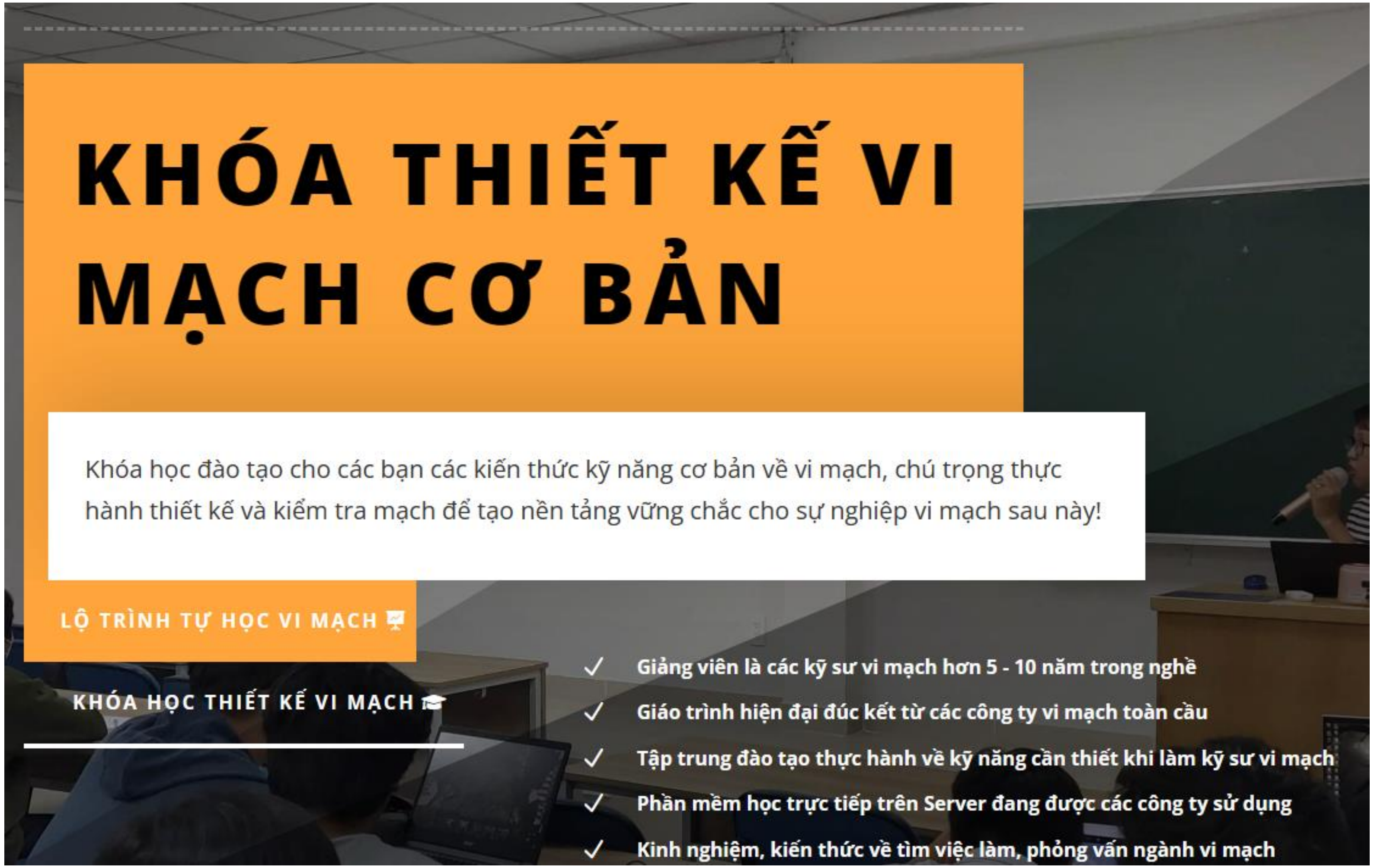
IC OVERVIEW
RTL DESIGN AND VERIFICATION

COURSE INTRODUCTION

[Khóa Học Thiết Kế Vi Mạch Cơ Bản - Trung Tâm Đào Tạo Thiết Kế Vi Mạch ICTC]



COURSE INTRODUCTION

SUMMARY

HOMEWORK

QUESTION

SELF-LEARNING

Session 2: Logic gates

1. Intro
2. Transistor
3. Basic logic gates
4. Boolean algebra
5. De Morgan’s Law

ICTC
IC TRAINING CENTER

LOGIC GATE

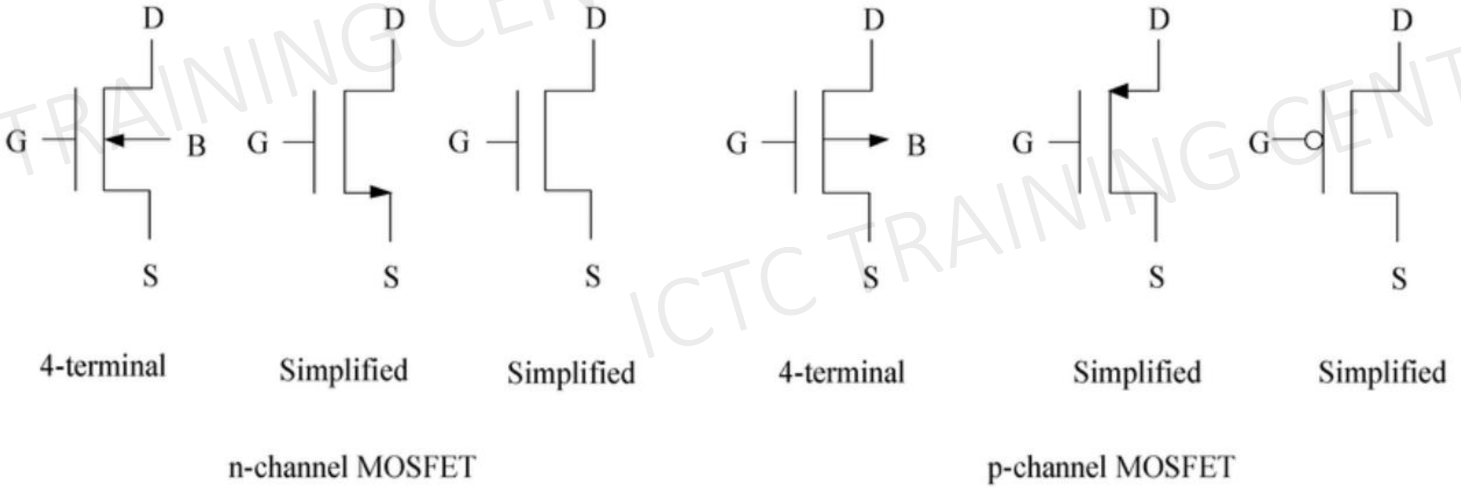
INTRO

- Digital circuit process on two values 0, 1.
- The value 0, 1 represent for Low or High voltage level.
- For example, a voltage supplies to a chip has range from 0 -> 3.3V. Then voltage range 0 ~ 0.5 is logic 0, range 2.9 ~ 3.3 is logic 1.

LOGIC GATE

TRANSISTOR

- Most of semiconductor devices nowadays are made from transistor MOS (metal-oxide-semiconductor).
- There are 2 types of MOS transistor: P type (Positive) and N type (Negative)
- Transistor acts as a switch to turn ON or turn OFF a circuit.



<u>Transistor N-type typical usage</u>

S is connected to GND (Logic 0)

G = 1 => switch is ON => Transistor is conducting

G = 0 => switch is OFF => Transistor is NOT conducting

<u>Transistor P-type typical usage</u>

D is connected to VDD (Logic 1)

G = 1 => switch is OFF => Transistor is NOT conducting

G = 0 => switch is ON => Transistor is conducting

- NOT gate (inverter), is a fundamental digital logic gate that performs the operation of logical negation. The output of a NOT gate is the opposite (complement) of its input. If the input is high (1), the output is low (0), and vice versa.

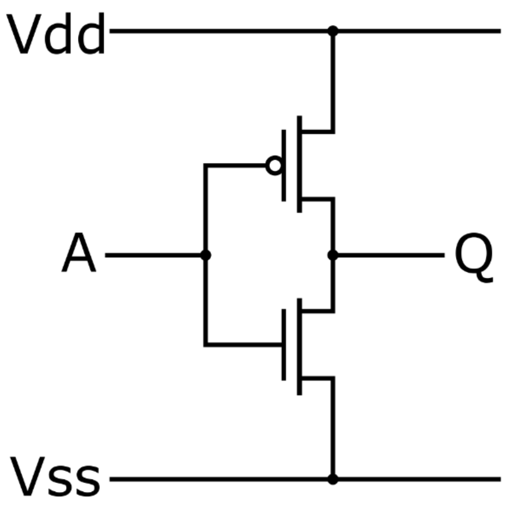
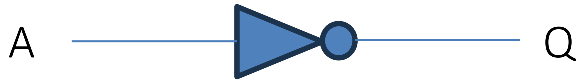


Figure: NOT gate from 2 transistors

__Transistor N-type typical usage__
S is connected to GND (Logic 0)
G = 1 => switch is ON => Transistor is conducting
G = 0 => switch is OFF => Transistor is NOT conducting

__Transistor P-type typical usage__
D is connected to VDD (Logic 1)
G = 1 => switch is OFF => Transistor is NOT conducting
G = 0 => switch is ON => Transistor is conducting

Truth table	
A	Q
0	1
1	0



Verilog code: Q = !A

LOGIC GATE

BASIC LOGIC GATES

- A NOR gate is a digital logic gate that performs the logical NOR (NOT-OR) operation. The output of a NOR gate is high (1) only when both of its inputs are low (0). If at least one input is high, the output is low (0).

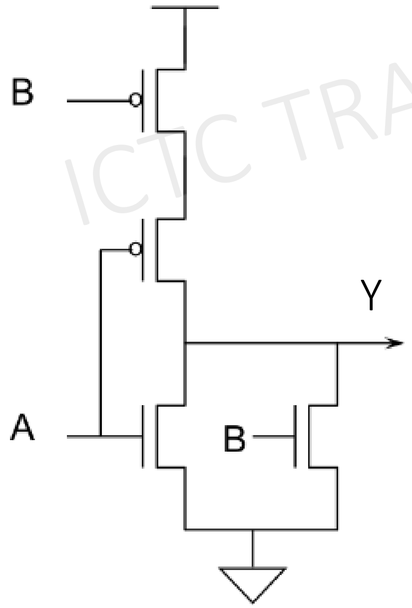
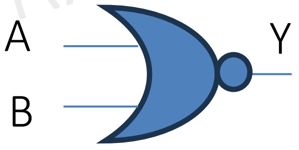


Figure: NOR gate

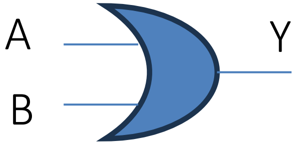
Transistor N-type typical usage
S is connected to GND (Logic 0)
G = 1 => switch is ON => Transistor is conducting
G = 0 => switch is OFF => Transistor is NOT conducting

Transistor P-type typical usage
D is connected to VDD (Logic 1)
G = 1 => switch is OFF => Transistor is NOT conducting
G = 0 => switch is ON => Transistor is conducting

Truth table (NOR)		
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



NOR gate
Verilog code: Y = !(A | B)



OR gate
Verilog code: Y = (A | B)

LOGIC GATE

BASIC LOGIC GATES

- A NAND gate is a digital logic gate that performs the logical NAND (NOT-AND) operation. The output of a NAND gate is low (0) only when both of its inputs are high (1). If at least one input is low, the output is high (1).

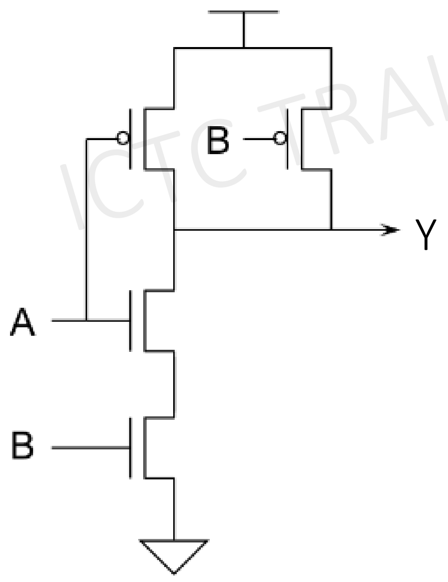


Figure: NAND gate

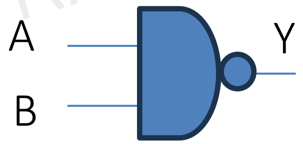
Transistor N-type typical usage

S is connected to GND (Logic 0)
G = 1 => switch is ON => Transistor is conducting
G = 0 => switch is OFF => Transistor is NOT conducting

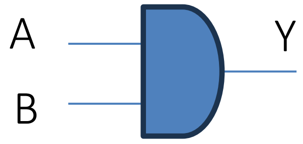
Transistor P-type typical usage

D is connected to VDD (Logic 1)
G = 1 => switch is OFF => Transistor is NOT conducting
G = 0 => switch is ON => Transistor is conducting

Truth table (NAND)		
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



NAND gate
Verilog code: Y = !(A & B)



AND gate
Verilog code: Y = (A & B)

LOGIC GATE

BOOLEAN'S ALGEBRA

- Boolean algebra is widely used in digital electronics, computer science, and information theory for designing and analyzing digital circuits and algorithms.
- In Boolean algebra, the basic operations include AND, OR, and NOT. These operations are applied to binary variables.
- Basic Boolean operations
  1. AND (.): A.B is true (1) only when both A and B are true
  2. OR (+): A+B is true (1) when at least one of A or B is true.
  3. NOT: $\overline{A}$ or A'. It produces the opposite value. If A is true, NOT A is false.

LOGIC GATE

BOOLEAN'S THEOREMS

- Idempotent (bất biến):

  $x + x = x$

  $x . x = x$

- Identity (đồng nhất):

  $x + 0 = x$

  $x . 1 = x$

- Commutative (giao hoán):

  $x + y = y + x$

  $x . y = y . X$

- Associative (kết hợp):

  $(x + y) + z = x + (y + z)$

  $(x . y) . z = x . (y . z)$

- Distributive (phân phối):

  $x . (y + z) = (x . y) + (x . z)$

  $x + (y . z) = (x + y) . (x + z)$

- Complement (bù):

  $x + x' = 1$

  $y . y' = 0$

- Dominative (thống trị):

  $x . 0 = 0$

  $x + 1 = 1$

- Absortive (hấp thu):

  $x + x . y = x$

  $x . (x + y) = x$

LOGIC GATE

BOOLEAN'S THEOREMS

Example: simplified below function

- $x + x'y$

$= (x + x') . (x + y)$

$= 1 . (x + y)$

$= x + y$

- $x . (x' + y)$

$= x.x' + x.y$

$= 0 + x.y$

$= x.y$

- $x.y + x'.z + y.z$

$= x.y + x'.z + y.z.(x+x')$

$= x.y + x'.z + y.z.x + y.z.x'$

$= x.y.(1+z) + x'.z.(1+y)$

$= x.y + x'.z$

Practice: simplify below functions

- x.(x'+y) + y'.(x+y')
- x.y + x.y.z + x'.y + x.y'.z
- x.(y + z.(x.y + x.z))
- (x+y).(x'+z)

- De Morgan's laws are a pair of fundamental rules in Boolean algebra. These laws describe the relationships between logical operations (AND, OR) and their negations (NOT) when applied to logical expressions. There are two De Morgan's laws:

$$\overline{A + B} = \overline{A} \,.\, \overline{B}$$

First Law

$$\overline{A \,.\, B} = \overline{A} + \overline{B}$$

Second Law

Note: "+" is OR and "." is AND

Another expression:

$$A + B = \overline{\overline{A} \,.\, \overline{B}}$$

$$A \,.\, B = \overline{\overline{A} + \overline{B}}$$

All the logic expressions can be represented by NOT, NOR, NAND gates.

LOGIC GATE

DE MORGAN’S LAW

Practice: let’s apply these laws to some logical expressions:

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

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

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

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

LOGIC GATE

BOOLEAN FUNCTION

- A Boolean function is a mathematical operation or expression that takes one or more binary inputs (variables that can have values of 0 or 1) and produces a binary output.
- For example: $F(A,B,C) = A.B + C'$ represents a Boolean function.

LOGIC GATE

BOOLEAN FUNCTION

- Boolean functions can be represented in 2 standard forms is SOP and POS.
- SOP: (Sum of Products) at this form, a Boolean function is expressed as the sum of multiple product terms, which contain all the variables, that makes the Boolean function become 1. In SOP form, A=0 is presented as A', A=1 is presented as A
- POS: (Product of Sums) at this form , a Boolean function is expressed as the product of multiple sum terms, which contain all the variables, that makes the Boolean function become 0. In POS form, A=0 is presented as A, A=1 is presented as A'.

- Example: $F(x,y,z) = x + y + x.z'$

x	y	z	z’	xz’	F(x,y,z)
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1

SOP : Sum of Products

$$F(x,y,z) = x'yz' + x'yz + xy'z' + xy'z + xyz' + xyz$$
$$= m2 + m3 + m4 + m5 + m6 + m7$$
$$= \sum(2,3,4,5,6,7)$$

POS: Product of Sums

$$F(x,y,z) = (x + y + z).(x + y + z')$$
$$= M0 . M1$$
$$= \prod(0,1)$$

LOGIC GATE

SUMMARY

SUMMARY:

- ❑ Transistor acts as a switch to turn ON or turn OFF a circuit.

Transistor N-type typical usage

S is connected to GND (Logic 0)
G = 1 => switch is ON => Transistor is conducting
G = 0 => switch is OFF => Transistor is NOT conducting

Transistor P-type typical usage

D is connected to VDD (Logic 1)
G = 1 => switch is OFF => Transistor is NOT conducting
G = 0 => switch is ON => Transistor is conducting

- ❑ NOT, NOR, NAND are basic gates and can be used to express any other elements.
- ❑ Boolean algebra is used for designing and analyzing digital circuits and algorithms.
- ❑ There are 2 type of Boolean function: SOP & POS.

LOGIC GATE

HOMEWORK

- A 2-input XOR gate, or Exclusive OR gate, is a digital logic gate that performs the exclusive OR operation. The output of an XOR gate is high (1) when the number of true inputs is odd, and it is low (0) when the number of true inputs is even. based on above description, write truth table, MOS diagram and logic for XOR gate.

- *Prove the SOP & POS expressions in above example are equivalent !!!

SOP : Sum of Products

$$F(x,y,z) = x'yz' + x'yz + xy'z' + xy'z + xyz' + xyz$$
$$= m2 + m3 + m4 + m5 + m6 + m7$$
$$= \sum(2,3,4,5,6,7)$$

POS: Product of Sums

$$F(x,y,z) = (x + y + z).(x + y + z')$$
$$= M0 . M1$$
$$= \prod(0,1)$$