JOINT INSTITUTE
交大密西根学院

ECE2700J SU23 RC1

Number Systems, Logic Gates

Wenyue Li
5/20/2024

Numbers Systems

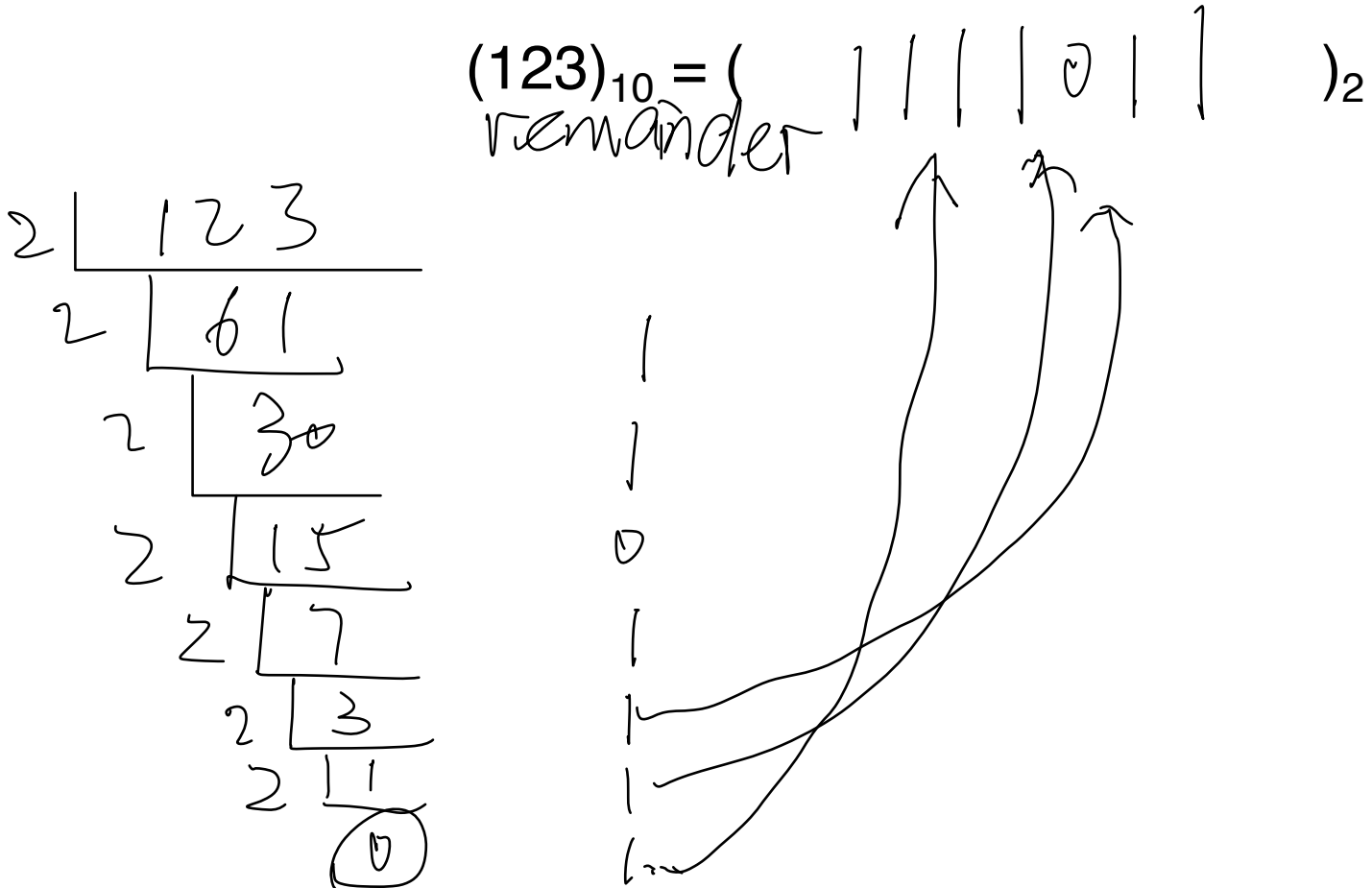
Basic concepts

- Binary: base 2
- Octal: base 8
- Decimal: base 10
- Hexadecimal: base 16

Numbers Systems

Conversion Exercises

1. Basic conversions



Numbers Systems

Conversion Exercises

2. Conversion with decimal points

$(123.45)_{10} = (1111011.01)_2$

$123 = 1111011$.

$0.45 \times 2 = 0.9$ → 0

$0.9 \times 2 = 1.8$ → 1

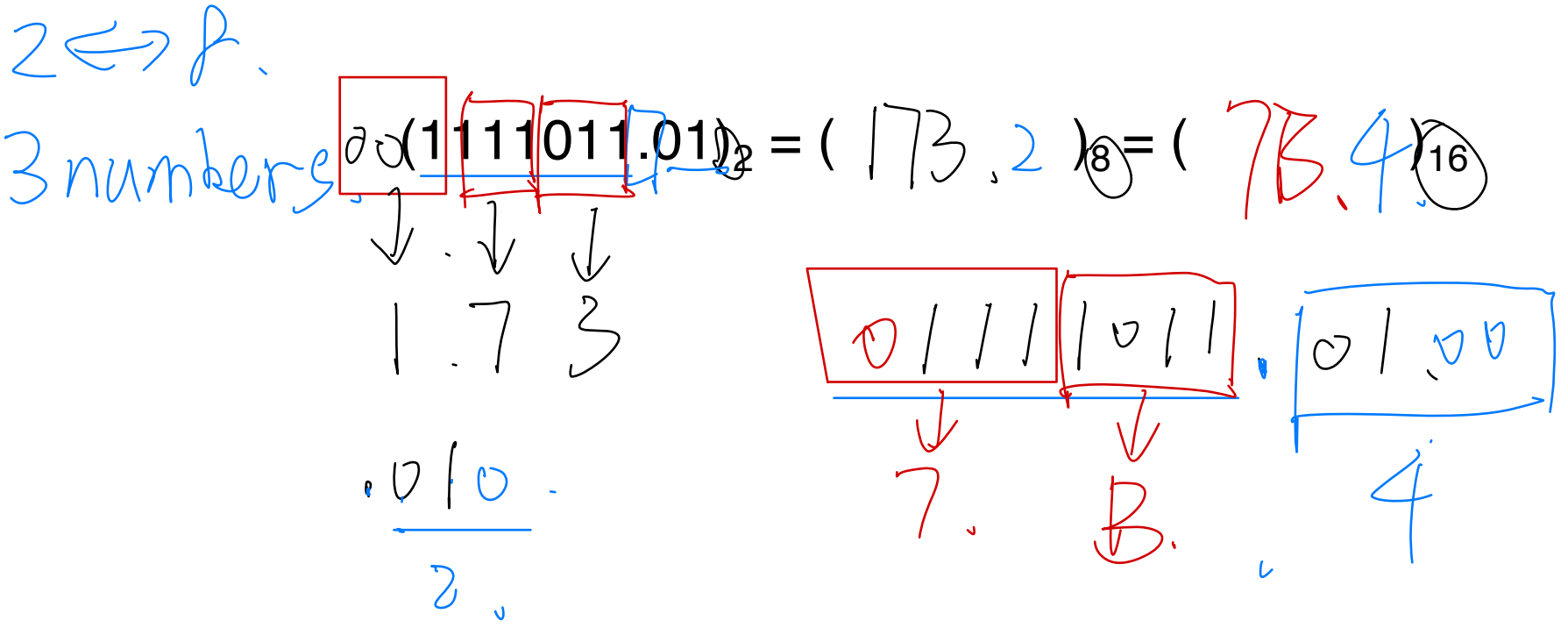
$0.8 \times 2 = 1.6$ → 1

Note: usually we reserve 3 or 4 digits after the decimal point.

Numbers Systems

Conversion Exercises

3. Conversion between binary and octal, binary and hexadecimal



Numbers Systems

Signed Numbers

There're three common ways to represent negative numbers

- 1. Sign-magnitude representation

0 ⟷ positive
1 ⟷ negative .

- 2. 1's complement representation: negate all the bits
- 3. **2's complement representation**: negate all the bits and add 1

Using $(+123)_{10} = (01111011)_2$ as an example

1. $(-123)_{10} = (\quad)_2$
2. $(-123)_{10} = (\quad)_2$
3. $(-123)_{10} = (\quad)_2$

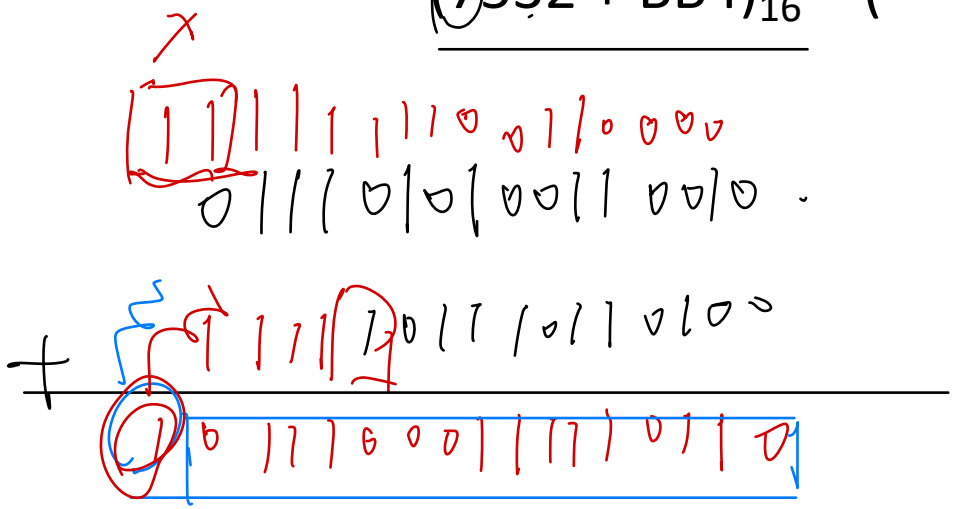
Numbers Systems

Sign extension

Extend the sign bit to the left. In 2's complement representation, we simply repeat the sign bit.

Exercise: assuming we're using 2's complement

$$(7532 + BB4)_{16} = (\qquad\qquad\qquad\qquad)_2$$



Numbers Systems

Overflow

When the result of an operation is too large to be represented in the number of bits available.

Question: what's the range of n-bit 2's complement representation?

1. If the signs of the operands are the same, but the sign of the result is different, then overflow occurs. If the signs of the operands are different, no overflow occurs.

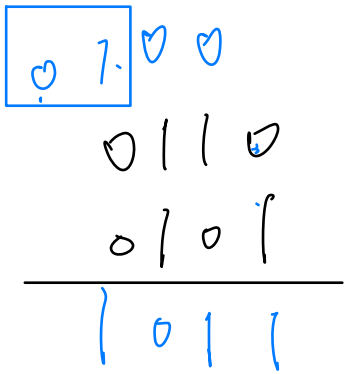
2. If the two most left carriers are different, then overflow occurs.

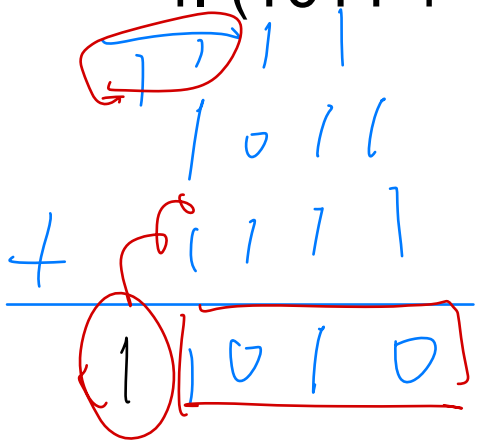
Numbers Systems

Overflow

Exercises: Compare the two calculation. Is there any overflow in the following operations?

1. $(7532 + BB4)_{16} = (70E6)_{16}$ ✗ overflow.
2. $(7532 + 0BB4)_{16} = (80E6)_{16}$ ✓ overflow.
3. $(0110 + 0101)_2 = (\quad)_2$ ✓ overflow.
4. $(1011 + 1111)_2 = (1010)_2$

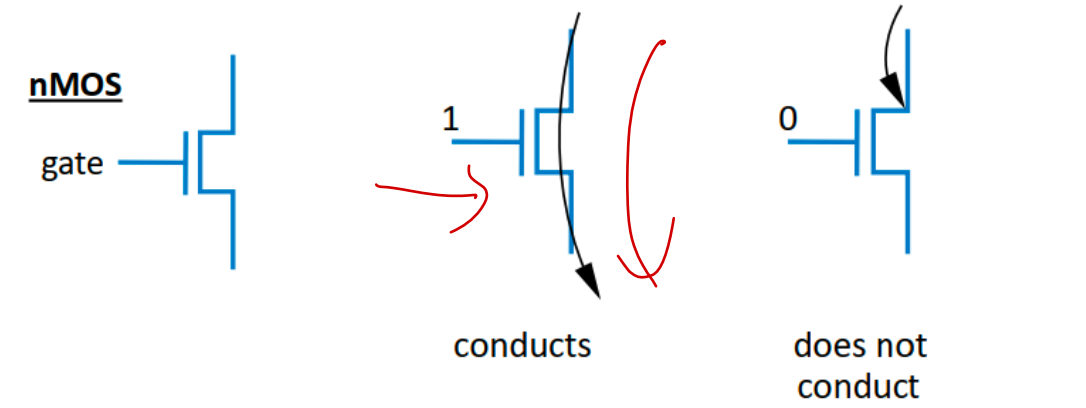


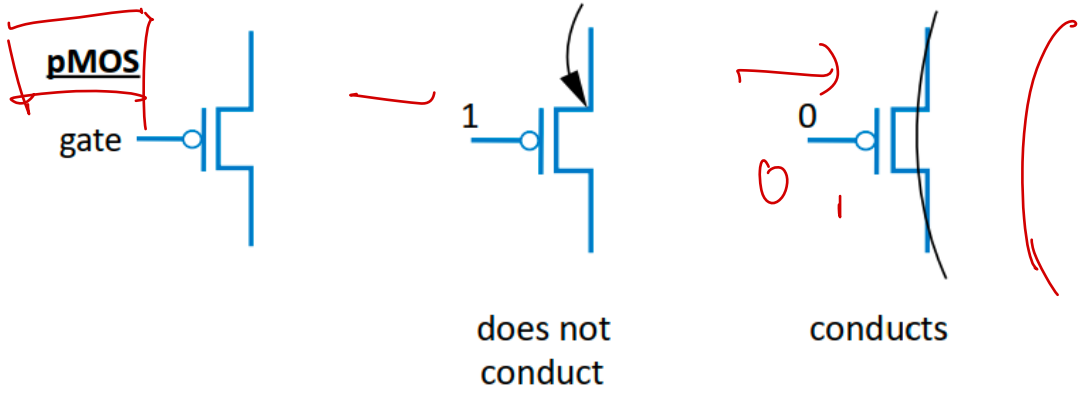


Logic gates

CMOS Transistor

nMOS / pMOS





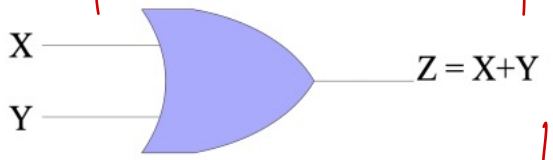
Logic gates

Symbol and truth table of all logical gates: AND, NAND, OR, NOR, XOR, XNOR, NOT, *buffer*

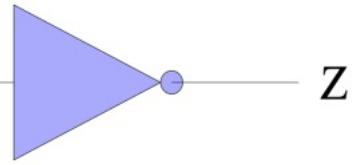
2-input AND gate

all 1 ⇒ 1.

one of input is 1. output will be 1.



2-input OR gate

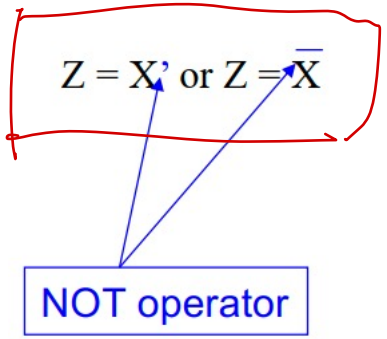


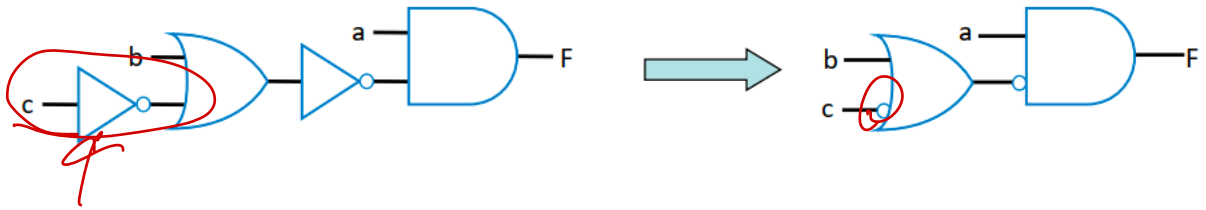
NOT gate/Inverter

- **Precedence of Logic Operations**

NOT > AND > OR

$Z = X'$





Logic gates

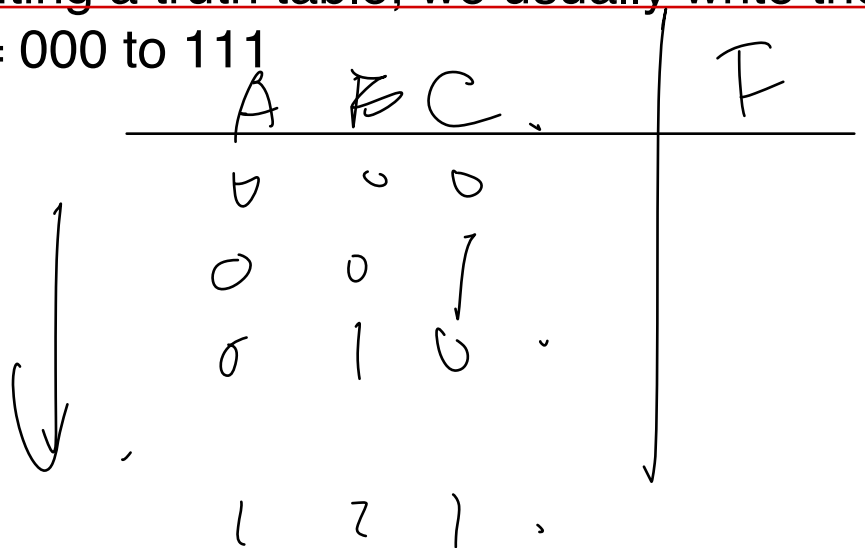
Truth table

Truth table: A table that shows the output of a logic gate for all possible input combinations.

- Number of rows = 2^n, where n is the number of inputs
- When writing a truth table, we usually write the inputs in binary order, i.e. abc = 000 to 111

3 inputs.

8

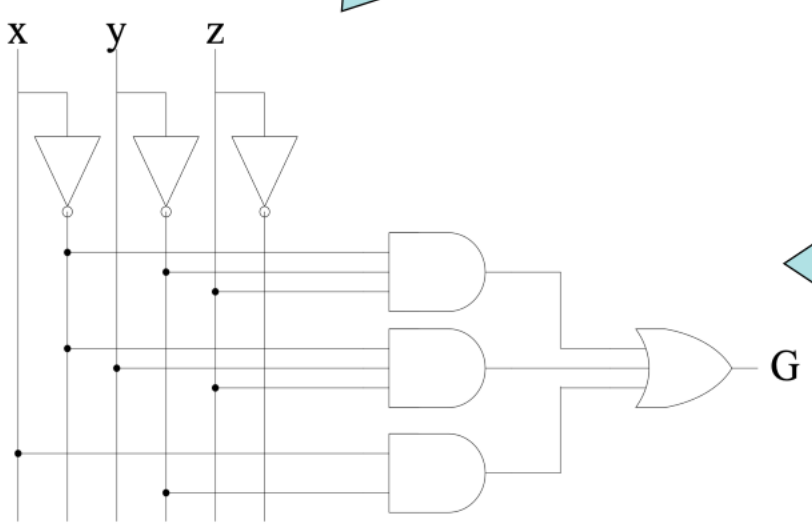


order.

Representations of I/O Relationship

Equation, Truth Table, & Circuit

$$G = x'y'z + x'yz + xy'$$



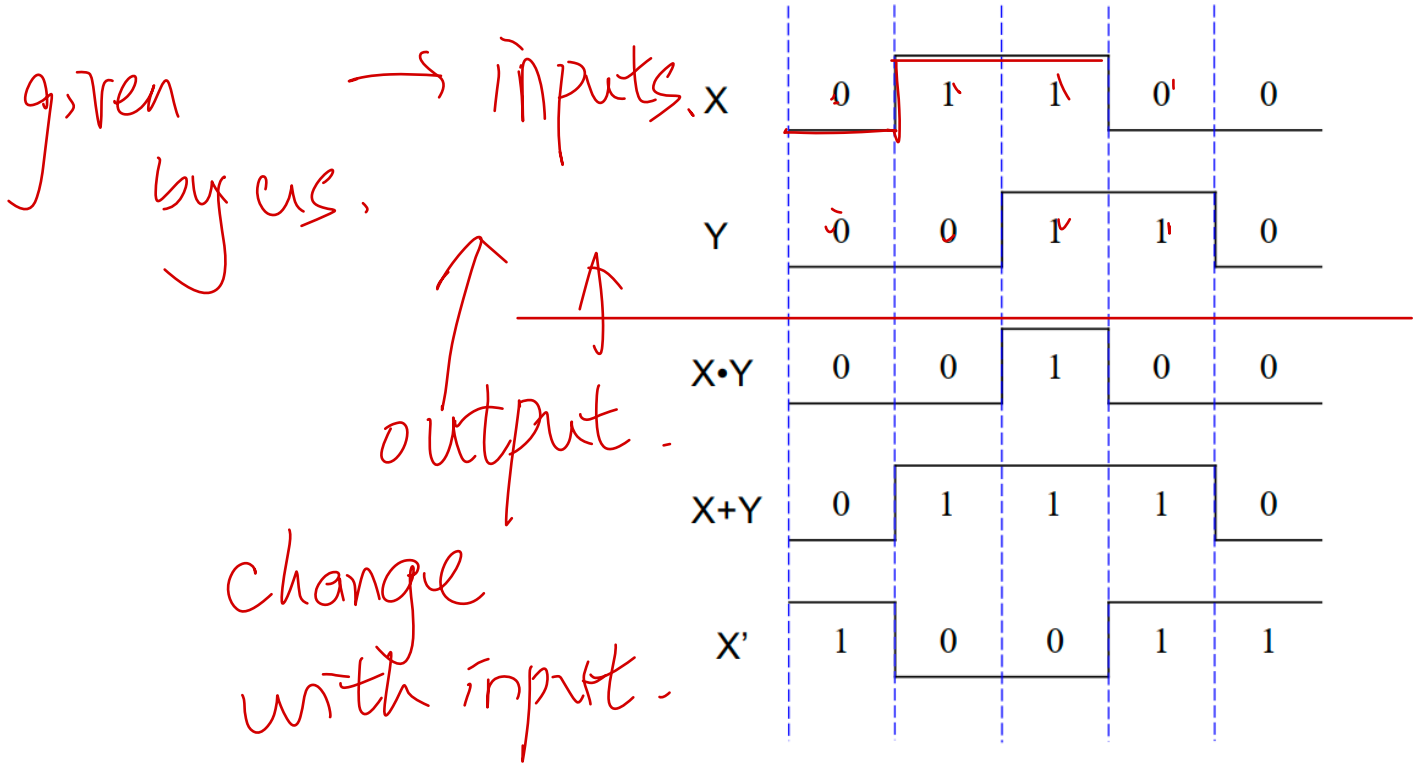
Gate Implementation

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

Truth Table

Timing Diagram

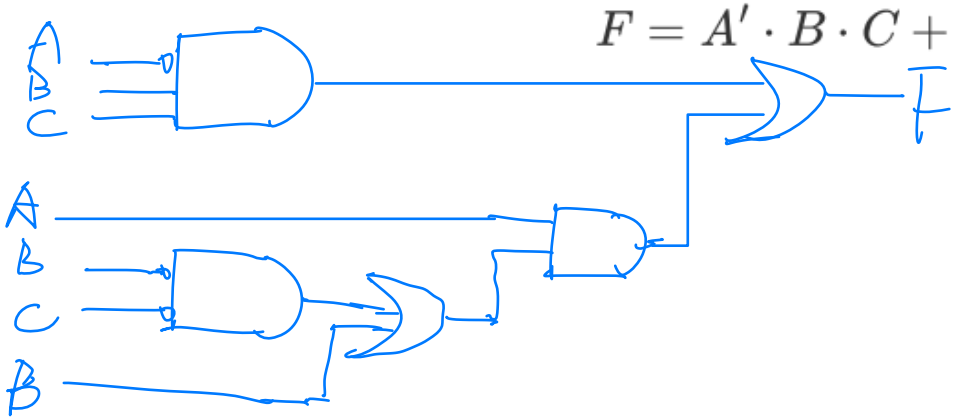
Timing diagram: a diagram that shows the response to changes on a signal in voltage levels with time



Logic gates

Exercise:

1. Conversion between logical equation, logical circuit, and truth table:



$$F = A' \cdot B \cdot C + A \cdot (B' \cdot C' + B)$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

2. Draw the timing diagram of the equation above with the given sequences

1. A = 1, B = 0, C = 1
2. A = 1, B = 1, C = 1
3. A = 0, B = 1, C = 1
4. A = 0, B = 1, C = 0
5. A = 1, B = 1, C = 0