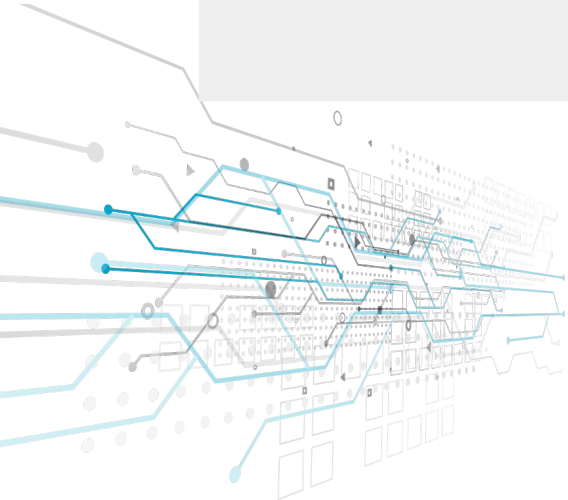
Ministry of Higher Education and Scientific Research
University of Larbi Ben M'Hidi, Oum El Bouaghi
Faculty of Exact Sciences and Natural and Life Sciences
Department of Mathematics and Computer Science

Computer Structure 2

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

2023-2024

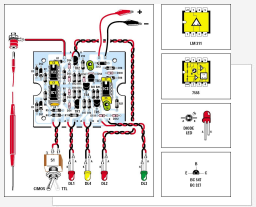
Target audience	1st year mathematics student + 1st year Common core Mathematics and computer science students .
Prerequisite	Students must have basic computer skills.
Assessment method	Assessment method: Exam (60%), Continuous assessment (40%)



Chapter 0: General Introduction

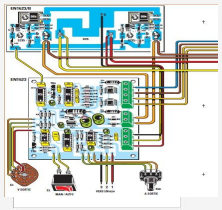
01

Computation and Arithmetic



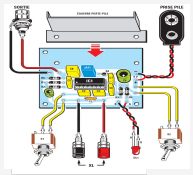
02

Sequence Control



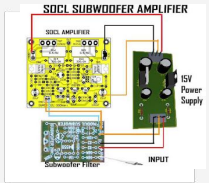
03

Decoding and Encoding

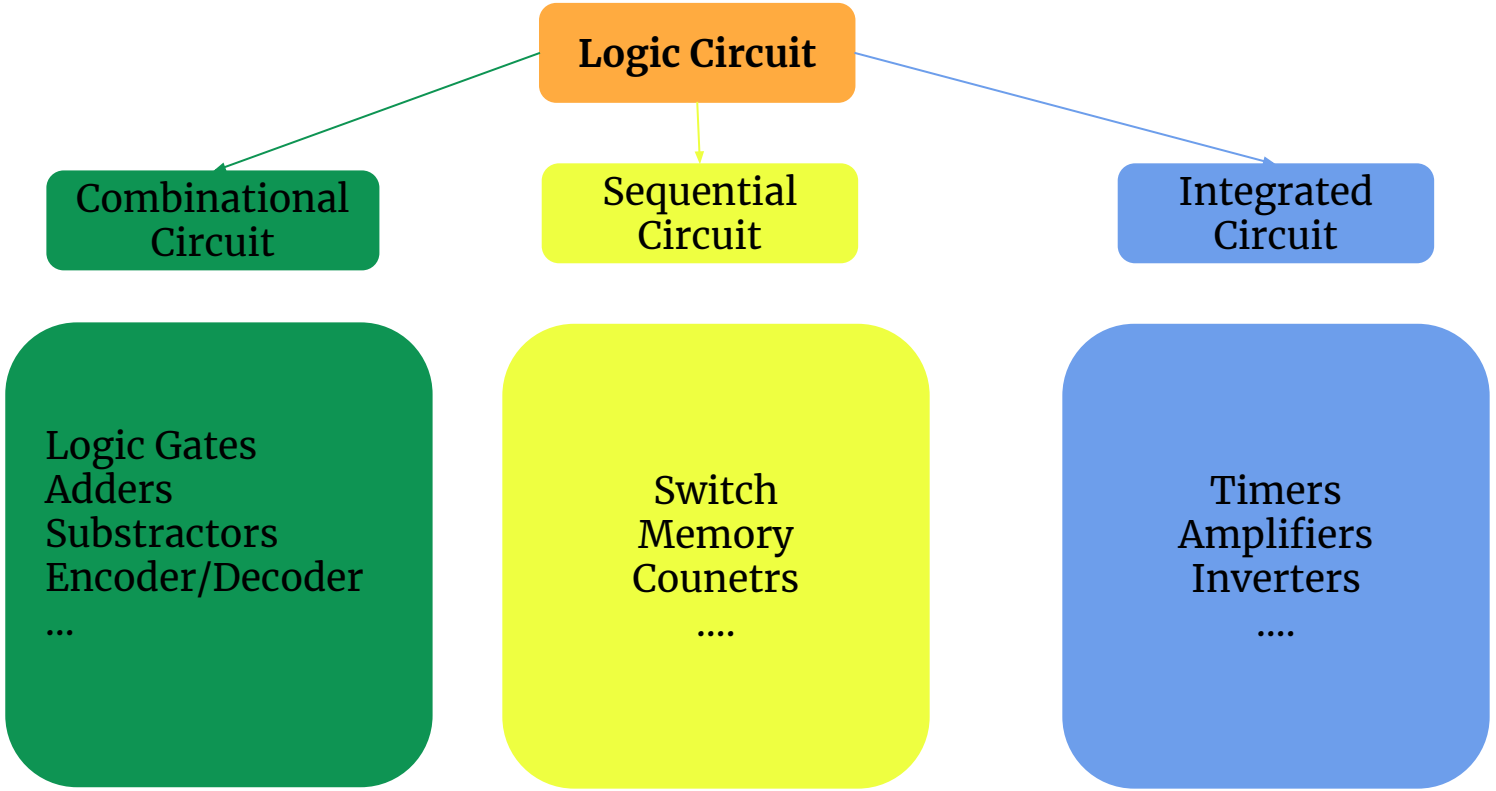


04

Communication Interfaces



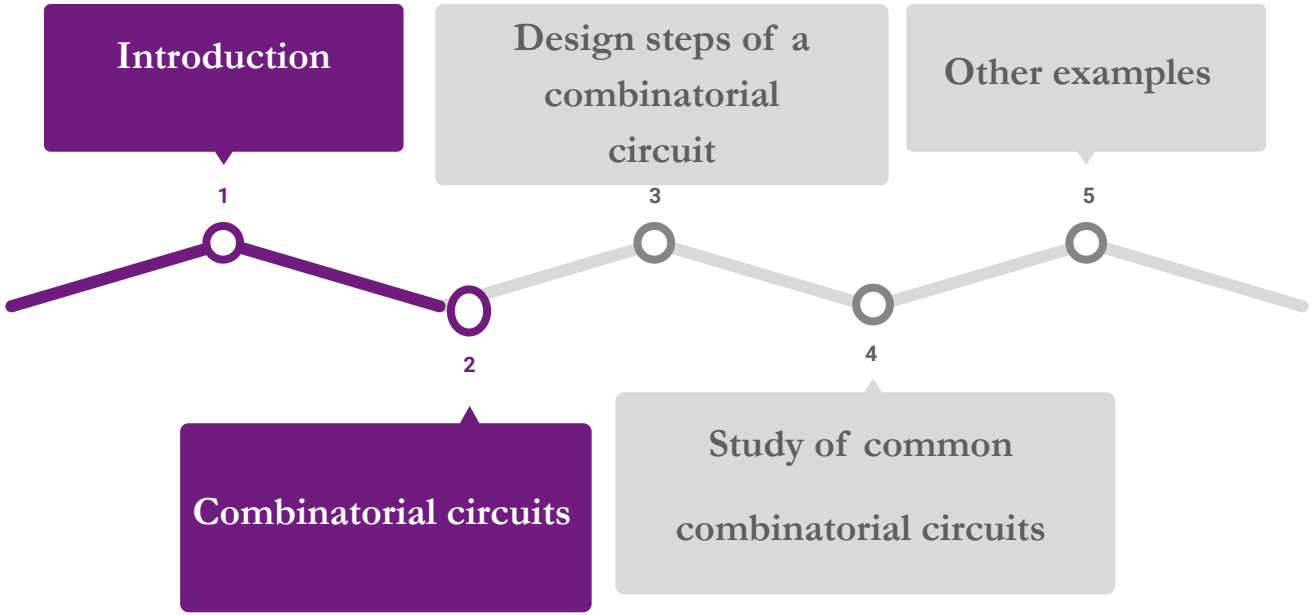
Chapter 0: General Introduction



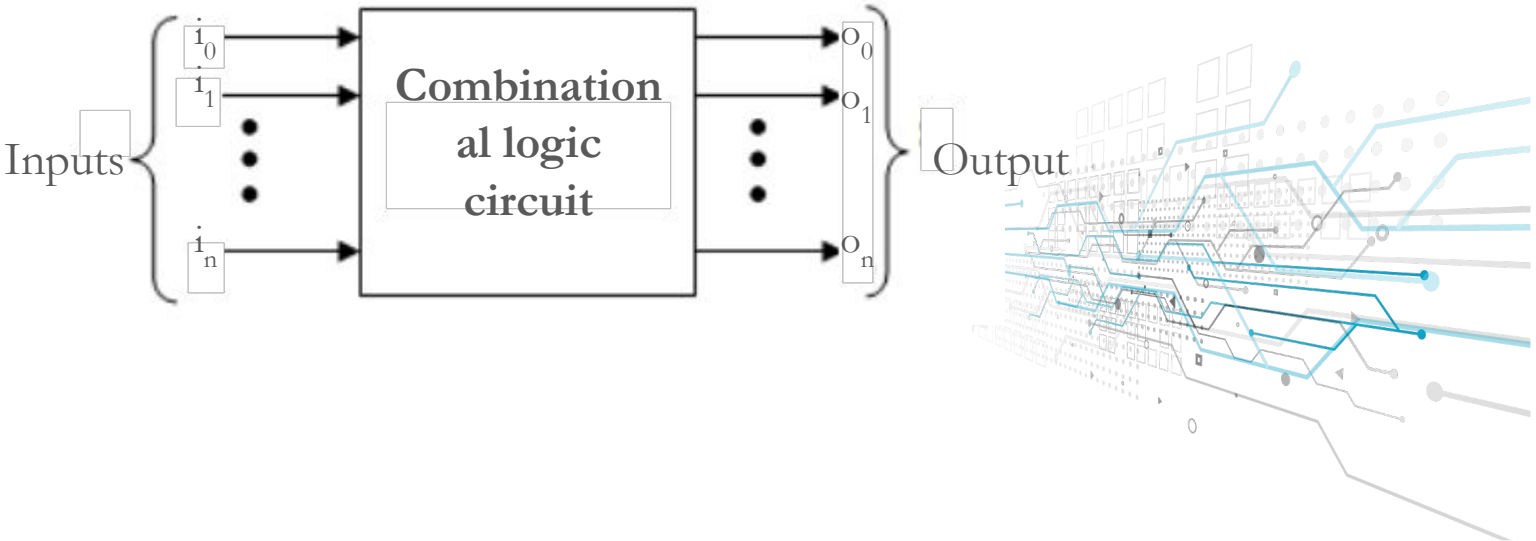
Outline:

Chapter 0	General Introduction
Chapter 1	Combinatorial logic
Chapter 2	Sequential logic
Chapter 3	Integrated circuits

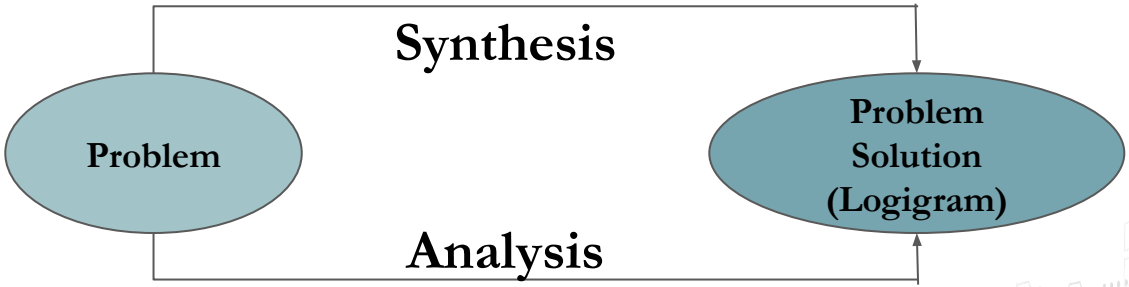
Chapter 1: Combinatorial logic



➔ **A combinational circuit** is a circuit that generates output solely based on the current input values, without any consideration for previous inputs or the circuit's state.



What is the difference between Analysis and Synthesis of a logical circuit?



The design or the Synthesis of a combinational circuit follow the following steps:

- ➔ Establish the truth table for each of the functions involved in the problem to be addressed and Establish the logical equations.
- ➔ Simplify the equations for each of the logical functions (Karnaugh tables or algebraically)
- ➔ Implement the logic circuit.

Combinational Circuit

Arithmetic	Transcoding	Switch	Comparison
Half adder	The encoders	Multiplexers	Comparators
Full adder	Priority encoders	Demultiplexers	
Half subtractor	Decoder		
Full subtractor			

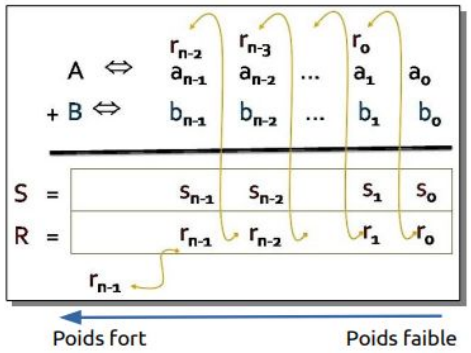
Arithmetic operators

➔ Half adder:

The task is to synthesize an addition circuit for two numbers, A and B, each represented by n bits.

This circuit should **take 2n input variables** (corresponding to the n-bit representations of numbers A and B) and produce **n+1 output bits** representing **the sum (S)** and a single bit indicating the final **carry of the calculation.**

principle



S=A+B

➔ Half adder:

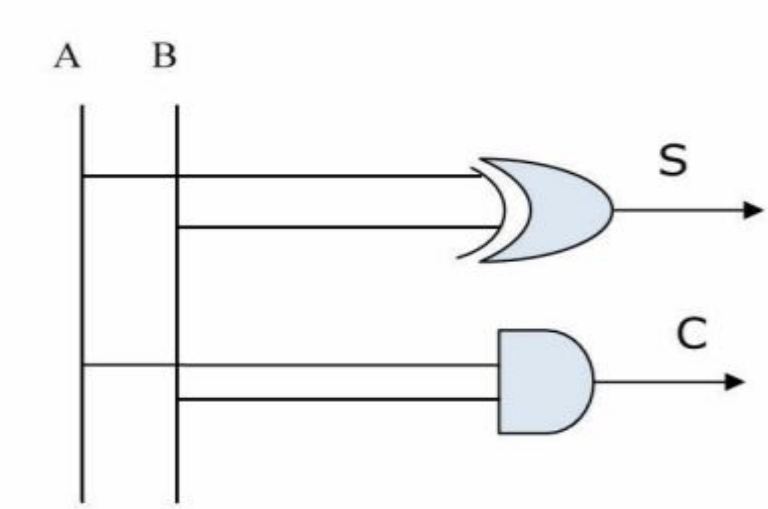
A. Truth Table

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

$S = A\overline{B} + \overline{A}B = A \oplus B$

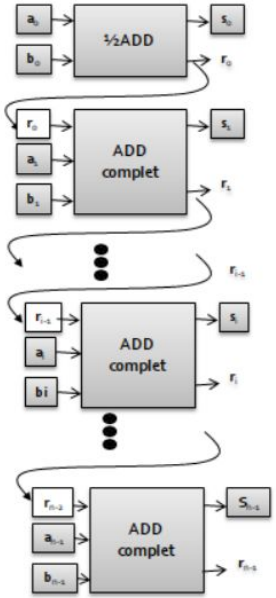
$C = A.B$

B. Implementation



➔ Full adder:

A full adder typically has three input bits: A, B and an incoming carry (C_{i-n}), and two output bits: the sum (S) and a carry-out (C).



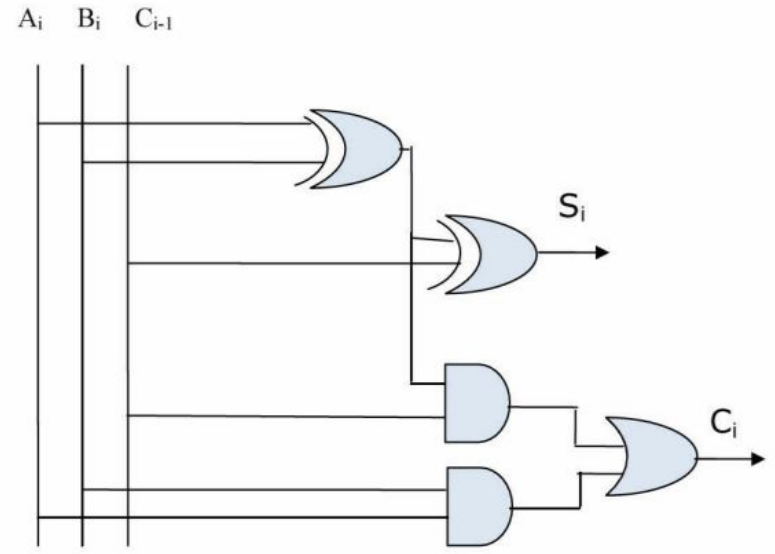
➔ Full adder:

A. Truth Table

C_{i-1}	B_i	A_i	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S_i = C_{i-1} \oplus B_i \oplus A_i$$
$$C_i = A_i B_i + C_{i-1}(A_i \oplus B_i)$$

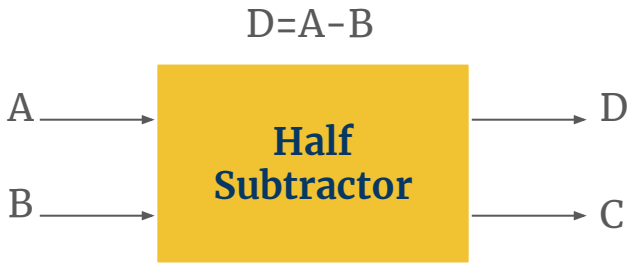
B. Implementation



➔ Half subtractor:

The task is to synthesize an subtractor circuit for two numbers, A and B, each represented by n bits.

This circuit is capable of subtracting one bit from another. Therefore, it outputs their difference (D) and the carry (C).



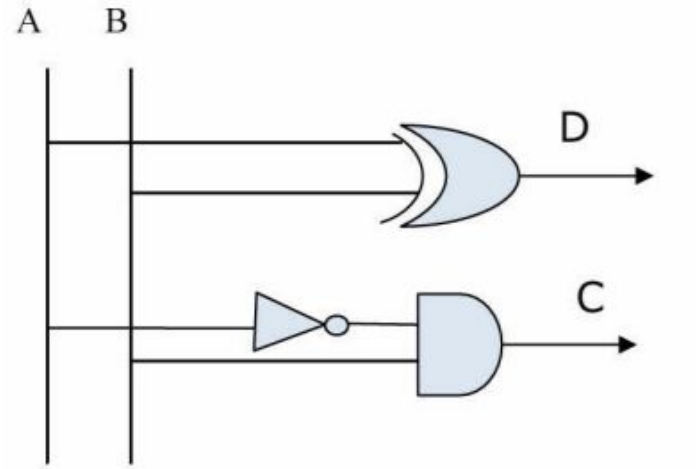
➔ Half subtractor:

A. Truth Table

A	B	D	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

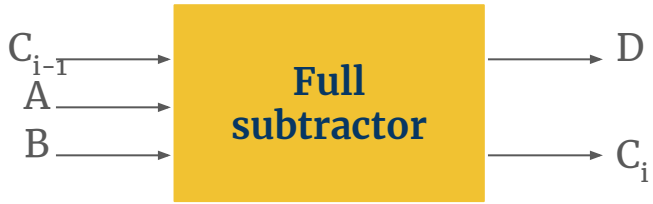
$$D = \overline{A}B + A\overline{B} = A \oplus B$$
$$C = \overline{A}B$$

B. Implementation



➔ Full subtractor :

A full **subtractor** has three input bits: A_i, B_i and a carry (C_{i-n}) of the previous step, and generates: the difference (D) and a carry-out of the corresponding step(C).



➔ **Full subtractor:**

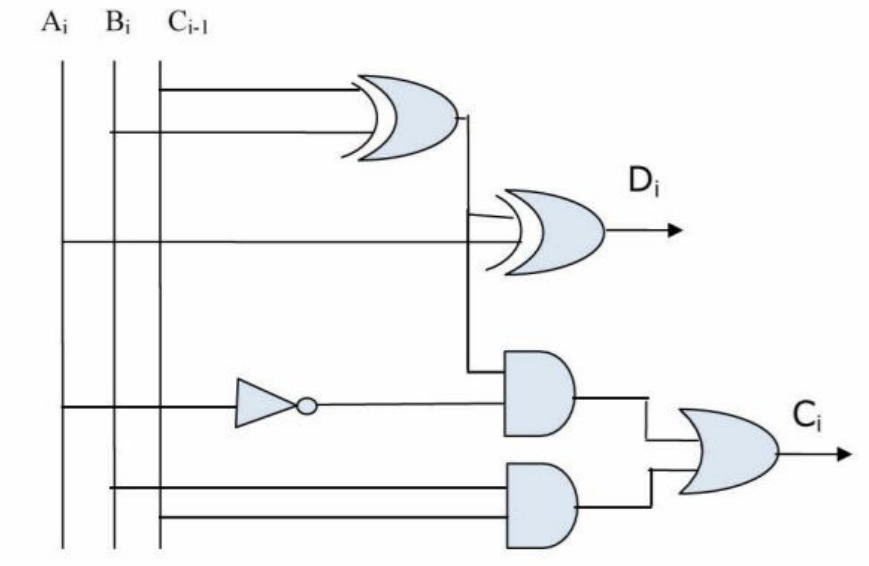
A. Truth Table

A_i	B_i	C_{i-1}	D_i	C_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = \overline{A}_i (B_i \oplus C_{i-1}) + B_i C_{i-1}$$

B. Implementation



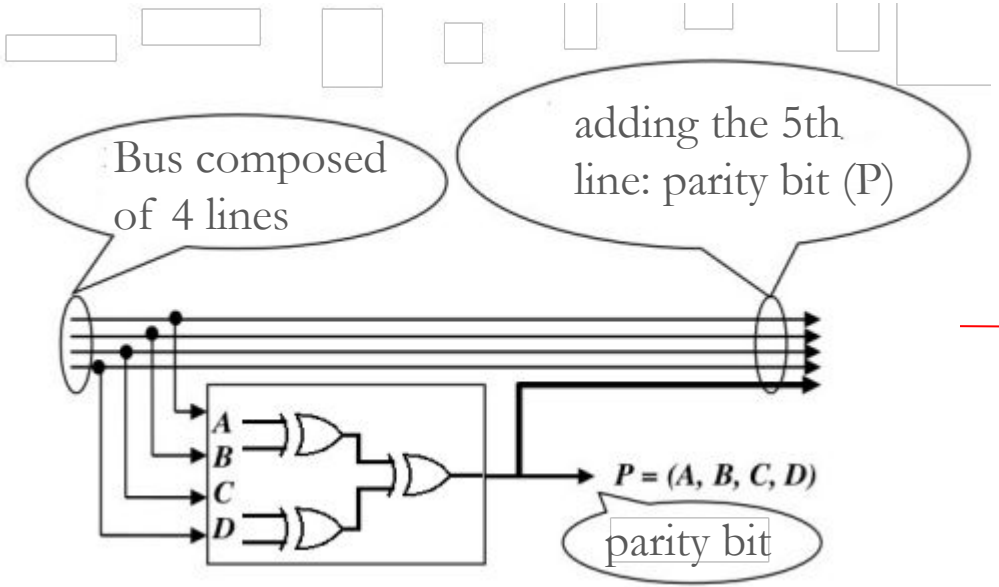
Exercice:

Design **a parity generator circuit**: A parity generator is defined as a logical circuit that ensures the number of '1' bits on a bus is always even.

Assuming a bus consists of 4 lines (A, B, C, and D), the parity generator introduces a 5th line named 'P,' whose logical state is configured to ensure an even number of '1' bits on this bus.

It is worth noting that a bus is a collection of wires, each carrying a single bit of information.

Solution:



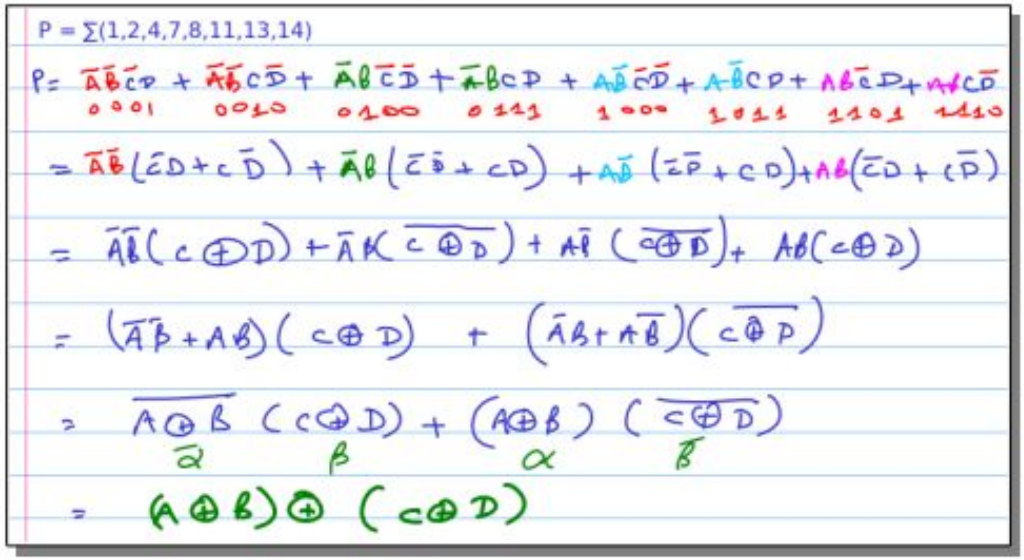
Truth Table

	ABCD	P
m_0	0000	0
m_1	0001	1
m_2	0010	1
m_3	0011	0
m_4	0100	1
m_5	0101	0
m_6	0110	0
m_7	0111	1
m_8	1000	1
m_9	1001	0
m_{10}	1010	0
m_{11}	1011	1
m_{12}	1100	0
m_{13}	1101	1
m_{14}	1110	1
m_{15}	1111	0

Solution:

	ABCD	P
m_0	0000	0
m_1	0001	1
m_2	0010	1
m_3	0011	0
m_4	0100	1
m_5	0101	0
m_6	0110	0
m_7	0111	1
m_8	1000	1
m_9	1001	0
m_{10}	1010	0
m_{11}	1011	1
m_{12}	1100	0
m_{13}	1101	1
m_{14}	1110	1
m_{15}	1111	0

$P = \sum(1,2,4,7,8,11,13,14)$



Solution:

$P = \sum(1,2,4,7,8,11,13,14)$

$$P = \underset{0001}{\bar{A}\bar{B}\bar{C}D} + \underset{0010}{\bar{A}\bar{B}C\bar{D}} + \underset{0100}{\bar{A}B\bar{C}\bar{D}} + \underset{0111}{\bar{A}BCD} + \underset{1000}{A\bar{B}\bar{C}\bar{D}} + \underset{1011}{A\bar{B}CD} + \underset{1101}{AB\bar{C}D} + \underset{1110}{ABC\bar{D}}$$

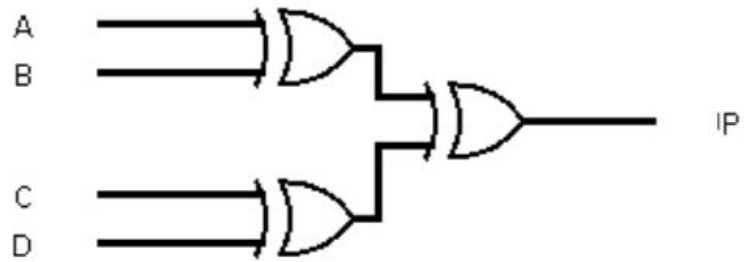
$$= \bar{A}\bar{B}(\bar{C}D + C\bar{D}) + \bar{A}B(\bar{C}\bar{D} + CD) + A\bar{B}(\bar{C}\bar{D} + CD) + AB(\bar{C}D + C\bar{D})$$

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

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

$$= \underset{\bar{\alpha}}{\overline{A \oplus B}}\ \underset{\beta}{(C \oplus D)} + \underset{\alpha}{(A \oplus B)}\ \underset{\bar{\beta}}{\overline{(C \oplus D)}}$$

$$= (A \oplus B) \oplus (C \oplus D)$$



Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Encoders:**

An encoder, or coder, is a combinational circuit with $\mathbf{2^n}$ inputs and **n** outputs.

Only a single input is active at a time, and the binary code equivalent to the number of this active input is delivered on the n outputs.

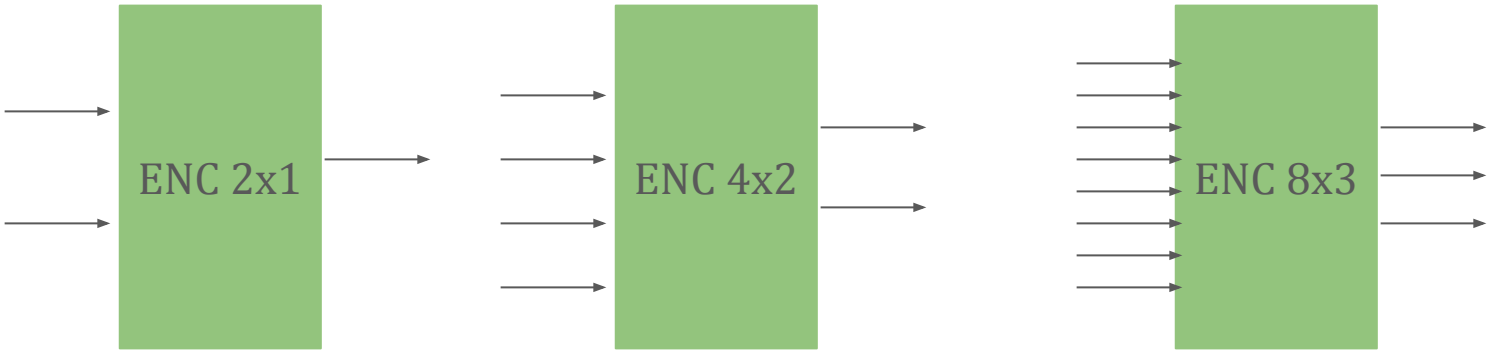
This circuit translates the rank of the active input into a binary code at the output.

It should be noted that in this type of circuit, only one input is set to 1 while the others are 0. This is known as the 'one-hot' input format.

Example of use: calculator, remote, keyboard …etc.

Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

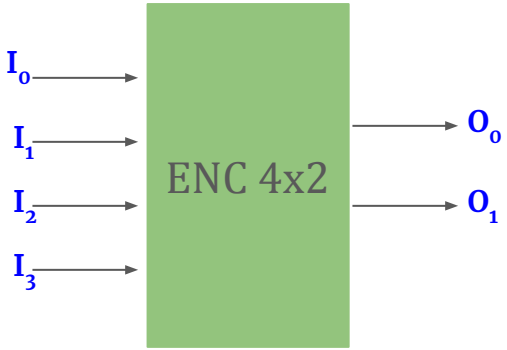
➔ **Encoders:**



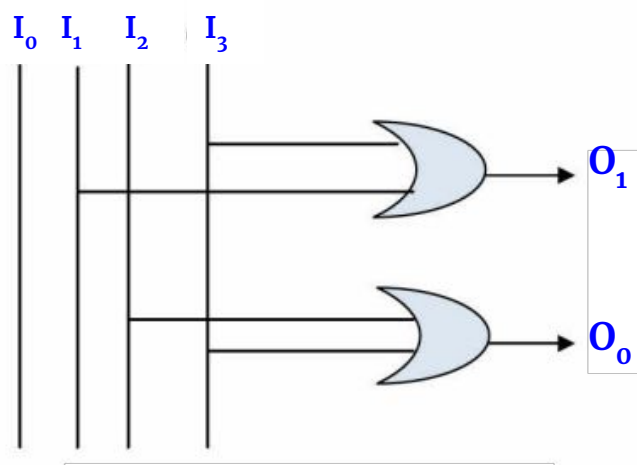
Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Encoders:**

Exemple:



I_0	I_1	I_2	I_3	O_0	O_1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

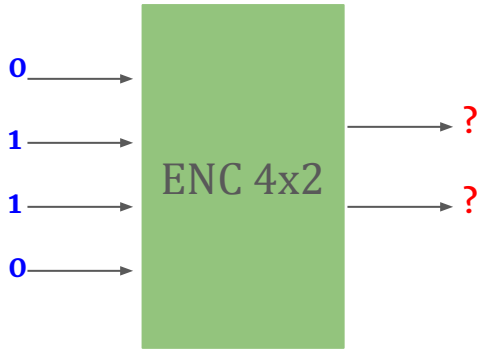


O_0=1 IF (I_2=1) OR (I_3=1) => $O_0 = I_2 + I_3$

O_1=1 IF (I_1=1) OR (I_3=1) => $O_1 = I_1 + I_3$

Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Priority encoders**



Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Priority encoders:**

A priority encoder is a combinational circuit that accepts multiple input lines and produces a binary code at its output corresponding to the highest-priority active input.

In other words, it encodes the highest-priority active input into binary code.

Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Priority encoders**

I_3	I_2	I_1	I_0	O_1	O_0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

e_3	e_2	e_1	e_0	S_1	S_0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1

Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Priority encoders**

$I_3 I_2$ \ $I_1 I_0$	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

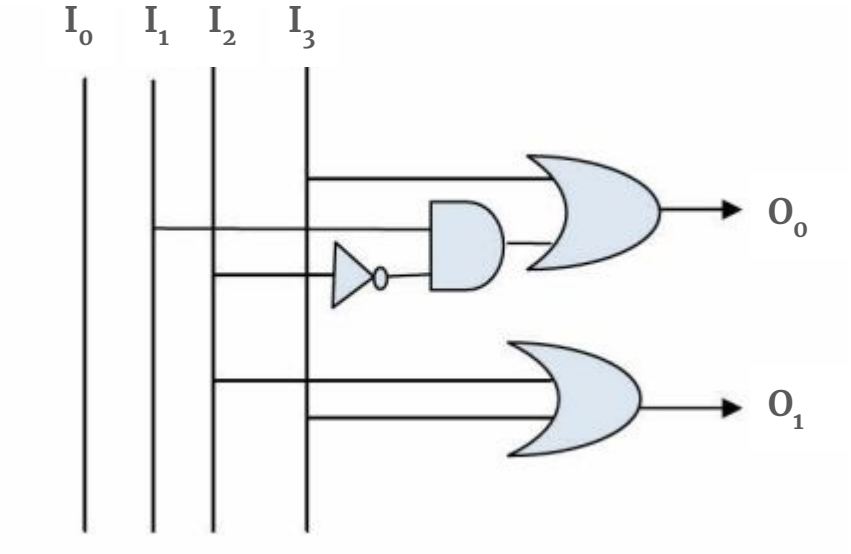
$O_0 = I_3 + \bar{I}_2 I_1$

$I_3 I_2$ \ $I_1 I_0$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$O_1 = I_3 + I_2$

Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Priority encoders:**

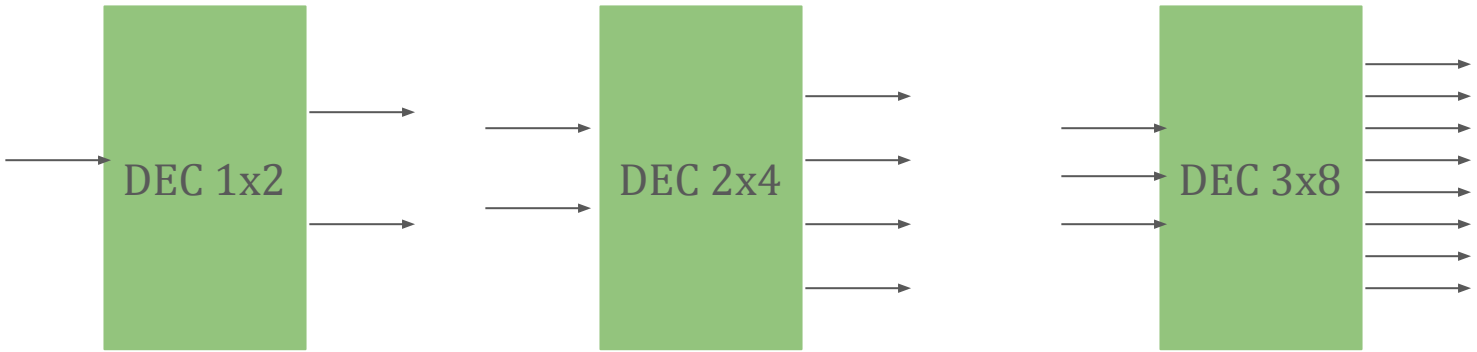


Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Decoders:**

A decoder is a combinational circuit that converts binary information from an input code to a set of output lines.

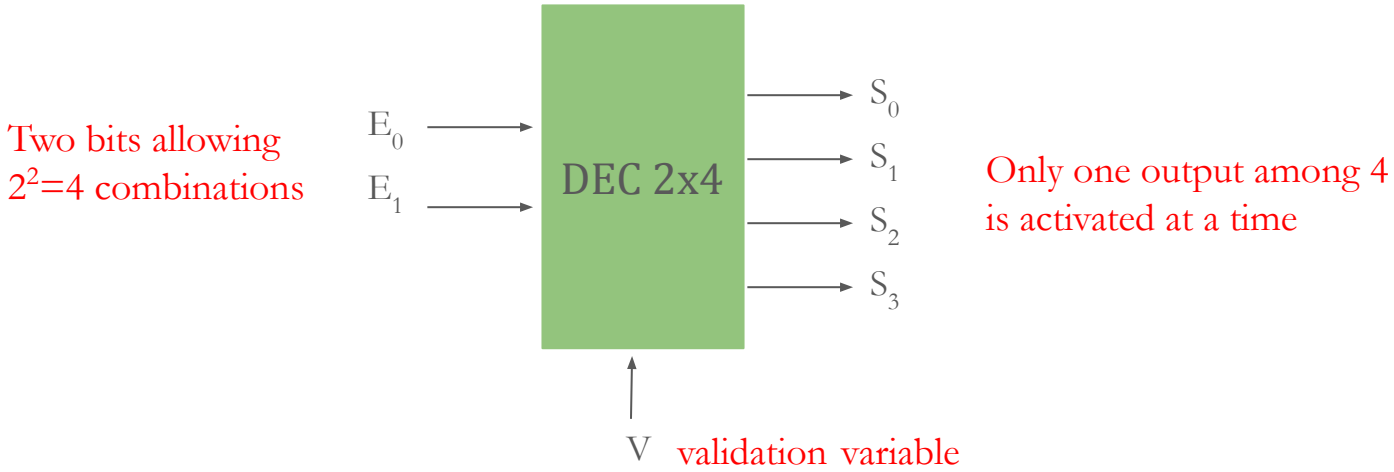
The number of output lines is M such as $M \leq n$, where n is the number of input lines with 2^n possible combination. (typically $M= 2^n$).



Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Decoders:**

Example: Decoder 4x2 (1 of 4 decoder)



Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Decoders:**

Example: Decoder 4x2 (1 of 4 decoder)

V	E_1	E_0	S_3	S_2	S_1	S_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

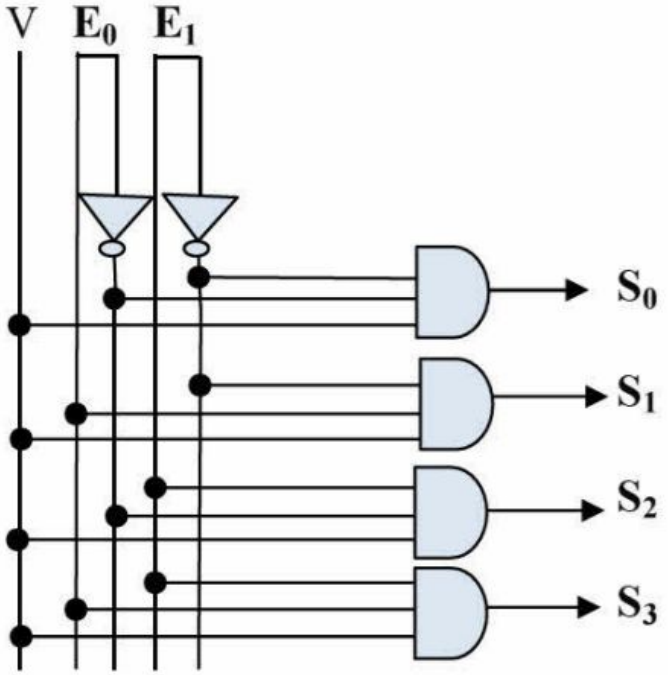
$$S_0 = V(\bar{E}_1\bar{E}_0)\ ;\ S_1 = V(\bar{E}_1 E_0)\ ;\ S_2 = V(E_1\bar{E}_0)\ ;\ S_3 = V(E_1 E_0)$$

Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Decoders:**

Example: Decoder 4x2 (1 of 4 decoder)

$$S_0 = V(\overline{E}_1\overline{E}_0) \; ; S_1 = V(\overline{E}_1 E_0) \; ; S_2 = V(E_1\overline{E}_0) \; ; S_3 = V(E_1 E_0)$$



Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Decoders:**

Example: **Decoder 4x2 (1 of 4 decoder)**

NB: A decoder can be used in two ways, where the output can represent the function:

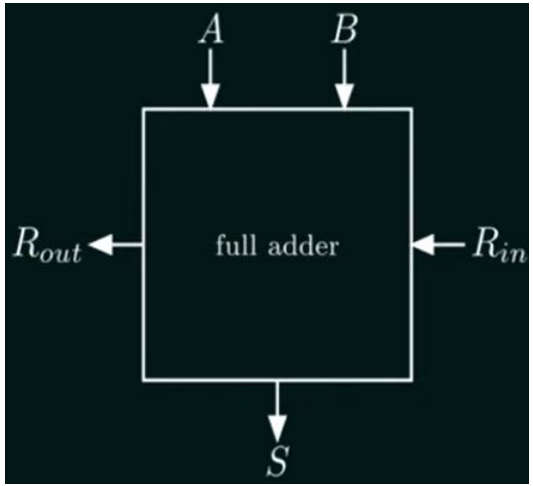
- Code converter: For each input code, there is a corresponding output code.
  For example: binary-to-octal, binary-to-decimal decoders, … etc.

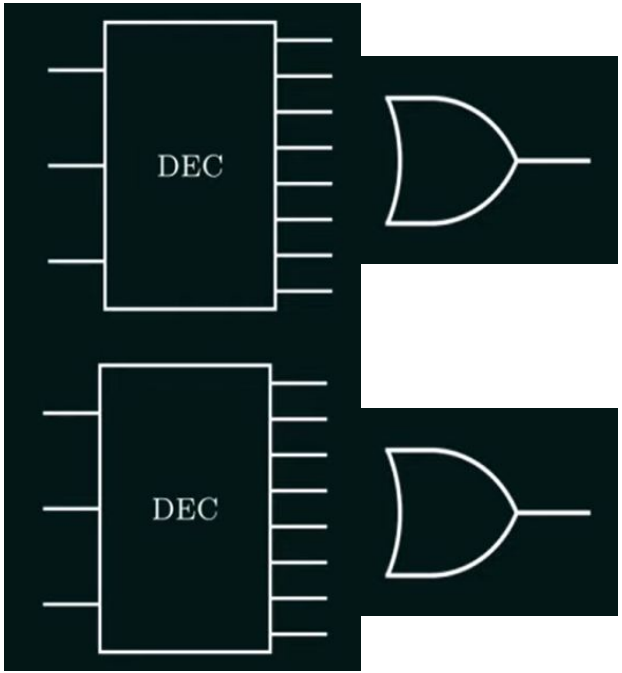
- Output selector: Only one output among the available **m** outputs is activated at a time based on the binary value presented at the input.

Exercise : Implement a full adder logical circuit utilizing decoders.

Objective:

A	B	R_{in}	S	R_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1





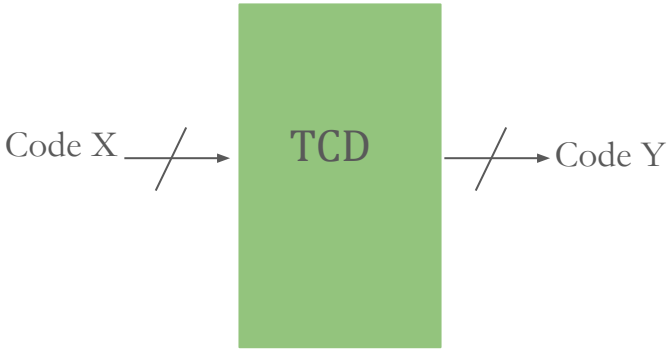
Solution:

- Why two decoders?
- According to the truth table: **S=∑(1,2,4,7)** and **R_{out}=∑(3,5,6,7)**
- How to choose the decoder?

Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Transcoders:**

An Transcoder is a combinational circuit which convert a code **X** (with **k bits** inputs) to another code **Y** (with **P bits** outputs). k and p are are independent.

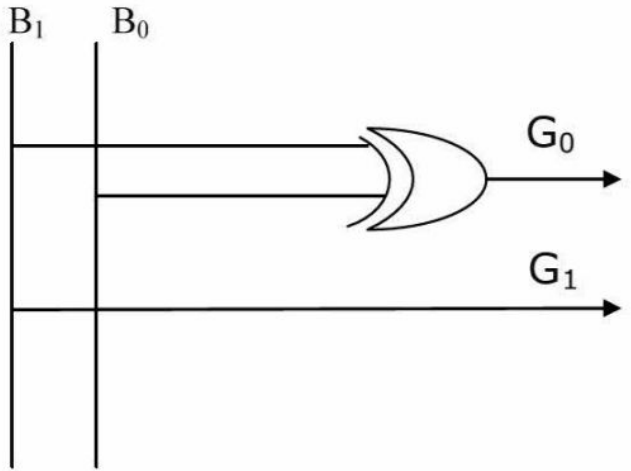


Transcoding operators : Transcoding is the process of converting one encoded format of digital data (code 1) to another (code 2).

➔ **Transcoders:**

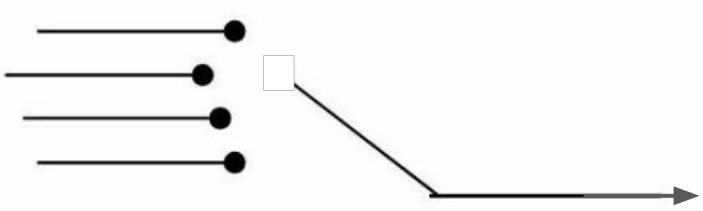
Example: Binary to Gray Transcoder in 2 bits

Binary		Gray	
B_1	B_0	G_1	G_0
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

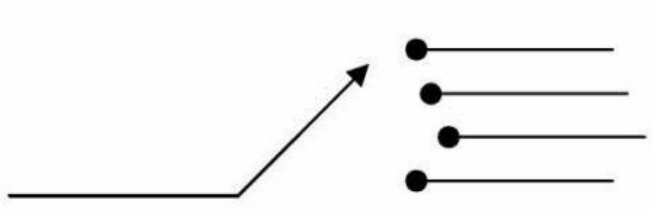


$$G_1 = B_1 \; ; \quad G_0 = B_1 \oplus B_0$$

switch operators : the multiplexer (switching) is a device which allows information to be transmitted on a single line from several sources or at a distance from several targets.



Multiplexer



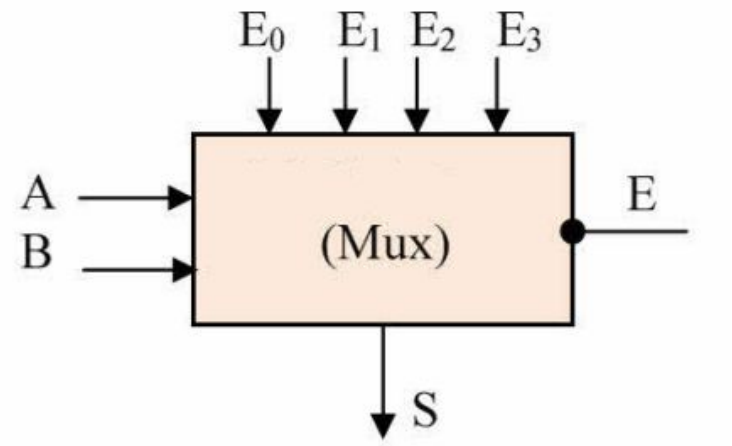
DeMultiplexer

➔ **Multiplexer (Mux):** this circuit selects one input from N and transmits the information carried by this line to a single output channel S.

switch operators : the multiplexer (switching) is a device which allows information to be transmitted on a single line from several sources or at a distance from several targets.

➔ **Multiplexer (Mux):** this circuit selects one input from N and transmits the information carried by this line to a single output channel S.

Example: Mux 4x1 => 4 inputs lignes

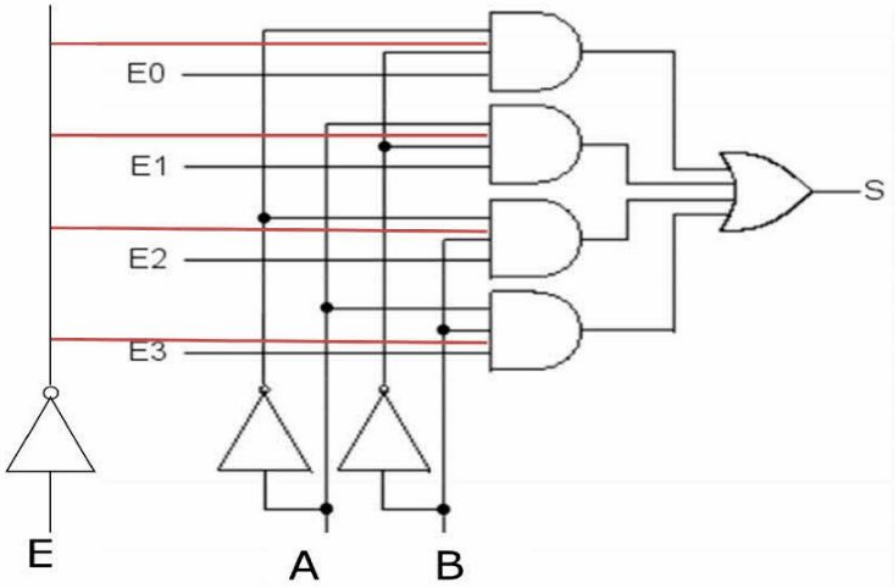


A	B	E	S
x	x	1	0
0	0	0	E_0
0	1	0	E_1
1	0	0	E_2
1	1	0	E_3

switch operators :

Example: Mux 4x1 => 4 inputs lignes

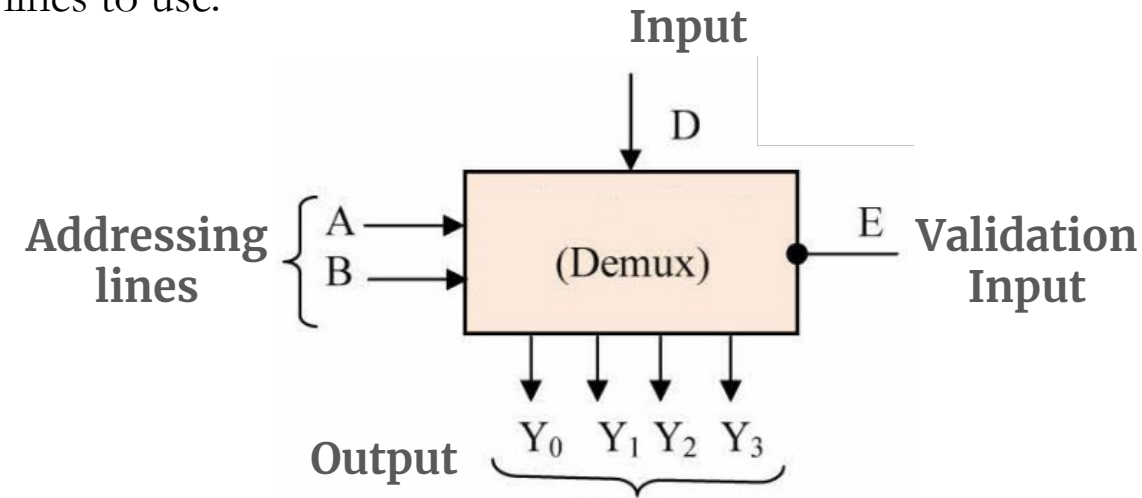
$$S = \bar{E}\,\bar{A}.\bar{B}.E_0 + \bar{E}.A.B.E_1 + \bar{E}\,A.\bar{B}.E_2 + \bar{E}\,A.B.E_3$$



switch operators : the multiplexer (switching) is a device which allows information to be transmitted on a single line from several sources or at a distance from several targets.

➔ **DeMultiplexer (DeMux):** A demultiplexer is a combinational circuit with one input and N outputs and witch match the input with only one output.
To be able to select this output we also need addressing lines: the code carried by these lines identifies the output lines to use.

Example: DeMux 1x4



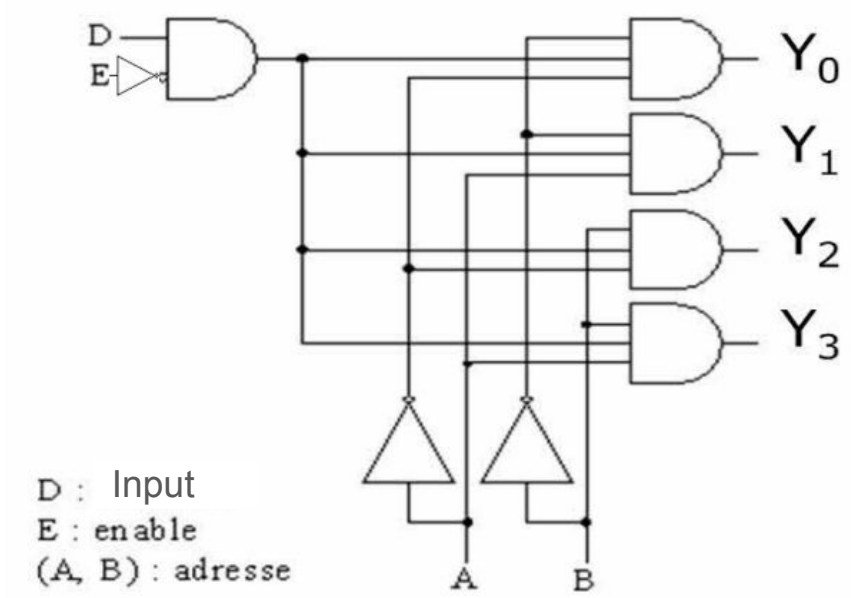
switch operators : the multiplexer (switching) is a device which allows information to be transmitted on a single line from several sources or at a distance from several targets.

Example: DeMux 1x4

A	B	E	Y_0	Y_1	Y_2	Y_3	
x	x	1	0	0	0	0	
0	0	0	D	0	0	0	$Y_0 = \overline{A}\overline{B}\overline{E}D$
0	1	0	0	D	0	0	$Y_1 = \overline{A}B\overline{E}D$
1	0	0	0	0	D	0	$Y_2 = A\overline{B}\overline{E}D$
1	1	0	0	0	0	D	$Y_3 = AB\overline{E}D$

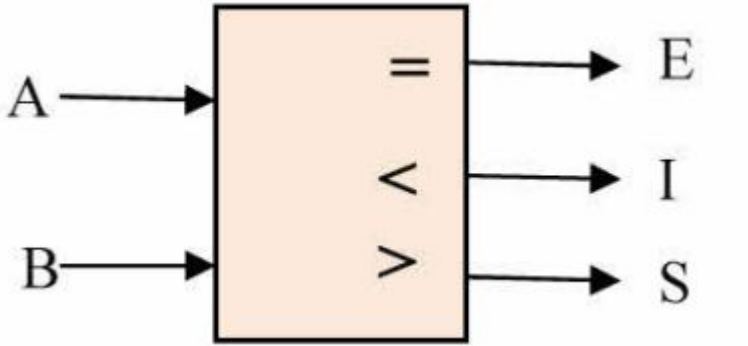
switch operators : the multiplexer (switching) is a device which allows information to be transmitted on a single line from several sources or at a distance from several targets.

Example: DeMux 1x4



comparison operators : the comparator is an operator capable of detecting equality and comparing two numbers.

Example: 1 bit comparator



comparison operators : the comparator is an operator capable of detecting equality and comparing two numbers.

Example: 1 bit comparator

A	B	E	I	S
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

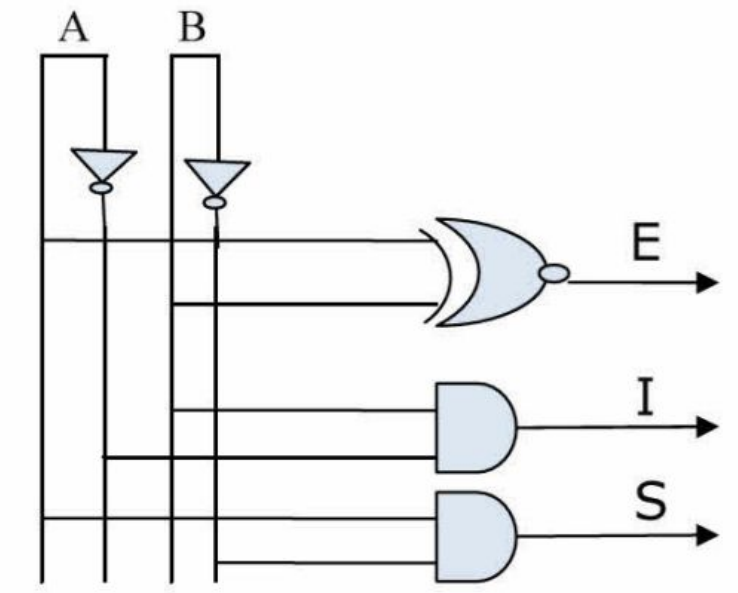
$$E = \overline{A}\,\overline{B} + AB = \overline{A \oplus B}$$

$$I = \overline{A}B$$

$$S = A\overline{B}$$

comparison operators : the comparator is an operator capable of detecting equality and comparing two numbers.

Example: 1 bit comparator



Exercise:

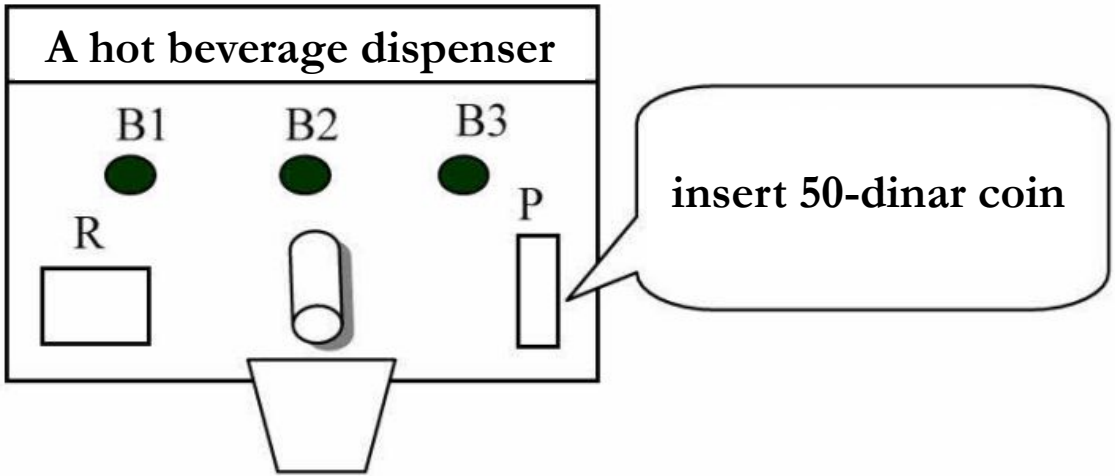
A hot beverage dispenser is designed to serve **coffee** or **Tea**, with or without **Milk**, or just **Milk alone**. Here's how it operates:

- There are three buttons: one for coffee (B1), one for tea (B2), and one for milk (B3).
- To get a drink without milk, simply press the corresponding button.
- If you want your drink with milk, press both the button for your chosen drink and the milk button simultaneously.
- Additionally, the dispenser only works if a 50-dinar coin has been inserted into the coin slot (P).
- If there's an incorrect operation after inserting the coin (e.g., pressing both coffee and tea buttons simultaneously), the coin will be returned (R).
- Since milk is provided free of charge, the coin will also be returned if only milk is chosen.

Exercise:

- Construct the truth table
- simplify using Karnaugh maps,

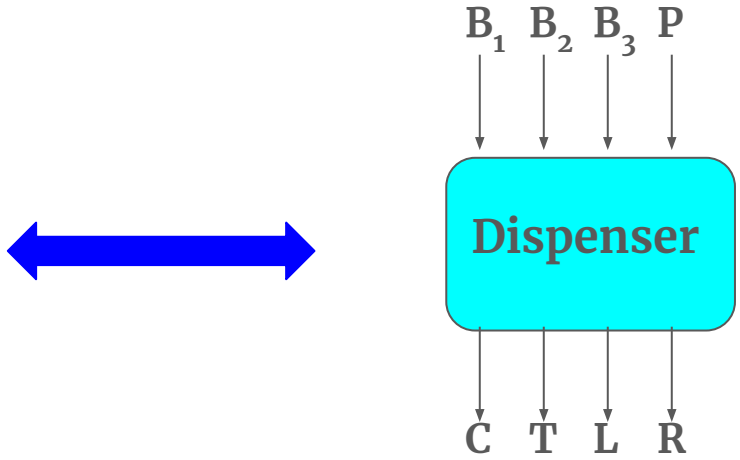
then deduce the simplified logical expressions for
R: coin return,
C: coffee distribution,
T: tea distribution,
L: milk distribution.



- Establish the logic diagram of the dispenser using NAND gates.

Solution:

P	B1	B2	B3	C	T	L	R
0	x	x	x	0	0	0	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	1	1
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	1	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	1



Solution:

$$C = PB_1\overline{B_2}$$

PB1 \ B2B3	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	0	0
10	0	0	0	0

$$T = P\bar{B}_1B_2$$

$$L = P\overline{B_2}B_3 + P\overline{B_1}B_3$$

$$R = P\overline{B_1}\,\overline{B_2} + PB_1B_2$$

Solution:

$$\bar{\bar{C}} = \overline{\overline{PB_1\overline{B_2}}} \qquad \bar{\bar{T}} = \overline{\overline{P\overline{B_1}B_2}}$$

$$\bar{\bar{L}} = \overline{\overline{P\overline{B_2}B_3 + P\overline{B_1}B_3}} = \overline{\overline{P\overline{B_2}B_3} \cdot \overline{P\overline{B_1}B_3}}$$

$$\bar{\bar{R}} = \overline{\overline{P\overline{B_1}\,\overline{B_2} + PB_1B_2}} = \overline{\overline{P\overline{B_1}\,\overline{B_2}} \cdot \overline{PB_1B_2}}$$