COMBINATIONAL LOGIC PART I: GATES & HALF ADDERS

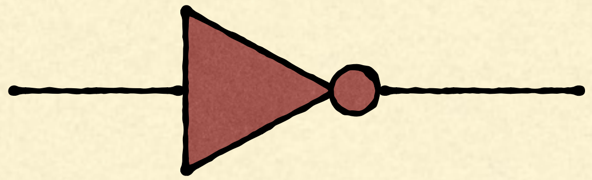
Ayman Hajja, PhD

HARDWARE DESIGN

- Next few weeks; how a modern processor is built, starting with basic elements as building blocks

COMBINATIONAL LOGIC SYMBOLS; 'NOT'

Boolean operator with only one variable that has the value one when the variable is zero and vice-versa

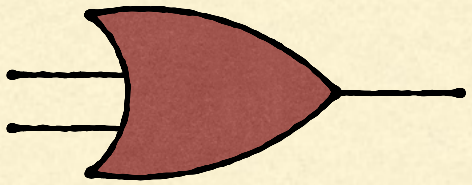


'NOT' Truth Table

A	~A
1	0
0	1

COMBINATIONAL LOGIC SYMBOLS; 'OR'

Boolean operator that gives the value one if at least one operand (or input) has a value of one, and otherwise has a value of zero.

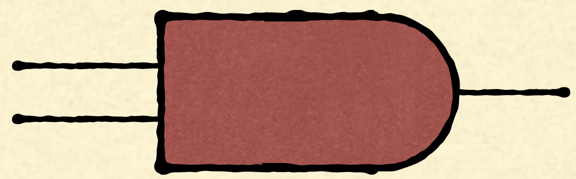


OR Truth Table

A	B	A + B
1	1	1
1	0	1
0	1	1
0	0	0

COMBINATIONAL LOGIC SYMBOLS; 'AND'

Boolean operator that gives the value one if and only if all the operands are one, and otherwise has a value of zero.

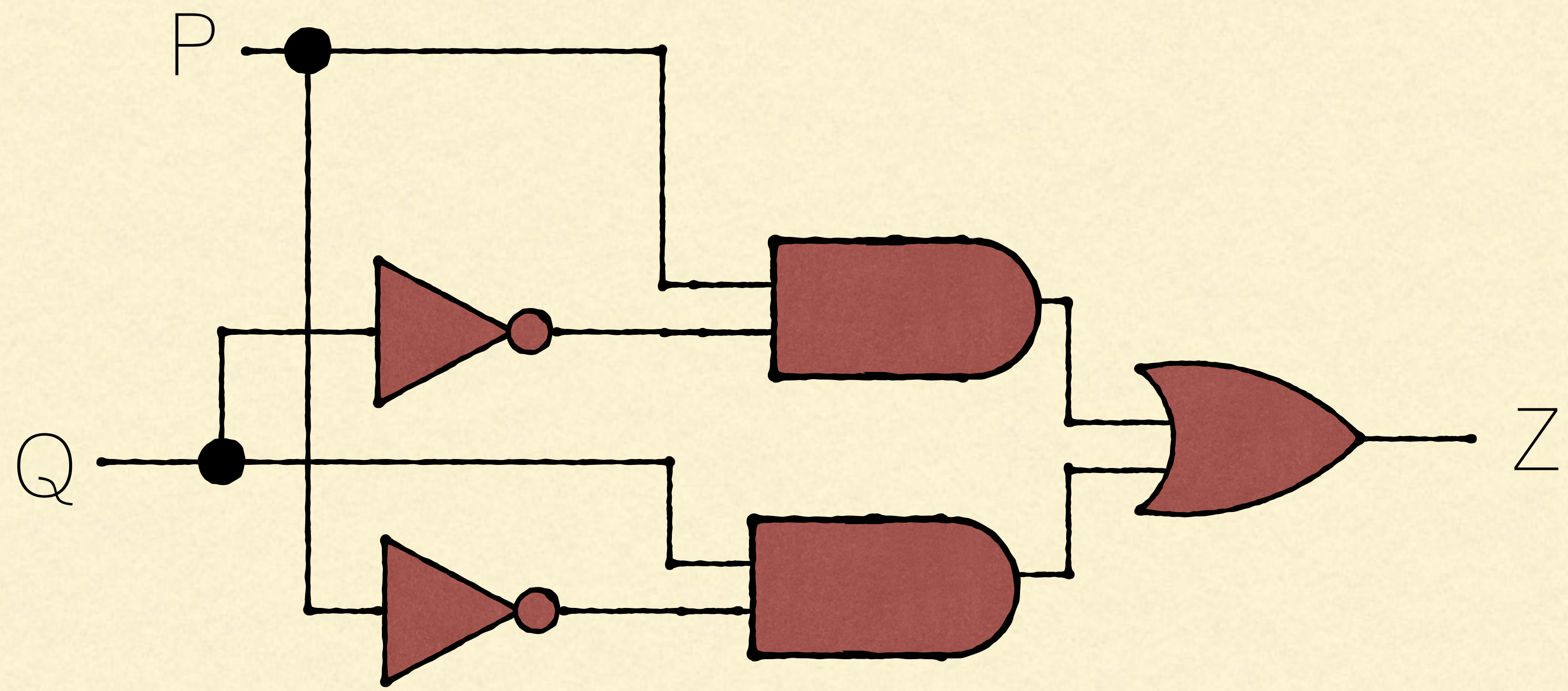


AND Truth Table

A	B	A . B
1	1	1
1	0	0
0	1	0
0	0	0

QUESTION

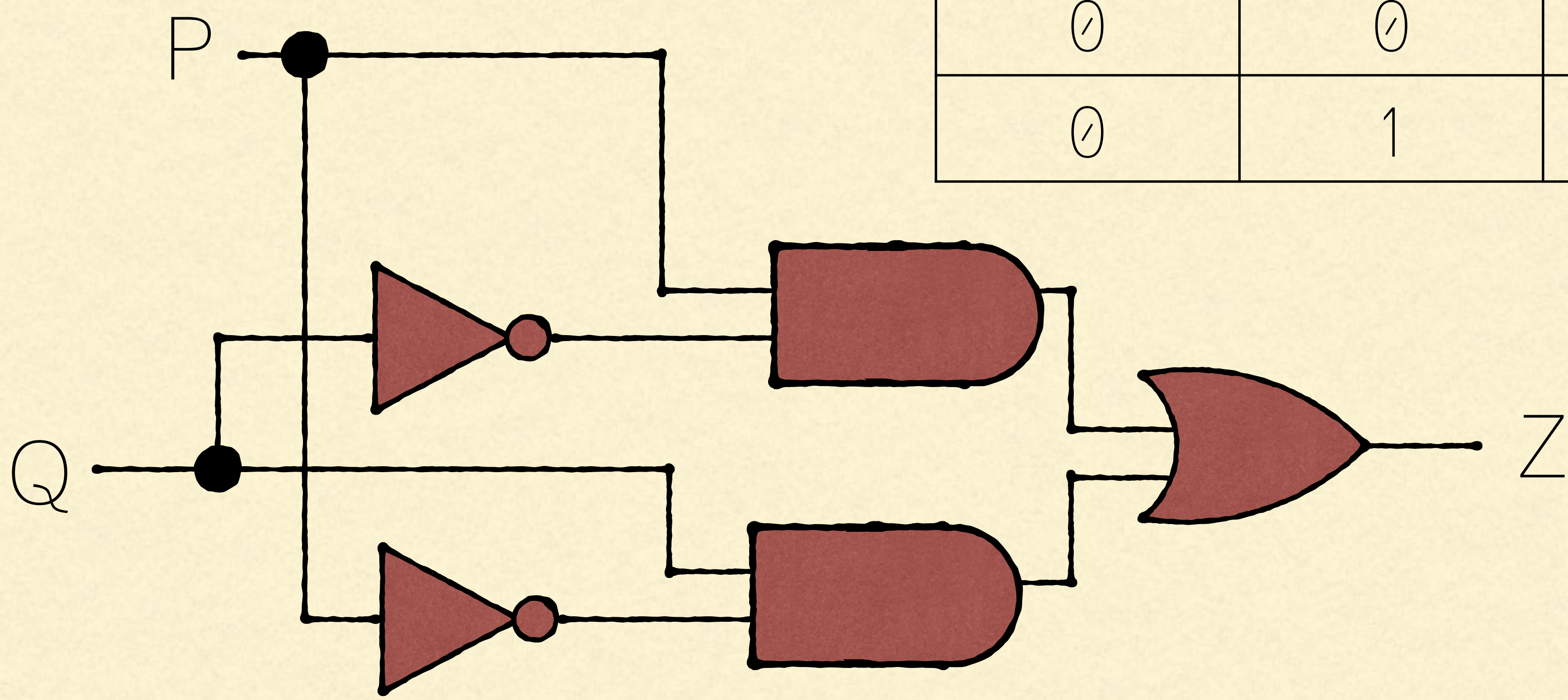
- What's the value of Z given that P is True (1) and Q is False (0)



QUESTION

- What about the other values?

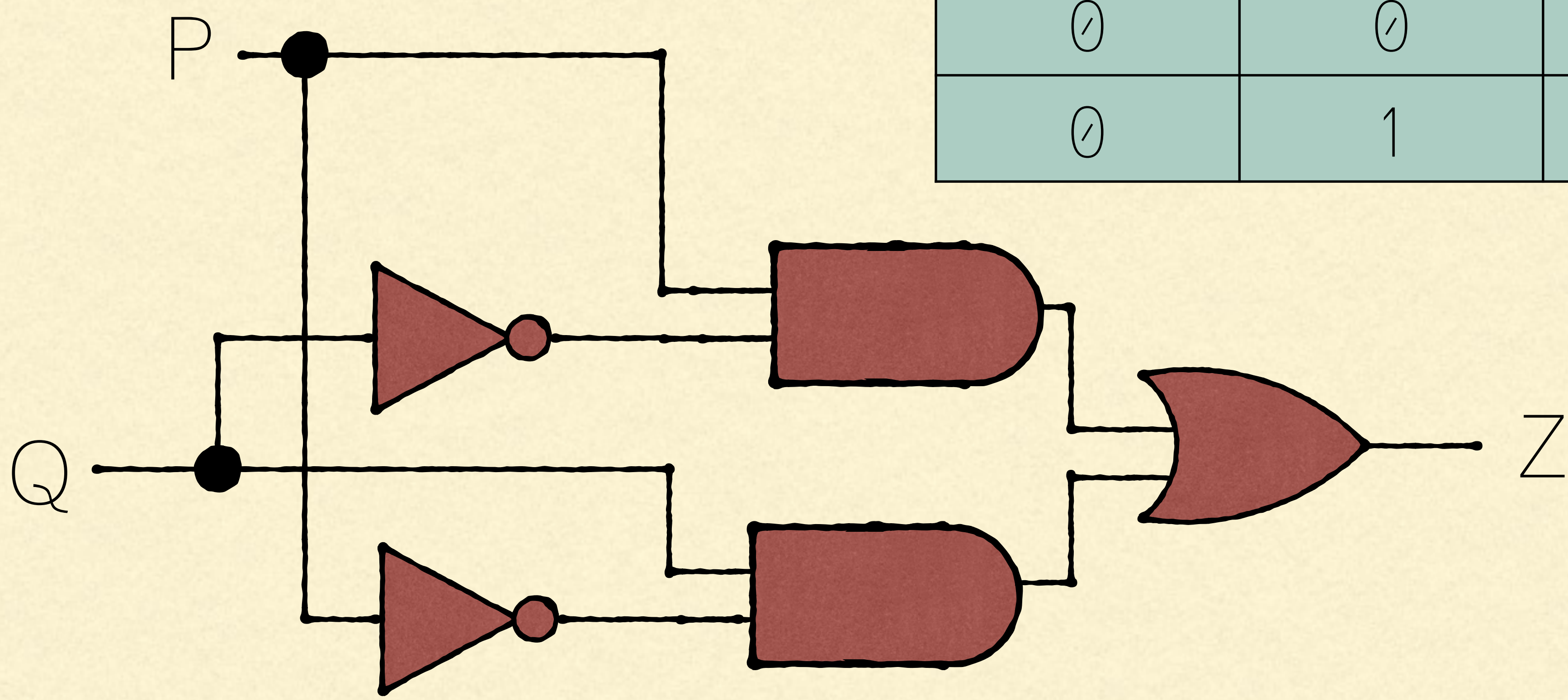
P	Q	Z
1	0	1
1	1	?
0	0	?
0	1	?



QUESTION

- What is this gate?

P	Q	Z
1	0	1
1	1	0
0	0	0
0	1	1

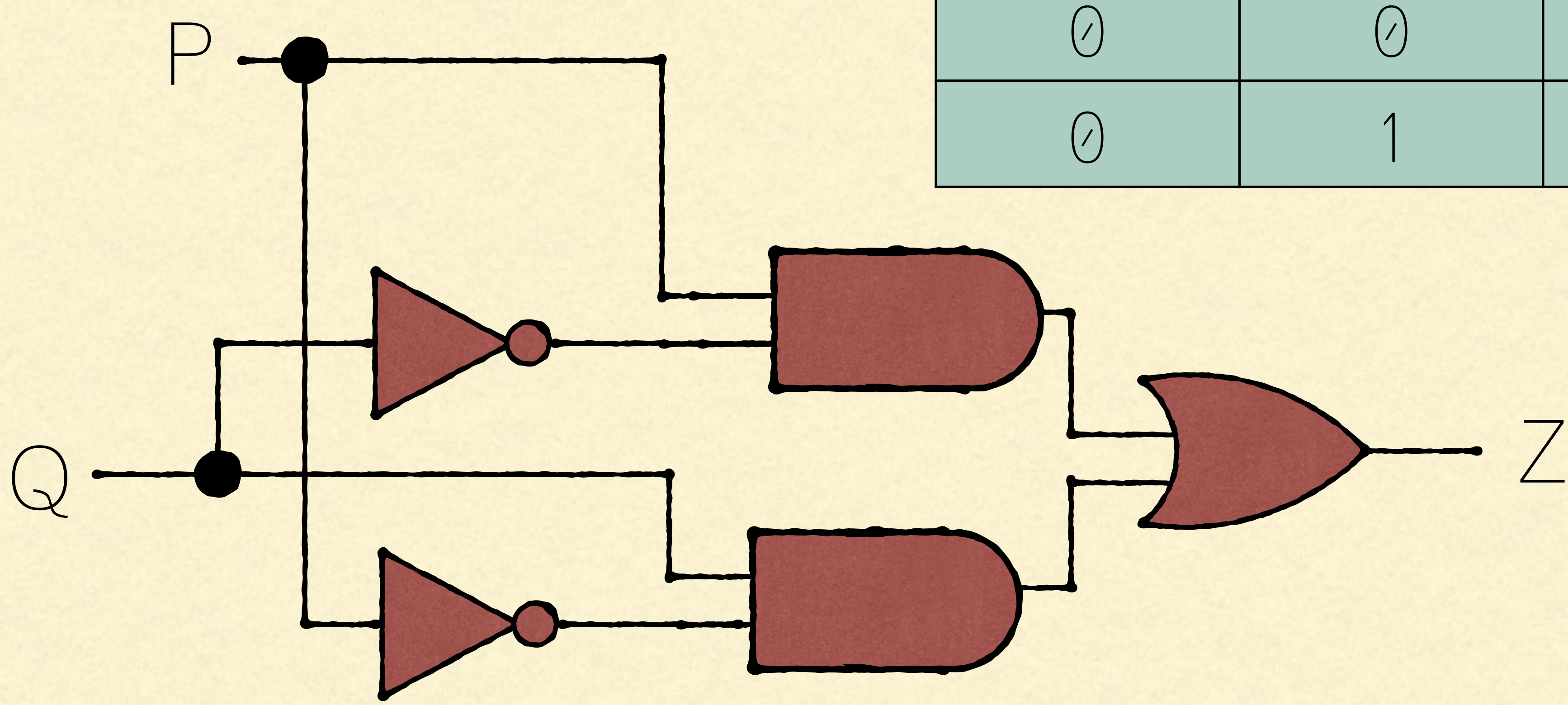


QUESTION

- What is this gate?

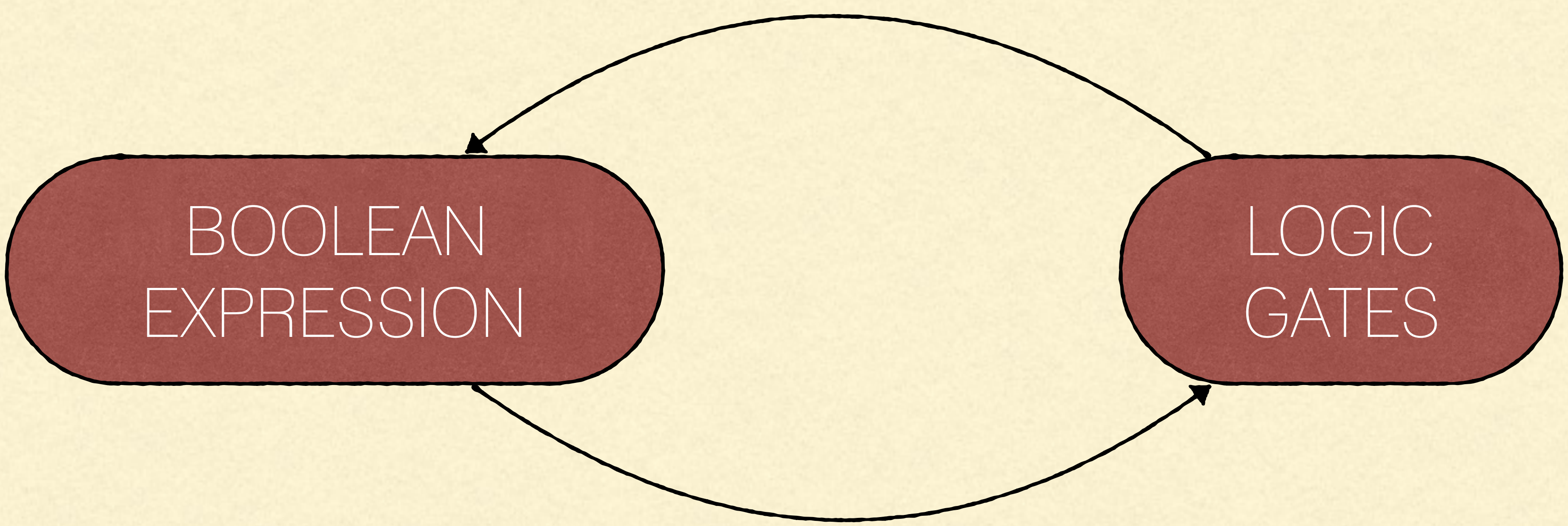
Exclusive Or

P	Q	Z
1	0	1
1	1	0
0	0	0
0	1	1



CONVERTING LOGIC GATES TO BOOLEAN EXPRESSIONS

- We can convert from any boolean expression to logic gates and vice-versa



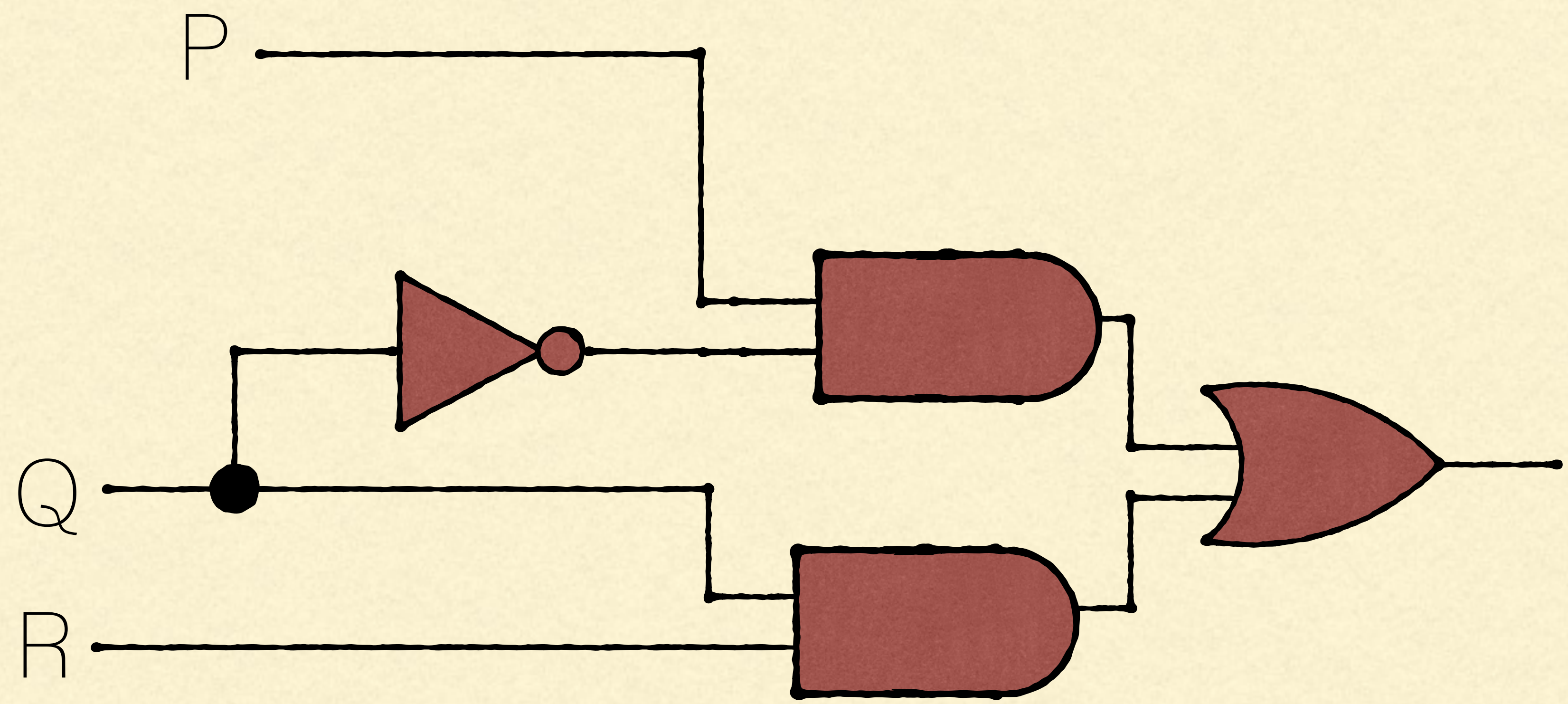
CONVERTING LOGIC GATES TO BOOLEAN EXPRESSIONS

- Convert the following boolean expression to logic gates:

$$\boxed{(P\ .\ \sim Q) + (Q\ .\ R)}$$

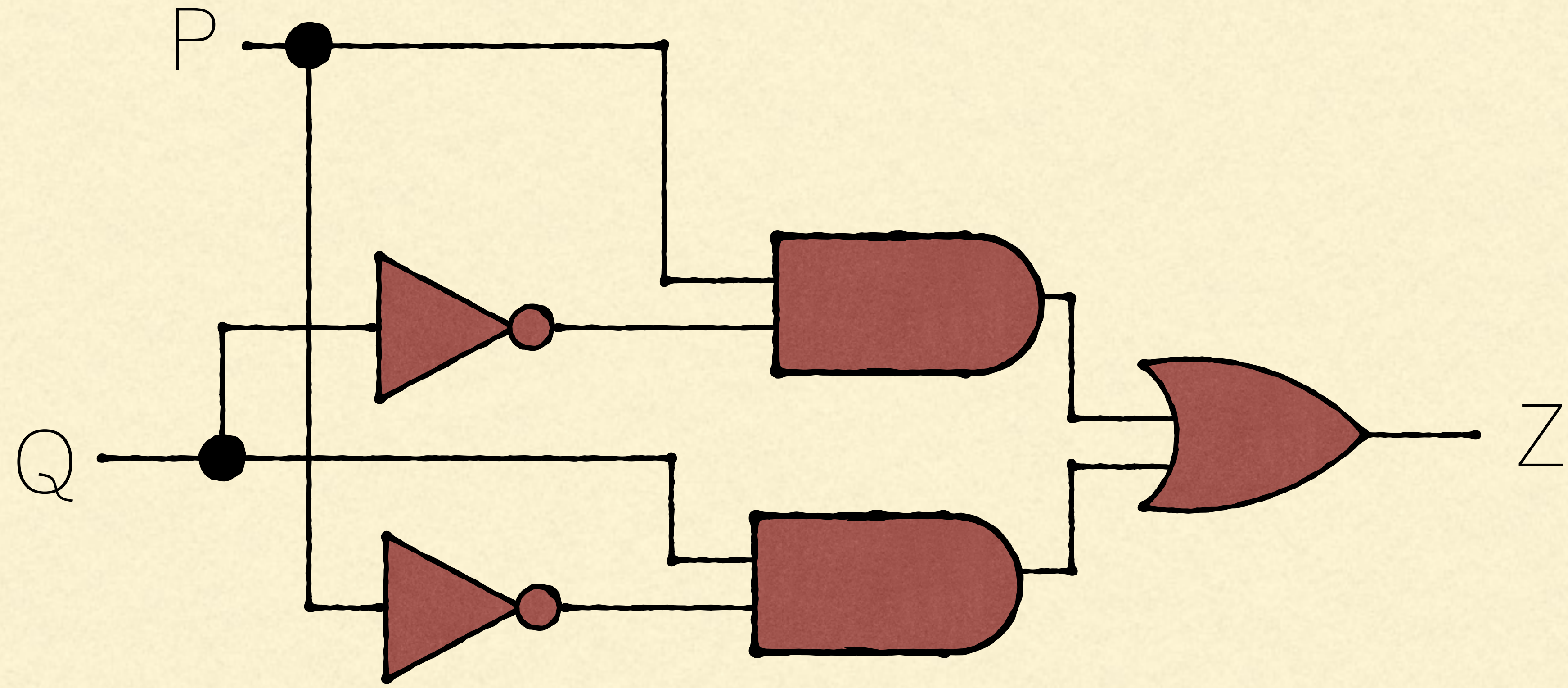
CONVERTING LOGIC GATES TO BOOLEAN EXPRESSIONS

(P . ~Q) + (Q . R)



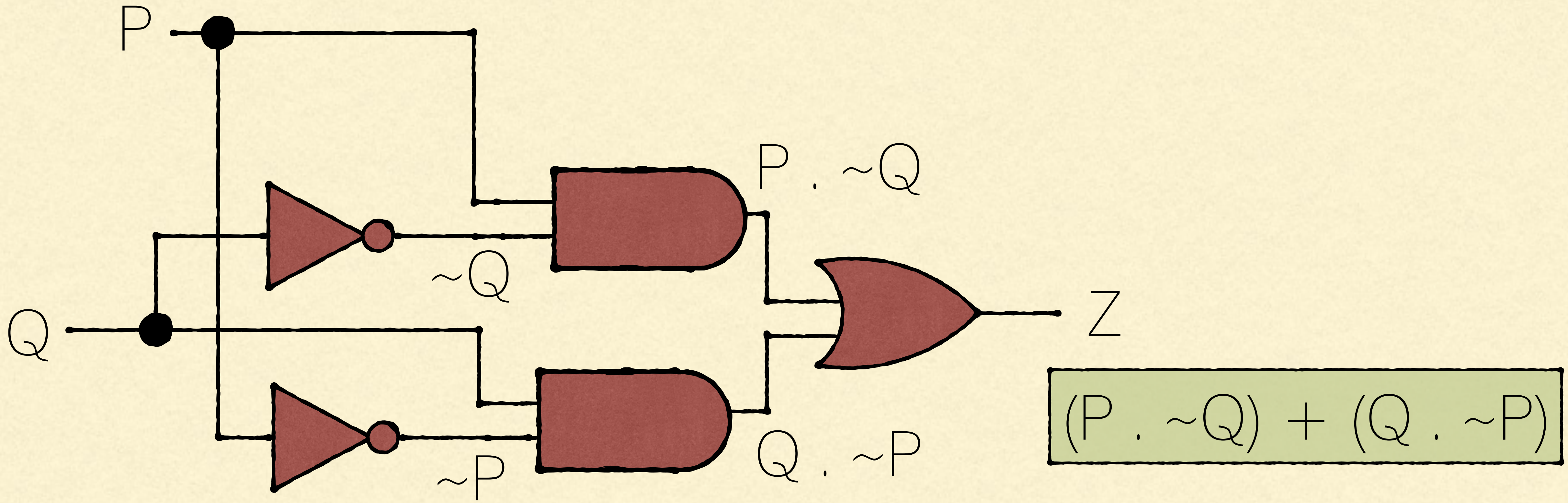
CONVERTING LOGIC GATES TO BOOLEAN EXPRESSIONS

- Convert the following gates to a boolean expression



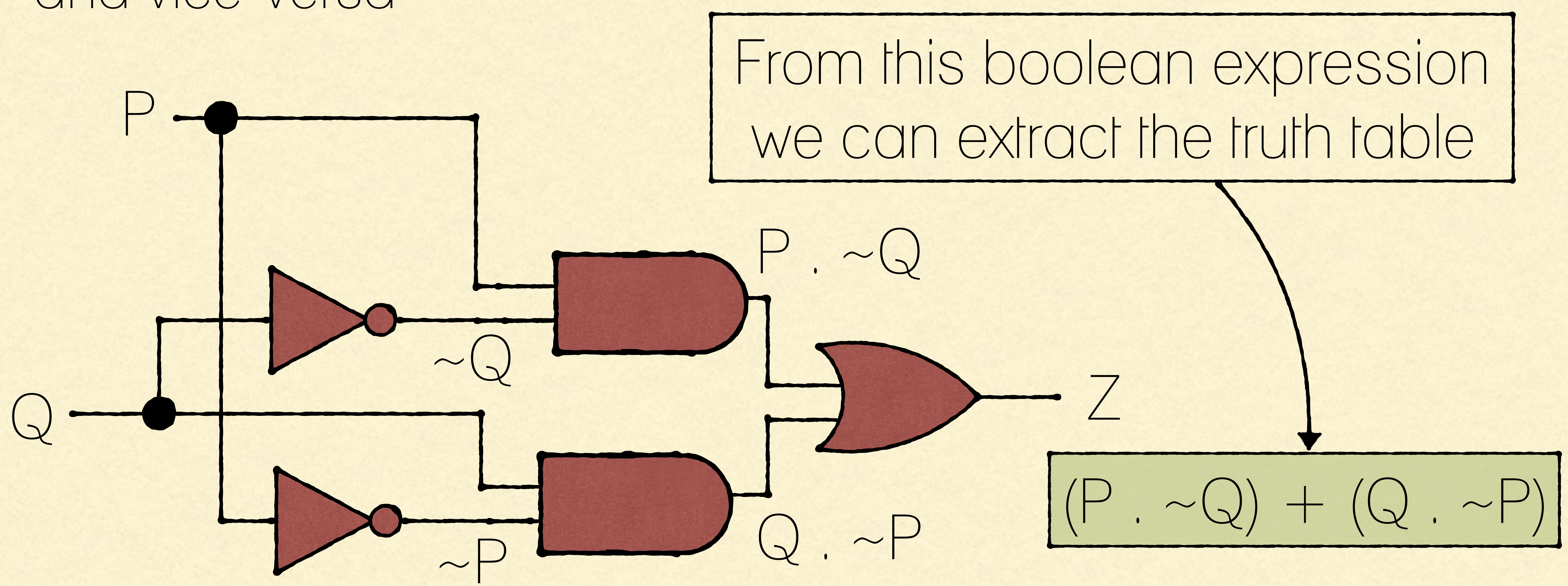
CONVERTING LOGIC GATES TO BOOLEAN EXPRESSIONS

- We can convert from any boolean expression to logic gates and vice-versa



CONVERTING LOGIC GATES TO BOOLEAN EXPRESSIONS

- We can convert from any boolean expression to logic gates and vice-versa



CONVERTING FROM BOOLEAN EXPRESSION TO TRUTH TABLE

$\boxed{(P \,.\, {\sim}Q) + (Q \,.\, {\sim}P) + R}$

CONVERTING FROM BOOLEAN EXPRESSION TO TRUTH TABLE

$(P \; . \; {\sim}Q) + (Q \; . \; {\sim}P) + R$

P	Q	R	Expression
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

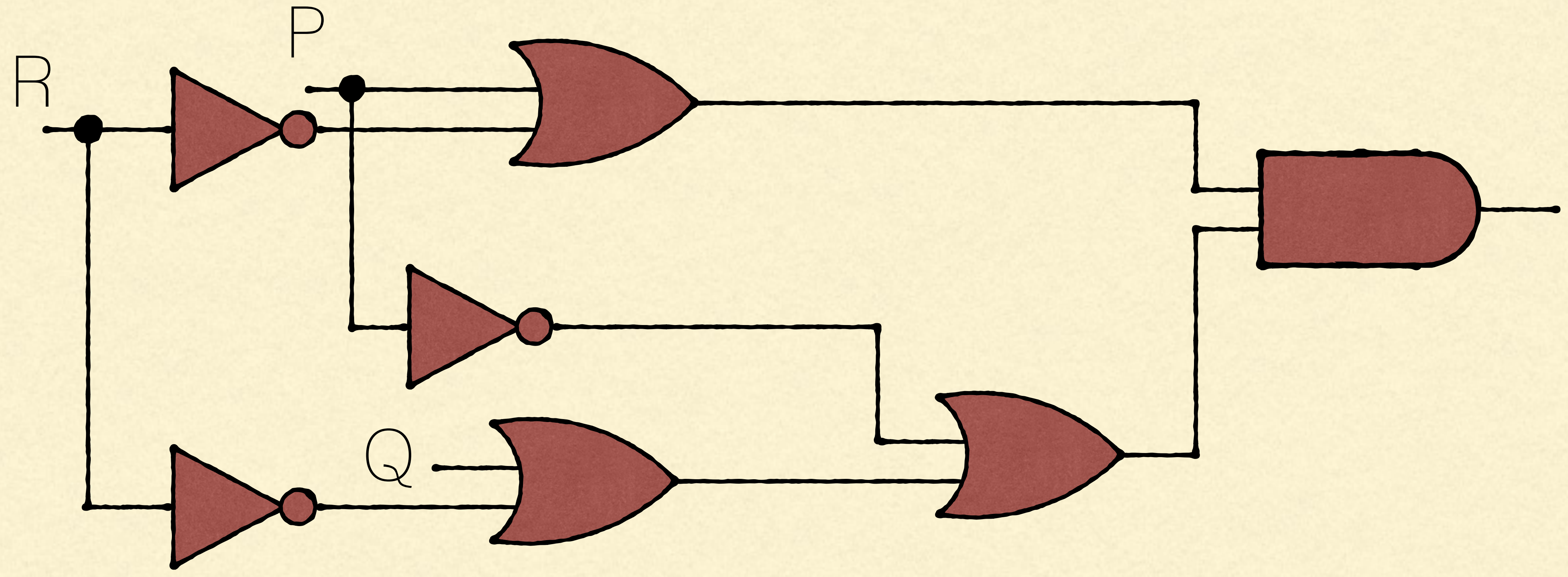
CONVERTING FROM BOOLEAN EXPRESSION TO TRUTH TABLE

$$\boxed{(P \,.\, {\sim}Q) + (Q \,.\, {\sim}P) + R}$$

P	Q	R	Expression
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

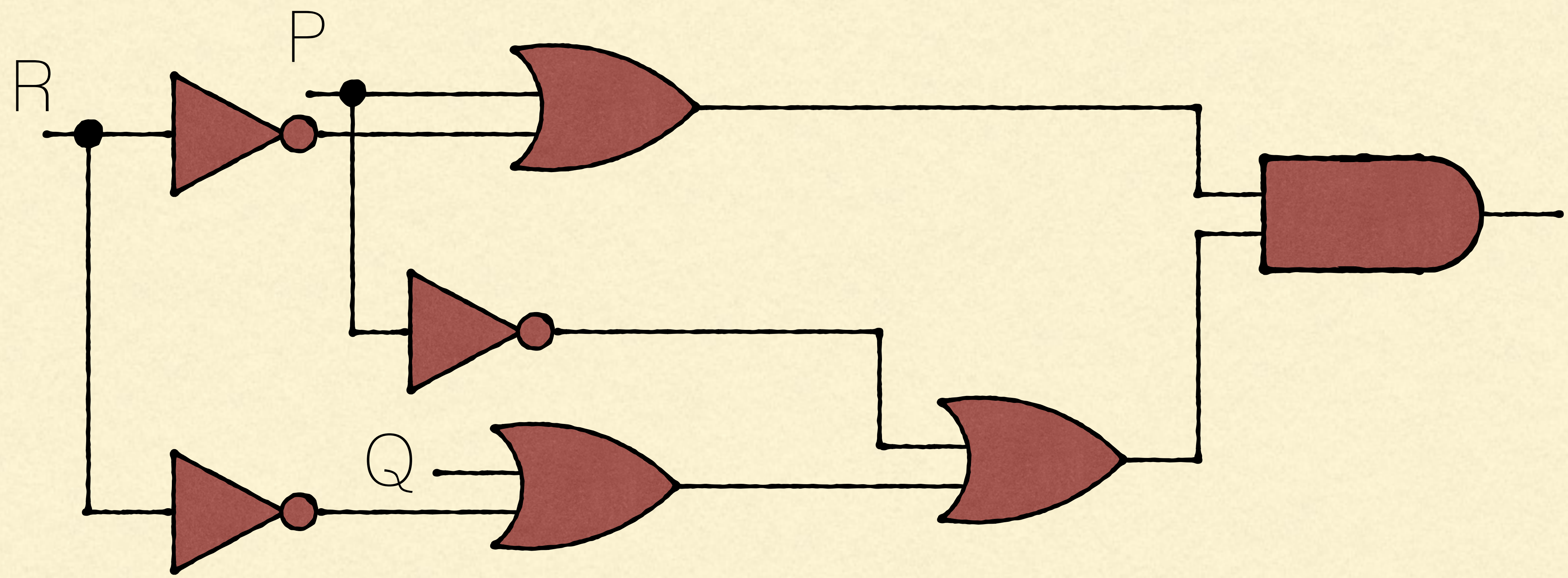
QUESTION

- Convert the following gates into a boolean expression



QUESTION

- Convert the following gates into a boolean expression



(P + ~R) . ((Q + ~R) + ~P)

CONVERTING FROM TRUTH TABLES TO BOOLEAN EXPRESSIONS

- Given a truth table, can we convert the truth table to a boolean expression?

P	Q	R	Expression
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

CONVERTING FROM TRUTH TABLES TO BOOLEAN EXPRESSIONS

- Given a truth table, can we convert the truth table to a boolean expression? Yes we can, using sum-of-products

P	Q	R	Expression
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

CONVERTING FROM TRUTH TABLES TO BOOLEAN EXPRESSIONS

- Given a truth table, can we convert the truth table to a boolean expression? Yes we can, using sum-of-products

P	Q	R	Expression	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	~P . Q . R
1	0	0	0	
1	0	1	1	
1	1	0	1	
1	1	1	1	

CONVERTING FROM TRUTH TABLES TO BOOLEAN EXPRESSIONS

- Given a truth table, can we convert the truth table to a boolean expression? Yes we can, using sum-of-products

P	Q	R	Expression	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	~P . Q . R
1	0	0	0	
1	0	1	1	P . ~Q . R
1	1	0	1	
1	1	1	1	

CONVERTING FROM TRUTH TABLES TO BOOLEAN EXPRESSIONS

- Given a truth table, can we convert the truth table to a boolean expression? Yes we can, using sum-of-products

P	Q	R	Expression	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	~P . Q . R
1	0	0	0	
1	0	1	1	P . ~Q . R
1	1	0	1	P . Q . ~R
1	1	1	1	

CONVERTING FROM TRUTH TABLES TO BOOLEAN EXPRESSIONS

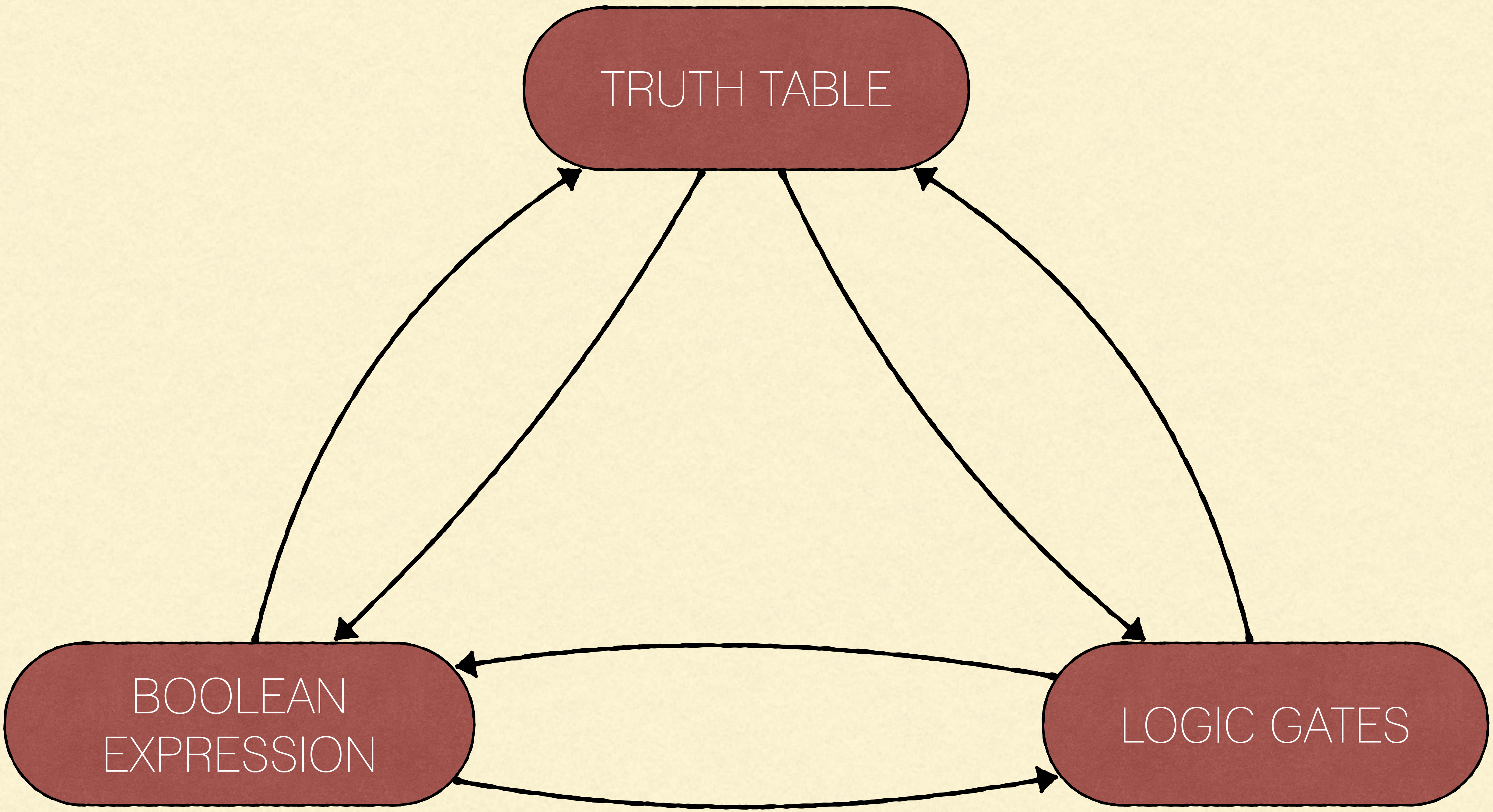
- Given a truth table, can we convert the truth table to a boolean expression? Yes we can, using sum-of-products

P	Q	R	Expression	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	~P . Q . R
1	0	0	0	
1	0	1	1	P . ~Q . R
1	1	0	1	P . Q . ~R
1	1	1	1	P . Q . R

CONVERTING FROM TRUTH TABLES TO BOOLEAN EXPRESSIONS

P	Q	R	Expression
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

~P . Q . R + P . ~Q . R +P . Q . ~R + P . Q . R


TRUTH TABLE
BOOLEAN EXPRESSION
LOGIC GATES

LOGICAL EQUIVALENCE

- The two boolean expressions p and q are called logically equivalent if p is true if and only if q is true; and if p is false if and only if q is false.
- In other words, p and q are called logically equivalent when their truth value always match.
- The symbol $\equiv$ refers to logical equivalence.

LOGICAL EQUIVALENCE

- One way to show whether two propositions are logically equivalent or not, is to construct the truth table for both and check if the truth values match!

EXAMPLE OF LOGICAL EQUIVALENCE

- Show that $\sim(p \vee q)$ and $\sim p \wedge \sim q$ are logically equivalent!
- We construct the truth table for both and check if they match:

P	Q	$P \vee Q$	$\sim(P \vee Q)$	$\sim P$	$\sim Q$	$\sim P \wedge \sim Q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

- Since the truth table for $\sim(p \vee q)$ and $\sim p \wedge \sim q$ are the same then they are logically equivalent

LAWS OF BOOLEAN ALGEBRA

- $X . \sim X = 0$
- $X . 0 = 0$
- $X . 1 = X$
- $X . X = X$
- $X . Y = Y . X$
- $X (Y + Z) = X Y + X Z$
- $\sim(X . Y) = \sim X + \sim Y$
- $X + \sim X = 1$
- $X + 0 = X$
- $X + 1 = 1$
- $X + X = X$
- $X + Y = Y + X$
- $X + Y Z = (X + Y) . (X + Z)$
- $\sim(X + Y) = \sim X . \sim Y$

BOOLEAN ALGEBRA SIMPLIFICATION EXAMPLE 01

- Simplify the following boolean expression;

  $y = a.b + a + c$

BOOLEAN ALGEBRA SIMPLIFICATION EXAMPLE 01

- Simplify the following boolean expression;

$$y = a.b + a + c$$

$$= a(b + 1) + c$$

$$= a(1) + c$$

$$= a + c$$

BOOLEAN ALGEBRA SIMPLIFICATION EXAMPLE 02

- Negate the following:
    - $P \wedge Q$
    - $(P \wedge Q) \vee (R \wedge S)$

BOOLEAN ALGEBRA SIMPLIFICATION EXAMPLE 02

- Negate the following:
    - $P \wedge Q$
        - $\sim P \vee \sim Q$
    - $(P \wedge Q) \vee (R \wedge S)$
        - $(\sim P \vee \sim Q) \wedge (\sim R \vee \sim S)$

BOOLEAN ALGEBRA SIMPLIFICATION EXAMPLE 04

- Show that $\sim(p \vee (\sim p \wedge q))$ and $\sim p \wedge \sim q$ are logically equivalent.
- Solution shown on next slide.

BOOLEAN ALGEBRA SIMPLIFICATION EXAMPLE 04

- Show that ~(p ∨ (~p ∧ q)) and ~p ∧ ~q are logically equivalent.
- ~(p ∨ (~p ∧ q)) is equivalent to ~p ∧ [~(~p ∧ q)] (De Morgan)
- ~p ∧ [~(~p ∧ q)] is equivalent to ~p ∧ [~(~p) ∨ ~q] (De Morgan)
- ~p ∧ (p ∨ ~q) (double negation law)
- (~p ∧ p) ∨ (~p ∧ ~q) (distributive law)
- False ∨ (~p ∧ ~q) (since ~p ∧ p is equivalent to False)
- False ∨ (~p ∧ ~q) is equivalent to ~p ∧ ~q (identity law)

TAUTOLOGY AND CONTRADICTION

- A compound proposition that is always true, no matter what the truth values of the propositions that occur in it, is called a tautology.
- A compound proposition that is always false is called a contradiction.

EXAMPLE OF A TAUTOLOGY AND A CONTRADICTION

Tautology

Contradiction

P	$\sim P$	$P \vee \sim P$	$P \wedge \sim P$
T	F	T	F
F	T	T	F

HALF ADDER

- Let's try to build a circuit that adds two bits. This circuit is called the half adder.
  - We start by constructing the truth table for out input/output

HALF ADDER

- Let's try to build a circuit that adds two bits. This circuit is called the half adder.
  - We start by constructing the truth table for out input/output

A	B	Result	Carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

HALF ADDER

- Let's try to build a circuit that adds two bits. This circuit is called the half adder.
  - We start by constructing the truth table for out input/output

A	B	Result	Carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

What gate(s) does the carry represent?

HALF ADDER

- Let's try to build a circuit that adds two bits. This circuit is called the half adder.
  - We start by constructing the truth table for out input/output

A	B	Result	Carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

What gate(s) does the carry represent?

AND gate

HALF ADDER

- Let's try to build a circuit that adds two bits. This circuit is called the half adder.
  - We start by constructing the truth table for out input/output

A	B	Result	Carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

What gate(s) does the Result represent?

HALF ADDER

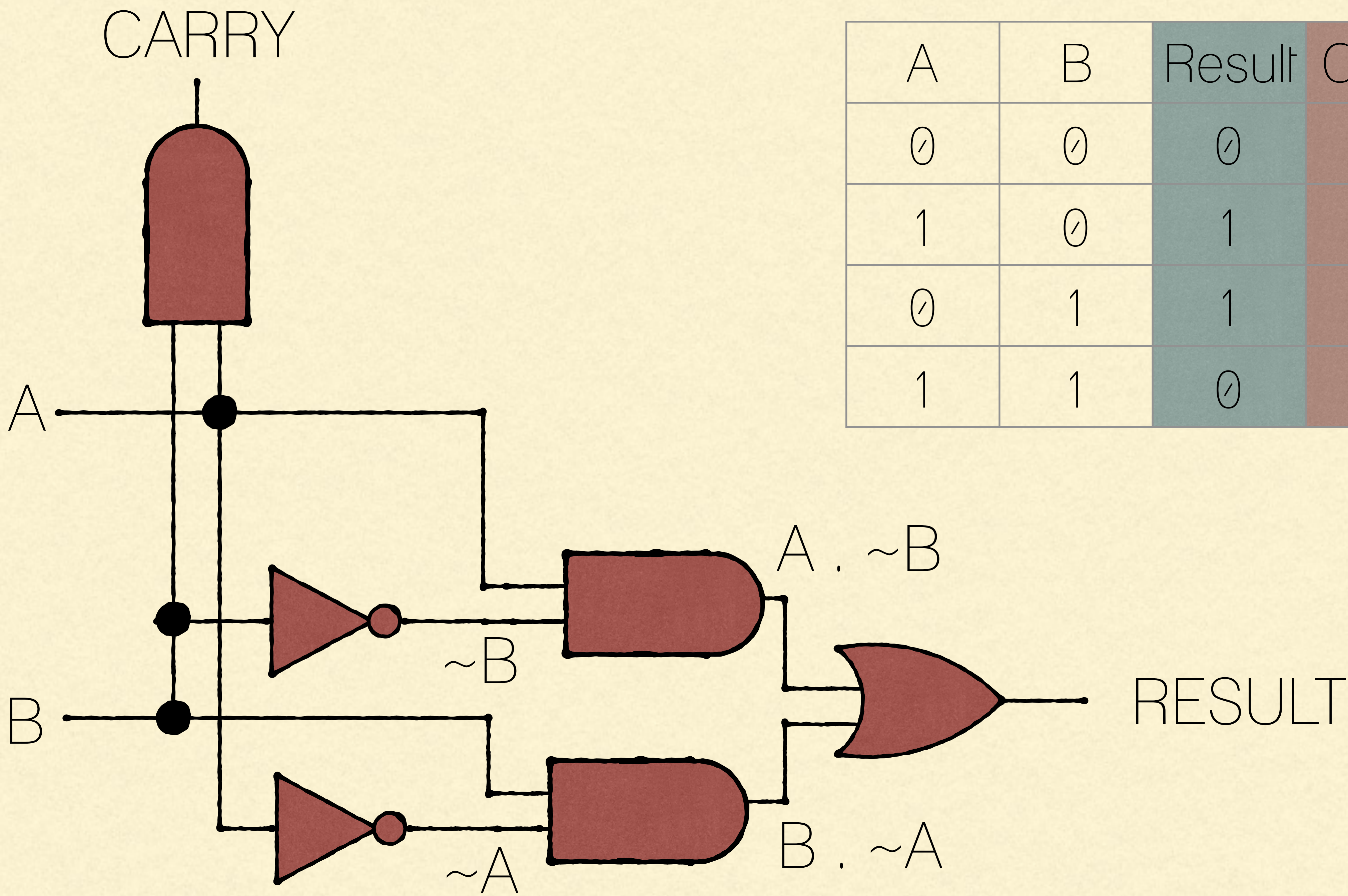
- Let's try to build a circuit that adds two bits. This circuit is called the half adder.
    - We start by constructing the truth table for out input/output

A	B	Result	Carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

What <u>gate(s)</u> does the Result represent?

EXCLUSIVE OR gate

BUILDING A HALF ADDER



A	B	Result	Carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

HALF ADDER

Can we build 32 half adders to add a 32-bit sequence?

HALF ADDER

Can we build 32 half adders to add a 32-bit sequence?

- No. The problem with a half-adder is that there it doesn't handle carries. We need a circuit that can add three bits.
    - That circuit is called a full adder.

FULL ADDER

- Here are the characteristics of a full adder.
  - Data inputs; 3 (let's call them X, Y, and CARRY IN)
  - Outputs; 2 (let's call them SUM, and CARRY OUT)
- How many rows would our truth table for the full adder be?

FULL ADDER

- Here are the characteristics of a full adder.
  - Data inputs; 3 (let's call them X, Y, and CARRY IN)
  - Outputs; 2 (let's call them SUM, and CARRY OUT)
- How many rows would our truth table for the full adder be?

THE ANSWER IS 8 (2^3)

FULL ADDER

- Here are the characteristics of a full adder.
  - Data inputs; 3 (let's call them X, Y, and CARRY IN)
  - Outputs; 2 (let's call them SUM, and CARRY OUT)

X	Y	CARRY IN	SUM	CARRY OUT
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

FULL ADDER

- Sum of products for the 'CARRY OUT';

$$X.Y.{\sim}C_{IN} + X.{\sim}Y.C_{IN} + {\sim}X.Y.C_{IN} + X.Y.C_{IN}$$

X	Y	CARRY IN	SUM	CARRY OUT
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

FULL ADDER

- Sum of products for the 'CARRY OUT';

$$X.Y.\sim C_{IN} + X.\sim Y.C_{IN} + \sim X.Y.C_{IN} + X.Y.C_{IN}$$

Can be further simplified to; $X.Y + Y.C_{IN} + X.C_{IN}$

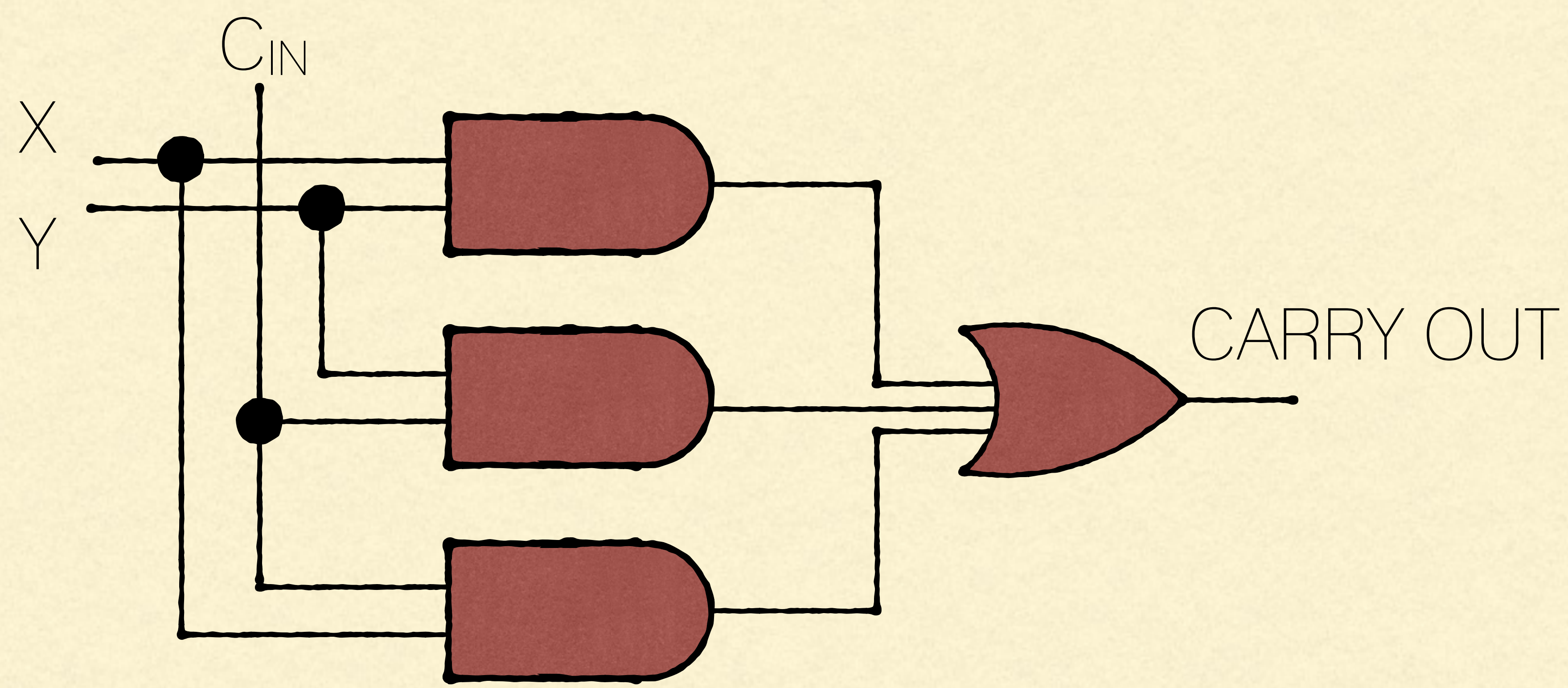
X	Y	CARRY IN	SUM	CARRY OUT
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

FULL ADDER

- Sum of products for the 'CARRY OUT';

$$X.Y.\sim C_{IN} + X.\sim Y.C_{IN} + \sim X.Y.C_{IN} + X.Y.C_{IN}$$

Can be further simplified to; $X.Y + Y.C_{IN} + X.C_{IN}$



FULL ADDER

- Sum of products for the 'SUM';

$$X.\sim Y.\sim C_{IN} + \sim X.Y.\sim C_{IN} + \sim X.\sim Y.C_{IN} + X.Y.C_{IN}$$

X	Y	CARRY IN	SUM	CARRY OUT
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

FULL ADDER

- Sum of products for the 'SUM';

$$X.\sim Y.\sim C_{IN} + \sim X.Y.\sim C_{IN} + \sim X.\sim Y.C_{IN} + X.Y.C_{IN}$$