Switching Circuit & Logic Design

Lecture 7 : Logic Gates & Boolean Algebra

Things to cover

- LOGIC GATES
  - Basic Gates
  - Universal Gates
  - XOR, XNOR

Book : Fundamentals of Digital Circuits, A. Anand Kumar

Prove the following

A + BC = (A+B).(A+C)

A + B = AB + A’B + AB’

Prove the following

A + BC = (A+B).(A+C) ⟶ Dual of first Distributive Law

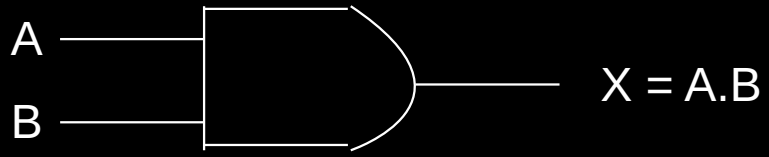
(A+B).(A+C)	= AA + AC + BA + BC	//Distributive Law
	= A + AC + BA + BC	//Idempotence Law
	= A.1 + AC + BA + BC	//Identity
	= A(1+C) + BA + BC	//Distributive
	= A.1 + BA + BC	//Dominance
	= A.1 + AB + BC	//Commutative
	= A (1+B) + BC	//Distributive
	= A.1 + BC	//Dominance
	= A + BC	//Identity

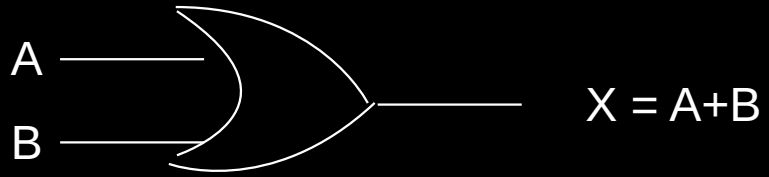
Prove the following

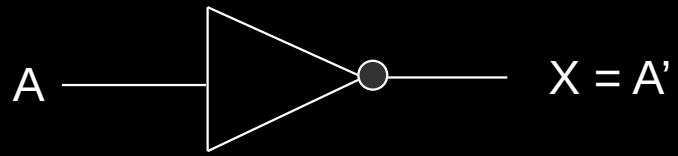
A + B = AB + A’B + AB’

AB + A’B + AB’	= BA + BA’ + AB’	// Commutative Law
	= B(A + A’) + AB’	// Distributive Law
	= B.1 + AB’	// Complementary Law
	= B + AB’	// Idempotent Law
	= (B+A).(B+B’)	// Distributive Law
	= (B+A).1	// Complementary Law
	= B+A	// Idempotent Law
	= A+B	// Commutative Law

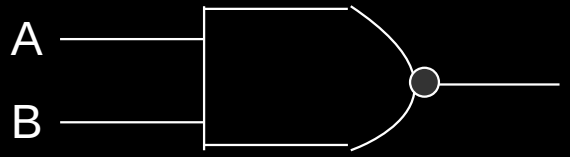
Basic Gates





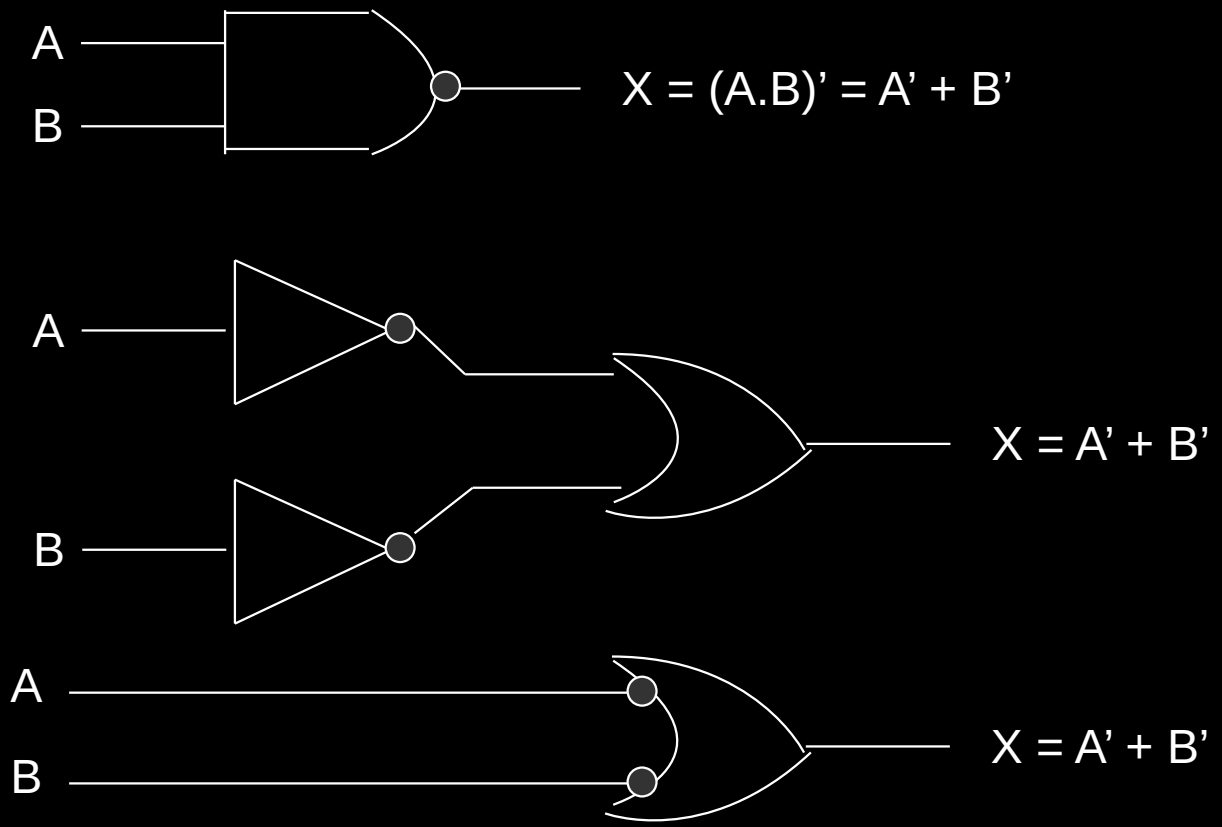


Universal Gates -- NAND

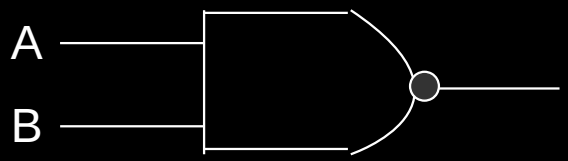


X = (A.B)’ = A’ + B’

Universal Gates -- NAND

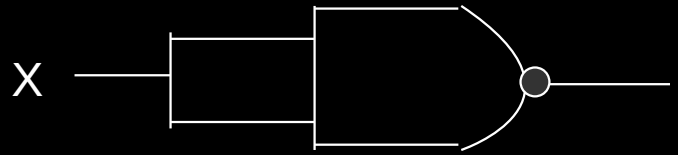


Implement Inverter

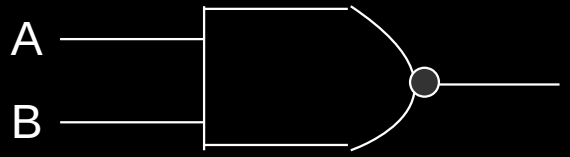


X = (A.B)’ = A’ + B’

If A = B, => X = (A.A)’ = A’



Implement AND, OR Gates

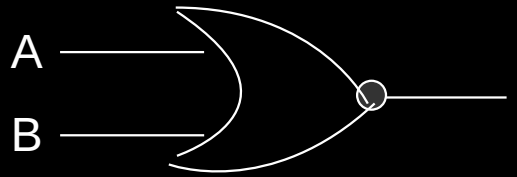


X = (A.B)’ = A’ + B’

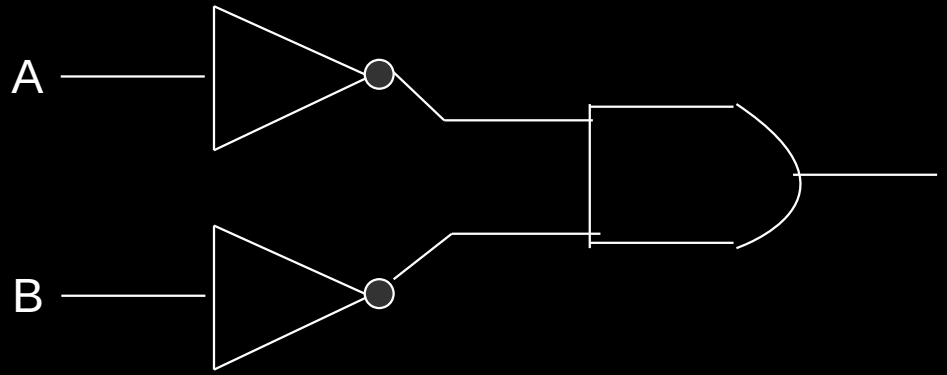
Universal Gates -- NOR

X = (A + B)’ = A’ . B’

Universal Gates -- NOR

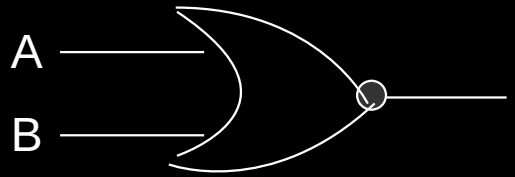


X = (A + B)’ = A’ . B’



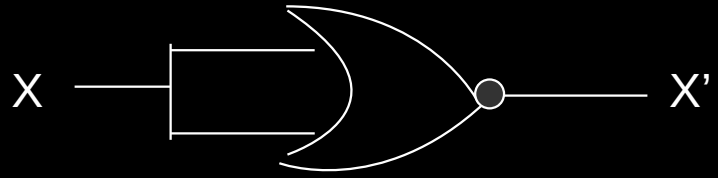
X = A’ . B’

Implement NOT

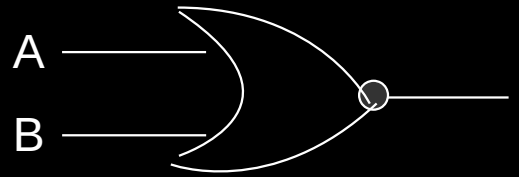


X = (A + B)’ = A’ . B’

If A = B, => X = (A+A)’ = A’

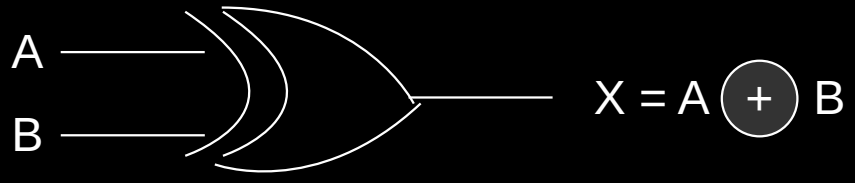


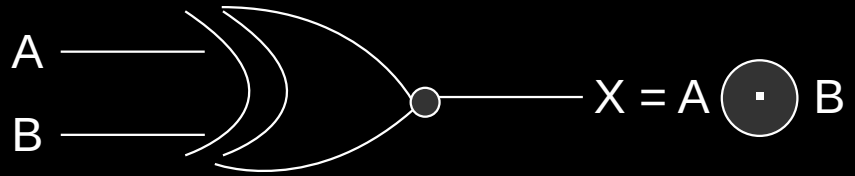
Implement AND, OR Gates



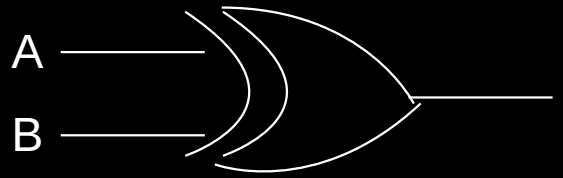
X = (A + B)’ = A’ . B’

XOR, XNOR Gate





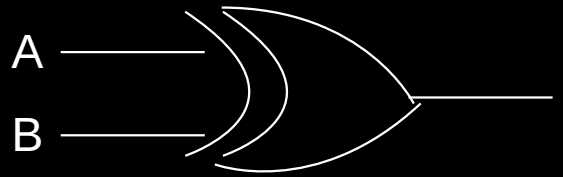
XOR Gate



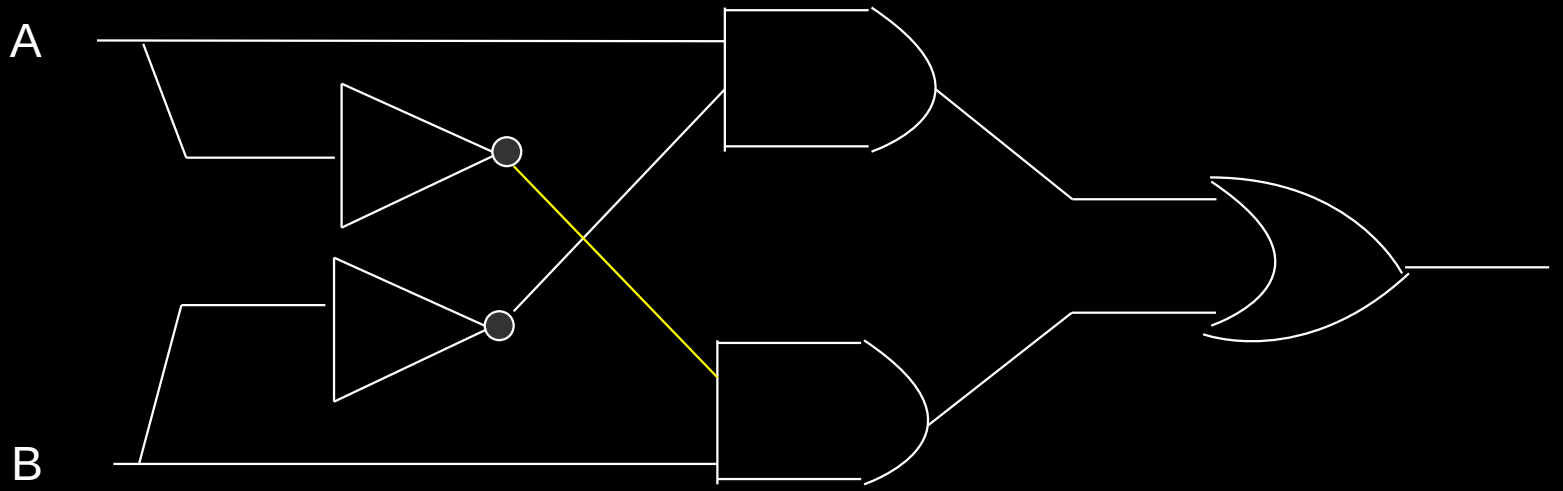
$X = A \oplus B = A'.B + A \cdot B'$

A	B	$X = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

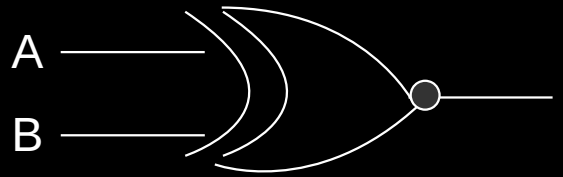
XOR Gate



$X = A \oplus B = A'.B + A.B'$



XNOR Gate



$X = A \odot B$

$= (A + B)'$

A	B	$X = A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

XNOR Gate