Digital Logic Design (CSNB163)

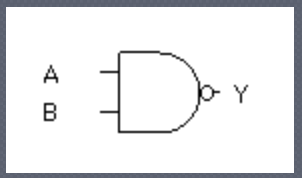
Module 7

Recaps.. NAND gate

- In Module 3 we have learned about NAND gate – it is a combination of **AND** operation **followed by NOT** operation
- Symbol

$$\overline{A \cdot B} = Y$$

- Logic Gate



- Truth table

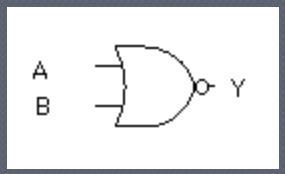
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Recaps… NOR gate

- In Module 3 we have learned about NOR gate – it is a combination of **OR** operation **followed by NOT** operation
- Symbol

  $A + B = Y$

- Logic Gate



- Truth table

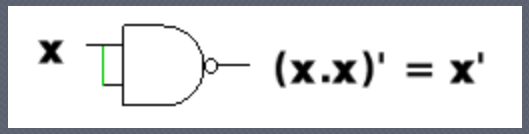
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Recaps… NAND and NOR gates

- In Module 3, we have also learned that digital circuits are **more frequently constructed with NAND or NOR gates** rather than with AND or OR gates.
- The main reason behind this is that NAND and NOR gates are **easier to fabricate** with electronic components.
- NAND and NOR gates are said to be **universal** gate because any digital logic design can be constructed just by using NAND and NOR gates.

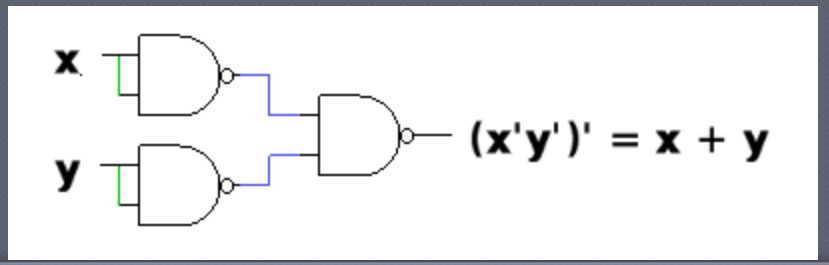
Universality of NAND gate

- In Module 3, we have learned that digital logic design are derived from 3 basic logic operations namely NOT, AND and OR.
- The universality of NAND gate allows it to represent:
  - NOT operation



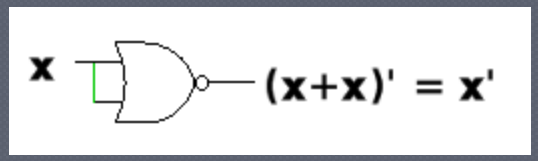
  - AND operation

  - OR operation

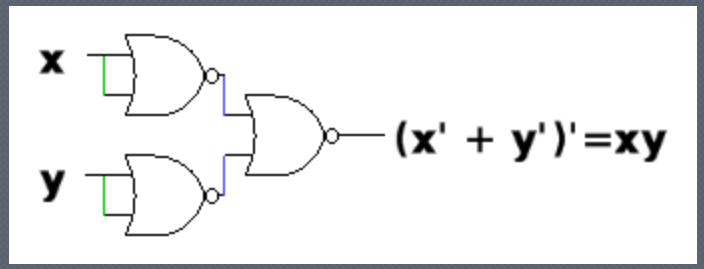


Universality of NOR gate

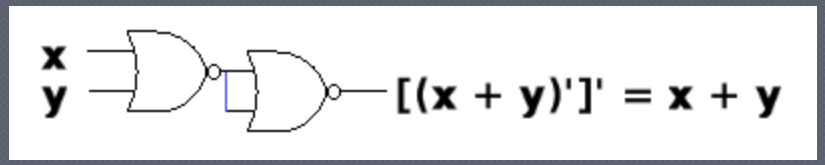
- In Module 3, we have learned that digital logic design are derived from 3 basic logic operations namely NOT, AND and OR.
- The universality of NOR gate allows it to represent:
  - NOT operation



  - AND operation



  - NOR operation

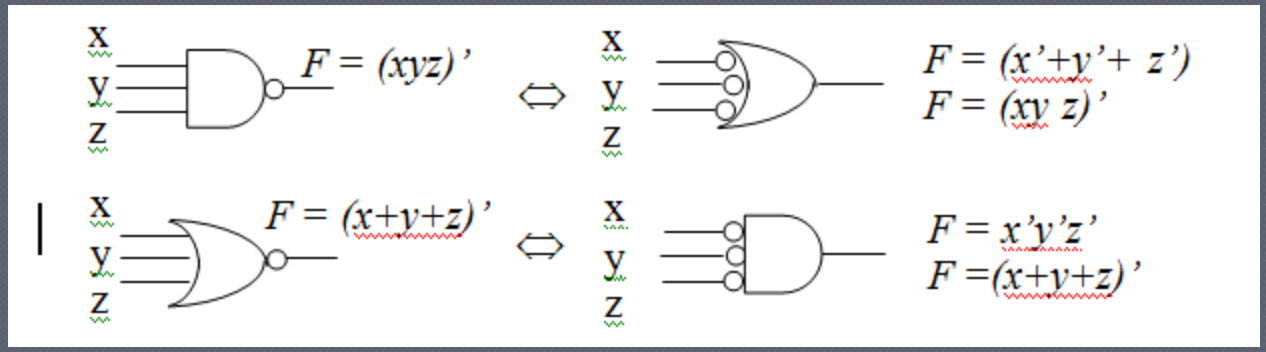


Recaps.. De Morgan's Theorem

- In Module 4, we have learned about DeMorgan's Law, whereby:

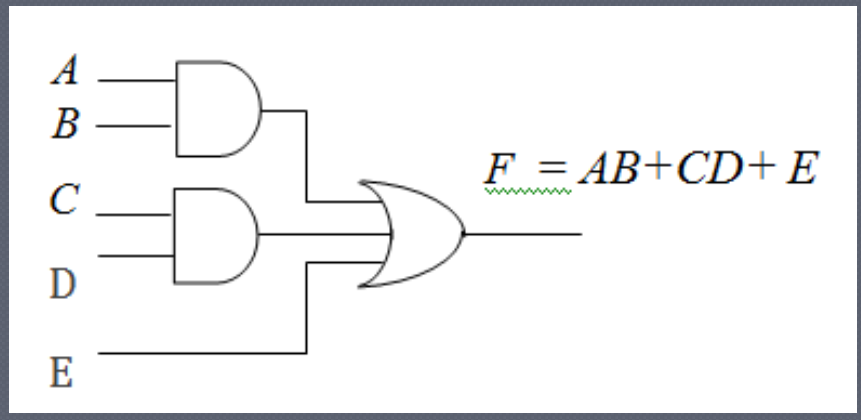
$$\overline{(a \cdot b)} = \overline{a} + \overline{b} \qquad \overline{(a + b)} = \overline{a} \cdot \overline{b}$$

- Thus, the following property holds true:

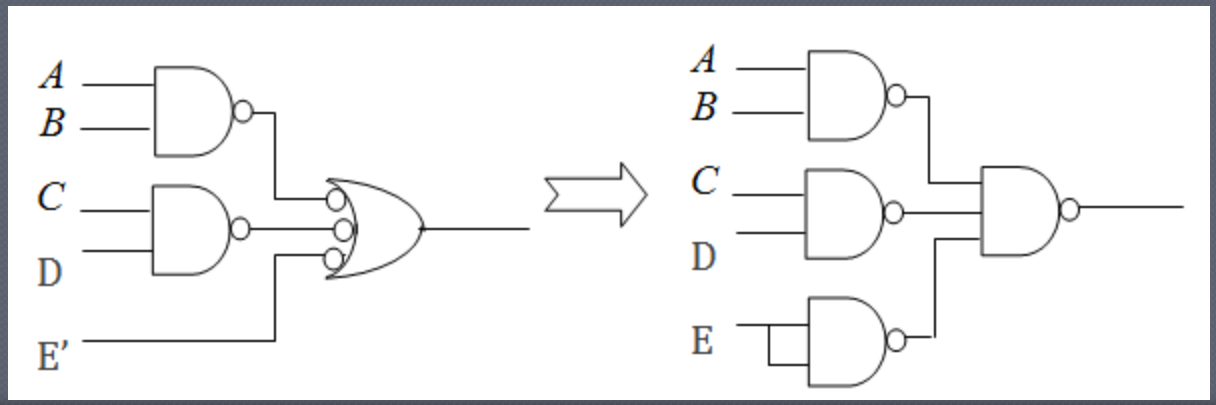


NAND Implementation

- Any **Sum of Products** Boolean expression can be implemented using **NAND gates**.
- E.g. F = AB + CD + E

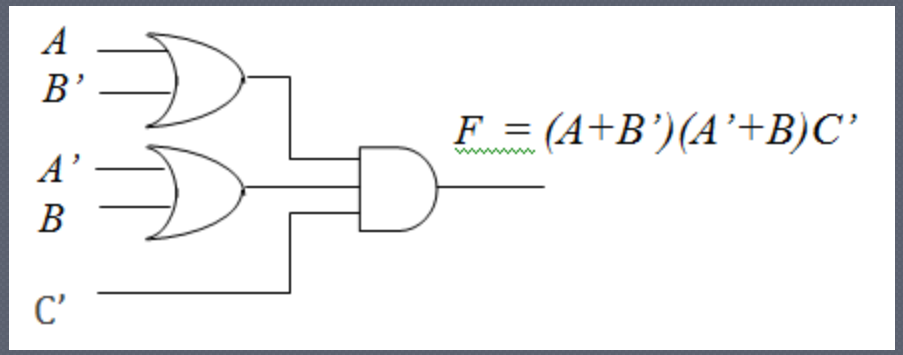


- NAND conversion:

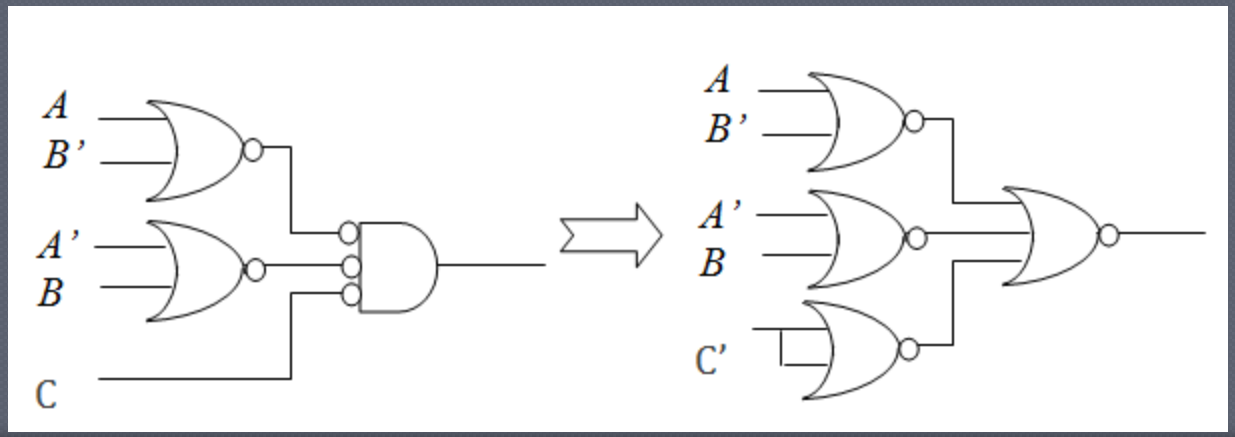


NOR Implementation

- Any **Product of Sum** Boolean expression can be implemented using **NOR gates**.
- E.g. F = (A + B') (A' + B) C'



- NOR conversion:

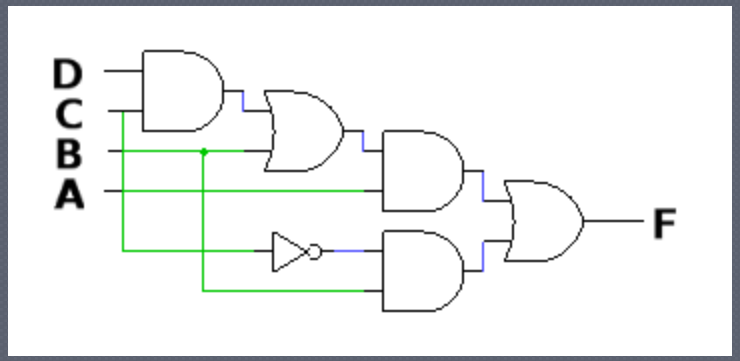


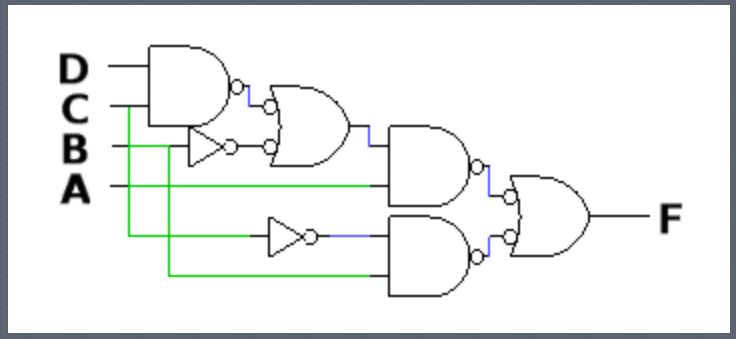
Exercise 1

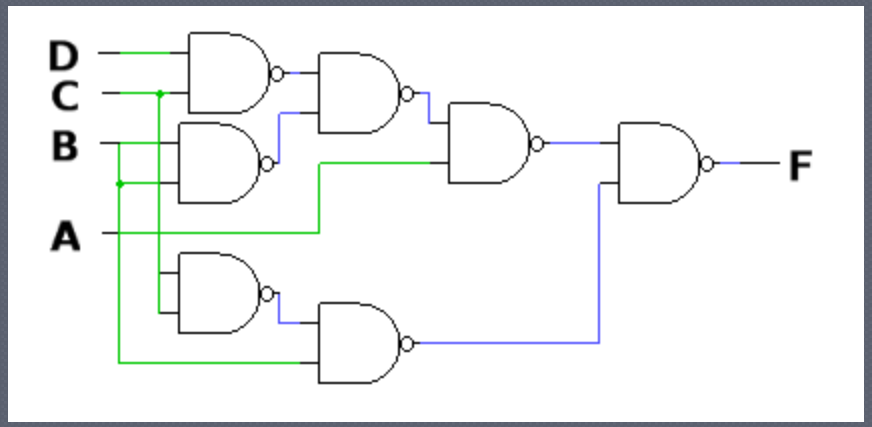
- Implement the following Boolean equation

$$F = A(CD + B)+BC'$$

using all NAND gates.







Wired Logic

- Wired Logic is a form of digital logic in which some logic functions are implemented by directly connecting together the outputs of one or more logic gates.
- Two basic wired logic functions are:
  - **AND – OR – INVERT**

    E.g. $F = (AB + CD + E)'$
  - **OR – AND - INVERT**

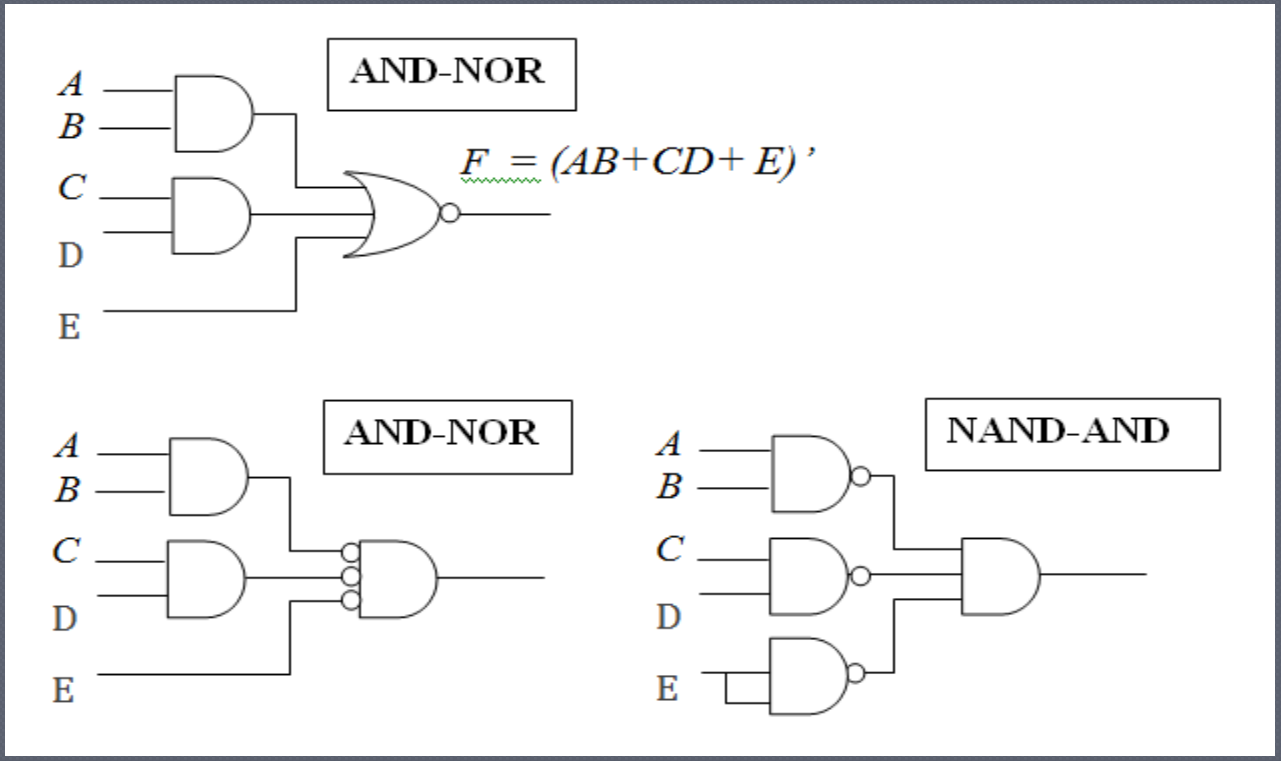
    E.g. $F = [(A+B)\ (C+D)E]'$

AND-OR–INVERT Implementation

- It requires a function in sum of products form.
- It requires 2 logic circuit function:

  (1) AND-NOR or **(2) NAND-AND**

- E.g.

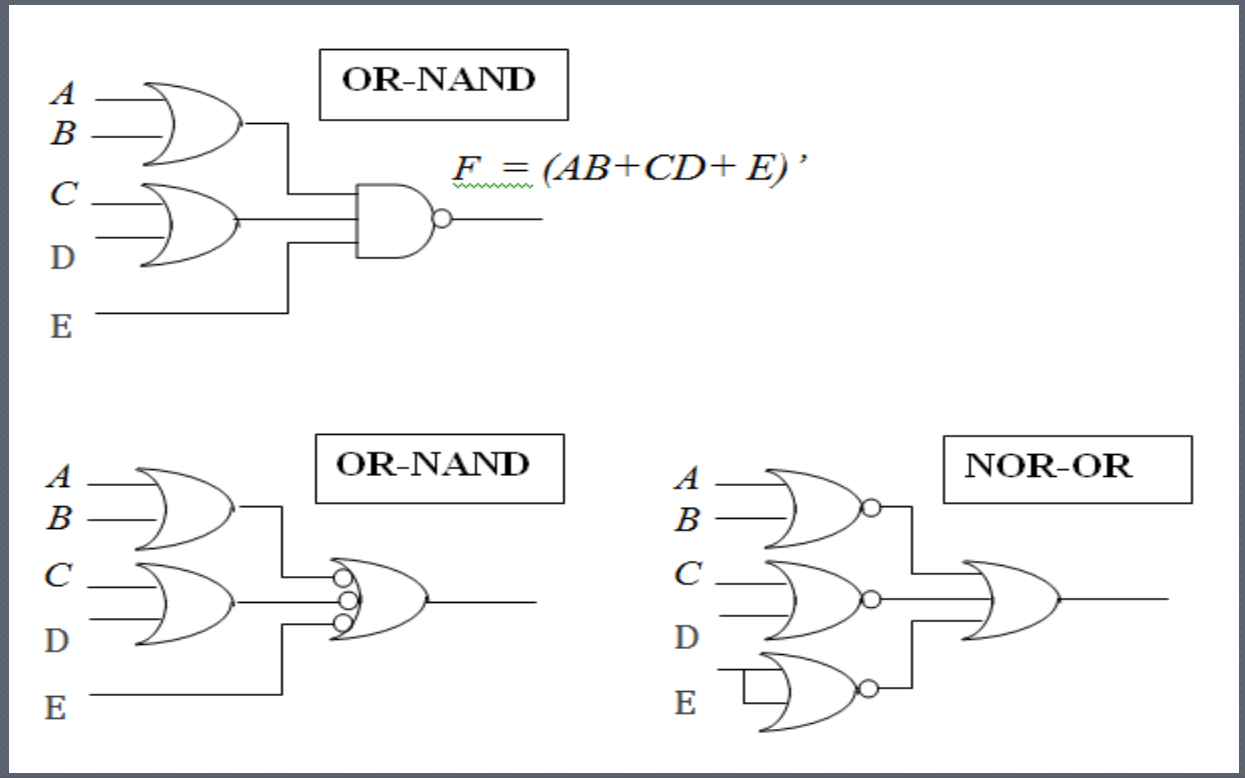


OR–AND–INVERT Implementation

- It requires a function in sum of products form.
- It requires 2 logic circuit function:

  (1) OR-NAND or **(2) NOR-OR**

- E.g.



Digital Logic Design (CSNB163)

End of Module 7