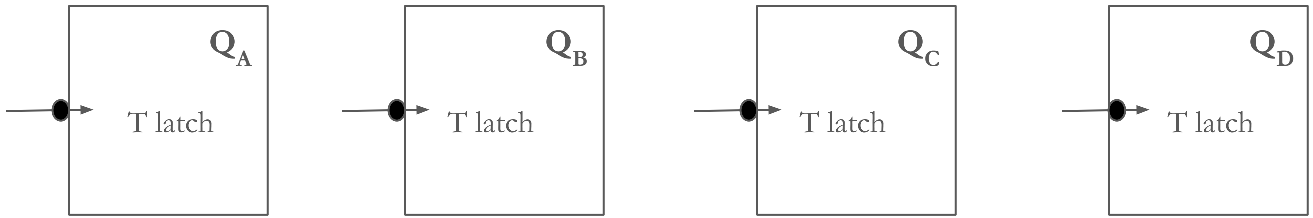
Ministry of Higher Education and Scientific Research
University of Larbi Ben M'Hidi, Oum El Bouaghi
Faculty of Exact Sciences and Natural and Life Sciences
Department of Mathematics and Computer Science

Computer Structure 1

Presented by: **Dr. NASRI/A**
nasri.ahlem1988@gmail.com
nasri.ahlem@univ-oeb.dz

2023-2024


Q_A
T latch
Q_B
T latch
Q_C
T latch
Q_D
T latch

Target audience	1st year mathematics student + 1st year Common core Mathematics and computer science students .
Prerequisite	Elementary mathematics.
Assessment method	Assessment method: Exam (60%), Continuous assessment (40%)

Outline:

Chapter 0 — **General Introduction and history:** Definitions, History and General Architecture of a Computer

Chapter 1 — **Numeral system:** Binary, Octal, Decimal, Hex

Chapter 2 — **Information representation :** Binary coding, Character representation and Numbers representation.

Chapter 3 — **Binary Boolean algebra:** Definition, Basic operators

Chapter 0: General Introduction and History

Outline:

1. **Definitions**
   - Computer Science;
   - Computer.
2. **History**
   - Before the 20th century;
   - 20th century;
   - 21th century.
3. **General Architecture of a Computer**
   - Central Processing Unit (CPU);
   - Random Access Memory (RAM);
   - Units of Memory Capacity Measurement
4. **recap**

Computer: is an automatic information processing machine that is capable of receiving, storing, processing, and outputting data and information under the control of a set of instructions, often referred to as programs or software.

Inputs → **Processing** → **Outputs**

Computer science (Computing): "It is the science of **processing information automatically**”.

- Using computer programs, algorithms, or automated systems to perform tasks related to information processing without **direct** human intervention.

Before the 20th century

1- Blaise Pascal (1642)
2- Gottfried Leibniz (1673)
3- George Boole (1854)
4- Claude Shannon (1984

20th century

1- Alan Turing (1936)
2- John Von Neumann (1945)

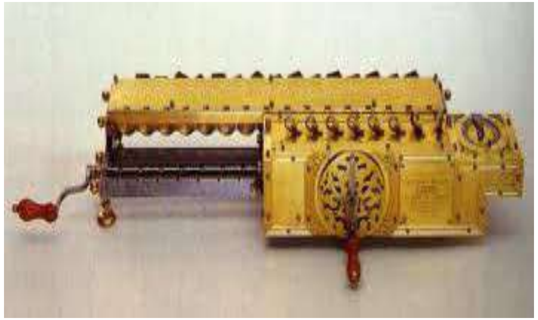
Birth of the Computer industry

1- 1st generation (1950)
2- 2nd generation (1960)
3- 3rd generation (1970)
4- 4th generation (1980)

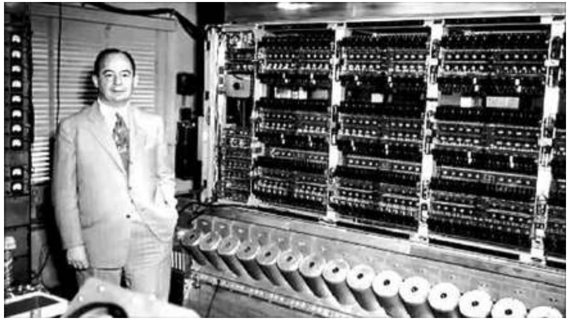
21th century

1- Central computers (mainframes)
2- Personal Computers or PC
3- Personal Digital Assistant

The Pascaline

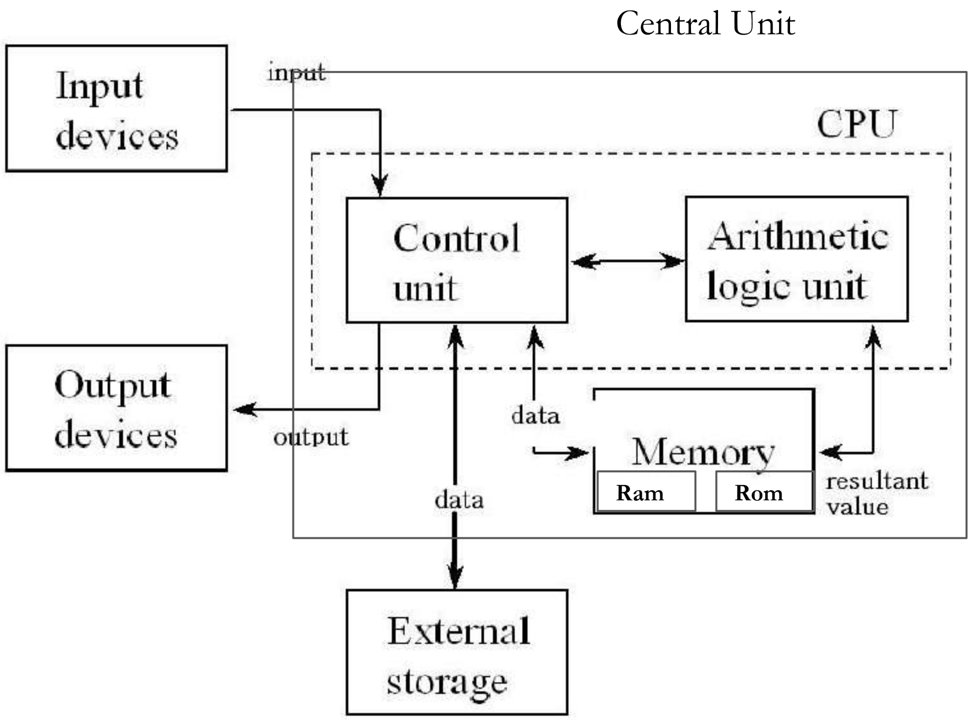


Leibniz's arithmetic machine



John von Neumann machine

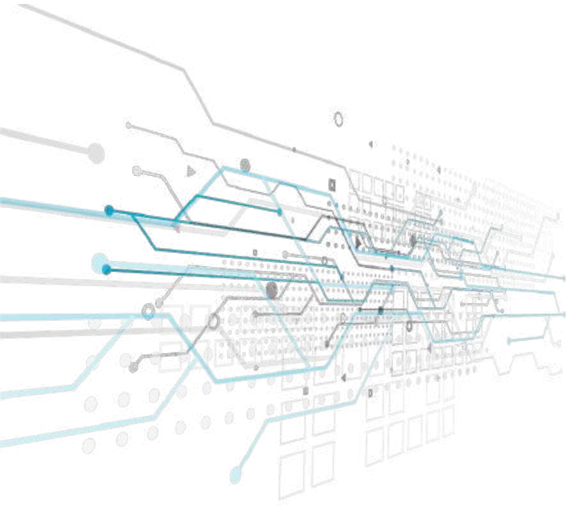
Hardware:


Central Unit
Input devices
input
CPU
Control unit
Arithmetic logic unit
Output devices
output
data
Memory
Ram
Rom
resultant value
data
External storage

Central Processing Unit (CPU)	**Arithmetic and Logic Unit (ALU)**	responsible for performing arithmetic and logical operations on data, which are essential for the computer to execute various tasks and processes.
	Control Unit	responsible for managing and coordinating the operations of the entire processor.
	Registers	used to store and manage data that the CPU needs to access quickly during its operations
Central Memory	**RAM**	(Random Access Memory):Volatile Memory; can be read and written in a standard way
	ROM	(Read-Only Memory): Non-Volatile Memory, a memory that can only be accessed for reading.

Buses	**Data Bus**	used to transmit data between the CPU, memory, and input/output devices. It allows for the transfer of binary data in both directions (read and write).
	Address Bus	used to transmit memory addresses. It determines the location in memory where data needs to be read from or written to.
	Control Bus	transmits a number of synchronization signals which ensure that the processor and the various online peripherals operate harmoniously.
External Storage	An external hard drive is a portable, standalone storage device that connects to the computer via USB, Thunderbolt, or other interfaces. It provides additional storage capacity. Ex: Solid-State Drive (SSD), USB Flash Drive, Memory Cards and SD Cards and Cloud Storage, … etc	

Peripherals	**Input Peripherals**	Keyboard, Mouse, Scanner, …
	Output Peripherals	Monitor (or Display), Printer, Projector, Speakers, …
	Input/Output Peripherals	Touchpad and Trackball, Modem, External memory



Software:

Operating system	the first point of contact between the computer and the user (human). It is software that consists of a set of basic applications required for the proper operation of the hardware: keyboard, screen, printer, and so on. Ex: Windows, Linux, IOS, Android, …
Apllications	a collection of programs that work together to give a service to the user. Microsoft Office, for example

Units of measurement:

Unit	Meaning
Byte, bit	**Capacity, size**: mainly used for memories (cache, RAM, disks).
Bit / second	**Speed** (bps) bit per second. used to calculate the speed of uploading.
Hertz	**Frequency**: number of events per second. Used for CPU Bus Frequency, Screen Refresh Rate, RAM Bus Frequency.

Units of measurement:

Units	value	in bytes
Byte	8 bits	1
Kb: kilo-Byte	1 024 Bytes	2^{10} bytes
Mb: mega-Byte	1 024 KB	2^{20} bytes
Gb: giga-Byte	1 024 MB	2^{30} bytes
Tb: tera-Byte	1 024 GB	2^{40}bytes

unit	value	in bps
Byte/second	8 Bps	2^{10} bps
Kbps: kilo-bit/ second	1 024 bps	2^{10} bps
Mbps: mega-bit/second	1 024 Kbps	2^{20} bps
Gbps: giga-bit/second	1 024 Mbps	2^{30} bps

unit	value	in Hertz
KHz: kilo-Hertz	1 000 Hz	10^3 Hz
MHz: mega-Hertz	1 000 KHz	10^6 Hz
GHz: giga-Hertz	1 000 MHz	10^9 Hz

Q1: What is a computer science ?

Q2: what are components of a computer ?

Q3: 1 GB= ? B / 10^6 B =? MB

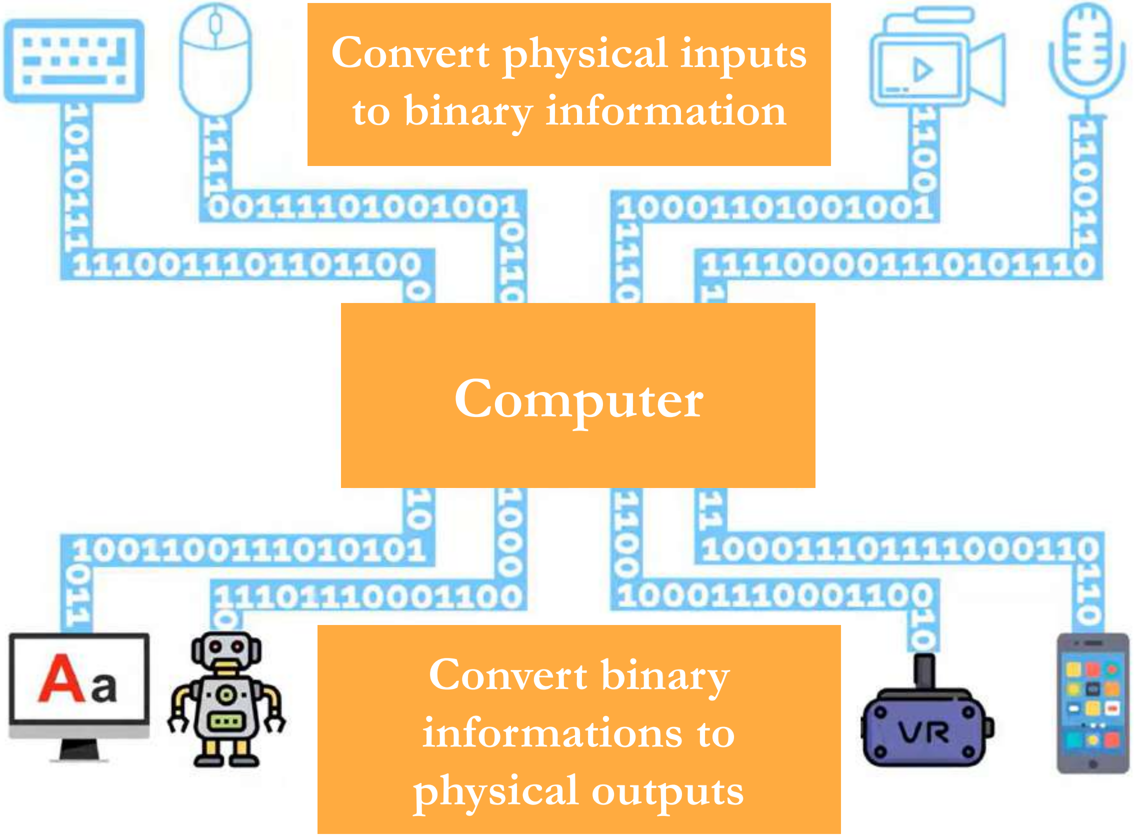
Ministry of Higher Education and Scientific Research
University of Larbi Ben M'Hidi, Oum El Bouaghi
Faculty of Exact Sciences and Natural and Life Sciences
Department of Mathematics and Computer Science

Computer Structure 1

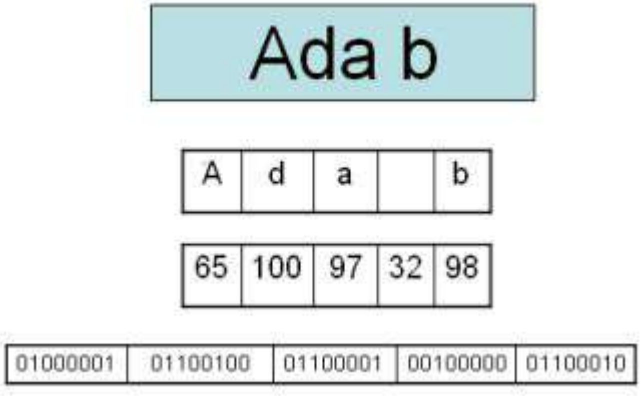
Presented by: **Dr. NASRI/A**
nasri.ahlem1988@gmail.com
nasri.ahlem@univ-oeb.dz

2023-2024

Chapter 1: Numeral systems


Convert physical inputs to binary information
Computer
Convert binary informations to physical outputs

Coding information => creating a **correspondence** between the (normal) **external** representation and its **internal** representation in the computer



Example of encoding of the character string "Ada b".

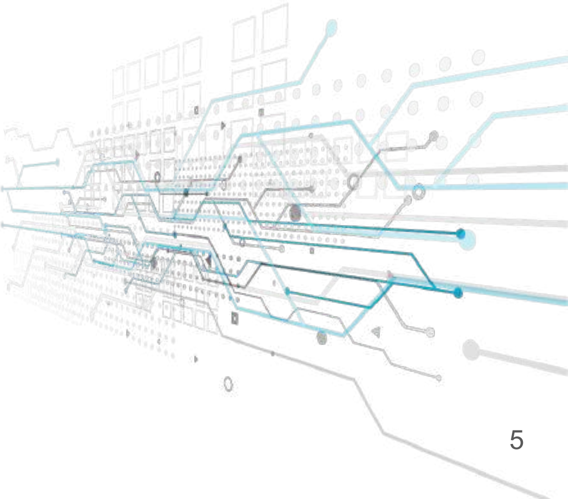
Base 2

Base 8

Base 10

Base 16

The binary system which uses two different numbers: 0, 1



Base 2

Base 8

Base 10

Base 16

The octal system which uses eight different digits: 0 to 7

Introduction

Base 2

Base 8

Base 10

Base 16

The decimal system using ten different digits (0 to 9)

Introduction

Base 2

Base 8

Base 10

Base 16

The hexadecimal system which uses sixteen different numbers: from 0 to 9, in addition to five characters A, B, C, D, E, F

Numeral system is characterized by:

- A base B>1.

- Different coefficients or symbols $\mathbf{a_i}$ such as: $0 \leq \mathbf{a_i} < B$.

	Binary	Octal	Decimal	Hexadecimal
Base	2	8	10	16
coefficients	0,1	0,1,2,3,4,5,6,7	0,1,2,3,4,5,6,7,8,9	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Base 10

B_{10}	2	0	2	3
Rank of the number	3	2	1	0
Power of base	10^3	10^2	10^1	10^0
	1000	100	10	1

Base 10

N_{10}	2	0	2	3
Rank of the number	3	2	1	0
Power of base	10^3	10^2	10^1	10^0
	1000	100	10	1

$$2023 = 3\times10^0 + 2\times10^1 + 0\times10^2 + 2\times10^3$$

Base b

Generalization					
N_b	…	a_3	a_2	a_1	a_0
Rank of the number	…	3	2	1	0
Power of base	…	b^3	b^2	b^1	b^0

$$N_b = a_0 \times b^0 + a_1 \times b^1 + a_2 \times b^2 + a_3 \times b^3 + \ldots$$

Exercise:

Here are the given numbers: 1010, 1020, 108141, 2A0GF00, 01AFB, CEE, BAC.

- Among these numbers, which ones can be the presentation of a number in base 2, 8, 10 or 16?

- Give the smallest base in which each number can be written?

Solution

	Base 2	Base 8	Base 10	Base 16	Smallest Base
1010	Yes	Yes	Yes	Yes	2
1020	No	Yes	Yes	Yes	3
108141	No	No	Yes	Yes	9
2A0GF00	No	No	No	No	17
01AFB	No	No	No	Yes	16
CEE	No	No	No	Yes	15
BAC	No	No	No	Yes	13

Polynomial Form:

$$N = (a_n a_{n-1} a_{n-2} \ldots\ldots. a_1 a_0)_B$$

strong weight   low weight

$$N = a_n B^n + a_{n-1} B^{n-1} + \ldots.. + a_1 B^1 + a_0 B^0 = \sum_{i=0}^{i=n} a_i B^i$$

Polynomial Form:

$$N = (\underbrace{a_n a_{n-1} a_{n-2} \ldots\ldots a_1 a_0}_{\textbf{Integer Part}}, \underbrace{a_{-1} a_{-2} \ldots\ldots a_{-m}}_{\textbf{Decimal part}})_B$$

$$N = a_n B^n + a_{n-1} B^{n-1} + \ldots\ldots + a_1 B^1 + a_0 B^0 + a_{-1} B^{-1} + a_{-2} B^{-2} + \ldots\ldots + a_{-m} B^{-m} = \sum_{i=-m}^{i=n} a_i B^i$$

Example:

8592

strong weight low weight

568,592

Integer Part Decimal part

$8592 = 8x10^3 + 5x10^2 + 9x10^1 + 2x10^0.$

$568.592 = 5x10^2 + 6x10^1 + 8x10^0 + 5x10^{-1} + 9x10^{-2}$
$+2x10^{-3}.$

Converting from base B to decimal:

Method 1: Polynomial expansion

- Express the number in polynomial form using base b,
- then sum the various terms of the polynomial representation of the number.

Example:

$$(12)_3 = 1 \times 3^1 + 2 \times 3^0 = 3 + 2 = (5)_{10}$$

$$(101101)_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 32 + 8 + 4 + 1 = (45)_{10}$$

Converting from base B to decimal:

Example:

$(111{,}01011)_2 =$

$(1254{,}1)_8 =$

$(A5F{,}6)_{16} =$

Converting from base B to decimal:

Example:

$(111,01011)_2 = 1\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4} + 1\times2^{-5}$

$= 4+2+1+0+1/4+0+1/16+1/32= (7, 34375)_{10}$

$(1254,1)_8 =$

$(A5F,6)_{16} =$

Converting from base B to decimal:

Example:

$(111{,}01011)_2 = 1\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4} + 1\times2^{-5}$

$= 4+2+1+0+1/4+0+1/16+1/32 = (7{,}\ 34375)_{10}$

$(1254{,}1)_8 = 1\times8^3 + 2\times8^2 + 5\times8^1 + 4\times8^0 + 1\times8^{-1} = 512+128+40+4+1/8 = (684{,}125)_{10}$

$(A5F{,}6)_8 =$

Converting from base B to decimal:

Example:

$(111{,}01011)_2 = 1\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4} + 1\times2^{-5}$

$= 4+2+1+0+1/4+0+1/16+1/32 = (7{,}34375)_{10}$

$(1254{,}1)_8 = 1\times8^3 + 2\times8^2 + 5\times8^1 + 4\times8^0 + 1\times8^{-1} = 512+128+40+4+1/8 = (684{,}125)_{10}$

$(A5F{,}6)_{16} = 10\times16^2 + 5\times16^1 + 15\times16^0 + 6\times16^{-1} = (2655{,}375)_{10}$

Converting from decimal to base B:

Converting Integer Part

Method 1: Successive Subtractions

- Begin by determining the nearest power of B to the decimal number, and then subtract this power from the number
- Then, repeat the same process with the result of the subtraction until reaching 0

Converting from decimal to base B:

Converting Integer Part

Method 1: Successive Subtractions

Example 1:

$$(5)_{10} = (?)_2$$

$$5 - 4 = 1 - 1 = 0;\, donc\ 5 = 4 + 1$$

$$5 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (101)_2$$

Converting from decimal to base B:

Converting Integer Part

Method 1: Successive Subtractions

Example 2:

$(45)_{10} = (?)_2$

$$45 - 32 = 13 - 8 = 5 - 4 = 1 - 1 = 0; \textit{donc } 45 = 32 + 8 + 4 + 1$$

$$45 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (101101)_2$$

Converting from decimal to base B:

Converting Integer Part

Method 1: Successive Subtractions

Example 3:

$(555)_{10} = (?)_2$

$$555 - 512 = 43 - 32 = 11 - 8 = 3 - 2 = 1 - 1 = 0; donc\ 555 = 512 + 32 + 8 + 2 + 1$$

$$555 = 1 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (1000101011)_2$$

Converting from decimal to base B:

Converting Integer Part

Method 1: Successive Subtractions

Example 3:

$(555)_{10} = (?)_9 = (?)_8 = (?)_{16}$

It is not easy to find the first power of 8, 16 or 9 close to 555 !!!

This is the main disadvantage of this method.

Converting from decimal to base B:

Converting Integer Part

Method 2: Successive division

- Divide the whole number and each successive quotient by B until you obtain a null quotient.
- The sequence of remainders, in the reverse order of their obtaining, gives the representation of the decimal number in the base system B.

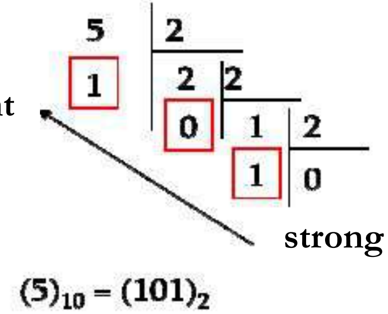
Converting from decimal to base B:

Converting Integer Part

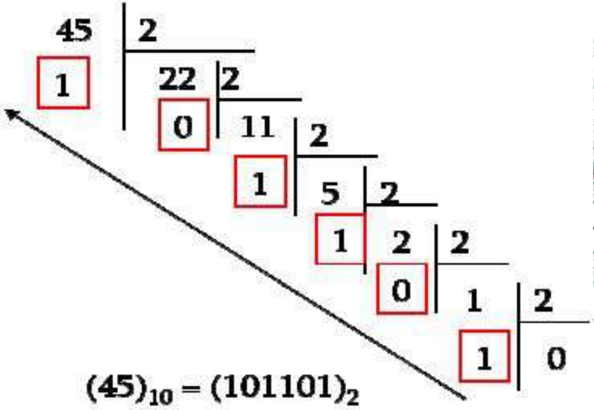
Method 2: Successive division

Example 1:

low weight



strong weight

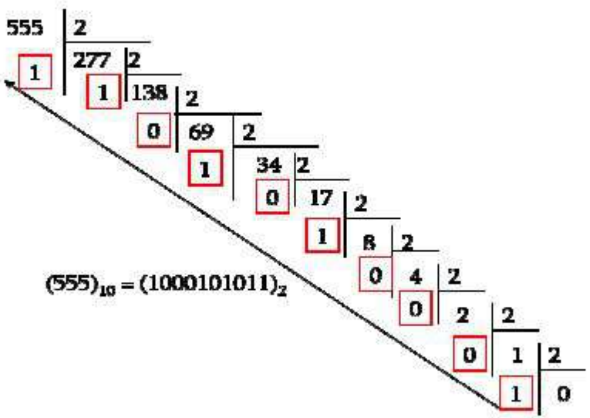


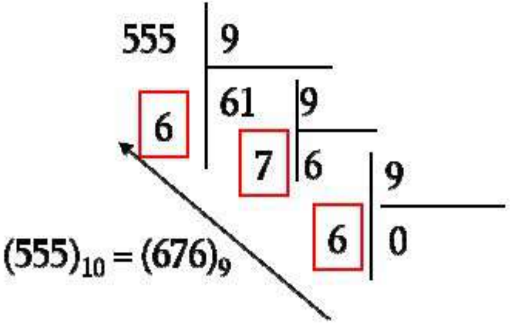
Converting from decimal to base B:

Converting Integer Part

Method 2: Successive division

Example 2:



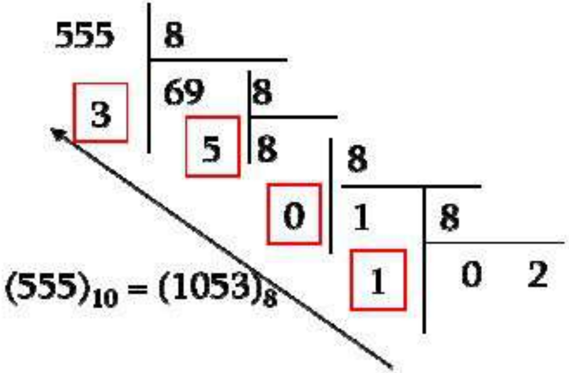


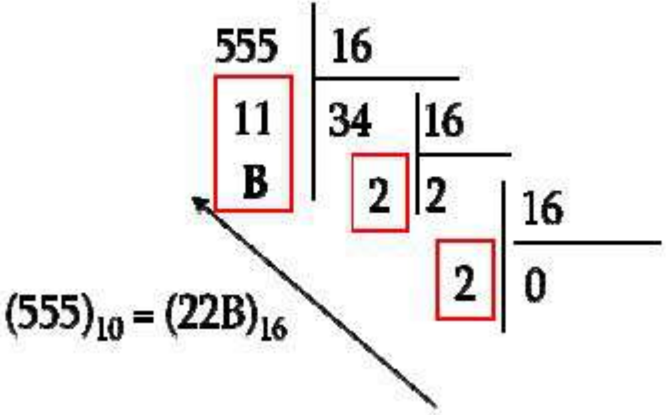
Converting from decimal to base B:

Converting Integer Part

Method 2: Successive division

Example 3:





Converting from decimal to base B:

Converting Decimal Part

Method 2: Successive multiplications

- Multiply the decimal part and the decimal parts of successive products by the base B until you obtain either a null decimal part or a repetition of one of the decimal parts.
- The finite, or infinitely repeated, sequence of the whole parts of the products obtained constitutes the representation of the decimal part in base b.

Converting from decimal to base B:

Converting Decimal Part

Method 2: Successive multiplications

Example1:

$(0,625)_{10}=(0,101)_2$

strong weight

$$
\begin{aligned}
0,625 * 2 &= \boxed{1},25 \\
0,25 * 2 &= \boxed{0},5 \\
0,5 * 2 &= \boxed{1},0
\end{aligned}
$$

$\downarrow$

low weight

$(0,6)_{10}=(?)_2$

$$
\begin{aligned}
0.6 * 2 &= 1.2 ; \\
0.2 * 2 &= 0.4 ; \\
0.4 * 2 &= 0.8 ; \\
0.8 * 2 &= 1.6 ; \\
0.6 * 2 &= 1.2 ; \\
0.2 * 2 &= 0.4 ; \\
0.4 * 2 &= 0.8 ; \\
0.8 * 2 &= 1.6 ; \ldots\ldots\ldots\ldots
\end{aligned}
$$

$(0.6)_{10}= (0.10011001)_2$

Converting from decimal to base B:

Converting Decimal Part

Method 2: Successive multiplications

Example2:

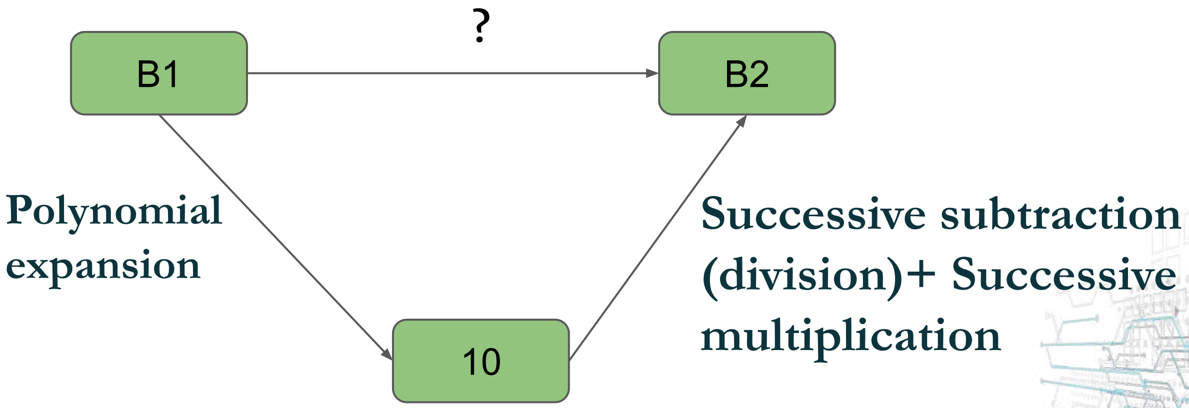
$(0.325)_{10}=(?)_2$

$0.325 * 2 = 0.650$
$0.650 * 2 = 1.300$
$0.300 * 2 = 0.600$
$0.600 * 2 = 1.200$
$0.200 * 2 = 0.400$
$0.400 * 2 = 0.800$
$0.800 * 2 = 1.600$
$0.600 * 2 = 1.200$
$0.200 * 2 = 0.400$
$0.400 * 2 = 0.800$
$0.800 * 2 = 1.600$

$(0.325)_{10}=(0101001)_2$

$(0.325)_{10}=(01010011001)_2.$

Converting from base B1 to base B2:



Converting from base B1 to base B2:

Example1:

$$(32,4)_5 = (?)_2$$

$$(32,4)_5 = 3 \times 5^1 + 2 \times 5^0 + 4 \times 5^{-1} = (17,8)_{10} = (10001,1100 \ldots \ldots)_2$$

Converting from base B1 to base B2:

Case of bases B and B^k (binary=>octal=>hexadecimal)

binary=>octal (do bursts on 3 bits)

Octal	0	1	2	3	4	5	6	7
Binary	000	001	010	011	100	101	110	111

Example:

$(37)_8 = (\underline{011}\ \underline{111})_2 = (11\ 111)_2$

$(1254,1)_8 = (\underline{001}\ \underline{010}\ \underline{101}\ \underline{100},\ \underline{001})_2 = (1\ 010\ 101\ 100,\ 001)_2$

$(1000101011)_2 = (\underline{001}\ \underline{000}\ \underline{101}\ \underline{011})_2 = (1053)_8$

$(1111000,01)_2 = (\underline{001}\ \underline{111}\ \underline{000},\ \underline{010})_2 = (170,2)_8$

Converting from base B1 to base B2:

Case of bases B and B^k (binary=>octal=>hexadecimal)

binary=>hexadecimal (do bursts on 4 bits)

Hexa	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Example:

$(A5F,6)_{16} = (\underline{1010}\ \underline{0101}\ \underline{1111}, \underline{0110})_2$

$(1000101011)_2 = (\underline{0010}\ \underline{0010}\ \underline{1011})_2 = (22B)_{16}$

$(1111000,01)_2 = (\underline{0111}\ \underline{1000}, \underline{0100})_2 = (78,4)_{16}$

Converting from base B1 to base B2:

Case of bases B and B^k (binary=>octal=>hexadecimal)

conversion	Method	Example
$2 \Rightarrow 8$	3 binary digits $\Rightarrow$ one octal digit	Binary $(\underline{101}\ \ \underline{110}\ \ \underline{011})_2$ $\downarrow\ \ \downarrow\ \ \downarrow$ Octal $(5\ \ 6\ \ 3)_8$
$8 \Rightarrow 2$	one octal digit $\Rightarrow$ 3 binary digits	Octal $(5\ \ 6\ \ 3)_8$ $\downarrow\ \ \downarrow\ \ \downarrow$ Binary $(\underline{101}\ \ \underline{110}\ \ \underline{011})_2$
$2 \Rightarrow 16$	4 binary digits $\Rightarrow$ one hexadecimal digit	Binary $(\underline{1010}\ \ \underline{0110}\ \ \underline{0011})_2$ $\downarrow\ \ \downarrow\ \ \downarrow$ Hexa $(A\ \ 6\ \ 3)_8$
$16 \Rightarrow 2$	one hexadecimal digit $\Rightarrow$ 4 binary digits	Hexa $(A\ \ 6\ \ 3)_{16}$ $\downarrow\ \ \downarrow\ \ \downarrow$ Binary $(\underline{1010}\ \ \underline{0110}\ \ \underline{0011})_2$

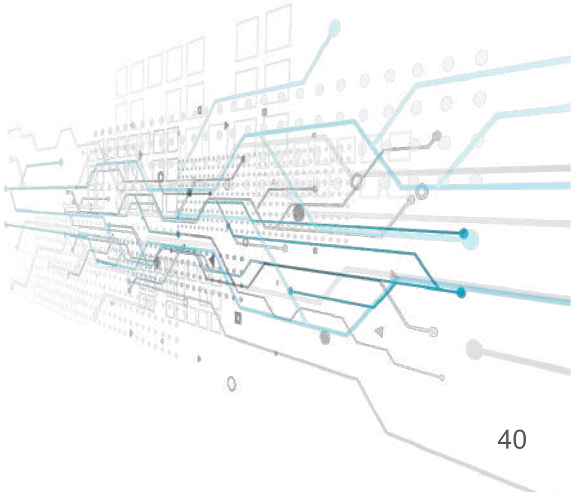
Converting a large decimal number to binary

To convert a large decimal number to binary, it's recommended to initially convert it to **octal** or **hexadecimal**, and then proceed to binary conversion => to reduce the number of operations (division or subtraction).

Example:

$(25693)_{10} = (?????????????)_2$

- **Method 1:** Successive divisions by 2
- **Method 2:** Successive subtractions
- **Method 3:** Convert to base 8 first
- **Method 4:** Convert to base 16 first



Converting a large decimal number to binary

Example:

$(25693)_{10} = (110010001011101)_2$

- **Method 1:** Successive divisions by 2 => **15 division operations**
- **Method 2:** Successive subtractions => **8 subtraction operations**
- **Method 3:** Convert to base 8 first

$$(25693)_{10} = (62135)_8 = (110\ 010\ 001\ 011\ 101)_2$$

Divisions successives par 8 => **5 division operations.**

- **Method 4:** Convert to base 16 first

(25693)10 = (645D)16 = (0110 0100 0101 1101)2

Successive divisions by 16=> **4 division operations**

Converting a large decimal number to binary

To convert a large decimal number to binary, it's recommended to initially convert it to **octal** or **hexadecimal**, and then proceed to binary conversion => to reduce the number of operations (division or subtraction).

In general:

Method	Successive divisions	Successive subtractions	Convert to base 8	Convert to base 16
Number of Arithmetic Operations	Nombre des bits	Number of 1 bits	Number of bits per 3	Number of bits per 4

Binary Addition

Whenever the result **exceeds 1**, it produces a **carryover** to the **adjacent column** with a **higher weight**.

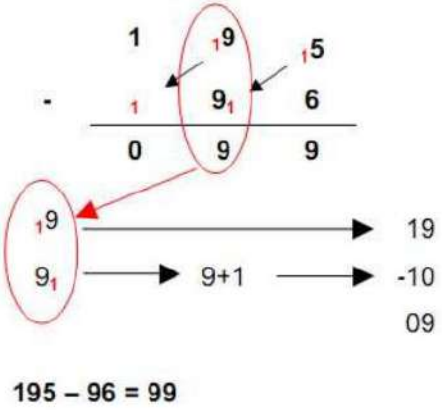
$A + B = 203 + 158 = 361$

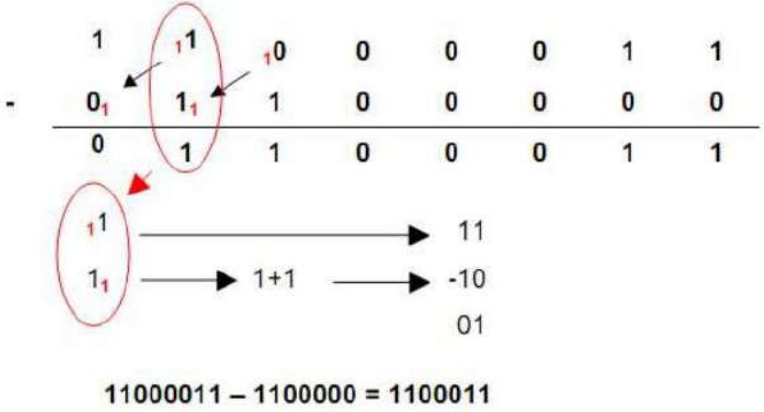
R	1			1	1	1	1		
A		1	1	0	0	1	0	1	1
+									
B		1	0	0	1	1	1	1	0
S	1	0	1	1	0	1	0	0	1

a	b	Sum	retainer
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Binary Subtraction

If the digit being subtracted has a **lower** numerical value than the subtracting digit, we **borrow** from the **column of higher weight.**





a	b	difference	retainer
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

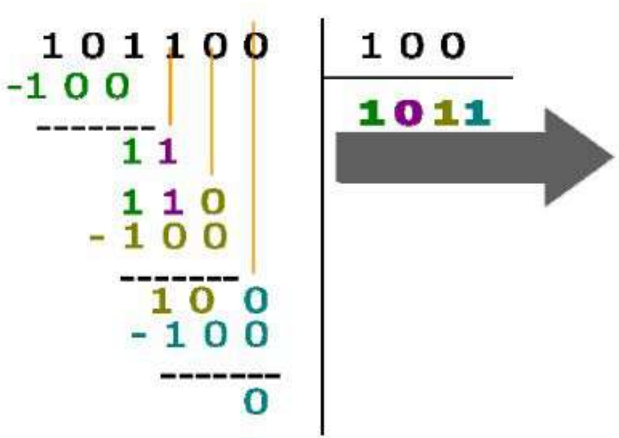
Binary Multiplication

Successive addition of the multiplican such that **multiplication by 0** results in a result equal to **0** and **multiplication by 1** results in the **copying of the multiplicand.**

```
            1 0 1 1
      x       1 1 0
        -----------
           1
            0 0 0 0
    1 1
          1 0 1 1 .
  1
    1 0 1 1 . .
  -------------
  1 0 0 0 0 1 0
```

Binary Division

Successive subtraction of dividend divisor.



Arithmetic Operations in a numeral system with any base

Performing the four basic arithmetic operations in a numeral system with a base other than 10 or 2 can be challenging. In such cases, when we have two numbers N1 and N2 with different bases, B1 and B2 respectively, it's helpful to convert them to either decimal or binary before proceeding with the desired operation

Exercise:

We have two numbers **A** and **B** represented in three positions as follows:

$$\mathbf{A = (a_3\ a_2\ a_1)_5;\ B = (b_3\ b_2\ b_1)_7}$$

1. What are the possible values for the coefficients a_i, b_i?

2. Knowing that $\mathbf{A + B = (138)_9}$, $\mathbf{A - B = (200)_6}$, Find the values of the coefficients a_i, b_i.

3. Transform A and B into binary then calculate A+B, A-B, A * (B/100), A/B.

Solution:

1. a_i<5 so, $a_i \in \{0, 1, 2, 3, 4\}$; b_i<7 so, $b_i \in \{0, 1, 2, 3, 4, 5, 6\}$;

2. $\begin{cases} A+B= \mathbf{(138)_9} \\ A-B= \mathbf{(200)_6} \end{cases}$ The sum of these two equations gives : $2A= \mathbf{(138)_9} + \mathbf{(200)_6}$

You must convert the two operands to the same base to perform the operation.

The best choice is to convert them to a decimal base.

Solution:

$$(138)_9 = 1\times 9^2 + 3\times 9^1 + 8\times 9^0 = (116)_{10}$$

$$(200)_6 = 2\times 6^2 + 0\times 6^1 + 0\times 6^0 = (72)_{10}$$

$$2A = 116 + 72 \leftrightarrow A = (94)_{10}$$

$$B = 116 - A \leftrightarrow B = (22)_{10}$$

$$\begin{cases} A = (94)_{10} = (??)_5 \\ B = (22)_{10} = (??)_7 \end{cases}$$

By the method of successive divisions, we find:

$$\begin{cases} A = (334)_5 \\ B = (031)_7 \end{cases}$$

By identification: $a_3 = 3 \;;\; a_2 = 3 \;;\; a_1 = 4$

$b_3 = 0 \;;\; b_2 = 3 \;;\; b_1 = 1$

Solution:

3.
$A = (94)_{10} = (1011110)_2$
$B = (22)_{10} = (10110)_2$

A	1	$^{1}0$	$^{1}1$	$^{1}1$	$^{1}1$	1	0
+							
B	0	0	1	0	1	1	0
=	1	1	1	0	1	0	0

A	1	0	1	1	1	1	0
-							
B	0	0	1	0	1	1	0
=	1	0	0	1	0	0	0

Solution:

			B						
1	0	1	1	0.	0	1	0	0	
		1	1			1	0	1.	1
		1	1	0					
			1	0.	0				

A					1	0	1	1	1	1	0
*											
B/100								1	0	1,	1
=											
					$^{11}1$	$^{0}0$	$^{1}1$	$^{1}1$	1	1	0
+				$^{0}1$	0	1	1	1	1	0	.
+			$^{11}0$	0	0	0	0	0	0	.	.
+	^{1}	$^{1}1$	0	1	1	1	1	0	.	.	.
=	1	0	0	0	0	0	0	1	0	1,	0

Solution:

A	B
1 0 1 1 1 1 0, 0 0	1 0 1 1 0
1	1 0 0, 0 0 1
1 1	
1 1 $_1$0 $_1$0 0	
- 1 $_1$0 $_1$1 1 0	
0 0 0 1 0	
....	

Ministry of Higher Education and Scientific Research
University of Larbi Ben M'Hidi, Oum El Bouaghi
Faculty of Exact Sciences and Natural and Life Sciences
Department of Mathematics and Computer Science

Computer Structure 1

Presented by: **Dr. NASRI/A**
nasri.ahlem1988@gmail.com
nasri.ahlem@univ-oeb.dz

2023-2024

Chapter 2: Internal Data Representation

1 **Introduction**

- introduce how information is represented in machine (Binary code)

2 **Binary coding**

- The reflected binary code
- The DCB code
- The code exceeds three.

3 **Character representation:**

- Code EBCDIC
- Code ASCII
- Code UTF.

4 **Representation of numbers**

- Integer
- Real

Types of information processed by the computer:

Numbers, instructions, images, animated image sequences,sounds, etc, always represented in binary form.

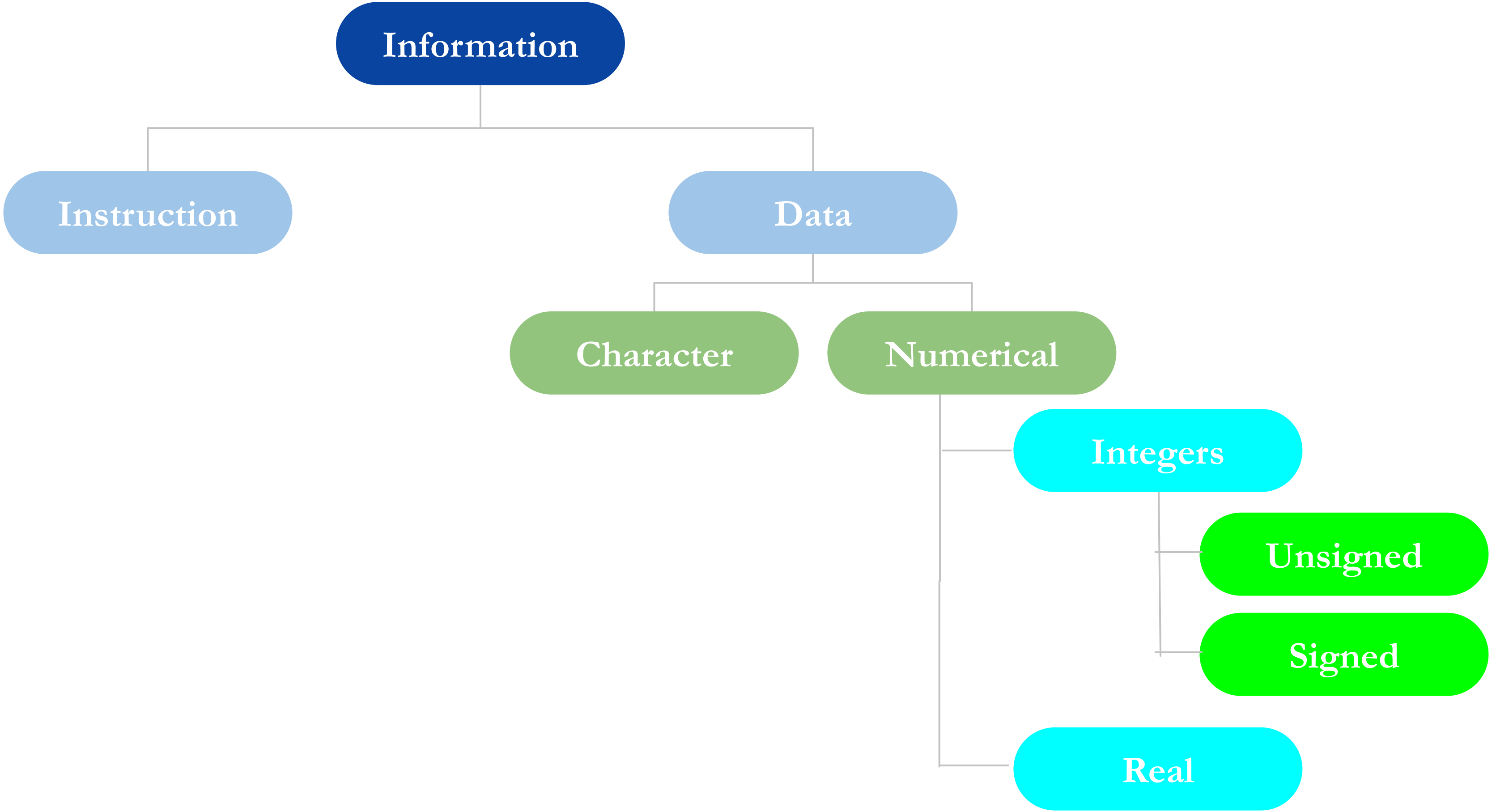
Advantages of binary:

- easy to achieve technically using bistables (systems in two equilibrium states: the existence or not of an electrical voltage +5V=1 and 0V=0).
- fundamental operations that are simple to perform, in the form of logic circuits.

The computer's **memory is limited** => **the bit sequence** which represents the information **is also limited**

Internal Data Representation

number of bits on which we must represent this data


Information
Instruction
Data
Character
Numerical
Integers
Unsigned
Signed
Real

1. The reflected binary code (GRAY code)

- **Gray code**, also known as **reflected binary code**, is a binary numeral system with a unique property that distinguishes it from traditional binary representation.

- In Gray code, adjacent numbers differ by **only one bit**, which makes it particularly useful in applications where you want to minimize errors during transitions between values (Karnaugh's tables, input output circuit, optical encoders and analog/digital converters, …)

1. The reflected binary code (GRAY code)

Principle:

1. Start with the leftmost (most significant) bit of the Gray code, which is the same as the leftmost bit in the binary representation.
2. Move from left to right through each bit in the Gray code.
3. For each bit, calculate the corresponding bit in the binary representation using the following instructions:
   - If the current bit in the Gray code is 0, keep the previous binary bit as it is.
   - If the current bit in the Gray code is 1, flip the previous binary bit.
4. Continue this process until you have converted all bits from Gray code to binary code.

1. The reflected binary code (GRAY code)

Example: We want to represent the first 8 numbers using the gray code.

Decimal Representation	Binary Representation	a	b	c
0	000	0	0	0
1	001	0	0	1
2	010	0	1	1
3	011	0	1	0
4	100	1	1	0
5	101	1	1	1
6	110	1	0	1
7	111	1	0	0

Mirror level 1

Mirror level 2

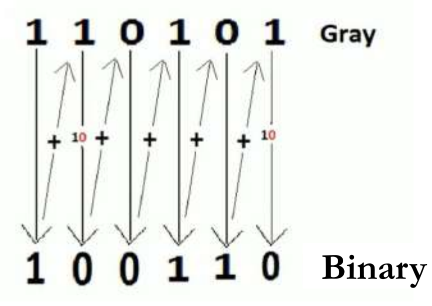
Gray Representation

1. The reflected binary code (GRAY code)

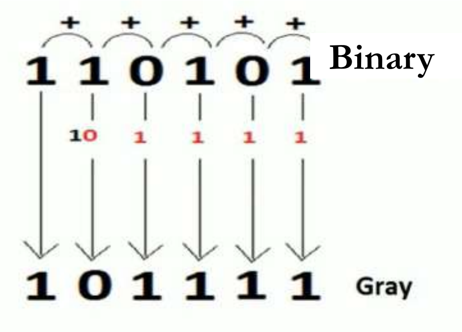
Converting GRAY to Binary

$(110101)_{Gray} = ?$

$(110101)_{Gray} = (100110)_2$



Converting Binary to GRAY



$(110101)_2 = ?$ $(110101)_2 = (101111)_{Gray}$

2. The Decimal Coded Binary code(DCB code)

Principle:

This is the most used code. Its principle is based on associating a binary code with each decimal digit on 4 bits.

Example:

$(512)_{10} = (1000000000)_2 = (0101\ 0001\ 0010)_{DCB}$

nbr of digit in decimal = 10

nbr of numbers we can represented using 4 bits = $2^4 = 16$

There are 6 unused configurations

Decimal	DCB	
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	
/	1010	Not used
/	1011	
/	1100	
/	1101	
/	1110	
/	1111	

3. The code exceeds three

Principale:

Exceeds three code is very similar to the DCB code. Its principle is based on associating each decimal digit with its binary equivalent plus 3.

Example:

$$(512)_{10} = (1000000000)_2 = [(0101+ 0011)\ (0001+0011)\ (0010+0011)]_{\text{Exceeds 3}}$$

$$= (1000\ 0100\ 0101)_{\text{Exceeds 3}}$$

1. Representation of Unsigned Integers:

An unsigned integer is represented by its binary equivalent and the non-significant bits are replaced by zeros.

Example: On 8-bit sequences

8 is represented by 00001000 138 is represented by 10001010

Evaluation:

The evaluation of a number represented in binary is done by converting the representation from the binary system to the decimal system (polynomial expansion).

Range of values:

The range of numbers that can be represented in n-bit binary is **$[0, 2^n-1]$**

The problem is how to tell the machine that a number is positive or negative??????

An **unsigned integer** is represented by a machine like its **equivalent in binary** (as seen in the previous section).
On the other hand, to represent **a signed integer**, there are several methods:

1. Signed absolute value
2. One's Complement
3. Two's Complement
4. With Excess

2. Representation of Signed Integers:

A- Signed absolute value

A Signed integer is represented on n bits by the binary equivalent of its absolute value on **(n-1)** bits and **the nth bit** represents **the sign** of the number.

By convention, **the positive sign** is represented by **0** and **the negative sign** is represented by **1**.

Example: On 4 bits 1 $\boxed{0001}$ 0 $\boxed{0001}$

Sign absolute value

representation of +1 on 4 bits

representation of -1 on 4 bits

2. Representation of Signed Integers:

A- Signed absolute value

Evaluation: A (a_n, a_{n-1}, a_{n-2}…. $a_2a_1a_0$) is the SAV representation of N in decimal. The evaluation of N is done according to the following algorithm:

If a_n=0 then (* the number is positive *)

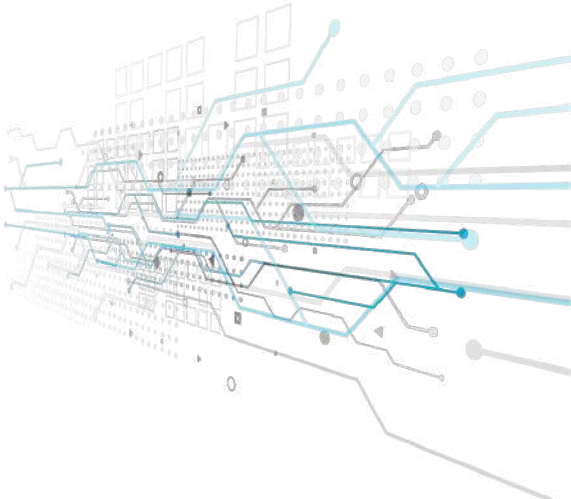
N ⟵ + Decimal Conversion (A)

Else (*the number is negative*)

$a_n \leftarrow 0$

N ⟵ - Decimal Conversion (A)

End



2. Representation of Signed Integers:

A- Signed absolute value

Range of values:

The range of numbers that can be represented in n-bit binary is $[-(2^{n-1}-1), +2^{n-1}-1]$

Example: On 3 bits, we have

The range of numbers

$-3 \leq N \leq +3$

$-(4-1) \leq N \leq +(4-1)$

$-(2^2-1) \leq N \leq +(2^2-1)$

$-(2^{(3-1)}-1) \leq N \leq +(2^{(3-1)}-1)$

NB: we have 2 representation of 0 (-0 and +0)!

SAV Value	Binary Value	Decimal Value
000	+ 00	+ 0
001	+ 01	+ 1
010	+ 10	+ 2
011	+ 11	+ 3
100	- 00	- 0
101	- 01	- 1
110	- 10	- 2
111	- 11	- 3

2. Representation of Signed Integers:

B- One's Complement

- A positive integer is represented by its binary equivalent on (n-1) bits the n^{th} bit represents the sign of the number which is 0.
- A negative integer is represented by its **one's complement.**

We call the one's complement of a number N the number N' such that: **N+N'= 2^n-1** where n is the number of bits of the representation of the number N.

Example:

N=1010. Its One's complement is: N'= $(2^4 - 1)$-N

$N' = (16-1)_{10} - (1010)_2 = (15)_{10} - (1010)_2 = (1111)_2 - (1010)_2 = (0101)_2 = (5)_{10}$

2. Representation of Signed Integers:

B- One’s Complement

NB: To find the one's complement of a number, simply

1. First write the absolute value of the number (|N|) on n bits.
2. Then invert all the bits of this number: if the bit is at 1 put 0 in its place and if the bit is at 0 put in its place a 1.

CA1(CA1(N))= N

On 4 bits

$$\begin{array}{cccc} \boxed{1} & 0 & 1 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \boxed{0} & 1 & 0 & 1 \end{array}$$

bit of sign

On 5 bits

$$\begin{array}{ccccc} 0 & 1 & 0 & 1 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 0 & 1 & 0 & 1 \end{array}$$

2. Representation of Signed Integers:

B- One's Complement

Evaluation: A (a_n, a_{n-1}, a_{n-2}.... $a_2a_1a_0$) is the 1C representation of decimal N.
The evaluation of N is done according to the following algorithm:

If a_n=0 then (* the number is positive *)
 N ⟵ + Decimal Conversion (A)
 Else (*the number is negative*)
 A ⟵ 1C(A)
 N ⟵ - Decimal Conversion (A)
End

2. Representation of Signed Integers:

B- One's Complement

Range of values:

The range of numbers that can be represented in n-bit binary is $[-(2^{n-1}-1), +2^{n-1}-1]$

Example: On 3 bits, we have

The range of numbers [-3,+3]

$-3 \leq N \leq +3$

$-(4-1) \leq N \leq +(4-1)$

$-(2^2-1) \leq N \leq +(2^2-1)$

$-(2^{(3-1)}-1) \leq N \leq +(2^{(3-1)}-1)$

NB: we have 2 representation of 0 (-0 and +0)!

1C Value	Binary Value	Decimal Value
000	+000	+0
001	+001	+1
010	+010	+2
011	+011	+3
100	-011	-3
101	-010	-2
110	-001	-1
111	-000	-0

2. Representation of Signed Integers:

C- Two's Complement

- A positive integer is represented by its binary equivalent on (n-1) bits the n^{th} bit represents the sign of the number which is 0.
- A negative integer is represented by its **tow's complement.**

We call the one's complement of a number N the number N' such that: **N'= 1C(N)+1**
where n is the number of bits of the representation of the number N.

N=1010 on 4 bits
CA1(N)=0101
CA2(N) = **CA1(N)+1** = 0101 + 1 = 0110

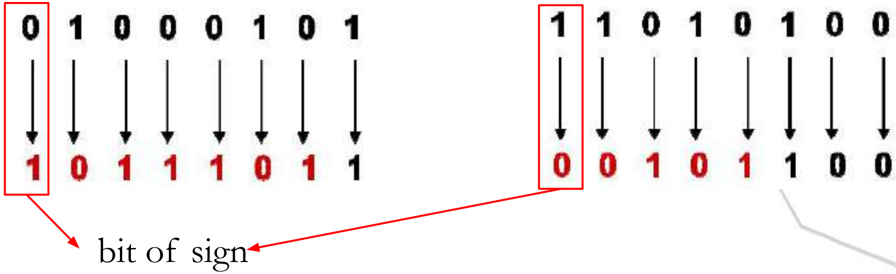
2. Representation of Signed Integers:

C- Two's Complement

NB: To find the one's complement of a number, simply

1. First write the absolute value of the number (|N|) on n bits.
2. Next, traverse the bits of this number starting from the lowest weight bit to the first 1 and invert the other bits that come after.

CA2(CA2(N))= N



2. Representation of Signed Integers:
C- Two's Complement

Evaluation: A $(a_n, a_{n-1}, a_{n-2} \dots a_2 a_1 a_0)$ is the 2C representation of decimal N.
The evaluation of N is done according to the following algorithm:

If $a_n = 0$ then (* the number is positive *)
  N ⟵ + Decimal Conversion (A)
  Else (*the number is negative*)
  A ⟵ 2C(A)
  N ⟵ - Decimal Conversion (A)
End

Exercise:

Use these different codings to code the following relative integers on 8 bits (if possible):

Decimal	Binary	Gray	DCB	exceeds 3	SAV	C1	C2
5							
-5							
32							
-17							
222							

2. Representation of Signed Integers:

D- Representation with excess

- Every number N is represented by the binary equivalent of the characteristic **C** such that: C **= N + Excess**.
- The excess **E** is chosen so that the sum C is always positive.
- In general, this representation is used to present the exponents of real numbers (IEEE 754 standard) with Excess = $2^{n-1}-1$.

Exemple: On 8 bit sequences: E= $2^7-1=127$

10 is represented by 10+127= $(137)_{10} = (10001001)_2$

-10 is represented by -10+127= $(117)_{10} = (01110101)_2$

2. Representation of Signed Integers:

D- Representation with excess

Evaluation:

The evaluation of a number represented with excess is done using the following relationship:

$$C = N + 2^{n-1} - 1 \leftrightarrow N = C - 2^{n-1} + 1$$

3. Representation Real numbers:

- A real number is made up of two parts: **the integer part** and the **fractional part** (the two parts are separated by a decimal point).

- The problem is how to tell the machine the position of the decimal point.??

- There are two methods for representing real numbers:

  - **Fixed point:** the position of the point is fixed
  - **Floating point:** the position of the point changes (dynamic)

3. Representation Real numbers:

A- Fixed point:

In this representation, the Integer Part (IP) is represented on **e** bits and the Fractional Part (FP) on **f** bits, in addition one bit is used for the sign.

Sign 1 bit	Integer Part (IP) e bits	Fractional Part (FP) f bits

Example: if e=3 and f=2 we will have the following values:

Binary values			Decimal values
Sign	IP	FP	
0	000	00	+ 0,0
0	000	01	+ 0,25
0	000	10	+ 0,5
0	000	11	+ 0,75
0	001	00	+ 1,0
……	…..	….	….

NB: In this representation the values are limited and we do not have a large precision !!

3. Representation Real numbers:

B- Floating point:

Each real number can be written as follows: $(-1)^S * (0,M) * B^E$

Where S: Sign (positive = 0, negative = 1), M: Mantissa, B: Base, E: Exponent.

S	E	M

Example:

$$15.6 = +\,0.156 * 10^{+2} = (-1)^0 * 0.156 * 10^{+2}$$

$$-(110{,}101)_2 = -\,(0{,}110101)_2 * 2^{+3} = (-1)^1 * (0{,}110101)_2 * 2^{+3}$$

$$(0{,}00101)_2 = +\,(0{,}101)_2 * 2^{-2} = (-1)^0 * (0{,}101)_2 * 2^{-2}$$

3. Representation Real numbers:

B- Floating point: Standard IEEE 754 (1985)

In this presentation, the real number can be written as follows:

$$(-1)^S \times (0,M) \times 2^E = (-1)^S \times (1,M_n) \times 2^E$$

S (1 bit)	E (e bits)	M_n (m bits)

Mn: Mantissa normalized to base 2 with a hidden bit. Indeed, the decimal point is placed after the most significant bit at 1,

for example: 11.01 => 1.101 => Mn= 101

- E: Exponent coded with excess 2^{e-1} -1

3. Representation Real numbers:

B- Floating point: Standard IEEE 754 (1985)

	32 Bits	64 Bits
Number of Sign bits (S)	1	1
Number of exponent bits (Eb)	8	11
Number of bits of the mantissa (Mn)	23	52
Exponent coding	with Excess $2^7 – 1 = 127$	with Excess $2^{10} – 1 = 1023$

3. Representation Real numbers:

B- Floating point: Standard IEEE 754 (1985)

Example: Code the number N = $(35.5)_{10}$ according to IEEE 754-32:

3. Representation Real numbers:

B- Floating point: Standard IEEE 754 (1985)

Example: Code the number $N = (35.5)_{10}$ according to IEEE 754-32:

- Converting N to binary: N = 100011.1
- Exponential form of N: $N = 1.000111 \times 2^5$
- Sign bit S = 0
  - Mantissa = 00011100000000000000000
  - Exponent coding: $C = E + 127 = 5 + 127 = 132 = (10000100)_2$
- Overall representation: (0 10000100 00011100000000000000000)

3. Representation of Characters:

The characters include : upper and lower case alphabetical letters (A..Z, a..z), numbers (0..9), punctuation (?,!, ..), special characters(%, @,..) and other symbols (>, ;, ..).

A- ASCII Code : (American Standard Code for Information Interchange)

ASCII code of 'A' is $(100\ 0001)_2 = (41)_{16} = (65)_{10}$

ASCII CONTROL CODE CHART

b7 b6 b5 / BITS / b4 b3 b2 b1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		SYMBOLS NUMBERS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL 0 0	16 DLE 10 20	32 SP 20 40	48 0 30 60	64 @ 40 100	80 P 50 120	96 ' 60 140	112 p 70 160
0 0 0 1	1 SOH 1 1	17 DC1 11 21	33 ! 21 41	49 1 31 61	65 A 41 101	81 Q 51 121	97 a 61 141	113 q 71 161
0 0 1 0	2 STX 2 2	18 DC2 12 22	34 " 22 42	50 2 32 62	66 B 42 102	82 R 52 122	98 b 62 142	114 r 72 162
0 0 1 1	3 ETX 3 3	19 DC3 13 23	35 # 23 43	51 3 33 63	67 C 43 103	83 S 53 123	99 c 63 143	115 s 73 163
0 1 0 0	4 EOT 4 4	20 DC4 14 24	36 $ 24 44	52 4 34 64	68 D 44 104	84 T 54 124	100 d 64 144	116 t 74 164
0 1 0 1	5 ENQ 5 5	21 NAK 15 25	37 % 25 45	53 5 35 65	69 E 45 105	85 U 55 125	101 e 65 145	117 u 75 165
0 1 1 0	6 ACK 6 6	22 SYN 16 26	38 & 26 46	54 6 36 66	70 F 46 106	86 V 56 126	102 f 66 146	118 v 76 166
0 1 1 1	7 BEL 7 7	23 ETB 17 27	39 ' 27 47	55 7 37 67	71 G 47 107	87 W 57 127	103 g 67 147	119 w 77 167
1 0 0 0	8 BS 8 10	24 CAN 18 30	40 (28 50	56 8 38 70	72 H 48 110	88 X 58 130	104 h 68 150	120 x 78 170
1 0 0 1	9 HT 9 11	25 EM 19 31	41) 29 51	57 9 39 71	73 I 49 111	89 Y 59 131	105 i 69 151	121 y 79 171
1 0 1 0	10 LF A 12	26 SUB 1A 32	42 * 2A 52	58 : 3A 72	74 J 4A 112	90 Z 5A 132	106 j 6A 152	122 z 7A 172
1 0 1 1	11 VT B 13	27 ESC 1B 33	43 + 2B 53	59 ; 3B 73	75 K 4B 113	91 [5B 133	107 k 6B 153	123 { 7B 173
1 1 0 0	12 FF C 14	28 FS 1C 34	44 , 2C 54	60 < 3C 74	76 L 4C 114	92 \ 5C 134	108 l 6C 154	124 | 7C 174
1 1 0 1	13 CR D 15	29 GS 1D 35	45 - 2D 55	61 = 3D 75	77 M 4D 115	93] 5D 135	109 m 6D 155	125 } 7D 175
1 1 1 0	14 SO E 16	30 RS 1E 36	46 . 2E 56	62 > 3E 76	78 N 4E 116	94 ^ 5E 136	110 n 6E 156	126 ~ 7E 176

Ministry of Higher Education and Scientific Research
University of Larbi Ben M'Hidi, Oum El Bouaghi
Faculty of Exact Sciences and Natural and Life Sciences
Department of Mathematics and Computer Science

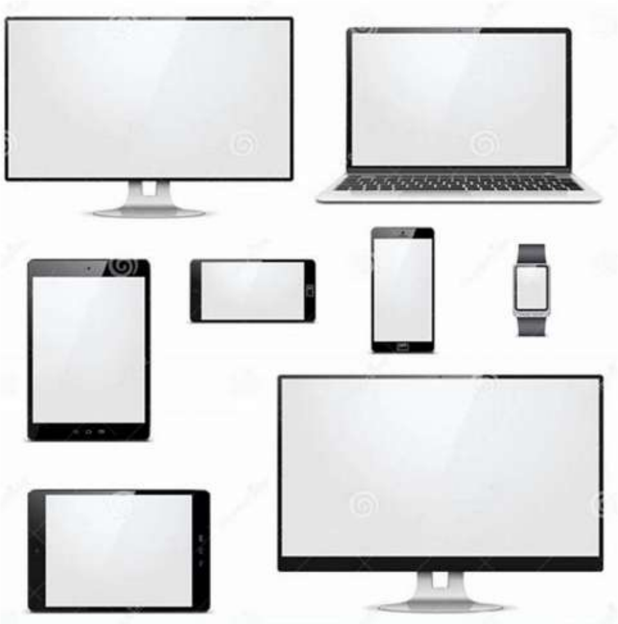
Computer Structure 1

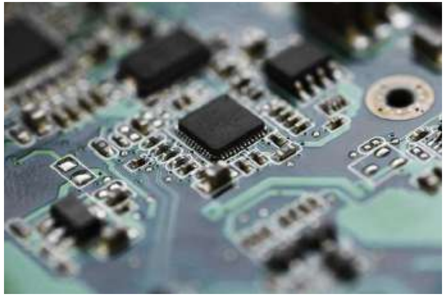
Presented by: **Dr. NASRI/A**
nasri.ahlem1988@gmail.com
nasri.ahlem@univ-oeb.dz

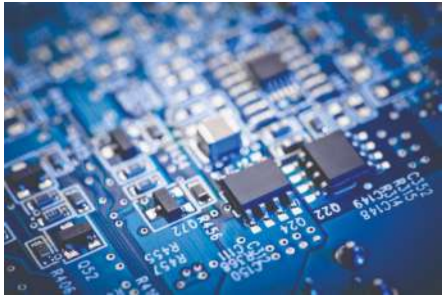
2023-2024

Chapter 3: Boolean Algebra

Introduction







Introduction

- To design and create such a circuit, we must have the mathematical model of its function performed. The mathematical model used is that of Boolean algebra (named after the English mathematician George Boole 1915 - 1864).

- George Boole developed an algebra for manipulating logical propositions using mathematical equations where the statements TRUE and FALSE are represented by the values 1 and 0, while that the AND and OR operators become algebraic multiplication and addition operators.

- Boolean algebra concerns the logic of binary systems.

A- Logical Variable:

A logical (or Boolean) variable is a variable that can take either the value 0 or the value 1. Generally, it is expressed by a single uppercase alphabetical character (A, B, S, etc.)

B- Logical Operators:

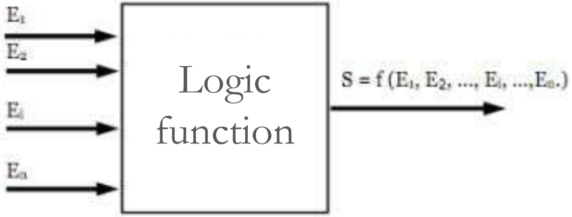
In Boolean Algebra, there are three basic operators: NOT, AND, OR.

C- Logic Gates:

A logic gate is an elementary electronic circuit enabling the function of a logic operator to be carried out.

D- Logic Function

It takes one or more Boolean variables as input, according to which it returns a Boolean value (1 or 0) as output. It is presented either by its logical expression or by its truth table.



D1- Logical Expression:

It is a combination of several variables via logical operators as well as parentheses.

D2- Truth table:

If a logical function has **n logical variables** then the truth table has:

- **(n+1) columns**: n input variables and one output variable
- **2^n lines**: In fact, the function has 2^n Boolean value combinations.

D- Logic Function

Exemple: The function F of three variables A, B and C

Logical Expression: F (A, B, C) = AB + C

Truth table of 4 columns and 8 rows

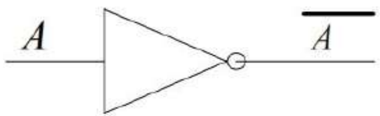
A	B	C	F(A,B,C)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1- Basic Operators:

All functions can be expressed using the three basic operators: NOT, AND and OR.

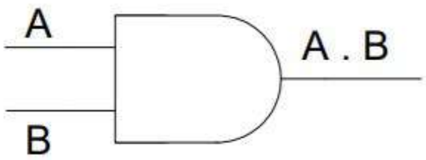
1.1- The NOT Operator (Negation):

NOT is a unary operator (a single variable) which has the role of inverting the value of a logical variable.

<table>
<tr><th>Representation</th><th>Truth table</th><th>logical gate</th></tr>
<tr>
<td>$F(A) = Not(A) = \overline{A}$</td>
<td>
<table>
<tr><td>A</td><td>$\overline{A}$</td></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td></tr>
</table>
</td>
<td>
</td>
</tr>
</table>

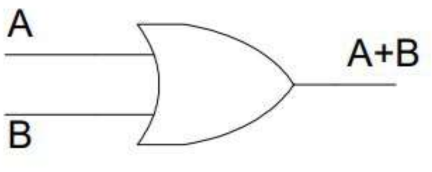
1.2- And Operator (Conjunction)

It is defined as follows: a AND b is TRUE if and only if a is TRUE and b is TRUE. This law is also denoted by a point '.'

<table>
<tr><th>Representation</th><th>Truth table</th><th>logical gate</th></tr>
<tr>
<td>F(A,B)= A*B = A . B = AB</td>
<td>
<table>
<tr><td>A</td><td>B</td><td>AB</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
<td>
</td>
</tr>
</table>

1.3- Or Operator (Disjunction)

It is defined as: a OR b is TRUE if and only if a is TRUE or b is TRUE.
In particular, if a is TRUE and b is also TRUE, then a OR b is true.
This law is also denoted by a plus (+).

Representation	Truth table	logical gate
F(A,B)= A+B	<table><tr><th>A</th><th>B</th><th>A+B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	

1.4- Algebraic Properties:

- In the definition of the AND , OR operators above, we just gave the basic definition with only two logical variables.
However, the AND operator (respectively OR) can produce the logical product (resp. the sum) of several logical variables (Example: A *B * C * D / A + B + C +D).

- To evaluate a logical expression (or function), we start by evaluating the sub-expressions between the parentheses, then the complement (NO), then the logical product (AND) and finally the logical sum (OR).

Associativity	$(a+b)+c=a+(b+c)=a+b+c$ As with normal operations, some parentheses are unnecessary: $(a.b).c=a.(b.c)=a.b.c$
Commutativity	$a+b=b+a$ The order is irrelevant: $a.b=b.a$
Distributivity	$a.(b+c)=a.b+a.c$ $a+(b.c)=(a+b)(a+c)$
Idempotence	$a+a+a+a+a\cdots+a=a$ $a.a.a.a.a\ldots.a=a$
Identity	$a+0=a$ $a.1=a$
Absorption	$a+1=1$ $a.0=0$
Simplification	$a+\overline{a}.b=a+b$ $a.(\overline{a}+b)=a.b$
Redundancy	$a.b+\overline{a}.c+b.c=a.b+\overline{a}.c$
Complimentary	$a=\overline{\overline{a}}$ $a.\overline{a}=0$ $a+\overline{a}=1$

1.4- Algebraic Properties:

Example:

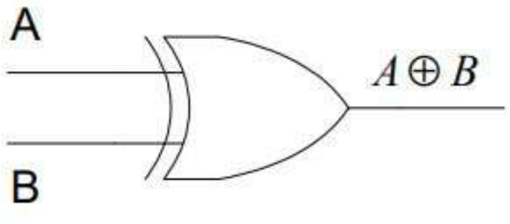
F(A,B)= $\overline{AB}$ (A + B)

A	B	A+B	AB	$\overline{AB}$	F(A,B)
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	0	0

2- Compound Operators:

2.1- The XOR Operator (Exclusive OR)

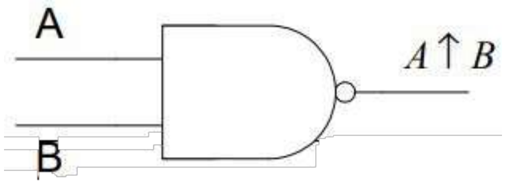
The XOR operator is 1 if one and only one entry is 1.

<table>
<tr><th>Representation</th><th>Truth table</th><th>logical gate</th></tr>
<tr>
<td>$F(A, B) = \overline{A}B + A\overline{B} = A \oplus B$</td>
<td>
<table>
<tr><td>A</td><td>A</td><td>$A \oplus B$</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td>
<td>
</td>
</tr>
</table>

2- Compound Operators:

2.2- The NAND Operator (NOT AND)

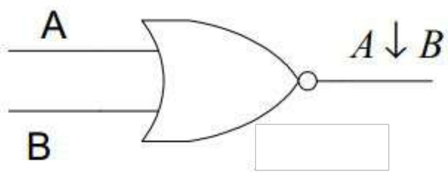
The NAND operator is 0 if all input variables are 1.

<table>
<tr><th>Representation</th><th>Truth table</th><th>logical gate</th></tr>
<tr>
<td>$F(A,B)=\overline{A * B} = A \uparrow B$</td>
<td>
<table>
<tr><td>A</td><td>A</td><td>$A \uparrow B$</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td>
<td>
</td>
</tr>
</table>

2- Compound Operators:

2.3- The NOR Operator (NOT OR)

The NOR operator is equal to 1 if all input variables are 0.

Representation	Truth table	logical gate
$F(A,B) = \overline{A + B} = A \downarrow B$	<table><tr><td>A</td><td>A</td><td>A ↓ B</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	

NB:

Two logical functions are identical if only if:

- We can show via the properties of Boolean algebra that their logical expressions are identical.
- Their truth tables are identical.

Example: prove that $A \oplus B = \overline{AB}\,(A + B)$

Example: prove that $A \oplus B = \overline{AB}\,(A + B)$

1st using truth table:

A	B		$A \oplus B$		$A + B$	AB	$\overline{AB}$	$\overline{AB}\,(A + B)$
0	0		**0**		0	0	1	**0**
0	1		**1**		1	0	1	**1**
1	0		**1**		1	0	1	**1**
1	1		**0**		1	1	0	**0**

Example: prove that $A \oplus B = \overline{AB}\,(A + B)$

2nd using algebraic Properties

$$\overline{AB}\,(A + B) = (\bar{A} + \bar{B})(A + B)$$

$$= \bar{A}A + \bar{A}B + \bar{B}A + \bar{B}B$$

$$= 0 + \bar{A}B + \bar{B}A + 0$$

$$= \bar{A}B + \bar{B}A$$

$$= \bar{A}B + A\bar{B}$$

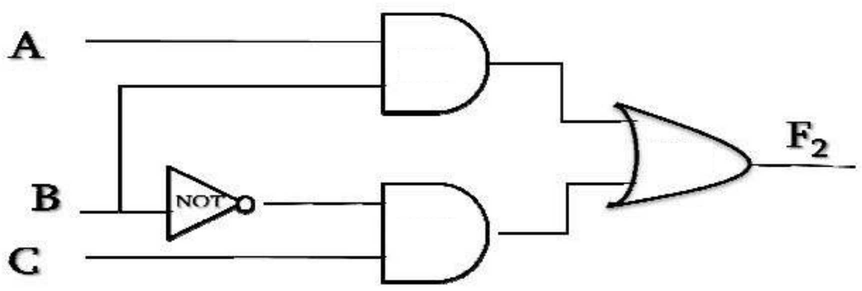
$$= A \oplus B$$

3- Logical Functions:

3.1- Logigram

- The logigram (or logic diagram) is the translation of the logical function into an electronic diagram.
- The principle consists of replacing each logical operator with the logic gate that corresponds to it

Example: $F_2(A,B,C) = AB + \bar{B}C$



3- Logical Functions:

3.2- extracting the logical expression of a function from its truth table

For a logical function with n variables:

- A minterm is the product of n variables (which can be complemented)
- A maxterm is the sum of n variables (which can be complemented)

From the truth table, the logical expression is defined either as:

- **The sum of minterms** where a minterm is determined for each value of the function equal to **1.**
- **The product of the maxterms** where a maxterm is determined for each value of the function equal to **0.**

Example:

Example:

A	B	C	F	Type of term	Expression of term
0	0	0	0	Maxterm	$A+B+C$
0	0	1	0	Maxterm	$A+B+\bar{C}$
0	1	0	0	Maxterm	$A+\bar{B}+C$
0	1	1	1	Minterm	$\bar{A}\,B\,C$
1	0	0	0	Maxterm	$\bar{A}+B+C$
1	0	1	1	Minterm	$A\,\bar{B}\,C$
1	1	0	1	Minterm	$A\,B\,\bar{C}$
1	1	1	1	Minterm	$A\,B\,C$

F = Sum of Minterm

F (A, B, C) = $\bar{A}\,B\,C + A\,\bar{B}\,C + A\,B\,\bar{C} + A\,B\,C$ → 1st canonical form (Disjunctive)

F (A, B, C) = $\sum(011, 101, 110, 111)$ = numerical form

F = Product of Maxterm

F (A, B, C) = $(A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+B+C)$ → 2nd canonical form (Conjunctive)

F (A, B, C) = $\prod(000, 001, 010, 100) = \prod(0, 1, 2, 4)$ → numerical form

3- Logical Functions:

3.3- Canonical Form

We call the canonical form of a function the form where each term of the function includes all the variables. There are two canonical forms:

- First Canonical Form: Disjunctive Form: which is the sum of the minterms; A disjunction. This form is the most used form.
- Second Canonical Form: Conjunctive Form: which is the product of the maxterms: a conjunction.

Note here that the first and second canonical forms are equivalent.

3- Logical Functions:

3.3- Canonical Form

Transition to canonical forms:

We can always reduce any logical function to one of the canonical forms.

To do, we add the missing variables in the terms which do not contain all the variables (the non-canonical terms).

This is possible using the rules of Boolean algebra:

- Multiply a term with an expression that is worth 1
- Add to a term with an expression that is 0
- Subsequently distribute

3- Logical Functions:

3.3- Canonical Form

Transition to canonical forms:

Example: $F_2(A,B,C) = \overline{AB\bar{C} + \bar{A}B}$

Conjonctive Form

$$F_{2C}(A,B,C) = \overline{AB\bar{C} + \bar{A}B}$$

$$= (\overline{AB\bar{C}})(\overline{\bar{A}B})$$

$$= (\bar{A} + \bar{B} + C)(A + \bar{B})$$

$$= (\bar{A} + \bar{B} + C)((A + \bar{B}) + C\bar{C})$$

$$= (\bar{A} + \bar{B} + C)(A + \bar{B} + C)(A + \bar{B} + \bar{C})$$

3- Logical Functions:

3.3- Canonical Form

Transition to canonical forms:

Example: $F_2(A,B,C) = \overline{AB\overline{C} \ + \ \bar{A}B}$

Disjonctive Form

$$F_{2D}(A,B,C) = \overline{AB\overline{C} \ + \ \bar{A}B}$$

$$= \left(\overline{AB\bar{C}}\right)\left(\overline{\bar{A}B}\right)$$

$$= (\bar{A} + \bar{B} + C)(A + \bar{B})$$

$$= \bar{A}A + \bar{A}\bar{B} + \bar{B}A + \bar{B}\bar{B} + CA + C\bar{B}$$

$$= 0 + \bar{A}\bar{B} + A\bar{B} + \bar{B} + AC + \bar{B}C$$

$$= \bar{A}\bar{B}(\boldsymbol{C} + \boldsymbol{\bar{C}}) + A\bar{B}(\boldsymbol{C} + \boldsymbol{\bar{C}}) + (\boldsymbol{A} + \boldsymbol{\bar{A}})\bar{B}(\boldsymbol{C} + \boldsymbol{\bar{C}}) + A(\boldsymbol{B} + \boldsymbol{\bar{B}})C + (\boldsymbol{A} + \boldsymbol{\bar{A}})\bar{B}C$$

$$= \boldsymbol{\bar{A}\bar{B}C} + \boldsymbol{\bar{A}\bar{B}\bar{C}} + \boldsymbol{A\bar{B}C} + \boldsymbol{A\bar{B}\bar{C}} + \boldsymbol{A\bar{B}C} + \boldsymbol{A\bar{B}\bar{C}} + \boldsymbol{\bar{A}\bar{B}C} + \boldsymbol{\bar{A}\bar{B}\bar{C}} + ABC + \boldsymbol{A\bar{B}C} + \boldsymbol{A\bar{B}C} + \boldsymbol{\bar{A}\bar{B}C}$$

$$= \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + ABC$$

3- Logical Functions:

3.4- Simplification

The canonical forms of a logical function are a correct definition of the function, but they can be simplified to:

- Express the same function with as few terms and as simple as possible.
- Perform the function with fewer electronic elements (logic gates).

Simplification can be achieved through either method:

- **The properties of Boolean algebra (Algebraic method).**
- **Map Karnaugh (graphic method).**

3- Logical Functions:

3.4- Simplification

The properties of Boolean algebra (Algebraic method).

- $F_1(A, B) = \bar{A}B + AB$
- $F_2(A, B) = (A + B)(A + \bar{B})$
- $F_3(A, B) = A + AB$
- $F_4(A, B) = A\,(A + B)$

3- Logical Functions:
3.4- Simplification
The properties of Boolean algebra (Algebraic method).

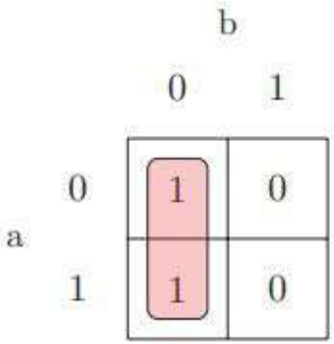
- $F_1(A,B) = \bar{A}B + AB = (\bar{A} + A)B = 1\,B = B$
- $F_2(A,B) = (A+B)(A+\bar{B}) = A\,A + A\,\bar{B} + B\,A + B\,\bar{B} = A + A\bar{B} + A\,B + 0 = A + A\,(\bar{B} + B) = A + A\,1 = A + A = A$
- $F_3(A,B) = A + AB = A\,(1+B) = A\,1 = A$
- $F_4(A,B) = A\,(A+B) = A\,A + A\,B = A + AB$

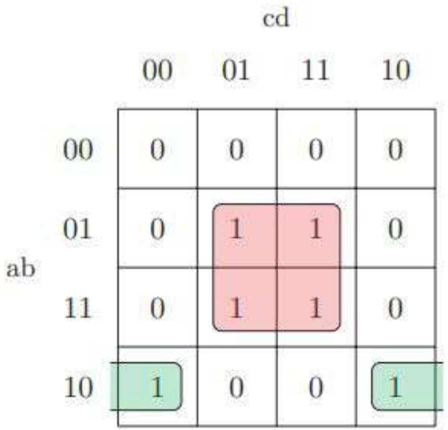
3- Logical Functions:

3.4- Simplification

Map Karnaugh (graphic method)

The Karnaugh map is a graphical tool for simplifying a logic equation or the process of going from a truth table to a corresponding circuit.





3- Logical Functions:
3.4- Simplification
Map Karnaugh (graphic method)

Method:

- Join adjacent "1" in groups of 1, 2, 4, 8,... etc. (2^n, n=0,1,2,... etc)
- The equation of the circuit is given by the sum of the products of the variables which do not change state in each grouping.

- Then, in the previous examples: $S_1 = \overline{b}$ and $S_2 = b.d + a.\overline{b}.\overline{d}$

NB: An output $\overline{S}$ is obtained by the groupings of zeros.

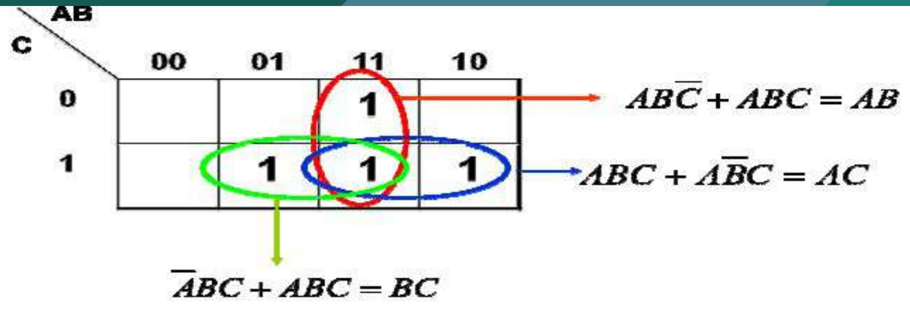
3- Logical Functions:

3.4- Simplification

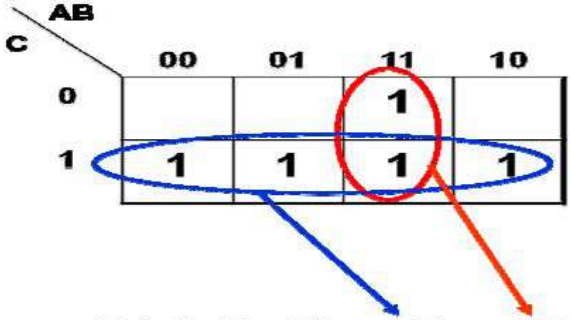
Exemple:

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

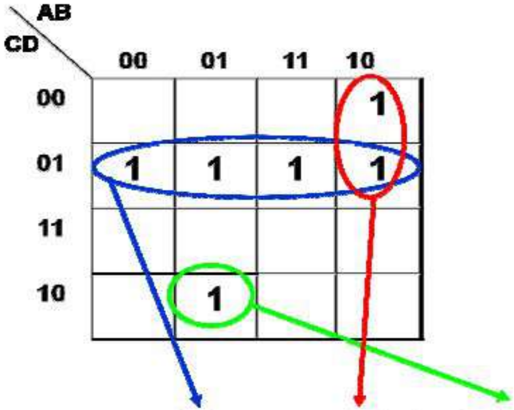
C \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1



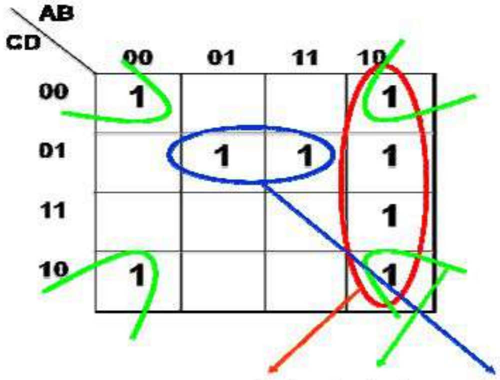
$$F(A,B,C) = AB + AC + BC$$



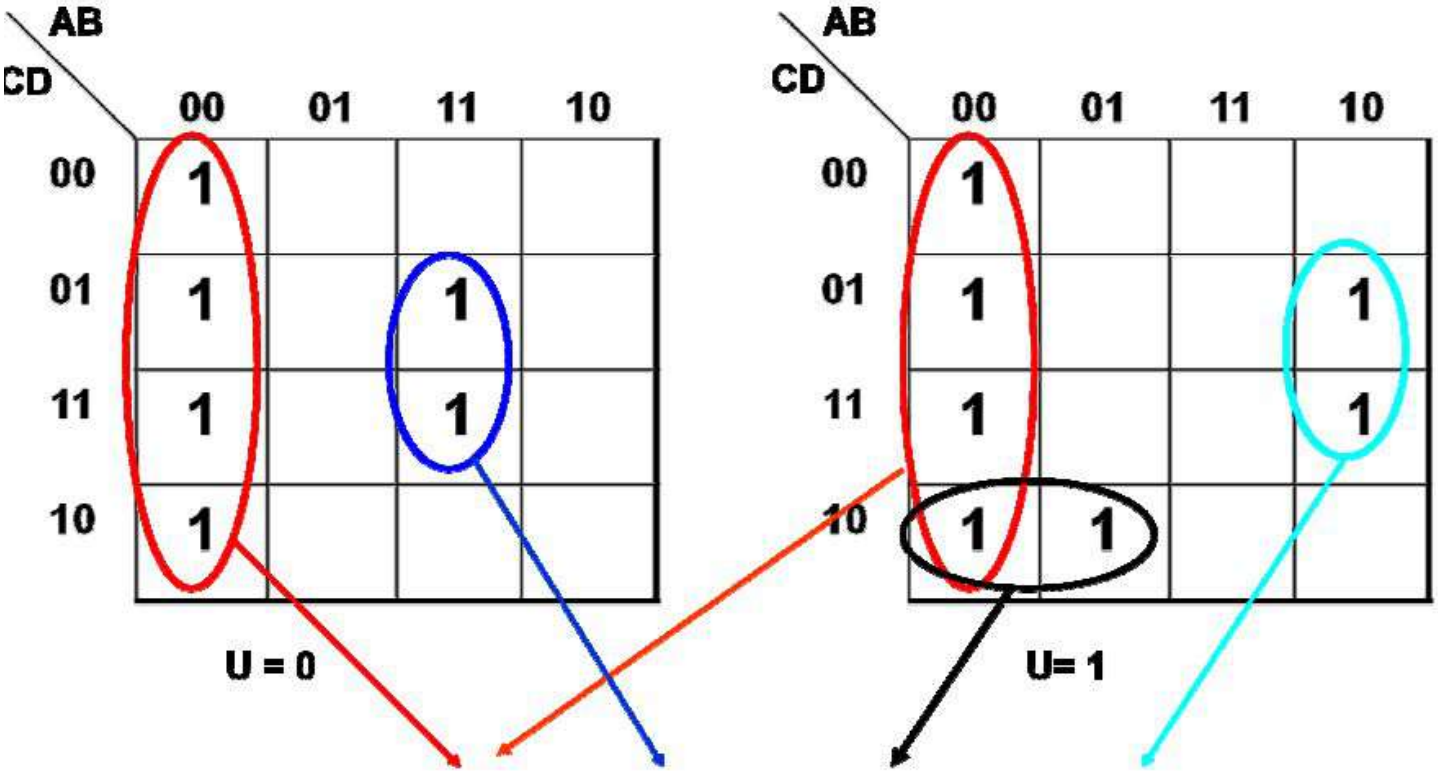
$$F(A,B,C) = C + AB$$



$$F(A,B,C,D) = \overline{C}.D + A.\overline{B}.\overline{C} + \overline{A}.B.C.\overline{D}$$



$$F(A,B,C,D) = A\overline{B} + \overline{B}\overline{D} + B\overline{C}D$$



$$F(A, B, C, D, U) = \overline{A}\,\overline{B} + A.B.D.\overline{U} + \overline{A}.C.\overline{D}.U + A.\overline{B}.D.U$$

3- Logical Functions:
3.4- Simplification
Example with 5 variables:

E = 0				
CD \ AB	00	01	11	10
00				
01				
11				
10				

E = 1				
CD \ AB	00	01	11	10
00				
01				
11				
10				

Map with 5 variables

3- Logical Functions:
3.4- Simplification
Example with 6 variables:

EF = 00				
AB / CD	00	01	11	10
00				
01				
11				
10				

EF = 01				
AB / CD	00	01	11	10
00				
01				
11				
10				

EF = 11				
AB / CD	00	01	11	10
00				
01				
11				
10				

EF = 10				
AB / CD	00	01	11	10
00				
01				
11				
10				

Map with 6 variables

3- Logical Functions:
3.4- Study of a logic function

Steps :

1. Truth table
2. Canonical Forms
3. Simplification (algebric or Karnaugh map)
4. logigram drawing
   (diagram of logic gates)

General Introduction

Exercise 1:

1. Provide a diagram illustrating the internal structure of a computer.
2. How is information presented in memory?
3. What is the basic unit of measurement for information?
4. What is the difference between RAM and ROM?
5. What is the difference between RAM and auxiliary memory?
6. How long does it take to download a 1 MB file using a 1 Mb/s ADSL connection?
7. Classify the following devices according to their type (input, output and input/output):

Keyboard, floppy disk, screen, DVD engraver, DVD player, scanner, hard disk, mouse, printer, modem, flash disk, digital screen, microphone, speaker.

Exercise 2:

Specify the units of measurement in the following data sheet:

- Intel Core™i5 (frequency 3.40 ....., cache memory 4 ....
- Windows 8.1 64 …..
- RAM 4 ………… with frequency of 1333 …… ......
- Hard disk 850 ……, transfer rate 4 ……
- Integrated network card (LAN) : 100 ……
- ADSL connection 2 ……
- WebCam: resolution 12 …….

Exercise 3:

Convert the following units:

- 2,4 GHz= …………… MHz= …………………..Hz.
- 4,7 GB = ……………. MB =........................... KB =........................... Bytes.
- 512 kb/s =...................... kB/s = ……………..Bytes/s.
- 2 TB = ………………..GB =........................... MB.
- 1Mb/s =.......................... kB/s =..........................bytes/s.

Exercise 4:

Yanis owns a flash disk with a storage capacity of 8 GB. He has saved three files on his flash disk with sizes of 1.87 GB, 4096 MB, and 300 MB, respectively. Yanis then lent his flash disk to Anya, requesting her to copy a 2 GB film onto it.

Is it possible for Anya to copy the film to the flash disk?

Why or why not?

By: Dr. A-NASRI

Solution TD0

Exercise 1:

1	**Recall the computer architecture (refer to the course) for a reminder.**
2	**Information is presented in memory as binary digital data, often represented in bits (0 and 1). These bits are organized into groups called bytes (or cells). Each cell has a unique address which allows direct access to any cell.**
3	**The basic unit of measurement for information is the bit (short for binary digit)**
4	RAM**: volatile memory; used for storing data and programs that are actively being used by Operating System, programs and applications; read/write memory; usually has a higher capacity compared to ROM.** ROM**: non-volatile memory; used for storing essential instructions and data such as a computer's BIOS or start instructions; read-only memory**
5	**Auxiliary memory (secondary storage), such as hard drives and SSDs, is used for long-term storage of data, programs, and files . non-volatile storage .** **RAM is much faster .** **RAM has a limited capacity compared to auxiliary memory** **Auxiliary memory is an external removable memory.**
6	**Time = Size/Speed = 1MB/1Mb/s = 1×8Mb 1Mb/s = 8s**
7	**Input:** Keyboard, PlayerDVD, scanner, mouse, microphone. **Output:** screen, printer, speaker. **Input/Ouput:** floppy disk; DVD engraver, hard disk, modem, flash disk, digital screen.

Keyboard, floppy disk, screen, DVD engraver, DVD player, scanner, hard disk, mouse, printer, modem, flash disk, digital screen, microphone, speaker.

Exercise 2:

Specify the units of measurement in the following data sheet:

- Intel Core™i5 (frequency 3.40 **GHz**,cache memory 4 **MB**
- Windows 8.1 64 **bits**
- RAM 4 **GB** with frequency of 1333 **MHz**
- Hard disk 850 GB, transfer rate 4 **MB/s**
- Integrated network card (LAN) : 100 **Mb/s**
- ADSL Connection 2 **Mb/s**

- WebCam: resolution 12 **Mega Pixel.**

Exercise 3:

Convert the following units:

- 2,4 GHz= $\mathbf{2.4 \times 10^3}$ MHz= $\mathbf{2.4 \times 10^9}$ Hz.
- 4,7 GB = $\mathbf{4.7 \times 2^{10}}$ MB = $\mathbf{4.7 \times 2^{20}}$ KB = $\mathbf{4.7 \times 2^{30}}$ Bytes.
- 512 kb/s = **512/8** kB/s = $\mathbf{64 \times 2^{10}}$ Bytes/s.
- 2 TB = $\mathbf{2{\times}2^{10}}$ GB = $\mathbf{2{\times}2^{20}}$ MB.
- 1Mb/s = $\mathbf{1 \times 2^{10}/8}$ kB/s = **1024/8 kB/s**= **128 kB/s** = $\mathbf{128 \times 1024}$ bytes/s.

Exercise 4:

Used space : 1,87 GB+ 4096 MB + 300 MB = 1,87+4,096+0,3 GB = **6.266 GB**
Free space : 8 GB - 6,266 GB= **1,734 GB.**
By comparing the free space with the Film size, we deduce that Anya cannot copy the film to the flash disk (2GB > 1,734 GB).

Numeral systems

Exercise 1:

Here are the given numbers: **1010, 1020, 108141, 2A0GF00, 01AFB, CEE, BAC.**

- Among these numbers, which ones can be the presentation of a number in **base 2, 8, 10 or 16**?
- Give the smallest base in which each number can be written?

Exercise 2:

Make the following conversions:

Base 10	Base 2	Base 8	Base 16	X	Base X
555				9	
120,25				4	
14,5				3	
35,05				7	
	101101			14	
	111.01011			6	
	101.101			3	
	1011011			4	
		37		5	
		1254,1		13	
		276,52		4	
			A2	11	
			A5F,6	12	
			B2CD,2A	7	
				5	32.4
				3	12
				12	24
				9	45

Exercise 3:

Determine the bases in which the following numbers are expressed:

Basex	**Base 10**	**x**
34	22	
75	117	
1110101	117	
24	14	
13	7	
70	56	
1111	40	
402	102	
135	75	
1023	75	

Exercice 4:

Find the solution to the following riddles:

- I am triple the value of 7 in decimal and I am an octal number. Who am I ?
- I am a binary number. If you convert me to decimal, I am double 11. Who am I ?
- I am a palindrome in base 3. If you convert me to decimal, I'm 130. What is my palindrome representation in base 3 ?

Exercise 5:

Determine the digits (x,y) such that the number N is written:

(x2y) in base 6 and (3x2) in base 5.

- Give the number N in the decimal system.

Exercise 6:

A clever hen mastered counting using a base 5 numeral system. She employed five symbols: **C, T, D, E**, and **O.** Each symbol corresponded to a specific numerical value.

- What numerical value did she give to each of these five letters, knowing that to name the decimal number **41346460**; She said "**COT COT CODET**".

Exercise 7:

We have two numbers A and B represented in three positions as follows:

$$\mathbf{A = (a3\ a2\ a1)_5; \qquad B = (b3\ b2\ b1)_7}$$

1. What are the possible values for the coefficients ai, bi?
2. Knowing that $\mathbf{A + B = (138)_9}$, $\mathbf{A - B = (200)_6}$, Find the values of the coefficients ai, bi.
3. Transform A and B into binary then calculate A+B, A-B, A * (B/100), A/B.

Exercise 8:

Perform the following operations:

$(1011.1101)_2 + (11.1)_2 = (?)_2$

$(1011.1101)_2 / (11.1)_2 = (?)_2$

$(1010.0101)_2 - (110.1001)_2 = (?)_2$

$(110)_2 * (1.01)_2 = (?)_2$

$(91B)_{16} + (6F2)_{16} = (?)_8$

$(3.6)_8 * (4.5)_8 = (?)_4$

$(340)_5 - (32)_7 = (?)_6$

Exercise 9:

In a personal computer, the usable memory words have the following hexadecimal addresses: from 0000 to 01FF and from 4001 to 7E00. What is the total decimal number of usable memory words?

Exercise 10:

Use these different codings to code the following relative integers on 8 bits (if possible) then on 16 bits:

Decimal	Binary	SVA	C1	C2	Excess
1					
-1					
-99					
-24					
127					
-128					
405					

Exercise 11: Code the following real numbers according to the IEEE standard 754-32:

$N1 = (-6.53125)_{10}$

$N2 = (-32.625)_{10}$

$N3 = (-\ 11.8561)_{10}$

$N4 = N1 - N2$

Exercice 12:

Convert to decimal the following binary number represented in floating point (IEEE754-32 bits)

Sign	Exponent	Mantissa
1	1000 0010	1010 1000 0000 0000 0000 000
1	1000 0100	1001 0100 0000 0000 0000 000
0	1000 1010	1111 1000 0000 0000 0000 000

By: Dr. A-NASRI

Solution TD2

Exercise 1: (Already seen in cours, it's just a reminder)

	Base 2	Base 8	Base 10	Base 16	Smallest Base
1010	**Yes**	**Yes**	**Yes**	**Yes**	**2**
1020	**No**	**Yes**	**Yes**	**Yes**	**3**
108141	**No**	**No**	**Yes**	**Yes**	**9**
2A0GF00	**No**	**No**	**No**	**No**	**17**
01AFB	**No**	**No**	**No**	**Yes**	**16**
CEE	**No**	**No**	**No**	**Yes**	**15**
BAC	**No**	**No**	**No**	**Yes**	**13**

Exercise 2:
Make the following conversions:

Base 10	**Base 2**	**Base 8**	**Base 16**	**X**	**Base X**
555	**1000101011**	**1053**	**22B**	9	**676**
120,25	**1111000.01**	**170,2**	**78,4**	4	**1320.1**
14,5	**1110.1**	**16.4**	**E,8**	3	**112.11111**
35,05	**100011.00001**	**43.031**	**23,0C**	7	**50.0231**
45	101101	**55**	**2D**	14	**33**
7.34375	111.01011	**7.274**	**7,58**	6	**11.20213**
5.625	101.101	**5.5000**	**5,A**	3	**12.12**
91	1011011	133	**5B**	4	**1123**
31	**11111**	37	**1F**	5	**111**
684.125	**1010101100.001**	1254,1	**2AC.2**	13	**408.18**

190,656	**10111110.1010**	276,52	**BE,A8**	4	**2332.222**
162	**10100010**	**242**	A2	11	**138**
2655,375	**101001011111.011**	**5137.3**	A5F,6	12	**1653.46**
45773,164	**1011001011001101.001**	**131315.124**	B2CD,2A	7	**250310.1101**
17,8	**10001.110011**	**21.63146315**	**11.CCCCCC**	5	32.4
5	**101**	**5**	**5**	3	12
28	**11100**	**34**	**1C**	12	24
41	**101001**	**51**	**29**	9	45

Exercise 3:

Determine the bases in which the following numbers are expressed:

1- $(34)_x = 3*x^1 + 4*x^0 = 22 \Rightarrow 3x+4=22 \Rightarrow 3x = 18$

X=6

2- The same process is applied for the rest …

Base x	**Base 10**	**x**
34	22	6
75	117	16
1110101	117	2
24	14	5
13	7	4
70	56	8
1111	40	3
402	102	5
135	75	7
1023	75	4

Exercice 4:

1- 7*3= 21; Convert 21 to octal: $(21)_{10}=(25)_8$

So, the octal number that is triple the value of 7 in decimal is 25.

2- 11×2=22; Convert 22 to binary: $(22)_{10}=(10110)_2$

So, the binary number that is double 11 in decimal is 10110.

3- convert 130 from decimal to ternary (base 3) => 11211 (palindrome).

Exercise 5:

$(N)_6$= x2y => $x \times 6^2 + 2\times 6^1 + y\times 6^0$ => 36x+12+y

$(N)_5$= 3x2 => $3\times 5^2 + x\times 5^1 + 2\times 5^0$ => 75+5x+2

36x+12+y = 75+5x+2 =>36x-5x=77-12-y => 31x= 65-y

y takes values from 0 to 4 =>

y = 0 => x?

y=1 => x?

y=2 => x?

y=3 => x=2

y=4 => x?

$(N)_6$= x2y => $(223)_6$= $(87)_{10}$

Exercise 6:

Convert 41346460 from decimal to base 5:

$(41346460)_{10}$= $(410\ 410\ 41320)_5$ = $(COT\ COT\ CODET)_5$

C=4

O=1

T=0

D=3

E=2

Exercise 7: (Already seen in cours, it's just a reminder)

1. $a_i < 5$ so, $a_i \in \{0, 1, 2, 3, 4\}$; $b_i < 7$ so, $b_i \in \{0, 1, 2, 3, 4, 5, 6\}$;

2. $\begin{cases} A+B = \mathbf{(138)_9} \\ A\text{-}B = \mathbf{(200)_6} \end{cases}$ The sum of these two equations gives : $2A = \mathbf{(138)_9} + \mathbf{(200)_6}$

You must convert the two operands to the same base to perform the operation.
The best choice is to convert them to a decimal base.

$$\left.\begin{aligned} (138)_9 &= 1\times9^2 + 3\times9^1 + 8\times9^0 = (116)_{10} \\ (200)_6 &= 2\times6^2 + 0\times6^1 + 0\times6^0 = (72)_{10} \end{aligned}\right\} \quad 2A = 116 + 72 \leftrightarrow A = (94)_{10}$$

$$B = 116 - A \leftrightarrow B = (22)_{10}$$

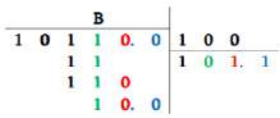
$\begin{cases} A = (94)_{10} = (??)_5 \\ B = (22)_{10} = (??)_7 \end{cases}$ By the method of successive divisions, we find: $\begin{cases} A = (334)_5 \\ B = (031)_7 \end{cases}$

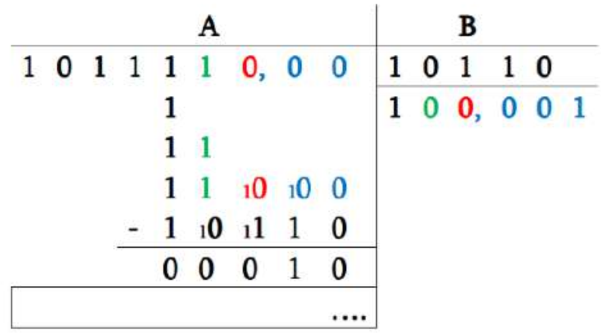
By identification: $a_3 = 3$; $a_2 = 3$; $a_1 = 4$
$b_3 = 0$; $b_2 = 3$; $b_1 = 1$

3. $\begin{cases} A = (94)_{10} = (1011110)_2 \\ B = (22)_{10} = (10110)_2 \end{cases}$

A	1	$^{1}0$	$^{1}1$	$^{1}1$	$^{1}1$	1	0
+							
B	0	0	1	0	1	1	0
=	1	1	1	0	1	0	0

A	1	0	1	1	1	1	0
-							
B	0	0	1	0	1	1	0
=	1	0	0	1	0	0	0





A					1	0	1	1	1	1	0
*											
B/100								1	0	1,	1
=											
					^{11}1	^{0}0	^{1}1	^{1}1	1	1	0
+				^{0}1	0	1	1	1	1	0	.
+			^{11}0	0	0	0	0	0	0	.	.
+	^{1}	^{1}1	0	1	1	1	1	0	.	.	.
=	1	0	0	0	0	0	0	1	0	1,	0

Exercise 8:

$(1011.1101)_2 + (11.1)_2 = (1111.0101)_2$

$(1010.0101)_2 - (110.1001)_2 = (11.11)_2$

$(110)_2 * (1.01)_2 = (111.1)_2$

$(91B)_{16} + (6F2)_{16} = (100D)_{16} = (10015)_8$

Exercise 9:

Decimal	Binary	SVA	C1	C2	Excess
1	00000001	00000001	00000001	00000001	10000000
-1	00000001	10000001	11111110	11111111	1111110
-99	01100011	11100011	10011100	10011101	00011100
-24	00011000	10011000	11100111	11101000	01100111
127	01111111	01111111	01111111	01111111	01111111
-128	0000000010000000	1000000010000000	1111111101111111	1111111110000000	1011101001110001

Exercice 10:

Example:

13.25

Method

- Integer part : $13 \Rightarrow 1101$
- Decimal part : $0,25 \Rightarrow 0,01$
- $(13.25)_{10} = (1101,01)_2$
- Normalization : $1101,01 \times 2^0$ <=> 0.110101×2^4
- Pseudo-normalization IEEE 754 : <=> 1.10101×2^3 (in format of 1,xxxx where xxx = pseudo mantissa)

Decomposition of Number into its various elementss:

- Sign bit: 0 (Number positif)
- Exponent on 8 bits biased by $127 \Rightarrow 3 + 127 = 130 \Rightarrow 1000\ 0010$
- Pseudo mantissa on 23 bits: 1010 1000 0000 0000 0000 0000

Sign	Biased exponent	Pseudo mantissa
0	1000 0010	1010 1000 0000 0000 0000 000

Decimal	Representation in IEEE 754-32 format		
	Sign	Exponent	Mantissa
N1=$(-6.53125)_{10}$	1	10000001	10001000000000000000000
N2 = $(-32.625)_{10}$	1	10000100	10100000000000000000000
N3= $(-11.8561)_{10}$	1	10000011	11100011000001111101000

Exercice 11:

1

Sign	exponent	Mantissa
1	1000 0010	1010 1000 0000 0000 0000 000
-	$130 = 127 + 3 \Rightarrow puissance3$	10101
-	2^3	$\times 1.10101$

The result is $-1.10101 \times 2^3 = (-1101.01)_2 = (-13.25)_{10}$

2

Sign	exponent	Mantissa
1	1000 0100	1001 0100 0000 0000 0000 000
-	$132 = 127 + 5 \Rightarrow puissance\,5$	1001 01
-	2^5	$\times 1.1001\ 01$

Result is $-1.1001\ 01 \times 2^5 = (-110010.1)_2 = (-50.5)_{10}$

3

Sign	exponent	Mantissa
0	10001010	11111000000000000000000
+	$138 = 127 + 11 \Rightarrow puissance\,11$	1111 1
+	2^{11}	$\times 1.1111\ 1$

The result is $+1.1111\ 1 \times 2^{11} = (+1111\ 1100\ 0000)_2 = (+16128)_{10}$

Boolean Algebra

Exercise 1: Draw the truth table of the following expressions:

- $F_1(A,B) = A + \overline{AB}$
- $F_2(A,B) = \overline{AB}\,(A+B)$
- $F_3(A,B,C) = ABC + AB\overline{C} + A\overline{B}C + A\overline{B}\overline{C}$

Exercise 2: Prove the following theorems by the truth table

Idempotence : $a + a + a + ... = a$

Identity $a + 0 = a \quad a.1 = a$

Absorption $a.0 = 0 \; a + 1 = 1$

Complementary $a + \overline{a} = 1 \; a.\overline{a} = 0$

$\overline{a.b} = \overline{a} + \overline{b}$

$\overline{a + b} = \overline{a}.\overline{b}$

Exercise 3: Prove the following equations using the properties of Boolean algebra:

$a + a.b = a$

$a.(a + b) = a$

$a + \overline{a}.b = a + b$

$(a + b)(a + \overline{b}) = a$

Exercise 4: Simplify the following equations using the properties of Boolean algebra:

$(a + b)(a + c)$

$(a + b)(\overline{a} + c)$

$\overline{\overline{a}.b + \overline{\overline{a} + b}}$

Exercise 5: Express the following functions in the first and second canonical form:

$f1(x, y, z) = xy + x\overline{z} + \overline{y}z$

f(a, b, c) = 1 if the count of variables at 1 is even

f(a, b, c, d) = 1 if at least two variables are equal to 1

Exercise 6: Simplify the functions of the previous exercice (exercice 5) using the Karnaugh map.

Exercise 7: Plot the logigrams of the functions of the previous exercice (exercice 5).

Exercise 8: Study the function: $F(x, y, z) = x \oplus (y + z)$

By: Dr. A-NASRI

Boolean Algebra

Exercise 1: Draw the truth table of the following expressions:

A	B	AB	$\overline{AB}$	$F(A,B)=A+\overline{AB}$
0	0	0	1	1
0	1	0	1	1
1	0	0	1	1
1	1	1	0	1

A	B	AB	$\overline{AB}$	A+B	$F(A,B)=\overline{AB}(A+B)$
0	0	0	1	0	0
0	1	0	1	1	1
1	0	0	1	1	1
1	1	1	0	1	0

A	B	C	$\overline{B}$	$\overline{C}$	ABC	$AB\overline{C}$	$A\overline{B}C$	$A\overline{B}\overline{C}$	$F(A,B,C)=ABC+AB\overline{C}+A\overline{B}C+A\overline{B}\overline{C}$
0	0	0	1	1	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	1	1
1	0	1	1	0	0	0	1	0	1
1	1	0	0	1	0	1	0	0	1
1	1	1	0	0	1	0	0	0	1

Exercise 2: Prove the following theorems by the truth table

1 Idempotence : $a + a + a + ... = a$

a	a	a		$a+a+a+a+a+a+a$	$a.a.a.a.a$
0	0	0		0	0
1	1	1		1	1

2 Identity $a + 0 = a$
$a.1 = a$

a	0	1		a+0	a.1
0	0	1		0	0
1	0	1		1	1

3 Absorption $a.0 = 0$
$a + 1 = 1$

a	0	1		$a.0$	$a+1$
0	0	1		0	1
1	0	1		0	1

4 Complementary $a + \overline{a} = 1$
$a.\overline{a} = 0$

a		$a+\overline{a}$	$a.\overline{a}$
0		1	0
1		1	0

$\overline{a.b} = \overline{a} + \overline{b}$

a	b	'a	b'		a.b	$\overline{a.b}$	$\overline{a}+\overline{b}$
0	0	1	1		0	1	1
0	1	1	0		0	1	1
1	0	0	1		0	1	1
1	1	0	0		1	0	0

$\overline{a+b} = \overline{a}.\overline{b}$

a	b	'a	b'	a+b	$\overline{a+b}$	$\overline{a}.\overline{b}$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

Exercise 3: Prove the following equations using the properties of Boolean algebra:

1° a+a.b= a. (1+b) (common factors)
= a.1 (absorption)
= a (identity)

2° a.(a+b)= a.a + a.b (distribution of . over +)
= a+ a.b (idempotence)
= a.(1+b) (common factors)
= a.1 (absorption)
= a (identity)

3°

$a+\overline{a}.b = (a+\overline{a}) . (a+b)$ (distribution of + over .)
$= 1. (a+b)$
$= (a+b)$

4°

$$(a+b).(a+\overline{b}) = a+b\overline{b}$$
$$=a$$

Exercise 4:

1° (a+b)(a+c) = a+(b.c) (distribution of + over .)

2°

$$(a+b).(\overline{a}+c) = a.\overline{a}+a.c+\overline{a}b+ b.c$$
$$=0+ac+\overline{a}b+b.c\,(a+\overline{a})$$
$$=ac+\overline{a}b+abc+\overline{a}bc$$
$$=\overline{a}b(1+c)+ac(1+b)$$
$$=\overline{a}b+ac$$

3°

$$\overline{\overline{a}.b+\overline{\overline{a}+b}}$$

$$\overline{\overline{a}.b+\overline{\overline{a}+b}}$$
$$=(\overline{\overline{a}.b}).(\overline{\overline{\overline{a}+b}})$$
$$=(\overline{\overline{a}}+\overline{b}).(\overline{a}+b)$$
$$=(a+\overline{b})(\overline{a}+b)$$
$$=a.\overline{a}+a.b+\overline{a}.\overline{b}+b.\overline{b}$$
$$=a.b+\overline{a}.\overline{b}$$

Exercise 5:

$\boxed{1}$ $f1(x,y,z) = xy + x\overline{z} + \overline{y}z$

x	y	z		f1	Minterm	Maxterm
0	0	0		0		$(x+y+z)$
0	0	1		1	$\overline{x}.\overline{y}z$	
0	1	0		0		$(x+.\overline{y}+z)$
0	1	1		0		$(x+.\overline{y}+.\overline{z})$
1	0	0		1	$x.\overline{y}.\overline{z}$	
1	0	1		1	$x.\overline{y}z$	
1	1	0		1	$xy.\overline{z}$	
1	1	1		1	xyz	

1st canonical form:
$F1 = \overline{x}.\overline{y}z + x.\overline{y}.\overline{z} + x.\overline{y}z + xy.\overline{z} + xyz$
2nd canonical form
$F1 = (x+y+z)\ (x+\overline{y}+z)(x+\overline{y}+\overline{z})$

$\boxed{2}$ F2(a, b, c) = 1 if the number of variables at 1 is even

a	b	c		f2	Minterm	Maxterm
0	0	0		1	$\overline{a}\overline{b}\overline{c}$	
0	0	1		0		$(a+b+\overline{c})$
0	1	0		0		$(a+\overline{b}+c)$
0	1	1		1	$\overline{a}bc$	
1	0	0		0		$(\overline{a}+b+c)$
1	0	1		1	$a\overline{b}c$	
1	1	0		1	$ab\overline{c}$	
1	1	1		0		$(\overline{a}+\overline{b}+\overline{c})$

1st canonical form
F2= $\overline{a}.\overline{b}.\overline{c} + \overline{a}bc + a.\overline{b}c + ab.\overline{c}$
2nd canonical form
F2= $(a+b+\overline{c})(a+\overline{b}+c)(\overline{a}+b+c)(\overline{a}+\overline{b}+\overline{c})$

a	b	c	d		f3	Minterm	Maxterm
0	0	0	0		0		$(a+b+c+d)$
0	0	0	1		0		$(a+b+c+\overline{d})$
0	0	1	0		0		$(a+b+\overline{c}+d)$
0	0	1	1		1	$\overline{a}\overline{b}cd$	
0	1	0	0		0		$(a+\overline{b}+c+d)$
0	1	0	1		1	$\overline{a}b\overline{c}d$	
0	1	1	0		1	$\overline{a}bc\overline{d}$	
0	1	1	1		1	$\overline{a}bcd$	
1	0	0	0		0		$(\overline{a}+b+c+d)$
1	0	0	1		1	$a\overline{b}\overline{c}d$	
1	0	1	0		1	$a\overline{b}c\overline{d}$	
1	0	1	1		1	$a\overline{b}cd$	
1	1	0	0		1	$ab\overline{c}\overline{d}$	
1	1	0	1		1	$ab\overline{c}d$	
1	1	1	0		1	$abc\overline{d}$	
1	1	1	1		1	$abcd$	

1st canonical form
$F3 = \overline{a}\overline{b}cd + \overline{a}b\overline{c}d + \overline{a}bc\overline{d} + \overline{a}bcd + a\overline{b}\overline{c}d + a\overline{b}c\overline{d} + a\overline{b}cd + ab\overline{c}\overline{d} + ab\overline{c}d + abc\overline{d} + abcd$

2nd Canonical form

$F3 = (a+b+c+d)(a+b+c+\overline{d})(a+b+\overline{c}+d)(a+\overline{b}+c+d)(\overline{a}+b+c+d)$

Exercice 6:

$\boxed{1}$ $f1(x,y,z) = xy + x\overline{z} + \overline{y}z$

		yz			
		00	01	11	10
x	0	0	1	0	0
	1	1	1	1	1

$f_1 = x + \overline{y}z$

2 f2(a, b, c) = 1 if the count of variables at 1 is even

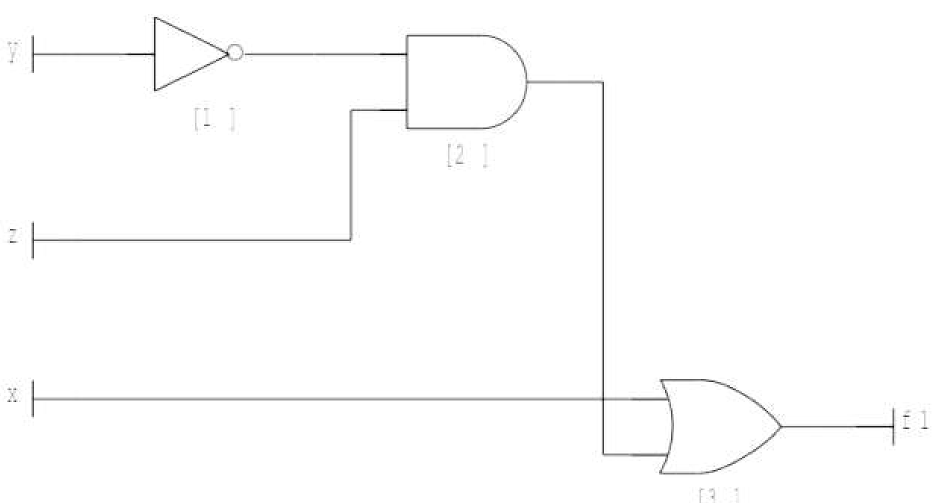
a \ bc	00	01	11	10
0	1	0	1	0
1	0	1	0	1

$$f_2 = \overline{a}\,\overline{b}\,\overline{c} + a\overline{b}c + \overline{a}bc + ab\overline{c}$$

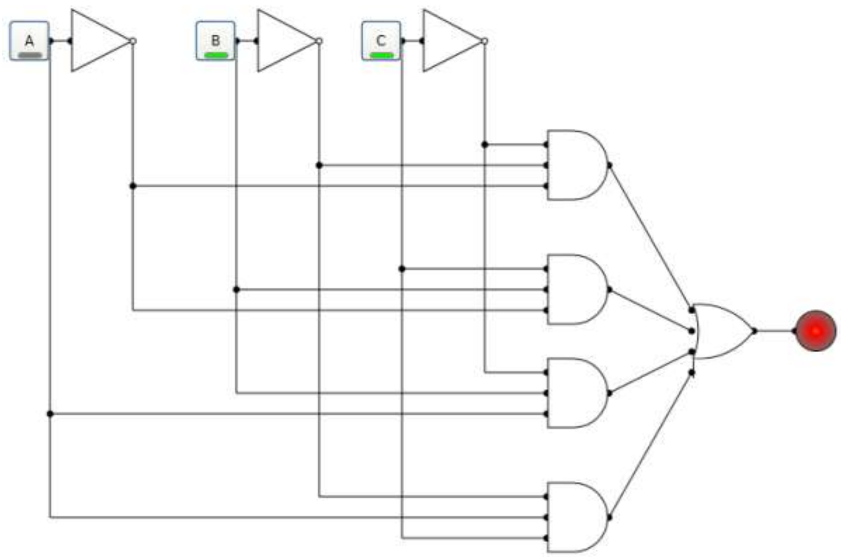
3 f3(a, b, c , d) = 1 if at least two variables are equal to 1

ab \ cd	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

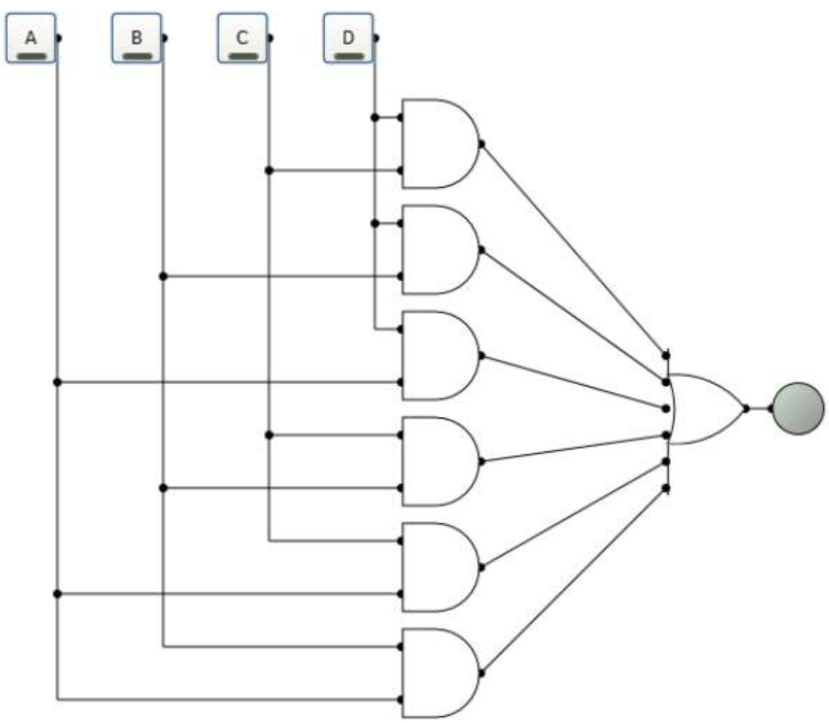
Exercice 7:



Logigramme de la fonction $f1(x, y, z) = xy + x\overline{z} + \overline{y}z$.



Logigram of function f2(a, b, c) = 1 if the count of variables at 1 is even.



Logigram of function f3(a, b, c , d) = 1 if at least two variables are equal to 1.

Exercice 8:

Truth table:

x	y	z		f4
0	0	0		0
0	0	1		0
0	1	0		1
0	1	1		1
1	0	0		1
1	0	1		0
1	1	0		0
1	1	1		0

Canonical forms:

1st canonical form:

$F4(x,y,z) = \overline{x}.y.\overline{z} + \overline{x}.y.z + x.\overline{y}.\overline{z}$

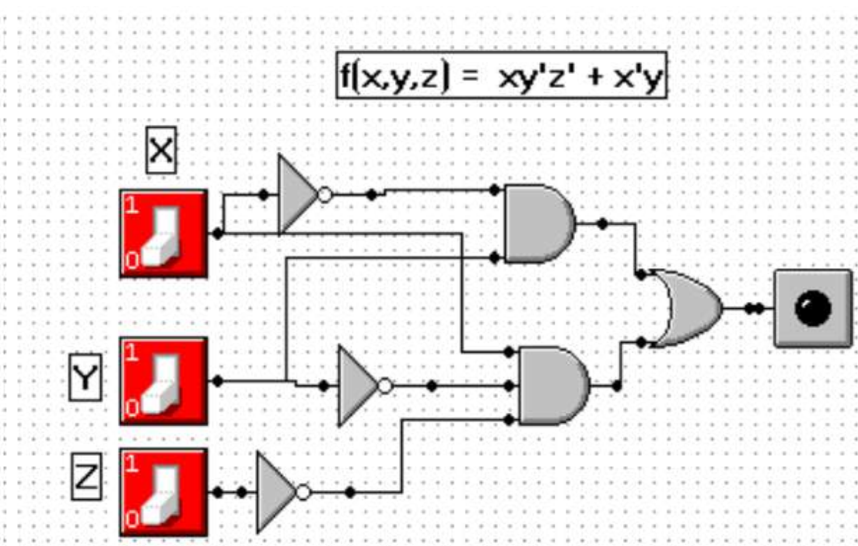
2nd canonical form:

$F4(x,y,z) = (x+y+z)(x+y+\overline{z})(x+\overline{y}+z)(\overline{x}+\overline{y}+z)(\overline{x}+\overline{y}+\overline{z})$

x \ yz	00	01	11	10
0	0	0	1	1
1	1	0	0	0

Simplification:

$f(x,y,z) = x.\overline{y}.\overline{z} + \overline{x}.y$



Logigram of $F(x,y,z) = x \oplus (y+z)$.