Duality

For compound proposition that contains only $\wedge \vee \neg$

Operator	Dual
$\wedge$	$\vee$
$\vee$	$\wedge$
T	F
F	T

Ex :

	Dual
$p \wedge \neg q \wedge \neg r$	$p \vee \neg q \vee \neg r$
$(p \wedge q \wedge r) \vee s$	$(p \vee q \vee r) \wedge s$
$(p \vee F) \wedge (q \vee T)$	$(p \wedge T) \vee (q \wedge F)$

The dual of proposition s is denoted by s^*

$$(s^*)^* \equiv s$$

$$A \Leftrightarrow B \qquad A^* \Leftrightarrow B^*$$

Logical Equivalences laws come in pairs

Disjunctive Normal Form (DNF)

A disjunction of conjunctions where:

1 - every variable or its negation is represented once in each conjunction (a minterm)

2 - each minterms appears only once

Ex:

$p \oplus q$ is equivalent to $(p \wedge \neg q) \vee (\neg p \wedge q)$

$(p \wedge \neg q) \vee (\neg p \wedge q)$ is the DNF for $p \oplus q$

How to obtain a DNF of any propositional expression ?

Method:

First: find the minterms of the DNF:

- Use the rows of the truth table where the proposition is True. Each row with True value makes a minterm in the DNF

For each minterm:

- If False (F) appears under a variable, use the negation of the propositional variable in the minterm
- If True (T) appears, use the propositional variable.

Conjunctive Normal Form (CNF)

A conjunction of disjunctions where:

1 - every variable or its negation is represented once in each disjunction (a minterm)

2 - each minterms appears only once

Ex:

$p \oplus q$ is equivalent to $(p \vee q) \wedge (\neg p \vee \neg q)$

$(p \vee q) \wedge (\neg p \vee \neg q)$ is the CNF for $p \oplus q$

How to obtain a CNF of any propositional expression ?

Method 1:

Obtain the DNF of the negation of the expression and use DeMorgan Law

Method 2:

First: find the minterms of the CNF:

- Use the rows of the truth table where the proposition is False. Each row with False value makes a minterm in the CNF

For each minterm:

- If True (T) appears under a variable, use the negation of the propositional variable in the minterm
- If False (F) appears, use the propositional variable.

A collection of logical operators is called

__Functionally Complete__ if every compound proposition is __logically equivalent__ to a compound proposition involving only these logical operators.

Ex: $\neg, \wedge, \vee$ form a functionally complete collection of logical operator

How: Convert any proposition into DNF or CNF

Ex: $\neg, \wedge$ form a functionally complete collection of L.O.

How: 1- Convert any proposition into CNF
2- Convert any $(P_1 \vee P_2 \ldots \vee P_n)$

into $\neg(\neg P_1 \wedge \neg P_2 \ldots \wedge \neg P_n)$

Ex: $\neg, \vee$ for a functionally complete collection

LOGIC AND BIT OPERATIONS

A <u>bit</u> has two possible values $\begin{cases} 0 & \text{false} \\ 1 & \text{true} \end{cases}$

<u>binary digit</u>

<u>Logic</u>	<u>Boolean Algebra</u>	
proposition	Boolean variable	
True	1	
False	0	
$\wedge$	AND	$\cdot$
$\vee$	OR	$+$
$\oplus$	XOR	
$\neg x$	$\bar{x}$	complement

Boolean Algebra

$1 + 1 = 1$

$1 + 0 = 1$

$0 + 1 = 1$

$0 + 0 = 0$

$1 \cdot 1 = 1$

$1 \cdot 0 = 0$

$0 \cdot 1 = 0$

$0 \cdot 0 = 0$

$\overline{0} = 1$

$\overline{1} = 0$

$\overline{\overline{x}} = x$ Law of double complement

$x + x = x$

$x \cdot x = x$

Idempotent

$x + 0 = x$

$x \cdot 1 = x$

Identity

$x + 1 = 1$

$x \cdot 0 = 0$

Dominance

$x + y = y + x$

$x \cdot y = y \cdot x$

Commutative

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

$x \cdot (y \cdot z) = (x \cdot y) \cdot z$

Associative

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

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

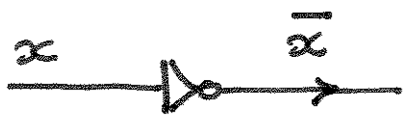
Distributive

$\overline{(xy)} = \overline{x} + \overline{y}$

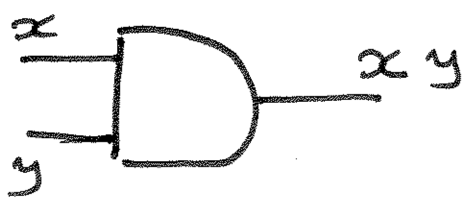
$\overline{(x + y)} = \overline{x}\ \overline{y}$

De Morgan law

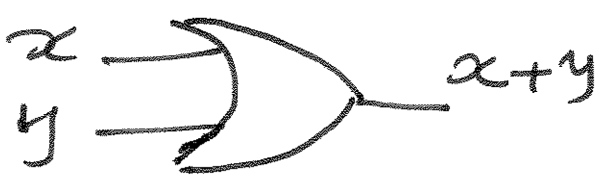
Logic Gates



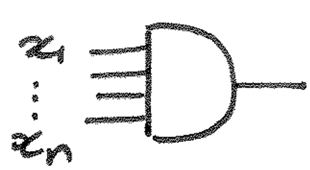
NOT

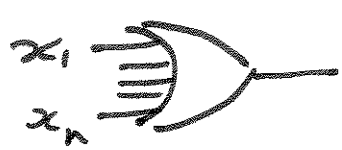


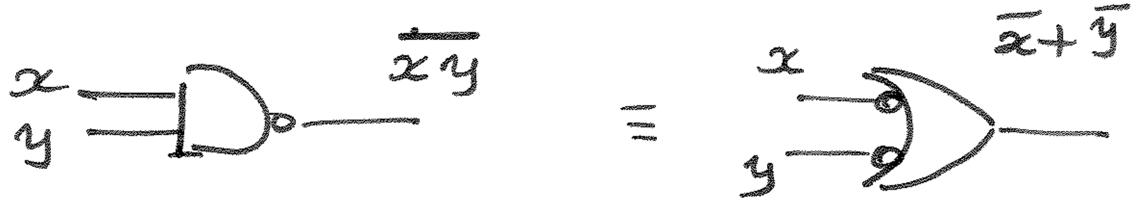
AND



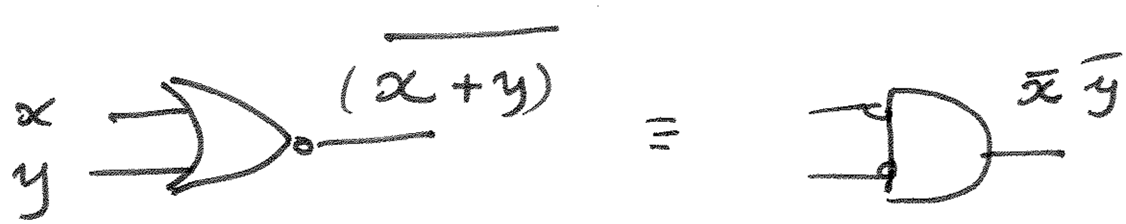
OR







NAND



NOR

Bit operations

bit string: sequence of bits

bit wise AND
bit wise OR
bit wise XOR