ME133 Lecture 12 2/23/23

Last time:

> Digital Signals
> Representation
> Combinational Logic

Today:

> Lab project

> Examples

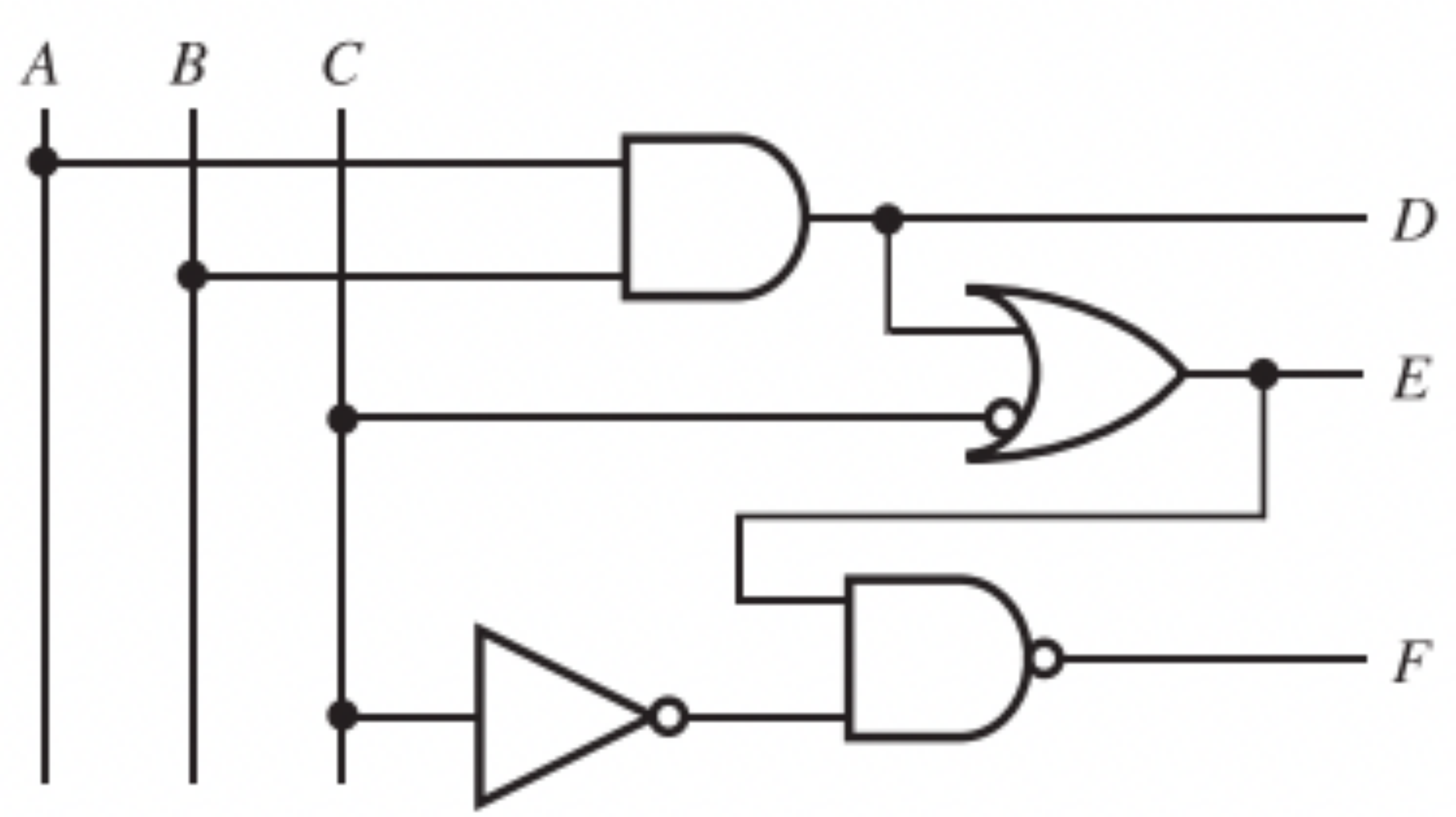
> Timing Diagrams

> Boolean Algebra

MakerSpace opens soon!

Please join the open house
if you’re available

Example



CMOS / transistors

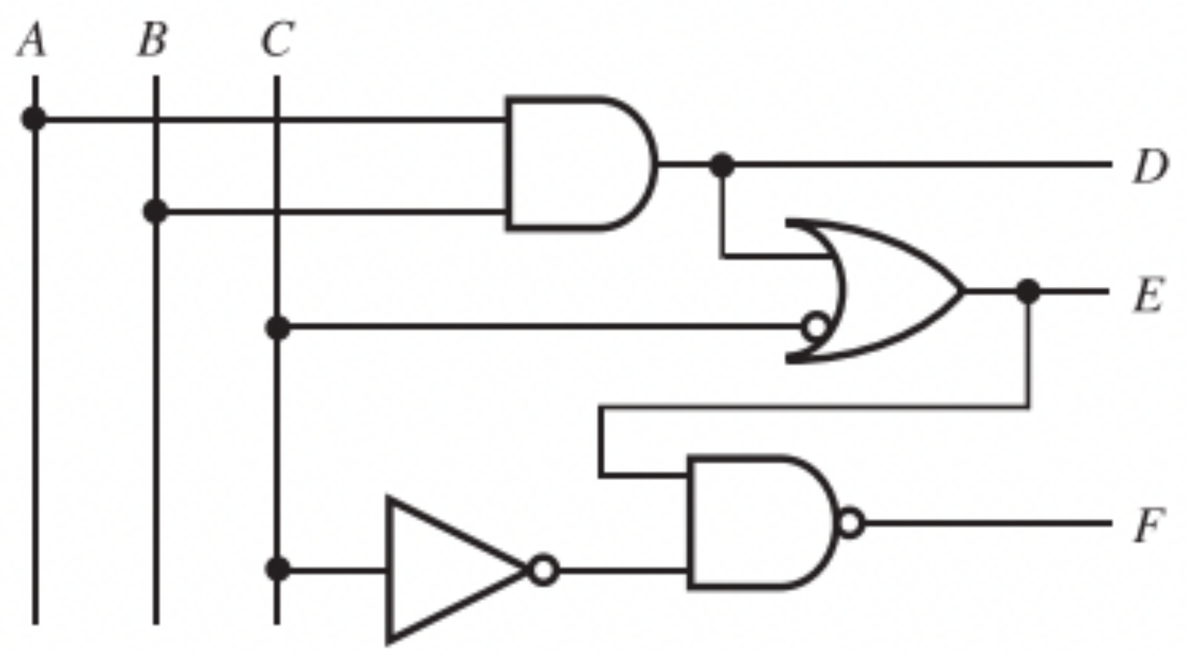
$$D = A \cdot B$$

$$E = D + \overline{C} = (A \cdot B) + \overline{C}$$

$$F = \overline{E \cdot \overline{C}}$$

$$= \overline{\left[(A \cdot B) + \overline{C}\right] \cdot \overline{C}}$$

Example



$$D = A \cdot B$$

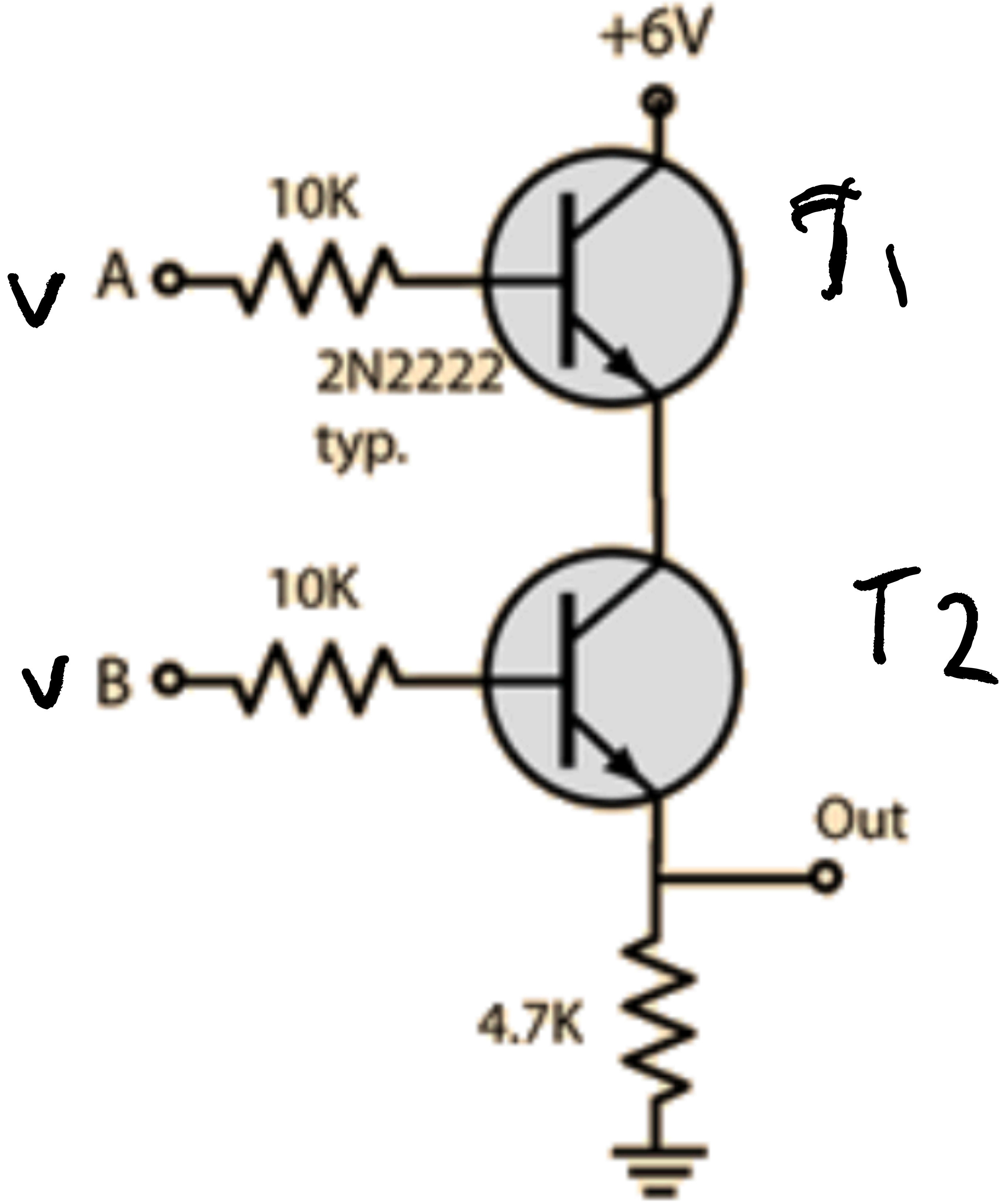
$$E = D + \overline{C} = (A \cdot B) + \overline{C}$$

$$F = \overline{E \cdot \overline{C}}$$

$$F = \overline{[(A \cdot B) + \overline{C}] \cdot \overline{C}}$$

A	B	C	D	E	F
0	0	0	0	1	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	1	1	0
1	1	1	1	1	1

How to make hardware gates?



(a) AND gate

$V_A = \text{Low} \Rightarrow T_1$ open

$V_C = \text{Low}$

$V_B = \text{Low} \Rightarrow T_2$ open

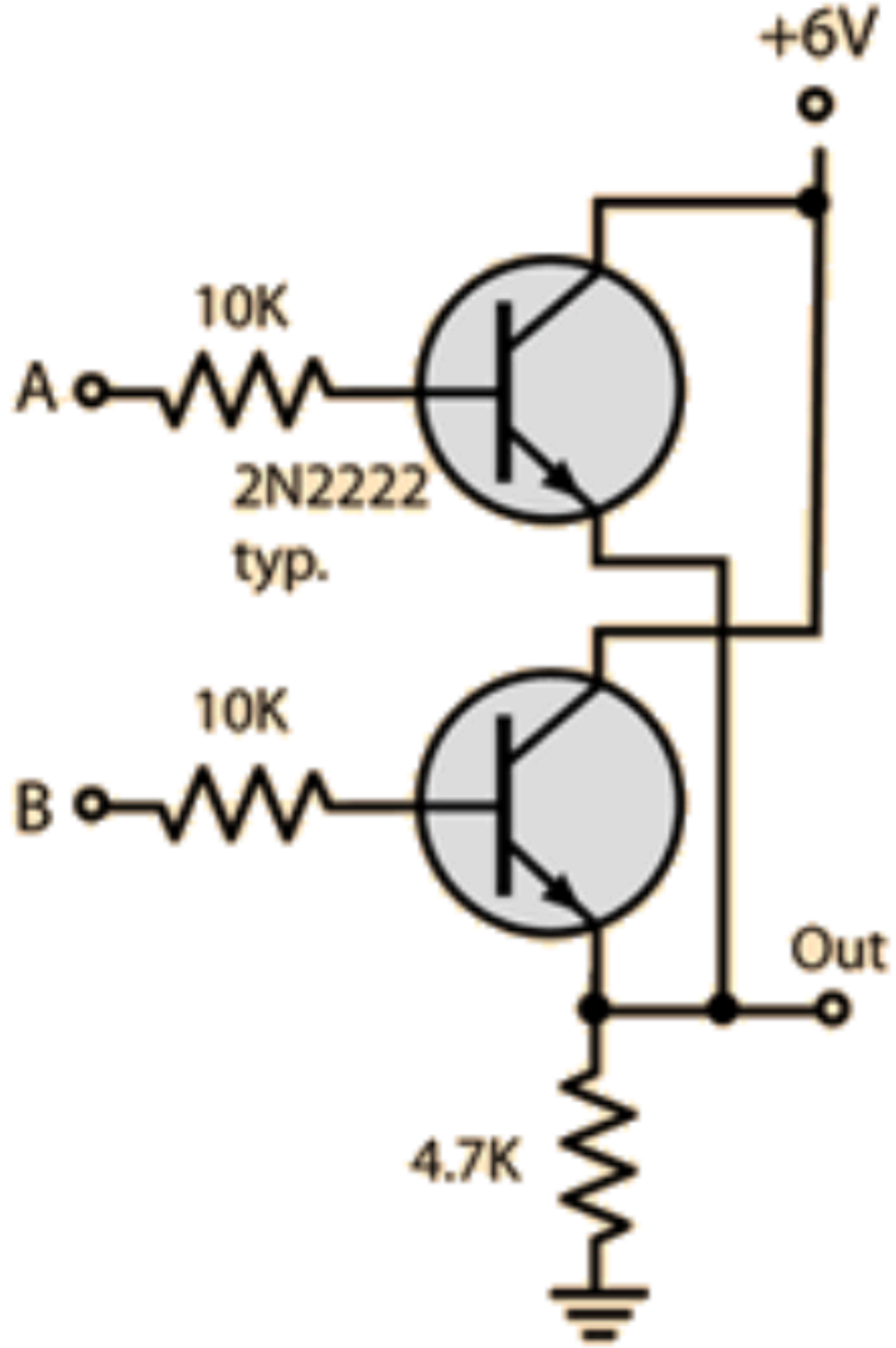
$V_C = \text{Low}$

iif $V_A = V_B = \text{High}$

$V_C = \text{High}$

V_A	V_B	V_C
0	0	0
1	0	0
0	1	0
1	1	1

How to make hardware gates?



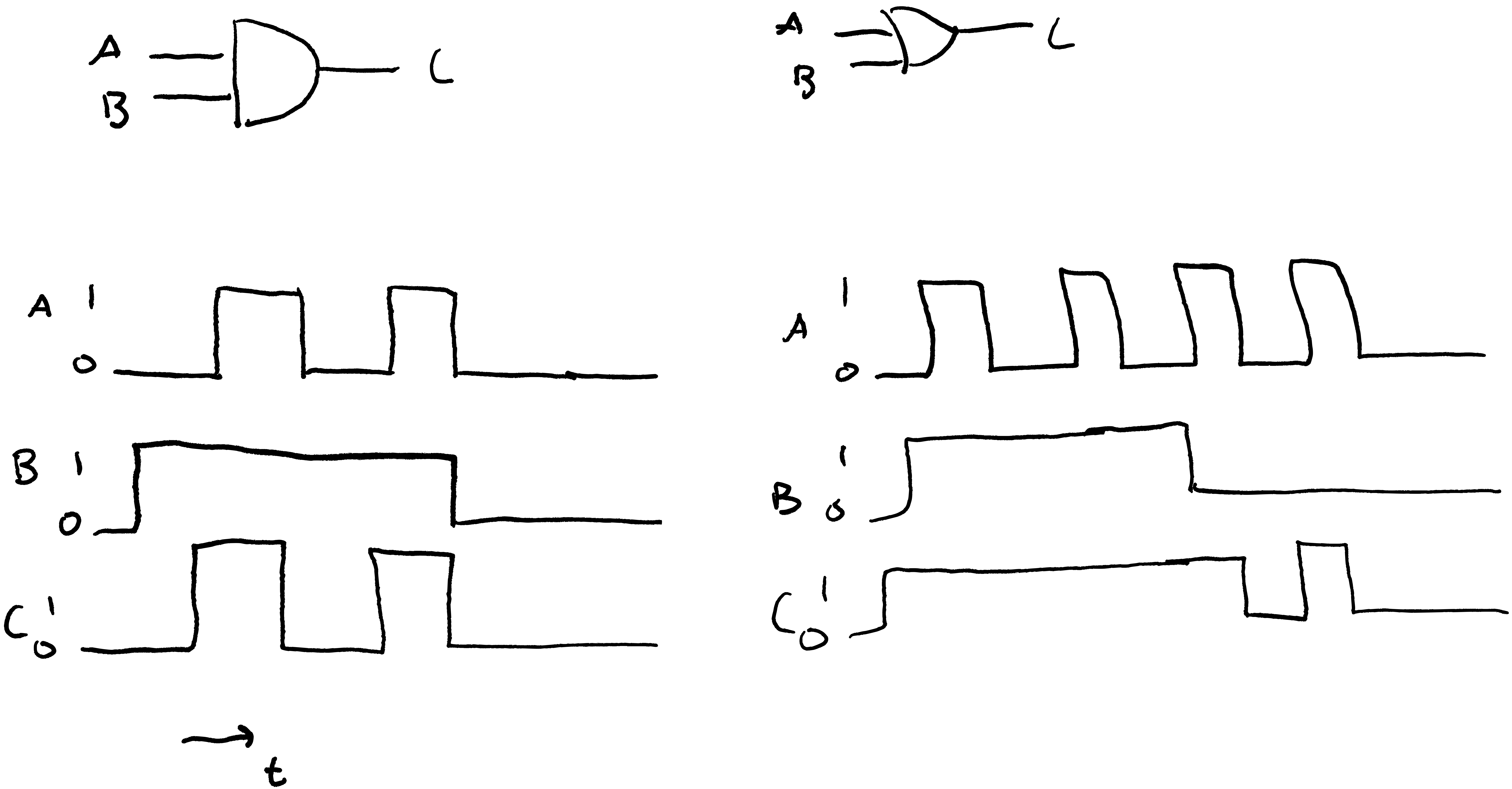
(b) OR gate

V_A = high
V_C = high

V_B = high
V_C = high

V_A	V_B	V_C
0	0	0
0	1	1
1	0	1
1	1	1

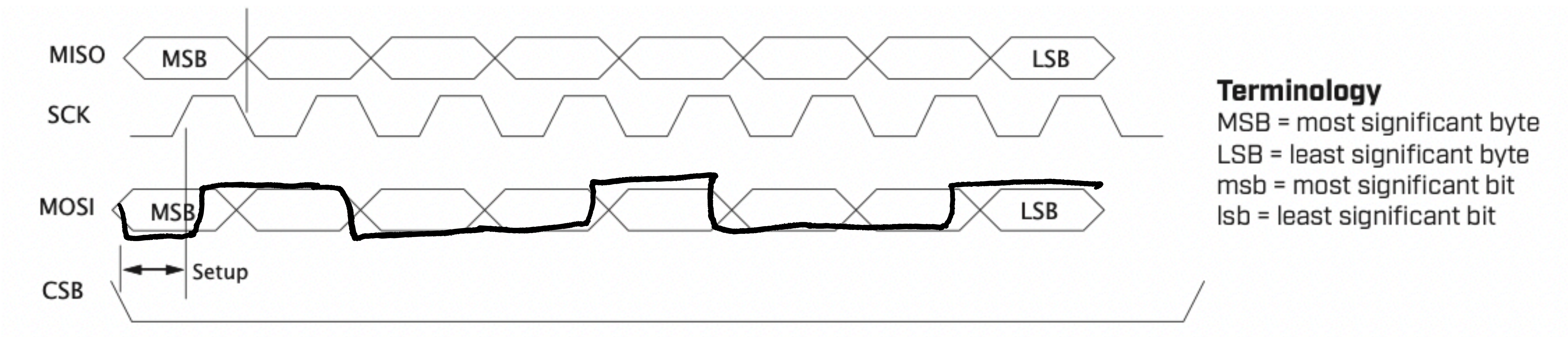
Timing Diagrams



'a' → 52 → 0100,1001

Timing Diagrams

SPI - serial peripheral Interface



Let's look at real timing diagrams from an encoder with SPI interface

Boolean Algebra

Boolean: binary, usually interpreted as 'TRUE' or 'FALSE'

Algebra: the study of variables & the rules for manipulating these variables in a formula

Boolean Algebra: rules for manipulating binary formulas

Three fundamental logic operators:

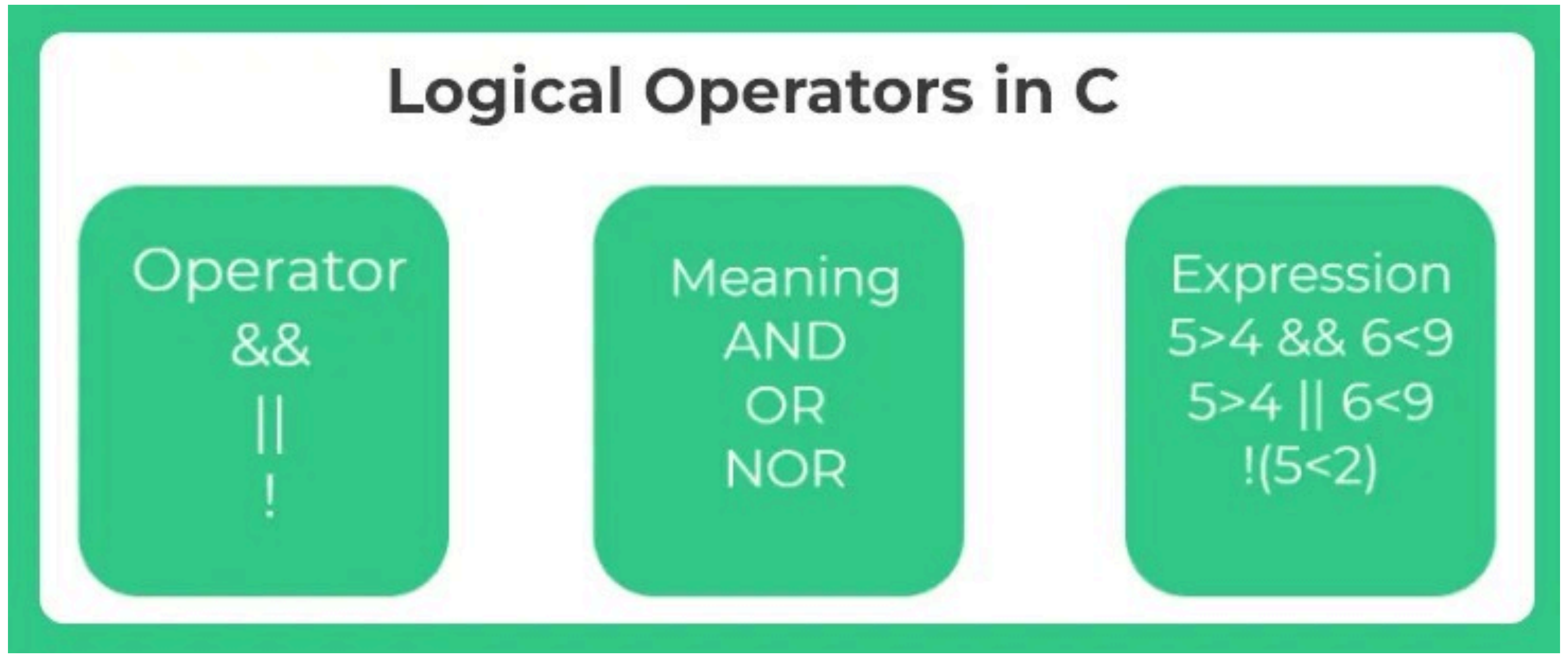
AND OR NOT

These can be realized with hardware gates or software (logical statements)

Boolean Algebra in Software

Logical Operators

<table>
<tr><th>Symbol</th><th>Role</th></tr>
<tr><td>&</td><td>Find logical AND</td></tr>
<tr><td>|</td><td>Find logical OR</td></tr>
<tr><td>&&</td><td>Find logical AND (with short-circuiting)</td></tr>
<tr><td>| |</td><td>Find logical OR (with short-circuiting)</td></tr>
<tr><td>~</td><td>Find logical NOT</td></tr>
</table>



$\underline{A + 0 = A}$

Fundamental Laws

$A = 0$ or 1

OR	AND	NOT
$\rightarrow$ $A + 0 = A$	$A \cdot 0 = 0$	
$A + 1 = 1$	$A \cdot 1 = A$	$\overline{\overline{A}} = A$
$A + A = A$	$A \cdot A = A$	
$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$	

Commutative, Associative and Distributive Laws

Commutative

$$A+B = B+A$$

$$A \cdot B = B \cdot A$$

Associative

$$(A+B)+C = A+(B+C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

Distributive

$$A \cdot (B+C)$$

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

$$A + (B \cdot C)$$

$$=$$

$$(A+B) \cdot (A+C)$$

De Morgan's Laws

Can convert ANDs to ORs or vice versa!

$$\overline{A+B+C\cdots} = \bar{A}\cdot\bar{B}\cdot\bar{C}\cdots$$

$$\overline{A\cdot B\cdot C\cdots} = \bar{A}+\bar{B}+\bar{C}\cdots$$

$$A+B+C\cdots = \overline{\bar{A}\cdot\bar{B}\cdot\bar{C}\cdots}$$

$$A\cdot B\cdot C\cdots = \overline{\bar{A}+\bar{B}+\bar{C}\cdots}$$

Other useful identities 6.17 - 6.24

Example $\quad \overset{1}{(A+\overline{A})} \cdot (A+B)$

Prove: $A + (\overline{A} \cdot B) = A + B$

Most straight forward way is to use truth table

A	$\overline{A}$	B	$\overline{A} \cdot B$	$A + (\overline{A} \cdot B)$	$A + B$
1	0	0	0	1	1
1	0	1	0	1	1
0	1	0	0	0	0
0	1	1	1	1	1

Example

Prove: $A + (\overline{A} \cdot B) = A + B$

Most straight forward way: use a truth table!

A	$\overline{A}$	B	$\overline{A} \cdot B$	$A + (\overline{A} \cdot B)$	$A + B$

Example: Simplifying a Boolean Expression

Use laws to collect like-terms — just like regular algebra!

$$\boxed{X = (A \cdot B \cdot C) + (B \cdot C) + (\overline{A} \cdot B)}$$

$\underline{7 \text{ operations}}$

→ associative

$$X = A \cdot (B \cdot C) + (B \cdot C) + (\overline{A} \cdot B)$$

→ distributive

$$X = (B \cdot C) \cdot (\overbrace{A + 1}^{1}) + (\overline{A} \cdot B)$$

$\underline{3 \text{ operations}}$

→ Fundamental laws $A + 1 = 1$

$$X = (B \cdot C) + (\overline{A} \cdot B) \longrightarrow \boxed{X = B \cdot (C + \overline{A})}$$

Example: Designing a logic network

Let's design a home alarm system using only logic gates

What we want:

1. Alarm to sound if window or doors are distrubed
   Mode-1 (sleeping)
2. Alarm to sound if " " " or
   motion is detect in house Mode-2 (vacation)
3. A disabled state where alarm is off mode-3 (off)

Assumptions:

That sensors are binary (motion - 1, no motion - 0
door/window is distrubed - 1, no distrubance
0

Example: Designing a logic network

Let's define our Boolean variables:

- A : state of door / windows
- B : state of motion dector
- Y : output to sound alarm (1 — alarm on, 0 - no sound)
- CD : 2-bit code

$$CD = \begin{cases} 0\ 1 & \text{Mode 1} \\ 1\ 0 & \text{Mode 2} \\ 0\ 0 & \text{mode 3} \end{cases}$$

Example: Designing a logic network

- *A*: state of the door and window sensors
- *B*: state of the motion detector
- *Y*: output used to sound the alarm
- *C D*: 2-bit code set by the user to select the operating state defined by

$$C\,D = \begin{cases} 0\,1 & \text{operating state 1} \\ 1\,0 & \text{operating state 2} \\ 0\,0 & \text{operating state 3} \end{cases}$$

Quasi-logic Statement

Activate alarm ($Y=1$) if $A=1$ <u>and</u> $CD=01$ <u>or</u>

$A=1$ <u>or</u> $B=1$ <u>and</u> $CD=10$

Example: Designing a logic network

Translate Quasi-logic Statement into Boolean Expression

Activate alarm (Y=1) if A=1 and CD = 01 or activate the alarm if A = 1 or B = 1 and CD = 10

$$\boxed{Y = A \cdot (\overline{C} \cdot D) + (A + B) \cdot (C \cdot \overline{D})}$$

simply!

Example: Designing a logic network

Simplify $$Y = A \cdot (\overline{C} \cdot D) + (A + B) \cdot (C \cdot \overline{D})$$

C	D	$(\overline{C} \cdot D)$	$(C \cdot \overline{D})$
0	0	0	0
1	0	0	1
0	1	1	0

$(\overline{C} \cdot D) = D$ and $(C \cdot \overline{D}) = C$

$$Y = (A \cdot D) + (A + B) \cdot C$$

Example: Designing a logic network

Convert to single gate: $Y = (A \cdot D) + (A + B) \cdot C$

Example: Designing a logic network

Now we can draw logic circuit:

$$Y = \overline{\overline{A \cdot D} \cdot \overline{(\overline{A} \cdot \overline{B})} \cdot C}$$