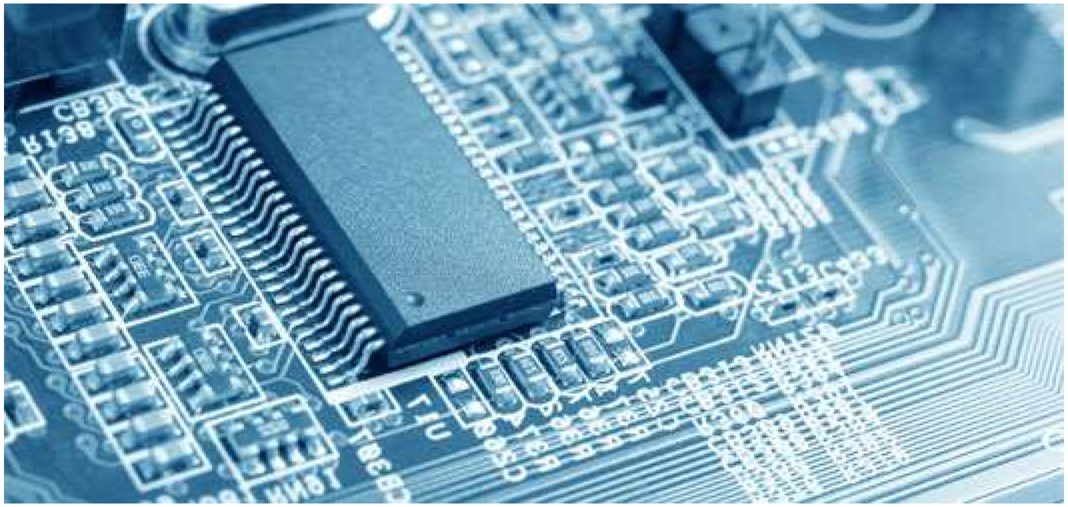
NAZARBAYEV UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY



ROBT206 – Microcontrollers with Lab

Lecture 9 – Additional Gates and Circuits

6 February, 2018

Topics

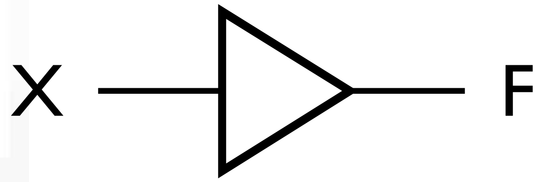
Today's Topics
Other Gate Types, NAND and NOR
Exclusive-OR Operator and Gates
High-Impedance Outputs

Other Gate Types

- Basic gates are ***AND***, ***OR***, ***NOT***.
- Why do we need other gates?
  - Implementation feasibility and low cost
  - Power in implementing Boolean functions
  - Convenient conceptual representation
- Gate classifications
  - Primitive gate - a gate that can be described using a single primitive operation type (AND or OR) plus an optional inversion(s).
  - Complex gate - a gate that requires more than one primitive operation type for its description
- Primitive gates will be covered first

Buffer

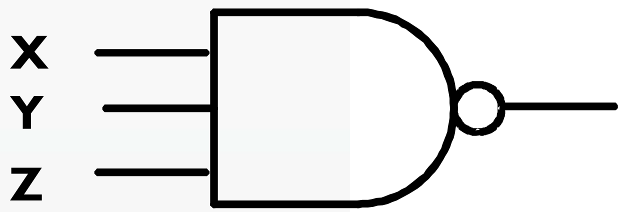
- A buffer is a gate with the function F = X:



- In terms of Boolean function, a buffer is the same as a connection!
- So why use it?
  - A buffer is an electronic amplifier used to improve circuit voltage levels and increase the speed of circuit operation.

NAND Gate

- The basic NAND gate has the following symbol, illustrated for three inputs:
  - AND-Invert (NAND)

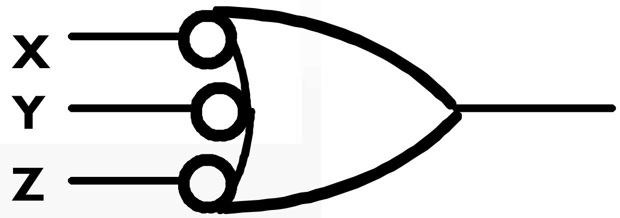


$$F(X,Y,Z) = \overline{X \cdot Y \cdot Z}$$

- NAND represents <u>NOT</u> <u>AND</u>, i. e., the AND function with a NOT applied. The symbol shown is an AND-Invert. The small circle ("bubble") represents the invert function.

NAND Gates

- Applying DeMorgan's Law gives Invert-OR (NAND)



$$F(X,Y,Z) = \overline{X} + \overline{Y} + \overline{Z}$$

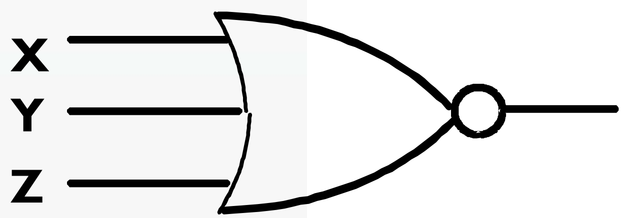
- This NAND symbol is called **Invert-OR**, since inputs are inverted and then **ORed** together.
- **AND-Invert** and **Invert-OR** both represent the NAND gate. Having both makes visualization of circuit function easier.
- A NAND gate with one input degenerates to an inverter.

NAND Gates

- The NAND gate is the natural implementation for CMOS (Complementary metal–oxide–semiconductor) technology in terms of chip area and speed.
- *Universal gate* - a gate type that can implement any Boolean function.
- NAND usually does not have a operation symbol defined since
  - the NAND operation is not associative, and
  - we have difficulty dealing with non-associative mathematics!

NOR Gate

- The basic NOR gate has the following symbol, illustrated for three inputs:
  - OR-Invert (NOR)

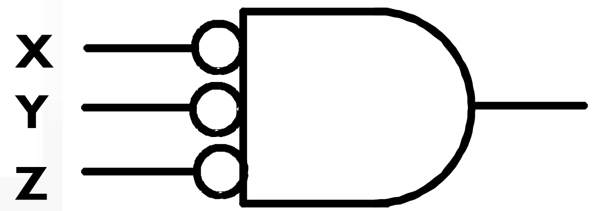


$$F(X,Y,Z) = \overline{X+Y+Z}$$

- NOR represents <u>NOT - OR</u>, i. e., the OR function with a NOT applied. The symbol shown is an OR-Invert.
- The small circle ("bubble") represents the invert function.

NOR Gate

- Applying DeMorgan's Law gives Invert-AND (NOR)



- This NOR symbol is called Invert-AND, since inputs are inverted and then ANDed together.
- OR-Invert and Invert-AND both represent the NOR gate. Having both makes visualization of circuit function easier.
- A NOR gate with one input degenerates to an inverter.

NOR Gate

- The NOR gate is a natural implementation for some technologies other than CMOS in terms of chip area and speed.
- The NOR gate is a universal gate
- NOR usually does not have a defined operation symbol since
  - the NOR operation is not associative, and
  - we have difficulty dealing with non-associative mathematics!

Exclusive OR/ Exclusive NOR

- The *eXclusive OR* (*XOR*) function is an important Boolean function used extensively in logic circuits.
- The *eXclusive NOR* function is the complement of the XOR function

Exclusive OR/ Exclusive NOR

- Uses for the XOR and XNORs gate include:
  - Adders/subtractors/multipliers
  - Counters/incrementers/decrementers
  - Parity generators/checkers
- Definitions
  - The XOR function is: $\mathbf{X} \oplus \mathbf{Y} = \mathbf{X}\overline{\mathbf{Y}} + \overline{\mathbf{X}}\mathbf{Y}$
  - The eXclusive NOR (XNOR) function, otherwise known as *equivalence* is: $\overline{\mathbf{X} \oplus \mathbf{Y}} = \mathbf{X}\mathbf{Y} + \overline{\mathbf{X}}\,\overline{\mathbf{Y}}$
- Strictly speaking, XOR and XNOR gates do no exist for more than two inputs. Instead, they are replaced by odd and even functions.

Truth Tables for XOR/XNOR

- Operator Rules: XOR

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

XNOR

X	Y	$\overline{(X \oplus Y)}$ or $X \equiv Y$
0	0	1
0	1	0
1	0	0
1	1	1

- The XOR function means:

  X OR Y, but NOT BOTH

XOR/XNOR

- The XOR function can be extended to 3 or more variables. For more than 2 variables, it is called an *odd function* or *modulo 2 sum* (*Mod 2 sum*), not an XOR:

$$X \oplus Y \oplus Z = \overline{X}\,\overline{Y}Z + \overline{X}Y\overline{Z} + X\overline{Y}\,\overline{Z} + XYZ$$

- The complement of the odd function is the even function.
- The XOR identities:

$$X \oplus 0 = X \qquad X \oplus 1 = \overline{X}$$

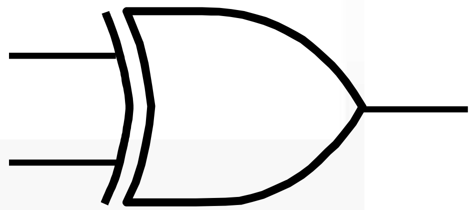
$$X \oplus X = 0 \qquad X \oplus \overline{X} = 1$$

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

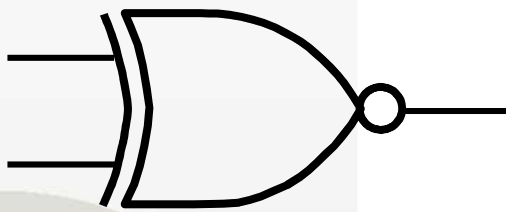
$$(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z) = X \oplus Y \oplus Z$$

Symbols For XOR and XNOR

- XOR symbol:



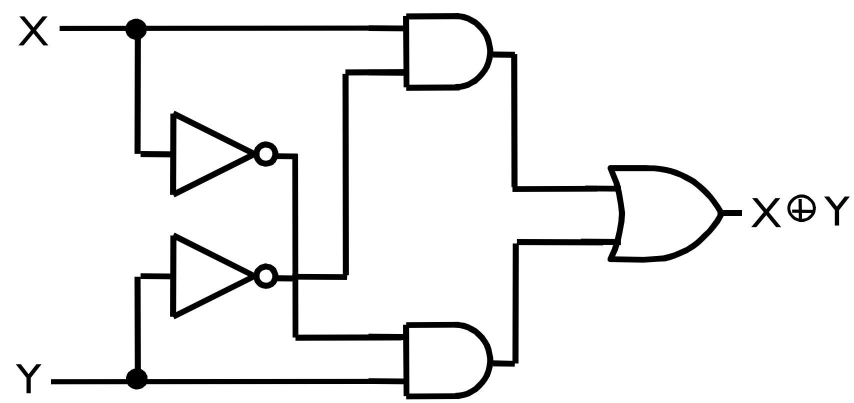
- XNOR symbol:



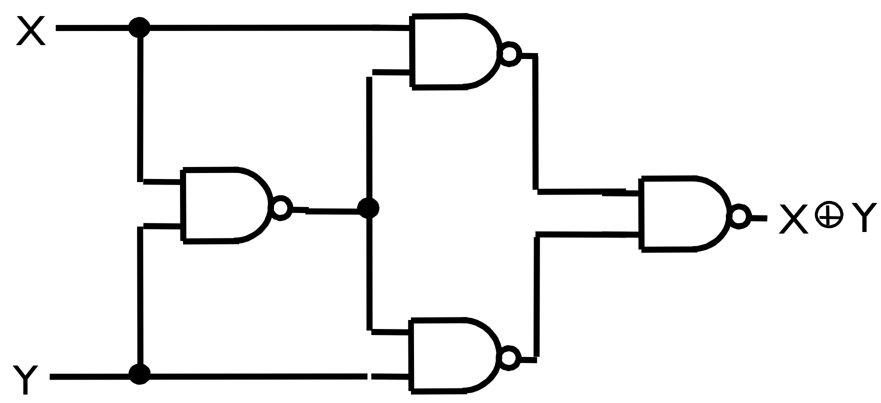
- Shaped symbols exist only for two inputs

XOR Implementations

- The simple SOP implementation uses the following structure:



- A NAND only implementation is:



Hi-Impedance Outputs

- Logic gates introduced thus far
  - have 1 and 0 output values,
  - cannot have their outputs connected together, and
  - transmit signals on connections in only one direction.

- Three-state logic adds a third logic value, Hi-Impedance (Hi-Z), giving three states: 0, 1, and Hi-Z on the outputs.

Hi-Impedance Outputs

- What is a Hi-Z value?
  - The Hi-Z value behaves as an open circuit
  - This means that, looking back into the circuit, the output appears to be disconnected.
  - It is as if a switch between the internal circuitry and the output has been opened.
- Hi-Z may appear on the output of any gate, but we restrict gates to:
  - a 3-state buffer, or
  - Optional: a transmission gate

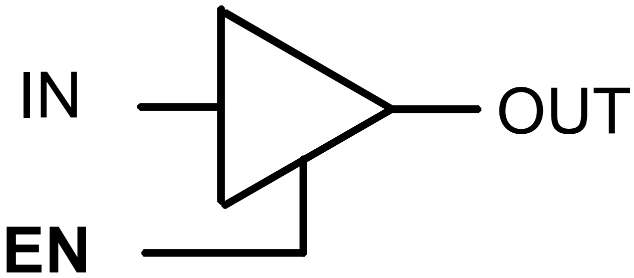
each of which has one data input and one control input.

The 3-State Buffer

- For the symbol and truth table, IN is the **data input**, and EN, the **control input**.
- For EN = 0, regardless of the value on IN (denoted by X), the output value is Hi-Z.
- For EN = 1, the output value follows the input value.
- Variations:
  - Data input, IN, can be inverted
  - Control input, EN, can be inverted

by addition of "bubbles" to signals.

Symbol



Truth Table

EN	IN	OUT
0	X	Hi-Z
1	0	0
1	1	1

Any Questions?