ECE 205 “Electrical and Electronics Circuits”

Spring 2024 – LECTURE 30

MWF – 12:00pm

Prof. Umberto Ravaioli

2062 ECE Building

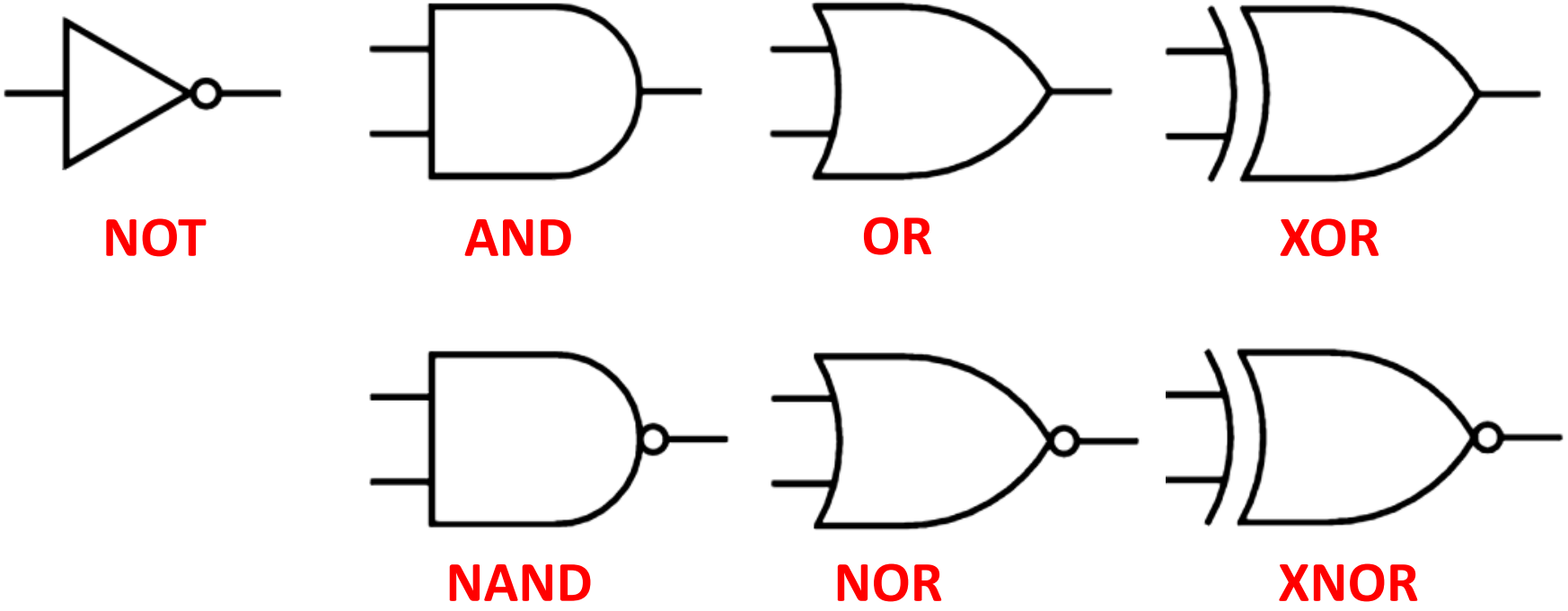
Lecture 30 – Summary

Learning Objectives

1. Boolean algebra
2. Logic network minimization practice
3. Universal gates combine to realize any other logic function

Logic Operations

Binary logic is based on a set of seven elementary logical operations with two inputs and one output. The elements which accomplish these operations are called "Logic Gates". They are represented with the symbols below in a logic circuit.



Boolean Algebra

Logic operations can be represented with formulas, using a special formalism called Boolean Algebra. The following table shows the Boolean notation.

OPERATOR	BOOLEAN ALGEBRA
NOT	$Y = \overline{A}$
AND	$Y = A\,B$
OR	$Y = A + B$
NAND	$Y = \overline{A\,B}$
NOR	$Y = \overline{A + B}$
XOR	$Y = A \oplus B$
XNOR	$Y = \overline{A \oplus B}$

NOTE: Some authors use $A.B$ for $A\,B$ and A' for $\overline{A}$

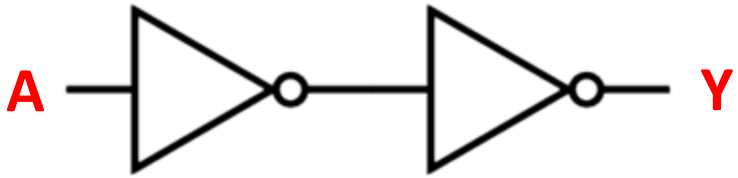
Boolean Algebra Simplifications Table

When a logic circuit is designed to obtain the desired behavior, it can be simplified by using the following laws to minimize the number of gates.

LAWS	AND	OR
Identity	$1\,A = A$	$0 + A = A$
Null	$0\,A = 0$	$1 + A = 1$
Idempotent	$A\,A = A$	$A + A = A$
Inverse (Complement)	$A\,\overline{A} = 0$	$A + \overline{A} = 1$
Commutative	$A\,B = B\,A$	$A + B = B + A$
Associative	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption	$A(A + B) = A$	$A + AB = A$ $A + \overline{A}B = A + B$

Involution Law

$$\overline{\overline{A}} = A$$



AND VERY IMPORTANT:

De Morgan Theorem

1) $$\overline{A + B} = \overline{A}\,\overline{B}$$

2) $$\overline{A\,B} = \overline{A} + \overline{B}$$

Example 1

Apply Distributive Law

$$AB(\overline{B}C + AC)$$

$$AB\overline{B}C + ABAC$$

Apply Commutative Law

$$AB\overline{B}C + AABC$$

Apply Idempotent Law

$$AB\overline{B}C + ABC$$

Apply Inverse Law
(a.k.a Complement Law)

$$A0C + ABC$$

Apply Null Law

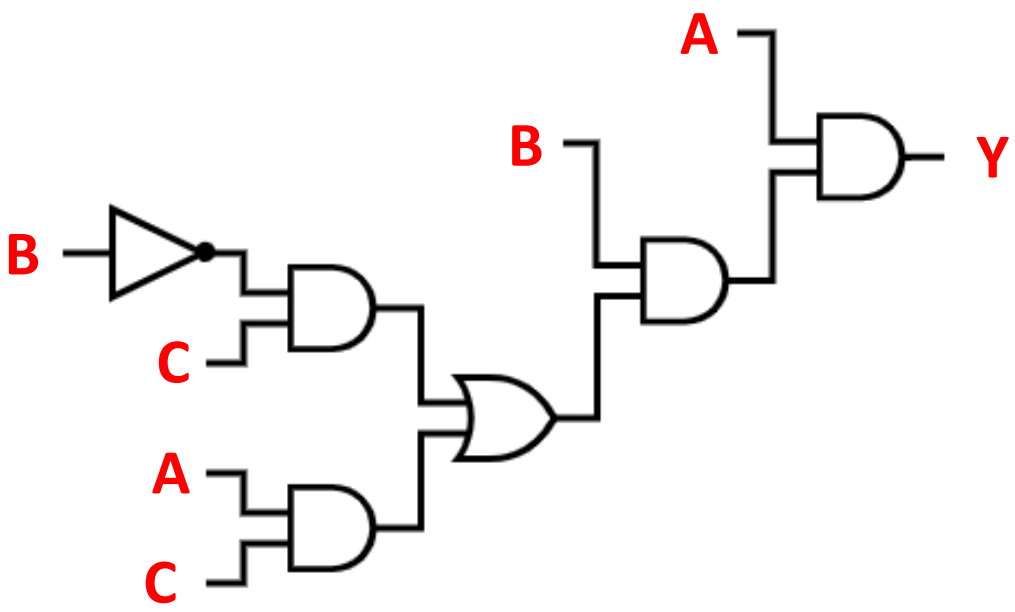
$$0 + ABC$$

Apply Identity Law

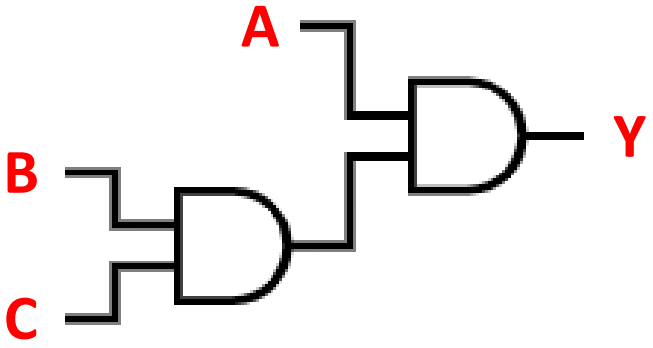
$$ABC$$

Example 1

$$\mathbf{AB(\overline{B}C + AC)}$$

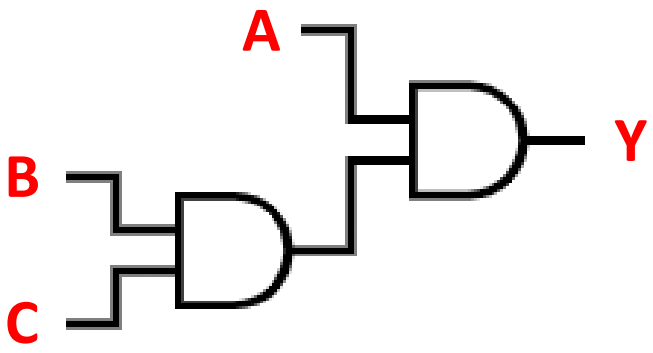


$$\mathbf{ABC}$$



Example 1

ABC



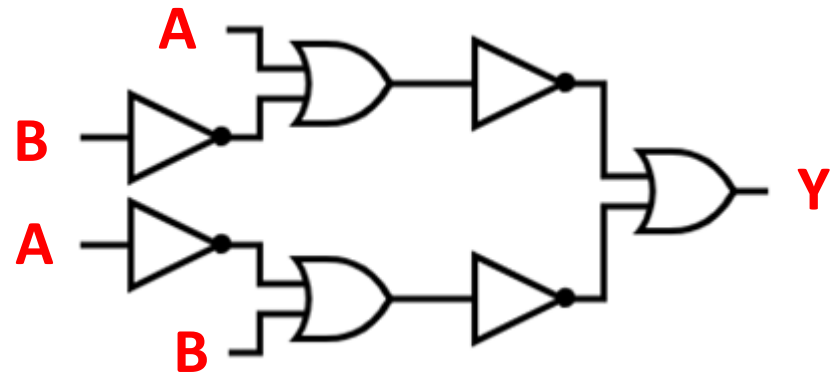
TRUTH TABLE

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

This is a 3-inputs AND. Only when all inputs are TRUE, the output is TRUE.

Example 2

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



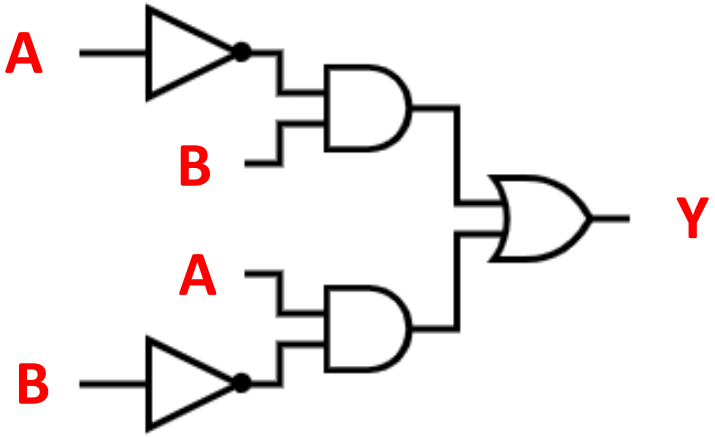
Apply De Morgan Theorem on both terms

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

$$\overline{A}\,\overline{\overline{B}} + \overline{\overline{A}}\,\overline{B}$$

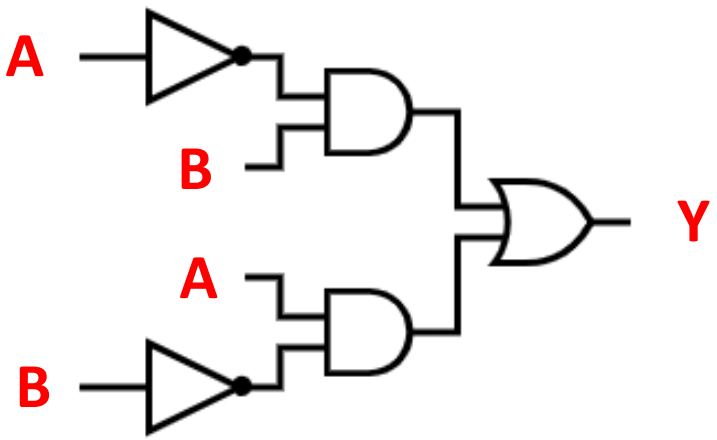
Apply Involution Law

$$\overline{A}\,B + A\,\overline{B}$$



Example 2

$$\overline{A}\,B + A\,\overline{B}$$



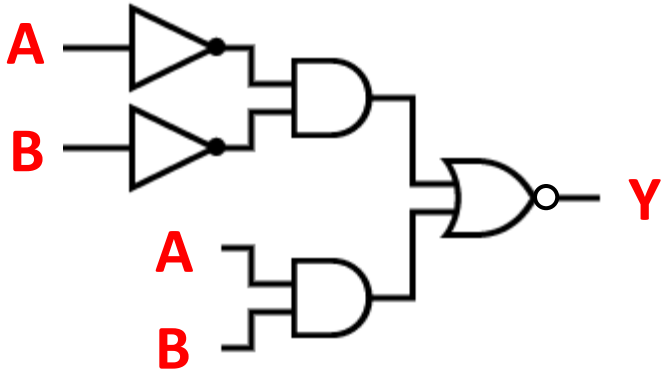
TRUTH TABLE

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

This is the Truth Table of the XOR

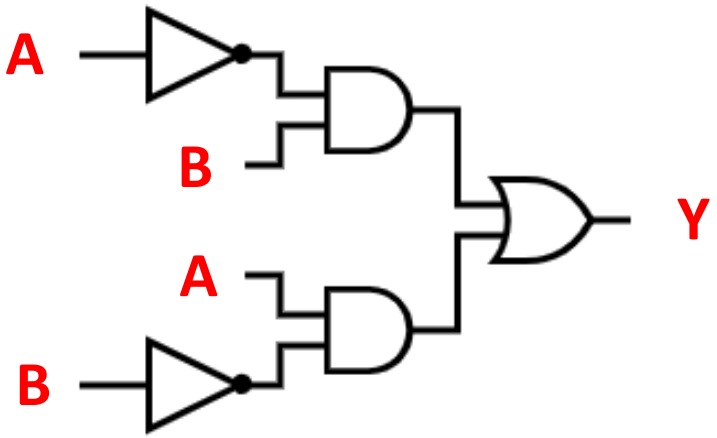
An equivalent realization giving the same truth table

$$\overline{\overline{A}\,\overline{B} + AB}$$



Example 2 Other equivalent circuits

$$\overline{A}\,B + A\,\overline{B}$$

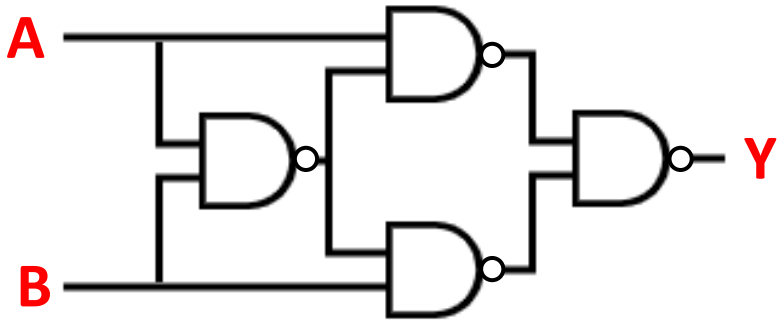


TRUTH TABLE

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

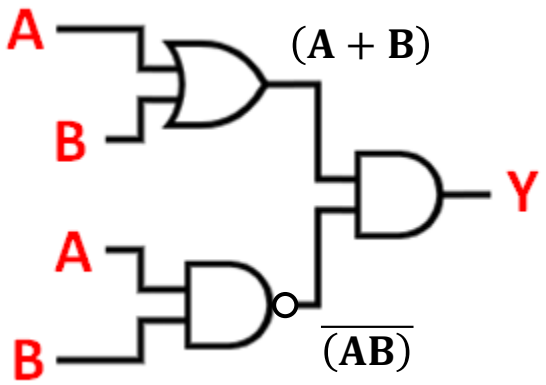
This is the Truth Table of the XOR

Realization only with NAND Gates



$$\overline{\left(A\,\overline{(AB)}\right)}\;\overline{\left(B\,\overline{(AB)}\right)}$$

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



Example 2

Prove

$$(A + B)\overline{(AB)} \Rightarrow \bar{A}\,B + A\,\bar{B}$$

Apply De Morgan Theorem

$$\overline{A\,B} = \bar{A} + \bar{B}$$

$$(A + B)(\bar{A} + \bar{B})$$

Apply Distribution Law

$$(\bar{A} + \bar{B})A + (\bar{A} + \bar{B})B$$

Apply Distribution Law

$$A\bar{A} + A\bar{B} + \bar{A}B + B\bar{B}$$

Apply Inverse Law

$$A\,\bar{A} = 0$$

$$0 + A\bar{B} + \bar{A}B + 0$$

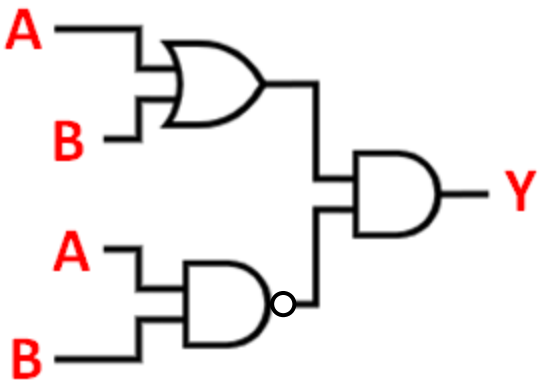
Apply Identity Law

$$0 + A = A$$

$$\bar{A}\,B + A\,\bar{B}$$

Example 2

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



XOR circuit realization with BJT

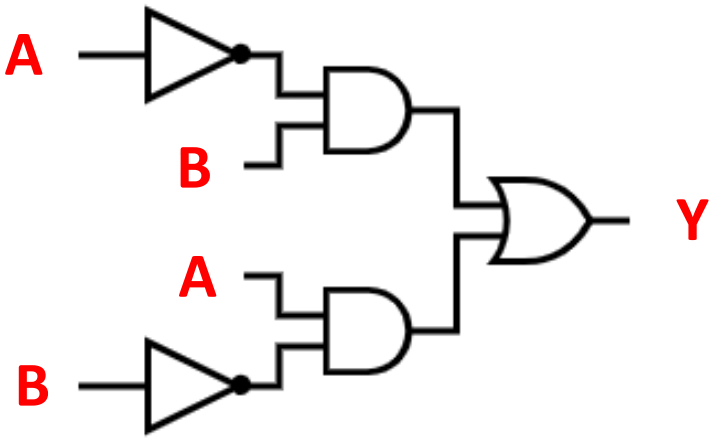
$V_{CC} \approx 5V$

OUTPUT: LED LIGHT with integrated current limiting resistor

http://sullystationtechnologies.com/npnxorgate.html

Example 2

$$\overline{A}\,B + A\,\overline{B}$$



TRUTH TABLE

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

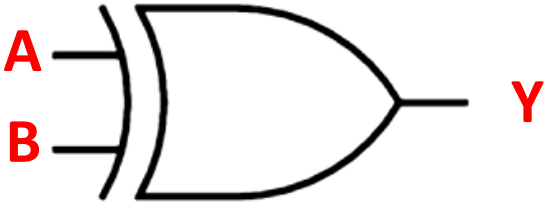
This is the Truth Table of the XOR

$$Y = A \oplus B$$

We have just designed one possible logic circuit to operate a light with two switches

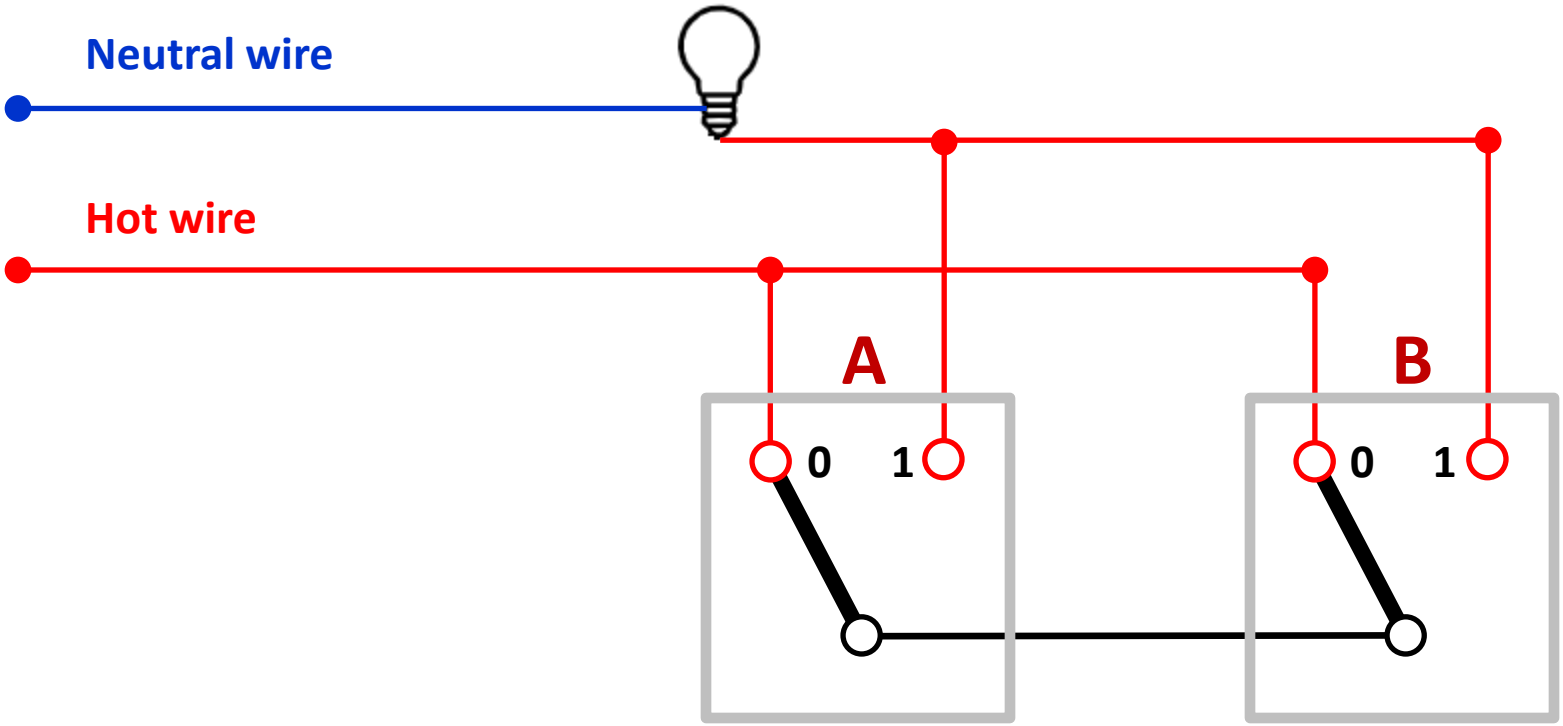
Example 2

Here is how an electrician implements the wiring of XOR with two-way switches



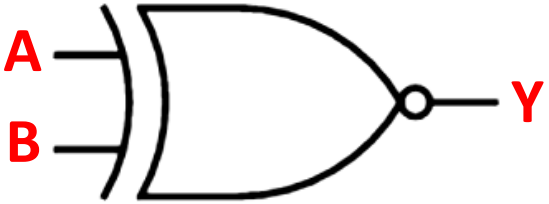
TRUTH TABLE

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



Example 2

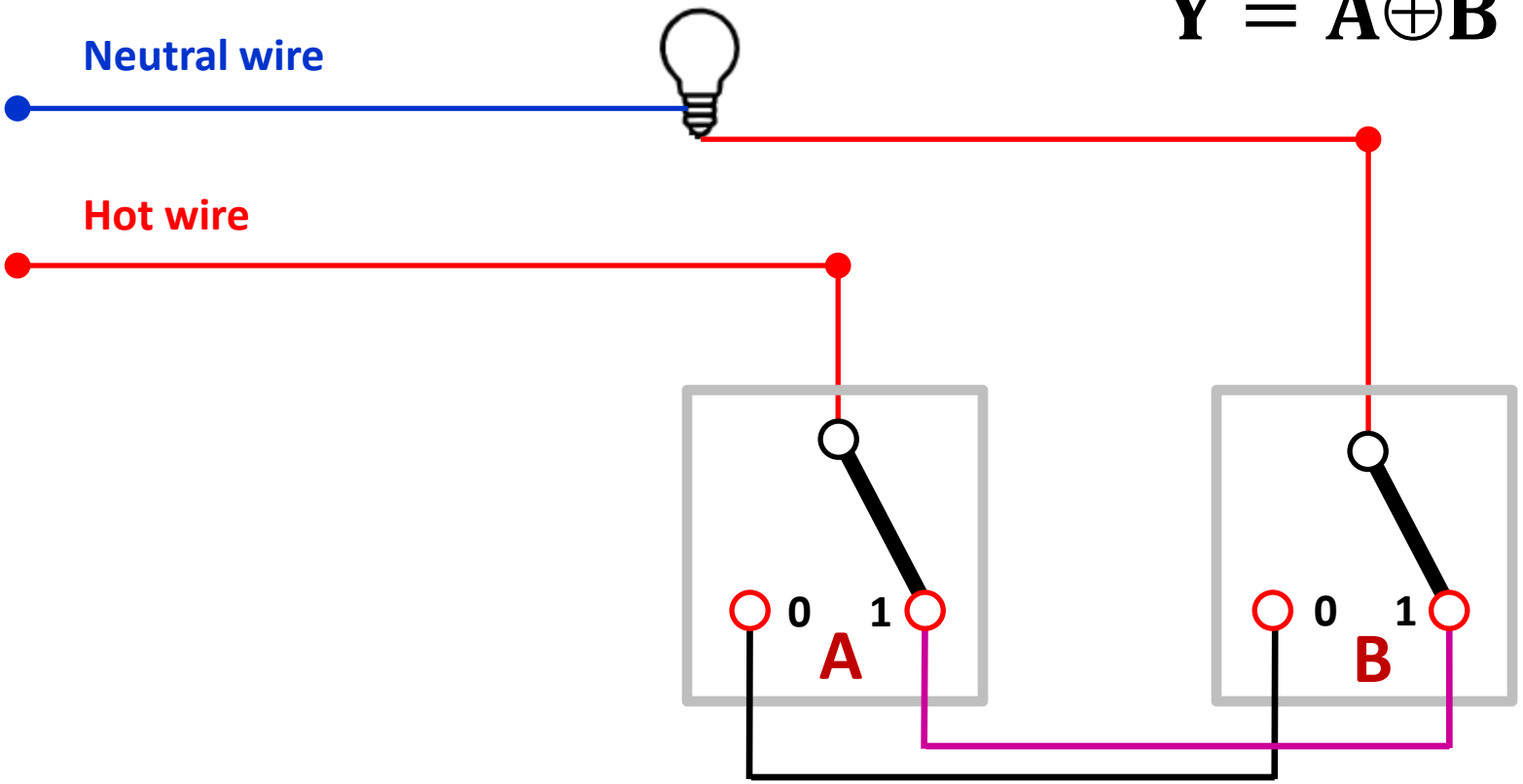
Here is how an electrician implements the wiring of XNOR with two-way switches



TRUTH TABLE

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

$$Y = \overline{A \oplus B}$$



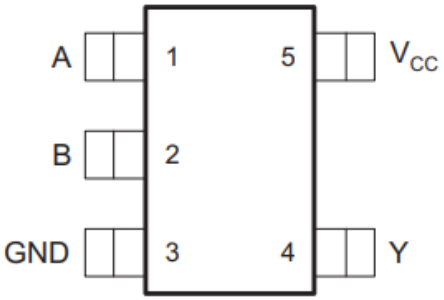
Example 2 XOR

TEXAS INSTRUMENTS

www.ti.com

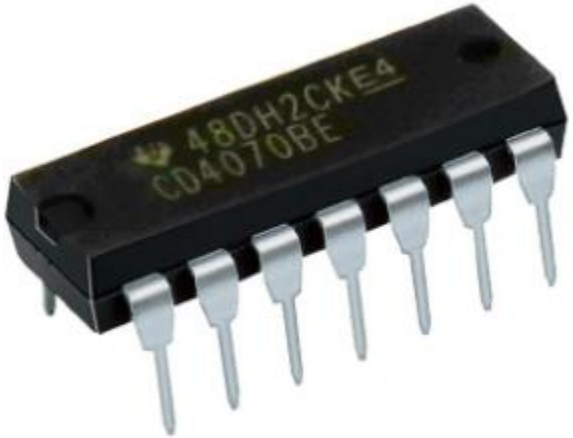
5 Pin Configuration and Functions

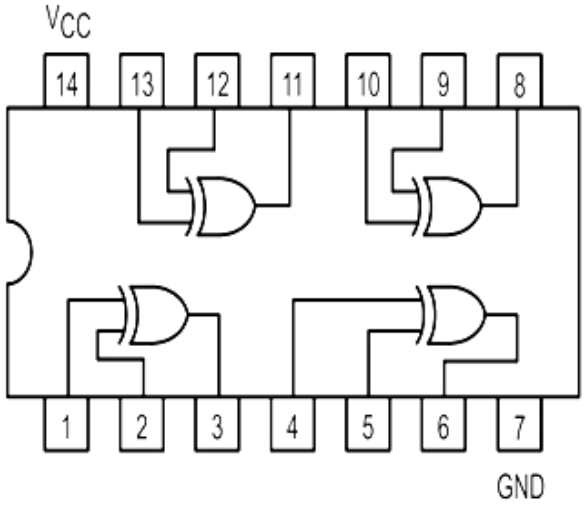
DBV Package
5-Pin SOT-23
Top View



TRUTH TABLE

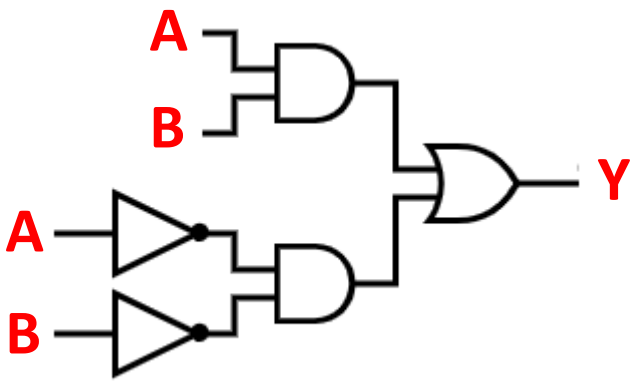
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0





Example 2 XNOR

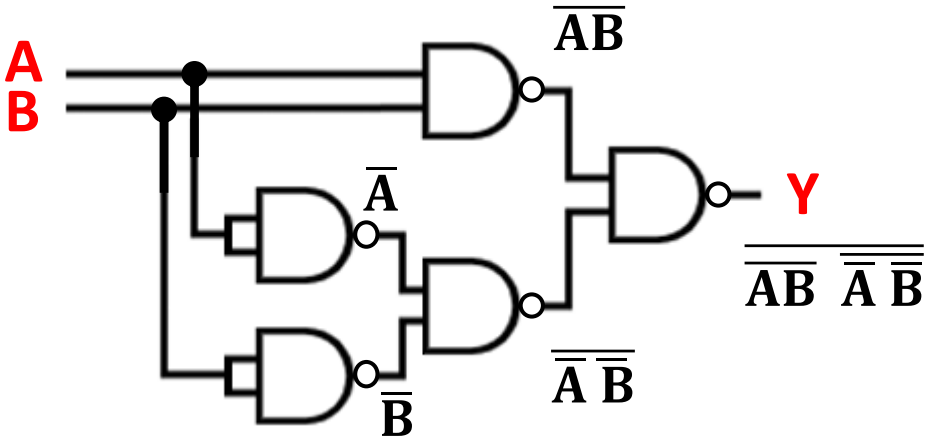
$$Y = \overline{A \oplus B} = AB + \overline{A}\,\overline{B}$$



TRUTH TABLE

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Realization only with NAND Gates



$$\overline{\overline{AB}\ \overline{A}\,\overline{B}}$$

De Morgan Theorem

$$\overline{\overline{AB}} + \overline{\overline{A}\,\overline{B}}$$

Involution Law

$$AB + \overline{A}\,\overline{B}$$

Example 3

$$(A + \bar{B} + \bar{C})(A + \bar{B} + C)\,(A + B + \bar{C})$$

Apply Distributive Law

$$A(A + \bar{B} + C)\,(A + B + \bar{C}) +$$ Apply Absorption Law Twice

$$\bar{B}(A + \bar{B} + C)\,(A + B + \bar{C}) +$$ Apply Absorption Law

$$\bar{C}(A + \bar{B} + C)\,(A + B + \bar{C})$$ Apply Absorption Law

$$A +$$
$$\bar{B}\,(A + B + \bar{C}) +$$
$$\bar{C}(A + \bar{B} + C)$$

⇒

Apply Inverse Law $A\,\bar{A} = 0$

$$A +$$
$$(\bar{B}A + \bar{B}B + \bar{B}\bar{C}) +$$
$$(\bar{C}A + \bar{C}\bar{B} + \bar{C}C)$$

Apply Distributive Law

Example 3

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

All OR operations, so parentheses can go away

Idempotent Law $A + A = A$

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

Absorption Law

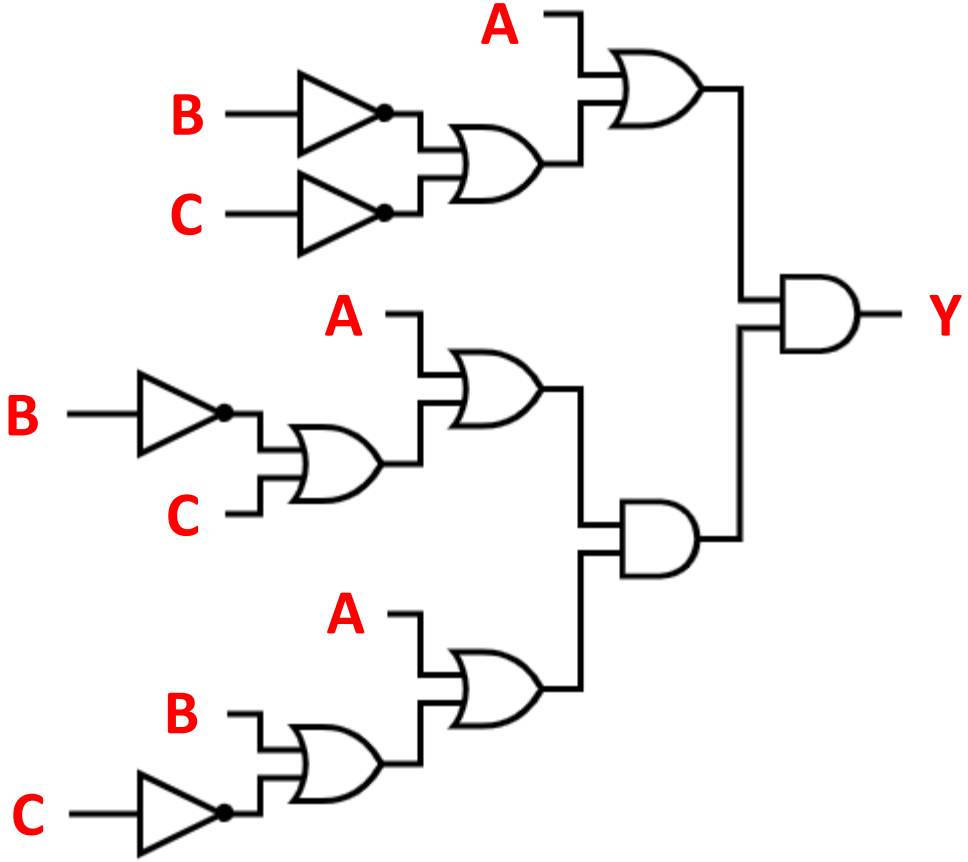
$$A + AB = A$$

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

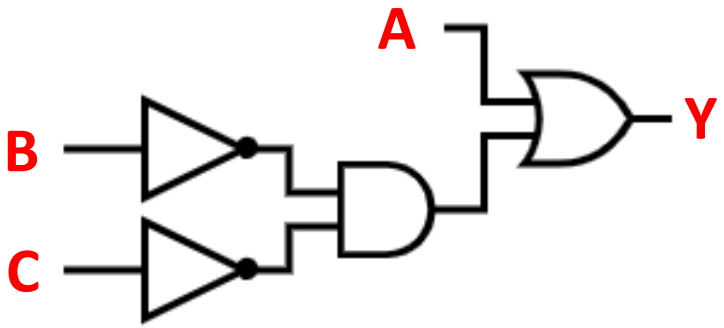
$$\underbrace{A + A\overline{C}} + \overline{B}\overline{C}$$

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

Example 3 $(A + \overline{B} + \overline{C})(A + \overline{B} + C)\ (A + B + \overline{C})$

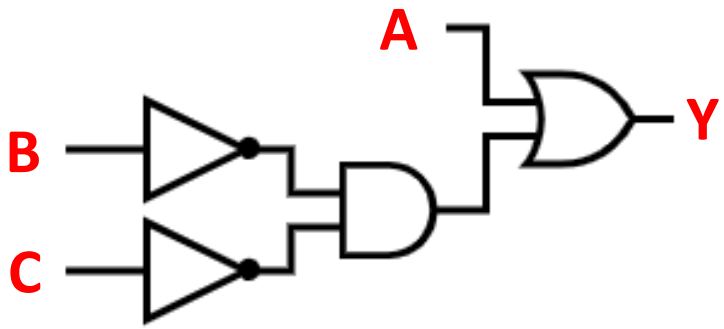


$\Rightarrow$ $A + \overline{B}\overline{C}$



Example 3

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



TRUTH TABLE

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

Example 4

$$B\,C + (C\,\overline{D} + A\,B)\,D$$

Apply Distributive Law

$$B\,C + (D\,C\,\overline{D} + D\,A\,B)$$

Apply Inverse (Complement) Law

$$A\,\overline{A} = 0$$

$$B\,C + 0\,C + D\,A\,B$$

Apply Null Law $0\,A = 0$

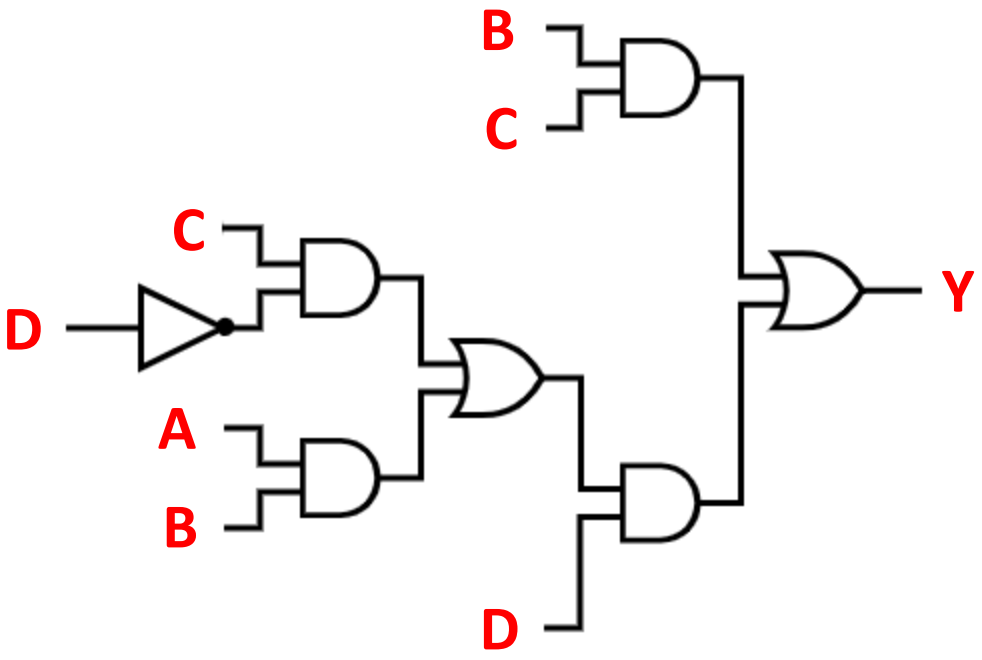
$$B\,C + 0 + D\,A\,B$$

Apply Identity Law $0 + A = A$

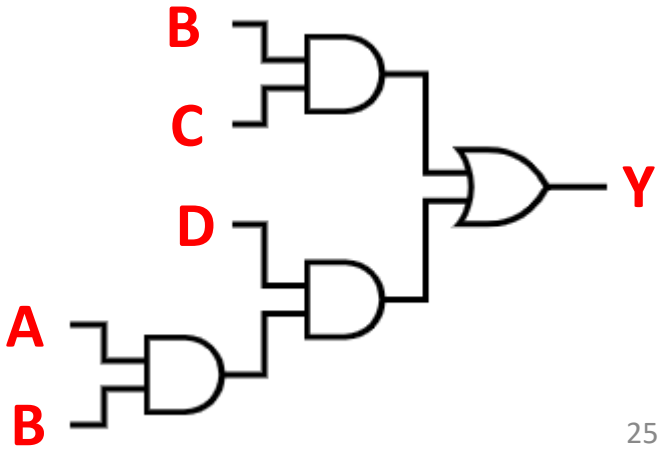
$$B\,C + D\,A\,B$$

Example 4

$$B\,C + (C\,\overline{D} + A\,B)\,D$$

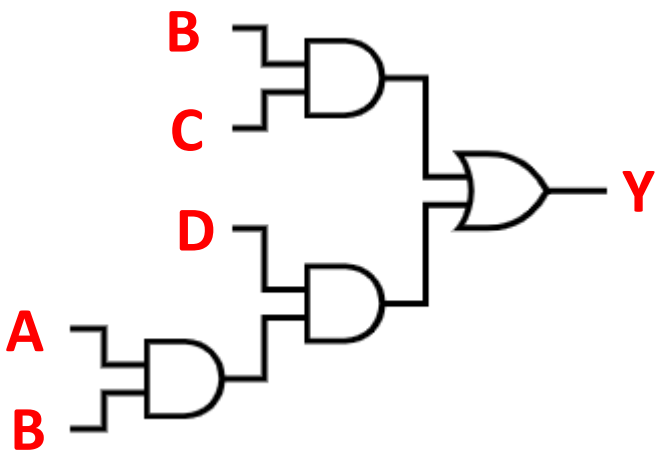


$$\Rightarrow \quad B\,C + D\,A\,B$$



Example 4

$$B\,C + D\,A\,B$$



TRUTH TABLE

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Example 5

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

De Morgan Theorem

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

Involution Law

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

Example 5

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

COMPLETE THE TRUTH TABLE

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

EASIER TO USE → $\overline{A} + B + C$

Example 5

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

COMPLETE THE TRUTH TABLE

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

$\overline{A} = 1$

$\overline{A} = 0$

ALWAYS AT LEAST ONE INPUT $= 1$

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

Example 5

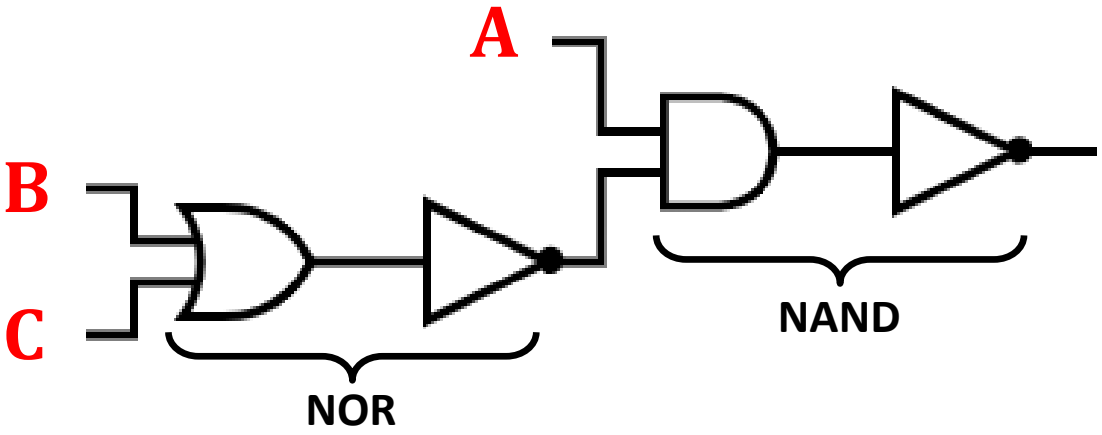
$$\overline{A(\overline{B+C})}$$

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

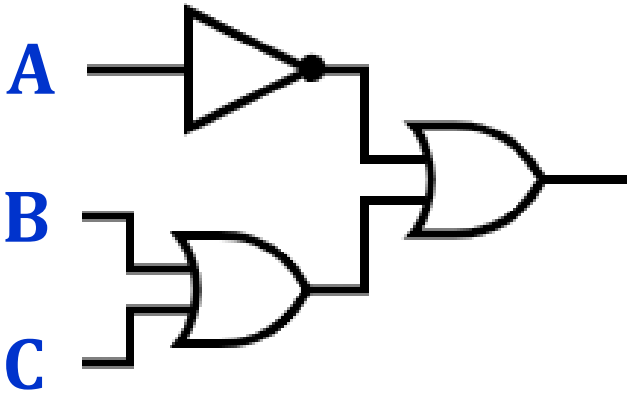
$$\overline{A} + B + C$$

Example 5

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



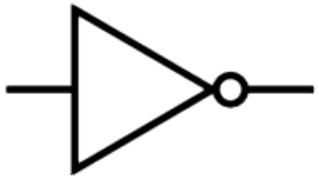
$$\overline{A}+B+C$$



Using standard gates to construct other logic functions

NOR and NAND are considered universal gates for the purpose of constructing other ones.

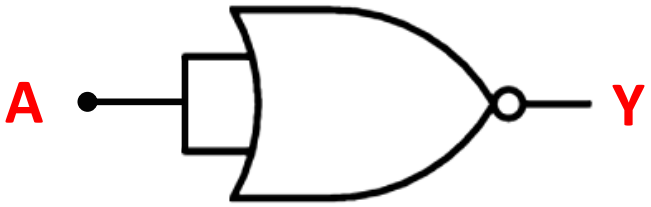
Construct the NOT gate



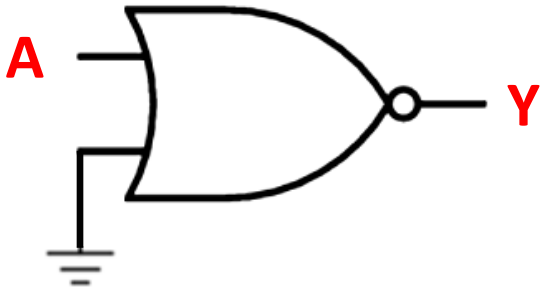
NOT

A	Y
0	1
1	0

Using NOR gates

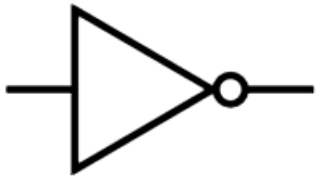


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

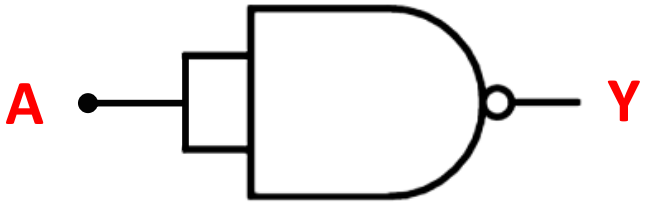
Construct the NOT gate



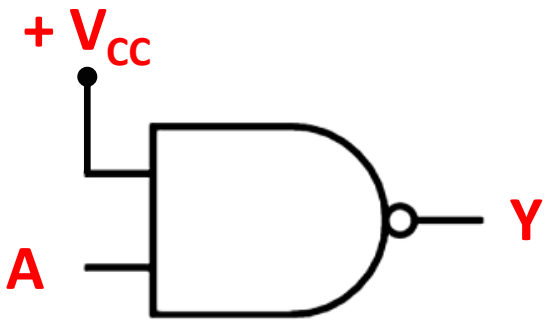
NOT

A	Y
0	1
1	0

Using NAND gates

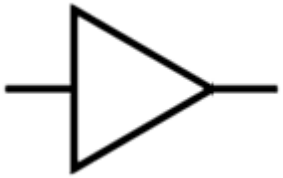


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

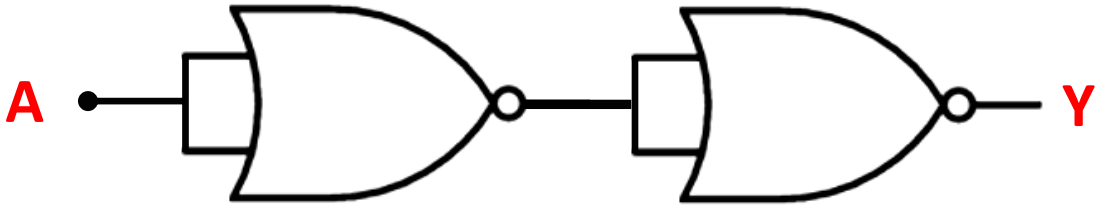
Construct the BUFFER gate

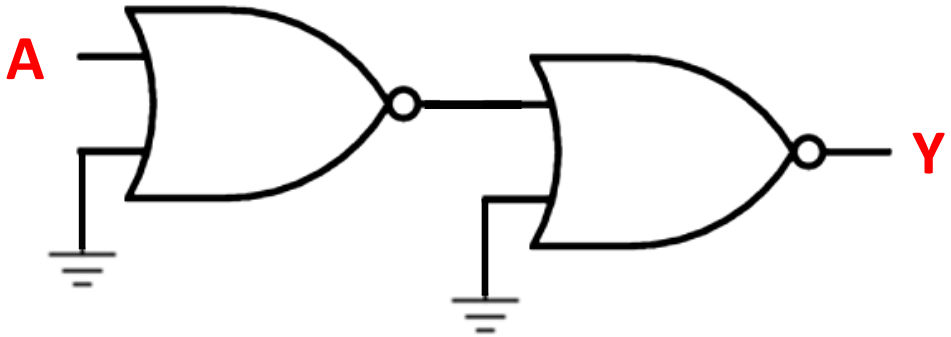


BUFFER

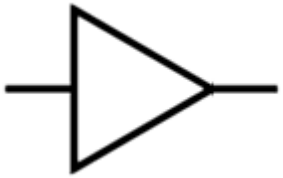
A	Y
0	0
1	1

Using NOR gates





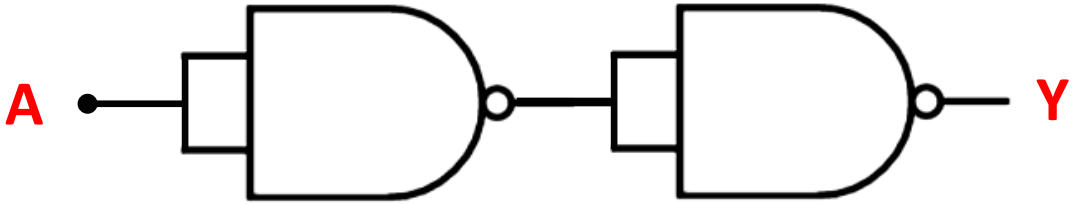
Construct the BUFFER gate

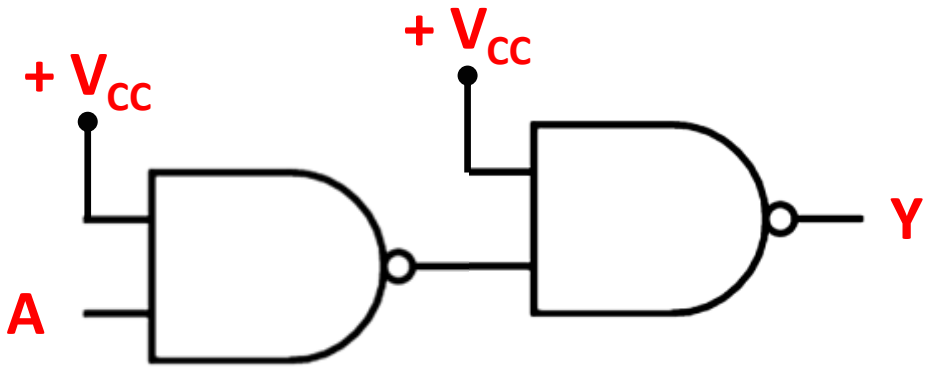


BUFFER

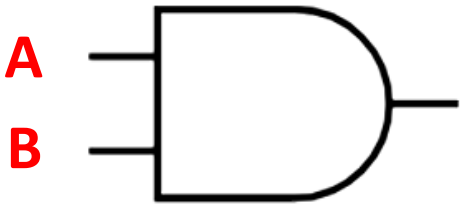
A	Y
0	0
1	1

Using NAND gates





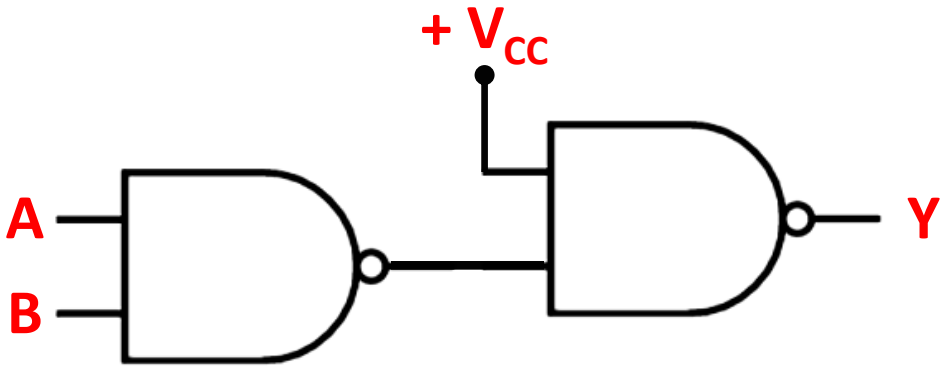
Construct the AND gate



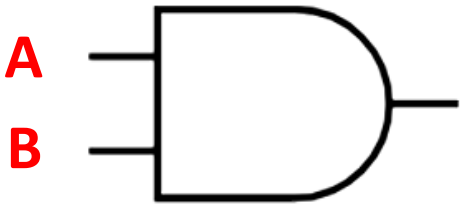
AND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Using NAND gates



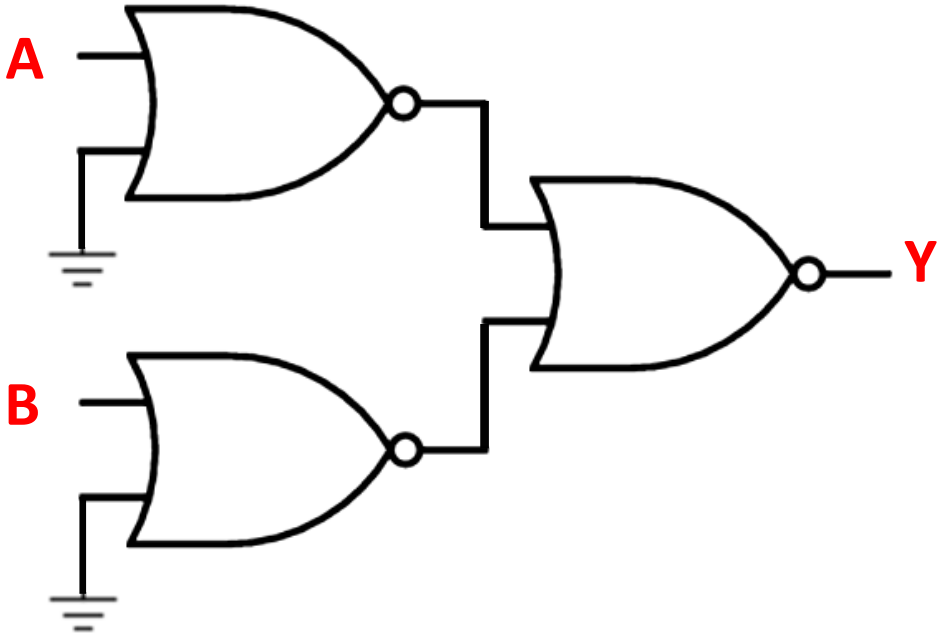
Construct the AND gate



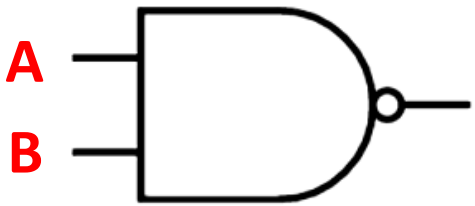
AND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Using NOR gates



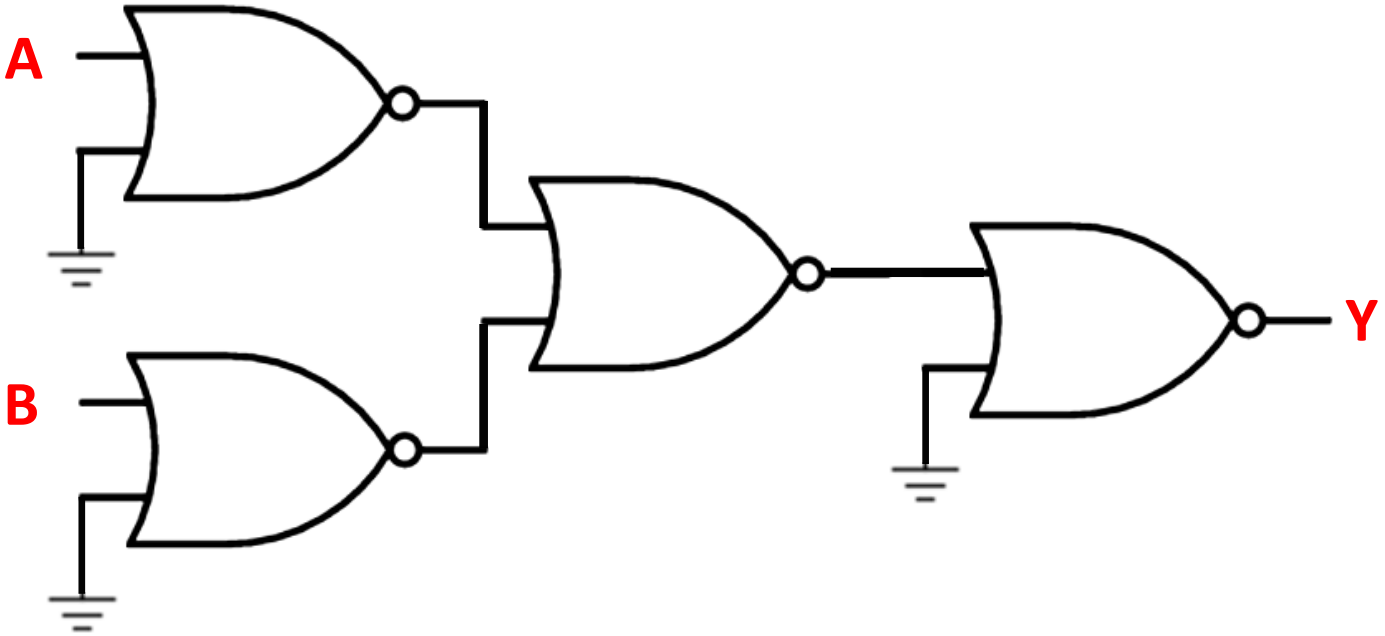
Construct the NAND gate



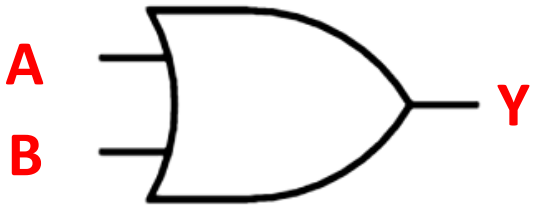
NAND

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Using NOR gates



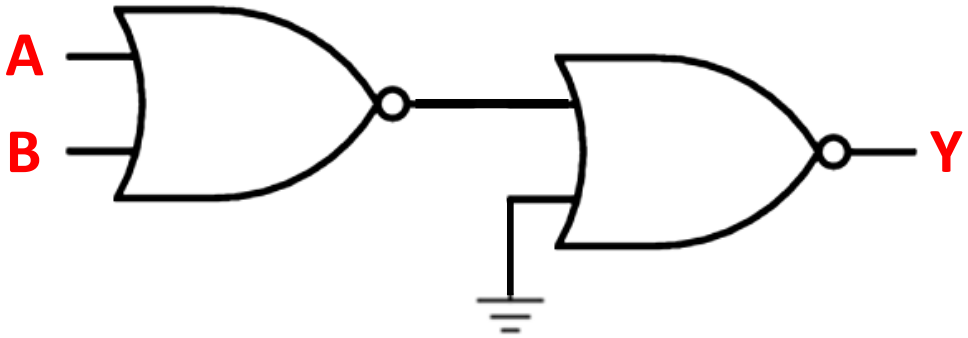
Construct the OR gate



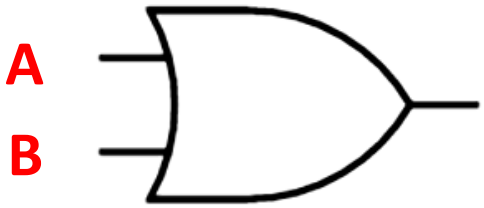
OR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Using NOR gates



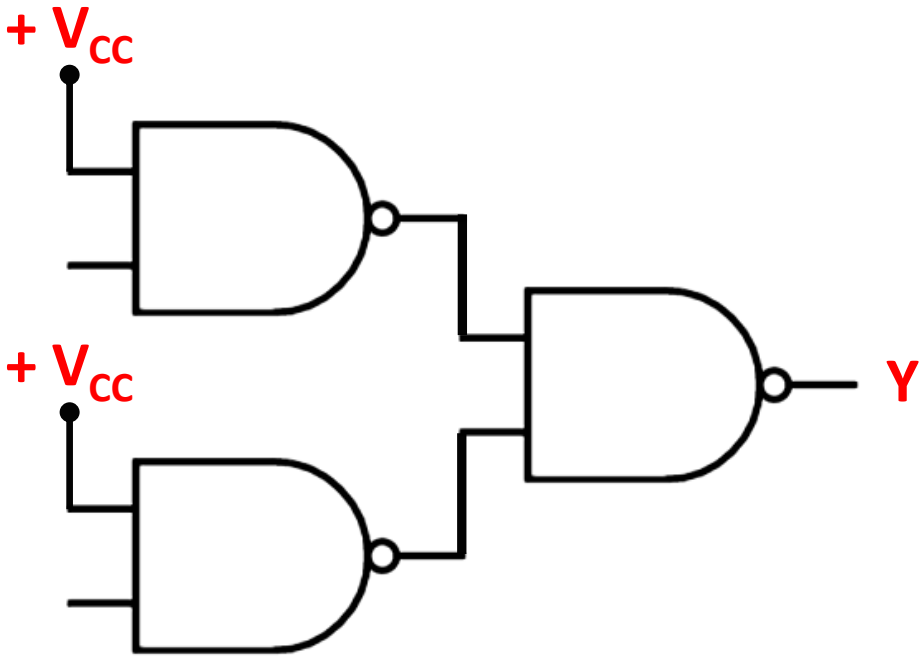
Construct the OR gate



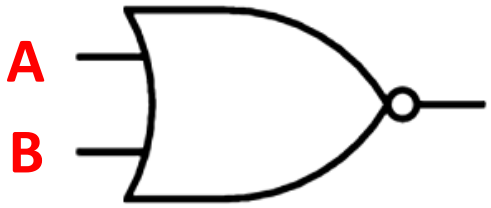
OR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Using NAND gates



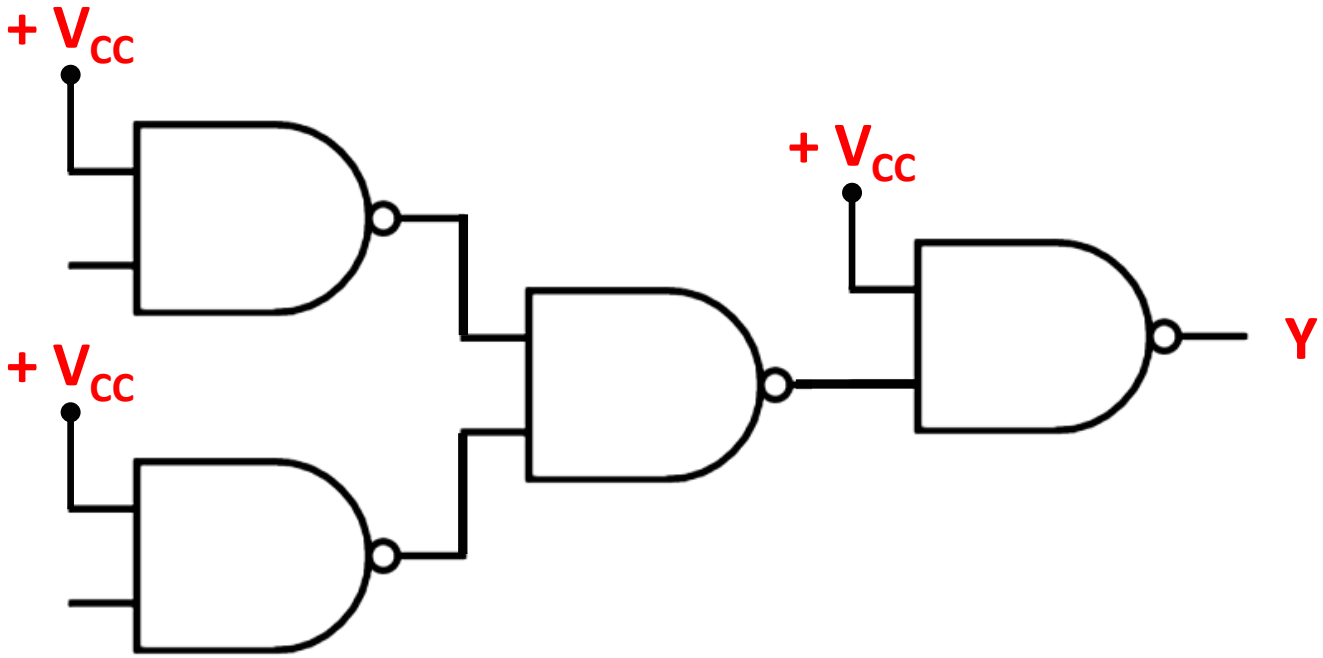
Construct the NOR gate



NOR

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Using NAND gates

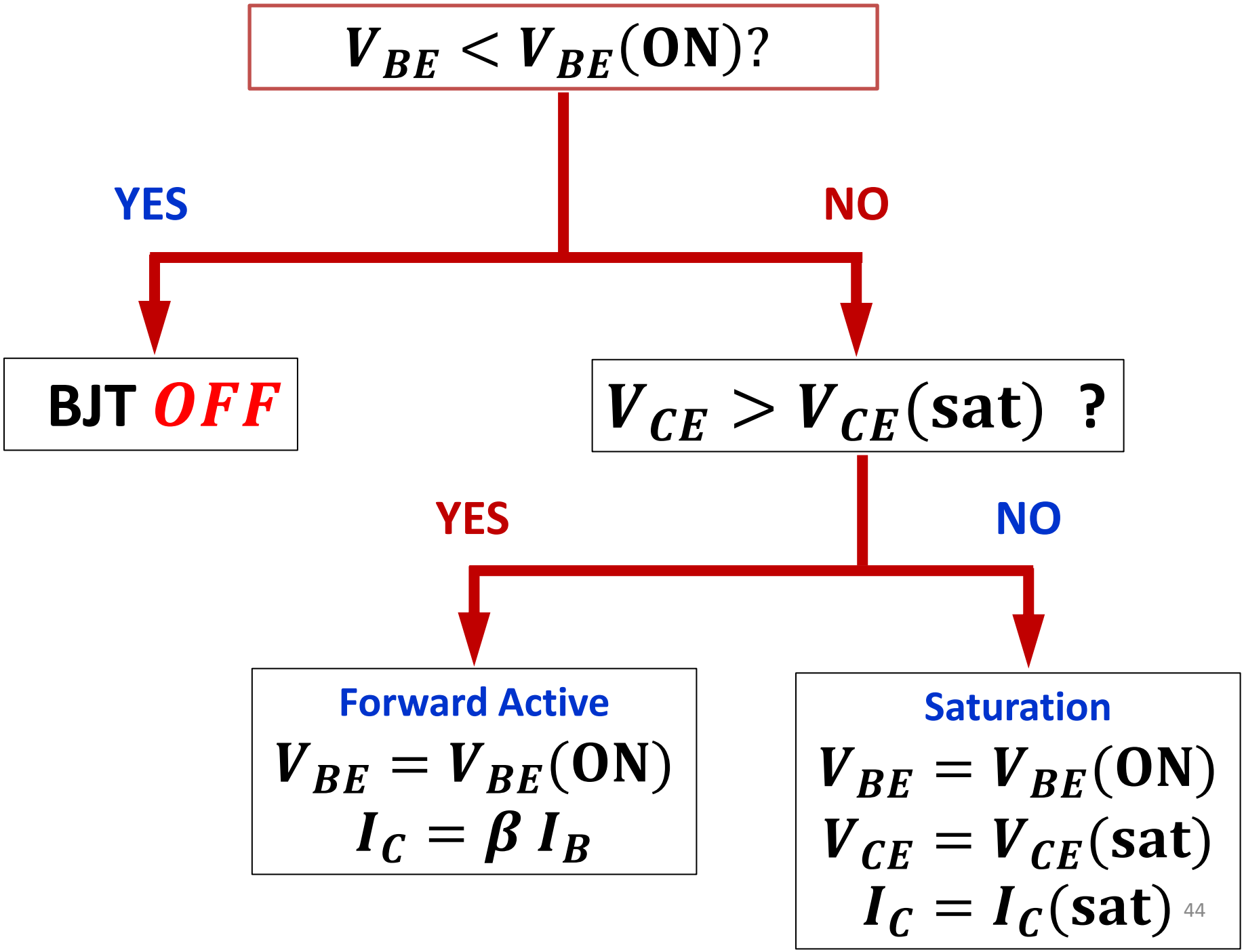


Logic Gates using BJT's

We consider logic gates made with BJTs connected by resistors (Resistor-Transistor-Logic or RTL). This was the earliest digital logic family for integrated circuits.

While there are much higher performance designs for BJT chips (e.g., Transistor-Transistor-Logic or TTL), RTL is still a good approach to prototype simple logic circuits with discrete components that can handle a fair amount of power in servo-mechanisms.

Intermediate between RTL and TTL, there was the Diode-Transistor-Logic (DTL) where inputs run through *p-n* junctions.

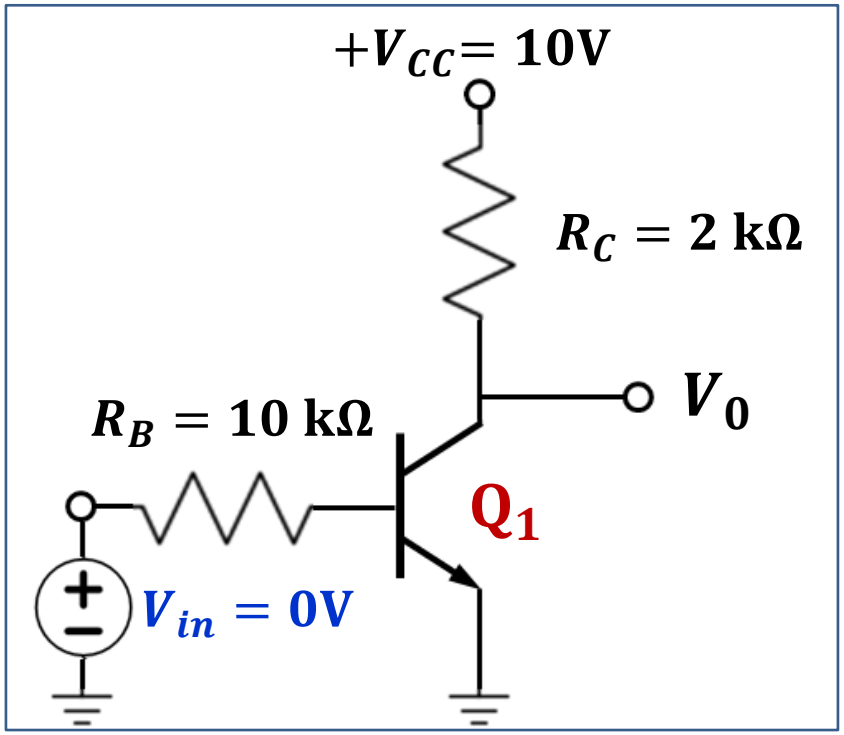

$V_{BE} < V_{BE}(\mathrm{ON})?$
YES
NO
BJT OFF
$V_{CE} > V_{CE}(\mathrm{sat})$?
YES
NO
Forward Active
$V_{BE} = V_{BE}(\mathrm{ON})$
$I_C = \beta I_B$
Saturation
$V_{BE} = V_{BE}(\mathrm{ON})$
$V_{CE} = V_{CE}(\mathrm{sat})$
$I_C = I_C(\mathrm{sat})$

Consider two cases

Assume $V_{BE}(\text{ON}) = 0.7\ \text{V}$

$V_{CE}(\text{sat}) = 0.2\ \text{V}$

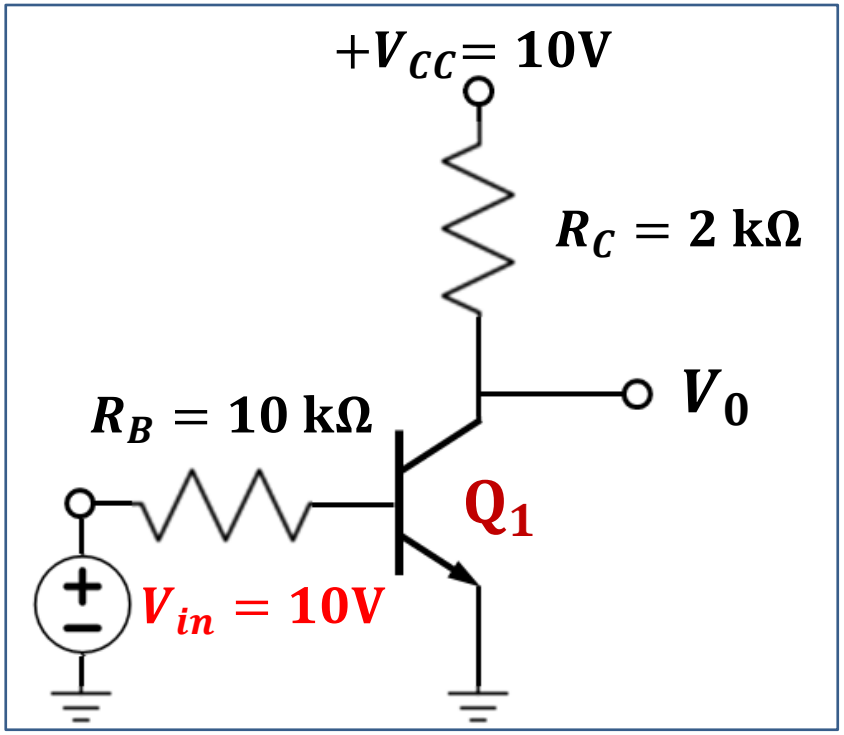
$\beta = 10$



$V_{BE} < V_{BE}(\text{ON})$

$V_0 = V_{CC} = 10\text{V}$

Q_1 OFF



$$I_B = \frac{10 - 0.7}{10\text{k}\Omega} = 0.93\text{mA}$$

$I_C = \beta I_B = 9.3\text{mA}$ Assuming FA mode

$$I_C(\text{sat}) = \frac{10 - 0.2}{2\text{k}\Omega} = 4.8\text{mA}$$

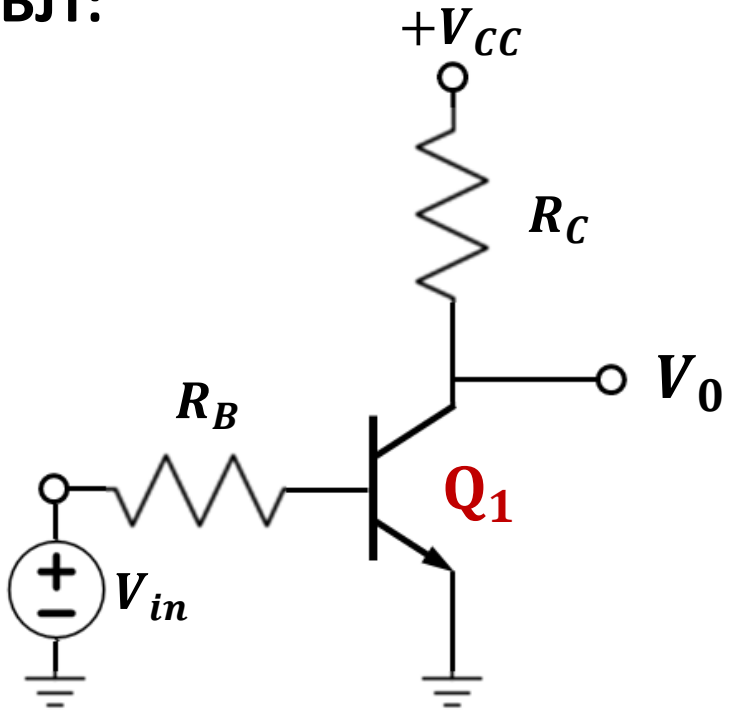
Q_1 SATURATION $V_0 = 0.2\ \text{V}$

Basic principle to design logic gates with BJT:

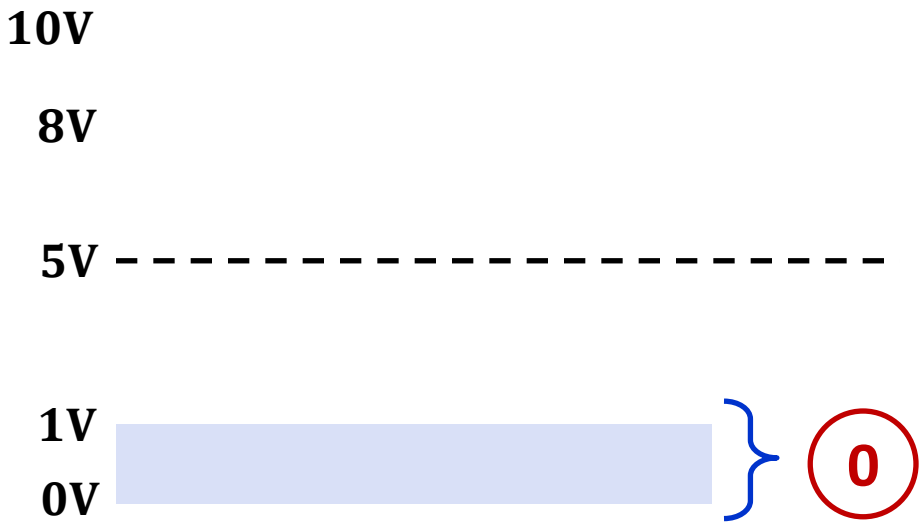
Two states of operation

(0) LOW $V_{in} \rightarrow$ HIGH V_0 (1)

Q_1 OFF



For a given technology one has to set reference voltage levels to accept logical states 0 and 1.



Basic principle to design logic gates with BJT:

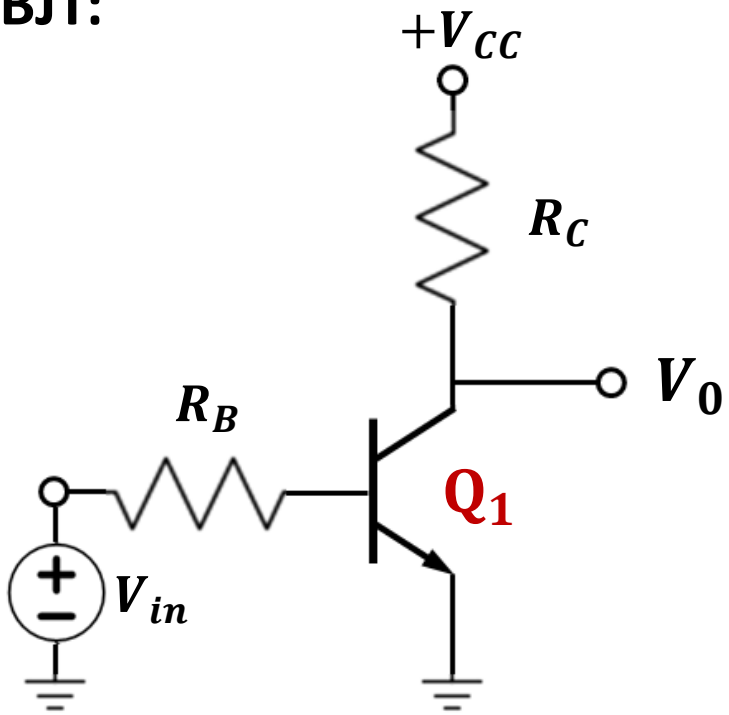
Two states of operation

(0) LOW $V_{in} \rightarrow$ HIGH V_0 (1)

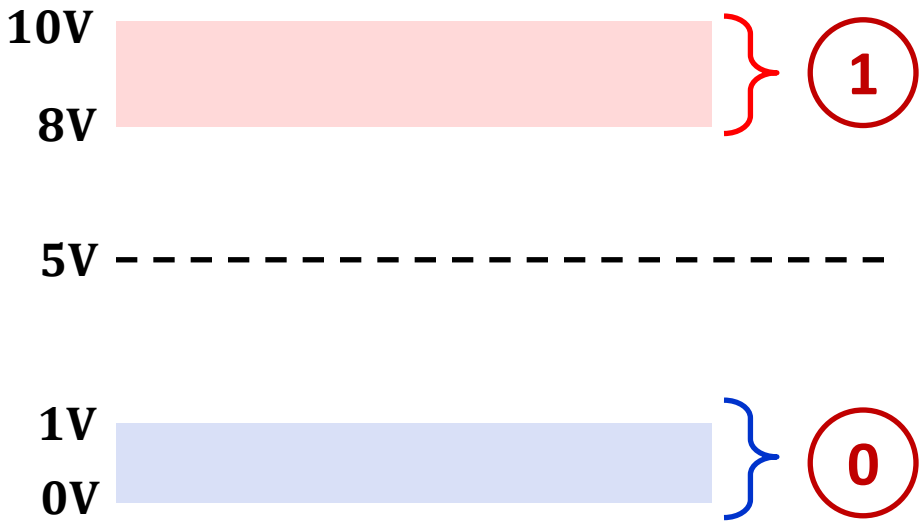
Q_1 OFF

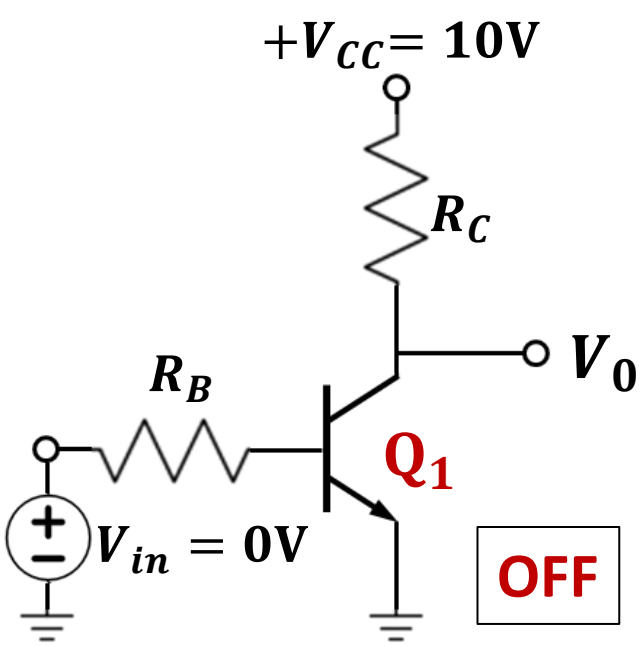
(1) HIGH $V_{in} \rightarrow$ LOW V_0 (0)

Q_1 SATURATION

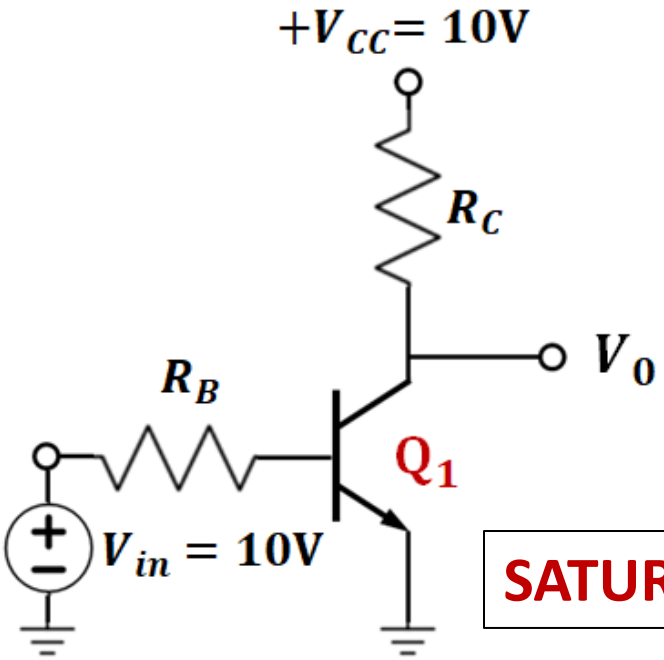


For a given technology one has to set reference voltage levels to accept logical states 0 and 1.



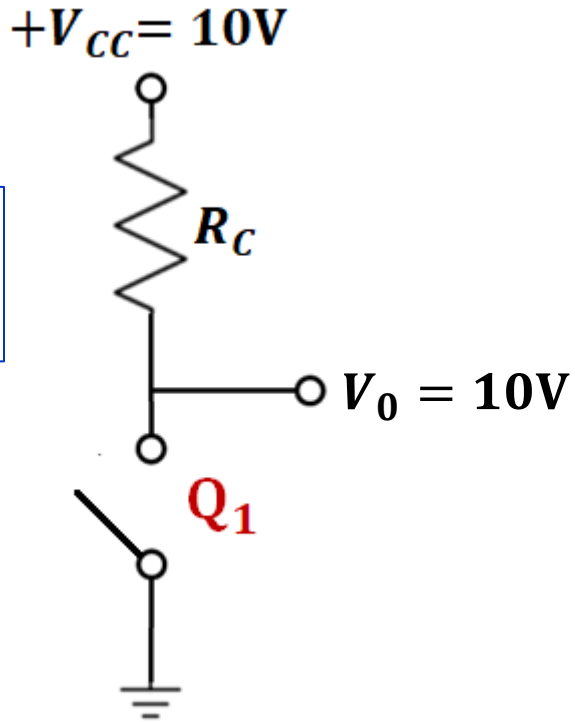


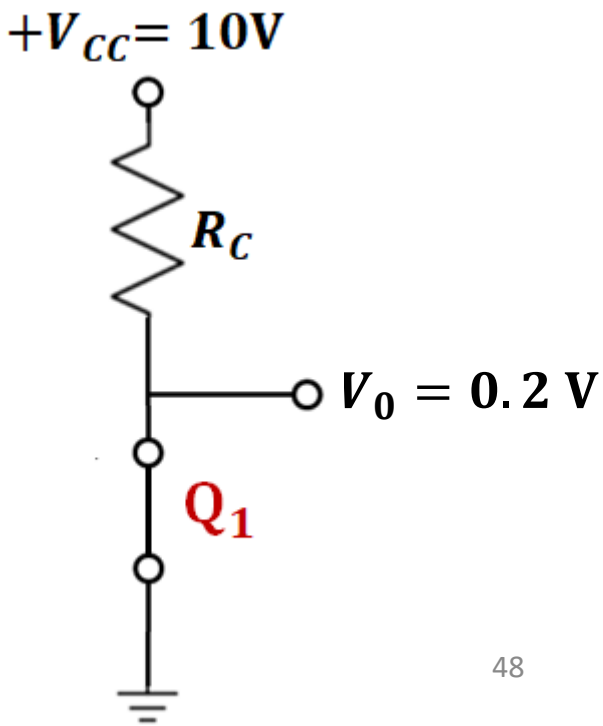
OFF



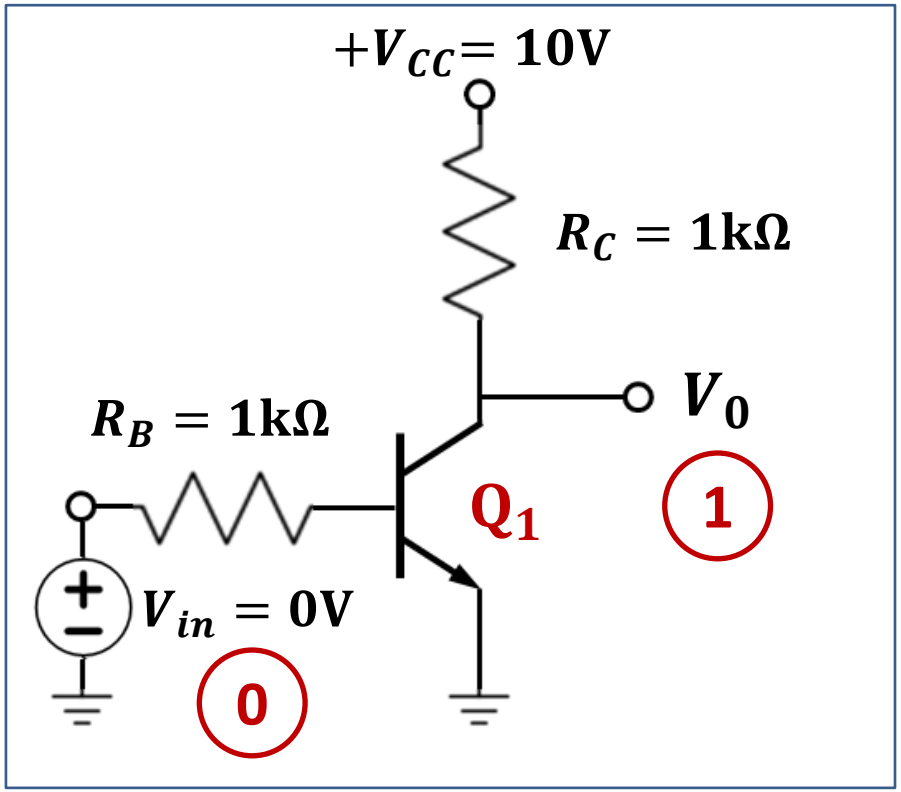
SATURATION

Transistors are like switches





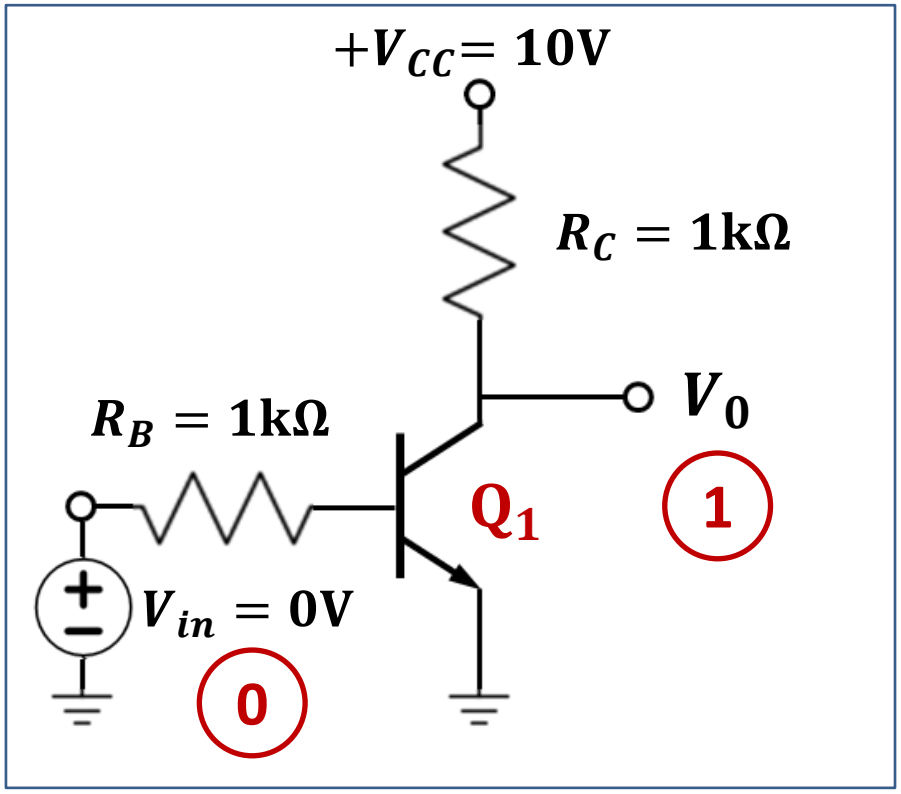
$V_{BE}(\text{ON}) = 0.7\ \text{V}$ $V_{CE}(\text{sat}) = 0.2\ \text{V}$ $\beta = 10$

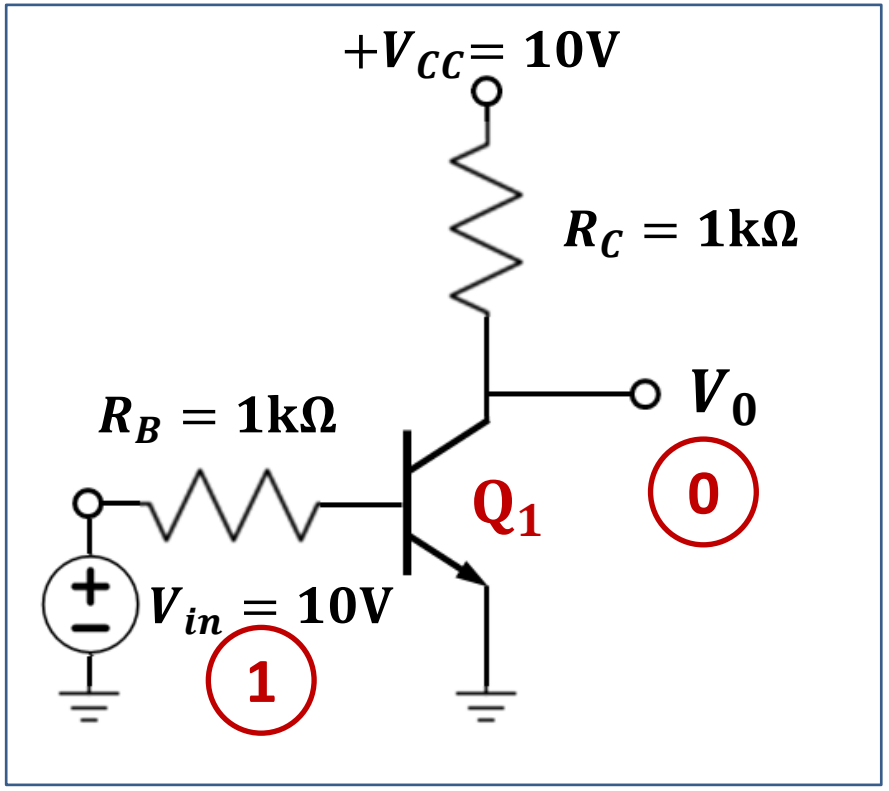


$V_0 = V_{CC} = 10\text{V}$ (1)

V_{in}	V_0
0 V (0)	10 V (1)

$V_{BE}(\text{ON}) = 0.7\ \text{V}$ $V_{CE}(\text{sat}) = 0.2\ \text{V}$ $\boxed{\beta = 10}$





$\boxed{V_0 = V_{CC} = 10\text{V}}$ (1)

V_{in}	V_0
0 V (0)	10 V (1)
10V (1)	0.2 V (0)

$$I_C(\text{sat}) = \frac{10 - 0.2}{1\text{k}\Omega} = 9.8\text{mA}$$

$$I_B = \frac{10 - 0.7}{1\text{k}\Omega} = 9.3\text{mA}$$

$$I_C = \beta I_B = 93\text{mA} \gg I_C(\text{sat})$$

$\boxed{V_0 = 0.2\ \text{V}}$ (0)

On the I-V curves