ME133 Lecture 11 2/21/23

Last time:

> Midterm

goal: grades by next Thursday

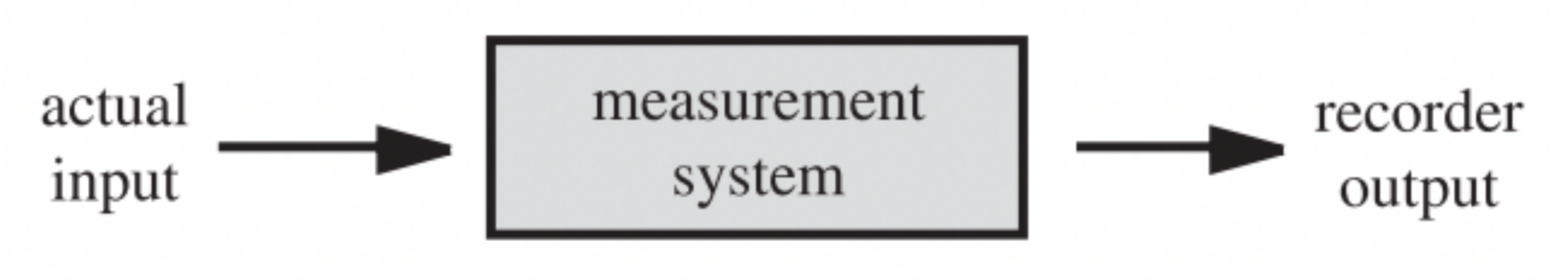
Today:

> Remarks on Second Order systems

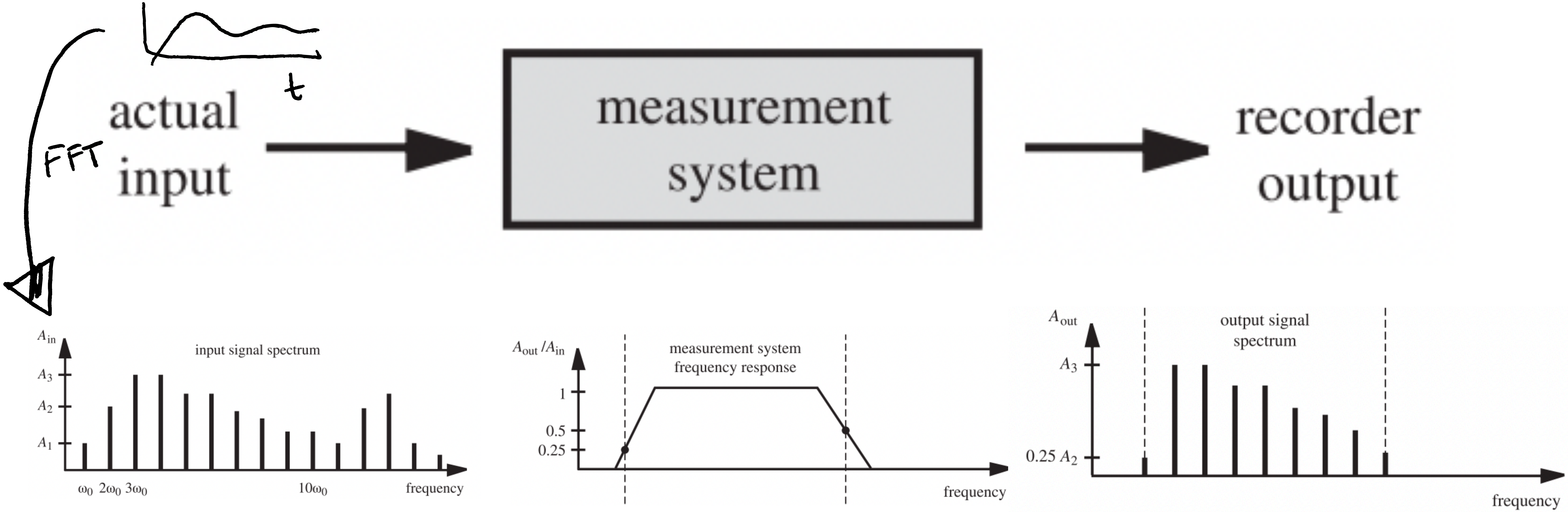
> Digital Circuits (Ch. 6)

> digital signals and representations
>combinational logic
> ~~timing diagrams~~

First, Summarize "System Response" (Ch. 4)



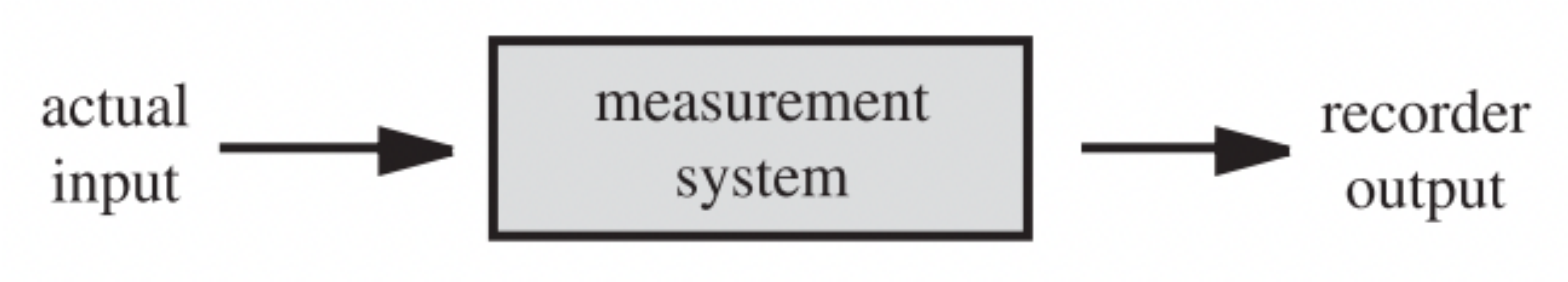
First, Summarize "System Response" (Ch. 4)



take away: if you know Freq. response you can predict the output given some input spectrum.

Collarly: you can build the Freq. response from input output data.

First, Summarize "System Response" (Ch. 4)



> zero-order

└→ gain.

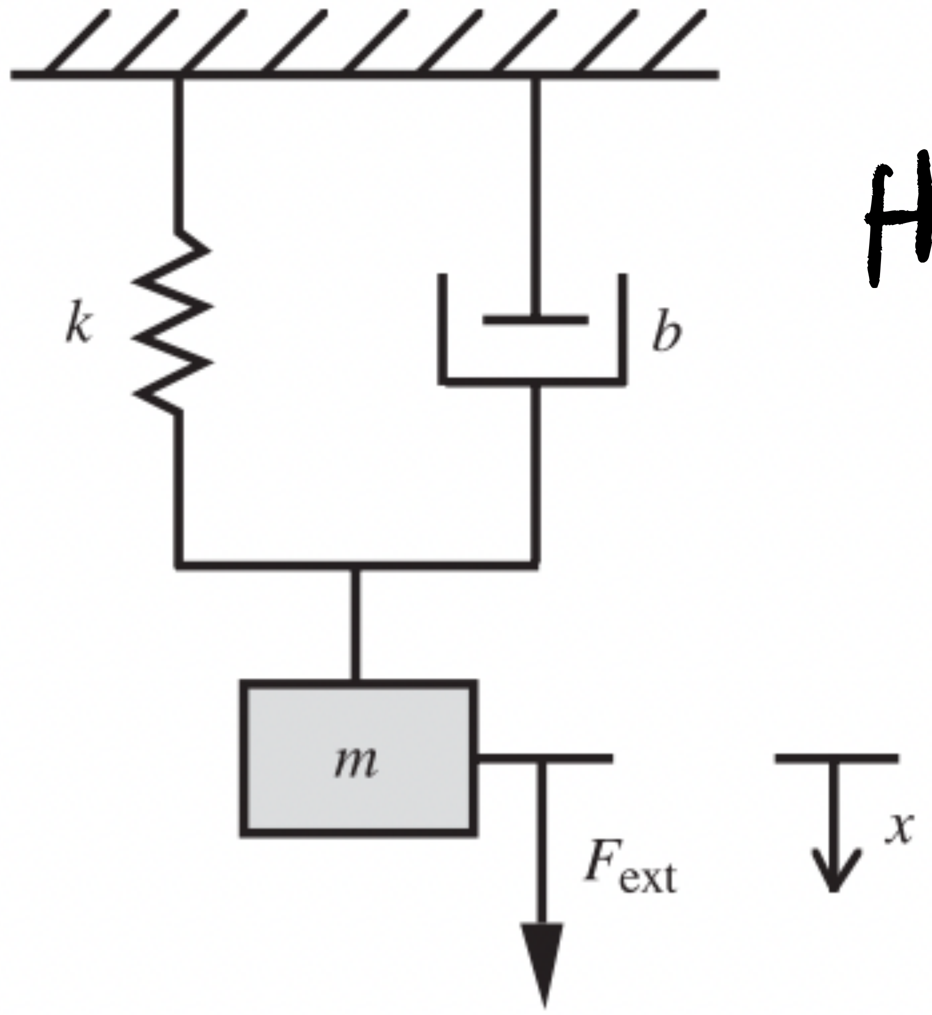
> first-order

└→ τ: time constant , gain.

First, Summarize “System Response” (Ch. 4)

The ~~next~~ most important system:

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = F_{ext}(t)$$



Harmonic oscillator

Key Idea of Second order Systems:

$$ms^2 + bs + k = 0$$

$$s_1 = -\frac{b}{2m} + \sqrt{\left(\frac{b}{2m}\right)^2 - \frac{k}{m}}$$

$$s_2 = -\frac{b}{2m} - \sqrt{\left(\frac{b}{2m}\right)^2 - \frac{k}{m}}$$

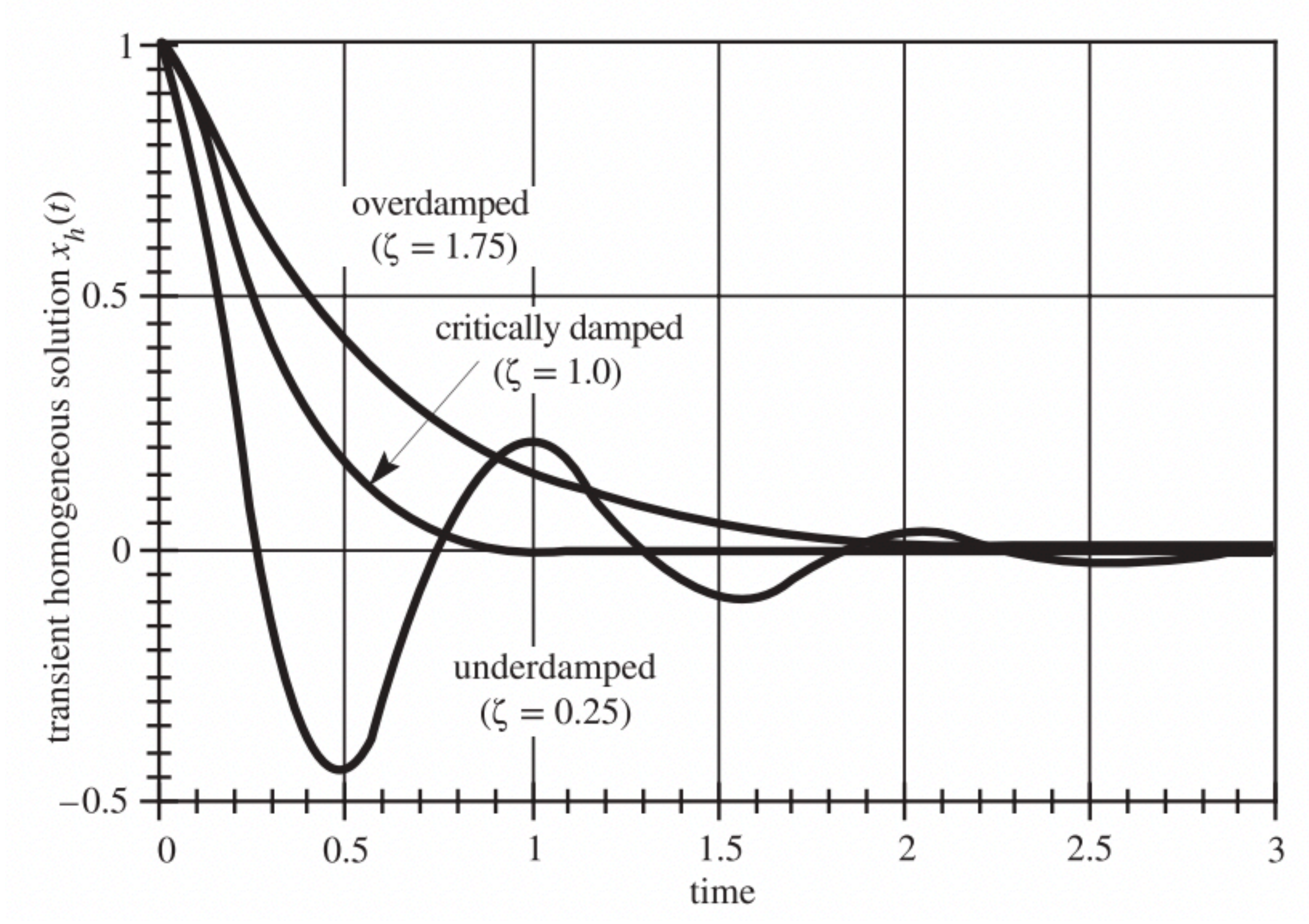
$$\zeta = \frac{b}{b_c} = \frac{b}{2\sqrt{km}} \qquad \omega_n = \sqrt{\frac{k}{m}}$$

$$s_1 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1}$$

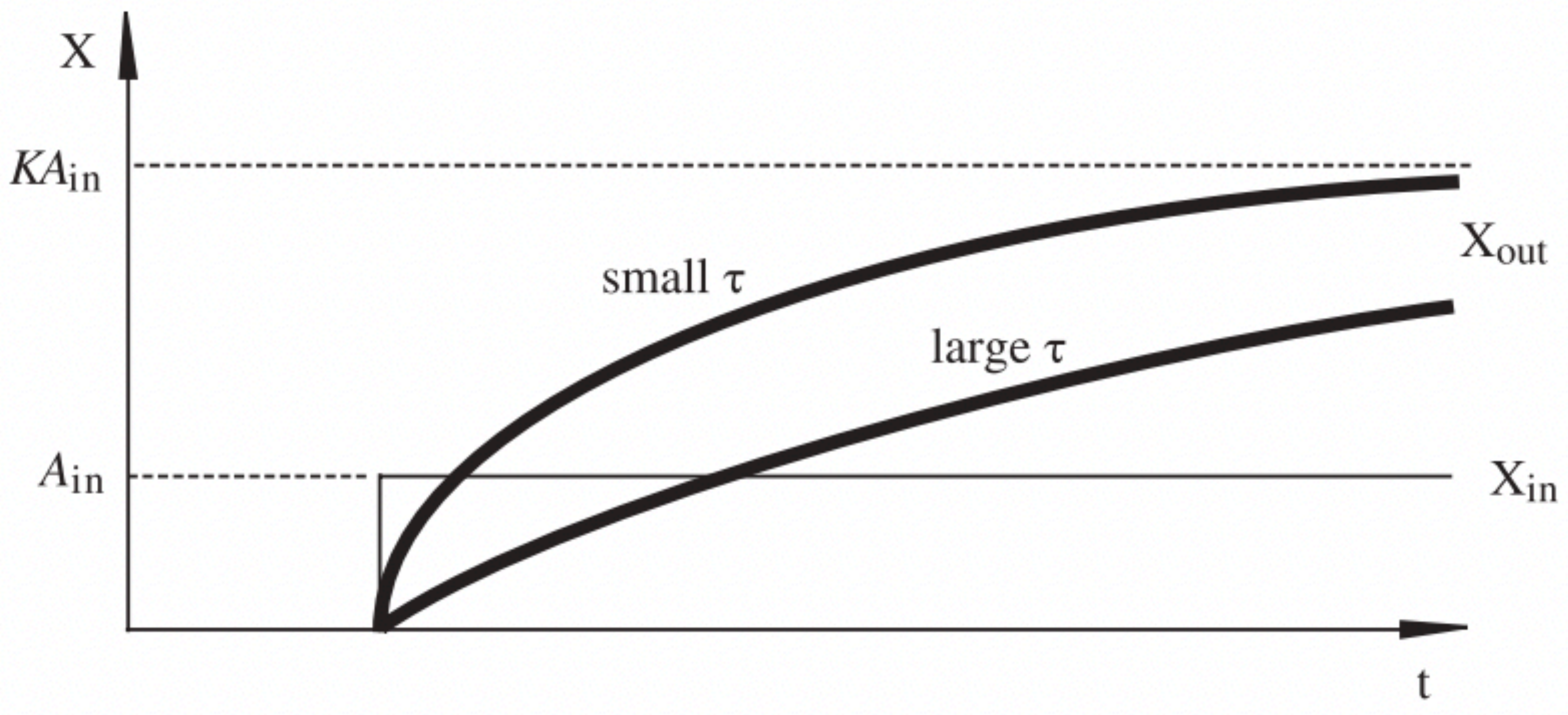
$$s_2 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}$$

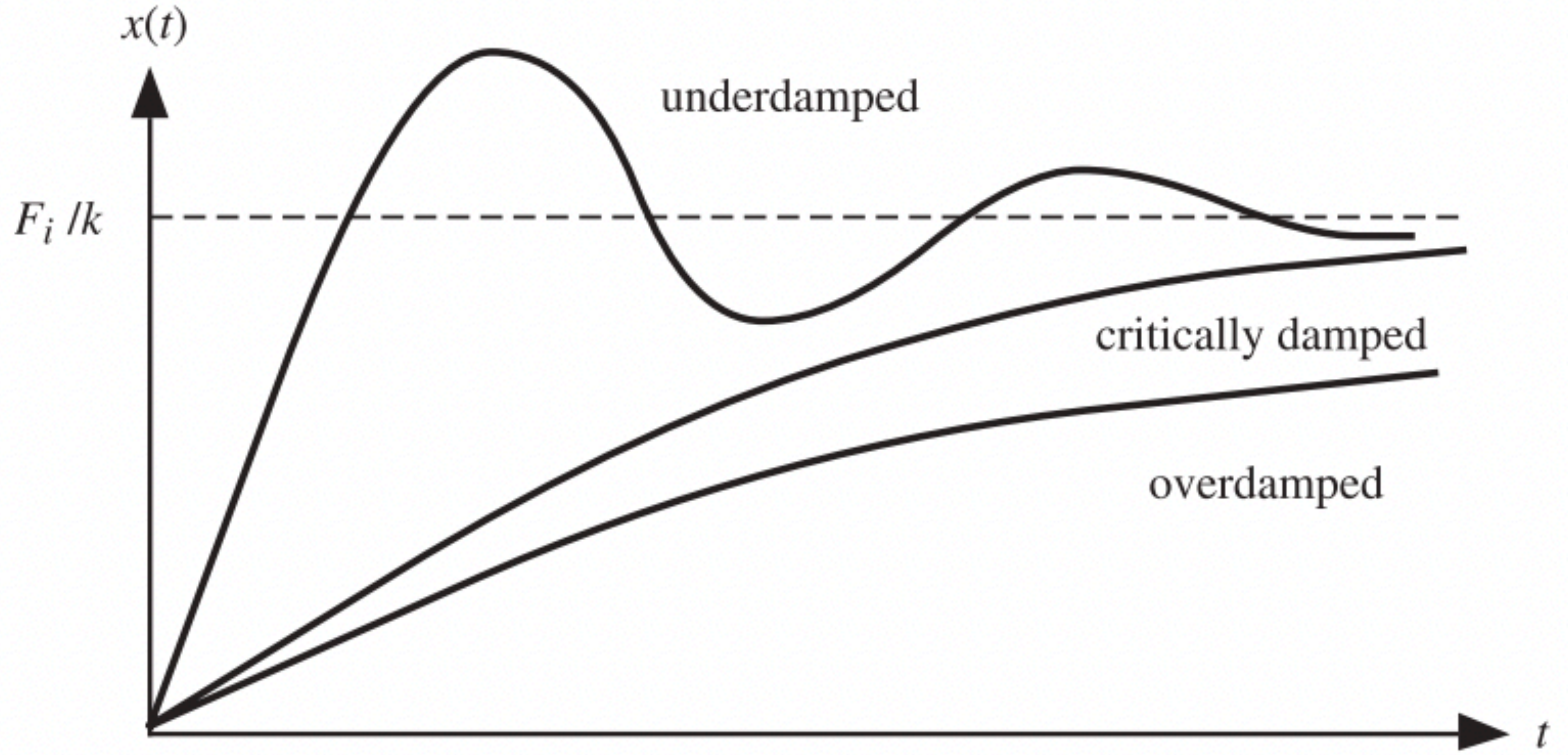
Roots of the Characteristic Eq. can be complex

Key Idea of Second order Systems:

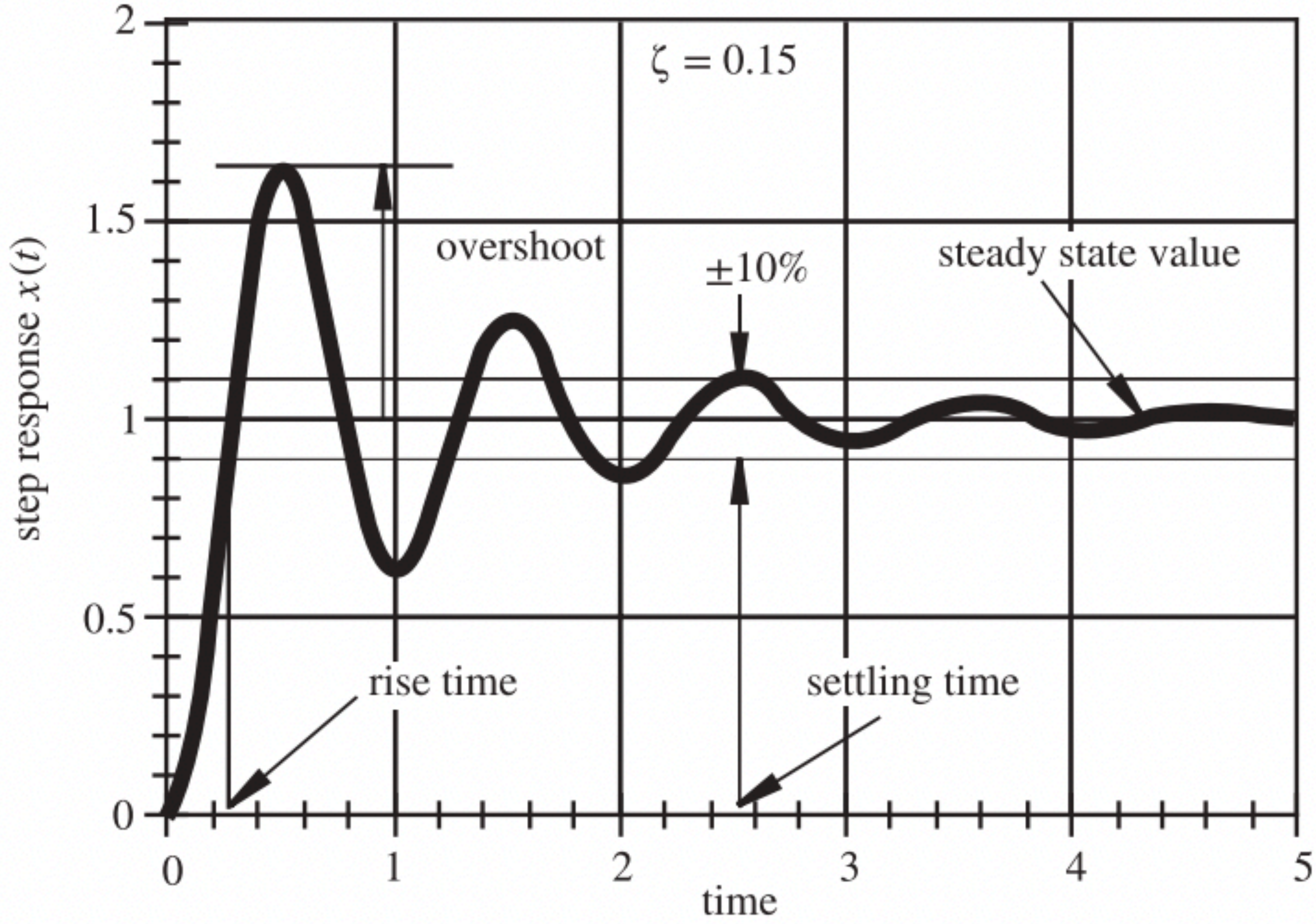


Compare First vs Second Order Step Response

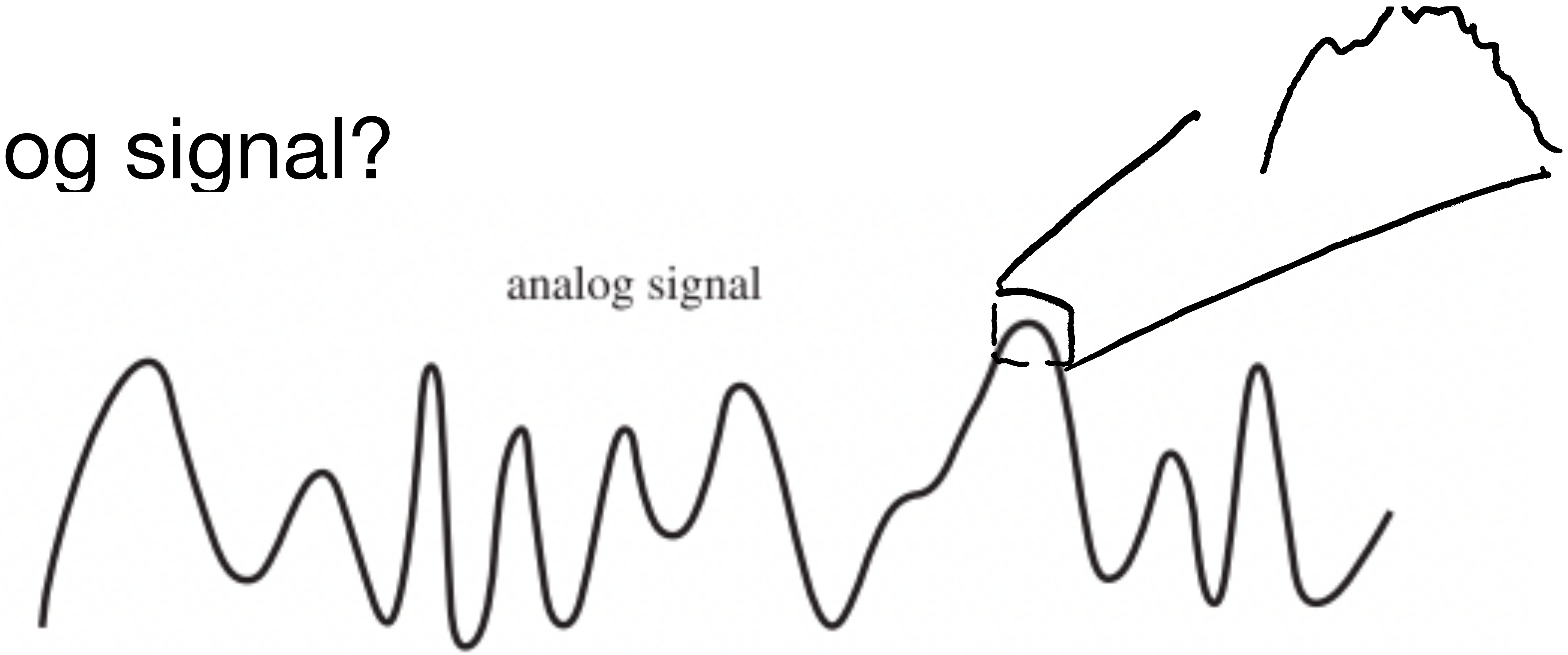




Features of Second Order Step Response

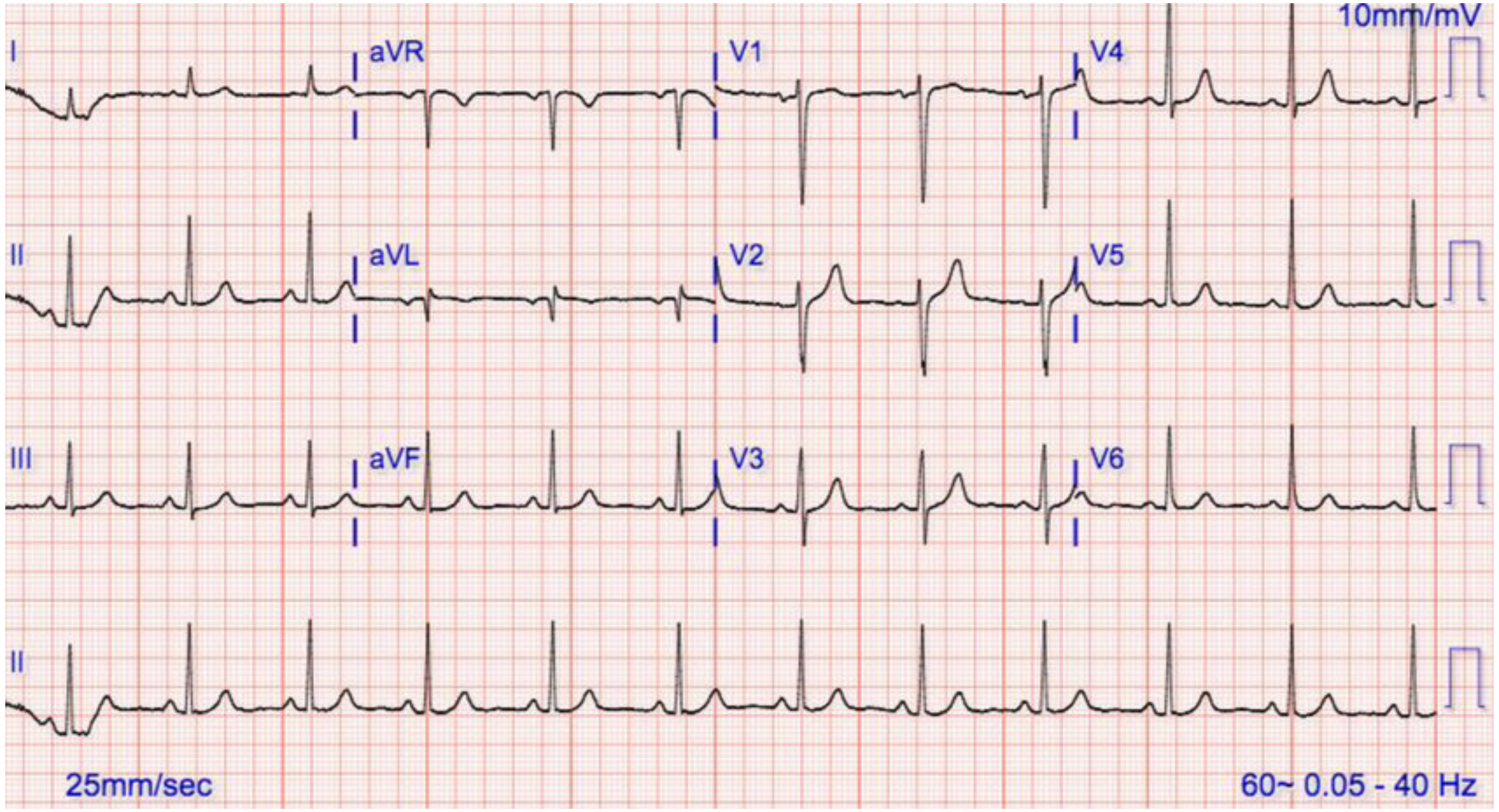


What is an analog signal?

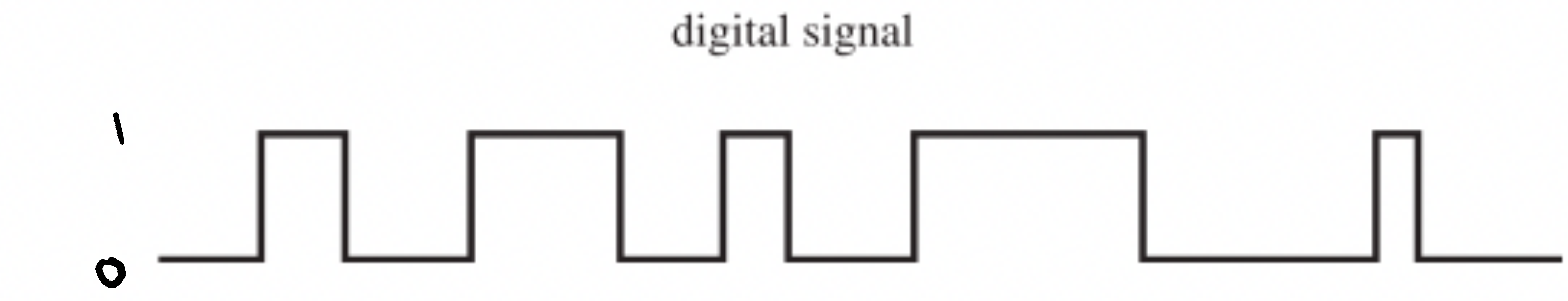


- changes continuously
- infinite values
- 'analog' → signal represents some other physical quantity

Analog Signal Example



What is a digital signal?

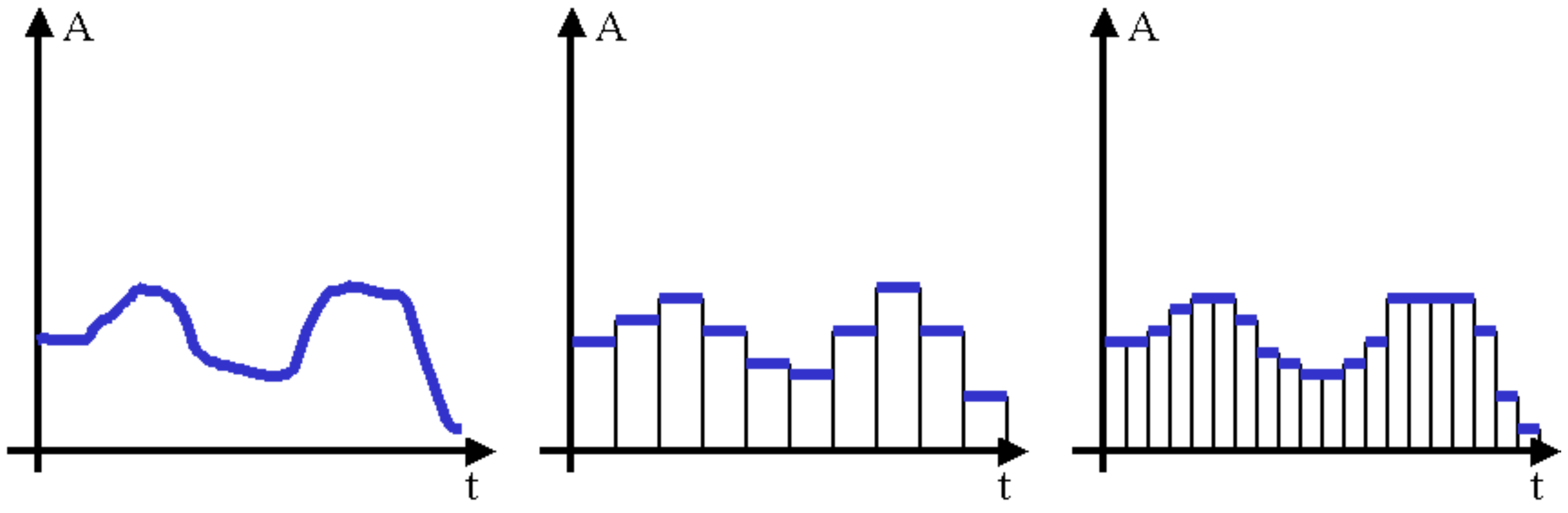


Binary
Signal

on vs off
true or false
0 or 1

- only take on "discrete"
  ⟶ certain values

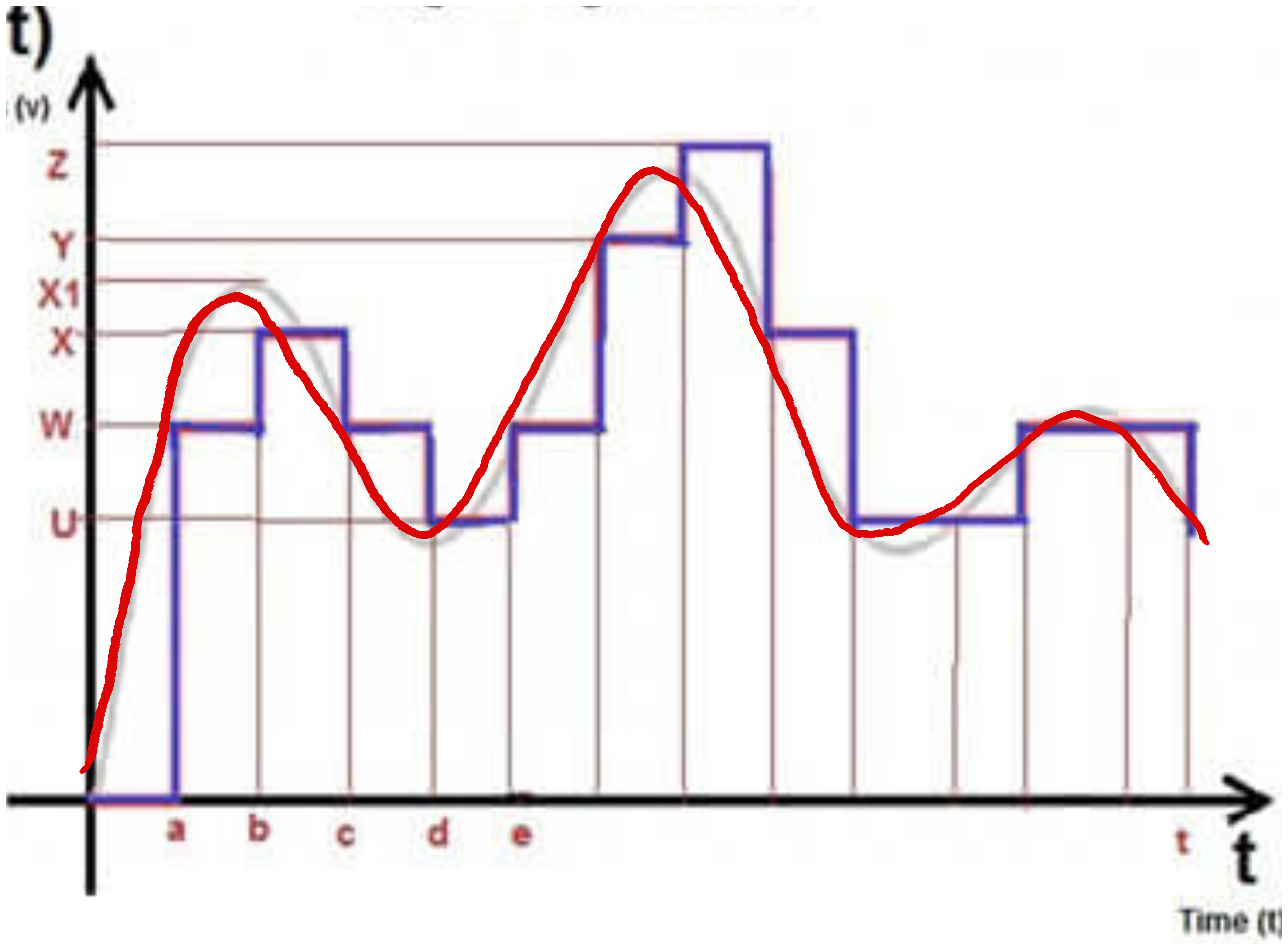
Digital signal examples:



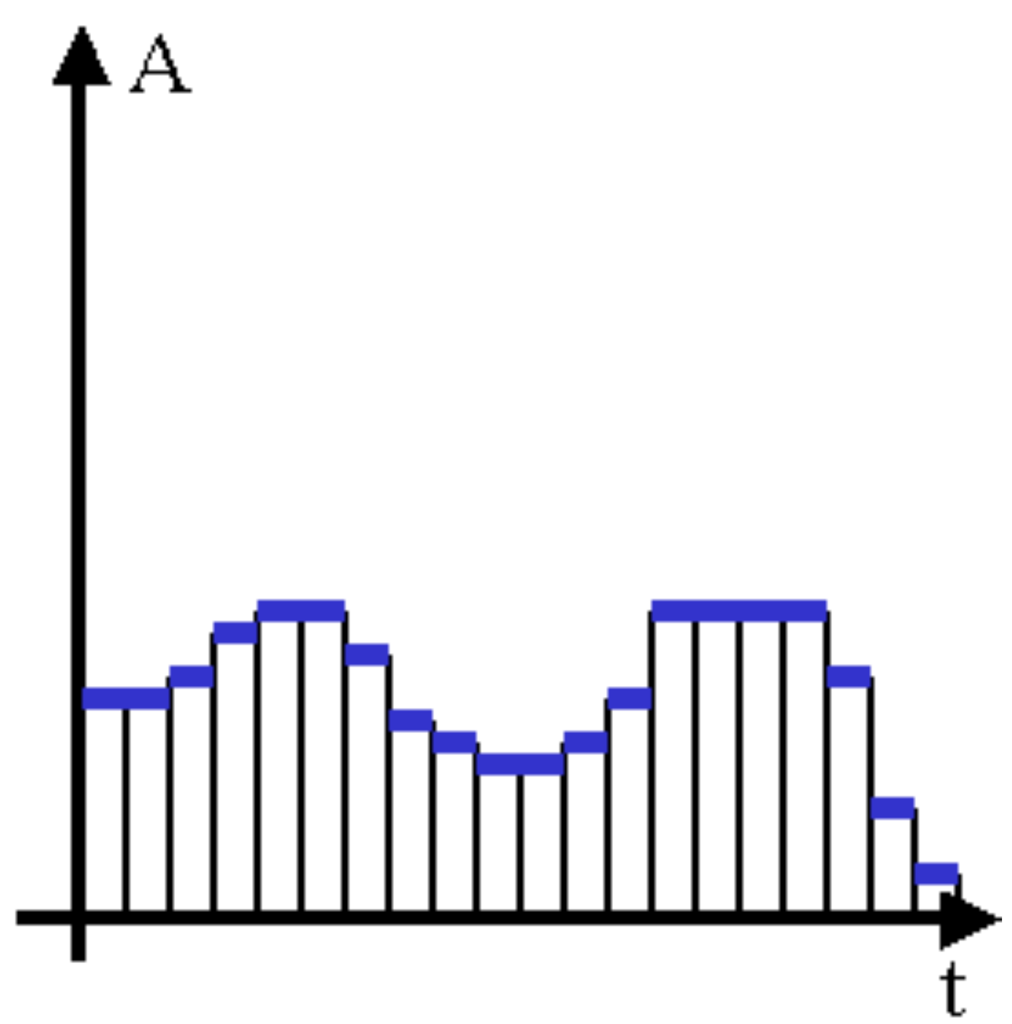
Analog signal – continuously varying

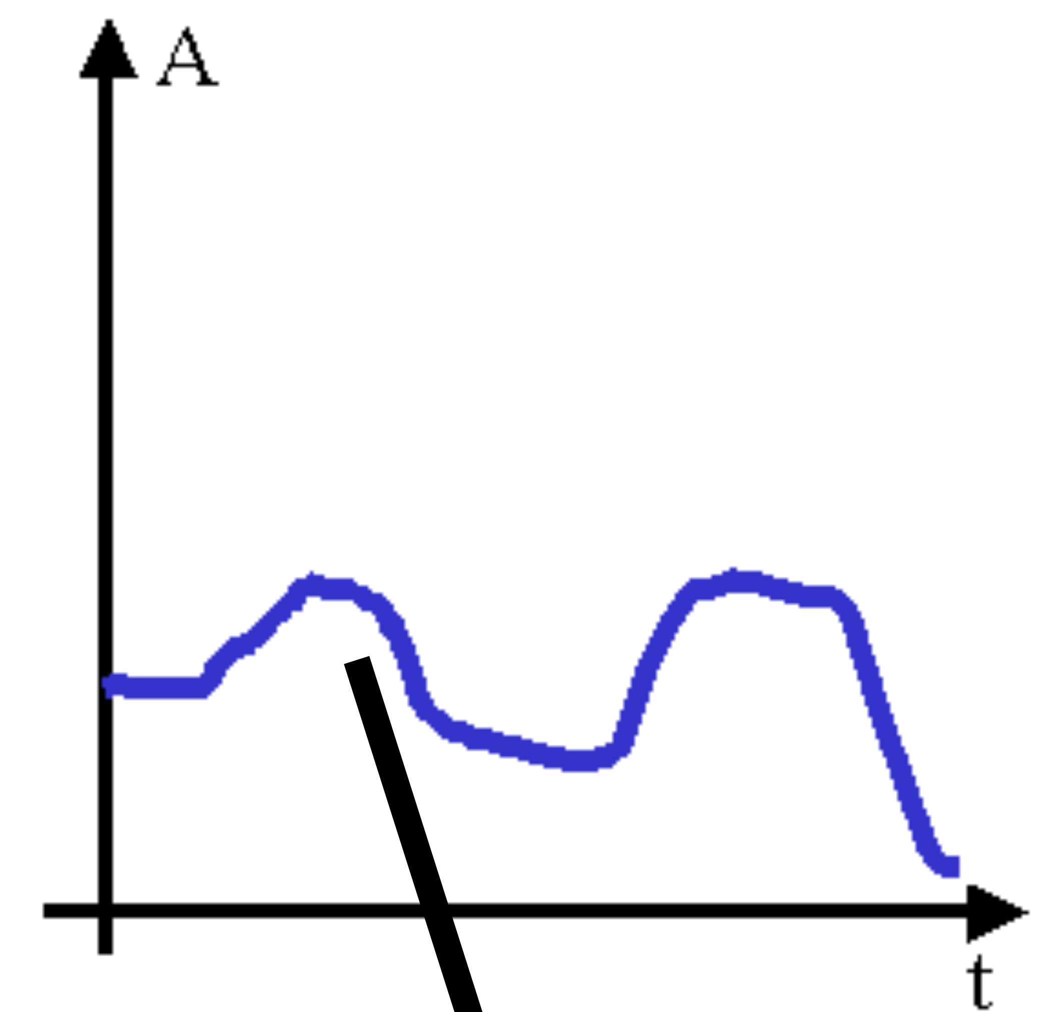
Digital signal – large time divisions

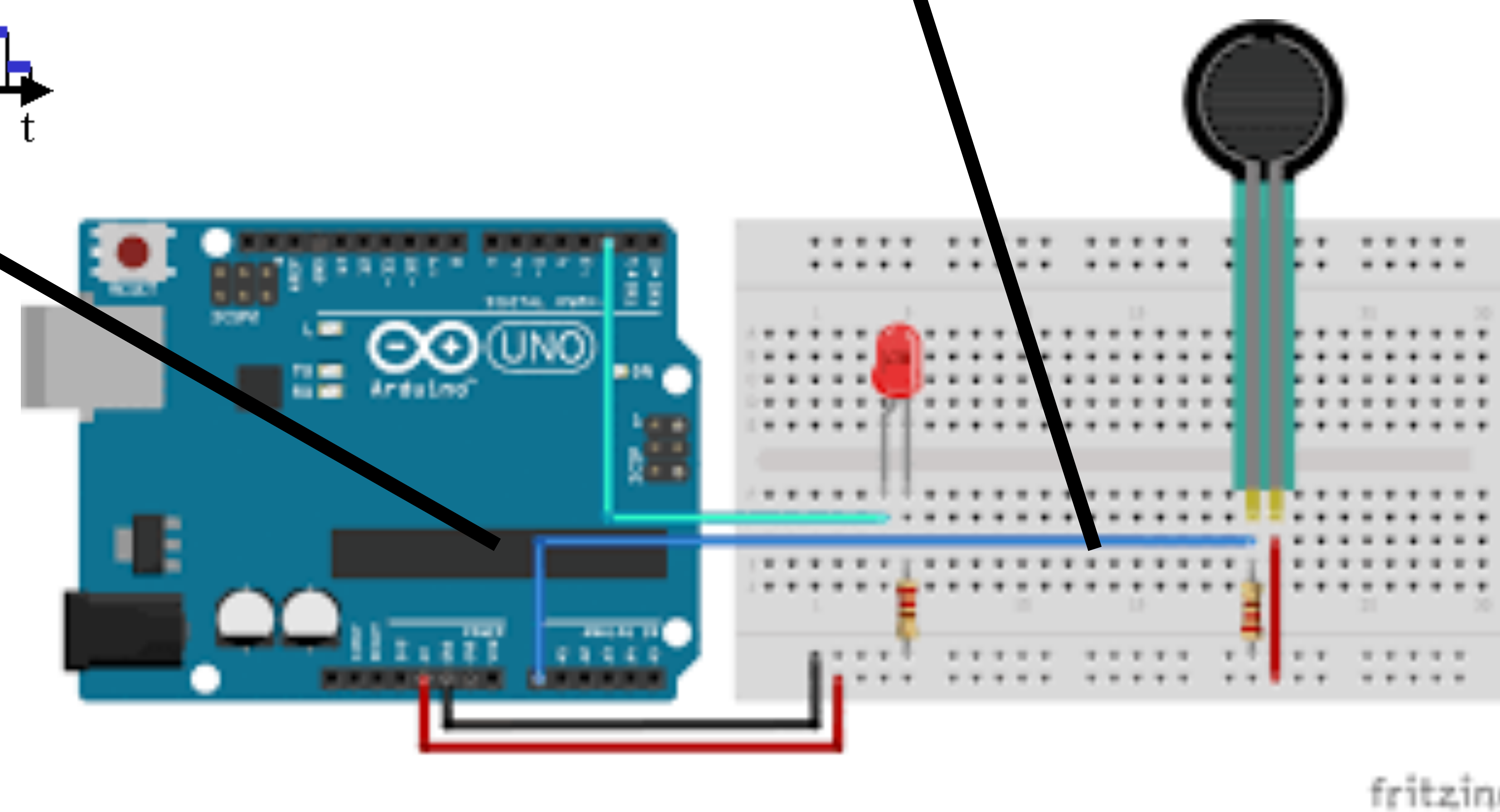
Digital signal – small time divisions



Digital signal examples:







Digital Representations: How to represent a digital number?

What does "base" of a number mean?

Base: # of possible symbols to represent a digit.

10 $\longrightarrow$ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Base representations

Base 10:

$$d_{n-1} \cdots d_3 d_2 d_1 d_0 = d_{n-1} \cdot 10^{n-1} + \vdots + d_3 \cdot 10^3 + d_2 \cdot 10^2 + d_1 \cdot 10^1 + d_0 \cdot 10^0$$

You can include fraction

$d_{-1}, d_{-2} \cdots$

Ex.

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

123

Binary: base 2 representation

Why? → fundamental operation of digital devices can only be on/off (we are using transistors)

$$d_{n-1} \cdots d_3 d_2 d_1 d_0 = d_{n-1} \cdot 2^{n-1} + \vdots + d_3 \cdot 2^3 + d_2 \cdot 2^2 + d_1 \cdot 2^1 + d_0 \cdot 2^0$$

$d_i = 1$ or 0

Example

convert

$$1101 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 + 1 \cdot 2^0$$

$$= 8 + 4 + 0 + 1$$

$$= \underline{\underline{13}}$$

digits in Binary

Bits!

MSB: most significant bit: farthest left bit
$\mapsto$ highest power of two

LSB: least significant bit: right most bit
$\mapsto$ lowest power of two
2^0

Everything in a computer is stored as binary

Byte - 2^8 (256) "Char"

('A', 'B', ..., 'c')

Char v = 'a';

v = 97

v == 'a'

v == 97

kilobyte - 1024 (2^{10}) Bytes

"Jabborwockey"

megabyte - (2^{20}) Bytes

Harry Potter

gigabyte - (2^{30}) Bytes

30min video

terabyte - (2^{40}) Bytes

→ largest consumer hard drive 2001

Petabyte - (2^{50}) Bytes

2000 years of MP3s

Binary Arithmetic

$$\begin{array}{r} 9 \\ +\ 3 \\ \hline 12 \end{array}$$

$9 = 1001$

$3 = 0011$

$$\begin{array}{r} {\scriptstyle 1} \\ 1001 \\ 0011 \\ \hline 1100 \end{array}$$

12!

$$\begin{array}{r} 9 \\ \times\ 3 \\ \hline 27 \end{array}$$

$$\begin{array}{r} 1001 \\ \times\ 0011 \\ \hline 1001 \\ 1001 \\ 0000 \\ 0000 \\ \hline 11011 \end{array}$$

$$\overbrace{1 \cdot 2^4}^{16} + \overbrace{1 \cdot 2^3}^{8} + 0 + \underbrace{1 \cdot 2^1}_{2} + \underbrace{1 \cdot 2^0}_{1}$$

27!

it [illegible]...

Binary is tedious to write/read: solution hexadecimal

Hexadecimal is extensively used for programming devices.

base 16 : { 0, ... 9,

{ A, B, C, D, E, F

{ 10, 11, 12, 13, 14, 15

$$0x9A = 9 \cdot 16^1 + 10 \cdot 16^0 =$$

$$123 = \underbrace{0111}_{7}\ \underbrace{1011}_{B} = 0x7B$$

ASCII : American Standard Code for Information exchange.

Combinational Logic (a subset within the Theory of Automata)

logic gates

Gate → controls the flow of signals

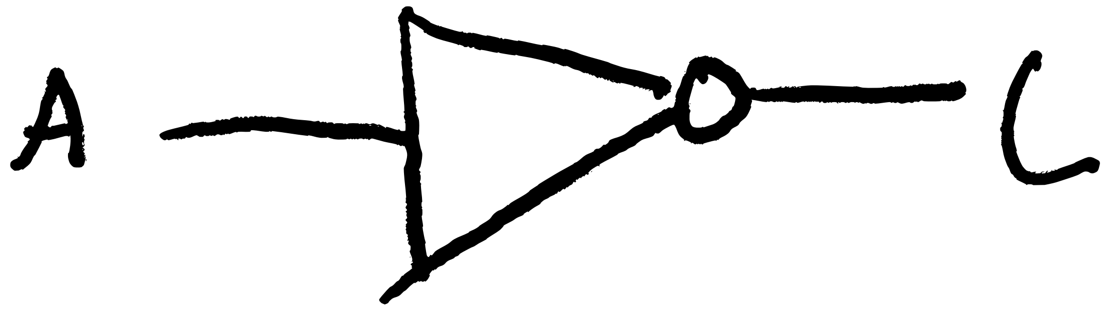
Truth table:

- a way to display all possible combos of input/output

Combinational logic operations (gates)

Inverter

Inverts signal

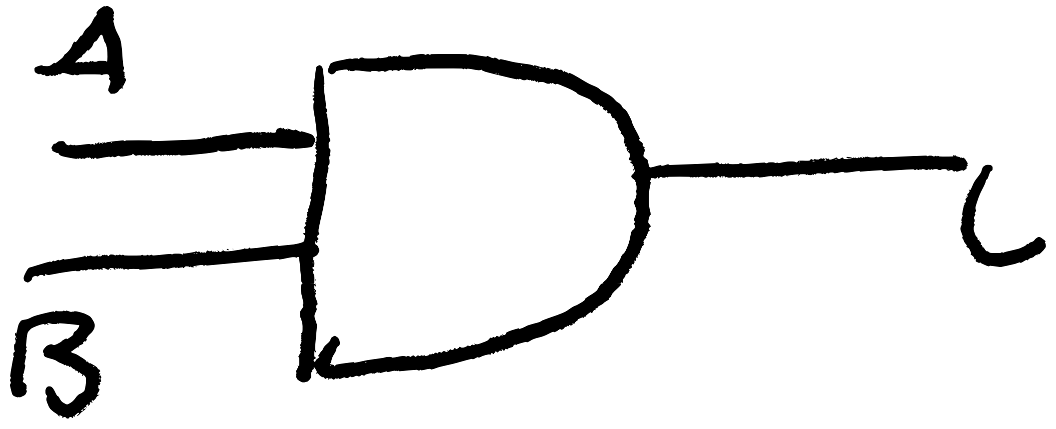


$C = \overline{A}$

A	C
0	1
1	0

AND gate

AND logic (&&)



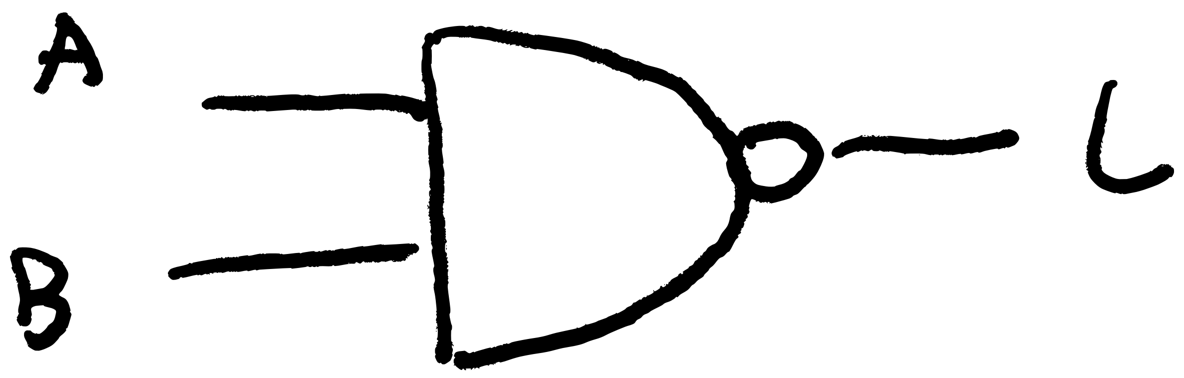
$C = A \cdot B$

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

Combinational logic operations (gates)

NAND gate

inverted AND



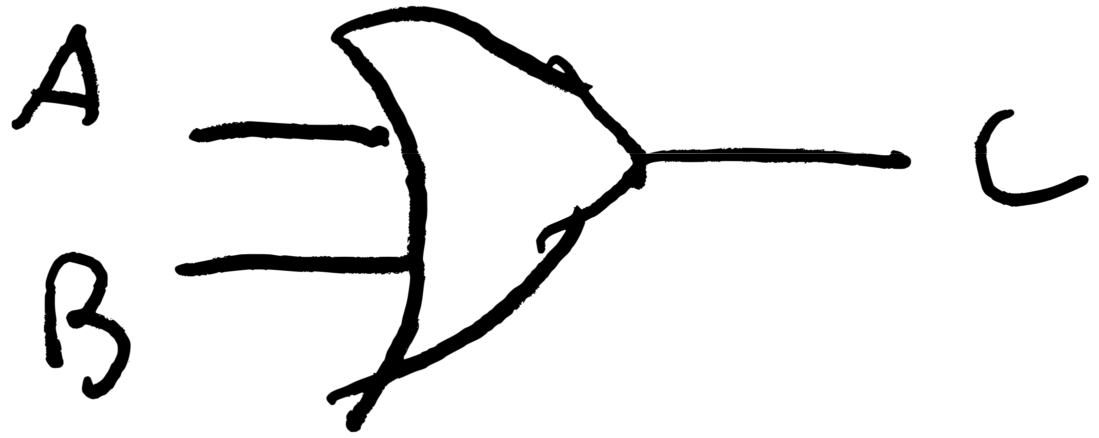
$C = \overline{A \cdot B}$

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

OR gate

OR logic

(||) matlab



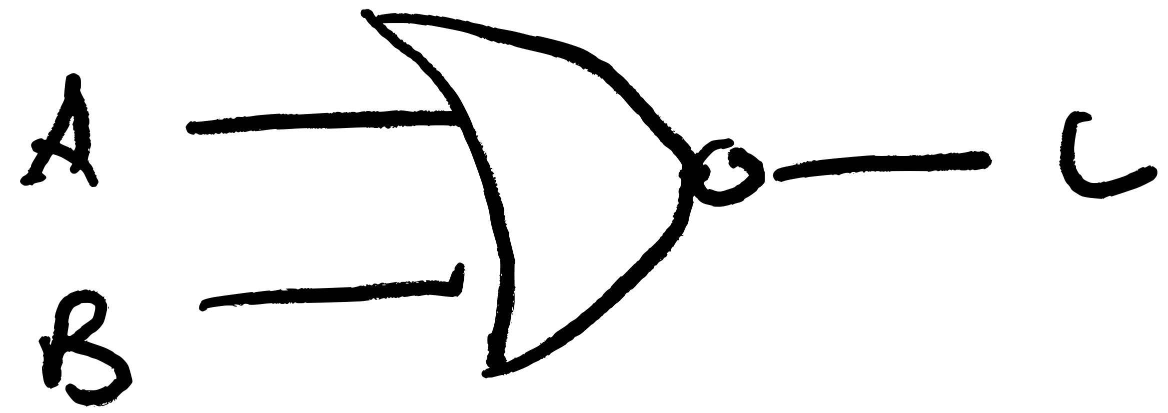
$C = A + B$

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

Combinational logic operations (gates)

NOR gate

Inverted OR

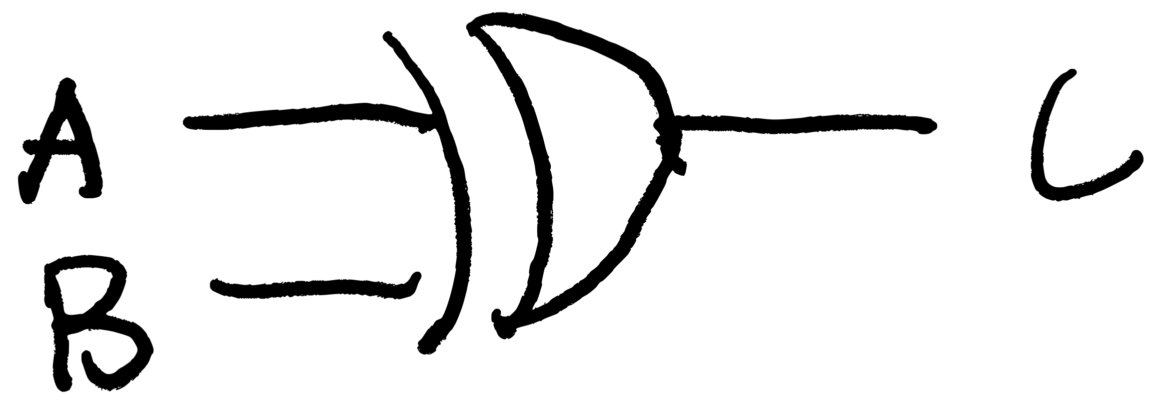


$$C = \overline{A+B}$$

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

XOR

Exclusive OR

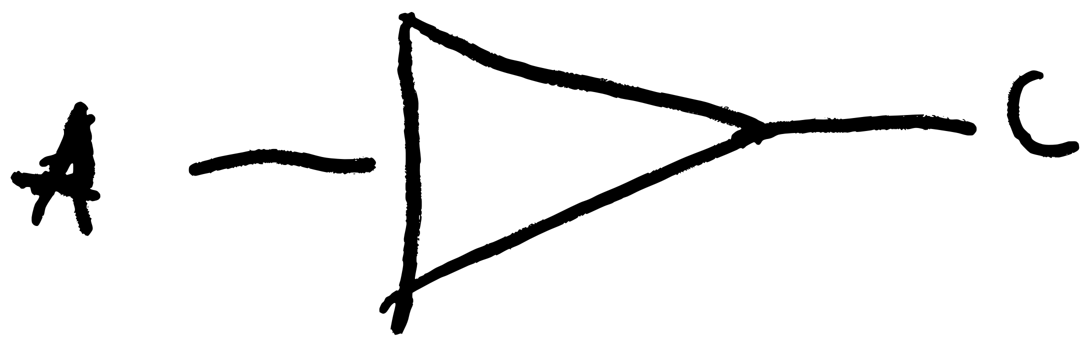


$$C = A \oplus B$$
$$= A \cdot \overline{B} + \overline{A} \cdot B$$

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

Buffer

Increases signal strength



$$C = A$$

A	C
0	0
1	1

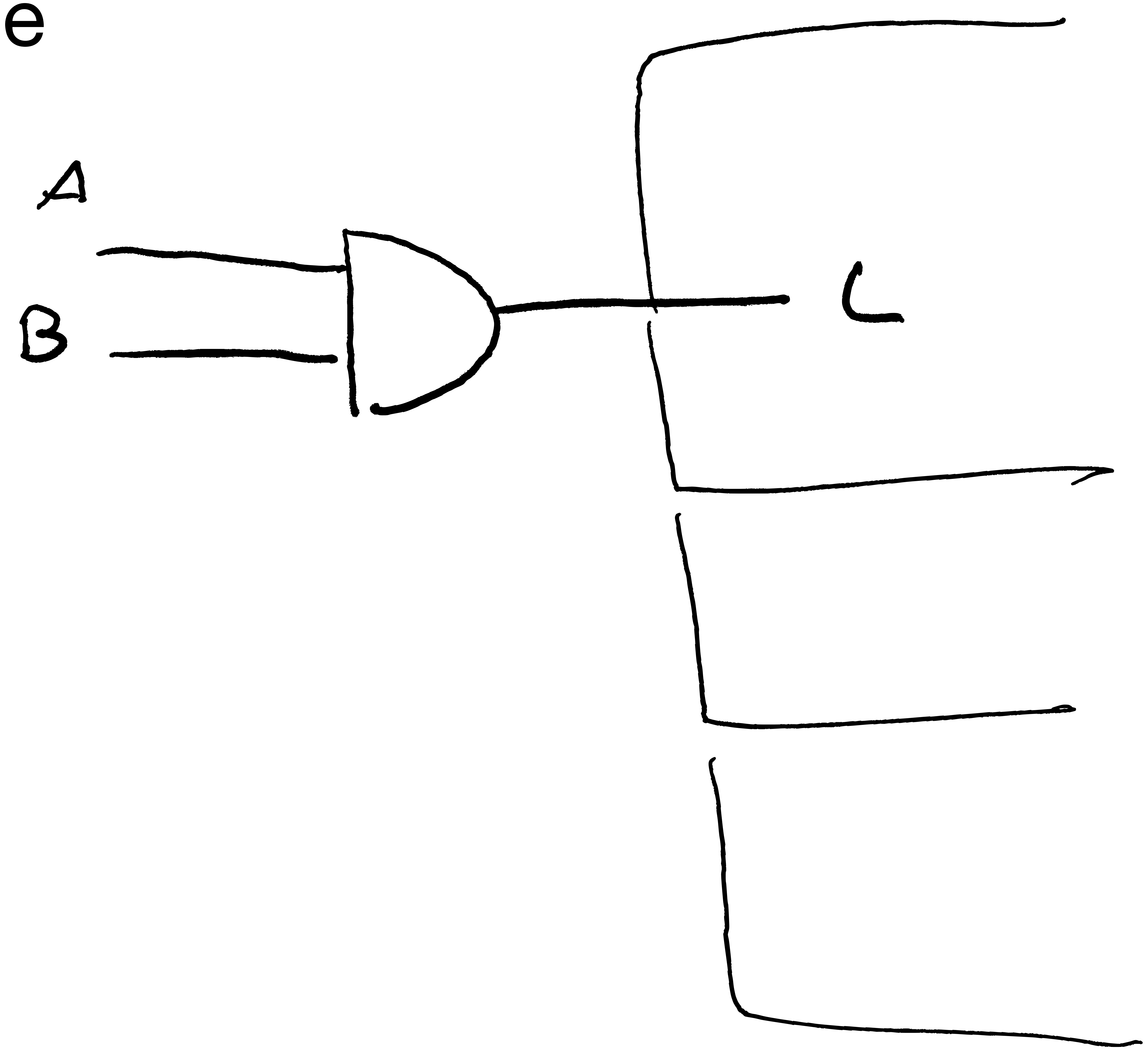
Notes on buffers

- used to increase current in logic circuits
- important to drive multiple inputs w/ single output

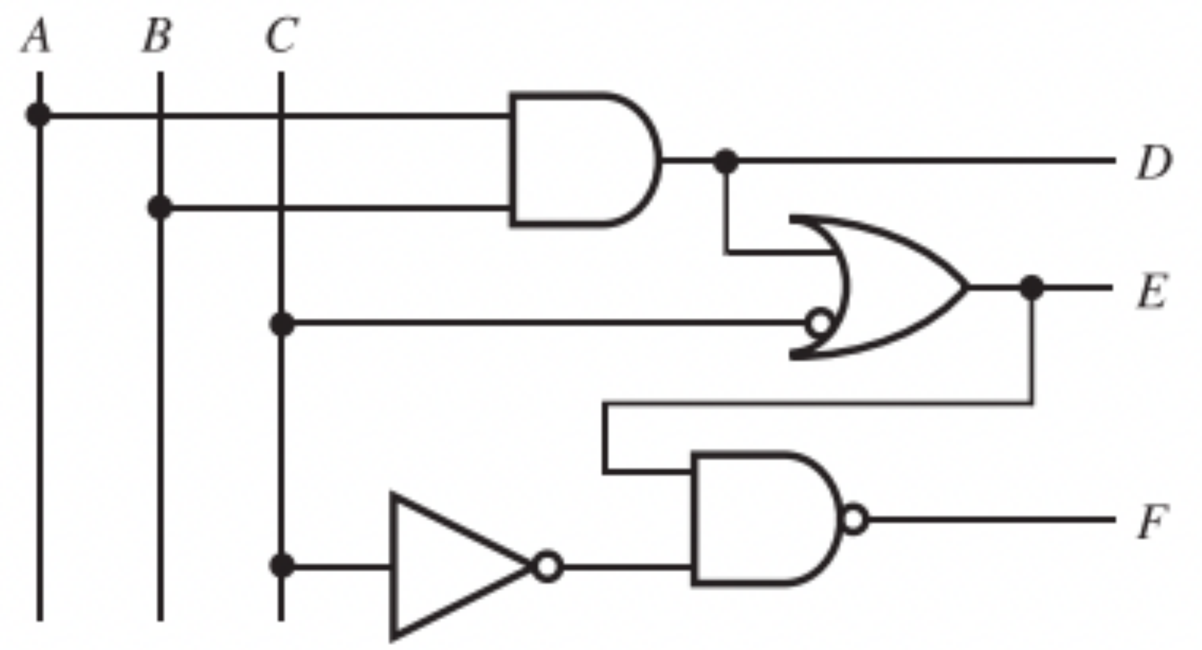
'fanout' $\longrightarrow$ # max of inputs driven by single output (~ 8)

- all logic circuits are manufactured as integrated circuits

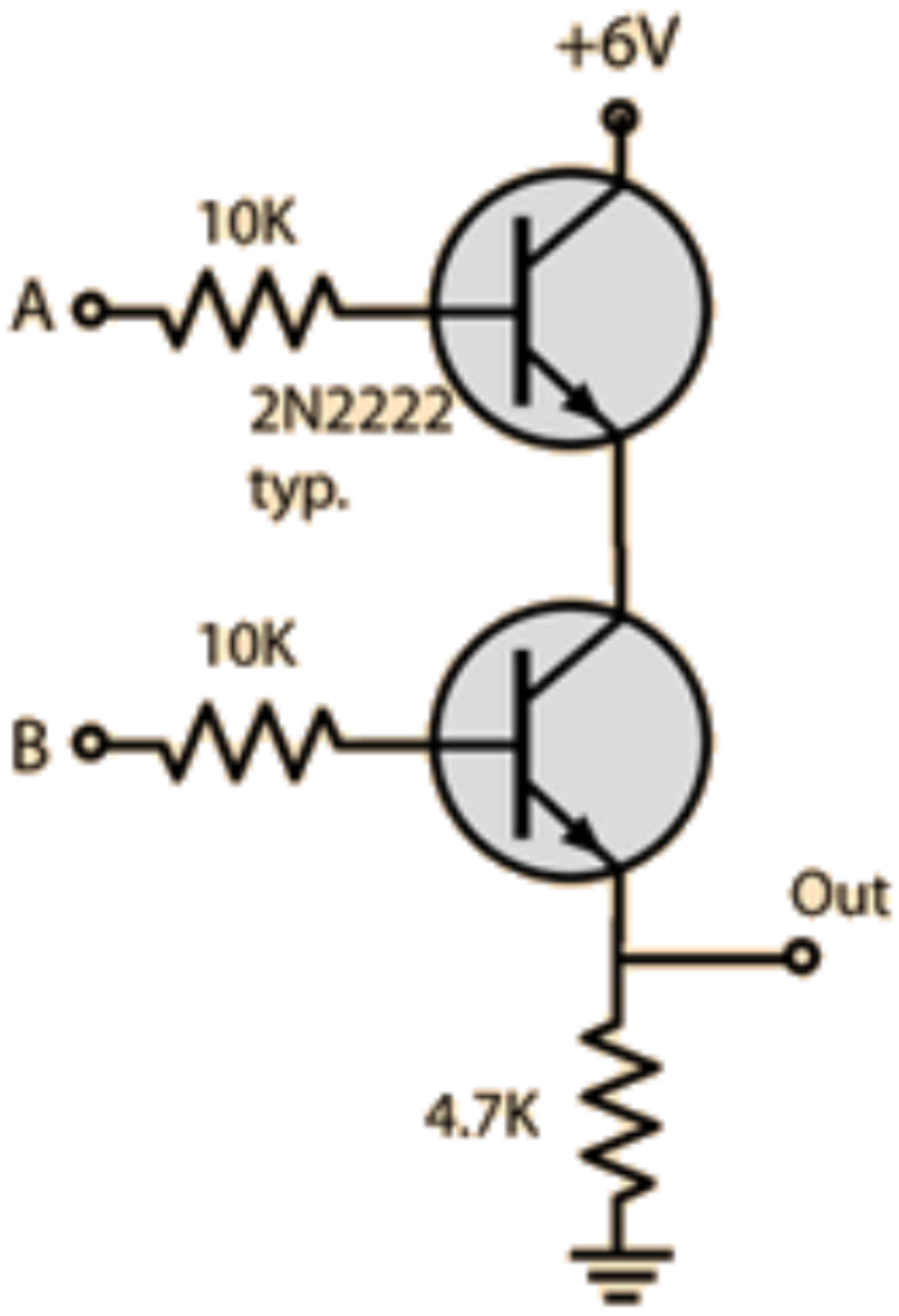
Example



Example

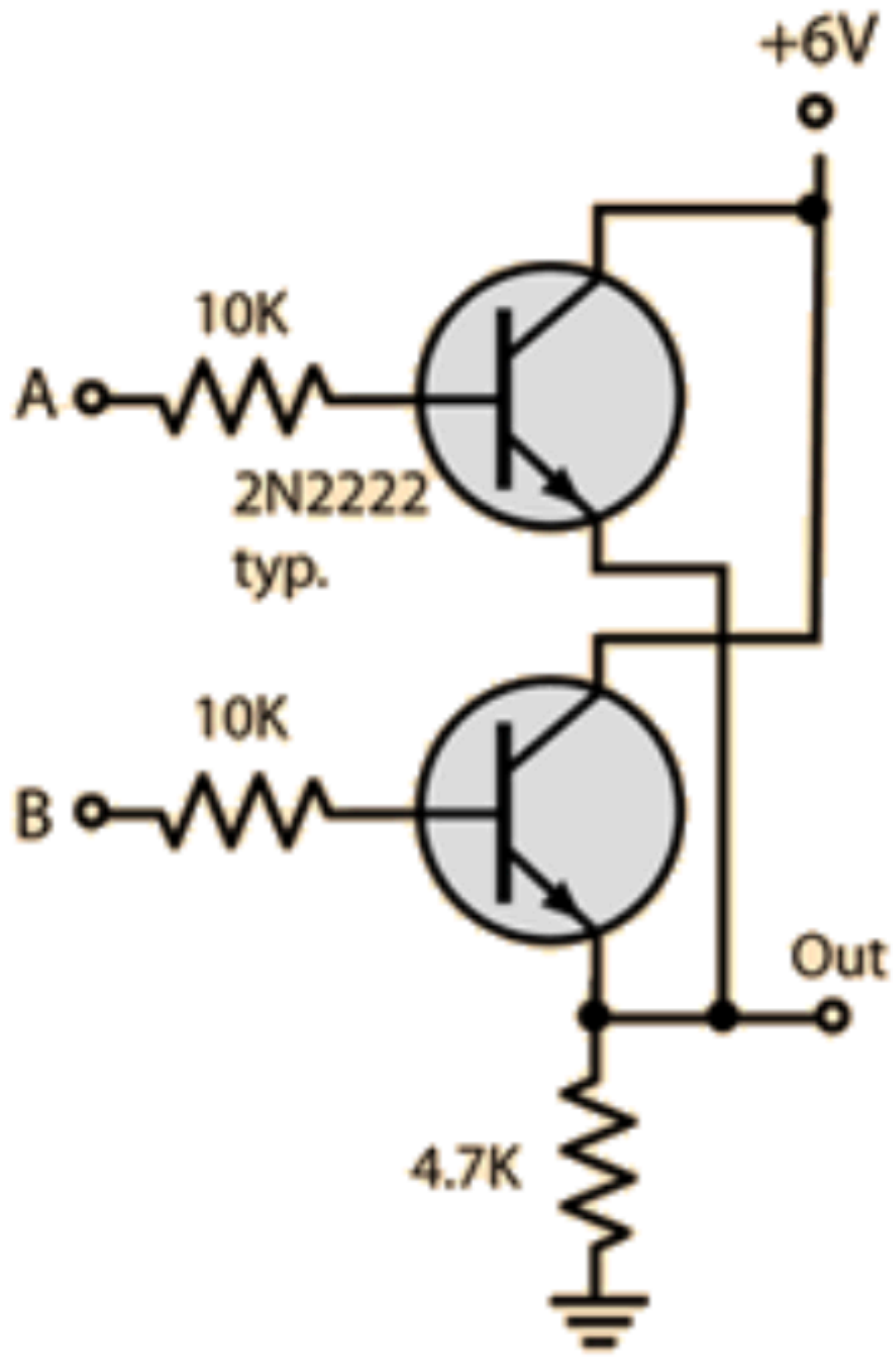


How to make hardware gates?



(a) AND gate

How to make hardware gates?



(b) OR gate