

Exclusive-OR

$\oplus$ denotes exclusive-OR

This is the definition of exclusive-OR:

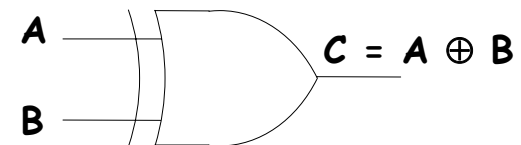
$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



Exclusive-OR Theorems

$$X \oplus 0 = X$$

$$X \oplus 1 = X'$$

$$X \oplus X = 0$$

$$X \oplus X' = 1$$

$$X \oplus Y = Y \oplus X$$

Commutative law

$$(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z) = X \oplus Y \oplus Z$$

Associative law

$$(X \oplus Y)' = X \oplus Y' = X' \oplus Y$$

Exclusive-OR can be expressed
using
AND, OR and NOT:

$$X \oplus Y = X'Y + XY'$$

**Exercise: Prove the above
expression using truth table.**

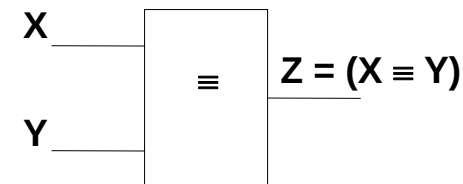
Equivalence Operation

$\equiv$ denotes equivalence

This is the definition of equivalence

$$\begin{array}{ll} (0 \equiv 0) = 1 & (0 \equiv 1) = 0 \\ (1 \equiv 0) = 0 & (1 \equiv 1) = 1 \end{array}$$

X	Y	$X \equiv Y$
0	0	1
0	1	0
1	0	0
1	1	1



Equivalence Theorems

$$(X \equiv 0) = X'$$

$$(X \equiv 1) = X$$

$$(X \equiv X) = 1$$

$$(X \equiv X') = 0$$

$$(X \equiv Y) = (Y \equiv X)$$

Commutative law

$$[(X \equiv Y) \equiv Z] = [X \equiv (Y \equiv Z)] = [X \equiv Y \equiv Z]$$

Associative law

$$(X \equiv Y)' = (X \equiv Y') = (X' \equiv Y)$$

Equivalence can be expressed
using
AND, OR and NOT:

$$(X \equiv Y) = XY + X'Y'$$

Exercise: Prove the above expression using truth table.

Exclusive-OR Complement

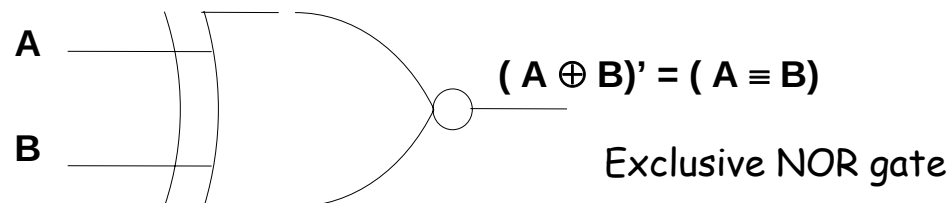
Exclusive-OR is the complement of Equivalence

$$(X \oplus Y)' = (X \equiv Y)$$

Proof:

$$\begin{aligned}(X \oplus Y)' &= (X'Y + XY')' \\&= (X + Y')(X' + Y) \\&= XX' + XY + X'Y' + YY' \\&= XY + X'Y' = (X \equiv Y)\end{aligned}$$

Since Equivalence is the complement of Exclusive-OR, here is another representation of the Equivalence Gate



Example

Convert to sum of products form:

$$(A + B) \equiv (A \oplus D) C$$

$$\text{Use } (X \equiv Y) = (XY + X'Y')$$

$$(A + B)(A \oplus D) C + (A + B)' [(A \oplus D) C]'$$

$$\text{Use } (XY)' = X' + Y' \text{ and } (X \oplus Y)' = X' + Y'$$

$$(A + B)(A \oplus D) C + (A' B') [(A \oplus D)' + C']$$

$$\text{Use } (X \oplus Y)' = (X \equiv Y)$$

$$(A + B)(A \oplus D) C + A' B' [(A \equiv D) + C']$$

Change $\equiv$ and $\oplus$ form

$$(A + B)(A'D + AD') C + A' B' [(AD + A'D') + C']$$

Multiply out

$$AA'CD + AACD' + A'BCD + ABCD' + AA'B'D + A'A'B'D' + A'B'C'$$

$$\text{Use } XX' = 0 \text{ and } XX = X$$

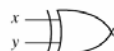
$$ACD' + A'BCD + ABCD' + A'B'D' + A'B'C'$$

$$\text{Use } X + XY = X$$

$$ACD' + A'BCD + A'B'D' + A'B'C'$$

Digital Logic Gates

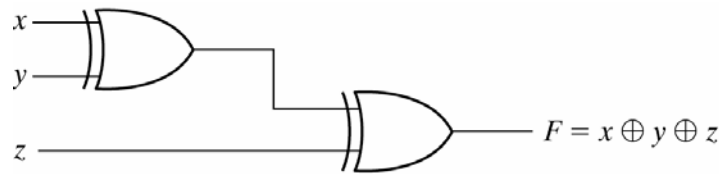
Commonly used digital logic gates are given in the following table

Name	Graphic symbol	Algebraic function	Truth table															
AND		$F = xy$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Inverter		$F = x'$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
Buffer		$F = x$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	
NAND		$F = (xy)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = (x + y)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	0
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

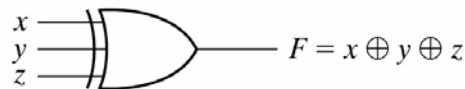
Extension to Multiple Inputs

- The commonly used gates, except the buffer and the inverter, can be extended to have more than two inputs if the binary operation is commutative and associative (e.g. AND, OR, XOR).
- NAND and NOR are commutative but not associative
 $[(x + y)' + z]' \neq [x + (y + z)']'$
- We define the multiple NOR and NAND gate as a complemented OR or AND gate.
e.g. three variable NOR gate $\rightarrow (x + y + z)'$

Example: 3 input XOR Gate



(a) Using 2-input gates



(b) 3-input gate

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

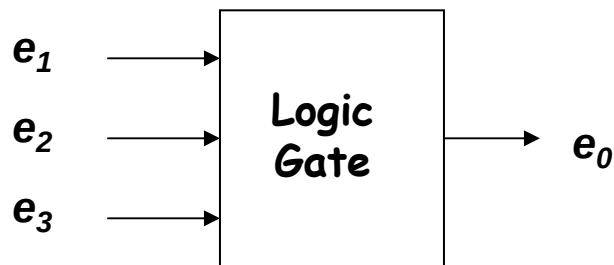
(c) Truth table

Integrated Circuits

- Integrated circuits are classified based on the levels of integration.
 - SSI < 10 gates
 - MSI 10 – 100 gates (e.g., decoders, adders, MUXs)
 - LSI 100 – few 1000s (e.g., processors, memory chips)
 - VLSI > 1000s
- Digital integrated circuits may also be classified by the specific circuit technology
 - TTL Transistor-Transistor-Logic
 - ECL Emitter-Coupled-Logic
 - CMOS Complementary Metal-Oxide-Semiconductor

Positive Logic and Negative Logic

Conversion of 0s and 1s to voltages



e_1	e_2	e_3	e_0
0	0	0	0
0	0	+V	0
0	+V	0	0
0	+V	+V	0
+V	0	0	0
+V	0	+V	0
+V	+V	0	0
+V	+V	+V	+V

Positive Logic

Let 0 volts represent logical 0
and +V volts represent logical 1

e_1	e_2	e_3	e_0	e_1	e_2	e_3	e_0
0	0	0	0	0	0	0	0
0	0	+V	0	0	0	1	0
0	+V	0	0	0	1	0	0
0	+V	+V	0	0	1	1	0
+V	0	0	0	1	0	0	0
+V	0	+V	0	1	0	1	0
+V	+V	0	0	1	1	0	0
+V	+V	+V	+V	1	1	1	1

The gate is a logical AND gate

Negative Logic

Let 0 volts represent logical 1
and +V volts represent logical 0

e_1	e_2	e_3	e_0	e_1	e_2	e_3	e_0
0	0	0	0	1	1	1	1
0	0	+V	0	1	1	0	1
0	+V	0	0	1	0	1	1
0	+V	+V	0	1	0	0	1
+V	0	0	0	0	1	1	1
+V	0	+V	0	0	1	0	1
+V	+V	0	0	0	0	1	1
+V	+V	+V	+V	0	0	0	0

The same gate is a logical OR gate

Positive Logic and Negative Logic

- If a network realizes a function f when inputs and outputs are defined with positive logic,
- The same network will realize the dual function f^D when defined with negative logic
- Please note the triangle symbol to indicate negative logic
- We will assume positive logic only in our course

x	y	F
L	L	L
L	H	L
H	L	L
H	H	H

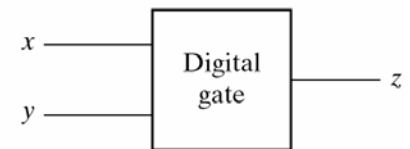
(a) Truth table with H and L

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

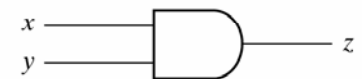
(c) Truth table for positive logic

x	y	z
1	1	1
1	0	1
0	1	1
0	0	0

(e) Truth table for negative logic



(b) Gate block diagram



(d) Positive logic AND gate



(f) Negative logic OR gate

Example

Factor to obtain a product of 3 terms:

$$A'B'C' + A'D + ABC' + AD$$

$Y'X$

YX

Use $XY + XY' = X$

$$A'B'C' + D + ABC'$$

$$C'(A'B' + AB) + D$$

Use $X + YZ = (X + Y)(X + Z)$

$$(C' + D) [(A'B' + AB) + D]$$

X

YZ

$$(C' + D) [(A'B' + A)(A'B' + B) + D]$$

YZ

X

YZ

X

$$(C' + D) [(A' + A)(B' + A)(A' + B)(B' + B) + D]$$

Use $X + X' = 1$

$$(C' + D) [(A + B')(A' + B) + D]$$

Use $X + YZ = (X + Y)(X + Z)$ again

$$(C' + D) (A + B' + D) (A' + B + D)$$

Another Example

Multiply out to obtain a sum of 3 terms:

$$(A' + B + D) (A + C) (A + B' + D) (A' + B)$$

X

Y

X

Use $X (X + Y) = X$

$$(A + C) (A + B' + D) (A' + B)$$

X

Z

X'

Y

Use $(X + Z) (X' + Y) = XY + X'Z$

$$(AB + A'C) (A + B' + D)$$

Multiply out

$$ABA + ABB' + ABD + A'CA + A'CB' + A'CD$$

Use $X \cdot X' = 0$ and $X \cdot X = X$

$$AB + ABD + A'B'C + A'CD$$

X

X Y

Use $X + XY = X$

$$AB + A'B'C + A'CD$$