ShanghaiTech University

EE 115B: Digital Circuits

Fall 2022

Lecture 4

Hengzhao Yang

September 20, 2022

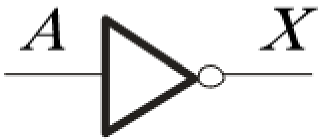
EE115 Analog and *Digital Circuits*

Topic 4: Logic Gates

Prof. Yajun Ha

ShanghaiTech University
Shanghai, China

The Inverter (1)



The inverter performs the Boolean **NOT** operation. When the input is LOW, the output is HIGH; when the input is HIGH, the output is LOW.

Input	Output
A	X
LOW (0)	HIGH (1)
HIGH (1)	LOW(0)

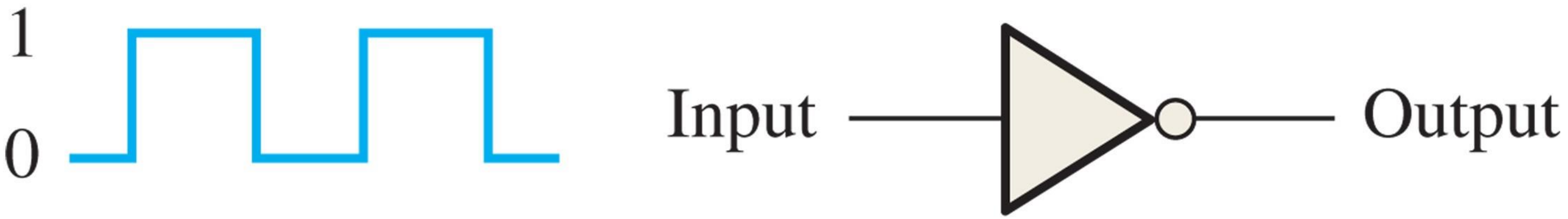
The **NOT** operation (complement) is shown with an overbar. Thus, the Boolean expression for an inverter is $X = \overline{A}$.

FIGURE 3-1 Standard logic symbols for the inverter (ANSI/IEEE Std. 91-1984/Std. 91a-1991).

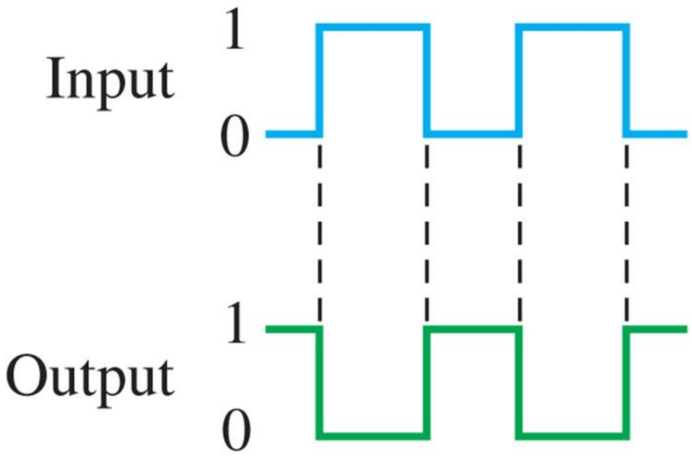
(a) Distinctive shape symbols with negation indicators

(b) Rectangular outline symbols with polarity indicators

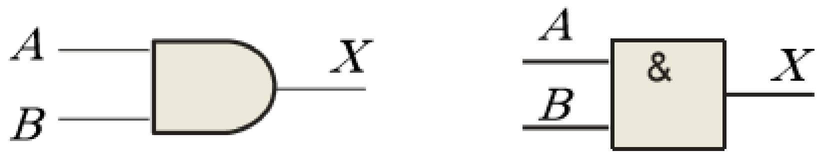
Example: Given the input, plot the output waveform.



Solution: See below for the timing diagram.



The AND Gate (1)



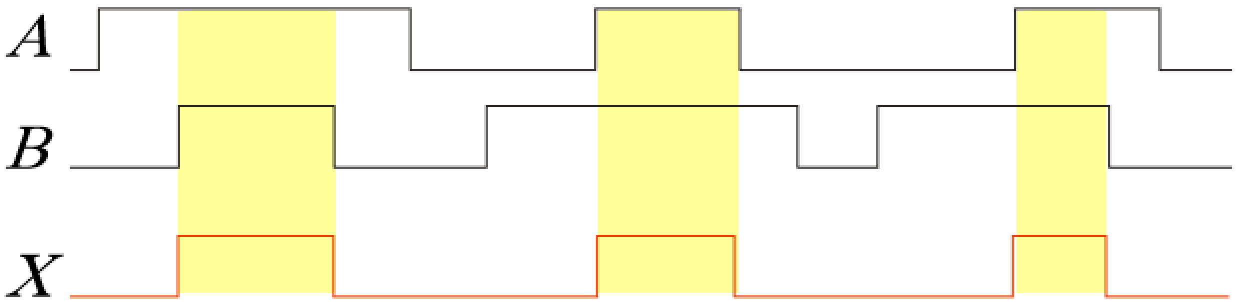
The **AND gate** produces a HIGH output when all inputs are HIGH; otherwise, the output is LOW. For a 2-input gate, the truth table is

Inputs		Output
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

The **AND** operation is usually shown with a dot between the variables but it may be implied (no dot). Thus, the AND operation is written as $X = A \cdot B$ or $X = AB$.

The AND Gate (3)

Example waveforms:



The AND operation is used in computer programming as a selective mask. If you want to retain certain bits of a binary number but reset the other bits to 0, you could set a mask with 1's in the position of the retained bits.

Example If the binary number 10100011 is ANDed with the mask 00001111, what is the result? 00000011

The OR Gate (1)

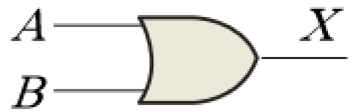
The **OR gate** produces a HIGH output if any input is HIGH; if all inputs are LOW, the output is LOW. For a 2-input gate, the truth table is

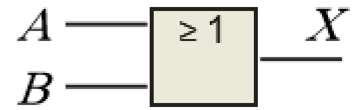
Inputs		Output
A	*B*	*X*
0	0	0
0	1	1
1	0	1
1	1	1

The **OR** operation is shown with a plus sign (+) between the variables. Thus, the OR operation is written as $X = A + B$.

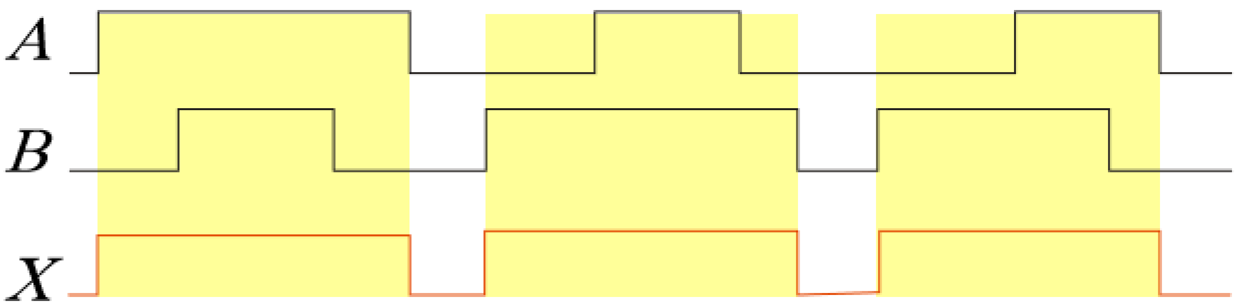
The OR Gate (2)

The OR Gate





Example waveforms:



The OR operation can be used in computer programming to set certain bits of a binary number to 1.

Example ASCII letters have a 1 in the bit 5 position for lower case letters and a 0 in this position for capitals. (Bit positions are numbered from right to left starting with 0.) What will be the result if you OR an ASCII letter with the 8-bit mask 00100000?

Solution The resulting letter will be lower case.

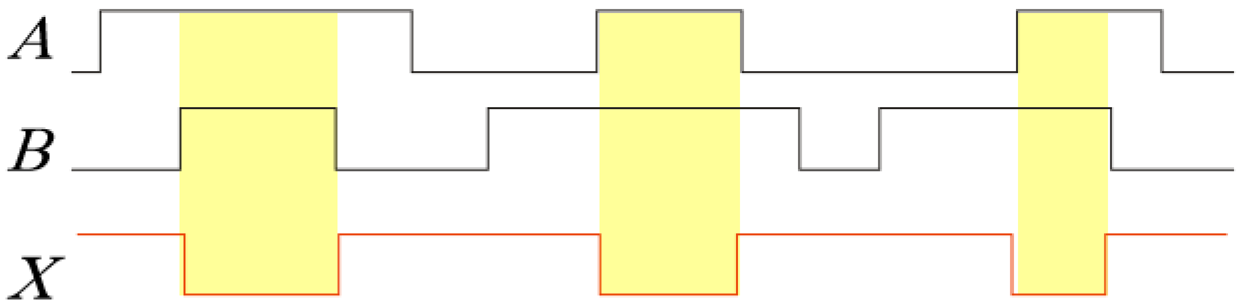
The **NAND gate** produces a LOW output when all inputs are HIGH; otherwise, the output is HIGH. For a 2-input gate, the truth table is

Inputs		Output
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

The **NAND** operation is shown with a dot between the variables and an overbar covering them. Thus, the NAND operation is written as $X = \overline{A \cdot B}$ (Alternatively, $X = \overline{AB}$.)

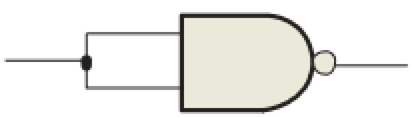
The NAND Gate (2)

Example waveforms:

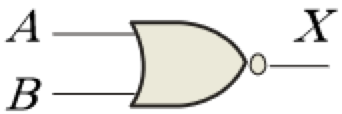


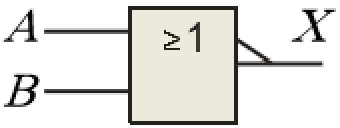
The NAND gate is particularly useful because it is a "universal" gate – all other basic gates can be constructed from NAND gates.

Question How would you connect a 2-input NAND gate to form a basic inverter?



The NOR Gate (1)





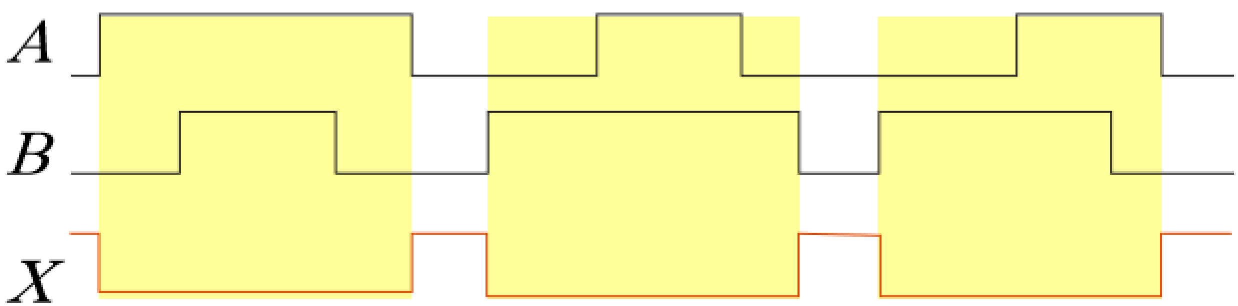
The **NOR gate** produces a LOW output if any input is HIGH; if all inputs are HIGH, the output is LOW. For a 2-input gate, the truth table is

Inputs		Output
A	*B*	*X*
0	0	1
0	1	0
1	0	0
1	1	0

The **NOR** operation is shown with a plus sign (+) between the variables and an overbar covering them. Thus, the NOR operation is written as $X = \overline{A + B}$.

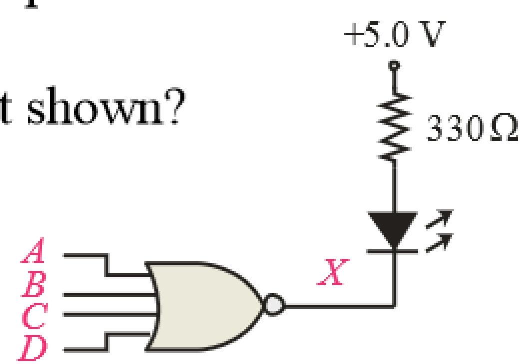
The NOR Gate (2)

Example waveforms:



The NOR operation will produce a LOW if any input is HIGH.

Example When is the LED is ON for the circuit shown?



Solution The LED will be on when any of the four inputs are HIGH.

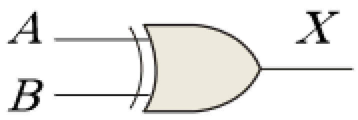
The XOR Gate (1)

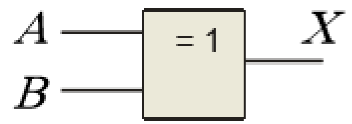
The **XOR gate** produces a HIGH output only when both inputs are at opposite logic levels. The truth table is

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

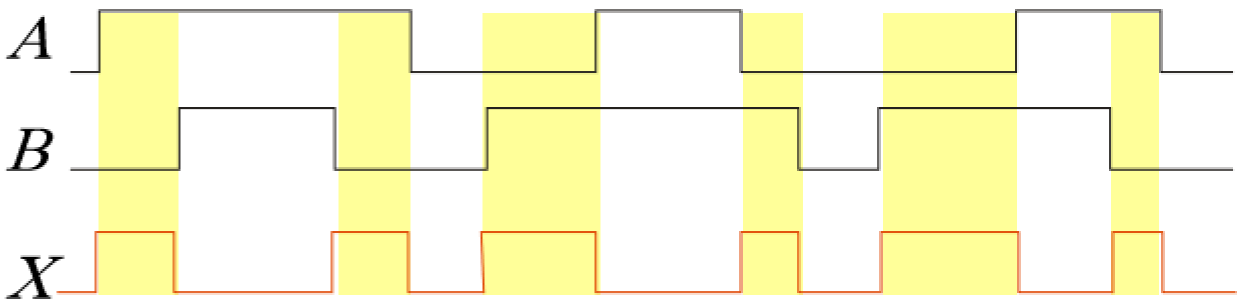
The **XOR** operation is written as $X = \overline{A}B + A\overline{B}$. Alternatively, it can be written with a circled plus sign between the variables as $X = A \oplus B$.

The XOR Gate (2)





Example waveforms:

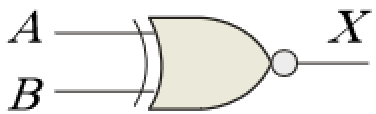


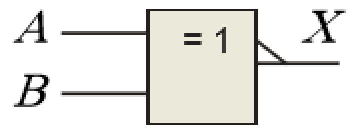
Notice that the XOR gate will produce a HIGH only when exactly one input is HIGH.

Question If the *A* and *B* waveforms are both inverted for the above waveforms, how is the output affected?

There is no change in the output.

The XNOR Gate (1)





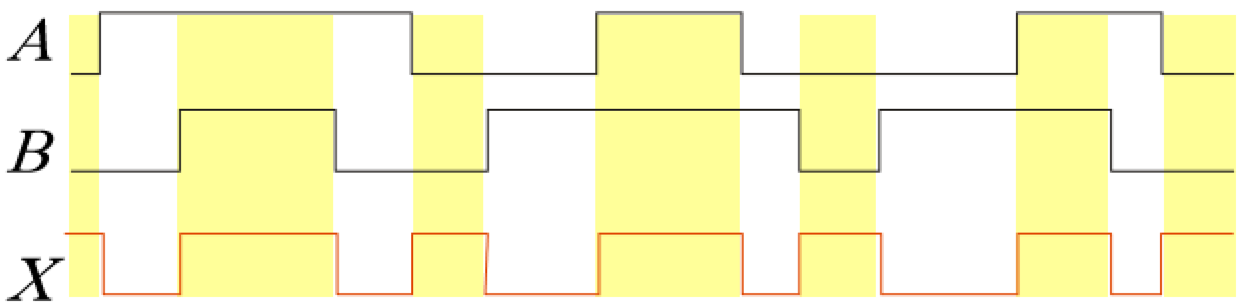
The **XNOR gate** produces a HIGH output only when both inputs are at the same logic level. The truth table is

Inputs		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

The **XNOR** operation shown as $X = \overline{A}\overline{B} + AB$. Alternatively, the XNOR operation can be shown with a circled dot between the variables. Thus, it can be shown as $X = A \odot B$.

The XNOR Gate (2)

Example waveforms:



Notice that the XNOR gate will produce a HIGH when both inputs are the same. This makes it useful for comparison functions.

Question If the *A* waveform is inverted but *B* remains the same, how is the output affected?

The output will be inverted.