Digital Logic

Lecture 7

2nd Stage

Computer Science Department

Faculty of Science

Soran University

Rafiq.maulud@soran.edu.iq

Fall 2022

Combinational Logic Circuits

Topics covered

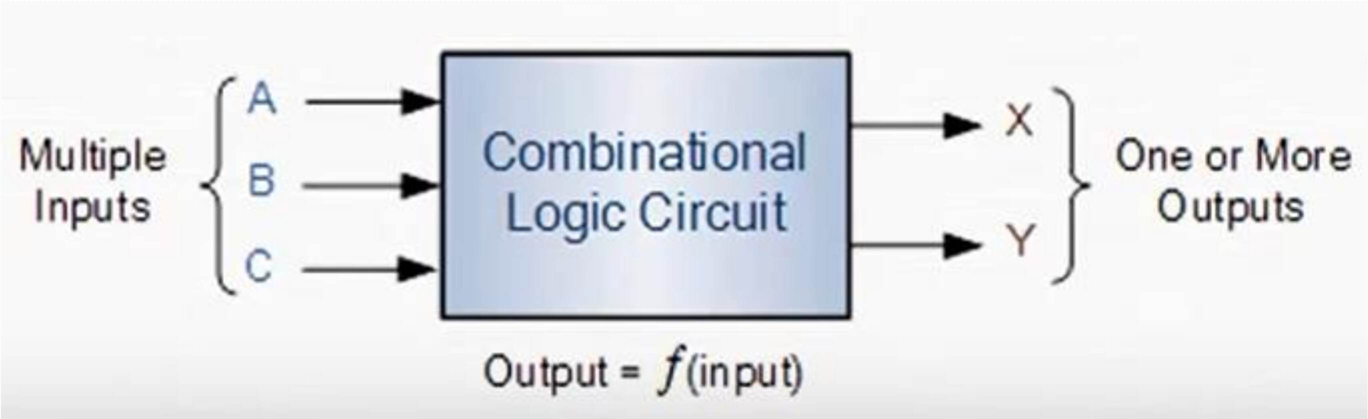
- Combinational Logic Circuits
  - Adder Circuits
    - The Half-Adder Circuit
    - The Full-Adder Circuit

  - Subtraction Circuits
    - Half Subtractors
    - Full Subtractors

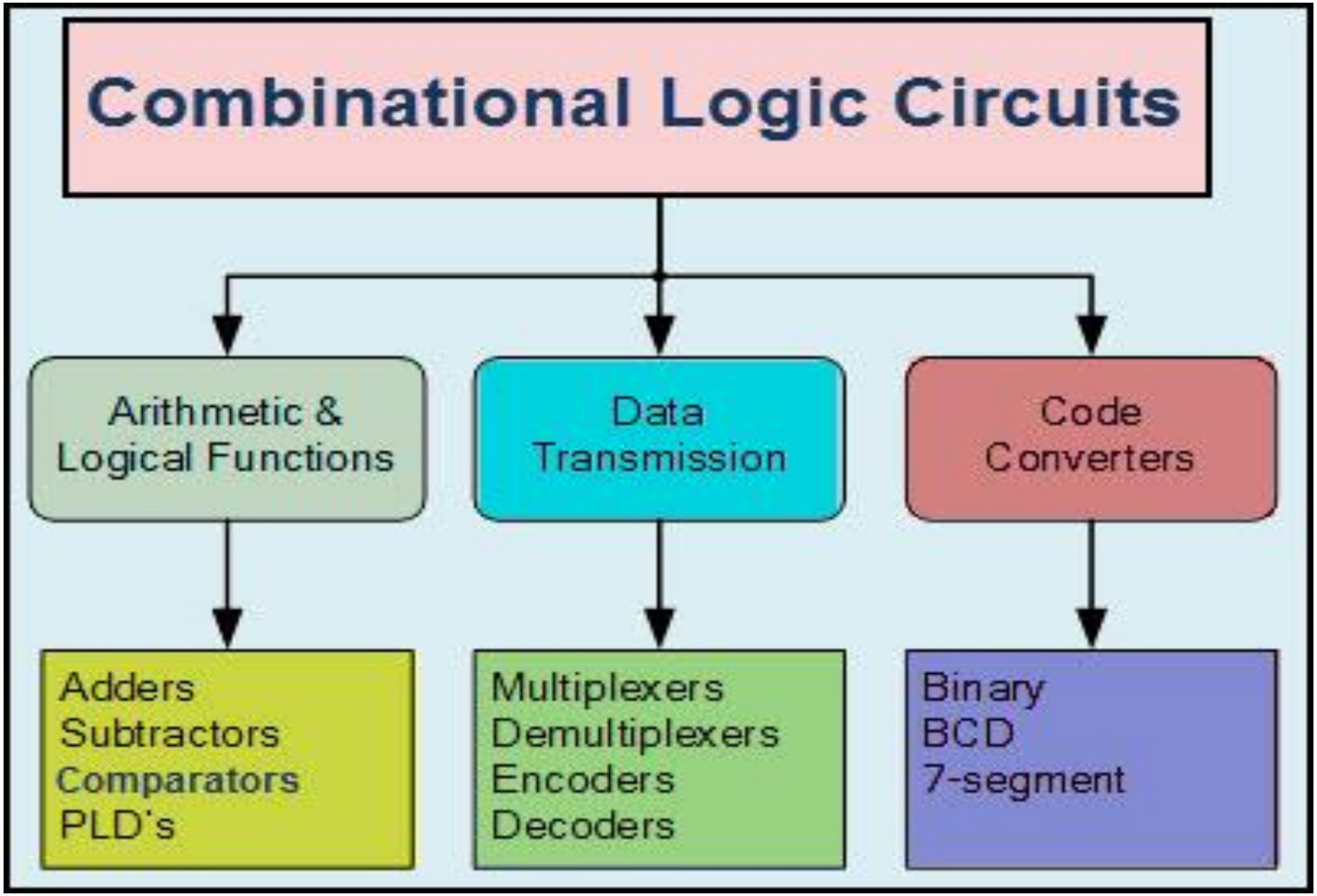
  - Digital Comparators
    - Single Bit Magnitude Comparator
    - 2-Bit Comparator
    - 4-Bit Comparator

Combinational Logic Circuits

Combinational Logic Circuits are made up from basic logic gates that are “combined” or connected together to produce more complicated switching circuits. These logic gates are the building blocks of combinational logic circuits.



Combinational Logic Circuits



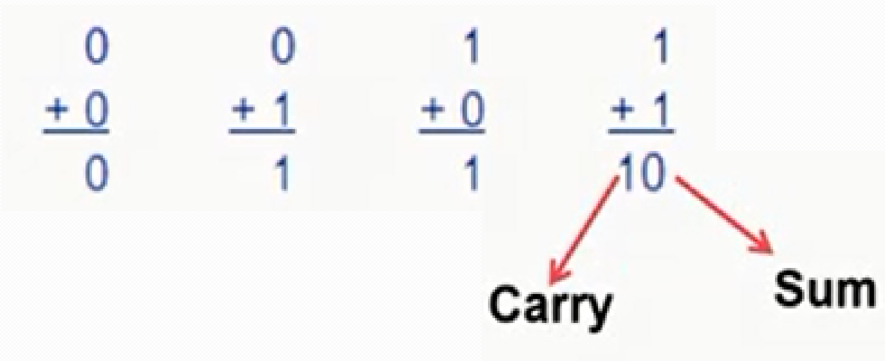
Binary Adder

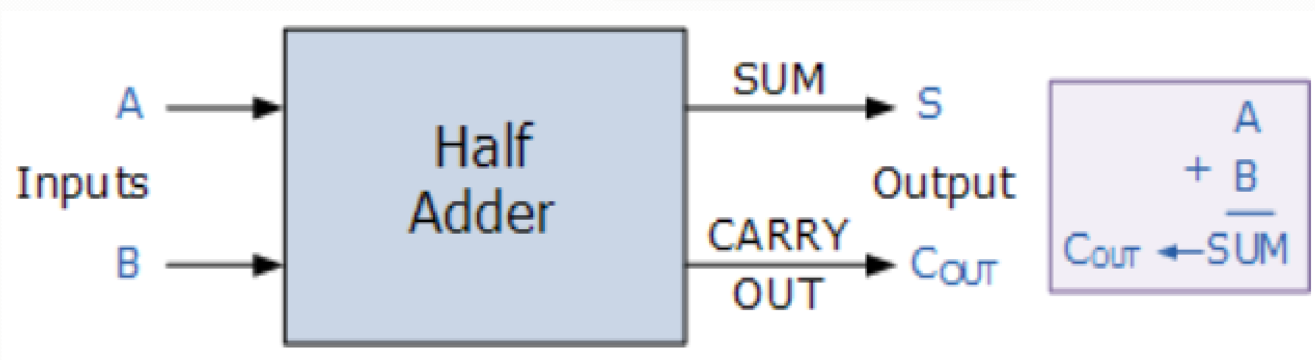
Binary Adders are arithmetic circuits in the form of **half-adders** and **full-adders** used to add together two binary digits.

1. Half adder
2. Full adder

Half Adder

A **half adder** is a logical circuit that performs an addition operation on two binary digits. The half adder produces a sum and a carry value which are both binary digits.





Block Diagram

Half Adder

- This 1-bit adder can be easily implemented with the help of **EXOR Gate** for the output '**SUM**' and an **AND Gate** for the **carry**.

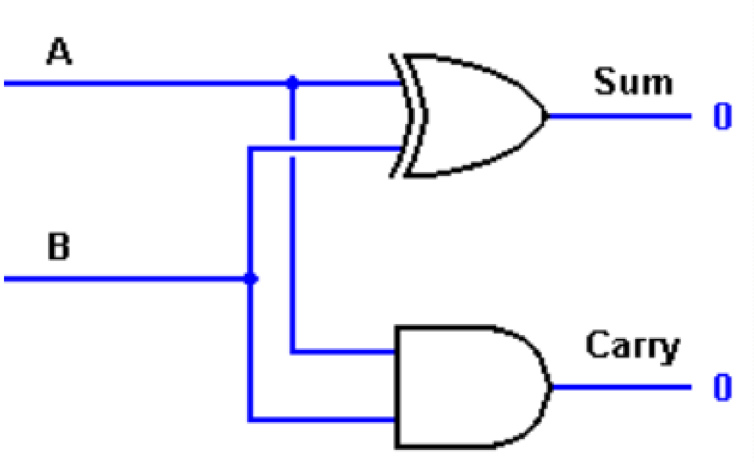
INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth Table

SUM $= A \oplus B$

Cout $= A.B$

Logic Function

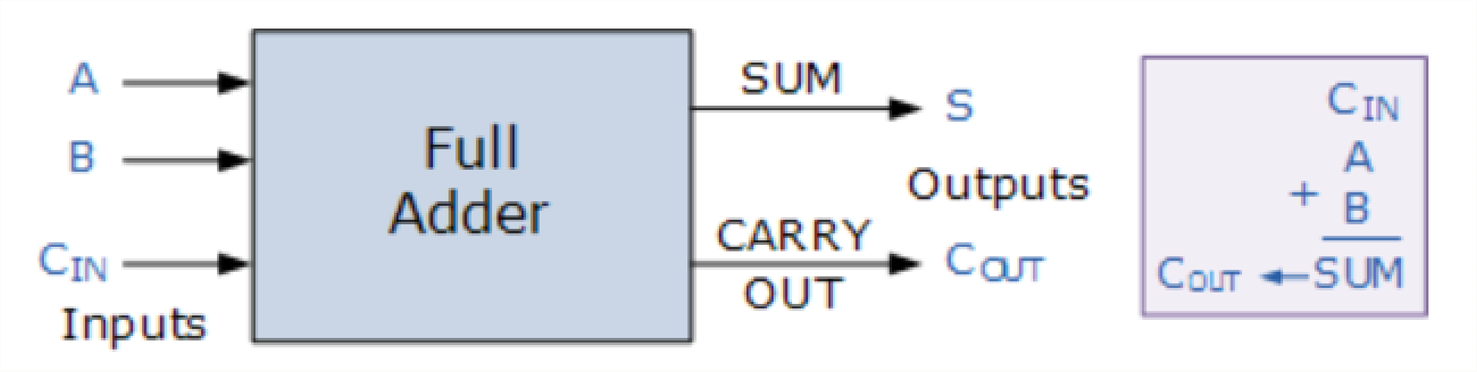


Logic Diagram

Full Adder

The main difference between the Full Adder and the previous Half Adder is that a full adder has three inputs. The same two single bit data inputs A and B as before plus an additional Carry-in (**C-in**) input to receive the carry from a previous stage as shown below.

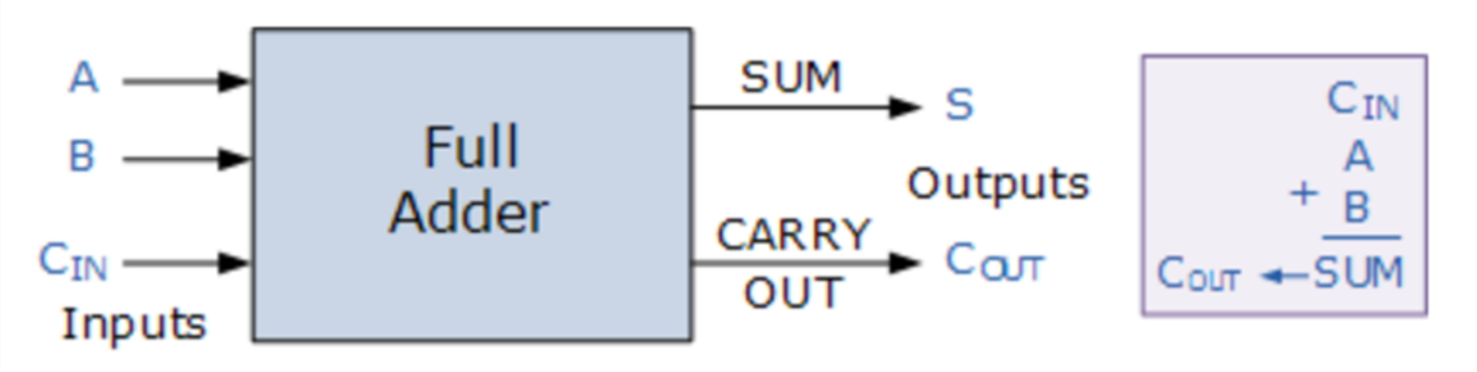
Full Adder Block Diagram



Full Adder

- When a full adder logic is designed we will be able to string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next.

- The output carry is designated as **Cout** and the normal output is designated as **S**.



Full Adder

$$S = \overline{A}\,\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\,\overline{C}_{in} + ABC_{in}$$

$$C = \overline{A}BC_{in} + A\overline{B}C_{in} + AB\overline{C}_{in} + ABC_{in}$$

SUM $= A \oplus B \oplus Cin$

Cout $= A.B + Cin(A \oplus B)$

INPUTS			OUTPUTS	
A	B	Cin	SUM	Cout
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

Full Adder

$$S = \overline{A}\,\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\,\overline{C}_{in} + ABC_{in}$$

$$C = \overline{A}BC_{in} + A\overline{B}C_{in} + AB\overline{C}_{in} + ABC_{in}$$

$$\begin{aligned}
\boxed{S} &= \overline{A}\,\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\,\overline{C}_{in} + ABC_{in} \\
&= (\overline{A}B + A\overline{B})\overline{C}_{in} + (\overline{A}\,\overline{B} + AB)C_{in} \\
&= (A \oplus B)\overline{C}_{in} + (\overline{A \oplus B})C_{in} \\
&= (A \oplus B) \oplus C_{in} = A \oplus B \oplus C_{in}
\end{aligned}$$

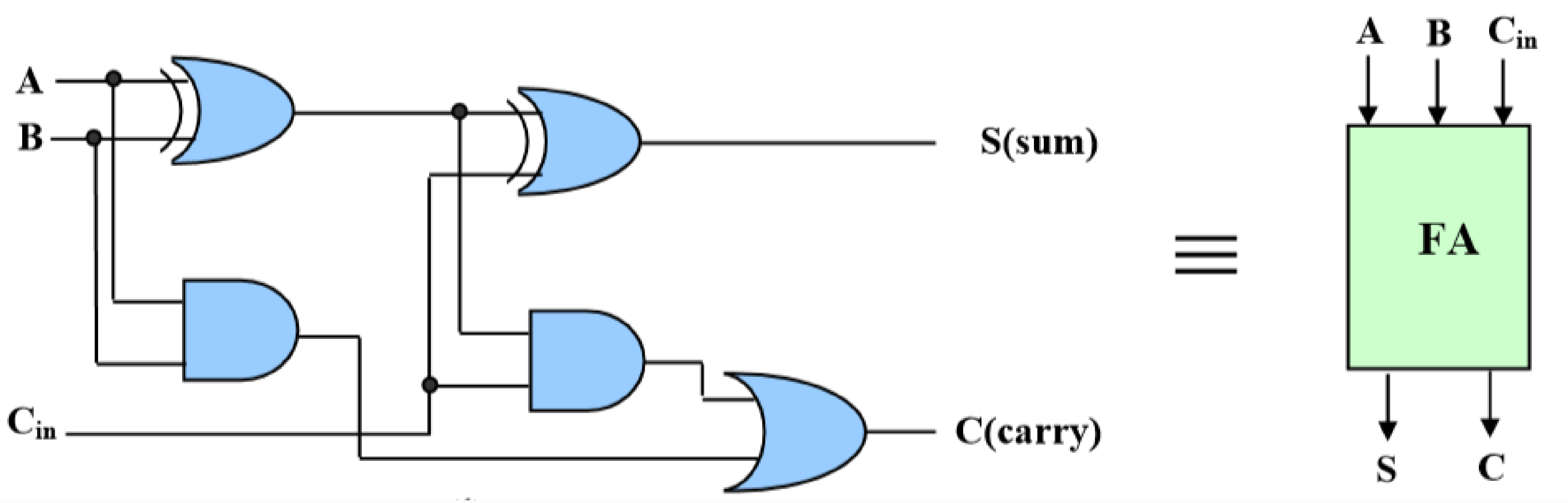
$$\begin{aligned}
\boxed{C} &= \overline{A}BC_{in} + A\overline{B}C_{in} + AB\overline{C}_{in} + ABC_{in} \\
&= (\overline{A}B + A\overline{B})C_{in} + AB(\overline{C}_{in} + C_{in}) \\
&= (A \oplus B)C_{in} + AB \Leftarrow (\overline{C}_{in} + C_{in} = 1)
\end{aligned}$$

INPUTS			OUTPUTS	
A	B	Cin	SUM	Cout
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

SUM $= A \oplus B \oplus Cin$

Cout $= A.B + Cin(A \oplus B)$

Full Adder



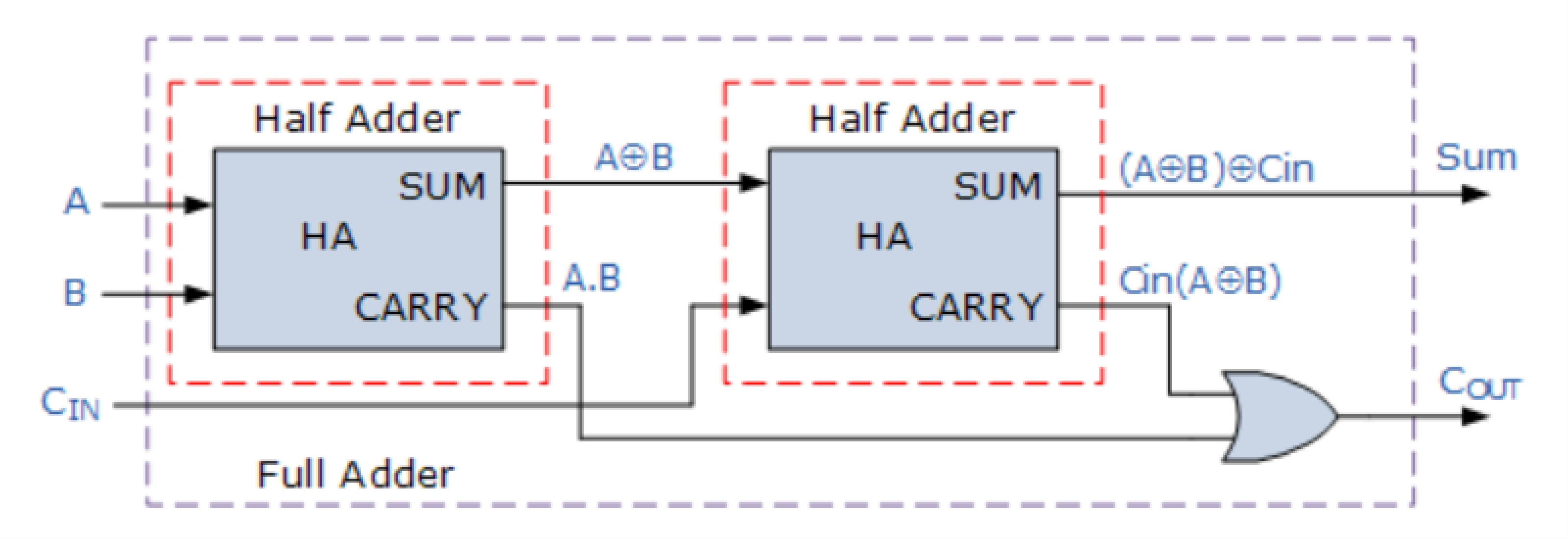
$$\textbf{SUM} = A \oplus B \oplus Cin$$

$$\textbf{Cout} = A.B + Cin(A \oplus B)$$

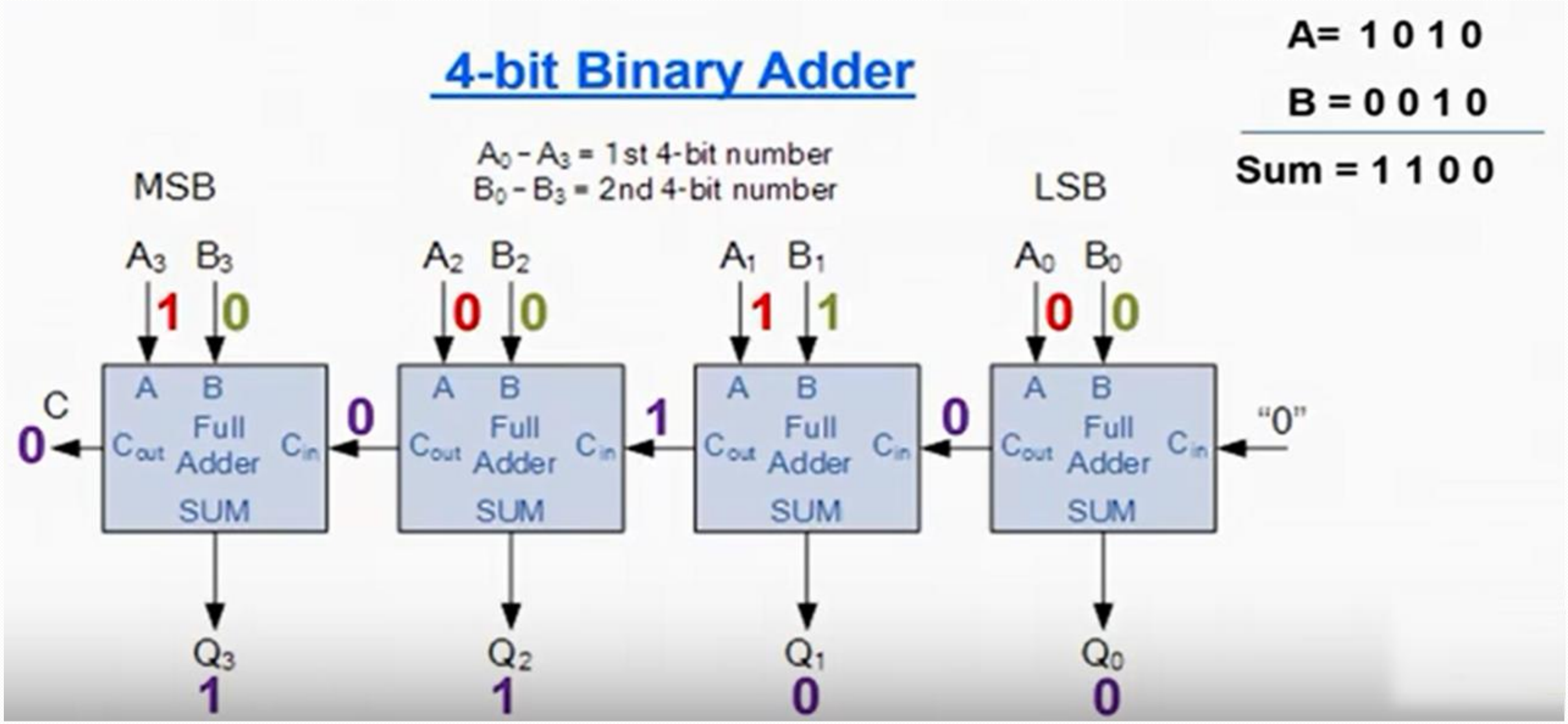
Full Adder

- In many ways, the full adder can be thought of as two half adders connected together, with the first half adder passing its carry to the second half adder as shown.

Full Adder Logic Diagram



4-bit Ripple Carry Adder



Half Subtractor

A	B	D	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

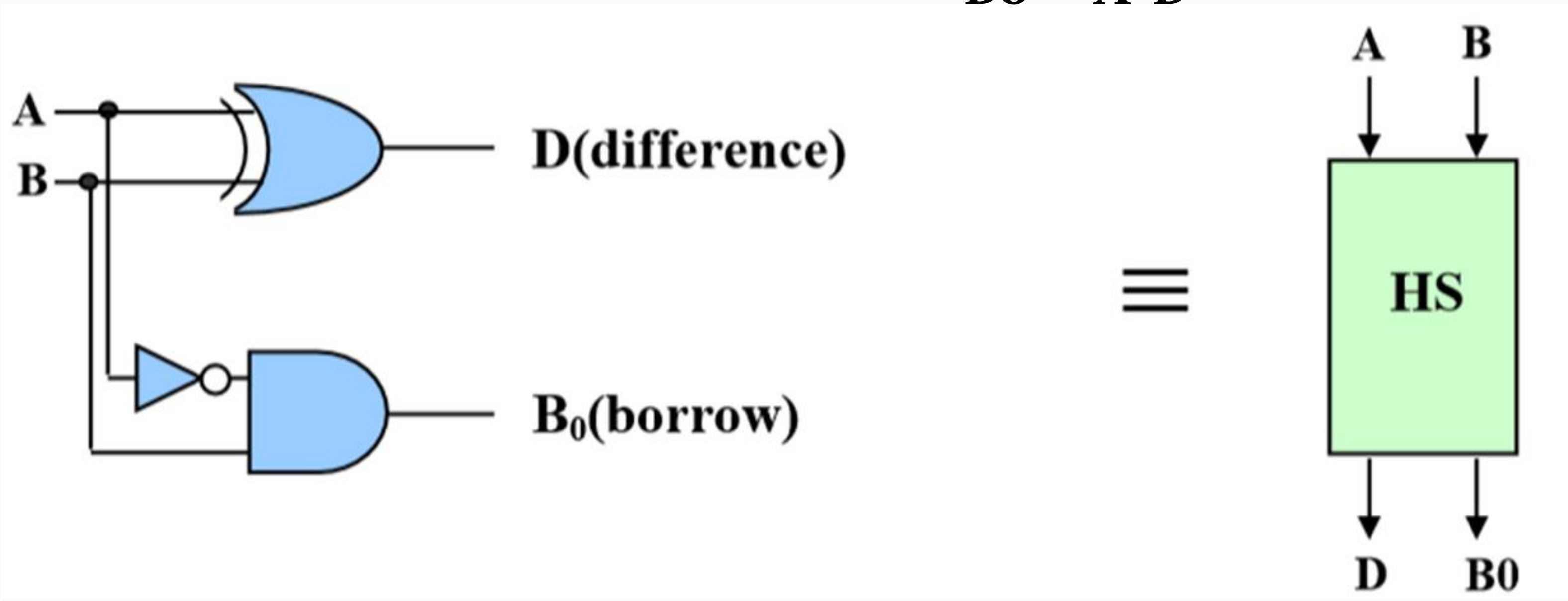
$D = \Sigma\, m(1,2)$

$D = A'B + AB'$

$D = A \oplus B$

$Bo = \Sigma\, m(1)$

$Bo = A'B$



Full Subtractor

$$D = \overline{A}\,\overline{B}B_{in} + \overline{A}B\overline{B}_{in} + A\overline{B}\,\overline{B}_{in} + ABB_{in}$$

$$D = (A \oplus B) \oplus B_{in} = A \oplus B \oplus B_{in}$$

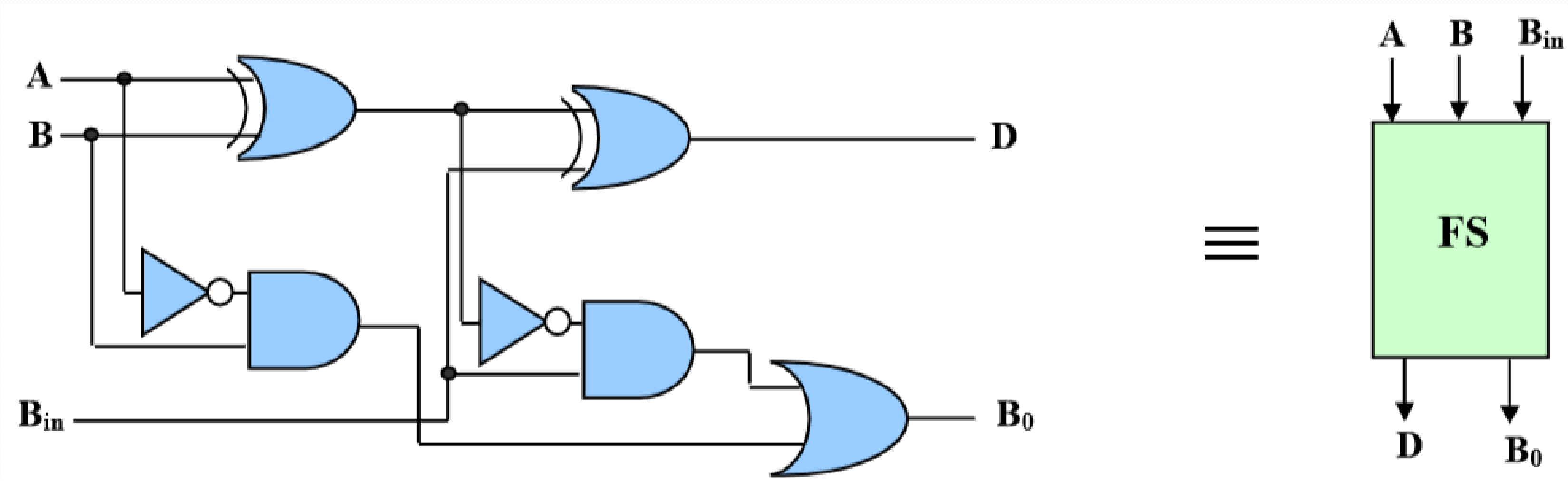
A	B	B_{in}	D	B_0
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

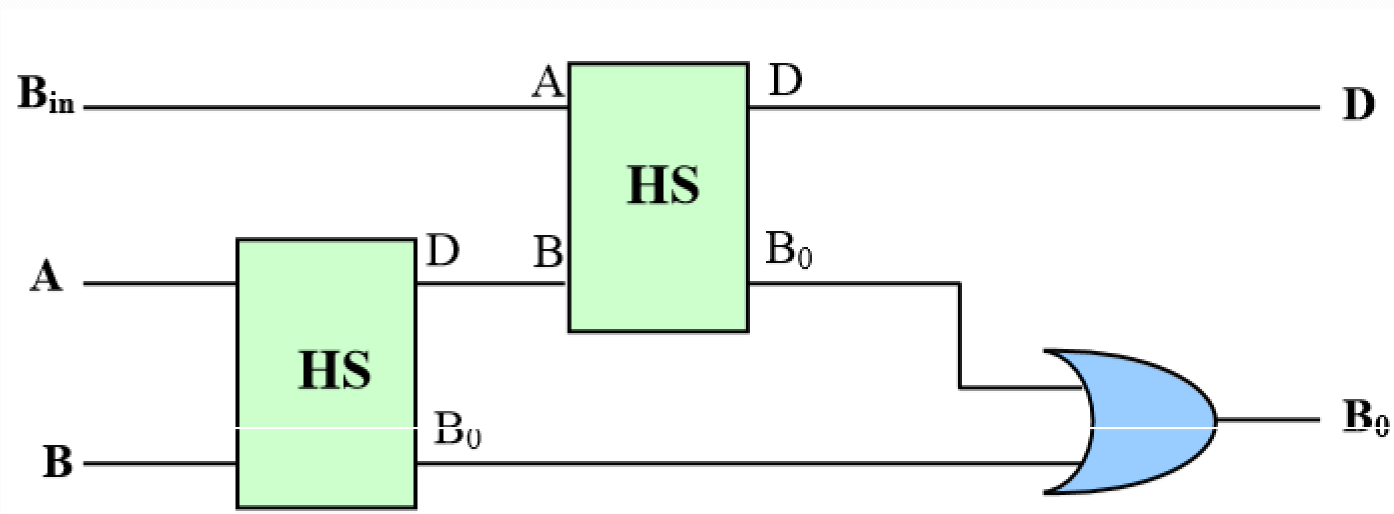
$$B_0 = \overline{A}\,\overline{B}B_{in} + \overline{A}B\overline{B}_{in} + \overline{A}BB_{in} + ABB_{in}$$

$$= B_{in}(\overline{AB} + AB) + \overline{A}B(\overline{B}_{in} + B_{in})$$

$$B_0 = B_{in}(\overline{A \oplus B}) + \overline{A}B \quad \Leftarrow \quad (\overline{B}_{in} + B_{in} = 1)$$

Full Subtractor





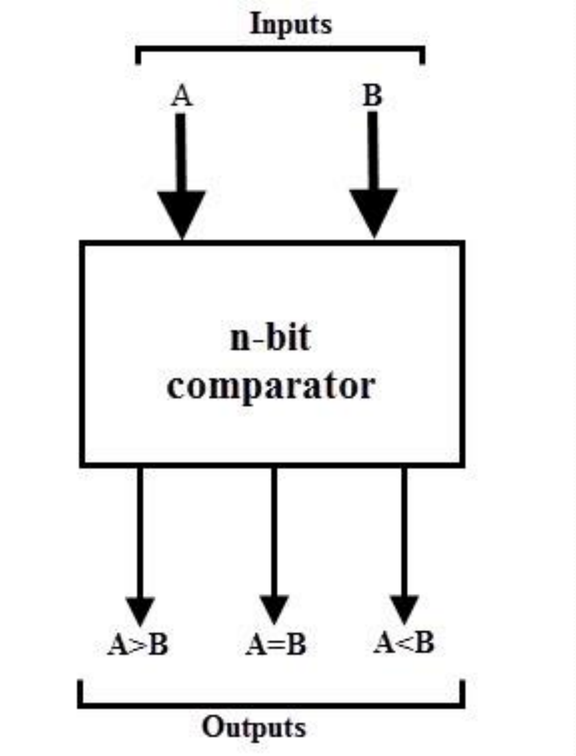
Practice

Use CircuitMaker to design a 4-bit adder and a 4-bit subtractor.

Digital Comparator

Digital Comparator is another very useful combinational logic circuit used to compare the value of two binary digits.

Digital or Binary Comparators are made up from standard AND, NOR and NOT gates that compare the digital signals present at their input terminals and produce an output depending upon the condition of those inputs.



Digital Comparator

These comparators can compare 2-bit, 4-bit and 8-bit numbers depending on the application requirement.

These are available in TTL as well as CMOS logic family ICs and some of these ICs include IC 7485 (4-bit comparator), IC 4585 (4-bit comparator in CMOS family) and IC 74AS885 (8-bit comparator).

Digital Comparator

- It is a combinational circuit that compares to numbers and determines their relative magnitude
- The output of comparator is usually 3 binary variables indicating: $A>B$

  $A=B$

  $A<B$

Digital Comparator

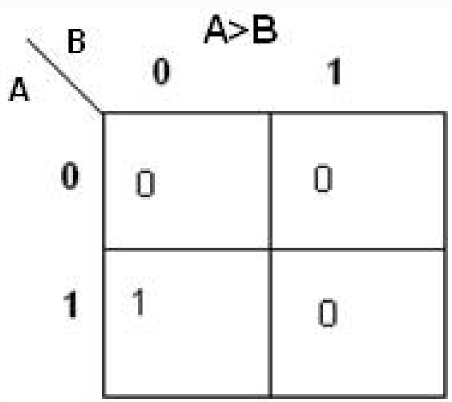
- **For example**, a magnitude comparator of two 1-bits, (A and B) inputs would produce the following three output conditions when compared to each other.

1-bit Digital Comparator Block Diagram

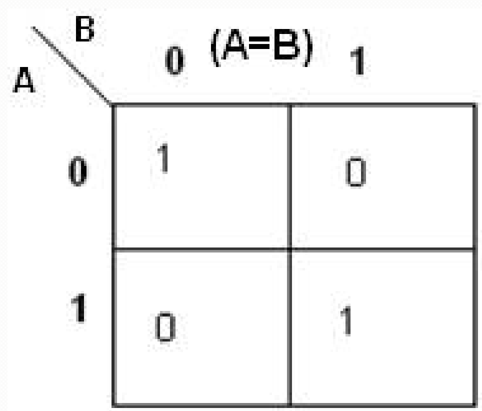
1-bit Digital Comparator Circuit

INPUTS		OUTPUTS		
A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

$A > B$, $A = B$, $A < B$

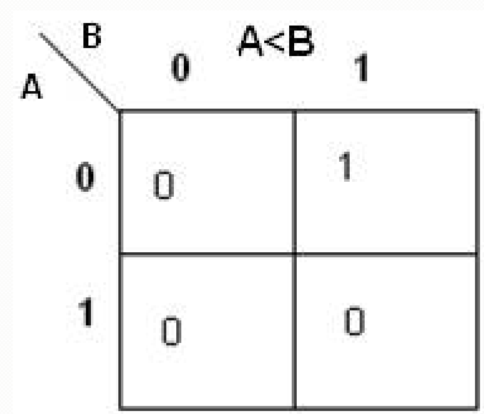


$A>B = A.\overline{B}$



$(A=B) = \overline{A}.\overline{B} + A.B$

$= A \text{ XNOR } B$

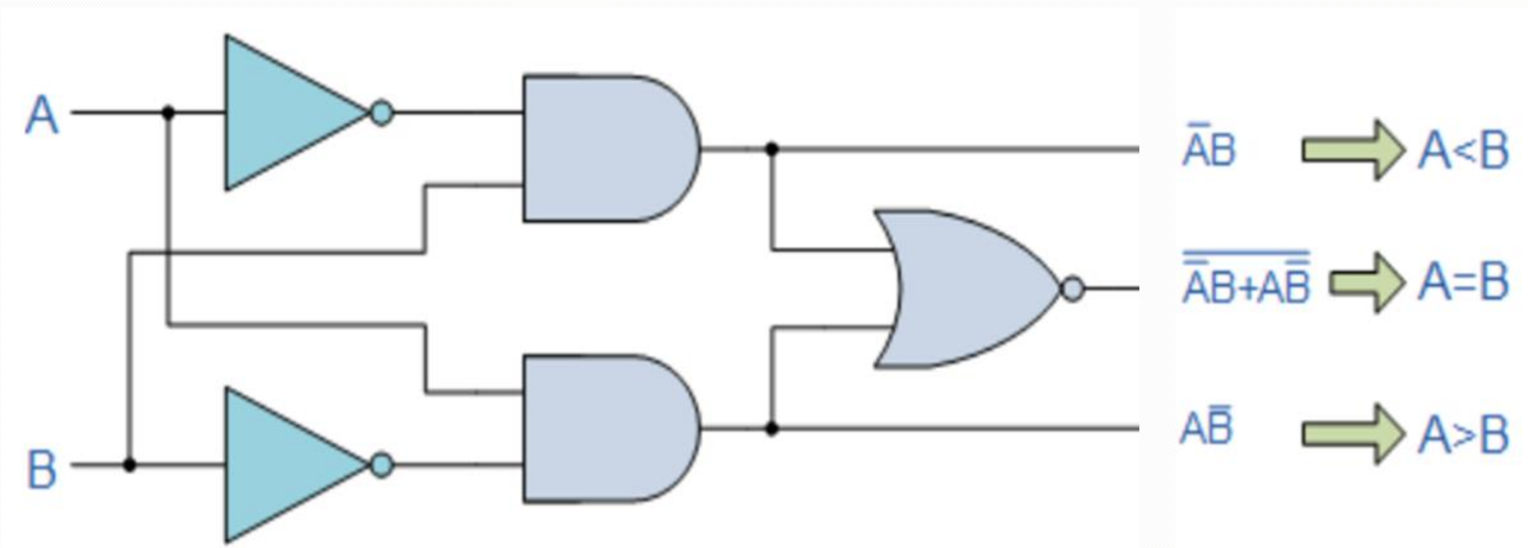


$A<B = \overline{A}.B$

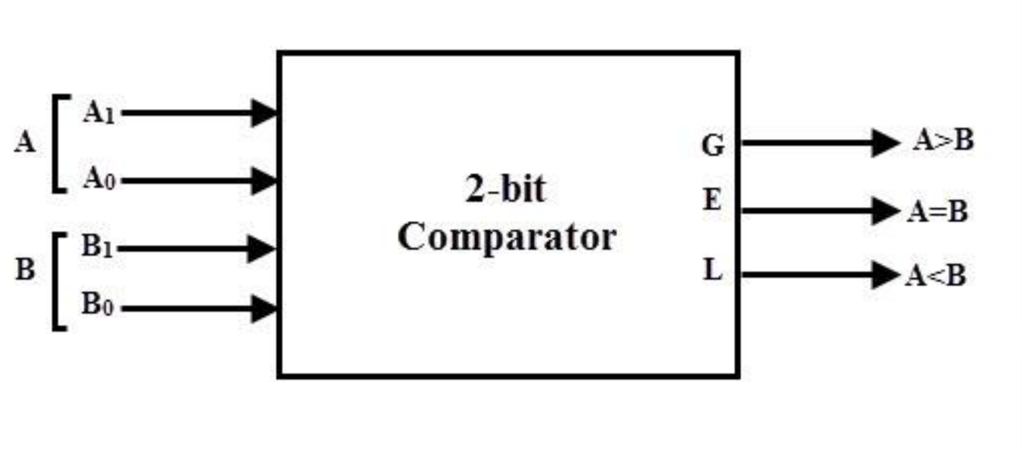
1-bit Digital Comparator Circuit

INPUTS		OUTPUTS		
A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

A > B, A = B, A < B



2-bit Digital Comparator Circuit



2-bit Digital Comparator Block Diagram

2-bit Digital Comparator Circuit

Inputs				Outputs		
A_1	A_0	B_1	B_0	$A>B$	$A=B$	$A<B$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

2-bit Digital Comparator Truth table

2-bit Digital Comparator Circuit

A>B

A1A0 \ B1B0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

$A>B = m(4, 8, 9, 12, 13, 14)$

$A>B: G = A0\,\overline{B1}\;\overline{B0} + A1\,\overline{B1} + A1\;A0\;\overline{B0}$

2-bit Digital Comparator Circuit

A =B

A1A0 \ B1B0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

A=B = m(0, 5, 10, 15)

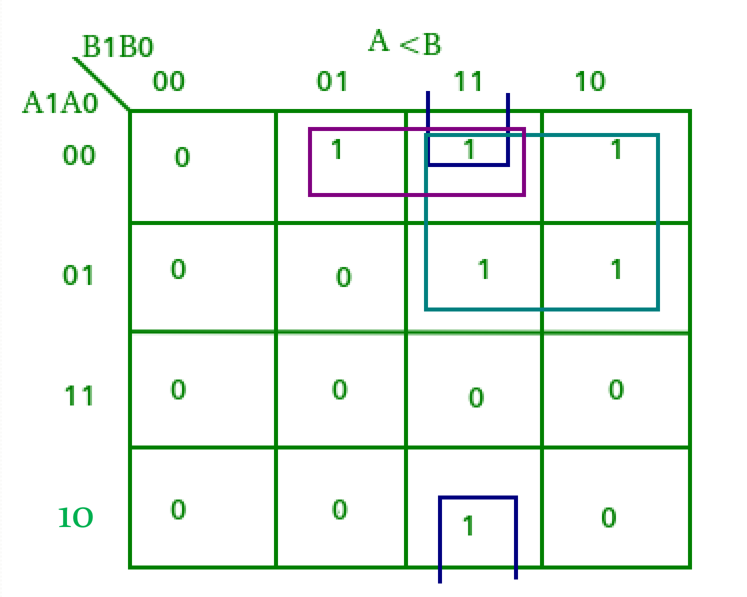
$$A = B: E = \overline{A1}\ \overline{A0}\ \overline{B1}\ \overline{B0} + \overline{A1}\ A0\ \overline{B1}\ B0 + A1\ A0\ B1\ B0 + A1\ \overline{A0}\ B1\ \overline{B0}$$

$$= \overline{A1}\ \overline{B1}\ (\overline{A0}\ \overline{B0} + A0\ B0) + A1\ B1\ (A0\ B0 + \overline{A0}\ \overline{B0})$$

$$= (A0\ B0 + \overline{A0}\ \overline{B0})\ (A1\ B1 + \overline{A1}\ \overline{B1})$$

$$= (A0 \text{ Ex-NOR } B0)\ (A1 \text{ Ex-NOR } B1)$$

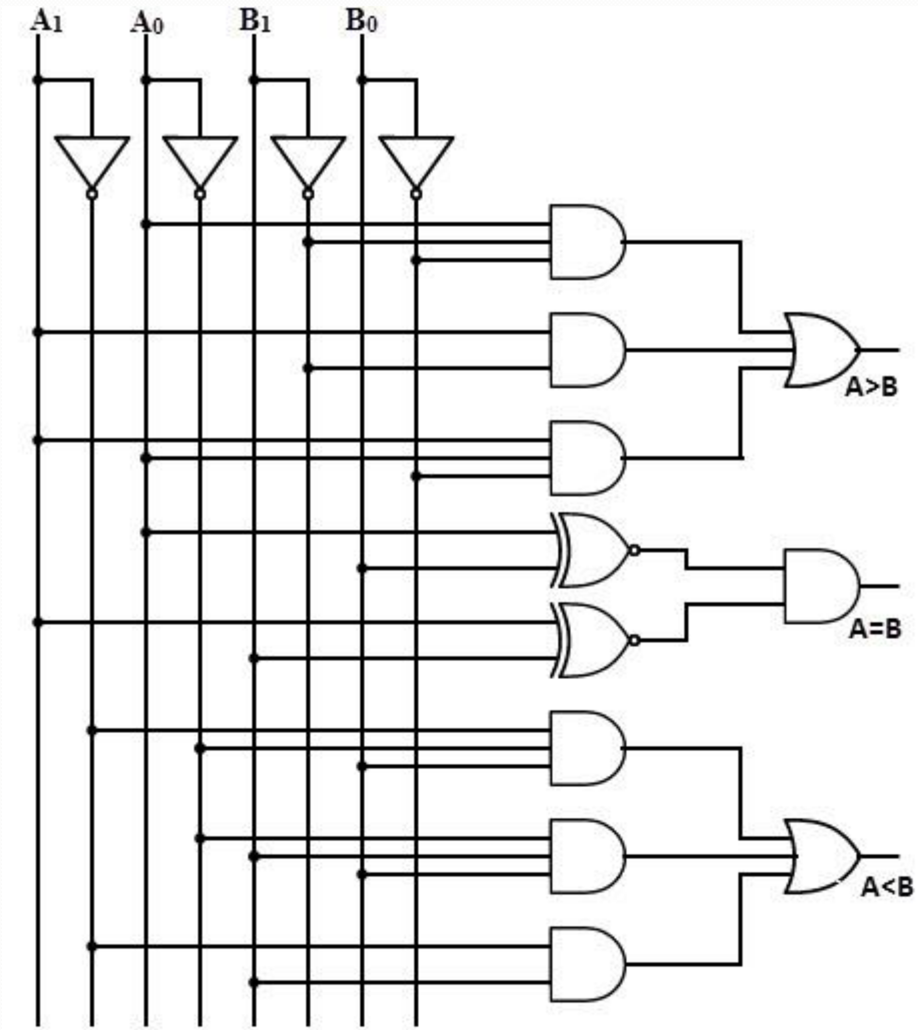
2-bit Digital Comparator Circuit



$A<B = m(1, 2, 3, 6, 7, 11)$

$$A < B: L = \overline{A1}\ B1 + \overline{A0}\ B1\ B0 + \overline{A1}\ \overline{A0}\ B0$$

2-bit Digital Comparator Circuit



Try

Design a four-bit combinational circuit 2's complementer using logic gates. (The output generates the 2's complement of the input binary number.)

Homework 8

Design a four-bit Multiplier using logic gates and show your work.

Deadline: December 2, 2022 @ 11:59 PM