1007ICT / 1807ICT / 7611ICT
Computer Systems & Networks

3A. Digital Logic an Circuits

Dr. Sven Venema

Dr. Vallipuram Muthukkumarasamy

Last Section: Data Representation

Topics Covered:

- Representing binary integers
- Conversion al
- Hexadecimal and octal r tions
- Binary number operatio
- One's complement and two's complement
- Representing characters, images and audio

Lecture Content

- Learning objectives
- Digital logic, Basic logic gates, Boolean algebra
- Combinatorial logic gates

Learning Objectives

At the end of this lecture you will have:

- Gained an understanding of basic logic gates
- Learnt the truth tables associated with the basic logic gates
- Gained an understanding of combinatorial logic gates
- Learnt the truth tables associated with combinatorial logic gates

Digital Logic (Section 2.2)

- All digital computers are built from a set of low level digital logic switches or ***Logic Gates***.
- Gates operate on binary signals that only have one of two values:

  - Signals from 0 to 2 volts is used to represent a binary 0 (**OFF**)
  - Signals f present a binary 1 (**ON**)

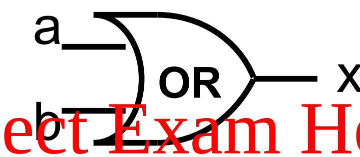
  - Signals ent an invalid state
- Three basic logic fun t can be applied to binary signals:

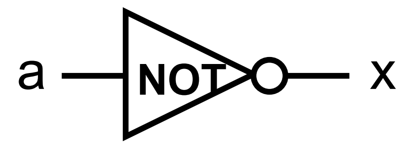
  - AND: output true if ALL inputs are true
  - OR: output true if ANY input is true
  - NOT: output is the inverse of the input
- More complex functions can be built from these three basic gates

Basic Logic Gates (Section 2.4)

Name	**AND**	**OR**	**NOT**

Symbol





Boolean expression

AND: x =

OR: b

NOT: $x = \overline{a}$

Truth Table

AND:

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

OR:

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

NOT:

A	X
0	1
1	0

Boolean Algebra

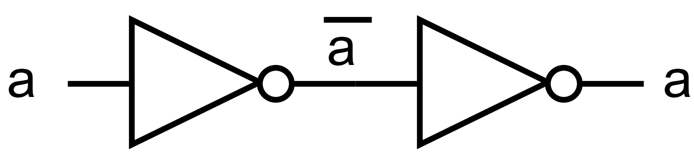
- There is a basic set of rules about combining simple binary functions.

OR

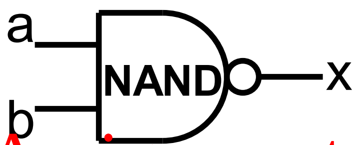
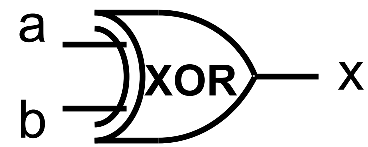
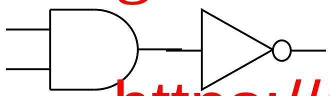
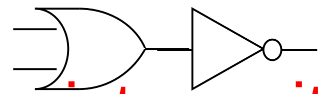
AND

- x OR
- x OR
- x OR $x = x$
- x OR $\overline{x} = 1$
- $\overline{(\overline{x})} = x$

- ND $0 = 0$
- D $1 = x$
- D $x = x$
- x AND $\overline{x} = 0$



Combinatorial Logic Gates

Name	**NAND**	**NOR**	**XOR**
Symbol			
Equivalent			Next Slide
Boolean expression	$x = \overline{a \text{ AND } b}$		x = a XOR b

Truth Table

NAND

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

NOR

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

XOR

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Boolean Algebra - 2

- This second set of rules are more powerful.

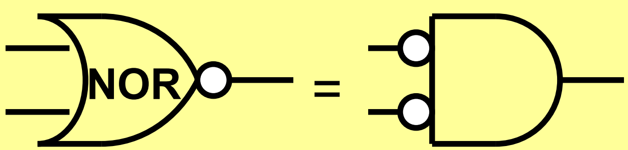
OR - form

$$\overline{(x \text{ OR } y)} = \overline{x} \text{ AND } \overline{y}$$

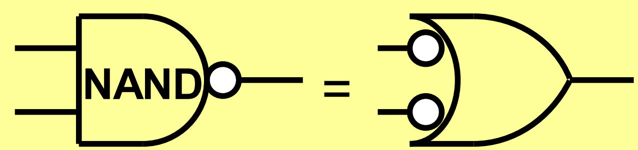
AND - form

$$\overline{(x \text{ AND } y)} = \overline{x} \text{ OR } \overline{y}$$

OR - form

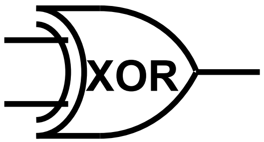


AND - form



DeMorgan's Theorem

The eXclusive-OR Gate (XOR)

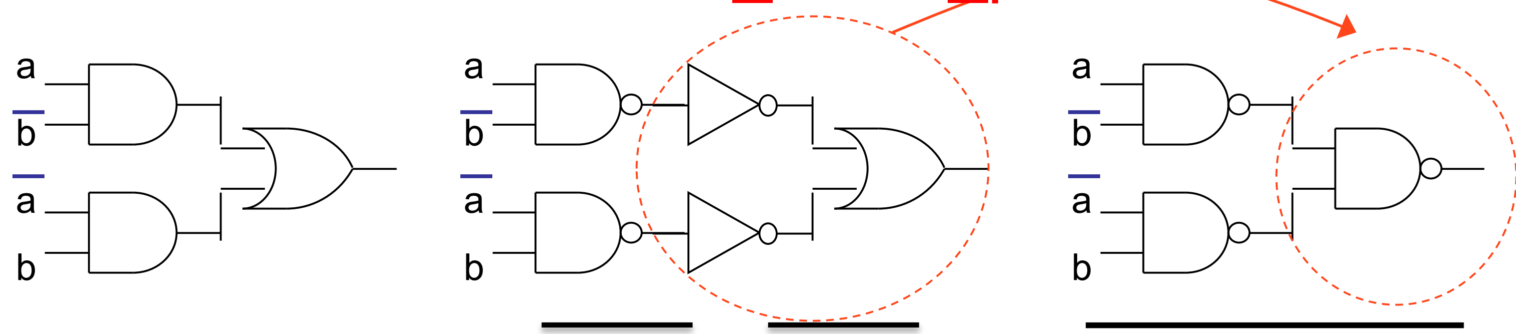


A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

- Looking at the truth table we see that the XOR function can be described as:
  - $x = (a \text{ AND } \overline{b}) \text{ OR } (\overline{a} \text{ AND } b)$
  - $x = a \text{ XOR } b$

- Thi ilt in 3 ways:

emorgan's Theorem



$x = (a \text{ AND } \overline{b}) \text{ OR } (\overline{a} \text{ AND } b)$ $x = \overline{\overline{(a \text{ AND } \overline{b})}} \text{ OR } \overline{\overline{(\overline{a} \text{ AND } b)}}$ $x = \overline{\overline{(a \text{ AND } \overline{b})} \text{ AND } \overline{(\overline{a} \text{ AND } b)}}$

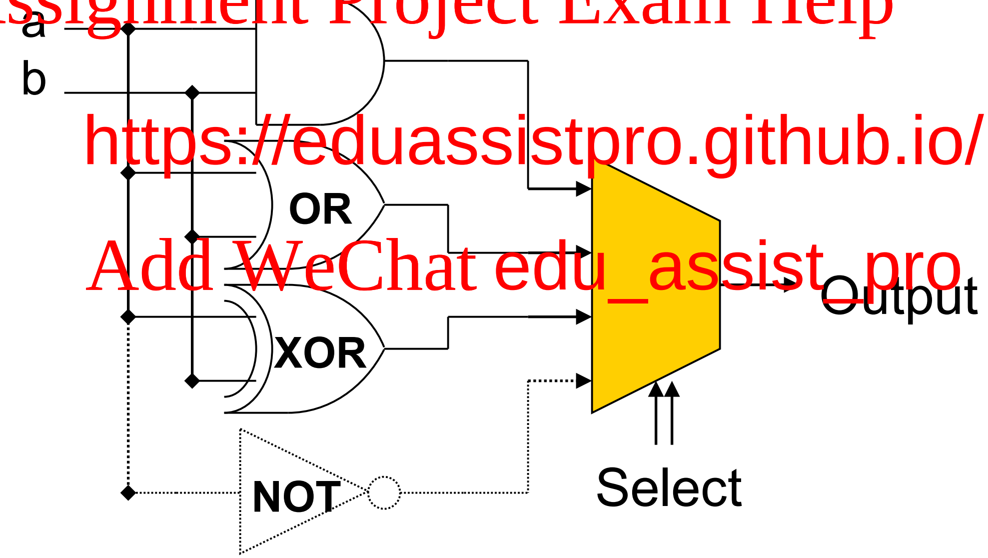

HEF4011BP
BB220 21
U+G06286


CD4081B-MIL
TEXAS INSTRUMENTS


Toshiba TC4071BP
1·A

Logic Unit

- Let's try to create a "programmable" logic unit that permits us to apply a predefined logic function to a given set of inputs.



- We need a function that lets us select what operation to perform

Summary

Have considered:

- Operation of basic logic gates
- Combinatorial logic gates, Truth tables

Next….

- Logic unit, Selection logic, Decoder logic
- Multiplexing and de ing