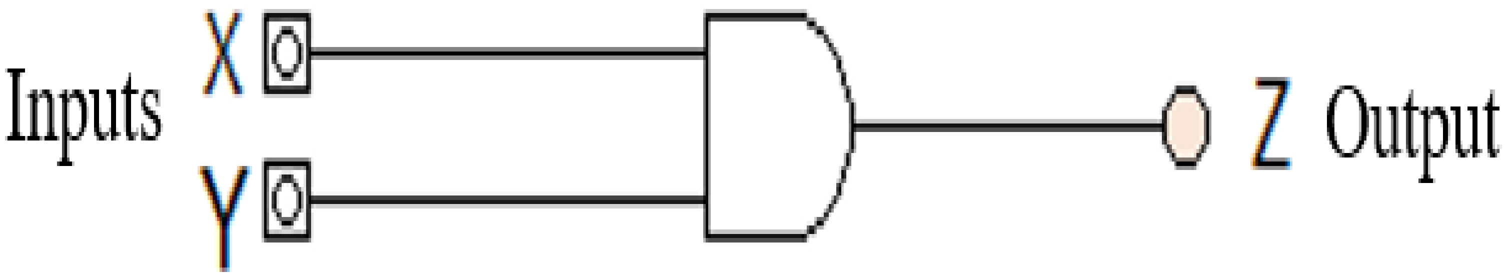
Unit 2

Logic gates: Logic gates are the basic building blocks of digital computer. Logic gates have one or more than one input and only one output. Input and output values are the logical values true (1) and false (0). Logic gates are also called combinational logic circuits. Basic logic gates are AND, OR and NOT.

1. **AND gate:** The output of AND gate is 1 if and only if all of the inputs are 1, otherwise the output value is 0.

Logic diagram of AND gate:



Algebraic Function:

Z = X AND Y

= X.Y

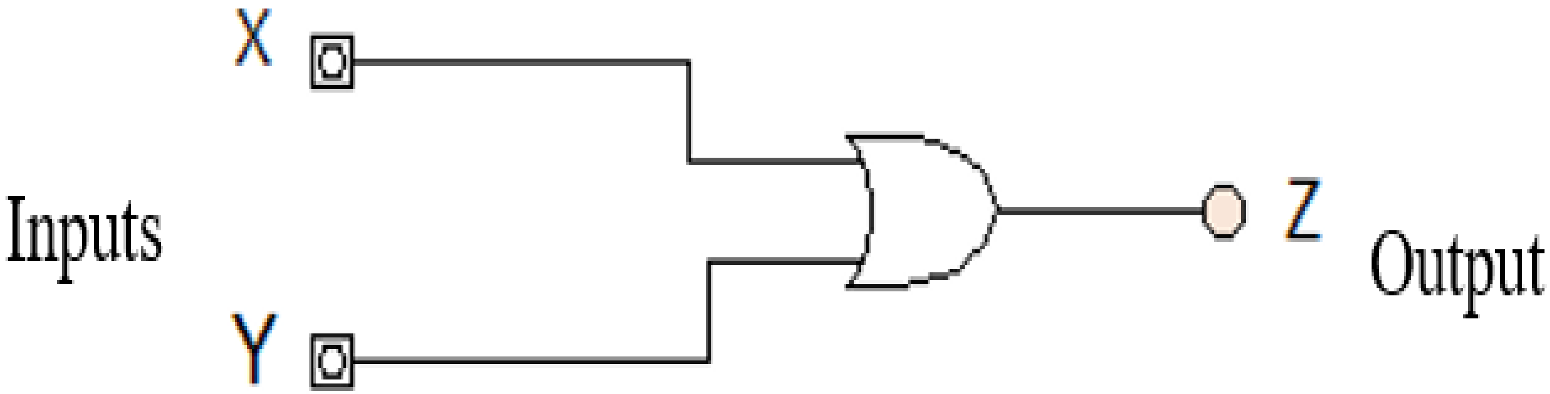
Truth Table:

2^n combination of variables. i.e. $2^2 = 4$ binary Combinations from 0 to 3.

Inputs		Outputs
X	**Y**	**Z**
0	0	0
0	1	0
1	0	0
1	1	1

2. **OR gate:** The output of OR gate is 1 if any one of the input value is 1 and output is 0 if all the inputs are 0.

Logic diagram of OR gate:



Algebraic Function:

$Z = X \text{ OR } Y$ or $X + Y$

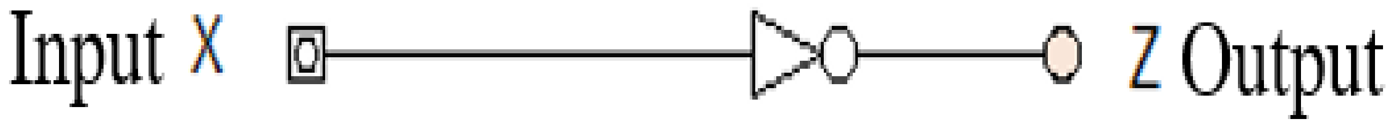
Truth Table:

2^n combination of variables. i.e. $2^2 = 4$ binary Combinations from 0 to 3.

Inputs		Outputs
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

3. **NOT gate (inverter):** A NOT has only one input and one output the output of NOT gate is the inversion of the input i.e. complement of the input.

Logic diagram of NOT gate:



Algebraic Function:

$$Z = \text{NOT } X$$
$$= \bar{X}$$

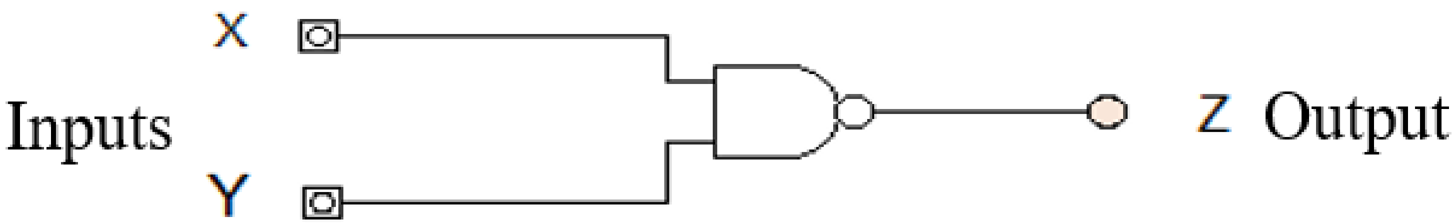
Truth Table:

Input	Output
X	Z
0	1
1	0

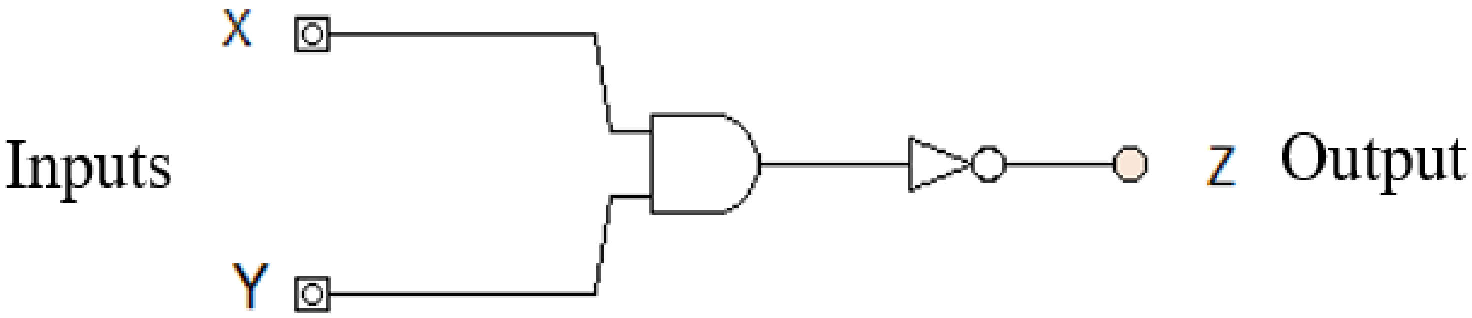
Combined Gates:

NAND gate: NAND gate is the combination of AND and NOT gates. An AND gate with inverter at the output. The output of NAND gate is the complement of AND gate.

Logic Diagram of NAND gate:



Or



Algebraic Function:

$$Z = \overline{X \text{ AND } Y}$$
$$= (X.Y)'$$

Activate \
Go to Settin

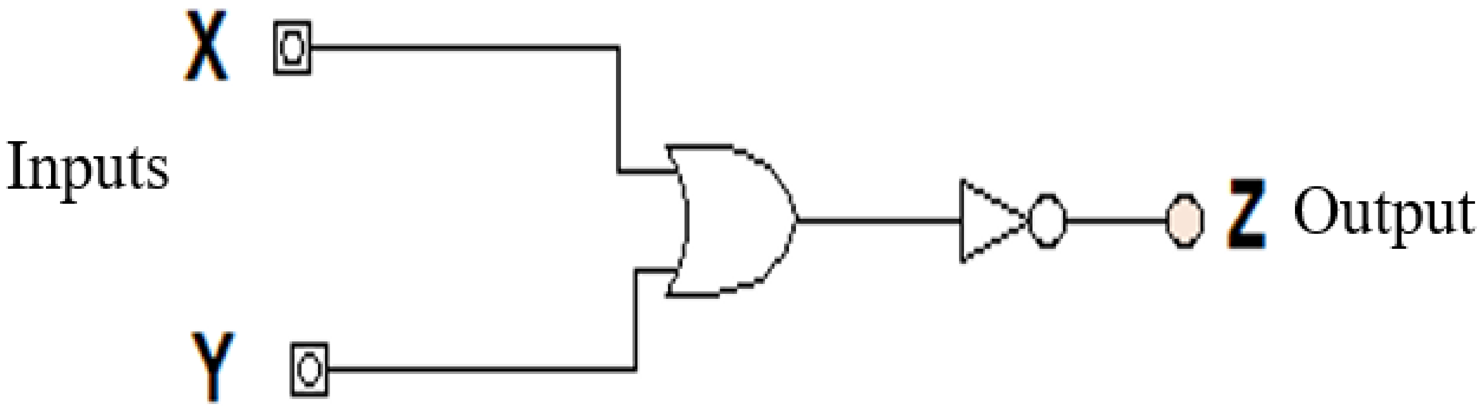
Truth Table:

2^n combination of variables. i.e. $2^2 = 4$ binary Combinations from 0 to 3.

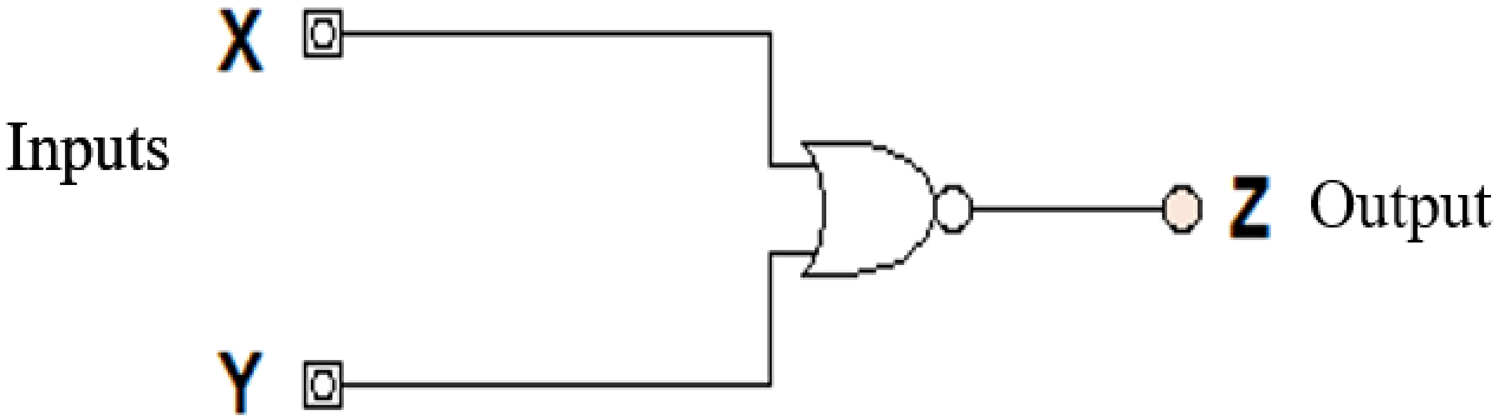
Inputs		Outputs
X	**Y**	**Z**
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate: NOR gate is the combination of OR and NOT gate. The output of NOR gate is the complement of OR gate.

Logic Diagram of NAND gate:



Or



Activate W
Go to Settings

Algebraic Function:

$$Z = \overline{X \text{ OR } Y}$$
$$= (X+Y)'$$

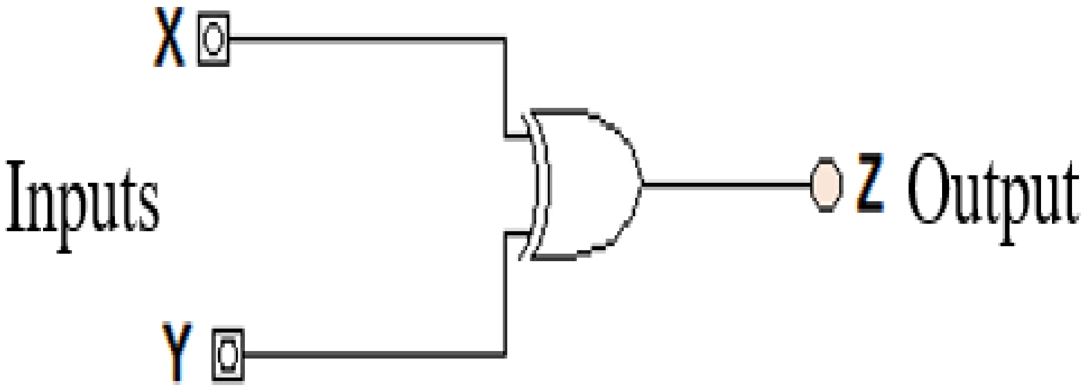
Truth Table:

2^n combination of variables. i.e. $2^2 = 4$ binary Combinations from 0 to 3.

Inputs		**Outputs**
X	**Y**	**Z**
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive – OR (XOR) Gate: An XOR gate gives an output value 1 when there are different input values and the output value is low when there are same input values.

Logic Diagram of XOR gate:



Algebraic Function:

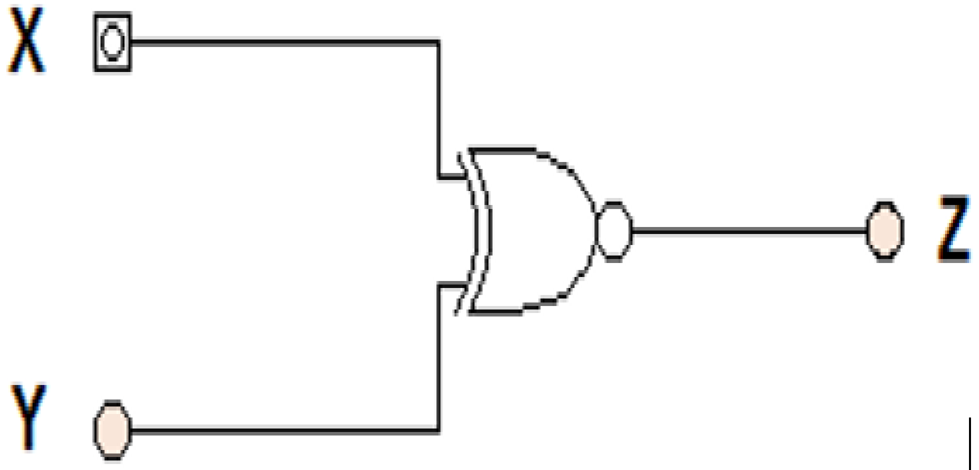
$$Z = X\overline{Y} + \overline{X}Y$$

Truth Table:

Inputs		Outputs
X	**Y**	**Z**
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive – NOR (X-NOR) Gate: An X-NOR gate gives an output value 1 when there are same input values and the output value is low when there are different input values.

Logic Diagram of XNOR gate:



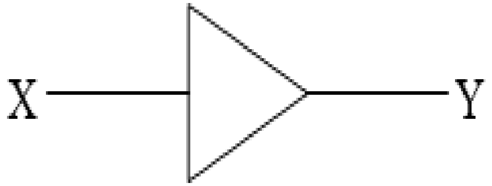
Algebraic Function:

$Z = XY + \bar{X}\bar{Y}$

Truth Table:

Inputs		Outputs
X	**Y**	**Z**
0	0	1
0	1	0
1	0	0
1	1	1

Buffer: Buffer produces the transfer function but does not produce any particular logic operation, since the binary value of the output is equal to the binary value of the input. Buffer will delay the time between input and output. It can also amplify the signal if the current is too weak.



Logic function: Y = X

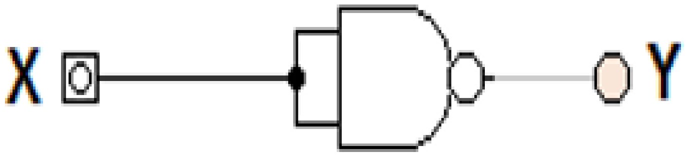
Truth Table:

Input	Output
X	Y
0	0
1	1

Universal Gates: NAND and NOR gates are called universal gates because, we can build any gate using NAND or NOR gates. Basic logic gates AND, OR and NOT can be realized by using only NAND or NOR gates.

NAND as a universal gate:

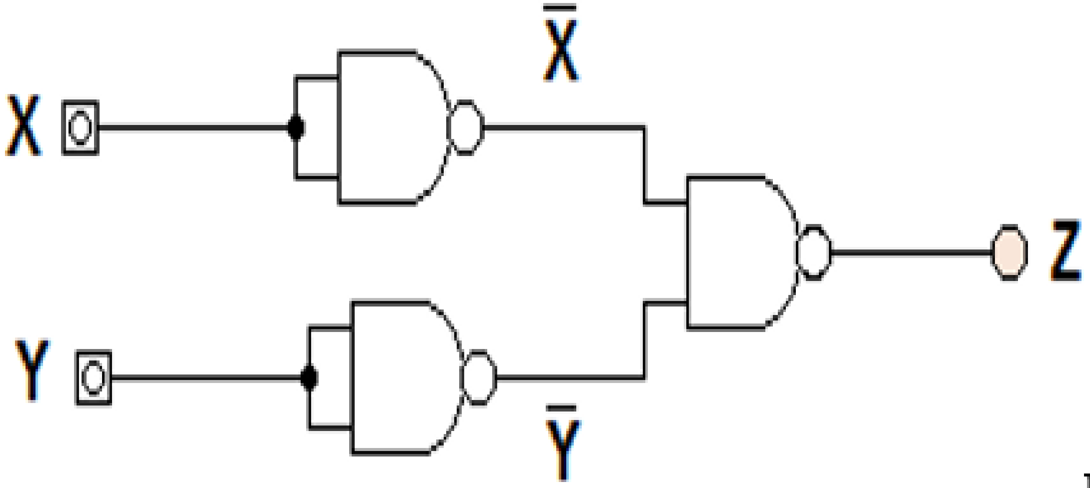
i) **NOT gate:** NOT gate can be realized using single input NAND gate.



$= \overline{X}$

Properties of NOT gate.

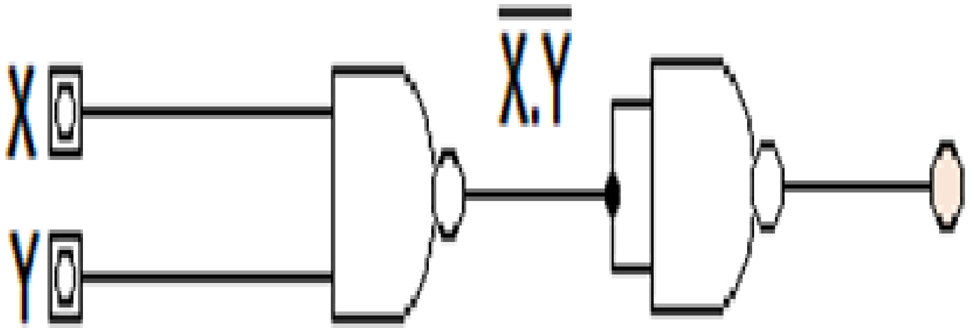
ii) **OR gate:** OR gate can be realized using three NAND gates.



$= \overline{\overline{X}.\overline{Y}} = \overline{\overline{X}} + \overline{\overline{Y}} = X + Y$

Properties of OR gate.

iii) AND gate: AND gate can be realized using two NAND gates.

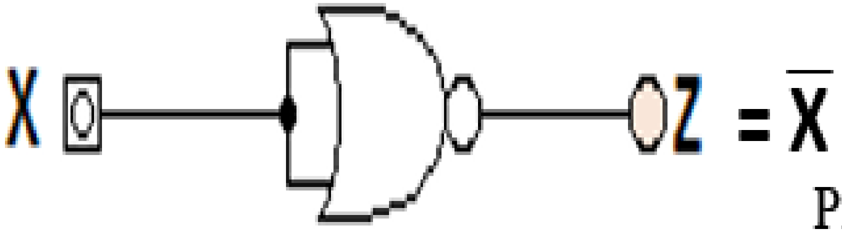


$$Z = \overline{\overline{X.Y}} = \overline{\overline{X} + \overline{Y}} = \overline{\overline{X}} \,.\, \overline{\overline{Y}} = X \,.\, Y$$

Properties of AND gate.

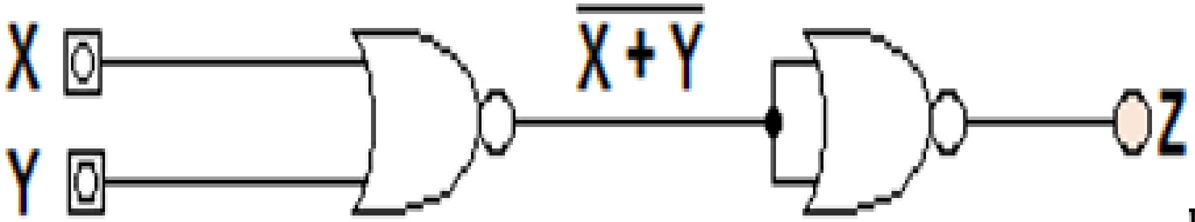
NOR as a universal gate:

i) **NOT gate:** NOT gate can be realized using a single input NOR gate.



Properties of NOT gate.

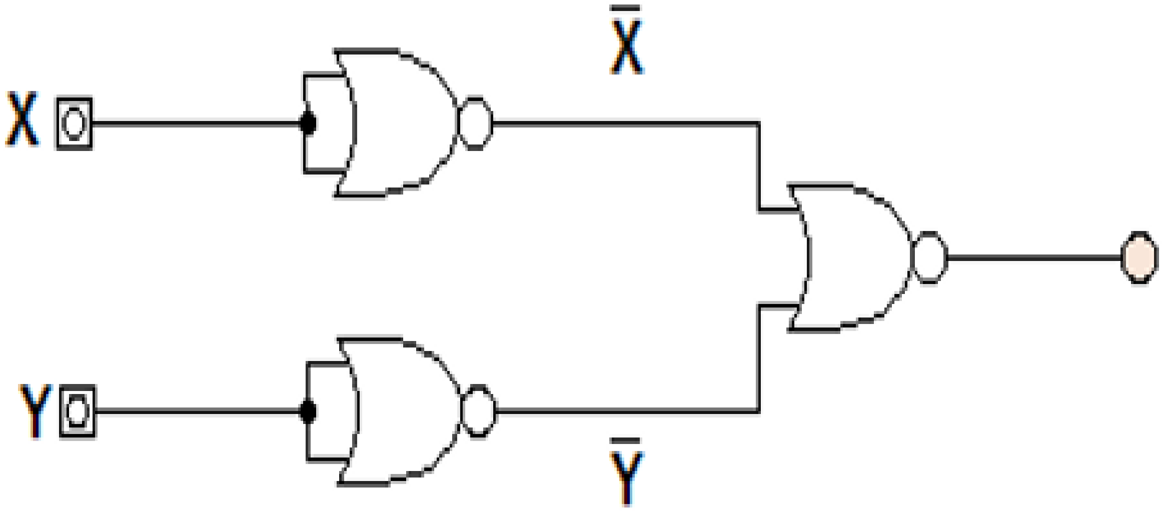
ii) **OR gate:** OR gate can be realized using two NOR gates.



$$= \overline{\overline{X+Y}} = \overline{\overline{X}\,.\,\overline{Y}} = \overline{\overline{X}} + \overline{\overline{Y}} = X + Y$$

Properties of OR gate.

iii) AND gate: AND gate can be realized using three NOR gates.

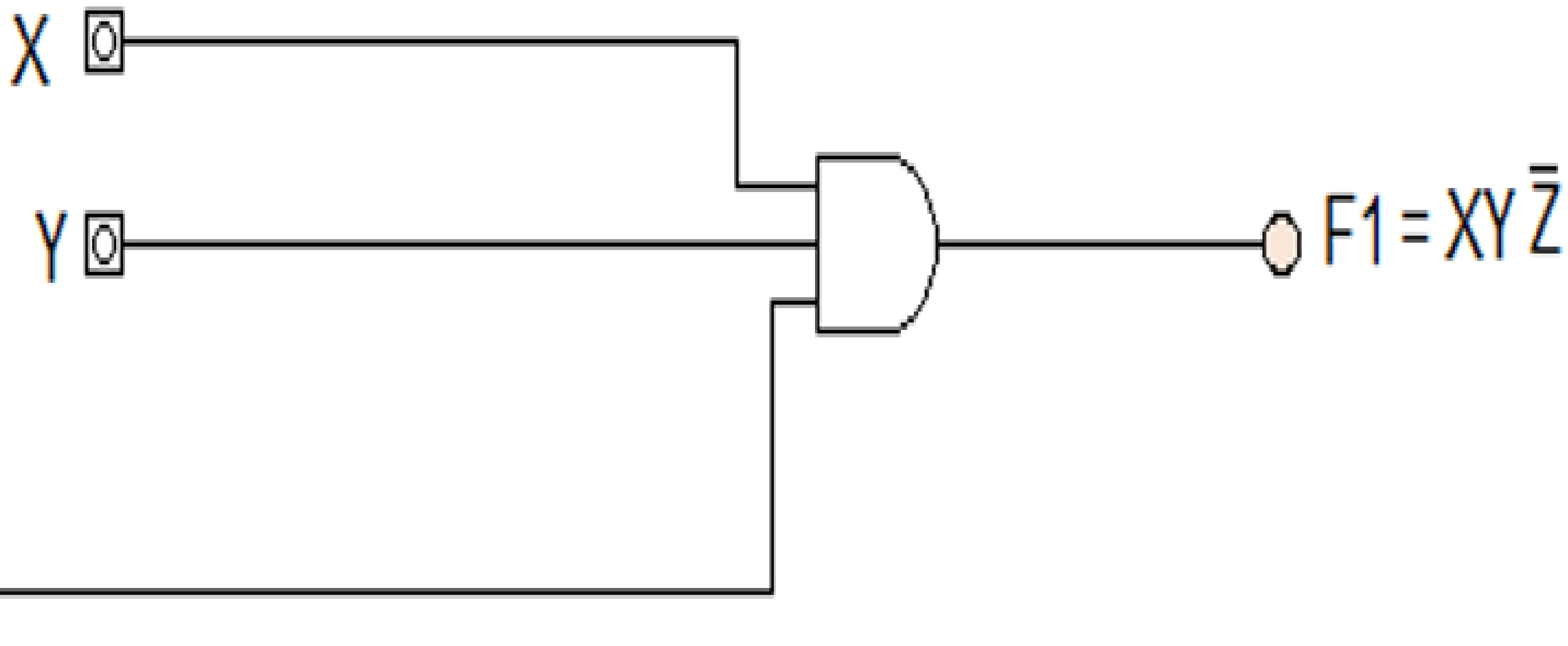


$$Z = \overline{\overline{X} + \overline{Y}} = \overline{\overline{X}} \,.\, \overline{\overline{Y}} = X \,.\, Y$$

Properties of AND gate.

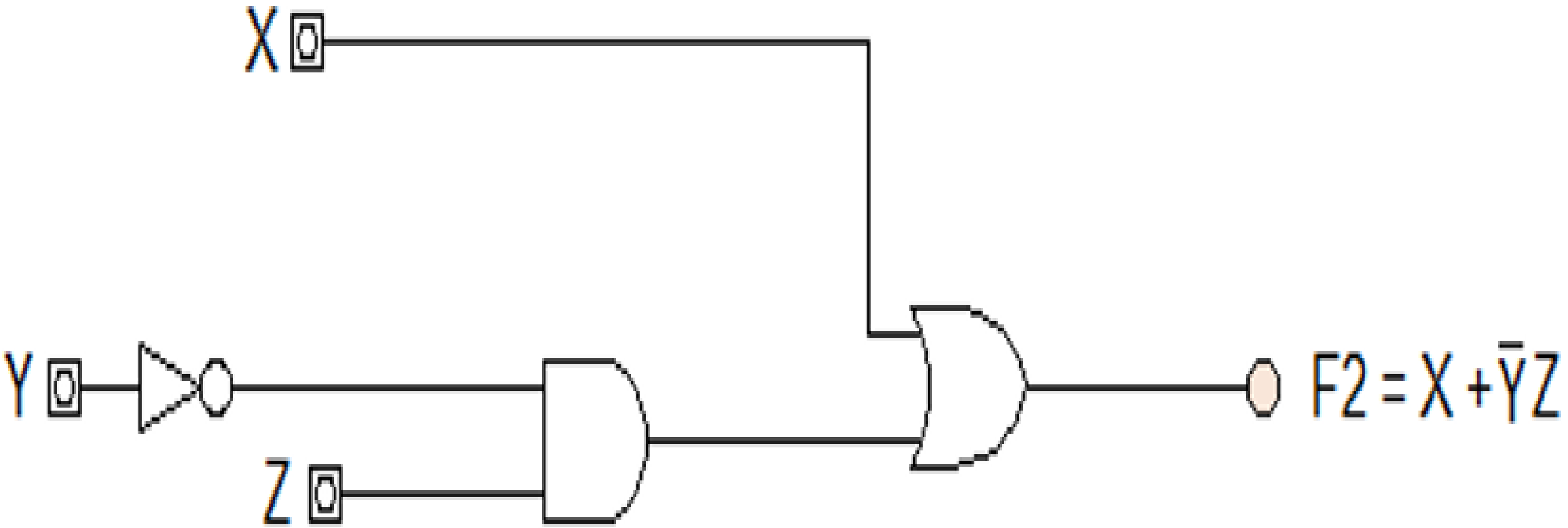
Implementation/realization of Boolean function with logic gates and truth table:

i) $F1 = XY\overline{Z}$



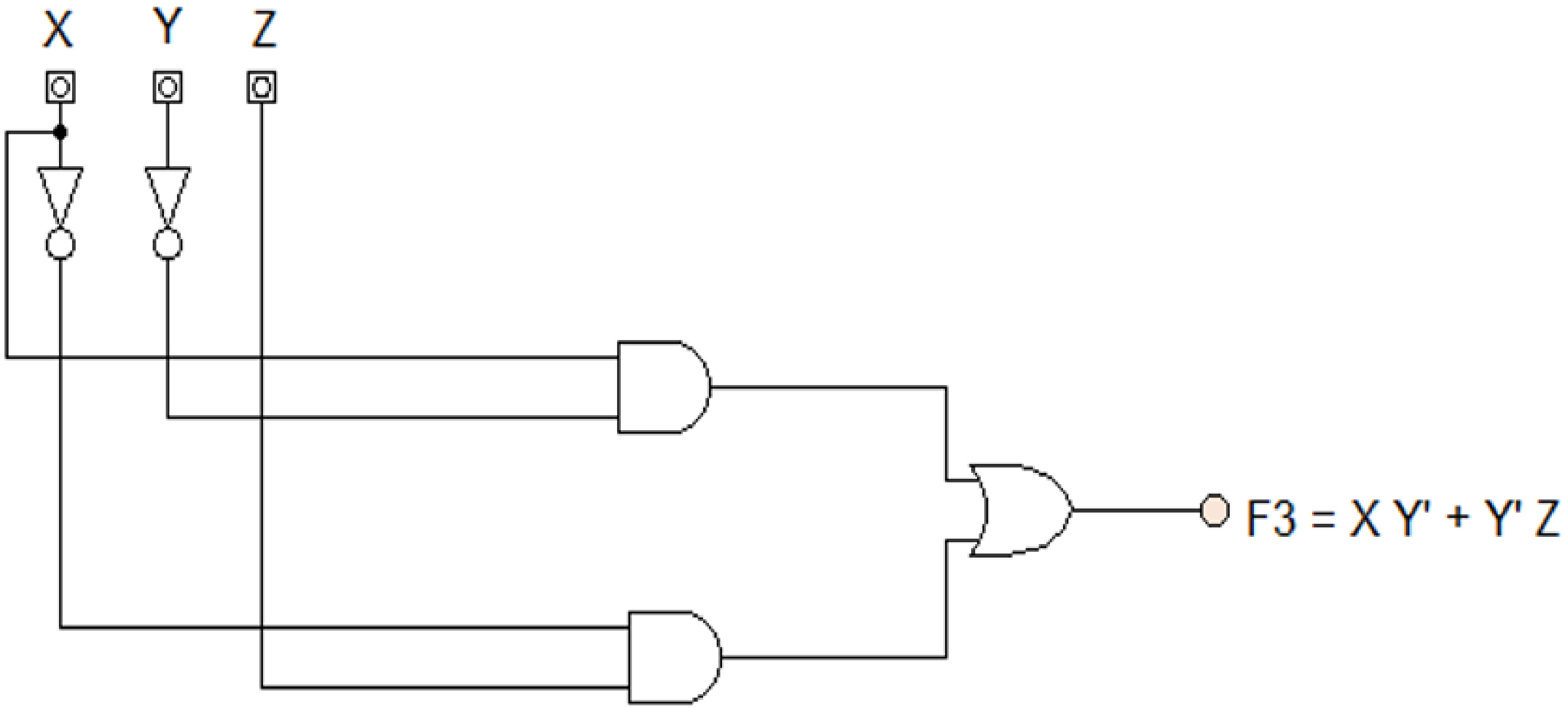
Implementation/realization of Boolean function with logic gates and truth table:

ii) $F2 = X + \bar{Y}Z$



Implementation/realization of Boolean function with logic gates and truth table:

iii) $F3 = X\bar{Y} + \bar{X}Z$



Implementation/realization of Boolean function with logic gates and truth table:

Truth table for F1, F2 and F3:

X	Y	Z	$F1 = XY\bar{Z}$	$F2 = X + \bar{Y}Z$	$F3 = X\bar{Y} + \bar{X}Z$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0

Boolean algebra: It is a method of expressing logic in a mathematical context. It a form of symbolic logic, in which variables have only the values "true (1)" or "false (0)". Relationship between two values are expressed by the Boolean operators AND, OR and NOT. Boolean algebra is used to solve and minimize logic equations. Thus a minimize equation uses the minimum number of logic gates and reduces the cost of operation.

Common Postulates of Boolean algebra :- The postulates of Boolean algebra is originated from the three basic logic operations AND, OR and NOT.

$$\left.\begin{array}{l} 1.\ 0.0 = 0 \\ 2.\ 0.1 = 0 \\ 3.\ 1.0 = 0 \\ 4.\ 1.1 = 1 \end{array}\right\} \text{Derived from AND operation}$$

$$\left.\begin{array}{l} 5.\ 0 + 0 = 0 \\ 6.\ 0 + 1 = 1 \\ 7.\ 1 + 0 = 1 \\ 8.\ 1 + 1 = 1 \end{array}\right\} \text{Derived from OR operation}$$

$$\left.\begin{array}{l} 9.\ \overline{0} = 1 \\ 10.\ \overline{1} = 0 \end{array}\right\} \text{Derived from NOT operation}$$

Activate Windows

Basic theorems of Boolean algebra:

1. $0 . X = 0$
2. $X . 0 = 0$
3. $1 . X = X$
4. $X . 1 = X$
5. $X + 0 = X$
6. $0 + X = X$
7. $X + 1 = 1$
8. $1 + X = 1$
9. $1 + \bar{X} = 1$
10. $X . X = X$
11. $X . \bar{X} = 0$
12. $X + X = X$
13. $X + \bar{X} = 1$
14. $\bar{\bar{X}} = X$

Basic theorems of Boolean algebra:

15. $X + Y = Y + X$
16. $X \, . \, Y = Y \, . \, X$

Commutative Laws

17. $X.(YZ) = (X.Y).Z$
18. $(X + Y) + Z = X + (Y + Z)$

Associative Laws

19. $X.(Y + Z) = X.Y + X.Z$
20. $X + YZ = (X + Y)\,(X + Z)$

Distributive Laws

21. $\overline{X + Y} = \overline{X} \, . \, \overline{Y}$
22. $\overline{X.\,Y} = \overline{X} + \overline{Y}$

De-Morgan's Theorems

23. $X + XY = X$
24. $X.(X + Y) = X$
25. $X\,Y + X\,Y' = X$

Absorption Laws

Basic theorems of Boolean algebra:

26. $(X + Y)(X + Y') = X$
27. $X + X'Y = X + Y$

Proof of Theorem17.

$X.(YZ) = (X.Y).Z$

L.H.S. $= X.(Y.Z)$

IF $X=0$,

$0.(YZ) = 0$ By Theorem 1

R.H.S $= (X.Y).Z$

IF $X = 0$,

$(0.Y).Z = 0.Z = 0$ By Theorem 1

Proved

Basic theorems of Boolean algebra:

Proof of Theorem 19.

$X.(Y + Z) = X.Y + X.Z$

L.H.S.= $X.(Y + Z)$

IF $X = 1$,

$1.(Y + Z) = 1.Y + 1.Z = Y + Z$ By Theorem 3

R.H.S = $X.Y + X.Z$

IF $X = 1$,

$1.Y + 1.Z = Y + Z$ By Theorem 3

Proved

Basic theorems of Boolean algebra:

Proof of Theorem 21 and 22(De-Morgan's theorem).

$$\overline{X+Y}=\overline{X}\,.\,\overline{Y}$$
$$\overline{X.\,Y}=\overline{X}+\overline{Y}$$

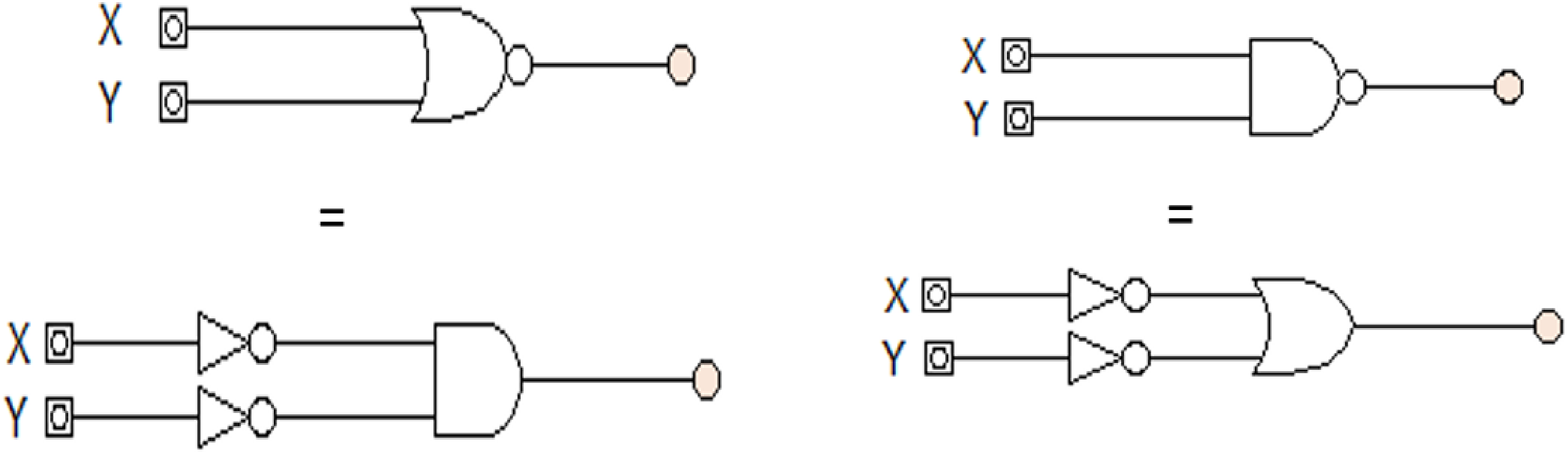
Proof by truth table:

X	Y	$\overline{X+Y}$	$\overline{X}\,.\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

X	Y	$\overline{X\,.\,Y}$	$\overline{X}+\overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Basic theorems of Boolean algebra:

Logic diagram for De-Morgan's theorem:



Hence, De- Morgan's theorems states that the complement of sum is equal to product of the complements and the complement of the product is equal to sum of the complements.

Basic theorems of Boolean algebra:

Proof of Theorem 24.

$X.(X + Y) = X$

$X.(X + Y)$

$= X.X + XY$

$= X + X.Y$ By theorem 10 i.e. $X.X = X$

$= X.(1 + Y) = X.1 = X$ Proved.

Simplification of Boolean Expression by algebraic method:

1. Simplify:

$$X\overline{Y}\,\overline{Z} + X\overline{Y}\,\overline{Z}W + X\overline{Z}$$

$= x\,y'\,z'(1 + w) + x\,z'$

$= x\,y'\,z' + x\,z'$ As, $1 + w = 1$

$= x\,z'\,(y' + 1) = x\,z'.1 = x\,z'$ As, $x + 1 = 1$

2. Simplify : $x + x'\,y + y' + x\,x'\,y + x'\,y'\,y$

$= x + x'\,y + y' + 0 + 0$ As, $x\,x'$ and $y\,y' = 0$

$= x + x'\,y + y' = x + y + y'$ As, $x + x'\,y = x + y$

$= x + 1$ As, $y + y' = 1$

$= 1$

Simplification of Boolean Expression by algebraic method:

3. Simplify: $z(y + z)(x + y + z)$

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

$= (y\,z + z)(x + y + z)$ As, $z\,z = z$

$= z\,(x + y + z)$ As, $z + y\,z = z$

$= x\,z + y\,z + z\,z = x\,z + y\,z + z = x\,z + z(y + 1) = x\,z + z = z(x + 1) = z.1 = z$

Different forms of Boolean Algebra:

1) Sum of Products (SOP): products terms are summed together.
   i) Minimal SOP: Minimum numbers of literals(variable)
      Ex. x y +x’ y +x y’
   ii) Expanded SOP: each products terms consists maximum number of Literal
      Ex. x y z + w x y z +x’ y z +w’ x y z
2) Product of Sums (POS): Summed terms are multiplied together.
   i) Minimal (POS): Minimum number of variables in sum terms.
      Ex. (x + y) (x’ +y) (x + y’)
   ii) Expanded POS: Maximum number of variable in sum terms.
      Ex. (x + y + z) (w + x + y + z) (w’ + x + y’ + z)

Canonical form of logic expression: When each term of logic expression contains All the variables, then that term is called canonical form. It is also called standard form of logic expression.

Minterm: It is canonical form of SOP expression. It is denoted m.
Maxterm: It is canonical form of POS expression. It is denoted by M.

Minterms and Maxterms for three variables:

X	y	z	Minterms	Notation	Maxterm	Notation
0	0	0	$x' y' z'$	m_0	$(x + y + z)$	M_0
0	0	1	$x' y' z$	m_1	$(x + y + z')$	M_1
0	1	0	$x' y z'$	m_2	$(x + y' + z)$	M_2
0	1	1	$x' y z$	m_3	$(x + y' + z')$	M_3
1	0	0	$x y' z'$	m_4	$(x' + y + z)$	M_4
1	0	1	$x y' z$	m_5	$(x' + y + z')$	M_5
1	1	0	$x y z'$	m_6	$(x' + y' + z)$	M_6
1	1	1	$x y z$	m_7	$(x' + y' + z')$	M_7

Example: Express the Boolean function F = A + B'C in sum of minters.

First Term: $A = A(B + B') = AB + AB'$

$= AB(C + C') + AB'(C + C')$

$= ABC + ABC' + AB'C + AB'C'$

Second Term: $B'C = B'C(A + A')$

$= AB'C + A'B'C$

Combining all the terms,

$= ABC+ABC'+AB'C+AB'C' +AB'C+A'B'C$

$=ABC+ABC'+AB'C+AB'C'+A'B'C$

$= m7+m6+m5+m4+m1$

$F(A, B, C) = \sum(1,4,5,6,7)$

Example: Express the Boolean function F = x y + x' z in product of maxterm.

F = x y + x' z

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

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

First Term: (x' + y)

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

Second term: (x + z)

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

Third term: (y + z)

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

Combining all the terms:

(x' + y + z) (x' + y + z') (x + y + z) (x + y' + z)

= M4 M5 M0 M2

= F(X, Y, Z) = $\pi(0,2,4,5)$