**Department of Biomedical Engineering**
**Digital Logic Design / Second stage**
**Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi**
**Email: hussam.jawad@uomus.edu.iq**

# Lecture Six

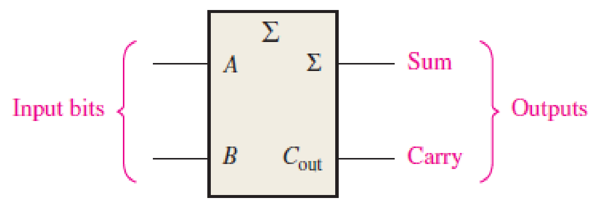
# Functions of Combinational Logic (Adder, Comparator)

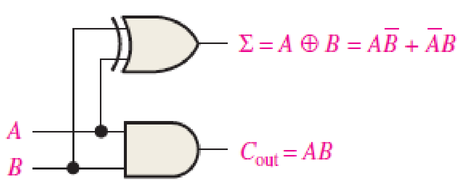
## Half and Full Adders

Adders are important in computers and also in other types of digital systems in which numerical data are processed. An understanding of the basic adder operation is fundamental to the study of digital systems. In this section, the half-adder and the full-adder are introduced.

### The Half-Adder

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 10$$




$$\Sigma = A \oplus B$$

$$C_{out} = AB$$

Half-adder truth table.

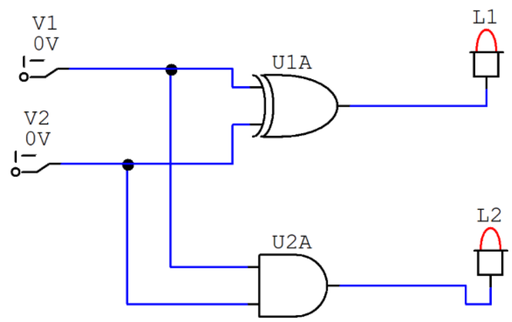
| $A$ | $B$ | $C_{out}$ | $\Sigma$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

***Example:*** Determine the sum ($\sum$) and the output carry ($C_{out}$) of a half-adder for each set of input bits:

(a) 0 +1 (b) 1+0 (c) 1+1

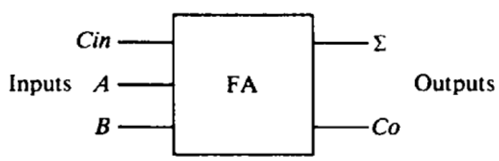
***Solution***

| A | B | $C_{out}$ | $\sum$ | Decimal number |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 2 |



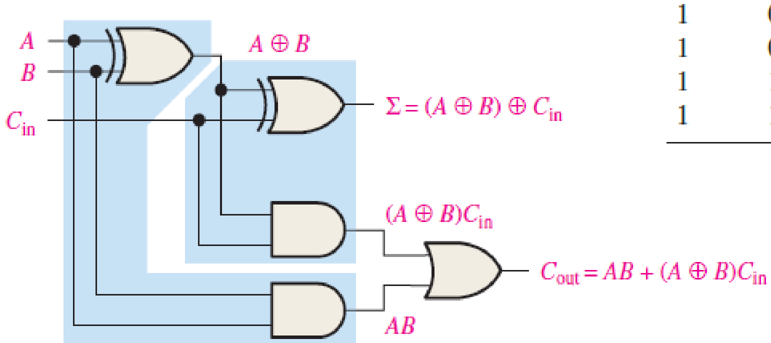

## The Full-Adder

The full adder accepts two input bits and an input carry and generates a sum output and an output carry.

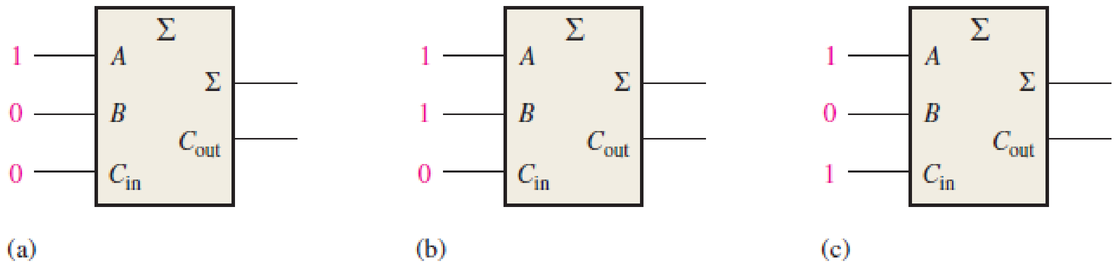



Full-adder truth table.

| $A$ | $B$ | $C_{in}$ | $C_{out}$ | $\Sigma$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |



**Example:** For each of the three full adders in the Figure below, determine the outputs for the inputs shown.




***Solution***

a- The input bits are $A = 1$, $B = 0$, and $Cin = 0$.

$$1 + 0 + 0 = 1 \text{ with no carry}$$

Therefore, $\sum = 1$ and $C_{out} = 0$.

b- The input bits are $A = 1$, $B = 1$, and $Cin = 0$.

$$1 + 1 + 0 = 0 \text{ with a carry of } 1$$

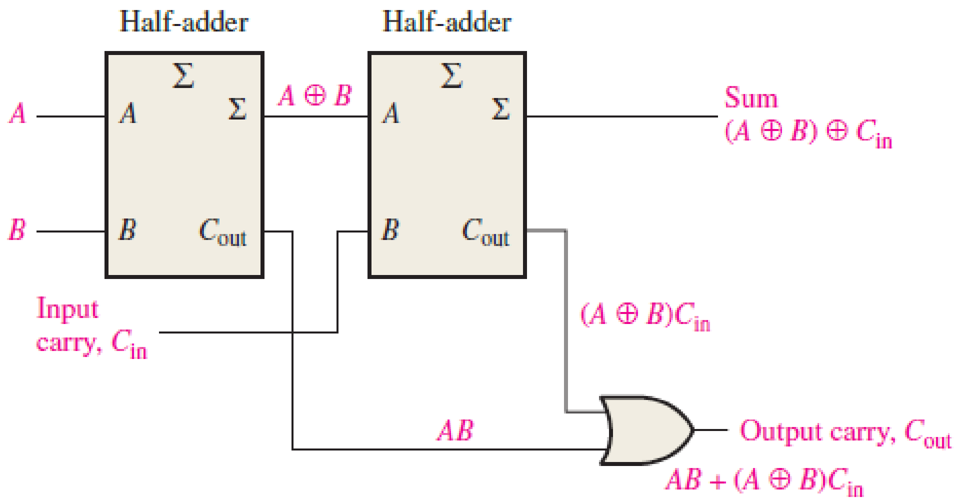
Therefore, $\sum = 0$ and $C_{out} = 1$.

c- The input bits are $A = 1$, $B = 0$, and $Cin = 1$.

$$1 + 0 + 1 = 0 \text{ with a carry of } 1$$

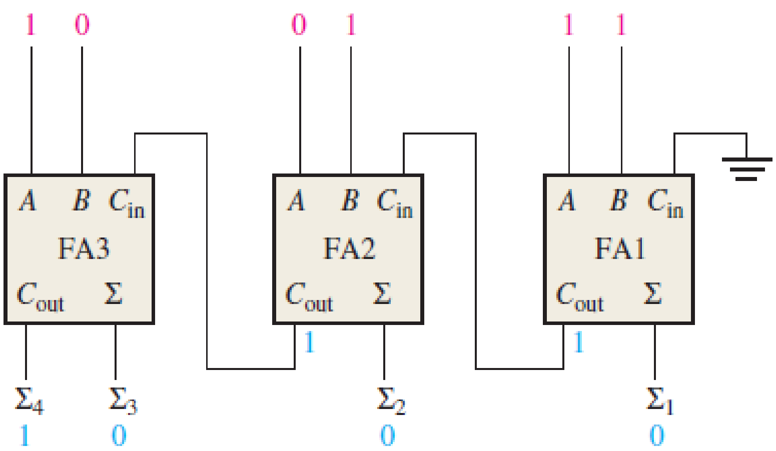
Therefore, $\sum = 0$ and $Cout = 1$.

## Parallel Binary Adders




***Example:*** Determine the sum generated by the 3-bit parallel adder and show the intermediate carriers when the binary numbers 101 and 011 are being added.

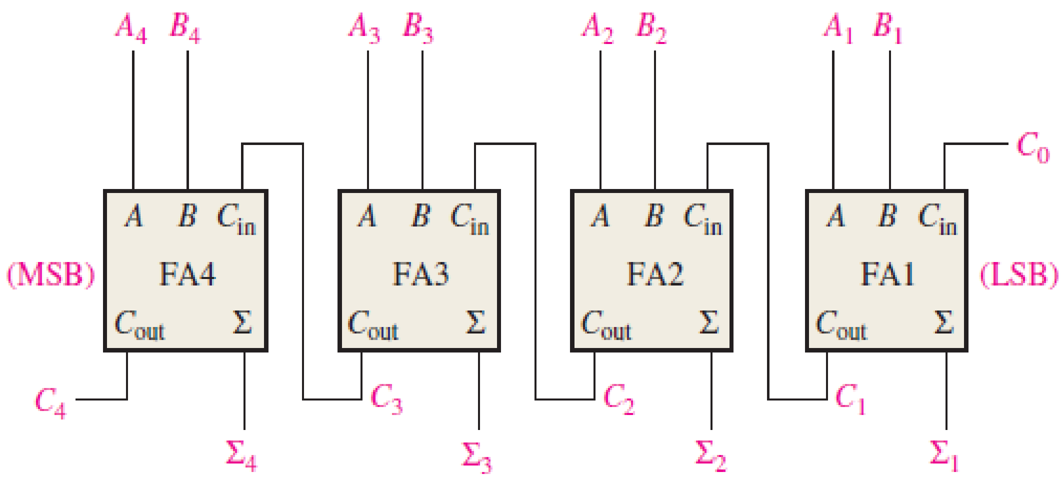
### *Solution*



### *4-bit parallel adder*

***Example*** Use the 4-bit parallel adder to find the sum and output carry for the addition of the following two 4-bit numbers if the input carry ($C_0$) is 0:

$$A_4A_3A_2A_1 = 1100 \text{ and } B_4B_3B_2B_1 = 1100$$




***Solution***

For $n = 1$: $A_1 = 0$, $B_1 = 0$, and $C_0 = 0$. From the 1st row,

$\sum = \mathbf{0}$ and $C1 = 0$

For $n = 2$: $A_2 = 0$, $B_2 = 0$, and $C_1 = 0$. From the 1st row of the table,

$\sum = \mathbf{0}$ and $C_2 = 0$

For $n = 3$: $A_3 = 1$, $B_3 = 1$, and $C_2 = 0$. From the 4th row of the table,

$\sum = \mathbf{0}$ and $C_3 = 1$

For $n = 4$: $A_4 = 1$, $B_4 = 1$, and $C_3 = 1$. From the last row of the table,

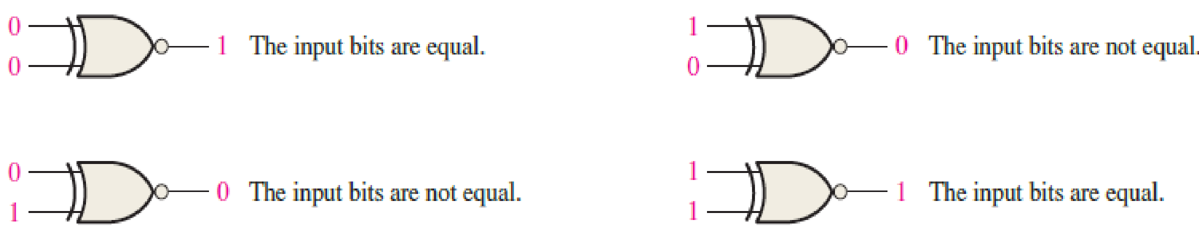
$\sum = \mathbf{1}$ and $C_4 = \mathbf{1}$

$C4$ becomes the output carry; the sum of 1100 and 1100 is **11000.**

## 2- Comparators

The basic function of a comparator is to compare the magnitudes of two binary quantities to determine the relationship of those quantities. In its simplest form, a comparator circuit determines whether two numbers are equal.

the exclusive-NOR gate can be used as a basic comparator because its output is a 0 if the two input bits are not equal and a 1 if the input bits are equal.

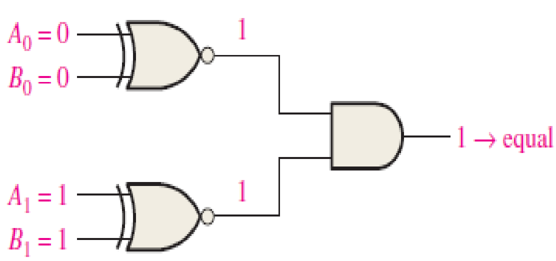



***Example***:

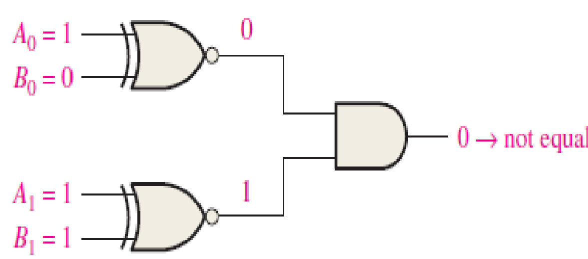
Compare the digital numbers to determine whether they are equal or not

- **a-** 10 and 10
- **b-** 11 and 10

***Solution***




(a)




(b)