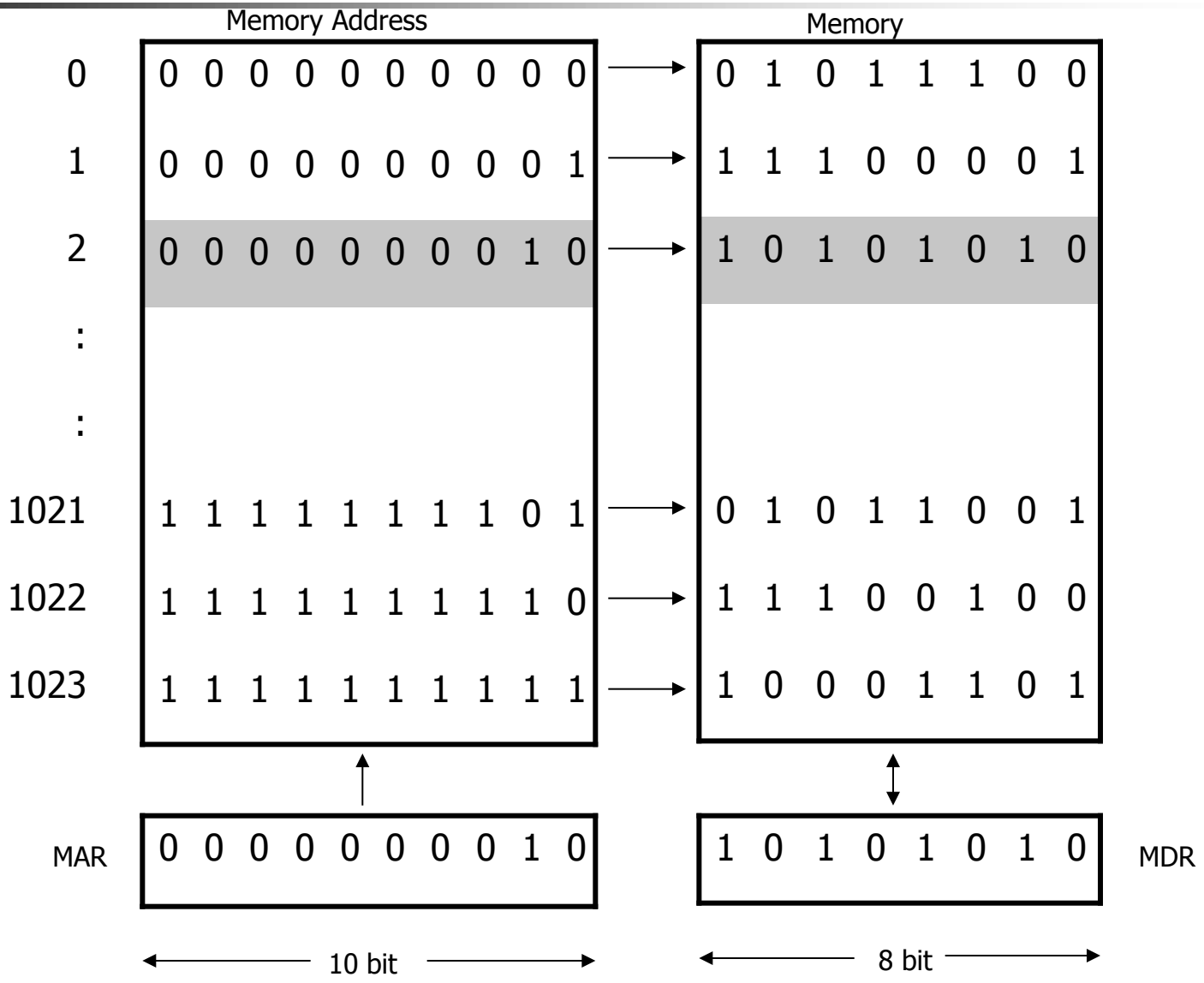
2^10=1024 (1K) words

Each word is 8-bit wide

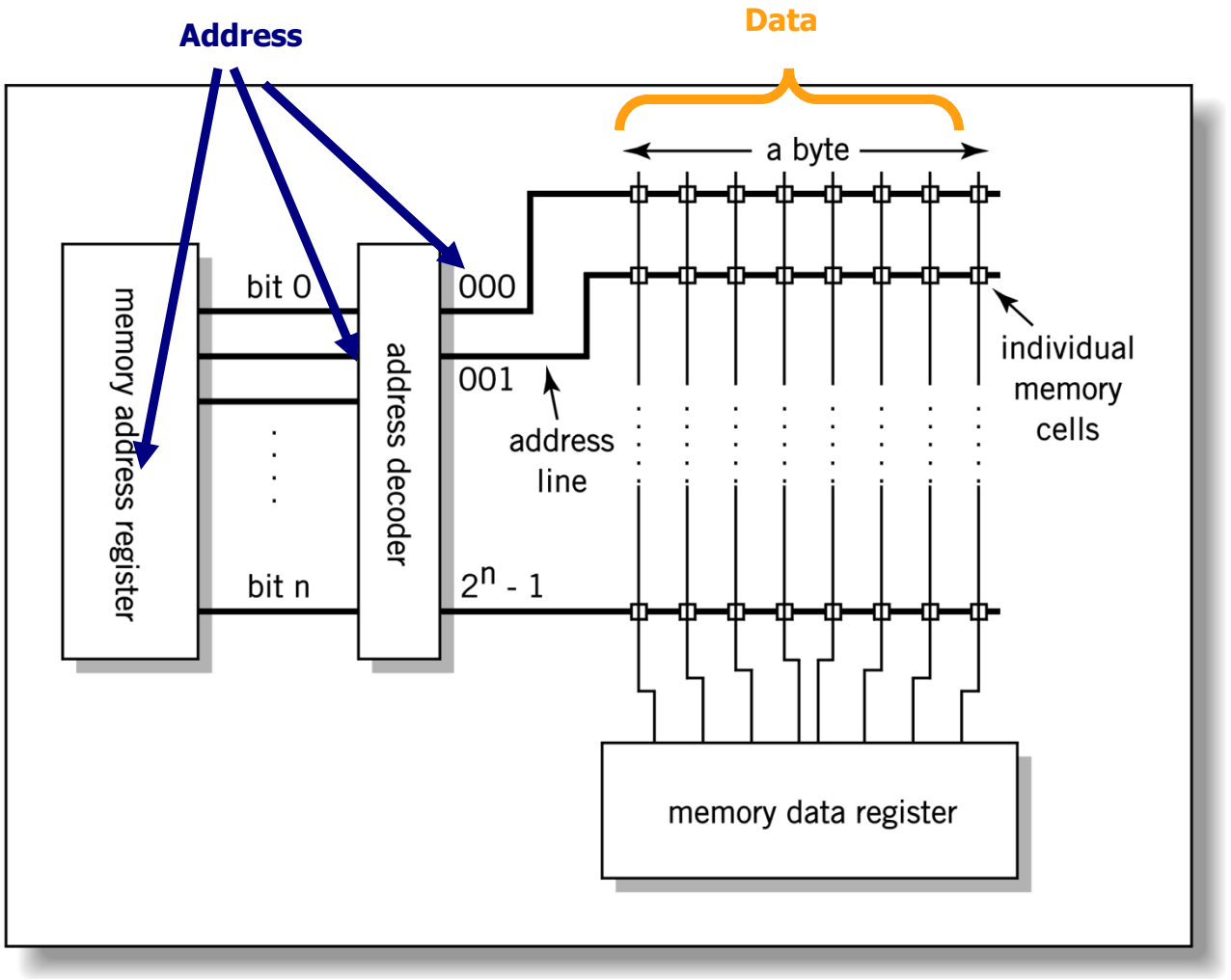
1KByte memory



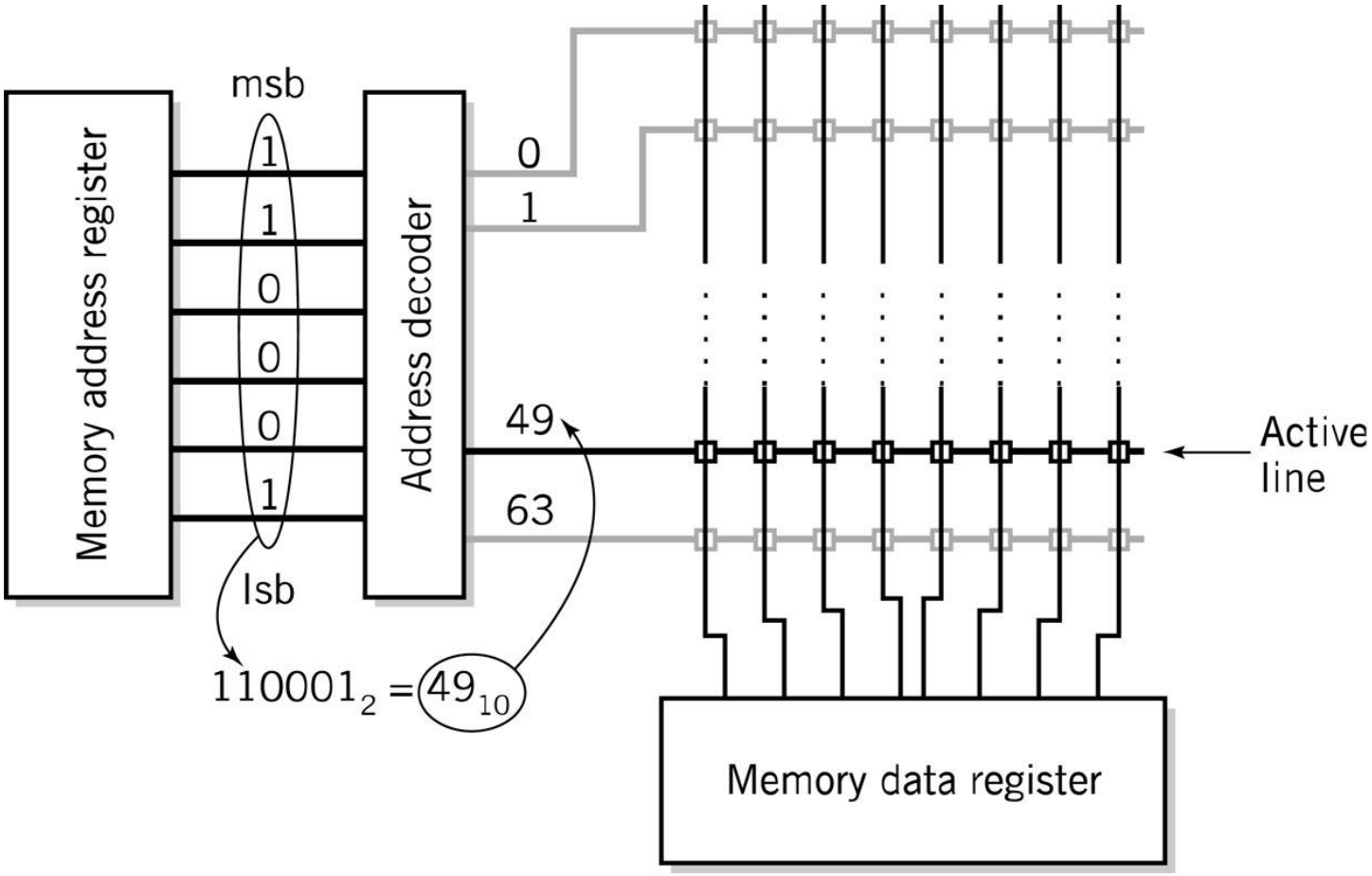
Example


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
MAR
0 0 0 0 0 0 0 0 1 0
1 0 1 0 1 0 1 0
MDR
10 bit
8 bit

Relationship between MAR, MDR and Memory

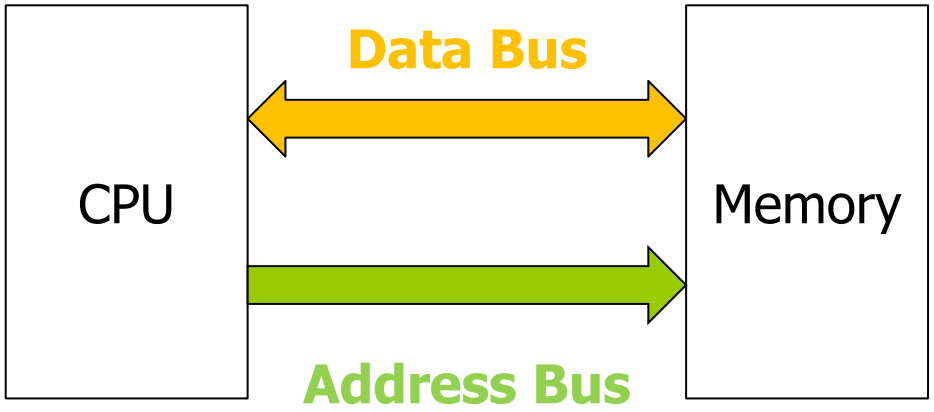


MAR-MDR Example



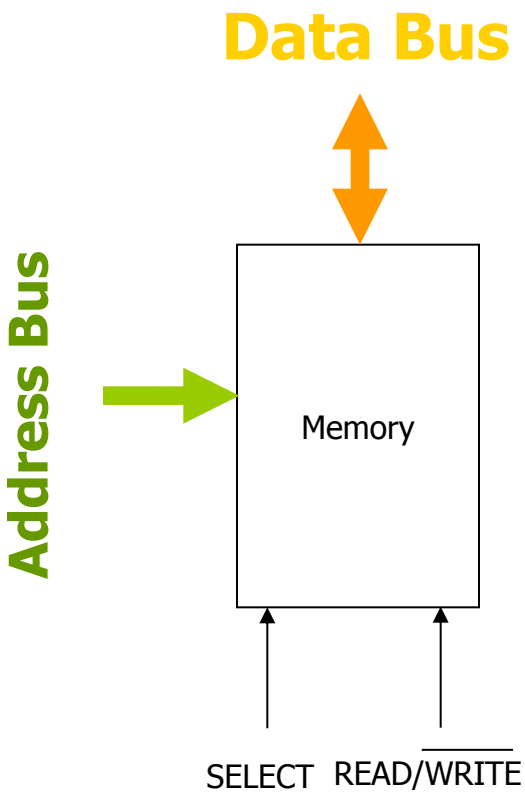
Memory Access

- Each memory location has a unique address
- Address from an instruction is copied to the MAR which finds the location in memory
- CPU determines if it is a store or retrieval
- Transfer takes place between the MDR and memory
- MDR is a two way register

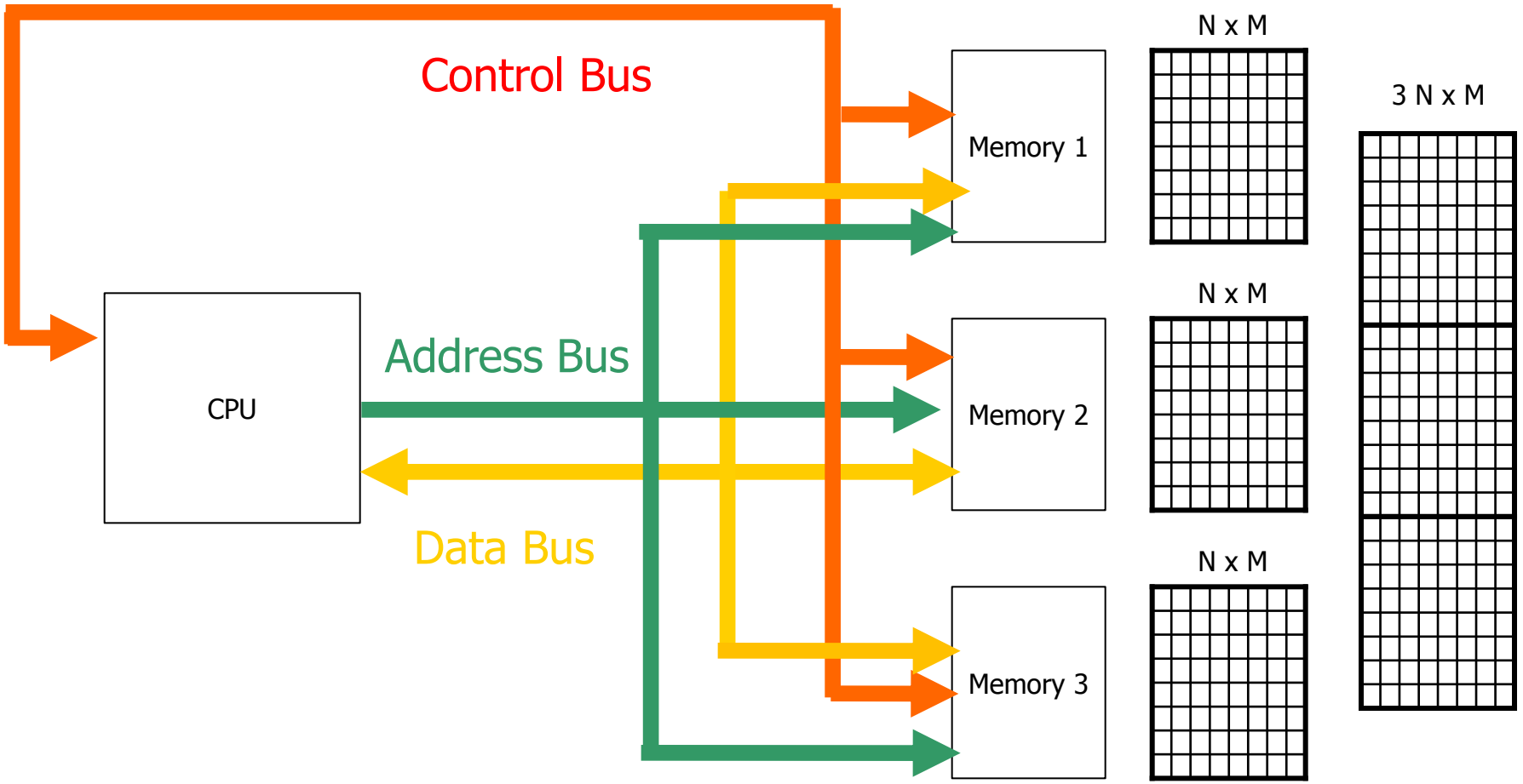


Memory Access

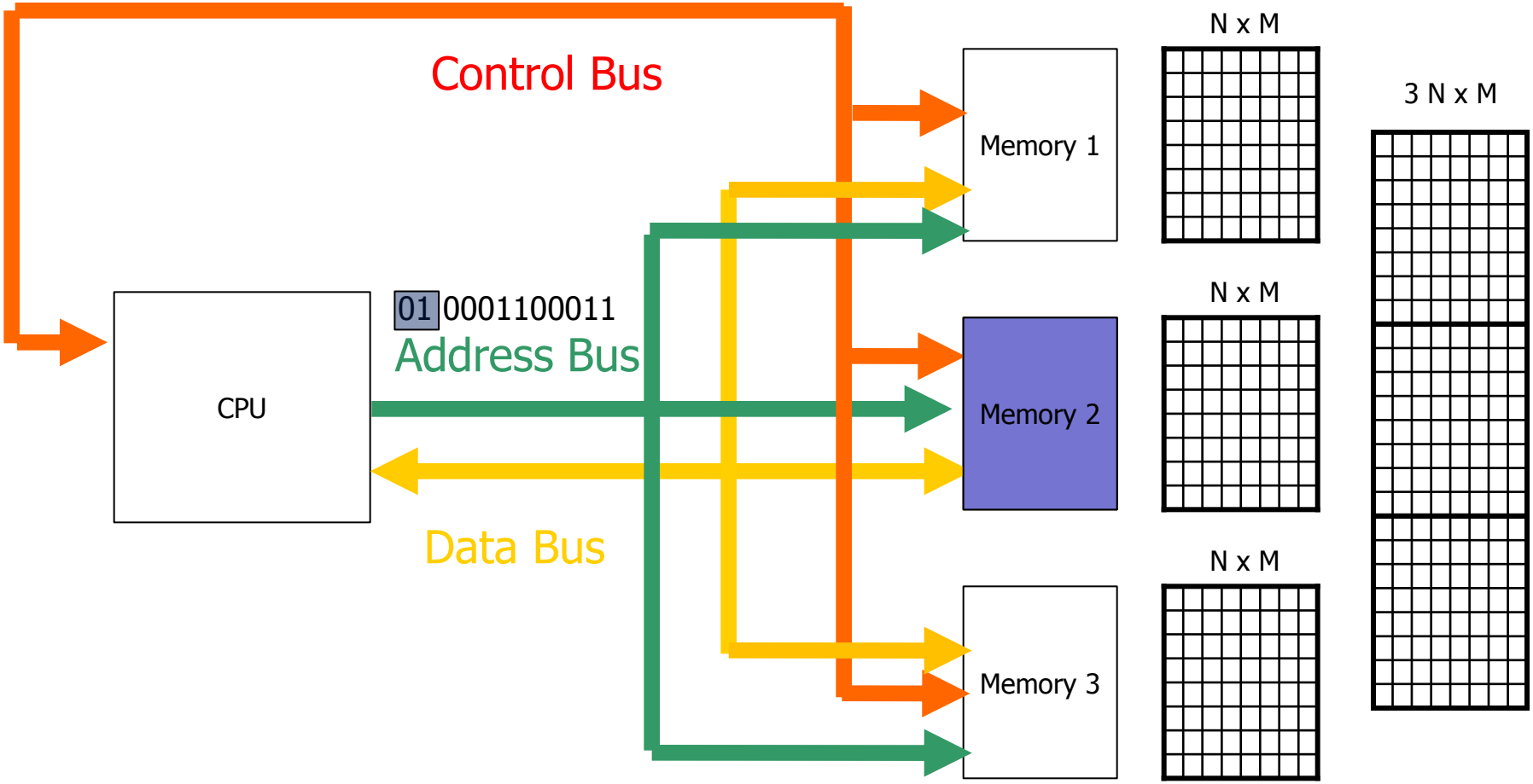
- For the microprocessor to access (Read or Write) information in memory (RAM or ROM), it needs to do the following:
  - Select the right memory chip (using part of the address bus).
  - Identify the memory location (using the rest of the address bus).
  - Access the data (using the data bus).



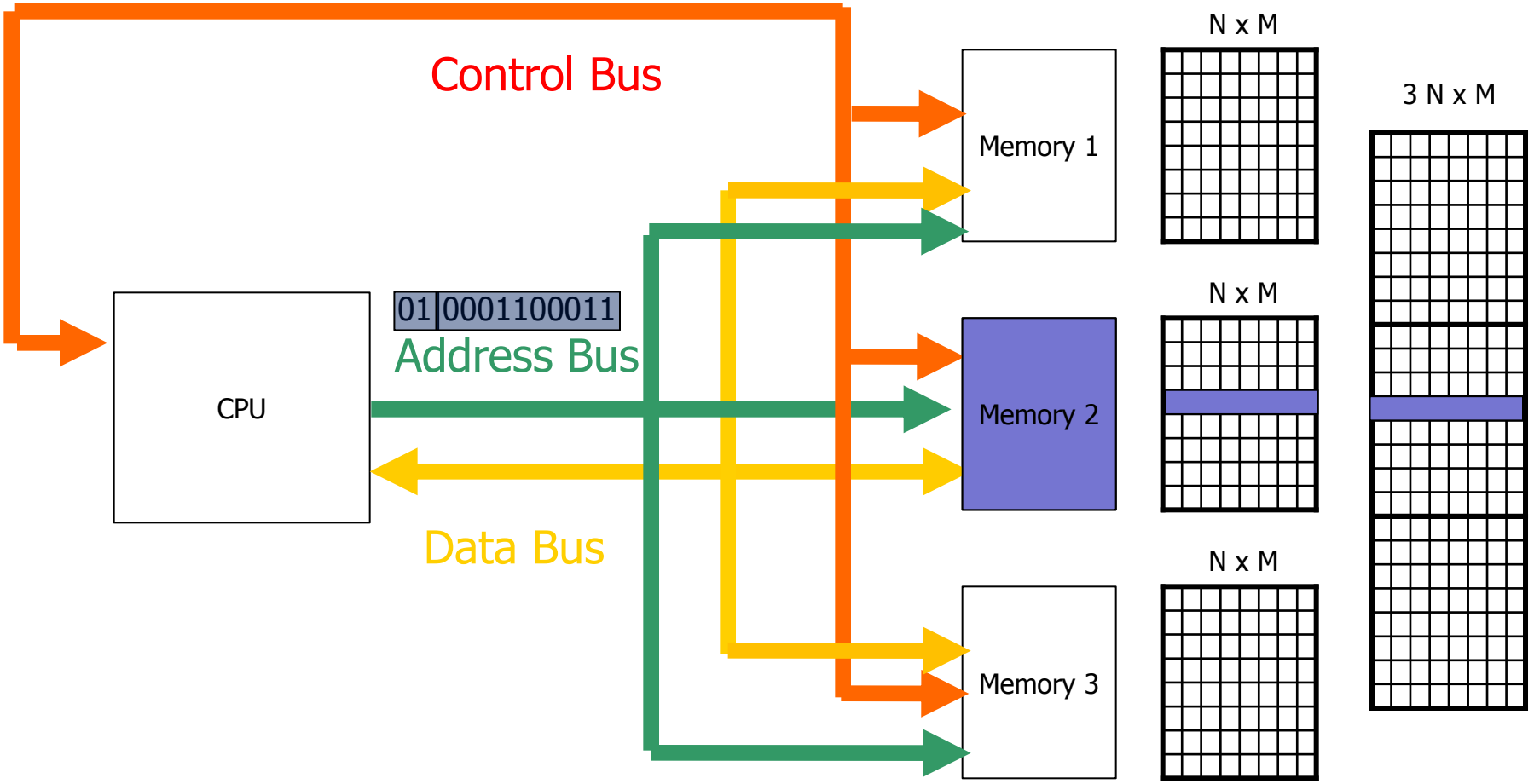
Memory Access



Memory Access

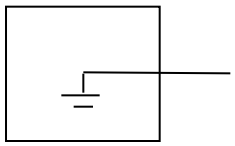


Memory Access

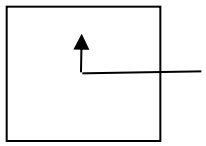


The Tri-State Buffers

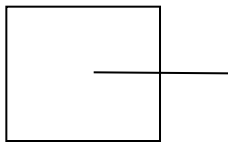
- An important circuit element that is used extensively in memory.
- This buffer is a logic circuit that has three states:
  - Logic 0, logic1, and high impedance.
  - When this circuit is in high impedance mode, it looks as if it is disconnected from the output completely.



The Output is Low



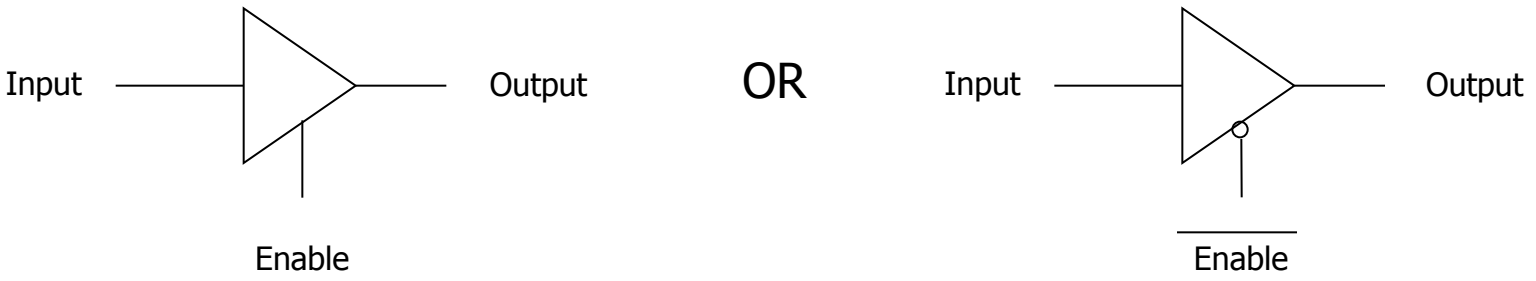
The Output is High



High Impedance

The Tri-State Buffer

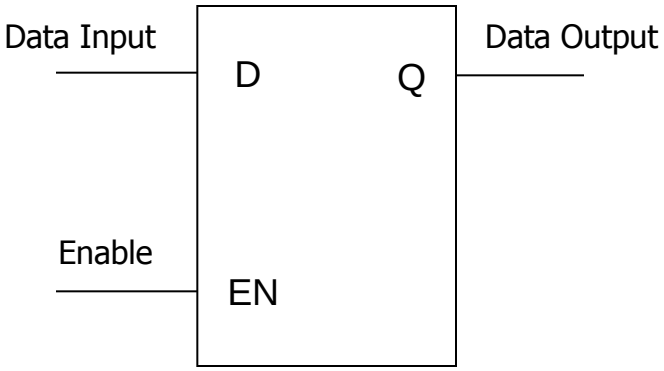
- This circuit has two inputs and one output.
  - The first input behaves like the normal input for the circuit.
  - The second input is an "enable".
    - If it is set high, the output follows the proper circuit behavior.
    - If it is set low, the output looks like a wire connected to nothing.



The Basic Memory Element

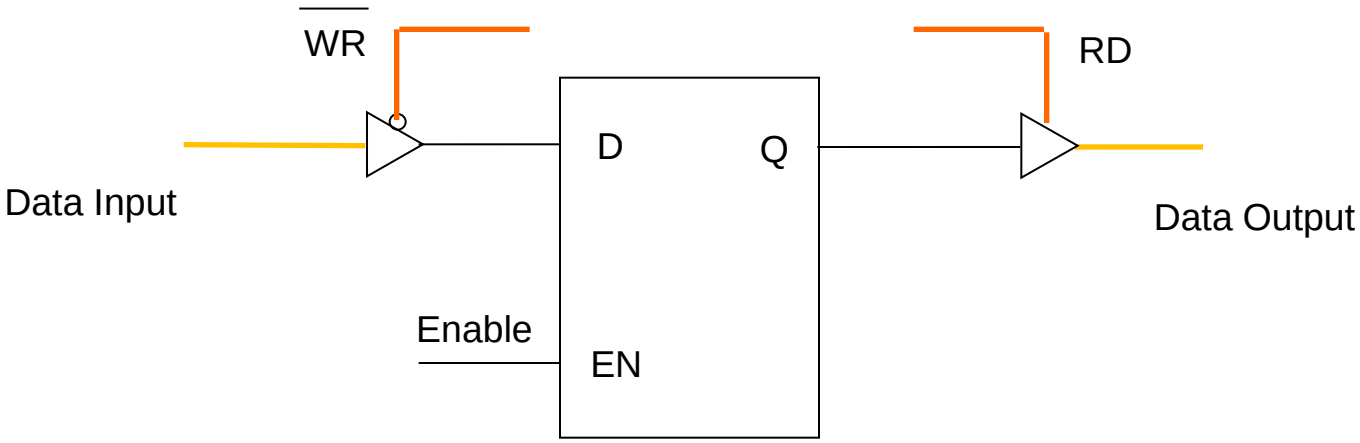
- The basic memory element is similar to a D latch.
  - Input
  - Output
  - Enable

D	Q(t+1)
0	0
1	1



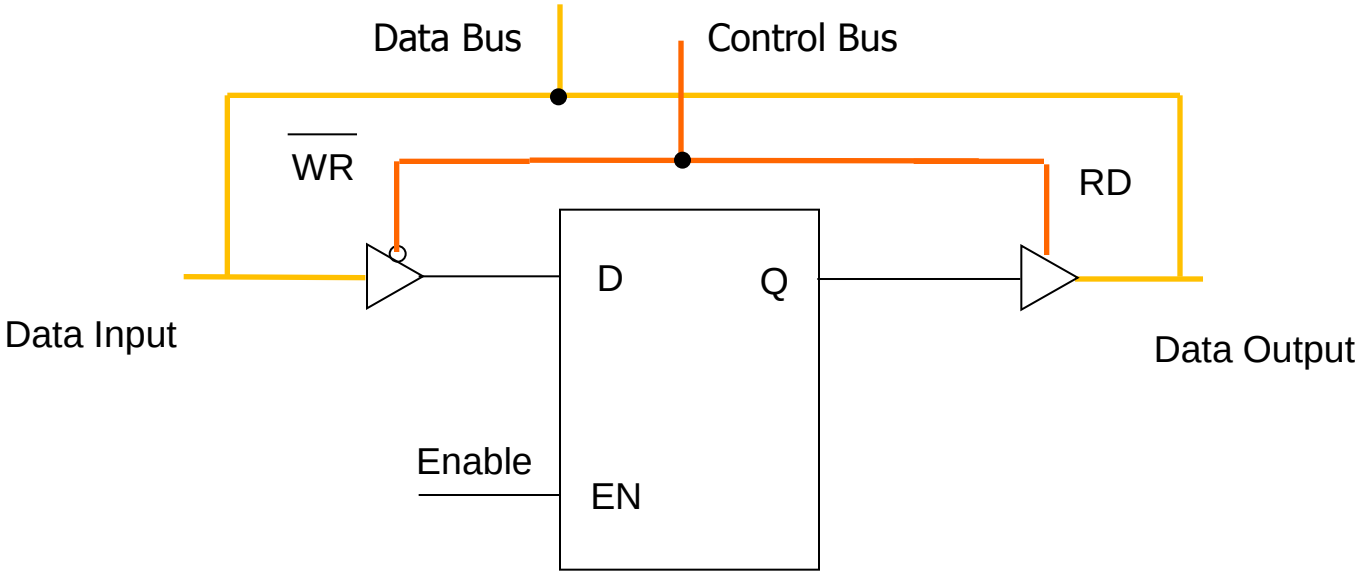
The Basic Memory Element

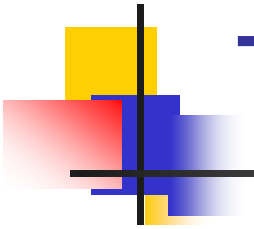
- Data is always present on the input and the output is always set to the content of the latch.
- tri-state buffers are added at the input and output of the latch.



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.



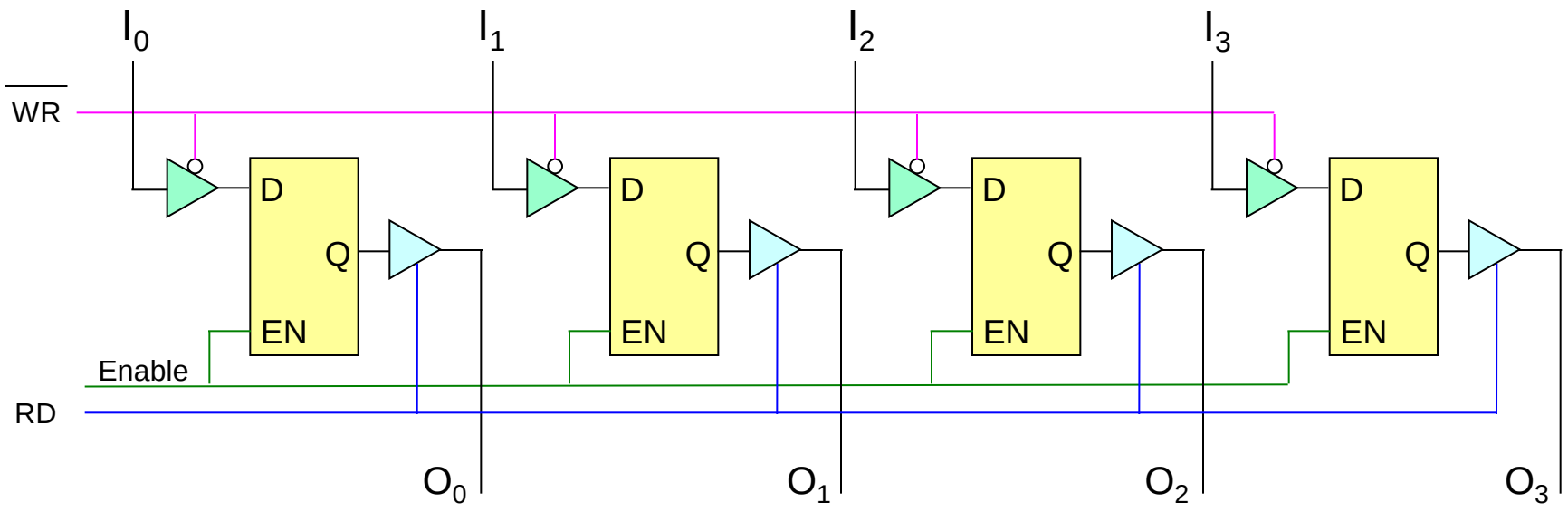


The Basic Memory Element

- 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.

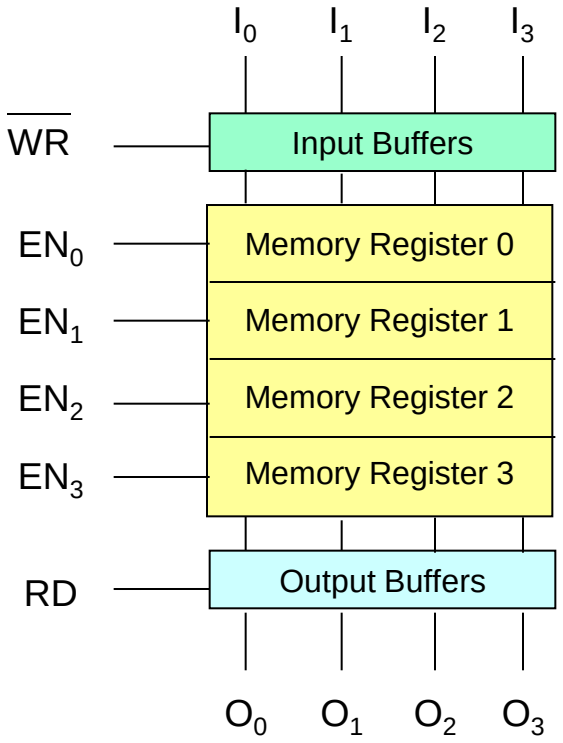
A Memory "Register"

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

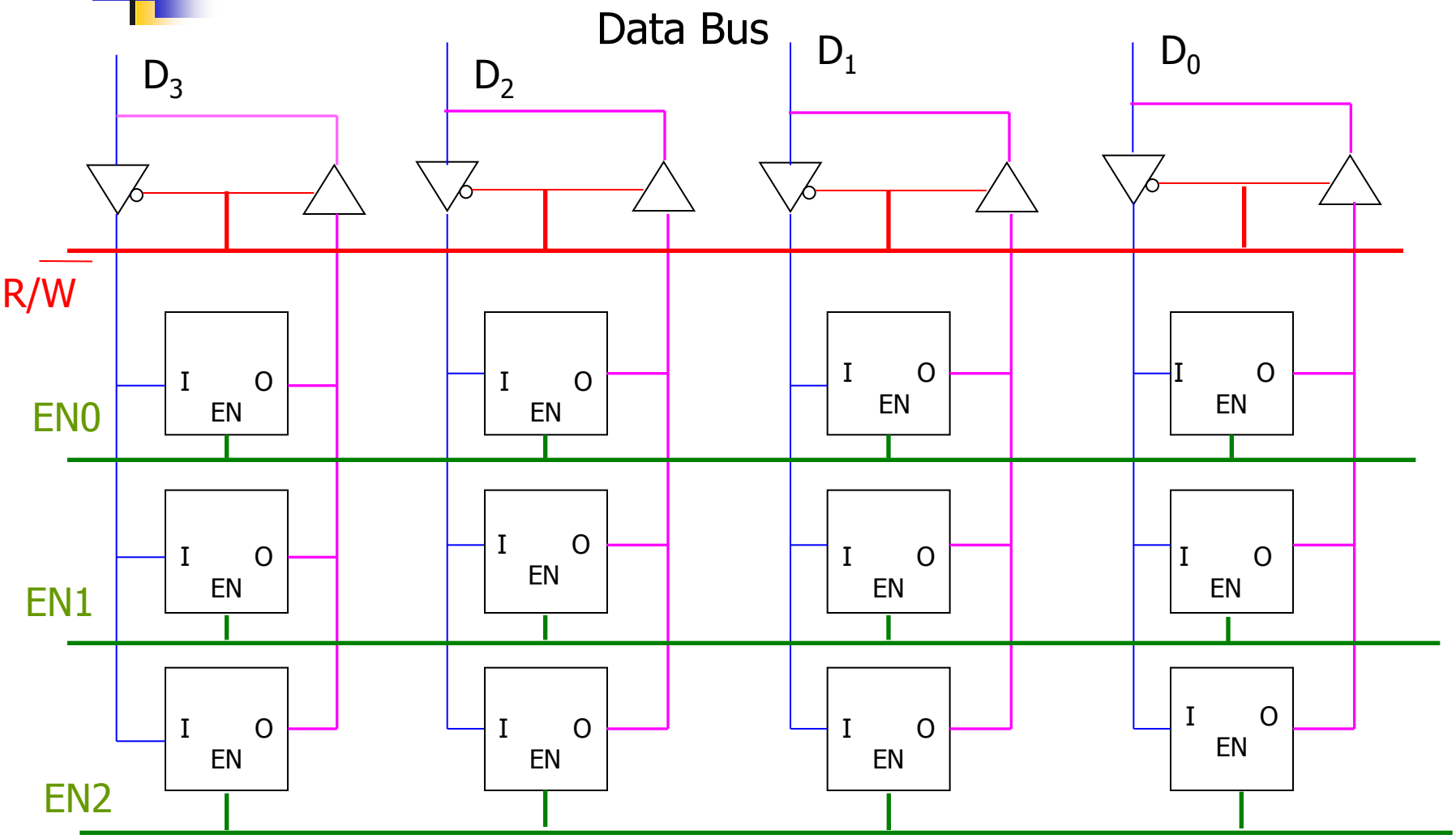


Memory Addressing

- 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 memory 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.

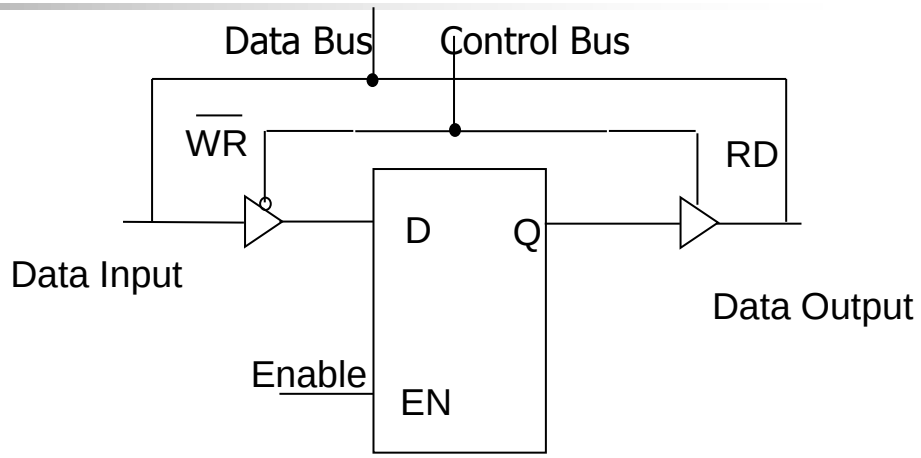


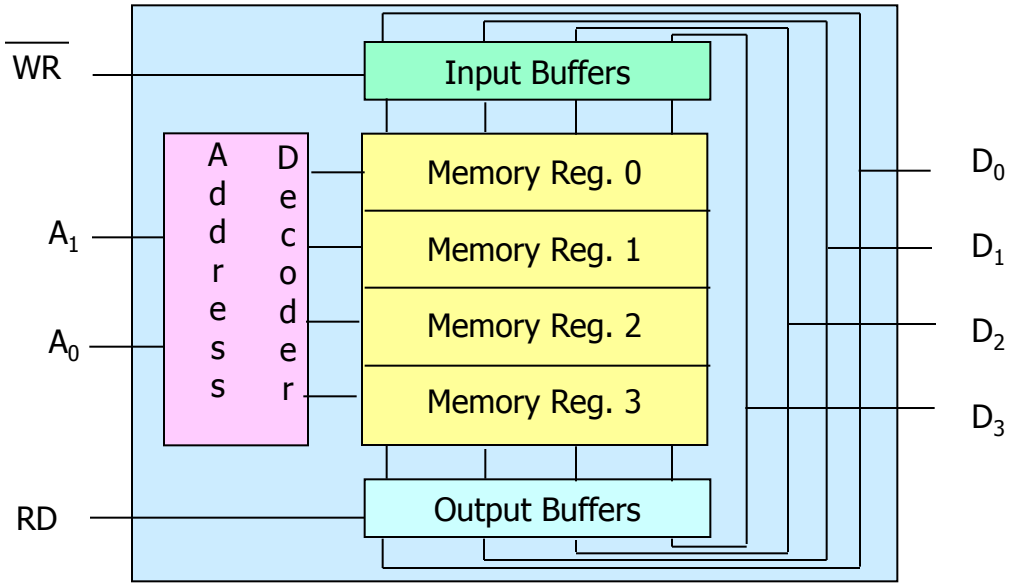
Memory Organization



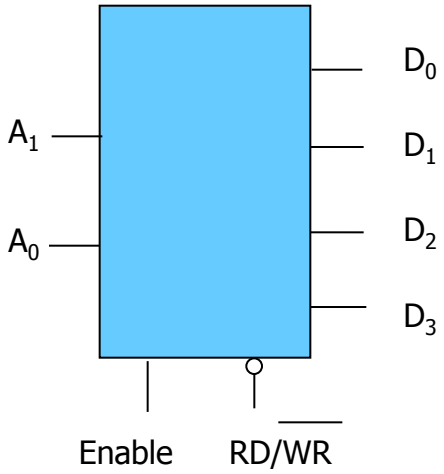
The Design of a Memory Chip

- 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.



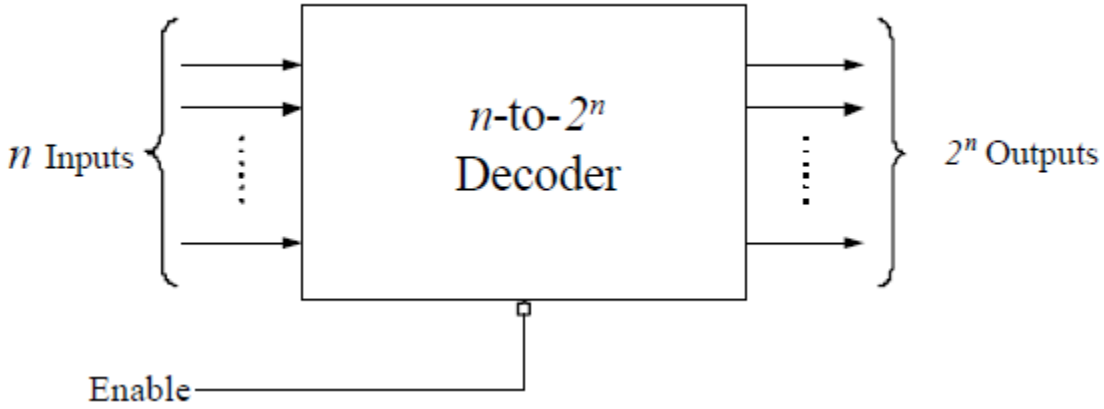


4x4 Memory



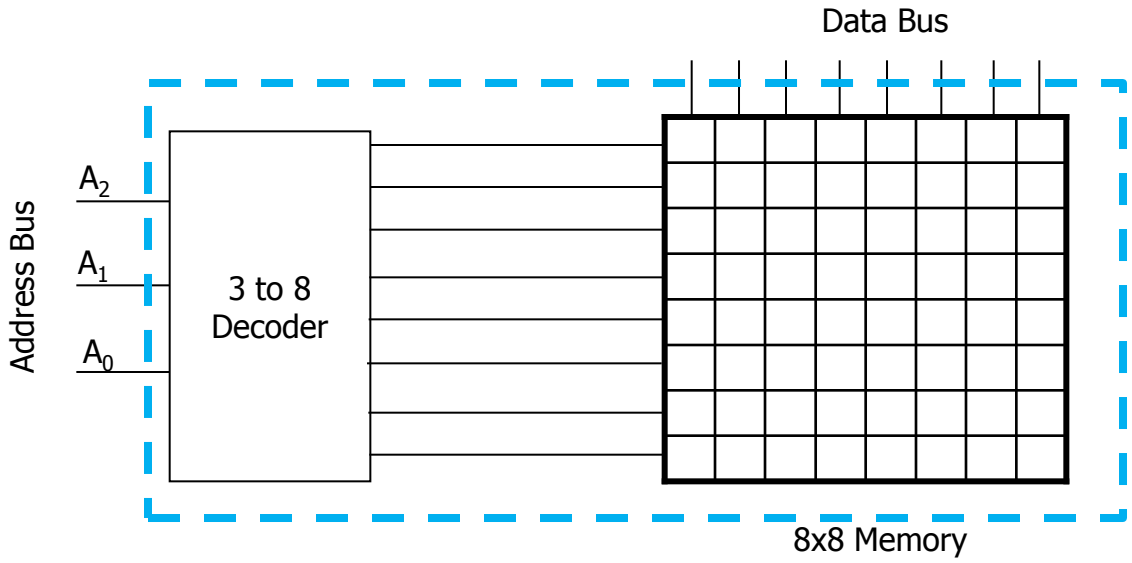
Memory Components

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

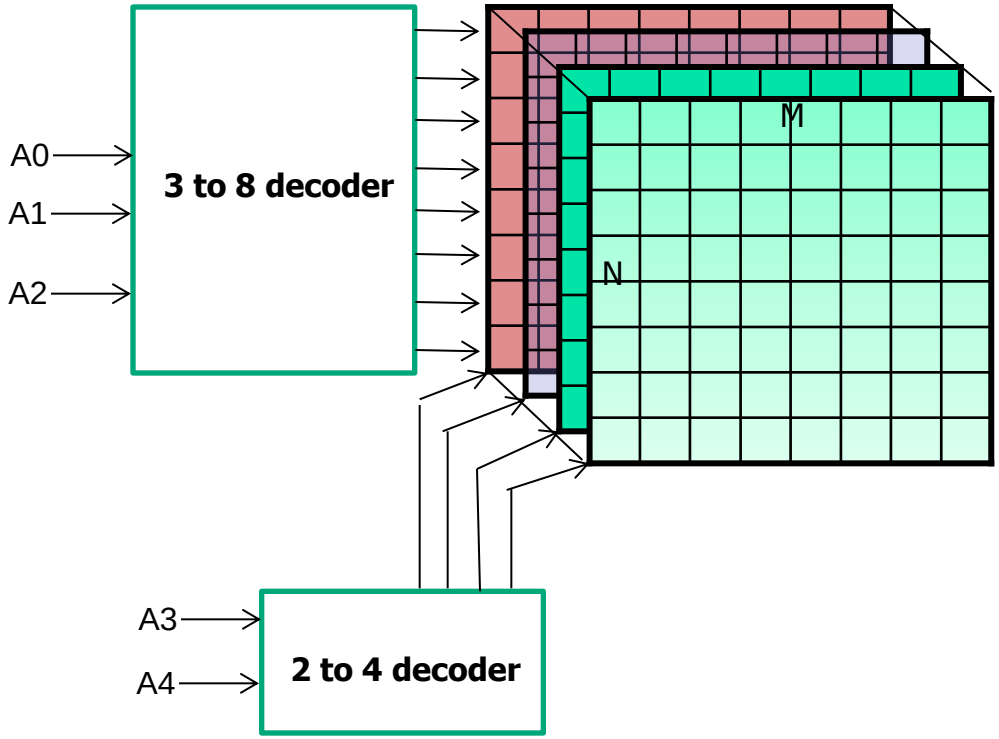


- n coded inputs to 2^n outputs

One Dimensional Addressing



Two Dimensional Addressing



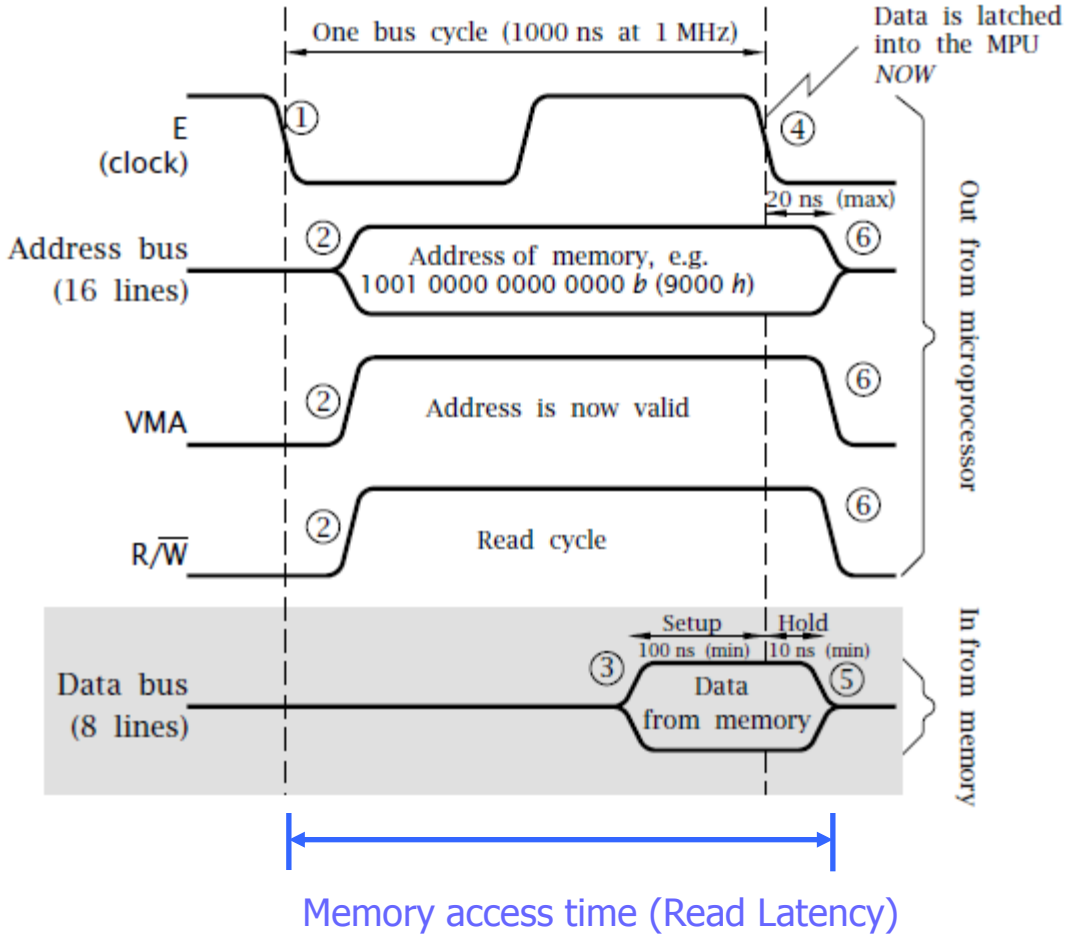
32X8 memory arranged in 2D configuration

This arrangement is more economical than 5-to-32 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)

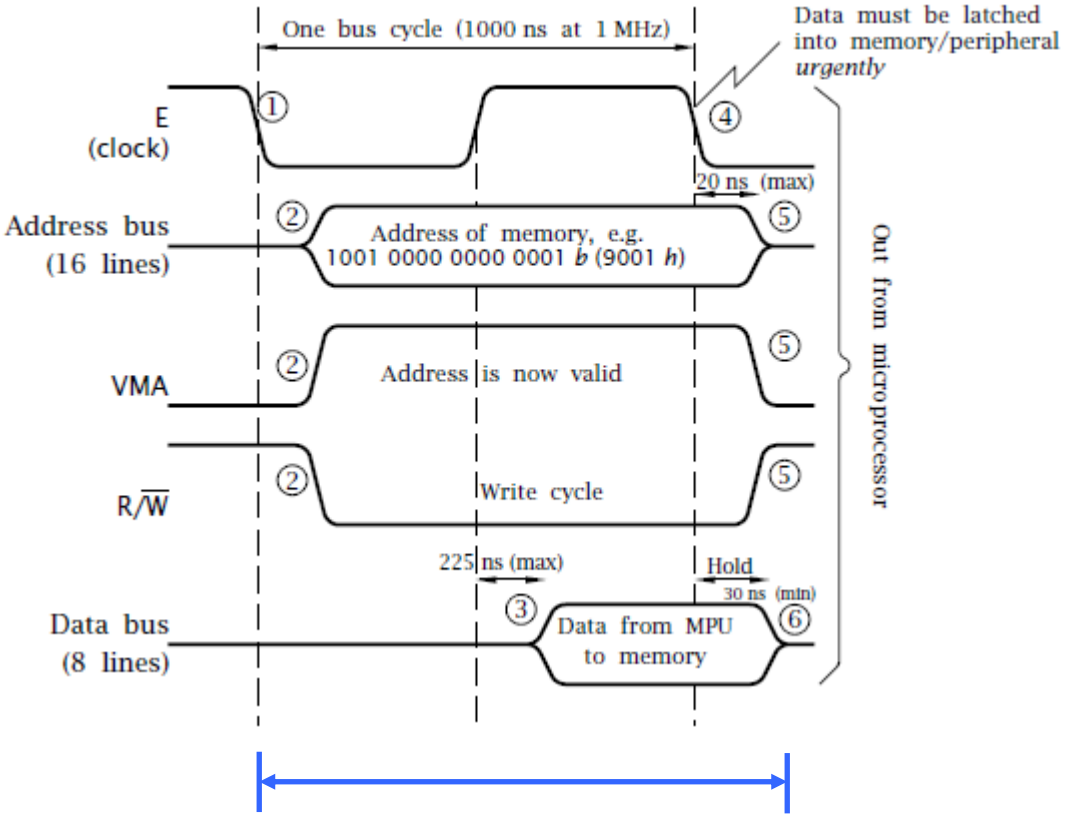


* Microprocessing Unit (MPU)
* Valid Memory Address (VMA)

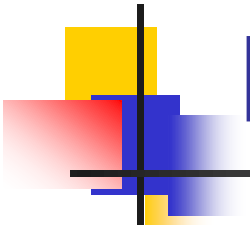
Memory Write

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)



References

- Lecture Slides: Dr. Şule Gündüz Öğüdücü
- Lecture Slides: Dr. Erdem Matoğlu
- Lecture Slides: Dr. Feza Buzluca
- Lecture Slides: Dr. Bassel Soudan
- Lecture Slides: Dr. Gökhan İnce