

Boolean Logic

सीबीएसई पाठ्यक्रम पर आधारित
कक्षा -11

अध्याय - 14



द्वारा:
संजीव भदौरिया
स्नातकोत्तर शिक्षक (संगणक विज्ञान)
के० वि० बाराबंकी (लखनऊ संभाग)

संजीव भदौरिया, के० वि० बाराबंकी

परिचय

- Boolean Logic को हम boolean algebra भी कह सकते हैं जिसका विकास George Boole ने किया था जोकि एक गणितज्ञ थे ।
- Boolean logic से बाद में बड़े बड़े रहस्य सुलझे और कई समस्याओं का समाधान मिला ।
- इसी logic की वजह से circuit निर्माण में मदद मिली ।
- यह पूरा का पूरा logic “हाँ” या “ना” (“YES” or “NO”) पर आधारित है जिसे हम आजकल true (T) या false (F) के रूप में भी जानते हैं ।
- कंप्यूटर system high voltage या low voltage पर अपने सन्देश प्रसारित करता है जिसे हम true या false के रूप में प्रयोग में ला सकते है और इसका परिपथ (circuit) बना सकते हैं ।
- इस अध्याय में हम इन्ही logic के बारे अध्ययन करेंगे।

Binary मानो वाली राशियाँ (Quantities)

- हम लोगों को दैनिक कार्यों में तार्किक (logic) निर्णय लेने पड़ते हैं जिनका उत्तर या तो हाँ (T) अथवा ना (F) में होता है | जैसे -
 - क्या मैं चाय पी लूँ ?
 - क्या मैं फिल्म देख लूँ?
 - सूरज पूरब से निकलता है ? इत्यादि |
- ऐसे निर्णय जिनका उत्तर सिर्फ “हाँ” या “ना” में मिलता है उन्हें binary decision कहते हैं |
- और “हाँ” (T) या “ना”(F) मानों को truth values कहते हैं
- Boolean variables का मान सिर्फ T या F में से ही कोई हो सकता है |
- इंदिरा गाँधी भारत की एक मात्र महिला प्रधान मंत्री थी | (इसका उत्तर सत्य है T) अतः यह एक logic कथन (statement) है |
- आप क्या कहना चाहते हैं ? (इसका उत्तर T या F में नहीं दिया जा सकता है) अतः यह एक logic statement नहीं है |

Logical Operations

- logical operation तीन प्रकार के होते हैं -
 - AND
 - OR
 - NOT
- इनका प्रयोग मिश्रित कथन (compound statement) बनाने के लिए किया जाता है ।
- जैसे -
 - He prefers coffee *not* tea.
 - He plays guitar *and* sitar.
 - I watch TV on sunday *or* I go for a movie.
- इनका प्रयोग logical variables को जोड़ने के लिए भी किया जाता है ।
- जैसे -

$X \text{ NOT } Y \text{ OR } Z$

$Y \text{ AND } Z \text{ OR } X$

Logical Operations

- Operator के बारे में पढ़ने से पहले हम कुछ बातें जान लें logic statements के बारे में इसके लिए एक उदहारण लेते हैं -

कथन - 1 : मैं चाय पीना चाहता हूँ।

कथन - 2 : चाय तुरंत हाज़िर है।

अब इसके समस्त पहलुओं के बारे में एक तालिका बनाते हैं -

(X) मैं चाय पीना चाहता हूँ। T T F F

(Y) चाय तुरंत हाज़िर है। T F T F

(R)(परिणाम) मैंने चाय पी। T F F F

अब यदि इसकी truth table बनाई जाए तो निम्नवत होगी

- Truth Table एक तालिका होती है जिसमे समस्त logical values की समस्त संभव truth values और समस्त संभव परिणाम होते हैं।

- Truth Table के जब सभी परिणाम true (1) आयें तो यह स्थिति **TAUTOLOGY** कहलाती है और यदि सभी परिणाम false (0) आयें तो यह स्थिति **FALLACY** कहलाती है।

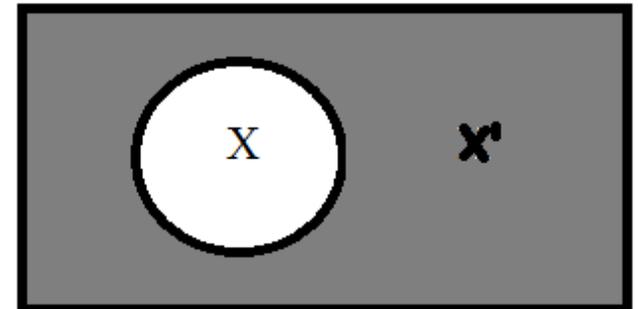


X	Y	R
1	1	1
1	0	0
0	1	0
0	0	0

Logical Operators (NOT)

- logical operator तीन प्रकार के होते हैं -
 - AND Operator
 - OR Operator
 - NOT Operator
- अब हम प्रत्येक operator के बारे में जानेंगे
- NOT operator किसी भी expression के ऊपर (') अथवा ($\bar{\quad}$) लगाकर प्रयोग किया जाता है जैसे X' अथवा \bar{X} | यह एक unary operator होता है | इसे complementation operator भी कहा जाता है |
- $0' = 1$ तथा $1' = 0$ होता है |
- NOT operator के लिए truth table तथा वेन आरेख (Venn Diagram) निम्नवत है -

X	X'
0	1
1	0



Logical Operators (AND)

- AND operator दो variables के बीच में (.) लगाकर प्रयोग किया जाता है जैसे X.Y
- यह boolean algebra का एक और महत्वपूर्ण operation प्रदर्शित करता है जिसे तार्किक गुणन (logical multiplication) कहते हैं |
- X.Y को X AND Y पढ़ा जाता है |
- AND operation के लिए नियम निम्नवत हैं -

$$0 . 0 = 0$$

$$0 . 1 = 0$$

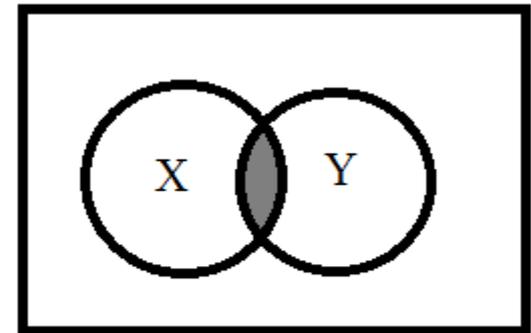
$$1 . 0 = 0$$

$$1 . 1 = 1$$

- AND operator के लिए truth table और वेन आरेख निम्नवत हैं -

X	Y	X.Y
0	0	0
0	1	0
1	0	0
1	1	1

यहाँ नोट करने वाली बात ये है की जब दोनों input 1 होंगे तभी परिणाम 1 होगा अन्यथा 0 |



छाया वाला भाग X.Y प्रदर्शित करता है |

Logical Operators (OR)

- OR operator दो variables के बीच में (+) लगाकर प्रयोग किया जाता है जैसे $X + Y$
- यह boolean algebra का एक और महत्वपूर्ण operation प्रदर्शित करता है जिसे तार्किक योग (logical addition) कहते हैं |
- $X + Y$ को X OR Y पढ़ा जाता है |
- OR operation के लिए नियम निम्नवत हैं -

$$0 + 0 = 0$$

$$0 + 1 = 1$$

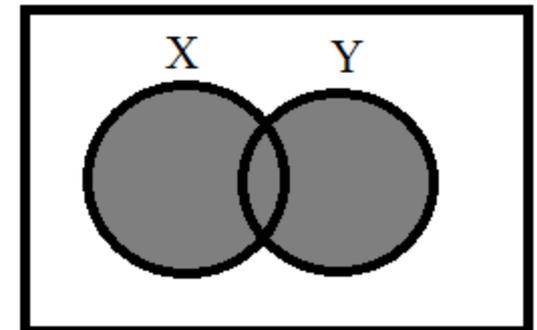
$$1 + 0 = 1$$

$$1 + 1 = 1$$

- AND operator के लिए truth table और वेन आरेख निम्नवत हैं -

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

यहाँ नोट करने वाली बात ये है की जब दोनों input में कोई भी एक input 1 होगा तो परिणाम 1 होगा अन्यथा 0 |



छाया वाला भाग $X+Y$ प्रदर्शित करता है |

Truth Table का प्रयोग करके Boolean Expression का मूल्यांकन

- Boolean Expression को बनाने के लिए logical variables के साथ logical operators का प्रयोग करते हैं | तथा इनका मूल्यांकन करने के लिए truth table का प्रयोग करते हैं | जैसे - $X + (Y.X)' + Z'$ एक boolean expression है | तो इसकी truth table निम्नवत होगी |

X	Y	Z	Y.Z	(Y.Z)'	Z'	X+(Y.Z)'	X+(Y.Z)'+Z'
0	0	0	0	1	1	1	1
0	0	1	0	1	0	1	1
0	1	0	0	1	1	1	1
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	1	1
1	1	1	1	0	0	1	1

Truth Table का प्रयोग करके Boolean Expression का मूल्यांकन

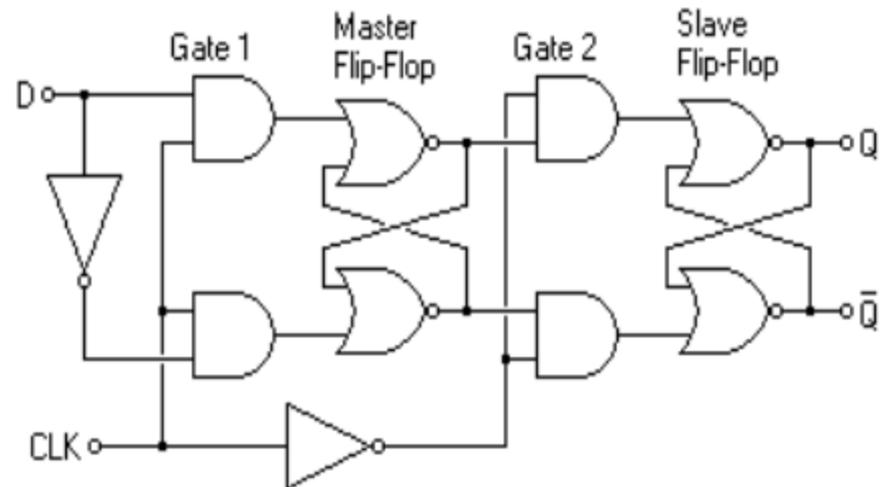
- Boolean logic का और truth table का प्रयोग करते हुए $(X+Y)' = X'.Y'$ को सत्यापित करिए |

X	Y	X+Y	$(X+Y)'$	X'	Y'	$X'.Y'$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

उपरोक्त तालिका में $(X+Y)'$ तथा $X'.Y'$ की तुलना करने पर ये सत्यापित हो जाता है कि दोनों का मान सामान है और बराबर हैं |

Basic Logic Gates

- जब *Shannon* ने दूरभाष के switching circuit में boolean logic का प्रयोग किया तो अभियंताओं (Engineers) ने पाया की boolean algebra को कंप्यूटर इलेक्ट्रॉनिक्स में भी प्रयोग किया जा सकता है | कंप्यूटर में boolean operations को पूरा करने के लिए logic gates का प्रयोग किया जाता है |
- Gate एक basic Electronic Circuit होता है जो एक या अधिक signal को input लेकर एक output signal बनाने के काम आता है |
- Gates एक प्रकार के द्विअवस्था (two-state) digital circuit होते हैं क्योंकि इनके input या output या तो low voltage (अर्थात 0) या high voltage (अर्थात 1) के रूप में होते हैं |
- Gates को logic gates भी कहते हैं क्योंकि इनको boolean logic के साथ समायोजित किया जा सकता है |
- Logic gates तीन प्रकार के होते हैं -
 1. Inverter (NOT Gate)
 2. OR Gate
 3. AND Gate



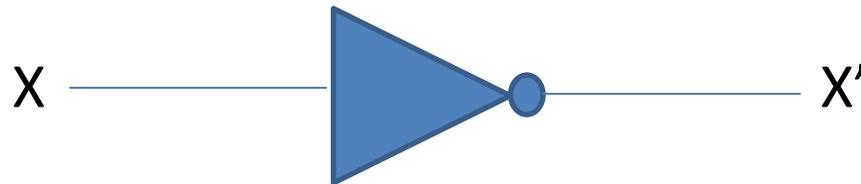
Inverter (NOT Gate)

- NOT गेट एक ही input लेता है और एक output देता है।
- इसका output दिए गए input के अवस्था के ठीक उल्टा होता है।
- इसे NOT गेट इसलिए कहा जाता है क्योंकि इसका output दिए गए input के सामान नहीं होता है।
- इसके output को input का complement(opposite) भी कहते हैं।
- Not गेट की truth table तथा इसका सांकेतिक चित्र निम्न है -

X	X'
Low	High
High	Low

या

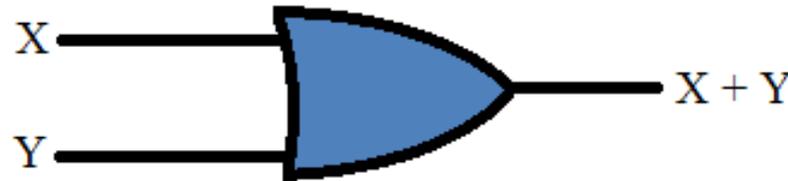
X	X'
0	1
1	0



OR Gate

- OR गेट दो या दो से अधिक input लेता है और एक output देता है |
- इसका output दिए गए input के logical योग होता है |
- यह किसी भी एक input signal के high होने पर high voltage output देता है | जब सभी input पर low voltage होगा तभी low voltage का output मिलता है |
- OR गेट की truth table तथा इसका सांकेतिक चित्र निम्न है -

X	Y	X + Y
Low	Low	Low
Low	High	High
High	Low	High
High	High	High



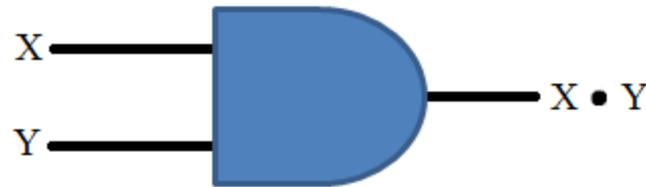
X	Y	X + Y
0	0	0
0	1	1
1	0	1
1	1	1

- यह गेट दो से अधिक input values के लिए भी ऐसे ही कार्य करता है |

AND Gate

- OR गेट दो या दो से अधिक input लेता है और एक output देता है |
- इसका output दिए गए inputs का तार्किक गुणन (logical multiplication) होता है |
- यह किसी भी एक input signal के low होने पर low voltage output देता है | जब सभी input पर high voltage होंगे तभी high voltage का output मिलता है |
- AND गेट की truth table तथा इसका सांकेतिक चित्र निम्न है -

X	Y	X.Y
Low	Low	Low
Low	High	Low
High	Low	Low
High	High	High



X	Y	X.Y
0	0	0
0	1	0
1	0	0
1	1	1

- यह गेट दो से अधिक input values के लिए भी ऐसे ही कार्य करता है |

Boolean logic की आधारभूत अभिधारणायें (Basic Postulates)

– Boolean algebra एक प्रकार का गणितीय निकाय (system) है जिसके कारण इसमें कुछ मूल नियम (fundamental laws) हैं जिन्हें हम basic postulates के नाम से जानते हैं। जो निम्न हैं -

I. यदि $X \neq 0$ तो $X = 1$ होगा तथा यदि $X \neq 1$ तो $X = 0$ होगा।

II. OR Relations (logical addition)

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 1$

III. AND Relations (logical multiplication)

- $0 \cdot 0 = 0$
- $0 \cdot 1 = 0$
- $1 \cdot 0 = 0$
- $1 \cdot 1 = 1$

IV. Complementary Rules:

- $0' = 1$
- $1' = 0$

द्वैतवाद का सिद्धांत (Principle of DUALITY)

- यह boolean logic का अत्यंत महत्वपूर्ण सिद्धांत है एक boolean relation से दूसरा boolean relation बनाया जा सकता है यदि हम निम्न बातों का पालन करें तो -
 1. प्रत्येक OR संकेत (+) को AND संकेत (.) से बदल दें |
 2. प्रत्येक AND संकेत (.) को OR संकेत (+) से बदल दें |
 3. प्रत्येक 0 को 1 में और प्रत्येक 1 को 0 से बदल दें |
- तो इस प्रकार हमें जो expression प्राप्त होगा वह मौलिक expression का द्वैत (dual) होगा | जैसे -
 - $0 + 0 = 0$ का dual $1 . 1 = 1$ होगा |
 - $0 + 1 = 1$ का dual $1 . 0 = 0$ होगा |
 - $1 + 0 = 1$ का dual $0 . 1 = 0$ होगा |
 - $1 + 1 = 1$ का dual $1 . 1 = 1$ होगा |

Boolean Algebra के प्रमुख सिद्धांत

I. 0 तथा 1 के गुण -

I. $0 + X = X$

II. $1 + X = 1$

III. $0 \cdot X = 0$

IV. $1 \cdot X = X$

$$0 + X = X$$

0	X	R
0	0	0
0	1	1

$$1 + X = 1$$

1	X	R
1	0	1
1	1	1

$$0 \cdot X = 0$$

0	X	R
0	0	0
0	1	0

$$1 \cdot X = X$$

1	X	R
1	0	0
1	1	1

II. Idempotence Law -

(a) $X + X = X$

X	X	R
0	0	0
1	1	1

(b) $X \cdot X = X$

X	X	R
0	0	0
1	1	1

Boolean Algebra के प्रमुख सिद्धांत

III. Involution –

$$(X')' = X$$

X	X'	(X')'
0	1	0
1	0	1

IV. Complementarity Law –

$$(a) X + X' = 1$$

X	X'	R
0	1	1
1	0	1

$$(b) X \cdot X' = 0$$

X	X'	R
0	1	0
1	0	0

V. Commutative Law –

$$(a) X + Y = Y + X$$

X	Y	X + Y	Y + X
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

$$(b) X \cdot Y = X \cdot Y$$

X	Y	X . Y	Y . X
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Boolean Algebra के प्रमुख सिद्धांत

VI. Associative Law –

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

X	Y	Z	Y + Z	X + Y	X+ (Y + Z)	(X + Y) + Z
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Boolean Algebra के प्रमुख सिद्धांत

VI. Associative Law –

$$(b) X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$$

X	Y	Z	Y . Z	X . Y	X . (Y . Z)	(X . Y) . Z
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	1	0	0
1	1	1	1	1	1	1

Boolean Algebra के प्रमुख सिद्धांत

VII. Distributive Law –

$$(a) X \cdot (Y + Z) = X \cdot Y + X \cdot Z$$

X	Y	Z	Y+Z	X.Y	X.Z	X.(Y+Z)	X.Y+X.Z
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

Boolean Algebra के प्रमुख सिद्धांत

VII. Distributive Law –

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

X	Y	Z	Y.Z	X+Y	X+Z	X+Y.Z	(X+Y).(X+Z)
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Handwritten proof of the distributive law $X + Y.Z = (X + Y) . (X + Z)$. The proof shows the right-hand side (R.H.S.) being simplified to match the left-hand side (L.H.S.).

$$\begin{aligned} \text{R.H.S. } & (X+Y).(X+Z) \\ &= X.X + X.Z + Y.X + Y.Z \\ &= X + X.Z + XY + YZ \\ &= X + XY + XZ + YZ \\ &= X(1+Y) + XZ + YZ \quad \text{1+Y=1} \\ &= X + XZ + YZ \\ &= X(1+Z) + YZ \quad \text{1+Z=1} \\ &= X + YZ \\ &= \text{L.H.S.} \end{aligned}$$

Boolean Algebra के प्रमुख सिद्धांत

VIII. Absorption Law –

$$(a) X + X \cdot Y = X$$

X	Y	X.Y	X+X.Y
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

$$(b) X \cdot (X + Y) = X$$

X	Y	X + Y	X.(X+Y)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

IX. कुछ अन्य नियम (Third Distributive Law)

$$X + X' \cdot Y = X + Y$$

X	Y	X'	X'.Y	X + X'.Y	X+Y
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Boolean Algebra के प्रमुख सिद्धांत

X. DEMORGAN's Law –

$$(a) (X+Y)' = X' \cdot Y'$$

X	Y	$(X+Y)'$	$X' \cdot Y'$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

$$(b) (X \cdot Y)' = X' + Y'$$

X	Y	$(X \cdot Y)'$	$X' + Y'$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

अभी तक हमने जितने भी नियम पढ़े उनका सत्यापन truth table से किया गया है | जबकि इनका सत्यापन गणितीय विधि से भी किया जा सकता है | Distributive Law का गणितीय सत्यापन भी दर्शाया गया है | उसी प्रकार हम किसी भी नियम का सत्यापन गणितीय विधि से कर सकते हैं |

Sum of Product : $ABC, XY, X'YZ'$ इत्यादि |

Product of Sum : $A+B+C, X+Y'+Z'$ इत्यादि हैं |

Boolean Expression को algebraic विधि के द्वारा कम करना

- $AB'CD' + AB'CD + ABCD' + ABCD$ को न्यूनतम रूप में लाओ ।

हल:

$$\begin{aligned} & AB'CD' + AB'CD + ABCD' + ABCD \\ &= AB'C(D'+D) + ABC(D'+D) \\ &= AB'C + ABC \\ &= AC(B'+B) = AC.1 = AC \end{aligned}$$

इसी प्रकार किसी भी boolean expression को न्यूनतम स्तर तक लाया जा सकता है ।

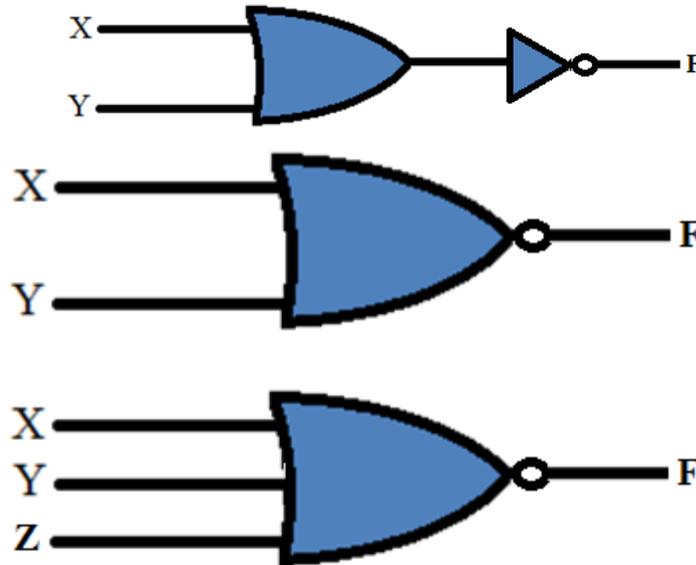
कुछ अन्य Logic Gates

- AND, OR, NOT Gates के अलावा कुछ Derived Gates भी होते हैं जैसे -
- NOR, NAND, XOR, XNOR Gates.

NOR Gate:

NOR गेट में दो या अधिक input signal होते हैं तथा एक ही output signal होता है | यदि समस्त input का मान 0 (low) है तो output signal 1 (high) होगा

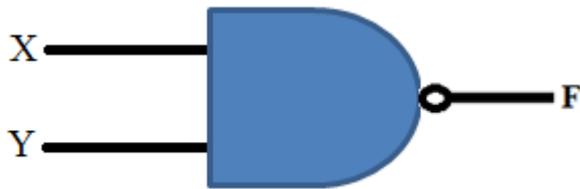
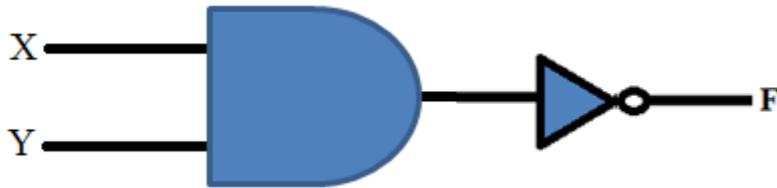
X	Y	F
0	0	1
0	1	0
1	0	0
1	1	0



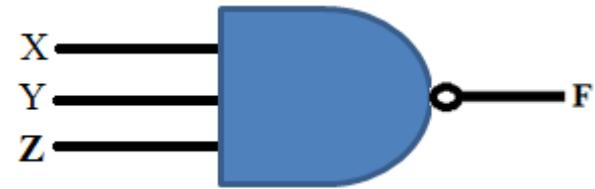
X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

NAND Gate

- NAND गेट में दो या अधिक input signal होते हैं तथा एक ही output signal होता है | यदि समस्त input का मान 1(high) है तो output signal 0(low) होगा | इसे भी universal गेट भी कहते हैं क्योंकि इसके द्वारा पुनः fundamental गेट बनाये जा सकते हैं |



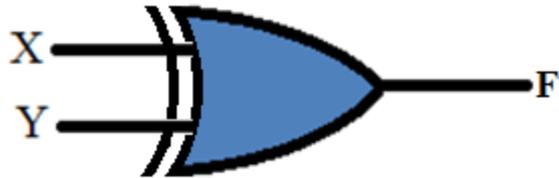
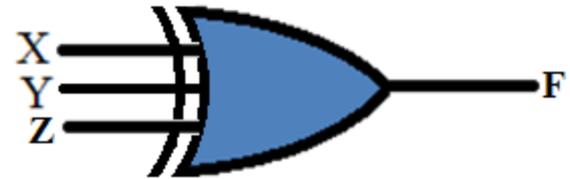
X	Y	F
0	0	1
0	1	1
1	0	1
1	1	0



X	Y	Z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

XOR Gate

- XOR Gate सिर्फ उन्हीं input के लिए 1 (high) मान देगा जिनमे सारे input signals में 1 की संख्या विषम (Odd) होगी | इसे universal गेट भी कहते हैं क्योंकि इसके द्वारा पुनः fundamental गेट बनाये जा सकते हैं |

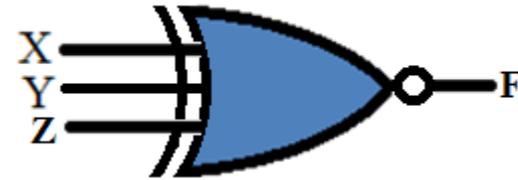
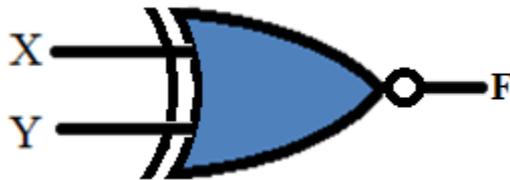


No of 1's	X	Y	F
Even	0	0	0
Odd	0	1	1
Odd	1	0	1
Even	1	1	0

No of 1's	X	Y	Z	F
Even	0	0	0	0
Odd	0	0	1	1
Odd	0	1	0	1
Even	0	1	1	0
Odd	1	0	0	1
Even	1	0	1	0
Even	1	1	0	0
Odd	1	1	1	1

XNOR Gate

- XOR Gate सिर्फ उन्हीं input के लिए 1 (high) मान देगा जिनमे सारे input signals में 1 की संख्या सम (Even) होगी |



No of 1's	X	Y	F
Even	0	0	1
Odd	0	1	0
Odd	1	0	0
Even	1	1	1

No of 1's	X	Y	Z	F
Even	0	0	0	1
Odd	0	0	1	0
Odd	0	1	0	0
Even	0	1	1	1
Odd	1	0	0	0
Even	1	0	1	1
Even	1	1	0	1
Odd	1	1	1	0

महत्वपूर्ण सवाल

1. NAND गेट से AND, NOT तथा OR गेट बनाइये व् इसका परिपथ चित्र बनाइये ।
2. NOR गेट से AND, NOT तथा OR गेट बनाइये व् इसका परिपथ चित्र बनाइये ।
3. निम्नलिखित के लिए logic circuit बनाइये ।
 - I. $F(A,B,C) = AB + AC' + A'B'C$
 - II. $F(X,Y,Z) = (X+Y) \cdot (X'+Z') \cdot (Y+Z)$

धन्यवाद

और अधिक पाठ्य-सामग्री हेतु निम्न लिंक पर क्लिक करें -

www.pythontrends.wordpress.com

एक शुरुआत pythontrends

पाइथन सीखें और सिखाएं

मुख्य पृष्ठ/Home

संपर्क/Contact

लेख/Articles

छायाचित्र/Images

विडियो/Video

अध्यायवार पाठ्यसामग्री/Lesson wise
Study Material

उपयोगी लिंक्स / Useful Links

पाइथन प्रोग्राम/Python Programs

नमस्ते दोस्तों ! /Hello Friends!



यह ब्लॉग उन बच्चों की मदद के लिए बनाया गया है जो python में प्रोग्रामिंग सीख रहे हैं | यह ब्लॉग द्विभाषीय होगा जिससे सीबीएसई बोर्ड के वे बच्चे जिन्हें अंग्रेजी भाषा में समस्या होती है उन्हें सही मार्गदर्शन करेगा तथा प्रोग्रामिंग में उनकी सहायता करेगा | जैसा की हम जानते हैं की हमारे देश में कई क्षेत्र और कई लोग ऐसे हैं जिनकी अंग्रेज़ी उतनी मज़बूत नहीं है क्यों कि ये हमारी मातृभाषा नहीं है | तो हमें कभी कभी अंग्रेज़ी के कठिन शब्दों को

