# Basic Electrical and Electronics Engineering

## Module 4

**Dr. Sonam Shrivastava/ Assistant Professor (Sr.) /SELECT**

**LECTURE 5**

# Module 4

# Digital Systems

## Lecture 4

## Topics to be covered

- **HALF ADDER**
- **FULL ADDER**

# Half Adder

**Half Adder**: is a combinational circuit that performs the addition of two bits, this circuit needs two binary inputs and two binary outputs.

| Inputs | | Outputs | |
|---|---|---|---|
| *X* | *Y* | *C* | *S* |
| *0* | *0* | *0* | *0* |
| *0* | *1* | *0* | *1* |
| *1* | *0* | *0* | *1* |
| *1* | *1* | *1* | *0* |

***Truth table***

*The simplified Boolean function from the truth table:*

$$\begin{cases} S = \bar{X}Y + X\bar{Y} \\ C = XY \end{cases} \quad 1\Big\}$$ (Using sum of product form)

Where $S$ is the sum and $C$ is the carry.

$$\begin{cases} S = X \oplus Y \\ C = XY \end{cases} \quad 2\Big\}$$ (Using **XOR** and **AND** Gates)

## Half Adder Truth Table:

| Inputs | | Outputs | |
|---|---|---|---|
| X | Y | S | C-out |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

$S(X,Y) = \Sigma\,(1,2)$

$S = X'Y + XY'$

$S = X \oplus Y$

$C\text{-out}(x,\ y,\ C\text{-in}) = \Sigma\,(3)$

$C\text{-out} = XY$

For Carry

| A \ B | | |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

Carry = AB

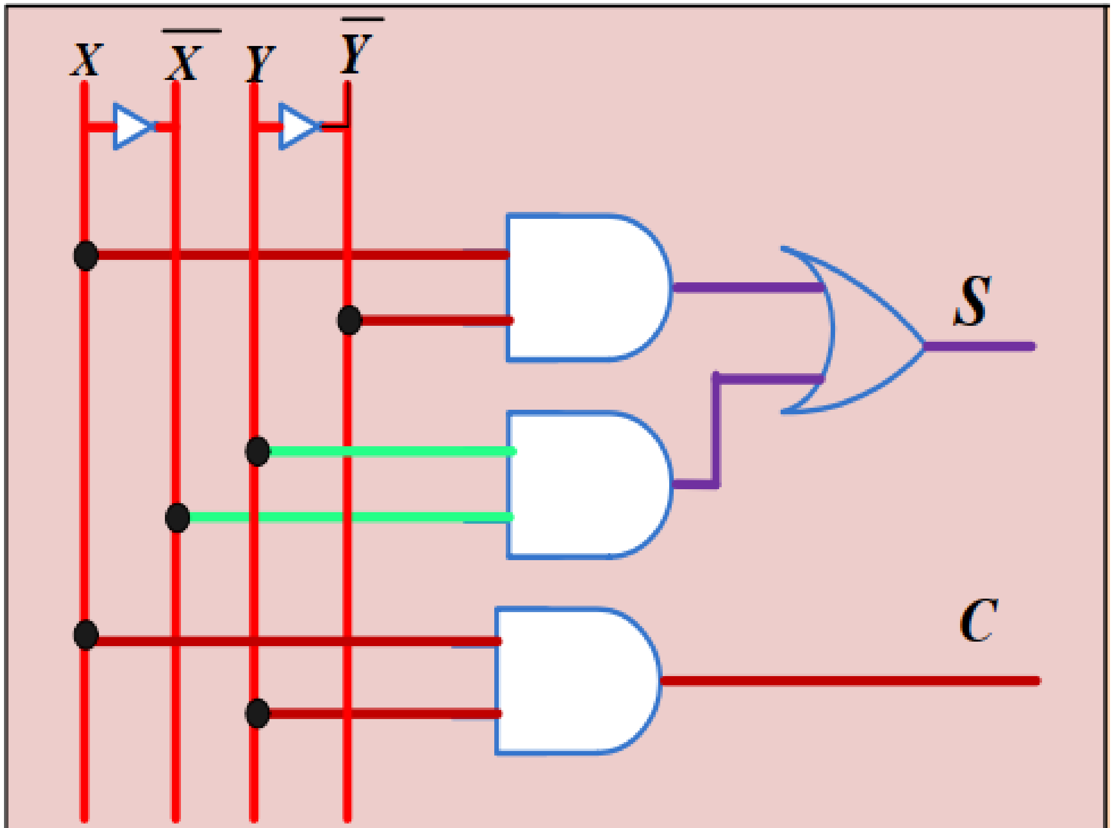
For Sum

| A \ B | | |
|---|---|---|
| | 0 | 1 |
| | 1 | 0 |

Sum = $A\overline{B}+\overline{A}B$

$=A\oplus B$

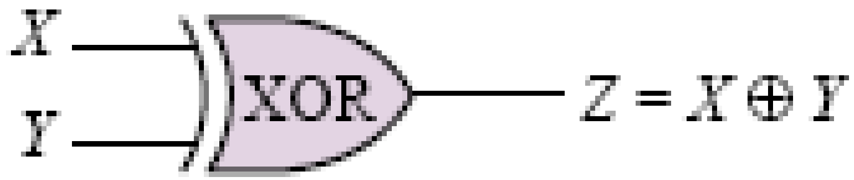
$$\begin{cases} S = \bar{X}Y + X\bar{Y} \\ C = XY \end{cases} \quad \color{red}{1}\Big\} \text{ (Using sum of product form)}$$




*Implementation of Half Adder using equation (1)*

# XOR (Exclusive OR) Gate

$$Z = X'Y + XY'$$

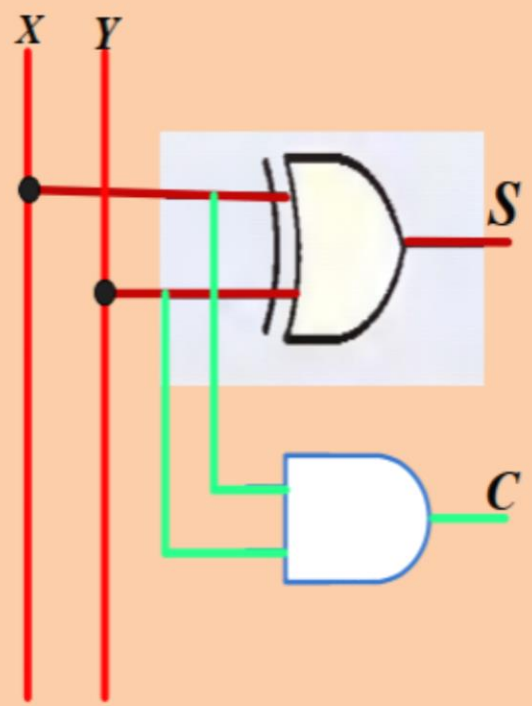



$$Z = X \oplus Y$$

| $X$ | $Y$ | $Z$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Truth table

$$\begin{cases} S = X \oplus Y \\ C = XY \end{cases} \qquad 2\}$$ (Using **XOR** and **AND** Gates)




*Implementation of Half Adder using equation (2)*

# Full Adder

**Full Adder** is a combinational circuit that performs the addition of three bits (two significant bits and previous carry).

- It consists of ***three inputs and two outputs***, two inputs are the bits to be added, the third input represents the carry form the previous position.
- The full adder is usually a component in a cascade of adders, which add 8, 16, etc, binary numbers.

# Full Adder Truth Table

| Inputs | | | Outputs | |
|---|---|---|---|---|
| $X$ | $Y$ | $C_{in}$ | $S$ | $C_{out}$ |
| 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 |

*Truth table for the full adder*

- The $S$ output is equal to *1* when only one input is equal to *1* or when all three inputs are equal to *1*.
- The $C_{out}$ output has a carry *1* if two or three inputs are equal to *1*.
- The Karnaugh maps and the simplified expression are shown in the following figures:

## Full Adder Truth Table

| Inputs | | | Outputs | |
|---|---|---|---|---|
| X | Y | C-in | S | C-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 |

S(X,Y, C-in) = $\Sigma$ (1,2,4,7)

C-out(x, y, C-in) = $\Sigma$ (3,5,6,7)

### Sum S

X spans columns 11 and 10; Y spans columns 01 and 11.

| C-in \ XY | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 2: 1 | 6 | 4: 1 |
| 1 | 1: 1 | 3 | 7: 1 | 5 |

S = X'Y'(C-in) + X'Y(C-in)' + XY'(C-in)' + XY(C-in)

S = X $\oplus$ Y $\oplus$ (C-in)

### Carry C-out

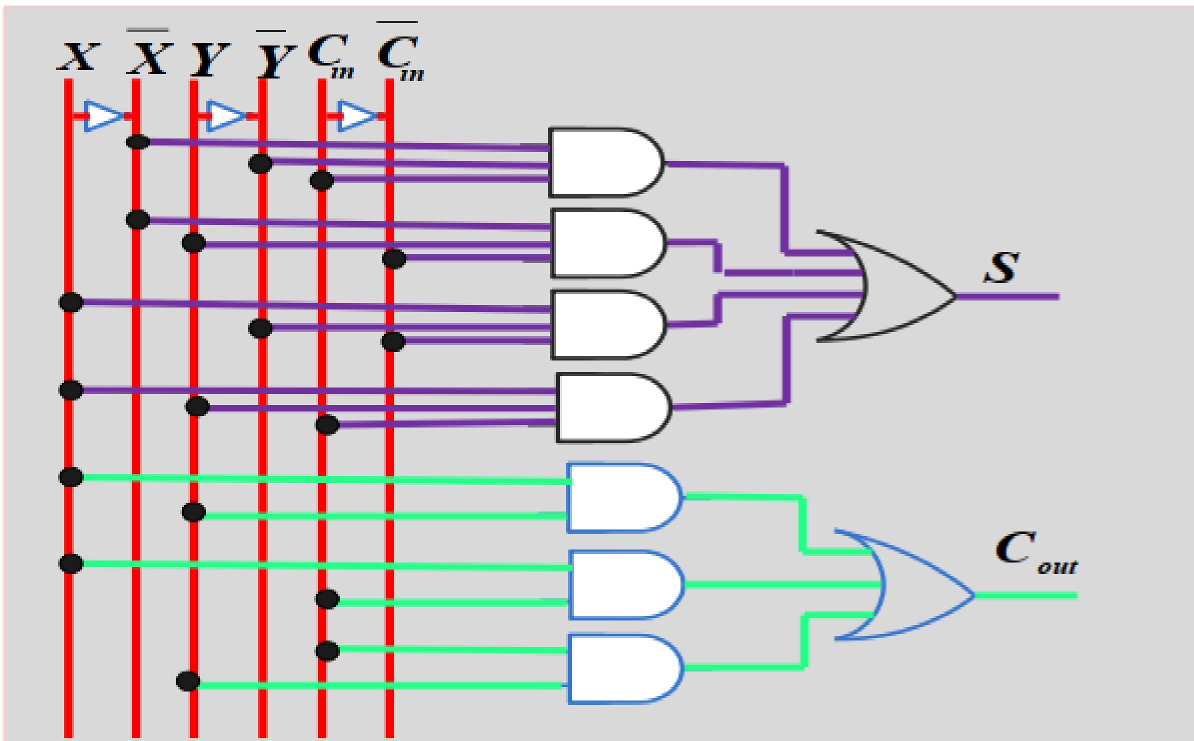
X spans columns 11 and 10; Y spans columns 01 and 11.

| C-in \ XY | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 2 | 6: 1 | 4 |
| 1 | 1 | 3: 1 | 7: 1 | 5: 1 |

C-out = XY + X(C-in) + Y(C-in)

$$\left\{\begin{array}{l} S = \bar{X}\,\bar{Y}C_{in} + \bar{X}Y\overline{C_{in}} + X\bar{Y}\,\overline{C_{in}} + XYC_{in} \\ \\ C_{out} = XY + XC_{in} + YC_{in} \end{array}\right\} \quad 1 \quad \textit{(Sum of products)}$$

- The ***logic diagrams*** for the full adder implemented in ***sum-of-products*** form are the following:




*logic diagram for the full adder*

- It can also be implemented using ***two half adders*** and ***one OR gate*** (using ***XOR*** gates).

$$\left\{\begin{array}{l} S = C_{in} \oplus (X \oplus Y) \\ C_{out=} C_{in} \cdot (X \oplus Y) + XY \end{array}\right\}$$

***Proof:***

***The sum:***

$$S = \bar{X}\,\bar{Y}C_{in} + \bar{X}Y\overline{C_{in}} + X\bar{Y}\,\overline{C_{in}} + XYC_{in}$$

$$= \overline{C_{in}}(\bar{X}Y + X\bar{Y}) + C_{in}(\bar{X}\,\bar{Y} + XY)$$

$$= \overline{C_{in}}(\bar{X}Y + X\bar{Y}) + C_{in}\overline{(\bar{X}Y + X\bar{Y})}$$

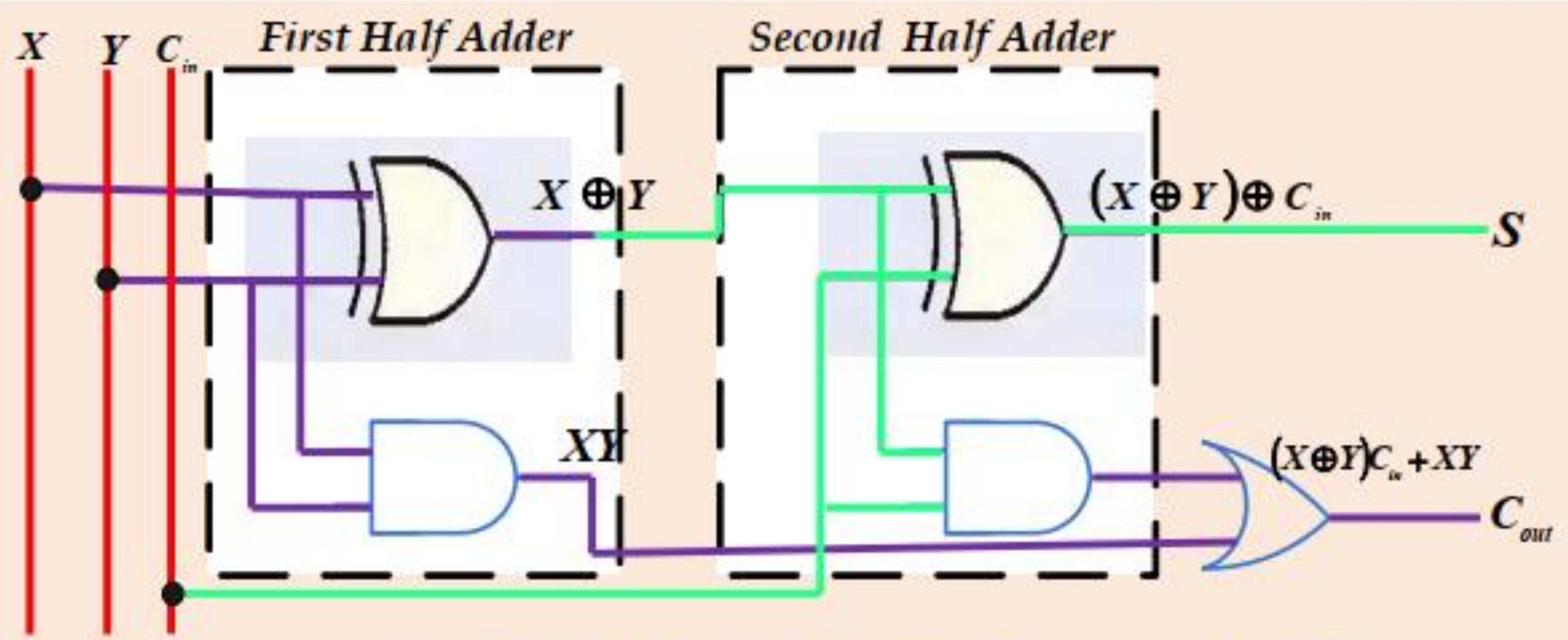
$$S = C_{in} \oplus (X \oplus Y)$$

***The carry output:***

$$C_{out} = \bar{X}YC_{in} + X\bar{Y}C_{in} + XYC_{in} + XY\,\overline{C_{in}}$$

$$= C_{in}(\bar{X}Y + X\bar{Y}) + XY(C_{in} + \overline{C_{in}})$$

$$C_{out=} C_{in} \cdot (X \oplus Y) + XY$$




Implementation of Full Adder with two Half Adders and an OR gate