VE270 Introduction to Logic Design

Recitation Class 1

CHEN Yuchi

UM-SJTU JI

May 20, 2020

Table of Content

Table of contents

Properties of analog signal and digital signal:

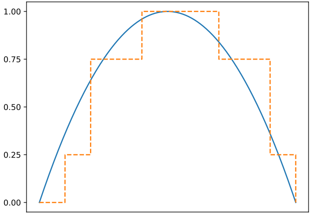
- Analog Signal: Infinite possible values.
  Example: Microphone, variable resistor...
- Digital Signal: Finite possible values.
  Example: Button, keypad...

In binary digital signal, only two possible values can be taken: 0 and 1.

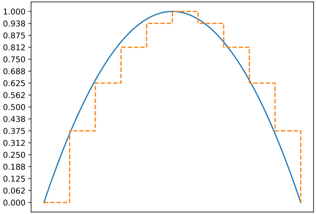
- Physically, binary values are represented by low voltages and high voltages, which is also how transistors are controlled.
- A larger value can be represented as combination of zeros and ones. Each digit is called a bit.
- Converting analog signal to digital signal (usually binary digital signal) is called digitization.
- Digital signal is easier to recover after distortion. That's one of the reasons digital signals are widely used.

Digital Signal

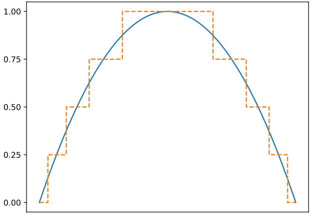
Sample Rate and Bit Depth



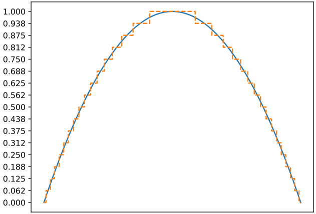
(c) Low Rate Low Depth



(d) Low Rate High Depth



(e) High Rate Low Depth



(f) High Rate High Depth

Table of contents

Binary System
Binary Number Encoding

In all number systems, each position of the number is associated with a weight. Here shows a decimal number and a binary number:

$$\begin{array}{cccccc} 5 & 4 & 3 & . & 2 & 1 \\ - & - & - & - & - & - \\ 10^2 & 10^1 & 10^0 & & 10^{-1} & 10^{-2} \end{array} \qquad \begin{array}{cccccc} 1 & 0 & 1 & . & 1 & 1 \\ - & - & - & - & - & - \\ 2^2 & 2^1 & 2^0 & & 2^{-1} & 2^{-2} \end{array}$$

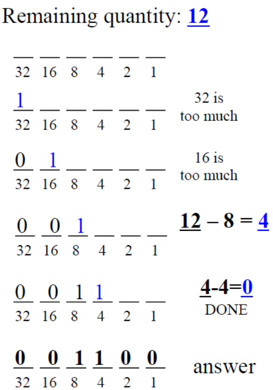
Thus, a binary number can be easily converted to decimal number by multiplying the digits with their weighs.

$$(101.11)_2 = 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times 0.5 + 1 \times 0.25 = (5.75)_{10}$$

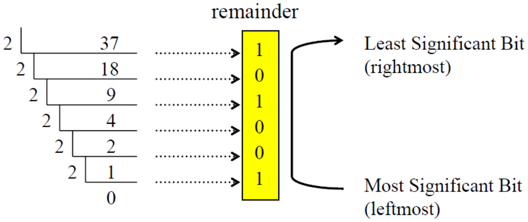
Binary System

Convert Decimal Number to Binary Number

There's general two ways two convert a decimal number to binary number.



(a) Subtraction Method

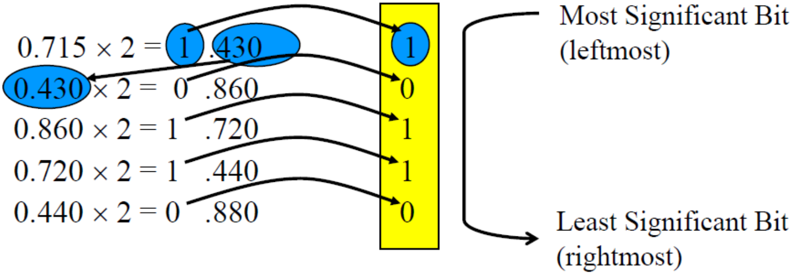


(b) Division Method

Binary System

Convert Decimal Number to Binary Number

There's also a method to convert the decimal part of decimal number to binary number.



$$(0.715)_{10} \approx (0.10110......)_2$$

Besides binary numbers, we also have octal and hexadecimal numbers. Since 8 and 16 are both powers of 2, converting numbers among these three base systems is easy.

$$
\begin{array}{ccccc}
(& \mathrm{B} & 8 & \mathrm{A} &)_{16} \\
 & - & - & - & \\
(& 1011 & 1000 & 1010 &)_2
\end{array}
\qquad
\begin{array}{ccccc}
(& 7 & 3 & 5 &)_8 \\
 & - & - & - & \\
(& 111 & 011 & 101 &)_2
\end{array}
$$

Remember to add more bits when converting binary numbers to octal or hexadecimal numbers. Take $(0101101)_2$ and $(01110)_2$ for example.

$$
\begin{array}{cccc}
(& 0010 & 1101 &)_2 \\
 & - & - & \\
(& 2 & \mathrm{D} &)_{16}
\end{array}
\qquad
\begin{array}{cccc}
(& 001 & 110 &)_2 \\
 & - & - & \\
(& 1 & 6 &)_8
\end{array}
$$

Please remember, for our course, octal numbers and hexadecimal numbers are regarded as a "special representation" of binary numbers.

The easiest way is to use decimal numbers as an intermediate. Because we are used to working with decimal numbers.

Subtraction method and division method do work in all cases, but you, as a human, may not good at subtraction and division without decimal number. For example:

$$(7\text{BA})_{16} = (?)_3$$

$$(7\text{BA})_{16}/(3)_{16} = (293)_{16} \cdots (1)_{16}$$

In this case, when you first calculate $7/3 = 2 \cdots 1$, remind the remaining 1 represents 16. $(16 + 11)/3 = 9$. You can find that you are still using decimal numbers. So, if you haven't memorized a multiplication table under base-16, don't do this.

This is a very very important concept.
Let's do some practice first. Assume 2's complement:

Example

$$
\begin{aligned}
(10110)_2 + (11011)_2 &= \\
(213)_8 + (66)_8 &= \\
(125)_8 + (324)_8 &= \\
(\mathrm{F3A})_{16} - (\mathrm{7F})_{16} &=
\end{aligned}
$$

This is a very very important concept.

Let's do some practice first. Assume 2's complement:

Example

$$
\begin{aligned}
(10110)_2 + (11011)_2 &= (10001)_2 \\
(213)_8 + (66)_8 &= (201)_8 \\
(125)_8 + (324)_8 &= (451)_8, \text{ overflow} \\
(\mathrm{F3A})_{16} - (\mathrm{7F})_{16} &= (\mathrm{EBB})_{16}
\end{aligned}
$$

If you don't do them correctly, pay attention to following content.

- Why do we use 2's complement?
  Because digital device doesn't know what a minus sign is.
- How can we write or read 2's complement?
  Write: Convert the absolute value to binary and add a zero on the left of it. If it's a positive number, keep the result and if it's a negative number, invert each bit and add 1 to it.
  Read: If the most significant bit is 0, directly convert it to decimal number. If the most significant bit is 1, invert each bit and add 1 to it. Then convert it to decimal number and add minus sign.
  Another way: Only treat MSB as negative.
- What is the range of 2's complement number?
  For a n bits 2's complement number, its range is $[-2^{n-1}, 2^{n-1}-1]$.
  Please take care of the maximum and minimum case.

- What about calculation?
  - Subtraction is finished by adding the 2's complement.
  - If the number of digits are different, you need to complete the shorter one. For positive numbers add zeros and for negative numbers add ones.
  - Consider either an infinite-bit case (overflow will not happen) or a finite-bit case (overflow may happen).
  - If the question doesn't ask particularly, when meeting an overflow, you can either indicate overflow and keep the overflowed answer or write the correct answer.
  - If the question indicates that there's finite bits or ask you to write out overflow, you must indicate an overflow if you meet one. Correct answer is optional.
  - Remember, we treat octal numbers and hexadecimal numbers as a representation of binary numbers.

If you are not sure about your calculation, use decimal number to check it.
Answer to the practice:

$$(10110)_2 + (11011)_2 = (110001)_2 \text{ or } (10001)_2$$

$$\begin{aligned}(213)_8 + (66)_8 =& (010\ 001\ 011)_2 + (110\ 110)_2 \\ =& (010\ 001\ 011)_2 + (111\ 110\ 110)_2 \\ =& (010\ 000\ 001)_2 = (201)_8\end{aligned}$$

$$\begin{aligned}(125)_8 + (324)_8 =& (001\ 010\ 101)_2 + (011\ 010\ 100)_2 \\ =& (100\ 101\ 001)_2 = (451)_8, \text{ overflow}\end{aligned}$$

$$\begin{aligned}(\mathrm{F3A})_{16} - (\mathrm{7F})_{16} =& (1111\ 0011\ 1010)_2 - (0111\ 1111)_2 \\ =& (1111\ 0011\ 1010)_2 + (1111\ 1000\ 0001)_2 \\ =& (1110\ 1011\ 1011)_2 = (\mathrm{EBB})_{16}\end{aligned}$$

Pause here.
Any questions?

Table of contents

Logic Gates
Transistors

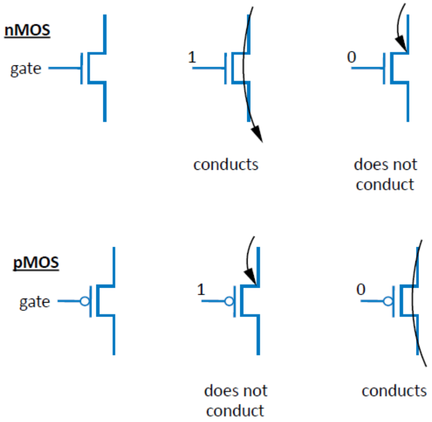
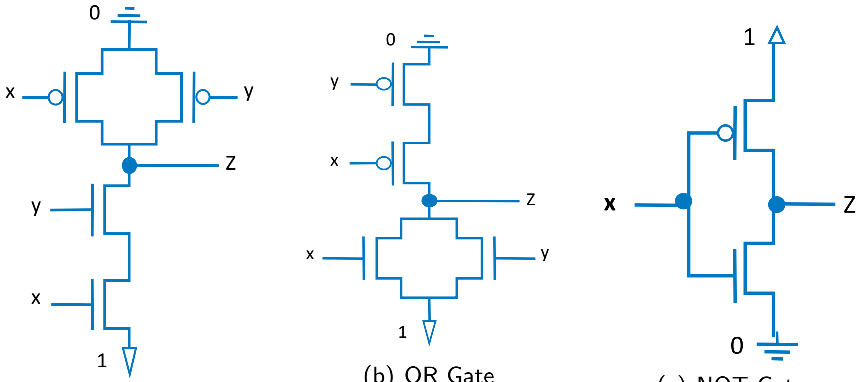


Figure: Two Kinds of Transistors



(a) AND Gate (b) OR Gate (c) NOT Gate

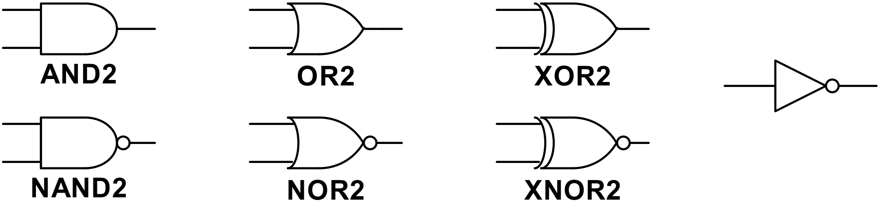


Figure: 7 Kinds of Logic Gates

I create them by Multisim and I keep the name there only for you to recognize. When you draw a logic gate, you needn't specify the name of it, as long as it's recognizable.

x	y	F
0	0	0
0	1	0
1	0	0
1	1	1

Table: AND

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1

Table: OR

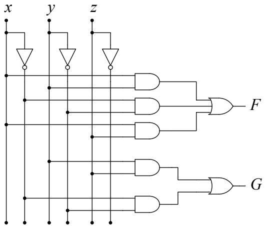
x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

Table: XOR

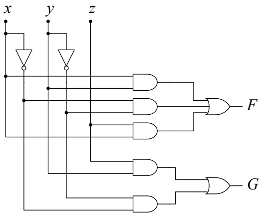
x	F
0	1
1	0

Table: NOT

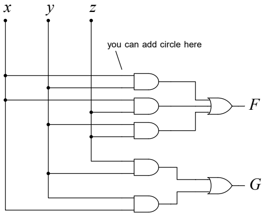
- When writing truth table, arrange the cases as a binary numbers plus one.
- Use 0 and 1. Don't use T/F or Vcc/GND. (In other course, follow its instruction.)
- You cannot miss any row.



(a) Draw Straight Line



(b) Avoid Floating Pin



(c) Make It Clearer

- Do not label in logic gates.
- Do not use arrows.
- Avoid multi-driver problem. (Very Serious)

NOT>AND>OR

Practice:

- $F = (a+b)c$
- $F = a'(b+c)c'$
- $F = a'b + b'c + c'd$

Two ways to detect overflow:

- Check the three sign bits of the three numbers.
- Use XOR gate to check the carries. (Generally used)

When drawing a time diagram:

- Be careful about the result of each gate.
- Be careful about when the input changes. There are dashed lines helping you.