

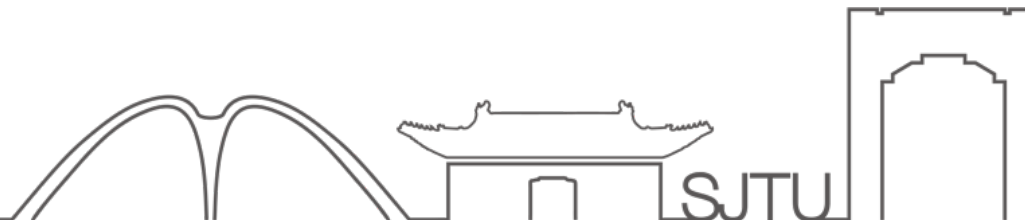


JOINT INSTITUTE
交大密西根学院

ECE2700J SU23 RC1

Number Systems, Logic Gates

Wenyue Li
5/20/2024



JOINT INSTITUTE
交大密西根学院

Numbers Systems

Basic concepts

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

Numbers Systems

Conversion Exercises

1. Basic conversions

$(123)_{10} = (\text{remainder } 11101)_2$

Handwritten conversion of $(123)_{10}$ to binary using the remainder method:

2		123
2		61
2		30
2		15
2		7
2		3
2		1
		0

The remainders, read from bottom to top, are 1, 1, 0, 1, 1, 1, 0. These are written as the binary number 11101_2 .

Numbers Systems

Conversion Exercises

2. Conversion with decimal points

$$\boxed{(123.45)}_{10} = (1111011.011)_2$$

$$13 = 1111011.$$

$$0.45 \times 2 = \boxed{0}9$$

$$0.9 \times 2 = \boxed{1}8$$

$$0.8 \times 2 = \boxed{1}6$$

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

$2 \leftrightarrow 8$

3 numbers

$$00(1111011.01)_2 = (173.2)_8 = (7B.4)_{16}$$

↓ ↓ ↓
1 7 3

00 | 0
—
2

0111 | 1011 | 01.00
↓ ↓ ↓
7 B 4

Numbers Systems

Signed Numbers

There're three common ways to represent negative numbers

- 1. Sign-magnitude representation $0 \leftrightarrow \text{positive}$
- 2. 1's complement representation: negate all the bits $1 \leftrightarrow \text{negative}$
- 3. **2's complement representation**: negate all the bits and add 1

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

1. $(-123)_{10} = (11111011)_2$
2. $(-123)_{10} = (10000101)_2$
3. $(-123)_{10} = (1000101)_2$

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

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

The image shows a handwritten binary addition problem. The first number, 7532, is represented in 16-bit 2's complement as 1111 1110 0100 0000. The second number, BB4, is represented as 1011 1011 0100. The addition is performed bit-by-bit from right to left, with carry bits indicated by blue arrows. The result, shown below a horizontal line, is 1011 1011 0100 0000, which is the 2's complement representation of BB4. This illustrates that adding a positive number to a negative number in 2's complement results in a negative number, and the sign bit of the result is the same as the sign bit of the negative operand.

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?

$$-2^{n-1} \sim 2^{n-1}-1$$

- 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.

Handwritten examples for rule 1:
 $4 \text{ (0100)} + 4 \text{ (0100)} = 8 \text{ (1000)}$ (overflow)
 $0111 \leftrightarrow 7$
 $1000 \leftrightarrow -8$ ✓

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

Handwritten example for rule 2:
 $0111 + 0101 \rightarrow 1100$ (overflow)

binary

Numbers Systems

Overflow

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

1. $(7532 + BB4)_{16} = (70E6)_{16}$ X overflow.

2. $(7532 + 0BB4)_{16} = (80E6)_{16}$ V overflow.

3. $(0110 + 0101)_2 = (\quad)_2$ V overflow.

4. $(1011 + 1111)_2 = (1010)_2$

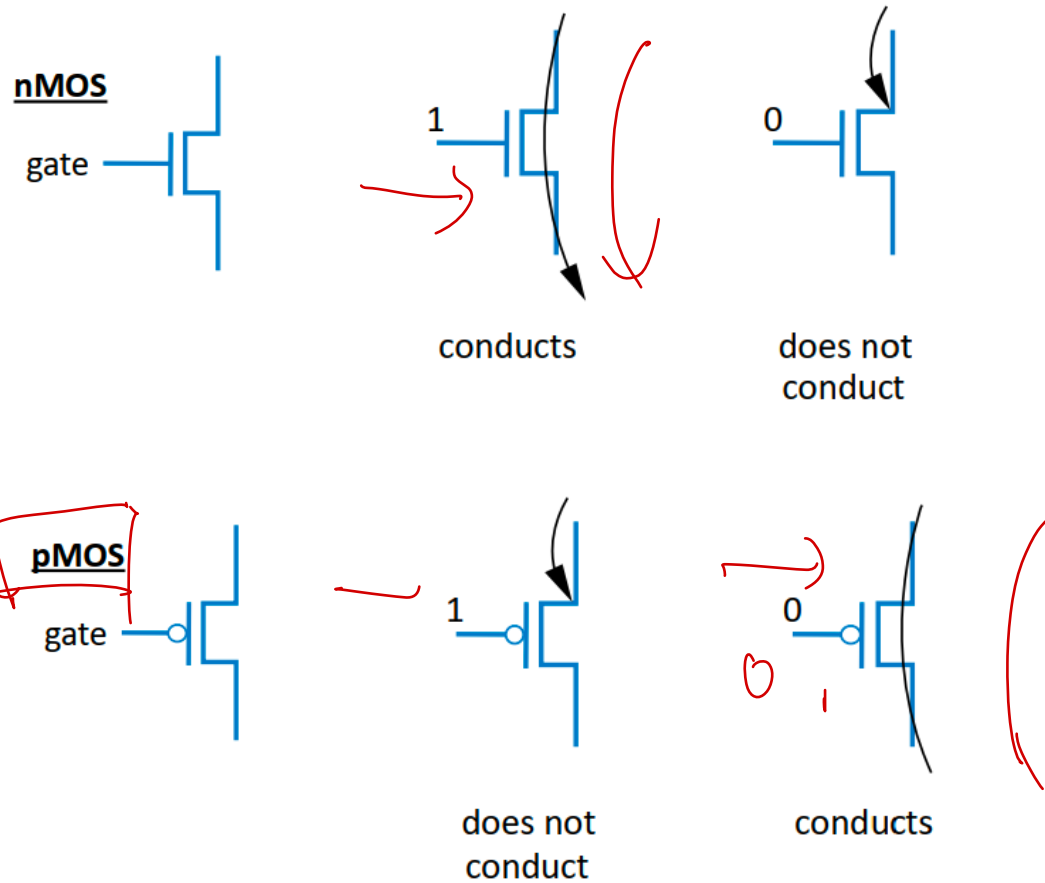
Handwritten binary addition for exercise 4:

$$\begin{array}{r} 1011 \\ + 1111 \\ \hline 11010 \end{array}$$

Logic gates

CMOS Transistor

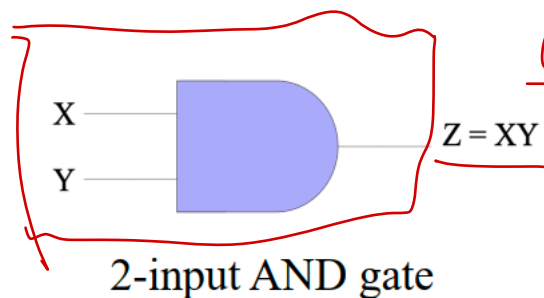
nMOS / pMOS



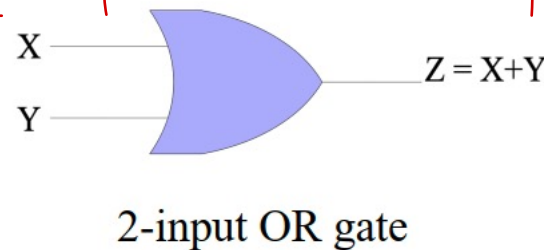
Logic gates

Logic gates

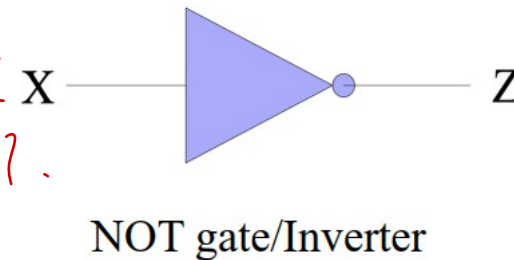
Symbol and truth table of all logical gates: AND, NAND, OR, NOR, XOR, XNOR, NOT, *buffer*



all 1 \rightarrow 1.



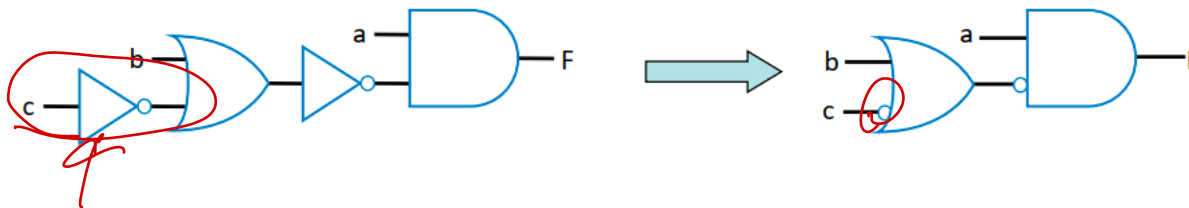
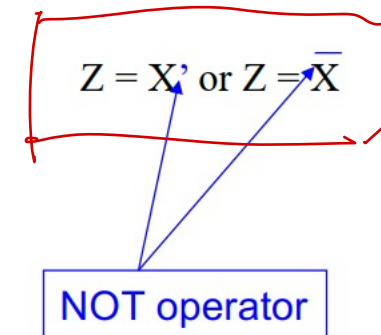
*one of input is 1.
output will be 1.*



- 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

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

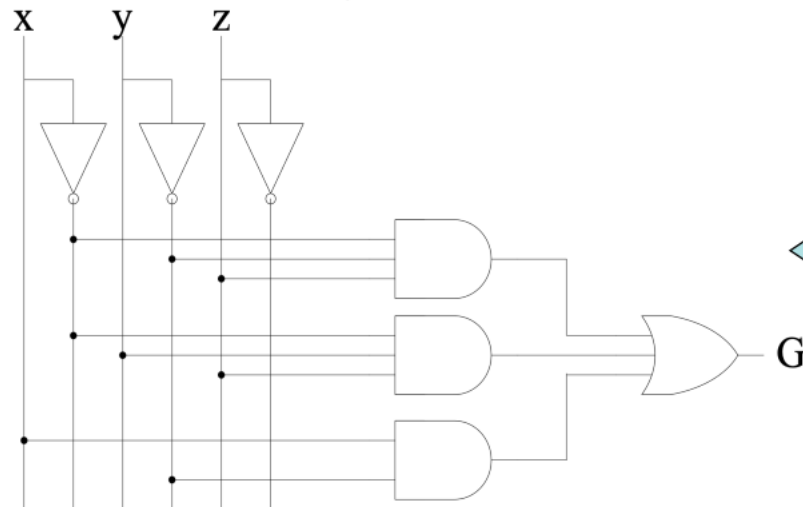
~~order~~

Logic gates

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	

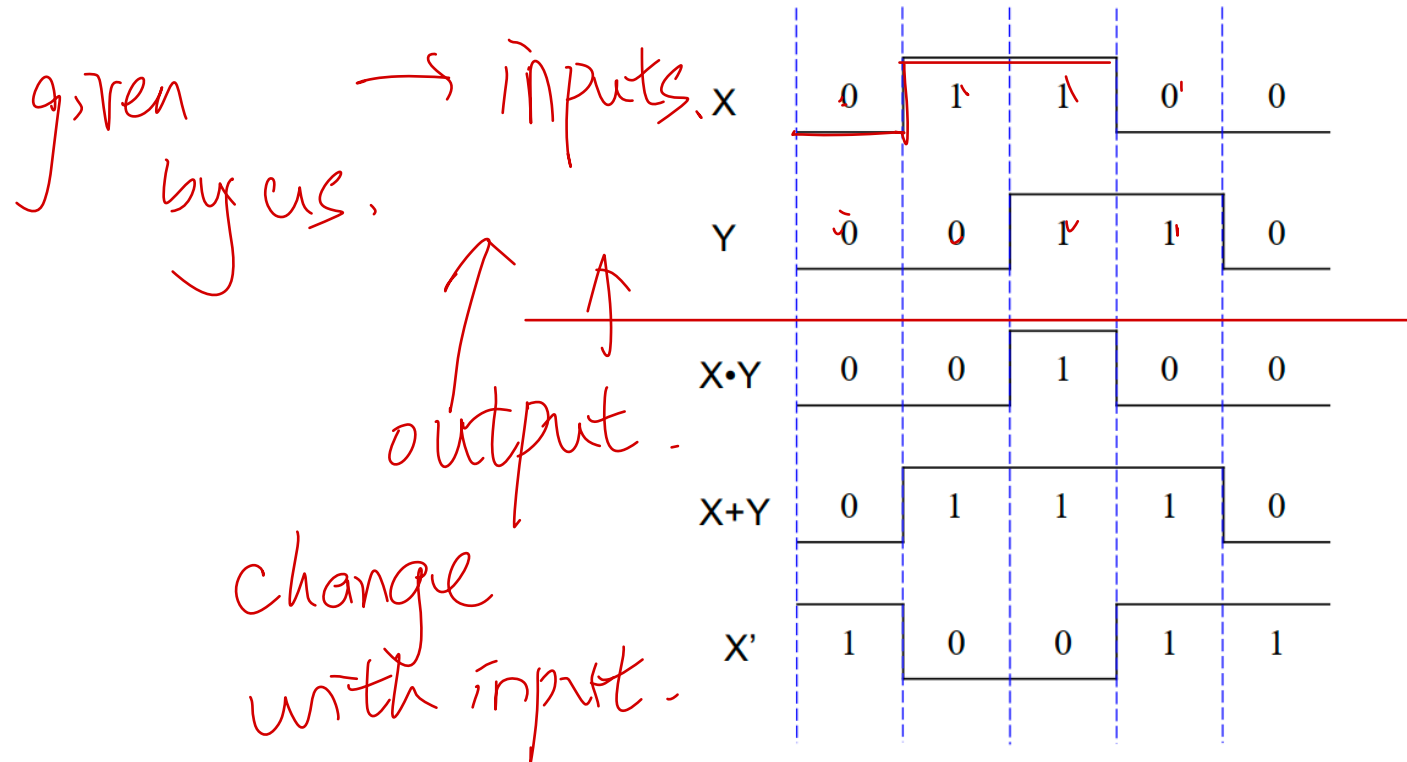
Truth Table



Logic gates

Timing Diagram

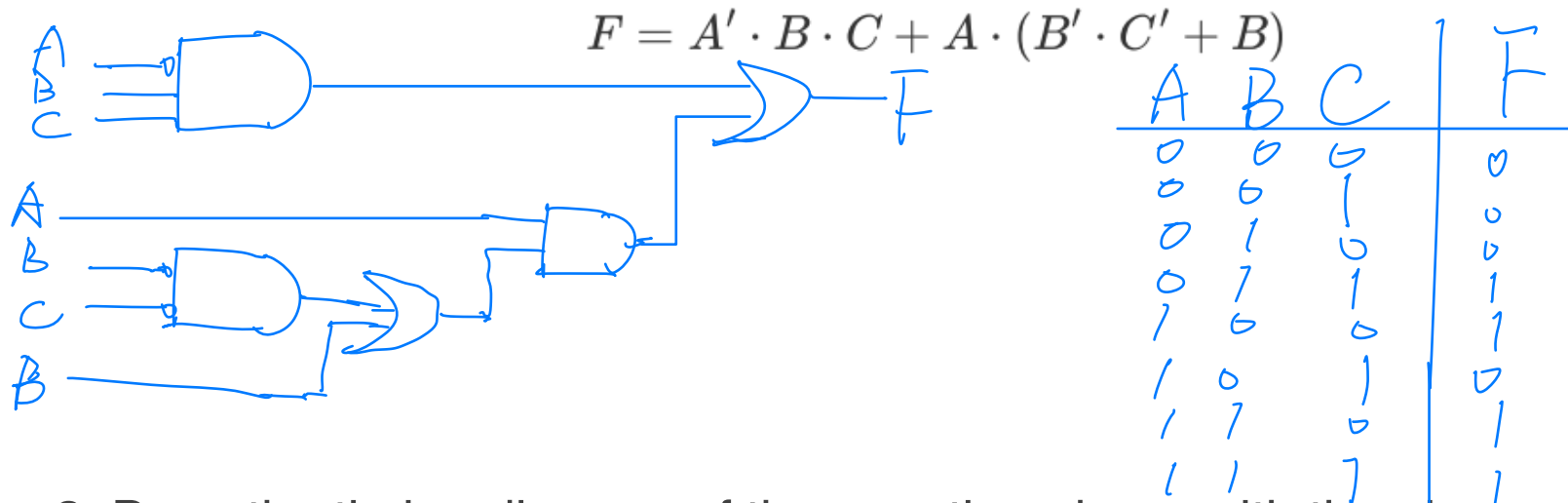
Timing diagram: a diagram that shows the response to changes on a signal in voltage levels with time



Logic gates

Exercise:

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



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

