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

Assignment Project Exam Help

https://powcoder.com

3A. Digital Logic and Digital Circuits

Add WeChat powcoder

Dr. Sven Venema

Dr. Vallipuram Muthukkumarasamy

Last Section: Data Representation

Topics Covered:

- Representing binary integers
- Conversion from binary to decimal
- Hexadecimal and octal representations
- Binary number operations
- One’s complement and two’s complement
- Representing characters, images and audio

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Lecture Content

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

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

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

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Digital Logic (Section 2.2)

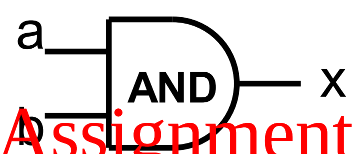
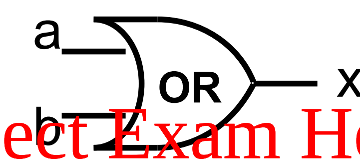
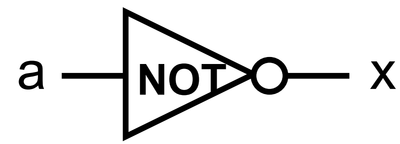
- All digital computers are built from a set of low level digital logic switches or ***Logic Gates***.
- Gates operate on binary signals that only have one of two values:
  - Signals from 0 to 2 volts is used to represent a binary 0 (**OFF**)
  - Signals from 3 to 5 volts is used to represent a binary 1 (**ON**)
  - Signals between 2 and 3 volts represent an invalid state

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

- Three basic logic functions that 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

© Ruben Gonzalez. Revised and updated by Sven Venema, Vallipuram Muthukkumarasamy, and Wee Lum Tan

Basic Logic Gates (Section 2.4)

Name	**AND**	**OR**	**NOT**
Symbol			
Boolean expression	x = a AND b	x = a OR b	$x = \overline{a}$

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

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

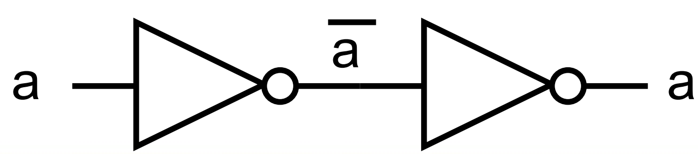
© Ruben Gonzalez. Revised and updated by Sven Venema, Vallipuram Muthukkumarasamy, and Wee Lum Tan

Boolean Algebra

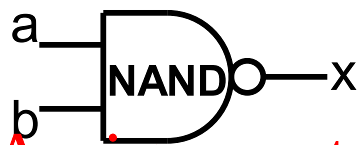
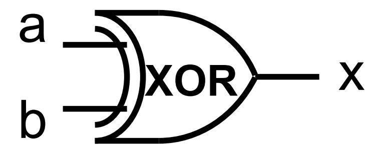
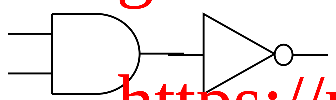
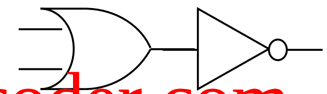
- There is a basic set of rules about combining simple binary functions.

OR	***AND***
• x OR $0 = x$	• x AND $0 = 0$
• x OR $1 = 1$	• x AND $1 = x$
• x OR $x = x$	• x AND $x = x$
• x OR $\overline{x} = 1$	• x AND $\overline{x} = 0$
• $\overline{(\overline{x})} = x$	

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder



Combinatorial Logic Gates

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

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

© Ruben Gonzalez. Revised and updated by Sven Venema, Vallipuram Muthukkumarasamy, and Wee Lum Tan

Boolean Algebra - 2

- This second set of rules are more powerful.

OR - form

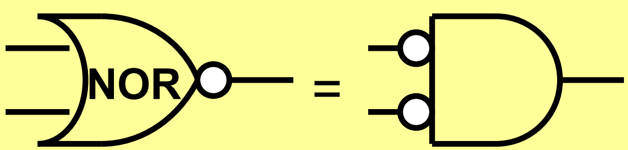
$\overline{(x \text{ OR } y)} = \overline{x} \text{ AND } \overline{y}$

AND - form

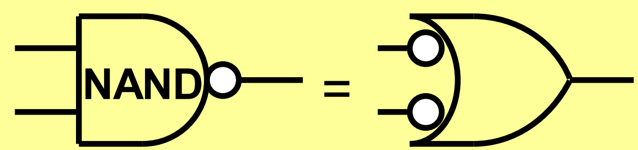
$\overline{(x \text{ AND } y)} = \overline{x} \text{ OR } \overline{y}$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

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

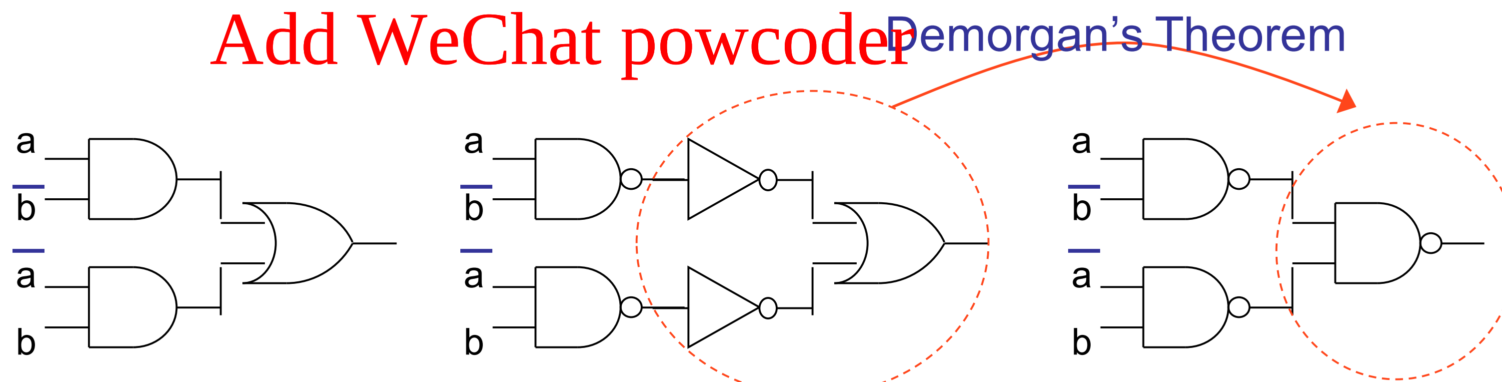
- Looking at the truth table we see that the XOR function can be described as:
  - x = (a AND $\bar{b}$) OR ($\bar{a}$ AND b)
  - x = a XOR b

Assignment Project Exam Help

https://powcoder.com

- This function can be built in 3 ways:

Add WeChat powcoder

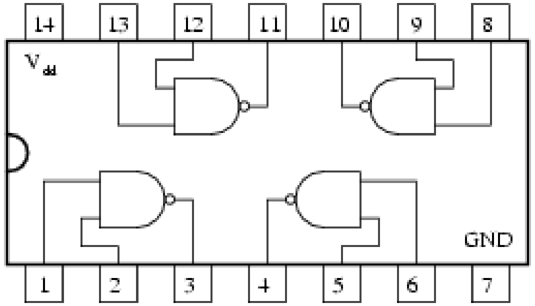


x = (a AND $\bar{b}$) OR ($\bar{a}$ AND b)

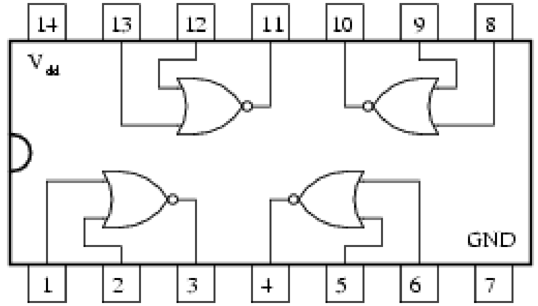
x = $\overline{\overline{(a \text{ AND } \bar{b})}}$ OR $\overline{\overline{(\bar{a} \text{ AND } b)}}$

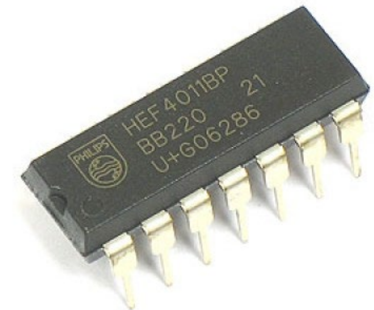
x = $\overline{\overline{(a \text{ AND } \bar{b})} \text{ AND } \overline{(\bar{a} \text{ AND } b)}}$

4011
Quad NAND gate

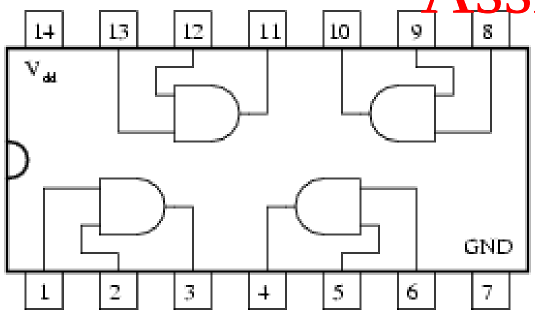


4001
Quad NOR gate

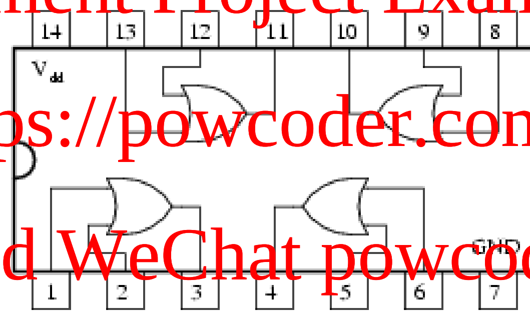


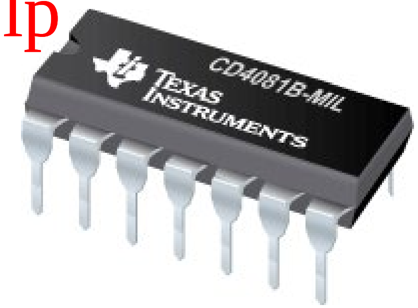


4081
Quad AND gate



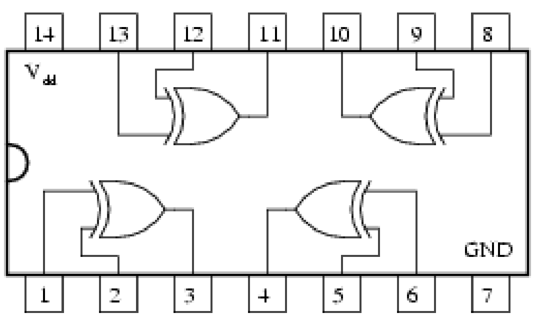
4071
Quad OR gate



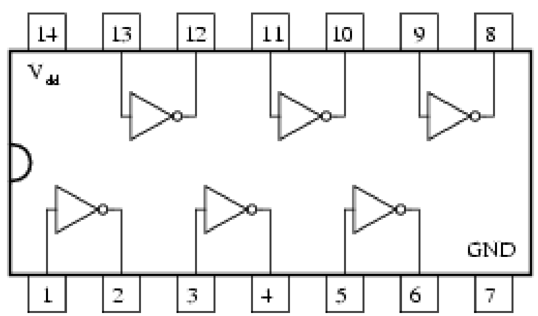


Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

4070
Quad XOR gate



4069
Hex inverter

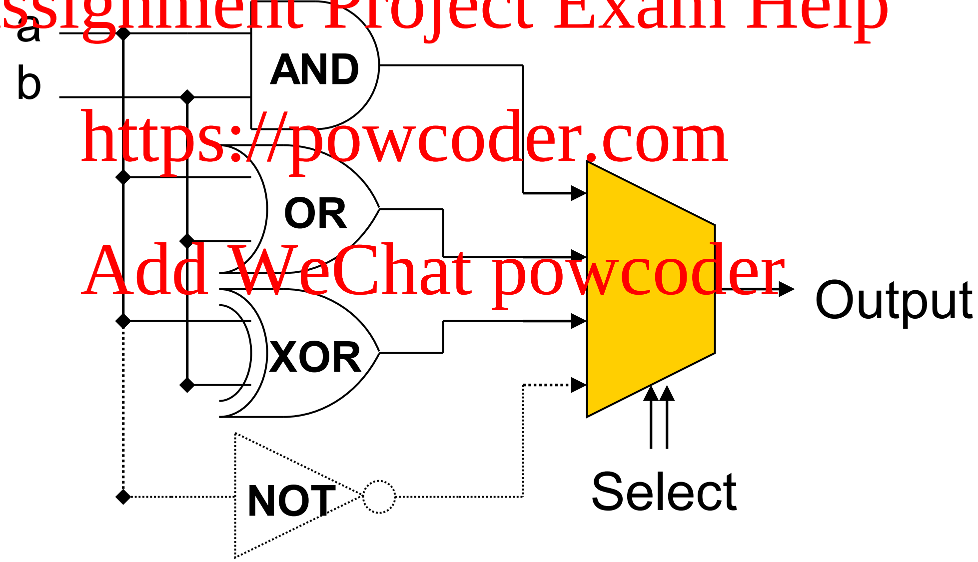


© Ruben Gonzalez. Revised and updated by Sven Venema, Vallipuram Muthukkumarasamy, and Wee Lum Tan

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.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder



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

© Ruben Gonzalez. Revised and updated by Sven Venema, Vallipuram Muthukkumarasamy, and Wee Lum Tan

Summary

Have considered:

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

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Next….

- Logic unit, Selection logic, Decoder logic
- Multiplexing and demultiplexing

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder