Digital Electronics (EE221)

Lecture -2

Boolean Operations

- Single variable operation
  - NOT
- Multiple variable operation
  - → Two input single output
    - → AND
    - → OR

NOT, AND, OR → Fundamental to Boolean Algebra.

Variable and Signals

Variable ┬ Binary variable
├→ '0' or '1'
└→ High or Low

Variable → Mathematical/Logical representation of an electrical signal

Example –
0 – 2.5 → Logical '0' / Low
2.5 – 5.0 → logical '1' / high

Variables and Signals

Variables are represented in letters — A, B, C

Variables can take only two values — '0' or '1'.

Any operation on variable(s) will finally lead to '0' or '1'.

We will mostly talk about variables.

NOTE: Variable state is measured through electrical voltage and current values

NOT Operation

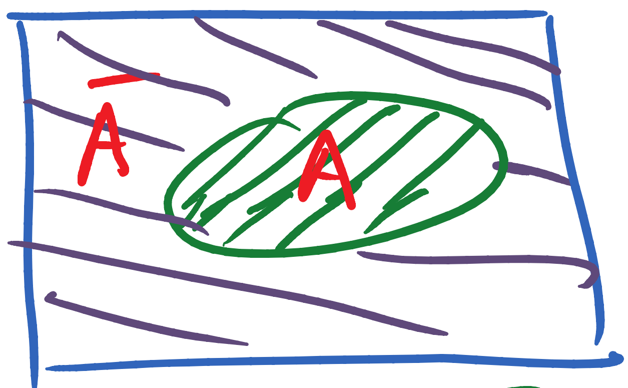
Representation –

$$NOT(A) \Longleftrightarrow \bar{A}$$

Logic symbol –

(NOT Gate)

VENN Diagram



Implemented through inverter circuit –

$A = 0, \ \bar{A} = 1$

$A = 1, \ \bar{A} = 0$

NOT $\Longleftrightarrow$ Inversion

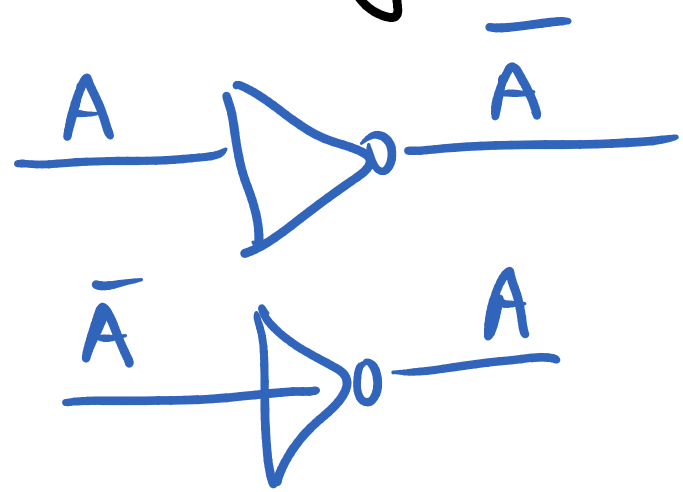
TRUTH TABLE

{All possible Inputs

A	B	C	D	...	output 1	output 2	...
0	0	0	0	...	1	1	...
0	0	0	1	...	0	1	...

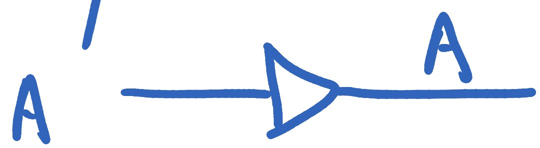
TRUTH TABLE for NOT gate —

A	$\bar{A}$
0	1
1	0



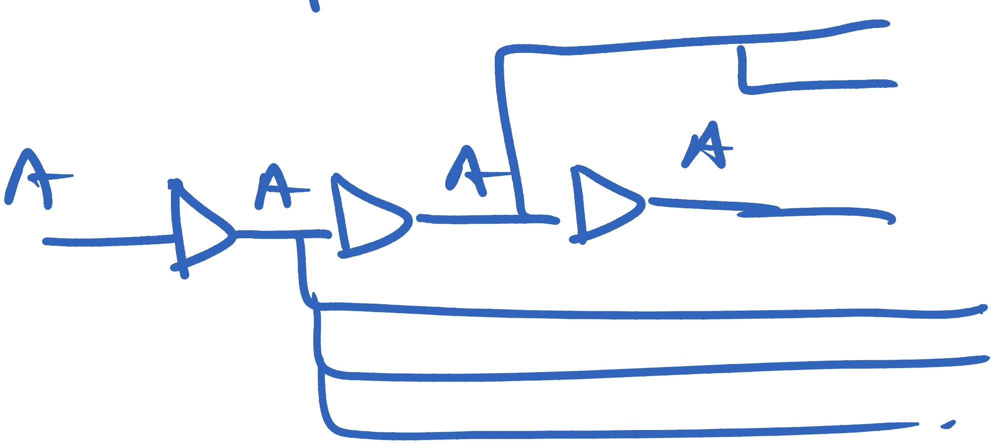
Buffer Gate

* Symbol –



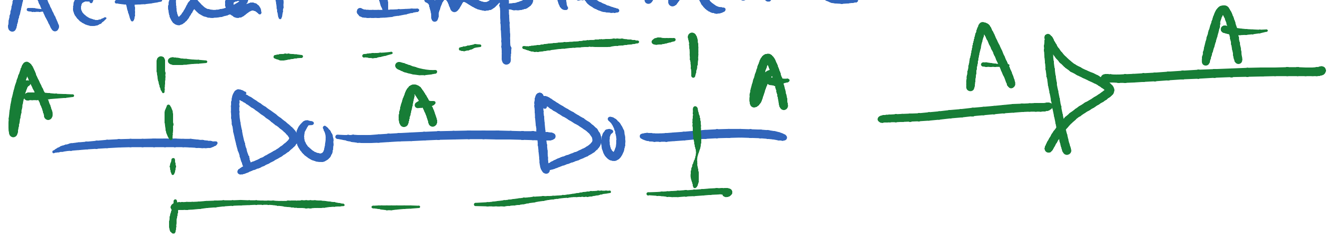
* Will be discussed later

* Improves signal level strength
* Improves noise immunity



Larger fan out from the same signal

* Actual Implementation

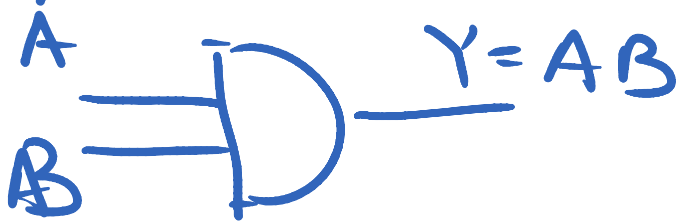


AND Operation

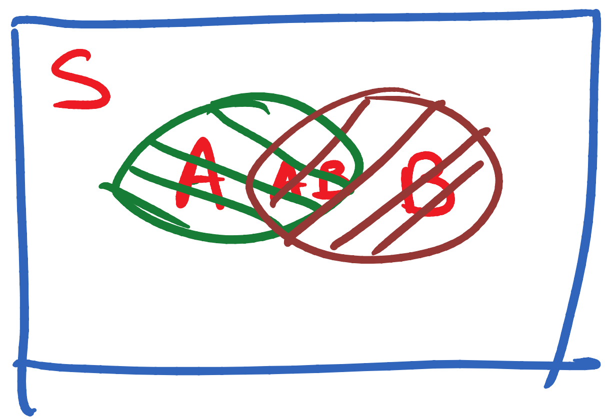
Representation —

$$AND(AB) = A.B = AB$$

Symbol —



VENN Diagram —



Truth table —

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

$$Y = AB$$

$$\bar{Y} = \bar{A}\,\bar{B} + \bar{A}B + A\bar{B}$$

$$Y = \bar{\bar{Y}} = \overline{\bar{A}\,\bar{B} + \bar{A}B + A\bar{B}}$$

Multiple Input AND operation.

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

2^N Number of combination for N variable input.

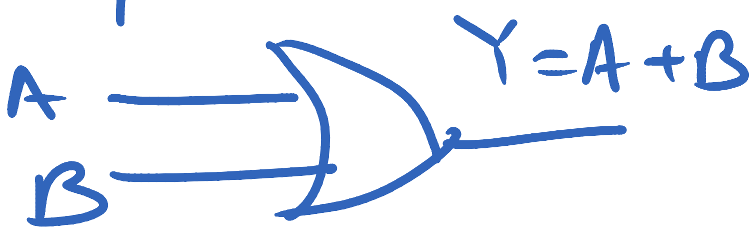
$Y = ABC$

OR Gate

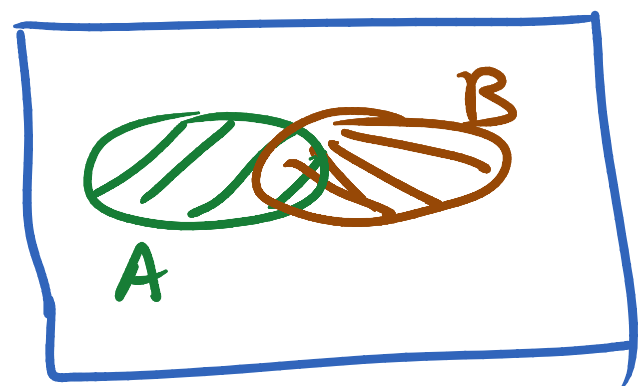
- Representation

$$OR(AB) = A + B$$

- Symbol



Venn Diagram



- Truth Table

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

$$Y = A + B$$

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

Identity Operation

By Definition —

$A + \bar{A} = 1$, $A + 0 = A$

$A + 1 = 1$

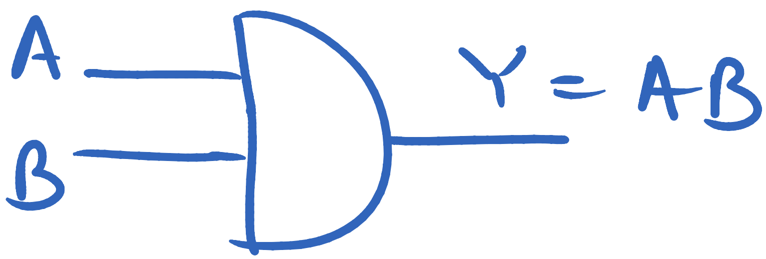
A	$\bar{A}$	Y
0	1	1
1	0	1

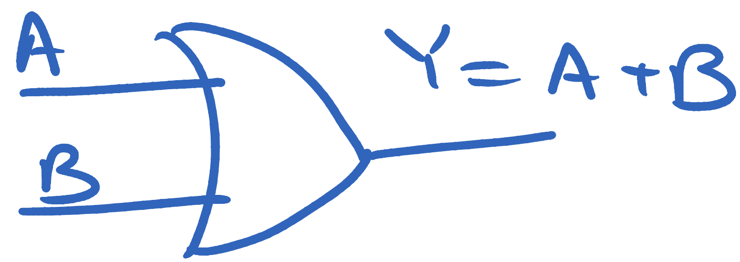
$A\bar{A} = 0$, $A \cdot 0 = 0$

$A + 0 = A$

A	$\bar{A}$	Y
0	1	0
1	0	0

AND and OR





$$\overline{Y} = \overline{AB}$$

$$\boxed{\overline{Y} = \overline{A}\,\overline{B} + A\overline{B} + \overline{A}B}$$

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

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

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

Any operation/function can be implemented in its regular form or dual form

Postulates for Algebraic Structure

1) Closure –

A set S is closed w.r.t a binary operation if for every pair of elements in S, the binary operation specifies a rule for obtaining a unique element is S.

2) Associative Law –

$(x * y) * z = x * (y * z), x, y, z \in S$

3) Commutative Law -

$$x * y = y * x \quad \forall x, y \in S$$

4) Identity element -

$$e * x = x * e = x, \quad \forall x \in S$$

5) Inverse -

$$x * y = e, \quad x, y \in S$$

6) Distributive Law -

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

Huntington Postulates (1904)

George Boole (1854) - Boolean Algebra

1) Closure w.r.t the operation +
closure w.r.t the operator .

2) Identity element w.r.t +,
$0 + x = x + 0 = x$
Identity element w.r.t to '.'
$1 . x = x . 1 = x$

3) Commutative, $x + y = y + x$
$x . y = y . x$

4) '.' distributive over '+' $x.(y+z) = x.y + x.z$
'+' distributive over '.' $x + (y.z) = (x+y).(x+z)$

5) Associative –

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

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

6) Inverse,

$$x + \bar{x} = 1$$

$$x \cdot \bar{x} = 0$$

Two valued Boolean algebra having a set of two elements 1 and 0, two binary operators with operation rules AND and OR, and a complement operator NOT.

Dulality Principle

- Huntington postulates have been listed pairs
- One part is obtained if the binary operator and identity elements are interchanged

Basic Theorems

P	$x + 0 = x$	$x \cdot 1 = x$
P	$x + \bar{x} = 1$	$x \cdot \bar{x} = 0$
T	$x + x = x$	$x \cdot x = x$
T	$x + 1 = 1$	$x \cdot 0 = x$
T	$\overline{(\bar{x})} = x$	
P	$x + y = y + x$	$x \cdot y = y \cdot x$
P	$(x+y)+z = (x+y)+z$	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
P	$x \cdot (y+z) = x \cdot y + x \cdot z$	$x + y \cdot z = (x+y) \cdot (x+z)$
T	$\overline{(x+y)} = \bar{x}\,\bar{y}$	$\overline{x \cdot y} = \bar{x} + \bar{y}$
T	$x + xy = x$	$x \cdot (x+y) = x$

Proof of the Theorems

$$x + x = (x + x) \cdot 1$$
$$= (x + x) \cdot (x + \bar{x})$$
$$= x + x \cdot \bar{x}$$
$$= x + 0$$
$$= x$$

$x \cdot x = x$ [By duality]

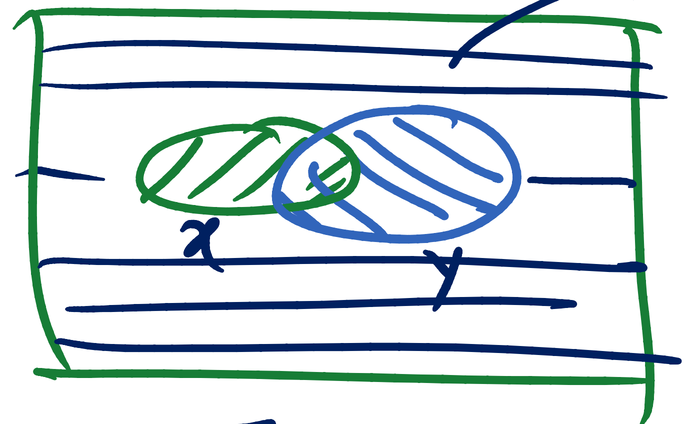
Proof of the Theorems

$$x+1 = 1\cdot(x+1)$$
$$= (x+\overline{x})(x+1)$$
$$= x+\overline{x}\cdot 1$$
$$= x+\overline{x}$$
$$= 1$$

$x.0 = 0$ [By duality]

De Morgan's Law -

$$\overline{x+y} = \overline{x} \cdot \overline{y}$$



$$\overline{x+y} = \overline{x} \cdot \overline{y}$$

x	y	$x+y$	$\overline{x+y}$	$\overline{x}$	$\overline{y}$	$\overline{x}\,\overline{y}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	1	1	0	0	0	0
1	0	1	0	0	1	0

$\overline{x \cdot y} = \overline{x} + \overline{y}$ [By Duality]

Boolean Functions

* An expression formed with binary variables

$$F_1 = x\bar{y} + \bar{x}yz + \bar{y}z$$

$$F_2 = xy + yz + zx$$

$$F_3 = ABCD + A\bar{B} + C\bar{D} + A\bar{C}$$

$$\vdots$$

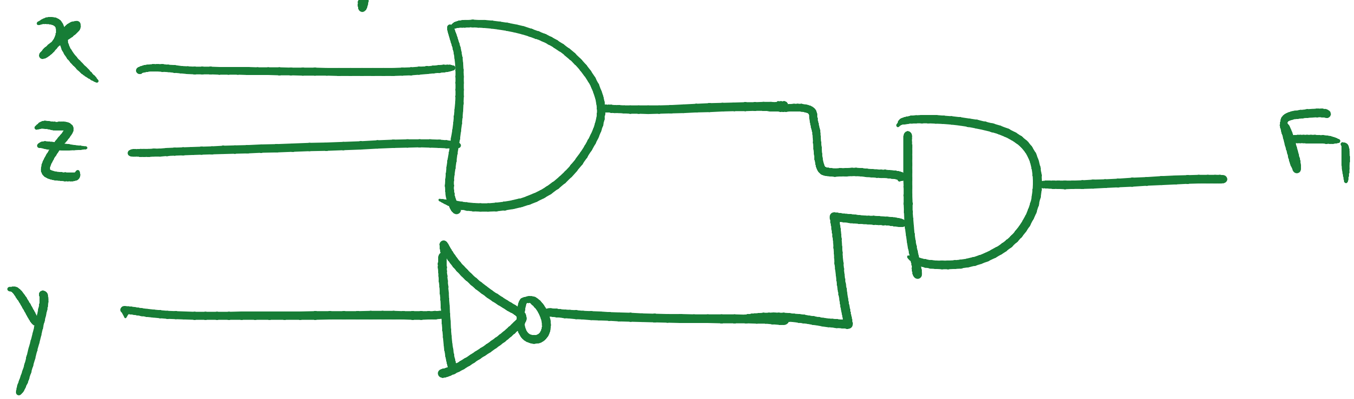
Truth Table

Any Boolean function can be represented in truth table.

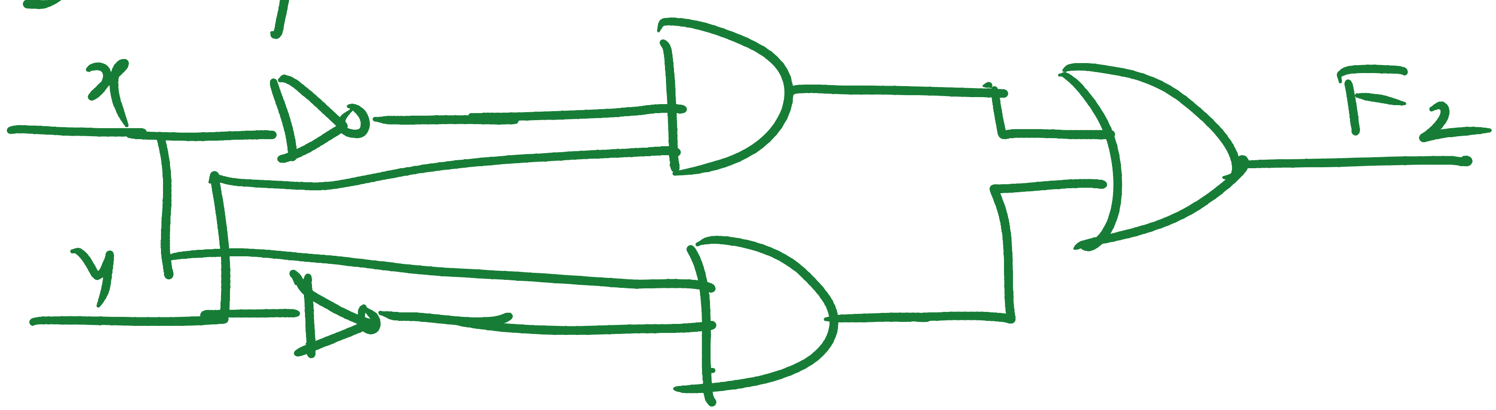
x y z	F_1	F_2	F_3
All Possible combinations	values of the functions (0,1)		

Implementation of Boolean function using Logic gates —

$F_1 = x\bar{y}z$



$F_2 = x\bar{y} + \bar{x}y$



Minimization of Boolean function

- Minimize the Boolean function for less costly hardware implementation
- Simplified implementation

$$F = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}$$

$$= \bar{x}z(\bar{y}+y) + x\bar{y}$$

$$= \bar{x}z + x\bar{y}$$

[Trial approach]

Complement of a Function —

$$\overline{A+B+C} = \bar{A}\,\bar{B}\,\bar{C}$$

$$\overline{A\cdot B\cdot C} = \bar{A}+\bar{B}+\bar{C}$$

(i) Take the dual of the function

(ii) Complement each literal

$$F_1 = \bar{x}\,y\,\bar{z} + \bar{x}\,\bar{y}\,z$$

$$\text{Dual}(F_1) = (\bar{x}+y+\bar{z})\cdot(\bar{x}+\bar{y}+z)$$

$$\bar{F_1} = (x+\bar{y}+z)\cdot(x+y+\bar{z})$$

Dual of a function

$$F_2 = x(\bar{y}\bar{z} + yz)$$

$$\text{Dual}(F_2) = x + (\bar{y} + \bar{z}) \cdot (y + z)$$

$$\bar{F_2} = \bar{x} + (y + z)(\bar{y} + \bar{z})$$

Any function can be implemented in its regular form or its complement form

Canonical and Standard Forms —

x	y	z	Min Term	Designation
0	0	0	$\bar{x}\,\bar{y}\,\bar{z}$	m_0
0	0	1	$\bar{x}\,\bar{y}\,z$	m_1
0	1	0	$\bar{x}\,y\,\bar{z}$	m_2
0	1	1	$\bar{x}\,y\,z$	m_3
1	0	0	$x\,\bar{y}\,\bar{z}$	m_4
1	0	1	$x\,\bar{y}\,z$	m_5
1	1	0	$x\,y\,\bar{z}$	m_6
1	1	1	$x\,y\,z$	m_7

Any Boolean function can be represented as sum of minterms.

Canonical and Standard Forms

$$F_1 = A + \bar{B} C$$

$$= A(B+\bar{B}) + (A+\bar{A})\bar{B} C$$

$$= AB(C+\bar{C}) + A\bar{B}(C+\bar{C}) + A\bar{B}C + \bar{A}\bar{B}C$$

$$= ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + \bar{A}\bar{B}C$$

$$= m_7 + m_6 + m_5 + m_4 + m_1$$

Canonical and Standard Forms

x	y	z	Maxterm	Designation
0	0	0	$x+y+z$	M_0
0	0	1	$x+y+\bar{z}$	M_1
0	1	0	$x+\bar{y}+z$	M_2
0	1	1	$x+\bar{y}+\bar{z}$	M_3
1	0	0	$\bar{x}+y+z$	M_4
1	0	1	$\bar{x}+y+\bar{z}$	M_5
1	1	0	$\bar{x}+\bar{y}+z$	M_6
1	1	1	$\bar{x}+\bar{y}+\bar{z}$	M_7

Implementation through product of Maxterms.

Canonical and Standard Forms

$$F = xy + \bar{x}z$$

$$= (xy + \bar{x})(xy + z)$$

$$= (x + \bar{x})(y + \bar{x}) \cdot (x + z)(y + z)$$

$$= (\bar{x} + y)(x + z)(y + z)$$

$$= (\bar{x} + y + z\bar{z})(x + z + y\bar{y})(y + z + x\bar{x})$$

$$= (x + y + z)(x + \bar{y} + z)(\bar{x} + y + z)(\bar{x} + y + \bar{z})$$

$$= M_0 M_2 M_4 M_5$$

Canonical Forms —

* Any function can be implemented either as sum of minterms or product of maxterms

$$F(A, B, C) = \sum(1, 2, 7\ 8)$$

$$F(x, y, z) = \prod(2, 6, 7)$$

Conversion Between Canonical Forms –

$$\overline{m_j} = M_j$$

$$F = \sum(0,2,4,5)$$

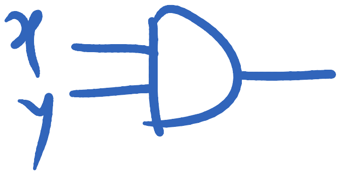
$$\Rightarrow \overline{F} = \sum(1,3,6,7)$$

$$\Rightarrow F = \prod(1,3,6,7)$$

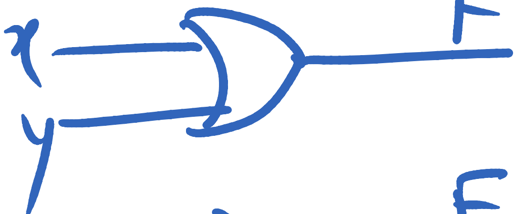
Sum of Minterms → AND-OR Implementation

Product of Maxterms → OR-AND Implementation

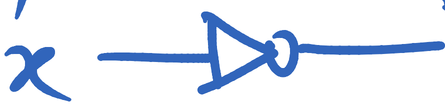
Basic Logic Family —



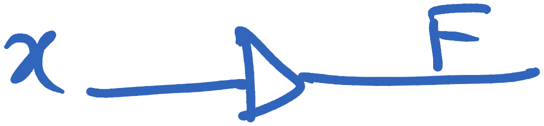
$F = xy$ AND



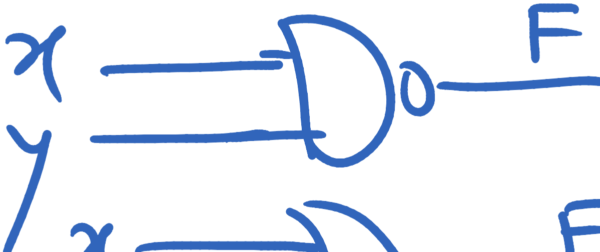
$F = x + y$, OR



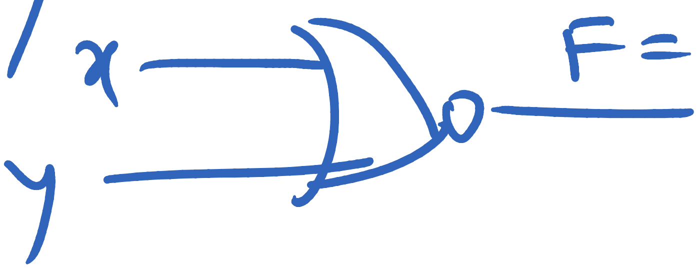
$F = \bar{x}$, Inverter



$F = x$, Buffer

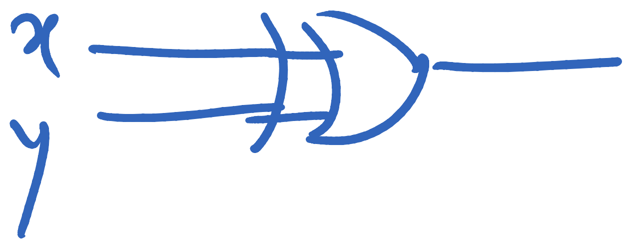


$F = \overline{xy}$, NAND



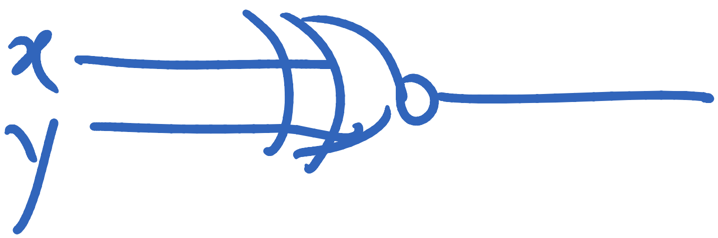
$F = \overline{x+y}$, NOR

Basic Logic Families —



$F = x\bar{y} + \bar{x}y$ Exclusive OR

$= x \oplus y$



$F = xy + \bar{x}\bar{y}$ Exclusive NOR

$= x \odot y$

- NAND and NOR gates are called universal gates
- Minimum number of such gates are to be used for implementation

Reference —

Digital Logic and Computer Design – M. Morris Mano,
PHI India
ISBN - 81 - 203 - 0417 - 9