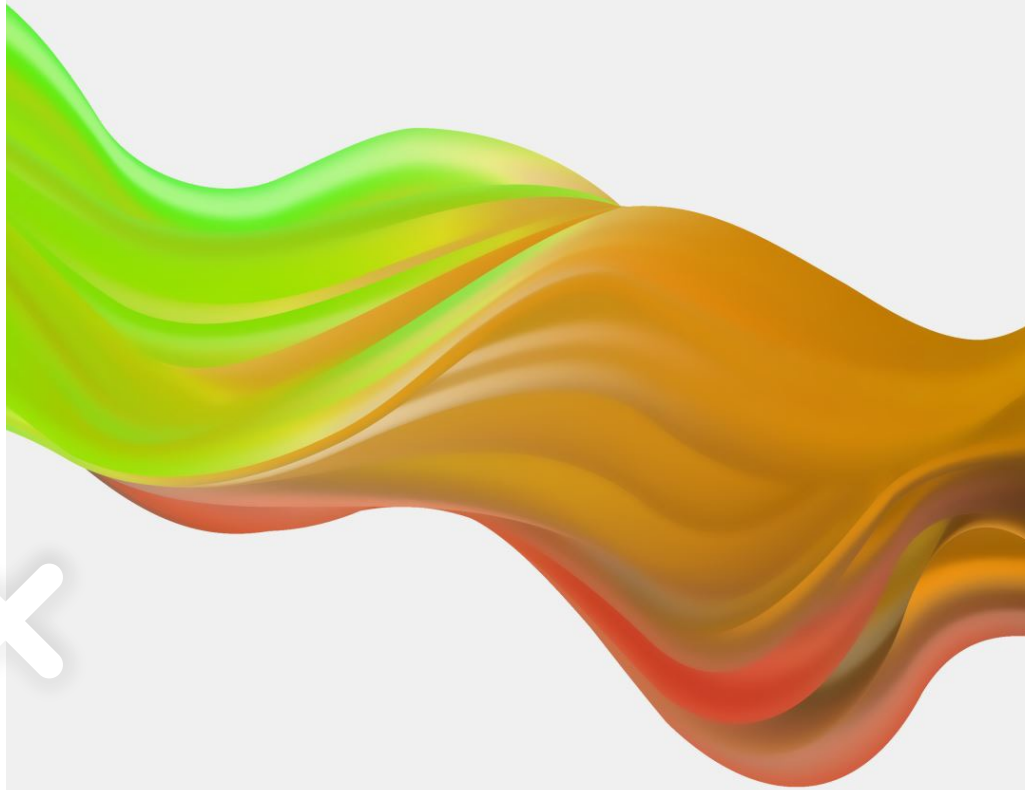
Computer Programming

MENG2020



Introduction

The History of Computing

780-850

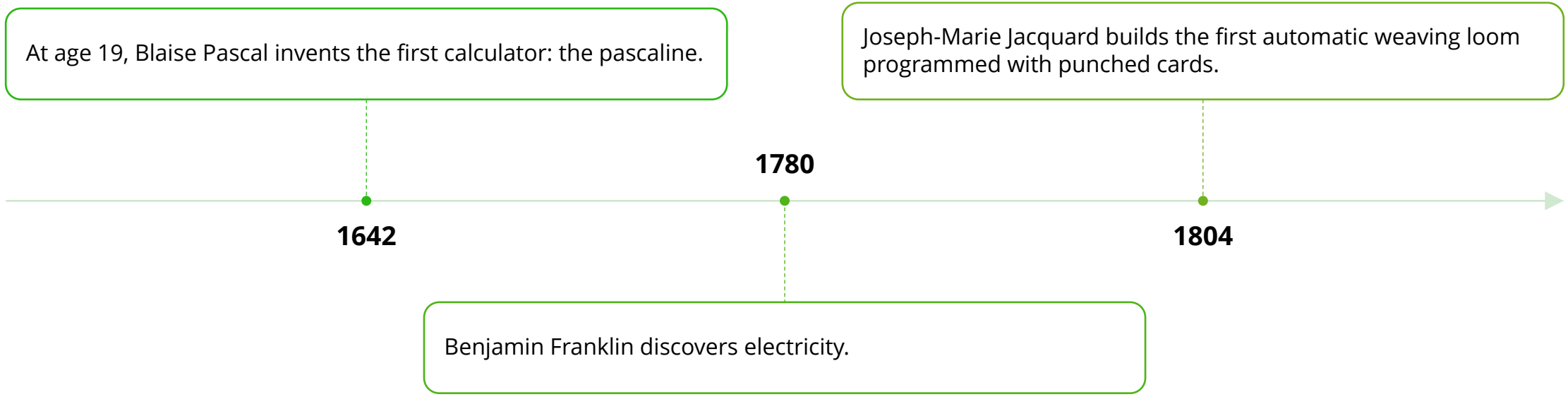
+ Muhammed ibn Musa Al-Khwarizmi

Tashkent cleric, mathematician and professor introduced the Hindu positional, decimal system and the use of zero into Arabic mathematics.

+ *About a thousand years later*

Ada Lovelace honoured him when she coined the word Algorithm.

And...


At age 19, Blaise Pascal invents the first calculator: the pascaline.
Joseph-Marie Jacquard builds the first automatic weaving loom programmed with punched cards.
1780
1642
1804
Benjamin Franklin discovers electricity.

History of Computing

- **Charles Babbage** (1791-1871)

  was an English Mathematician,philosopher,inventor Known to some as the "Father of Computing" for his contributions to the basic design of the computer through his Analytical machine. His previous Difference Engine was a special purpose device intended for the production of tables.

  He originated the concept of a programmable computer.

- **Ada Lovelace** (1815-1852)

  she is mainly known for having written a description of Charles Babbage's early mechanical general-purpose computer, the analytical engine. She is today appreciated as the "first programmer" since she was writing programs that is, manipulating symbols according to rules for a machine that Babbage had not yet built. She also foresaw the capability of computers to go beyond mere calculating or number-crunching while others, including Babbage himself, focused only on these capabilities.

1854

- **George Boole** creates the algebra that bears his name today (boolean).

AND	T	F
T	T	F
F	F	F

OR	T	F
T	T	T
F	T	F

1939-1945

- During WWII, Germany used the **Enigma** machine to encode its transmissions.

- 1945 : The first bug! –
- found on Sep 9th at 15:45 by Grace Hopper, then working on the Mark II computer at Harvard University. She became the highest ranking female Navy person of her time (Rear Admiral) and a role model to thousands of young women. The bug now resides at the National Museum of American History in Washington DC

1996-1998

+ **New services still!**

Computer Systems

Computers are electronic systems that can transmit, store and manipulate information (data).

Data can be numeric, character, graphic and sound.

To manipulate data, a computer needs a set of instructions called a program.

Algorithms

An algorithm is a series of instructions on how to solve the problem. We need algorithms before we can write programs.

Algorithms have the following characteristics:

Precision: must give the correct results.

Uniqueness: same inputs will give the same results.

Finiteness: it must stop eventually.

Input: most algorithms need input.

Output: the answers are the outputs.

Generality: must work with multiple data sets

Another Complex Algorithm

Sorting mail:

1. Get piece of mail from mail box
2. If piece is personal
    2.1 Read it

else

    if piece is a magazine

        2.1.1 Put in magazine rack

    else

        if piece is a bill

            2.1.1.1 Pay it

        else

        if piece is junk mail

                2.1.1.1.1 Throw it away

Writing an Algorithm

We will now learn how to write an algorithm to solve a simple problem:

Sort the following numbers in ascending order:

7 2 8 3 5

Think about how you would solve this problem for a moment.

Sorting is a common problem handled by computers.

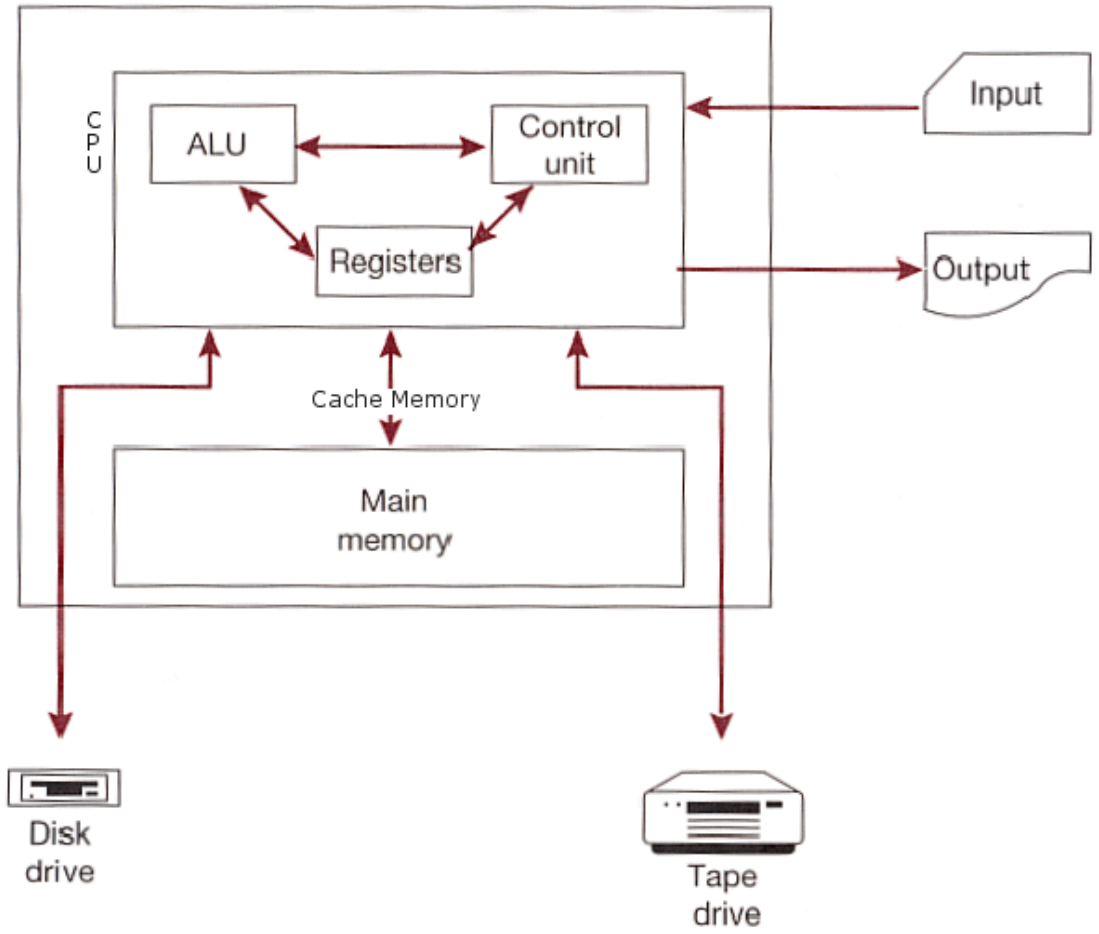
Introduction - A Look at Digital Computation

A. Hardware

Physical components that together provide the functionality required by a computing machine

- CPU (Central Processing Unit)
  - ALU (Arithmetic Logic Unit)
  - Internal Registers
- Memory
  - RAM (Random Access Memory) aka Read/Write Memory
  - ROM (Read Only Memory)
  - Cache
- I/O Device
  - Keyboard
  - Mouse
  - Display
  - Printer

Components of a Computer



Storage Types

CPU registers: Only a few cells on CPU.

Read-Only Memory (ROM): Thousands of cells containing pre-set computer instructions.

Main memory (RAM): Billions of cells on circuits separate from CPU. Memory location picked-up at random and access time is the same.

Secondary storage: Hundreds and thousands of billions of cells on disks (magnetic or optic), flash drives or tapes.

Secondary storage is not volatile (data is kept even when power is off).

B. Software

Programming Languages

Procedural

- Pascal
- Fortran
- Basic
- Turing
- C

Object Oriented

- Java
- C++

Applications

- Word Processors
- Spreadsheets
- SAS
- MATLAB

Operating Systems

- Windows
- Unix
- Linux
- MAC-OS

DIGITAL SYSTEMS

Digital systems: Systems that use a binary-like notation,

Number representation can be done in different bases, the most frequently used include:

Base 2 - Binary Number Representation

All numbers are shown using only 2 digits, i.e., 0 and 1

Base 8 - Octal Representation (Oct for short)

All numbers are shown using only 8 digits, i.e., 0, 1, 2, ..., 6, 7

Base 10 - Decimal Representation

All numbers are shown using only 10 digits, i.e., 0, 1, 2, ..., 8, 9

Base 16 - Hexadecimal representation (Hex for short)

All numbers are shown using only 16 digits, i.e., 0, 1, ..., 8, 9, A, B, C, D, E, F

Binary System

To count in binary is similar to the usual count in decimal: You start at 0 and at 9 you run out of digits so you use two instead like 10. In binary we run out of digits at 1 instead of 9. So we count in binary from 0 to 10 decimal with 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, and 1010.

Like in decimal, we can pad with zeros to the left without changing the value:
0000 1001 is still a binary equivalent of 9.

Integer numbers shown in different bases

Decimal	Binary	Oct	Hex
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11

Internal Representations

Bit: Binary digit (0 or 1).

Byte: A group of 8 bits. One character.

Word: The width of a memory cell.

Nibble: A group of 4 bits. Used to make bit sequences easier to read.

Each byte of main memory has an address.

All numbers and all characters are represented in binary code.

Integer Numbers (Unsigned)

All numbers are converted in binary:

ex: 9 = 1001

So 9, the integer in a 32-bit system, would look like:
(the spaces between the groups of 4 bits are only there for clarity)
0000 0000 0000 0000 0000 0000 0000 1001

Always place the number on the right and pad the unused bits with zeros.

Can you find the decimal value of this number?
0000 0000 0000 0000 0000 0000 0010 1101

BASE CONVERSIONS

A. Other bases to Decimal

For any given number (M) in base (n) such as

$M)_n = m_{d-1}\ m_{d-2}\ m_{d-3} \ldots m_1\ m_0$

the conversion can be easily as follows:

$M)_{10} = m_{d-1} \times n^{d-1} + m_{d-2} \times n^{d-2} + m_{d-3} \times n^{d-3} + \ldots + m_1 \times n^1 + m_0$

For example:

$725)_8 = 7x8^2 + 2x8^1 + 5 = 469)_{10}$

$101001)_2 = 1x2^5 + 1X2^3 + 1 = 41)_{10}$

$1FBA)_{16} = 1x16^3 + Fx16^2 + Bx16^1 + A$

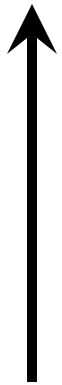
$= 1X4096 + 15x256 + 11x16 + 10 = 8122)_{10}$

B. Conversion Decimal to other bases

Can be easily done by consecutive division where dividend is the number itself, the divisor is the new base and the quotient is being discarded and at each step of division the reminder is noted.

For example:

$25)_{10} =?)_2$



+ 25/2=12 1 (This the least significant bit)
+ 12/2=6 0
+ 6/2=3 0
+ 3/2=1 1
+ 1/2=0 1 (This the most significant bit)

+ $25)_{10} =11001)_2$

Integer Numbers (Signed)

- There are many ways to represent negative integers. The simplest method is called **sign-and-magnitude**. It means using the leftmost bit as a sign bit (1 for negative, 0 for positive). In a 32-bit system that means that the first bit represents the sign, and the other 31 the absolute value of the number.
- For example –9 would be:
  1000 0000 0000 0000 0000 0000 0000 1001
- This method has one major drawback, it would allow for two zero values –0, and +0. For that reason, it is not used.

```
+0: 0000 0000 0000 0000 0000 0000 0000 0000

-0: 1000 0000 0000 0000 0000 0000 0000 0000
```

Integer Numbers (Signed)

- Without a doubt, your personal computer uses a method known as **two's complement**. A little more complicated than sign-and-magnitude but not that much and it has only one zero value.
- To have –9 in two's complement, you invert all the bits in the representation of 9 and add 1.
- You can achieve the same result by starting from the right; find the first 1 and invert all the bits to the left of that one.

```
+9 = 0000 0000 0000 0000 0000 0000 0000 1001
−9 = 1111 1111 1111 1111 1111 1111 1111 0111
```

Hexadecimal Numbers (HEX)

- In computer science, the base-16 numeral system is often used to represent groups of 4 bits. It uses 16 distinct symbols, most often the symbols 0–9 to represent values zero to nine, and A, B, C, D, E, F to represent values ten to fifteen.

Example:

+9 = 0000 0000 0000 0000 0000 0000 0000 1001
0 0 0 0 0 0 0 9

-9 = 1111 1111 1111 1111 1111 1111 1111 0111
F F F F F F F 7

Characters

Characters are expressed using the ASCII code:

'A' = 65 = 01000001

'g' = 103 = 01100111

'$' = 36 = 00100100

'+' = 43 = 00101011

Note that digits expressed in ASCII are different than integers or doubles.

'9' = 57 = 00111001

See the ASCII code at **www.asciitable.com**

Programming Languages

Many languages exist to program instructions to the computer.

1st generation: Machine languages (pure binary) Example: *10101110101010101011101010101*1

2nd generation: Assembly languages (mnemonic codes)
Example: *MV R1,R3*

3rd generation: High-level languages (C, Fortran, Java)
Example: *y = x * x + 2;*

4th generation: Closer to human languages than typical high-level programming languages (MATLAB, SQL)
Example: *find all records where name is "Kim"*

Solving Problems

- Here are the steps required to solve a problem with a computer program:

1. Define the problem
2. Analyze the problem.
3. Design a solution.
4. Implement the solution.
5. Test the program.
6. Update and maintain the program.

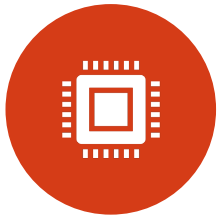
Why MATLAB for MENG2020?

It is a very popular software and language with students, engineers, and scientists in universities, research institutes and industries all over the world. It is being used by many advanced courses in the science program.

It is powerful and easy to use.

Programs are platform-independent. Programs written on one platform will run on all the other platforms without change (Windows, Unix, Linux, Macintosh).

MATLAB contains an extensive library of predefined functions.



MATLAB has advanced capabilities for plotting and graphics making it a visually attractive environment for new programmers.

MATLAB Avantages

Its basic data element is the matrix (like 2 two-dimensional grid). A simple integer is considered an matrix of one row and one column. Several mathematical operations that work on arrays or matrices are built-in to the Matlab environment. For example, cross-products, dot-products, determinants, inverse matrices.

Vectorized operations. Adding two arrays together needs only one command, instead of a for or while loop.

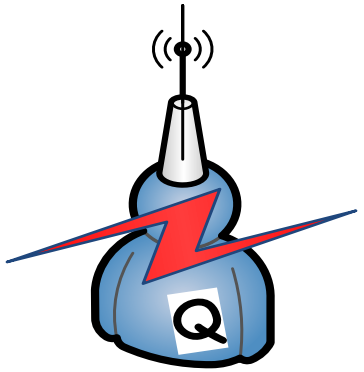
The graphical output is optimized for interaction. You can plot your data very easily, and then change colors, sizes, scales, etc, by using the graphical interactive tools.

Matlab's functionality can be greatly expanded by the addition of toolboxes. These are sets of specific functions that provided more specialized functionality. Ex: Excel link allows data to be written in a format recognized by Excel, Statistics Toolbox allows more specialized statistical manipulation of data (Anova, Basic Fits, etc)

Matlab Drawbacks

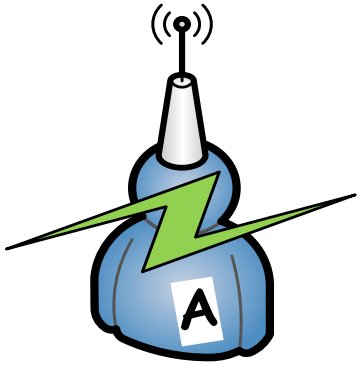
- It uses a large amount of memory and on slow computers it is very hard to use.
- It sits “on top” of Windows, getting as much CPU time as Windows allows it to have. This makes real-time applications very complicated.
- MATLAB is an interpreted language. The main disadvantage of interpreted languages is execution speed. When a language is compiled, all of the code is analyzed and processed efficiently, before the programmer distributes the application. With an interpreted language, the computer running the program has to analyze and interpret the code before it can be executed (each and every time), resulting in slower processing performance.
- For applications where performance is critical, a fast language like C is a much better choice.

converting between decimal and binary



$230)_{10} =?)_{2}$

$1100111)_{2} =?)_{10}$



- The number 230 can be expressed as:
  128 + 64 + 32 + 4 + 2
  So, the answer is: 11100110

- The number 1100111 represents:
  64 + 32 + 4 + 2 + 1
  So, the answer is: 103

EXAMPLE

- $(41)_{10} = (101001)_2$

A look at the memory locations –bytes

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	1
0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	1

Word: Depending on the type of hardware being used it can be 8, 16 or 32 bits. The width of a memory cell.

MEMORY

A collection of fixed number of storage locations where each location can be accessed using a unique address. Each location can also have a fixed number of bits.

RAM: Random Access Memory

- Static RAM
- Dynamic RAM

ROM: Read Only Memory

read only memory

Therefore, there are 2 types of operations that we'd like to perform on memory, i.e., read and write.

The following is an example of a memory with ***n*** locations. As you can see memory locations are consecutively numbered.

CPU

CPU has two main parts:

- registers
- ALU

A. Register

Registers are storage units inside the CPU. Registers are used to store the data and the result of operations.

B. ALU

ALU is the unit responsible for 2 types of operations that the CPU performs:

- Arithmetic
- Logical

hence the name ALU (Arithmetic Logical Unit)

Address bus to provide memory address

Data bus to facilitate exchange of data between CPU and memory

CPU

uses the data and the program stored in the memory to accomplish a requested task. To do so, CPU should first provide the address of the location in the memory that is going to be accessed (accessing can mean either reading or writing). CPU does that by putting the desired location address on a so-called address bus. The length of address bus therefore will indicate the max size of the memory, i.e.,

- If the address bus is 2-bit, CPU can access up to 4 (2^2) locations
- If the address bus is 3-bit, CPU can access up to 8 (2^3) locations
- If the address bus is 4-bit, CPU can access up to 16 (2^4) locations
- If the address bus is 8-bit, CPU can access up to 256 (2^8) locations

CPU – MEMORY LAY-OUT

If the address bus is 16-bit, CPU can access up to 65536 (2^{16}) locations

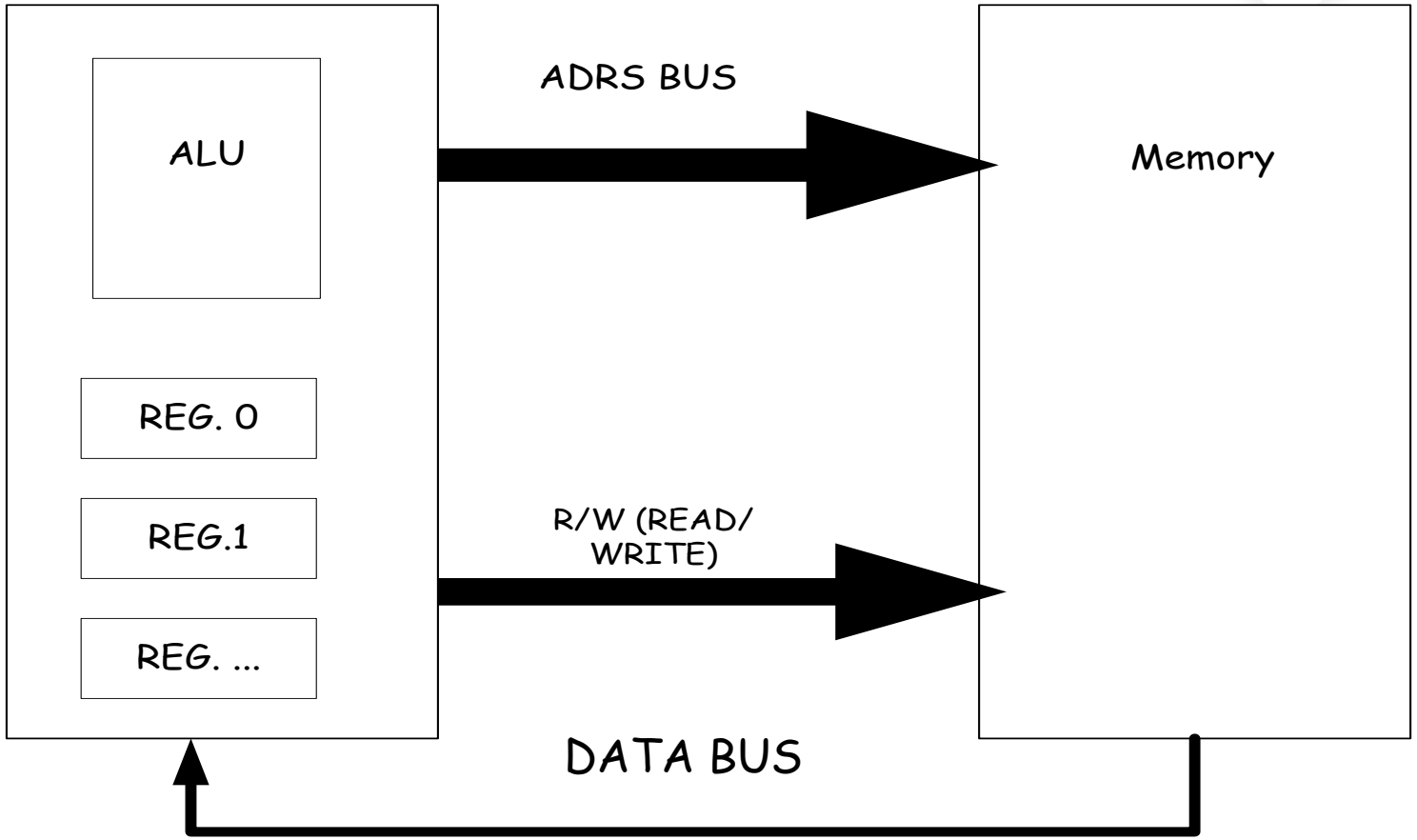
If the address bus is n-bit, CPU can access up to (2^{n}) locations

When the desired location in the memory is addressed, depending on the type of access, 2 things might happen:

In case of a read: address is provided by the CPU, the desired location is accessed and its content will be read. This is done by putting the content of the memory on a so-called data-bus.

In case of a write: address is provided by the CPU, the desired location is accessed. At the same type CPU should put the data on the data bus. This data then will be written to the desired location.

Simple interaction between CPU & Memory



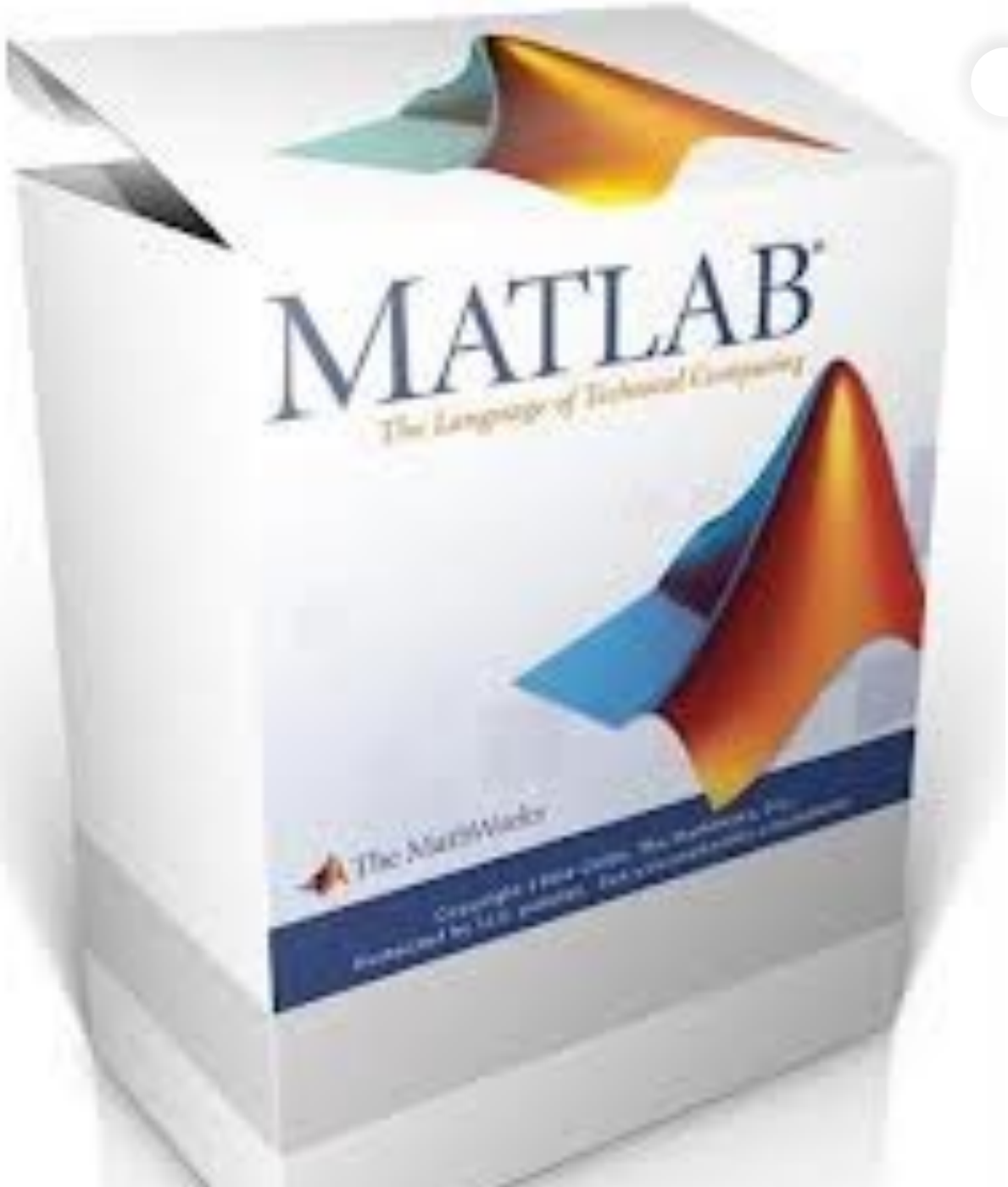
READ – WRITE OPERATIONS

READ:

- CPU puts the address on the address bus
- CPU issues read
- MEMORY puts the data on the data bus
- CPU gets the data from the data bus

WRITE:

- CPU puts the address on the address bus
- CPU puts the data on the data bus
- CPU issues write
- CPU puts the data into the specified memory location



Objectives

Understand what MATLAB is and why it is widely used in engineering and science

Start the MATLAB program and solve simple problems in the command window

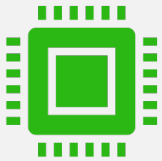
What is MATLAB?

MATLAB is one of a number of commercially available, sophisticated mathematical computation tools

Others include

- Maple
- Mathematica
- MathCad

What Is MATLAB?



MATLAB is both a powerful computational environment and a programming language that easily handles matrix and complex arithmetic.

It is a large software package that has many advanced features built in, and it has become a standard tool for many working in science or engineering disciplines.

Among other things, it allows easy plotting in both two and three dimensions.

MATLAB is good at:

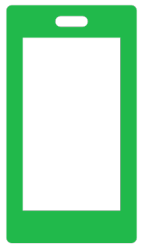
Numerical calculations

- Especially:matrices

Graphics

MATLAB stands for **Mat**rix **Lab**oratory

Why MATLAB



Easy to use

Built in programming language

MATLAB was originally written in Fortran, then later rewritten in C

Release Numbers

MATLAB is updated regularly

The Mathworks packages their software in groups, called releases, New releases are issued twice a year in the spring and in the fall

Release 2022 a includes	MATLAB 9.12.0	A number of specialized “toolboxes”