Université Concordia University
Department of Electrical & Computer Engineering

COEN 212: DIGITAL SYSTEMS DESIGN I
Lecture 3: Logic Gates

Instructor: Dr. Reza Soleymani, Office: EV-5.125,
Telephone: 848-2424 ext.: 4103.

Lecture 3:
Objectives of this lecture

Université Concordia University
Department of Electrical & Computer Engineering

- Basic forms of Boolean expressions: Canonical and standard forms.
- Logic Gates.
- Multiple input gates.

Lecture 3:
Reading for this lecture

- Digital Design by M. Morris R. Mano and Michael D. Ciletti, 6th Edition, Pearson, 2018:
  - Chapter 2 (2.6 to 2.9)

Lecture 3: Literals

Université Concordia University
Department of Electrical & Computer Engineering

- In a Boolean expression, each term is represented by a gate and each variable in a term is an input to that gate.
- Each variable appearing in a function, whether in complimented or original form is called a literal.
- For example, $F = x'y'z + xyz' + x'z'$ has three terms and 8 literals.

Université Concordia University
Department of Electrical & Computer Engineering

- **Standard forms:**
  - Sum of products
  - Product of sums.

- **Standard sum of product form:**
  - function is represented as the sum (OR) of several terms.
  - Each term is the product (AND) of one or more literals.
  - example,

$$F_1 = x + x'y + xyz + x'y'z'$$

is in Sum of Products form.

Université Concordia University
Department of Electrical & Computer Engineering

- **Standard Product of Sum form:**

- a function is represented as product (AND) of several factors.
- Each factor is the sum (OR) several terms.
- Example

$$F_2 = x(x' + y')(x + z')(x + y + z)$$

is in Product of Sums form.

Lecture 3:
Canonic Forms: minterms

- With two variables x and y, we can form 4 products or xy, $x'y$, xy', $x'y'$ each called a standard product or a minterm.
- With n variables, there are 2^n minterms enumerated using numbers 0 to $2^n - 1$ (in binary)
- minterms corresponding to each row of the truth table is formed by ANDing the variables (for 1's) or their complements (for 0's).
- Example: $n = 3$

x	y	z	Term	Designation
0	0	0	$x'y'z'$	m_0
0	0	1	$x'y'z$	m_1
0	1	0	$x'yz'$	m_2
0	1	1	$x'yz$	m_3
1	0	0	$xy'z'$	m_4
1	0	1	$xy'z$	m_5
1	1	0	xyz'	m_6
1	1	1	xyz	m_7

Lecture 3:
Canonic Forms: Maxterms

Université Concordia University
Department of Electrical & Computer Engineering

- A standard sum or Maxterm is formed by ORing the n variables. Those with 0 value appear uncomplemented and those with value 1 appear complemented.
- Example:

x	y	z	Maxterm	Designation
0	0	0	$x+y+z$	M_0
0	0	1	$x+y+z'$	M_1
0	1	0	$x+y'+z$	M_2
0	1	1	$x+y'+z'$	M_3
1	0	0	$x'+y+z$	M_4
1	0	1	$x'+y+z'$	M_5
1	1	0	$x'+y'+z$	M_6
1	1	1	$x'+y'+z'$	M_7

- Note that each Maxterm is the complement of the corresponding minterm.

Lecture 3:
Canonic Sum of Product Forms

Université Concordia University
Department of Electrical & Computer Engineering

A Boolean function can be expressed algebraically by adding the minterms for those rows of the truth table for which the function is 1.

Example:

x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$F_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$ or $F_1(x, y, z) = \sum(1,4,7)$.

and,

$F_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$

or $F_2(x, y, z) = \sum(3,5,6,7)$.

Lecture 3:
Canonic Product of Sum Forms

Université Concordia University
Department of Electrical & Computer Engineering

- AND the Maxterms corresponding to those rows of the truth table for which the function is equal to 0.

$F_1 = (x + y + x).(x + y' + z).(x + y' + z').(x' + y + z').(x' + y' + z) = M_0.M_2.M_3.M_5.M_6.$

- We can also add together the minterms corresponding rows where F_1 is zero and then take the compleme, i.e.,
  - $F_1{}' = x'y'z' + x'yz' + x'yz + xy'z + xyz',$
  - $F_1 = (x'y'z' + x'yz' + x'yz + xy'z + xyz')' = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z) = M_0M_2M_3M_5M_6$
- This can be written as $F_1(x, y, x) = \prod(0,2,3,5,6).$

- *Similarly,* $F_2 = (x + y + z)(x + y + z')(x + y' + z)(x' + y + z) = M_0.M_1.M_2.M_4 = \prod(0,1,2,4)$

Lecture 3: Transforming Canonical forms

Concordia University
Department of Electrical & Computer Engineering

- A function represented in sum of minterms can be converted to a product of Maxterms form and vice versa.
- Note that:
  - a sum of minterms expression has those minterms for which the function is 1.
  - The complement of the function consists of the rest of the terms.
  - The complement of the complement will have Maxterms for these rows (the rest of the terms).
  - The complement of the complement of the function is the function itself.
- For example, in the previous example,
  - $F_1 = \sum(1,4,7)$.
  - $F_1{}'$ is the sum of the rest of minterms, i.e., $F_1{}' = \sum(0,2,3,5,6)$.
  - So, $F_1 = (F_1{}')' = \prod(0,2,3,5,6)$.

Lecture 3: Other Logic Operations

- n variables take 2^n values, i.e., the truth table for an n-bit function has 2^n rows.
- Each row's value can be a 1 or a 0. So, there are 2^{2^n} functions with n binary inputs.
- For $n = 2$, we have four values for x and y. So, there are $2^4 = 16$ functions.

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	**0**	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	**1**	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	**0**	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	**1**	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- Each column defines a different gate (or function).
- For example: Column 4 (F_1) is AND and 10 (F_7) is OR.

Lecture 3:
Other Logic Operations

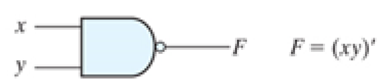
Université Concordia University
Department of Electrical & Computer Engineering

Boolean Functions	Operator Symbol	Name	Comments
$F_0 = 0$		Null	Binary constant 0
$F_1 = xy$	$x \cdot y$	AND	x and y
$F_2 = xy'$	x/y	Inhibition	x, but not y
$F_3 = x$		Transfer	x
$F_4 = x'y$	y/x	Inhibition	y, but not x
$F_5 = y$		Transfer	y
$F_6 = xy' + x'y$	$x \oplus y$	Exclusive-OR	x or y, but not both
$F_7 = x + y$	$x + y$	OR	x or y
$F_8 = (x + y)'$	$x \downarrow y$	NOR	Not-OR
$F_9 = xy + x'y'$	$(x \oplus y)'$	Equivalence	x equals y
$F_{10} = y'$	y'	Complement	Not y
$F_{11} = x + y'$	$x \subset y$	Implication	If y, then x
$F_{12} = x'$	x'	Complement	Not x
$F_{13} = x' + y$	$x \supset y$	Implication	If x, then y
$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
$F_{15} = 1$		Identity	Binary constant 1

Lecture 3:
Other Logic Operations

- We have learned about F_1 (AND) and F_7 (OR) already. F_{10} and F_{12} are also NOT for variables y and x, respectively.
- Some other functions that are important are:
  - F_8: $(x + y)'$ called NOR: NOT-OR

  - F_{14}: $(x.y)'$ called NAND: NOT-AND

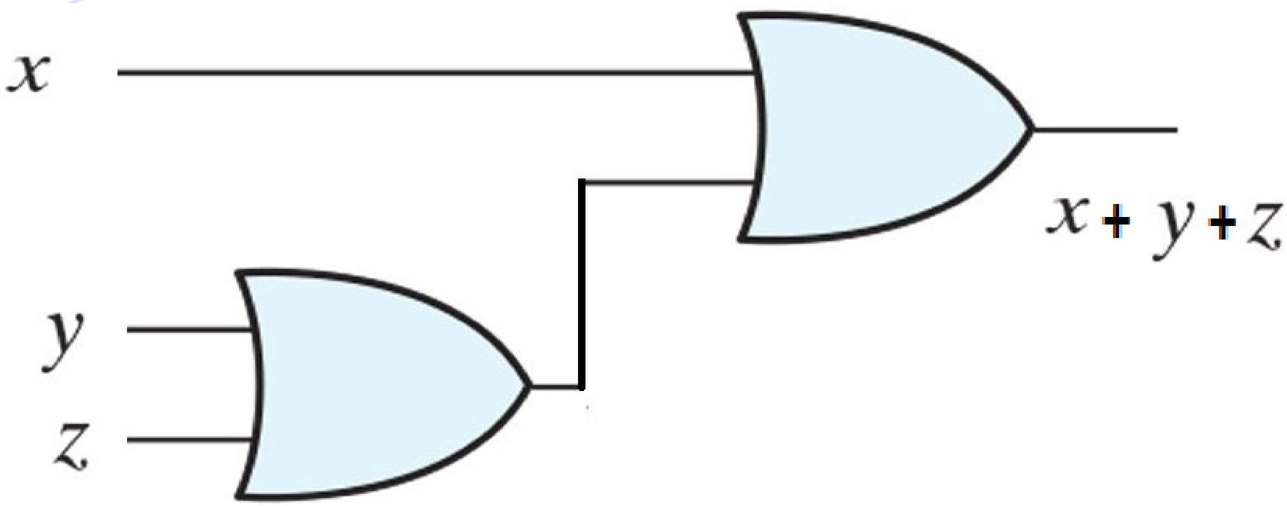


  - F_6: $xy' + x'y$ called Exclusive OR: XOR

  - F_9: $xy + x'y'$ called X-NC .

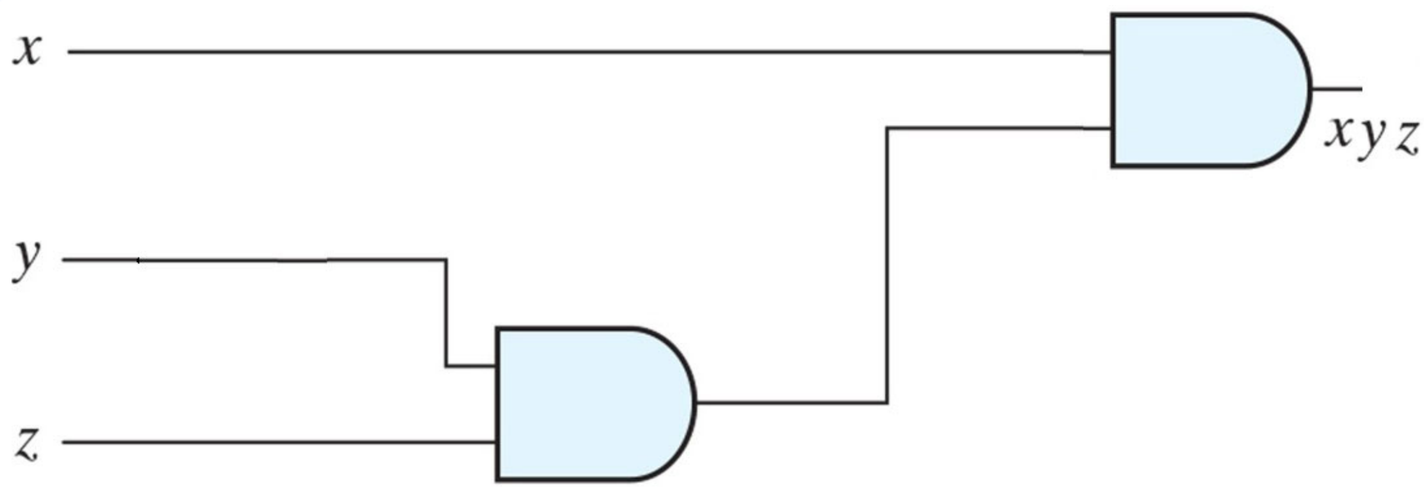
Lecture 3:
Multiple Input Gates: OR

- AND gates and OR gates are commutative and associative, i.e., $(x + y) + z = x + (y + z)$. So, a 3-input OR gate can be made from two OR gates with two inputs each.



Lecture 3:
Multiple Input Gates: AND

- Similarly a 3-input AND gate can be written with no ambiguity as $(x.y).z$ or $x.(y.z)$.
- So, it can be implemented using two 2-input AND gates:



Lecture 3:
Multiple Input Gates: NOR and NAND

Université Concordia University
Department of Electrical & Computer Engineering

- NAND and NOR operations are not associative, e.g., for NOR gates

$$(x \downarrow y) \downarrow z = [(x+y)' + z]' = (x+y)z' = xz' + yz'$$

and

$$x \downarrow (y \downarrow z) = [x + (y+z)']' = x'(y+z) = x'y + x'z$$

So,

$$(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$$

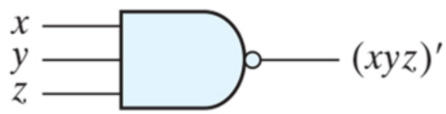
- To avoid this difficulty NAND and NOR gates are implemented using complemented (inverted) AND and OR gates, i.e.,

$$x \uparrow y \uparrow z = (xyz)'$$
$$x \downarrow y \downarrow z = (x+y+z)'$$

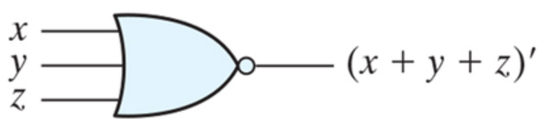
Lecture 3:
Multiple Input Gates: NOR and NAND

Université Concordia University
Department of Electrical & Computer Engineering

- 3-input NAND gate:



- 3-input NOR gate:

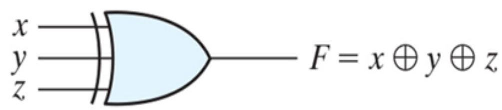


Lecture 3:
Multiple Input Gates: XOR

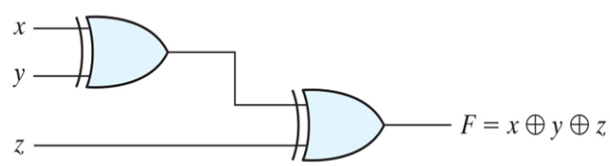
Université Concordia University
Department of Electrical & Computer Engineering

- XOR and X-NOR (equivalence) are both commutative and associative. So, they can be extended to more than two inputs.

- 3-input XOR gate:
  - Symbol



  - Implementation



Lecture 3: Knowledge Check

Université Concordia University
Department of Electrical & Computer Engineering

- **Question 1:** Implement the function for the following truth table:

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

- **Question 2:** Complement of $F = x'y + y'x$ is:

a) $F' = (x'+y')(x + y)$ b) $F' = xy + x'y'$ c) $F' = x'y + xy$ b) None

Question 3: Derive the truth table of:

$$F = x'y'z + xy'z' + x'yz' + xyz$$