ENGR 101
Engineering Technology

Dr. Kerese Manueli

School of Engineering and Computer Science
Victoria University of Wellington

Victoria
UNIVERSITY OF WELLINGTON
Te Whare Wānanga
o te Ūpoko o te Ika a Māui
CAPITAL CITY UNIVERSITY

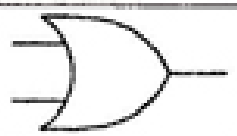
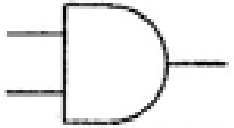
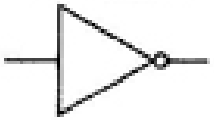
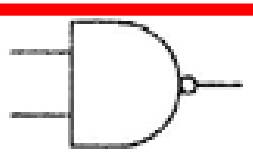
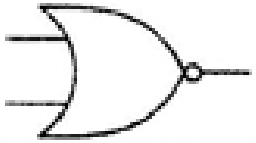
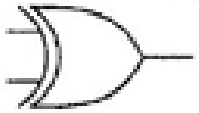
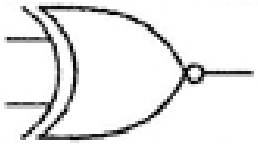
Week 3 Lecture 5b

- Main topics
  - Introduction to Engineering Technology
  - Number system
  - Logic Gates
  - Boolean Algebra

- Course web page:
  https://ecs.wgtn.ac.nz/Courses/XMUT101_2021T1/

- kerese@ecs.vuw.ac.nz

Logic Gates Symbols

Gate	Symbol
OR	
AND	
NOT	
NAND	
NOR	
EX-OR or X-OR	
EX-NOR or X-NOR	

Exclusive OR

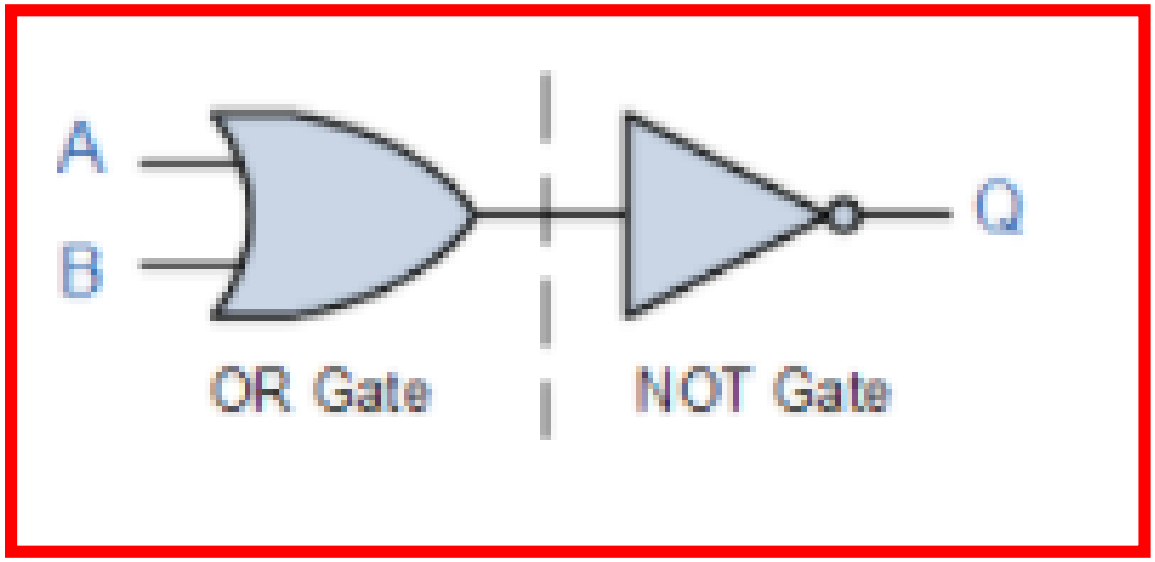
Exclusive NOR

NOR Gates

- Combine basic OR and NOT gate.

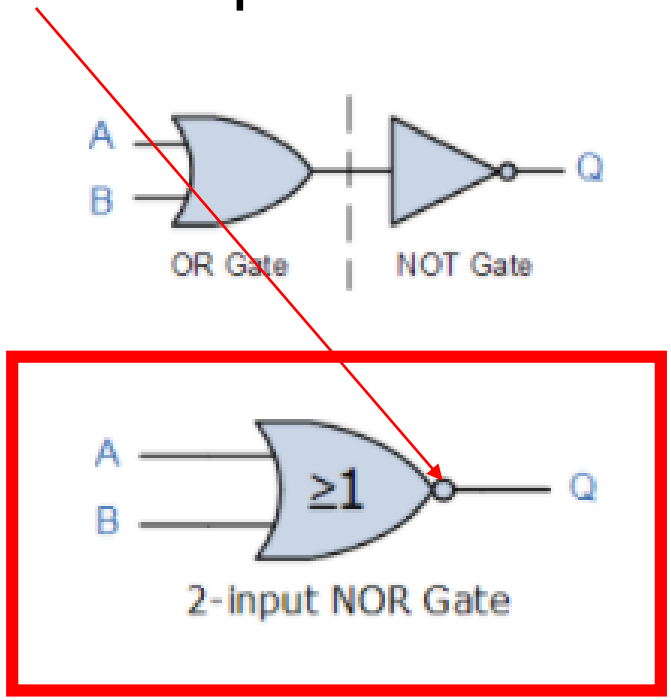
NOR Gates

- Combine basic OR and NOT gate.
- Not OR - NOR gate is an inverted OR gate.

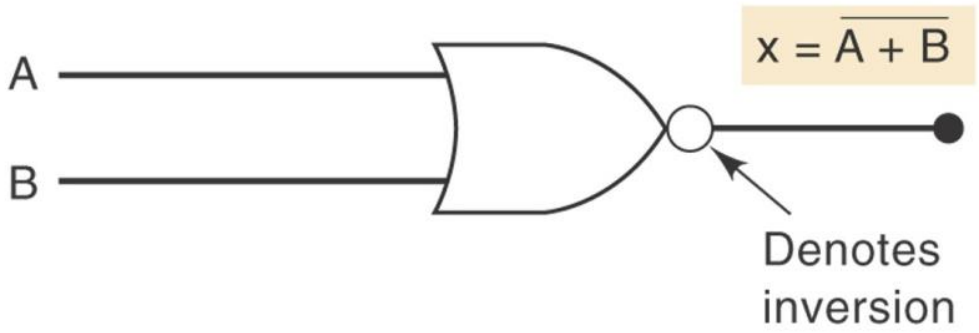


NOR Gates

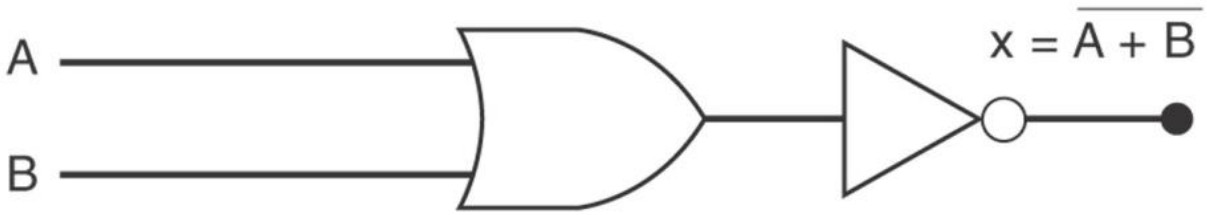
- Combine basic OR and NOT gate.
- Not OR - NOR gate is an inverted OR gate.
- An inversion “bubble” is placed at the output of the OR gate.



(a) NOR symbol



(b) Equivalent circuit; (c) Truth table.



(b)

OR NOR

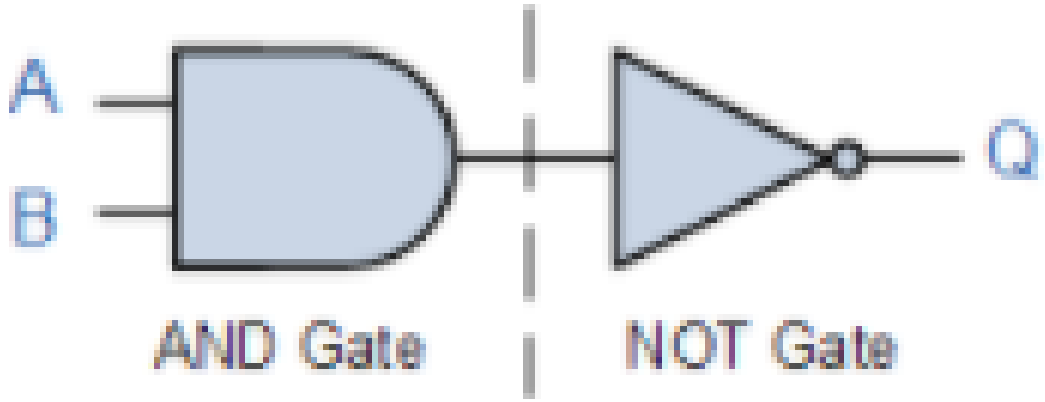
A	B	A + B	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

(c)

NOR Truth table

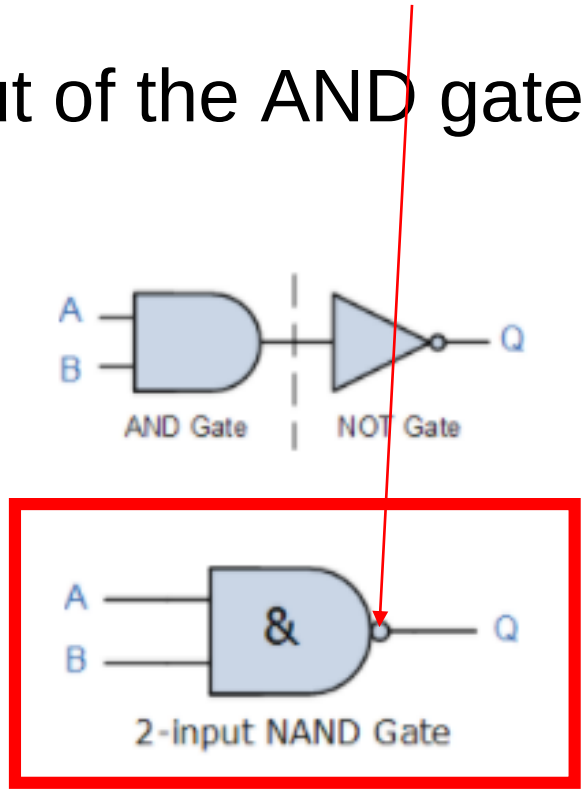
NAND Gates

- The NAND gate is an inverted AND gate.

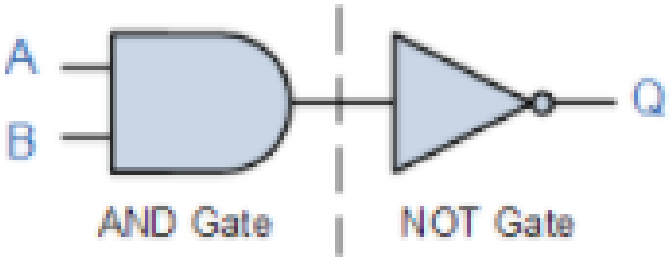


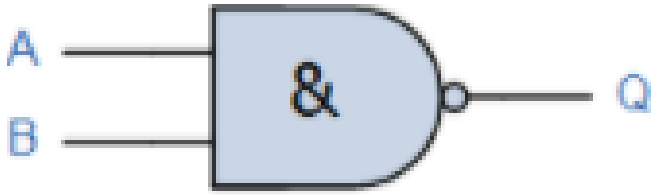
NAND Gates

- The NAND gate is an inverted AND gate.
- An inversion “bubble” is placed at the output of the AND gate.



NAND Gates





2-Input NAND Gate

NAND Truth table

A	B	A • B	$\overline{A \cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Boolean Algebra

- *Boolean algebra* is defined as a closed algebraic system containing a set K with two or more elements and the two operators: • (AND) & + (OR).

Boolean Algebra

- A *Boolean algebra* is defined as a closed algebraic system containing a set K or two or more elements and the two operators, . and +.
- Useful for identifying and *minimizing* circuit functionality

Boolean Algebra

- A *Boolean algebra* is defined as a closed algebraic system containing a set K or two or more elements and the two operators, . and +.
- Useful for identifying and *minimizing* circuit functionality
- Identity elements:

  a + 0 = a

  a . 1 = a

Boolean Algebra

- A *Boolean algebra* is defined as a closed algebraic system containing a set K or two or more elements and the two operators, . and +.
- Useful for identifying and *minimizing* circuit functionality
- Identity elements:

  a + 0 = a

  a . 1 = a

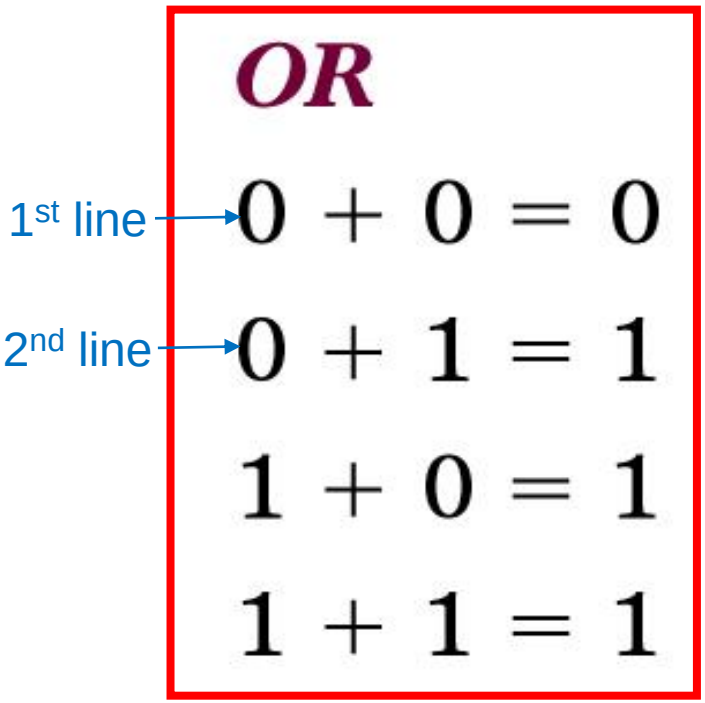
- 0 is the identity element for the + operation.

Boolean Algebra

- A *Boolean algebra* is defined as a closed algebraic system containing a set K or two or more elements and the two operators, . and +.
- Useful for identifying and *minimizing* circuit functionality
- Identity elements
  - a + 0 = a
  - a . 1 = a
- 0 is the identity element for the + operation.

  1 is the identity element for the . operation.

Boolean Algebra for OR gate



Boolean Algebra for OR, AND

OR	*AND*
$0 + 0 = 0$	$0 \cdot 0 = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$
$1 + 1 = 1$	$1 \cdot 1 = 1$

Boolean Algebra for OR, AND & NOT

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

Ordering Boolean Functions

How to interpret A • B + C?

The above expression is read as:

A “AND” B “OR” C

Ordering Boolean Functions

- How to interpret A • B + C?
  1. Is it A•B ORed with C ? (A•B) + C
  2. Is it A ANDed with B+C ? A • (B+C)

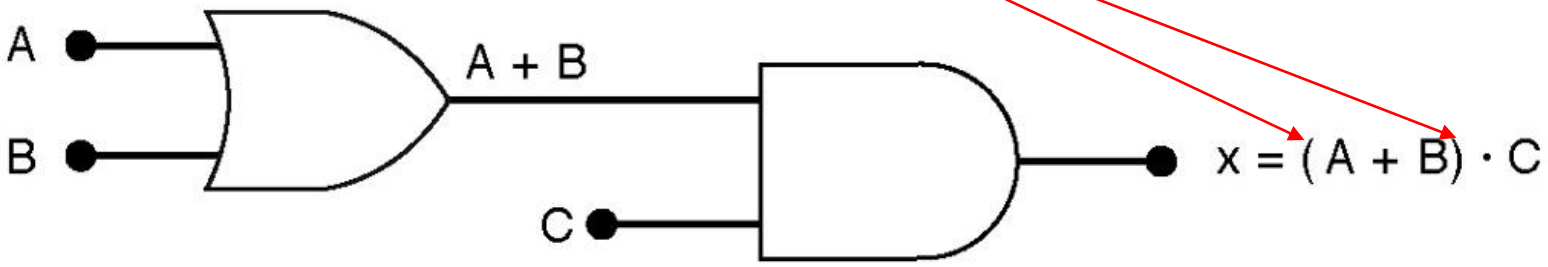
Ordering Boolean Functions

- How to interpret A•B+C?
  - Is it A•B ORed with C ?
  - Is it A ANDed with B+C ?
- Order of precedence for Boolean algebra:

  AND before OR.

Ordering Boolean Functions

- How to interpret A•B+C?
  - Is it A•B ORed with C ?
  - Is it A ANDed with B+C ?
- Order of precedence for Boolean algebra: AND before OR.
- Note that parentheses (or normal brackets) are needed here :



Commutativity and Associativity

- The Commutative Property:

  For every a and b in K,

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

  (2) $a \cdot b = b \cdot a$

Commutativity and Associativity

- The Commutative Property:

  For every a and b in K,

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

  (2) $a \cdot b = b \cdot a$

- The Associative Property:

  For every a, b, and c in K,

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

  (2) $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

Distributivity of the Operators and Complements

- The Distributive Property:

  For every a, b, and c in K,

  $a + (b \cdot c) = (a + b) \cdot (a + c)$

  $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Distributivity of the Operators and Complements

- The Distributive Property:

  For every a, b, and c in K,

  $a + (b \cdot c) = (a + b) \cdot (a + c)$

  $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

- The Existence of the Complement:

  For every a in K there exists a unique element called a'

  (*complement of a*) such that,

  $a + a' = 1$

  $a \cdot a' = 0$

Distributivity of the Operators and Complements

- The Distributive Property:

  For every a, b, and c in K,

  $a + (b \cdot c) = (a + b) \cdot (a + c)$

  $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

- The Existence of the Complement:

  For every a in K there exists a unique element called a' (*complement of a*) such that,

  $a + a' = 1$

  $a \cdot a' = 0$

- To simplify notation, the . operator is frequently omitted. When two elements are written next to each other, the AND (.) operator is implied…

$$a + b \cdot c = (a + b) \cdot (a + c)$$

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

Duality

- The principle of *duality* states that if an expression is valid in Boolean algebra, the dual of that expression is also valid.

Duality

- The principle of *duality* is an important concept. This says that if an expression is valid in Boolean algebra, the dual of that expression is also valid.
- To form the dual of an expression, replace all + operators with . operators, all . operators with + operators, all ones with zeros, and all zeros with ones.

Duality

- The principle of *duality* is an important concept. This says that if an expression is valid in Boolean algebra, the dual of that expression is also valid.
- To form the dual of an expression, replace all + operators with . operators, all . operators with + operators, all ones with zeros, and all zeros with ones.
- Form the dual of the expression

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

Duality

- The principle of *duality* is an important concept. This says that if an expression is valid in Boolean algebra, the dual of that expression is also valid.
- To form the dual of an expression, replace all + operators with . operators, all . operators with + operators, all ones with zeros, and all zeros with ones.
- Form the dual of the expression

  a + (bc) = (a + b)(a + c)

- Following the replacement rules…

  a (b + c) = a b + a c

Duality

- The principle of *duality* is an important concept. This says that if an expression is valid in Boolean algebra, the dual of that expression is also valid.
- To form the dual of an expression, replace all + operators with . operators, all . operators with + operators, all ones with zeros, and all zeros with ones.
- Form the dual of the expression

  $a + (bc) = (a + b)(a + c)$
- Following the replacement rules…

  $a(b + c) = ab + ac$
- Take care not to alter the location of the parentheses if they are present.

Involution

- This theorem states:

$$a'' = a$$

Involution

- This theorem states:
  a'' = a
- Remember that aa' = 0 and a+a'=1.
  - Therefore, a' is the complement of a and a is also the complement of a'.
  - As the complement of a' is unique, it follows that a''=a.

Involution

- This theorem states:

  a'' = a
- Remember that aa' = 0 and a+a'=1.
  - Therefore, a' is the complement of a and a is also the complement of a'.
  - As the complement of a' is unique, it follows that a''=a.
- Taking the double inverse of a value will give the initial value.

Absorption

- This theorem states:

$a + ab = a$

$a(a+b) = a$

Absorption

- This theorem states:

  a + ab = a a(a+b) = a

- To prove the first half of this theorem:

$$
\begin{aligned}
a + ab &= a \cdot 1 + ab \\
&= a\,(1 + b) \\
&= a\,(b + 1) \\
&= a\,(1) \\
a + ab &= a
\end{aligned}
$$

DeMorgan's Theorem

- A key theorem in simplifying Boolean algebra expression. It states:

$(a + b)' = a'b'$

$(ab)' = a' + b'$

DeMorgan's Theorem

- A key theorem in simplifying Boolean algebra expression is DeMorgan's Theorem. It states:

  $(a + b)' = a'b'$ $(ab)' = a' + b'$

- Complement the expression: $(a(b + z(x + a'))$ and simplify.

DeMorgan's Theorem

- A key theorem in simplifying Boolean algebra expression is DeMorgan's Theorem. It states:

1 $(a + b)' = a'b'$ 2 $(ab)' = a' + b'$

- Complement the expression $(a(b + z(x + a')))'$ and simplify.

$(a(b+z(x + a')))'$ $= a' + (b + z(x + a'))'$

$\overline{(a\ (b + z\ (x + \bar{a})))}$

DeMorgan's Theorem

- A key theorem in simplifying Boolean algebra expression is DeMorgan's Theorem. It states:

1 $(a + b)' = a'b'$ 2 $(ab)' = a' + b'$

- Complement the expression $(a(b + z(x + a')))'$ and simplify.

$(a(b+z(x + a')))'$ $= a' + (b + z(x + a'))'$

$= a' + b'(z(x + a'))'$

DeMorgan's Theorem

- A key theorem in simplifying Boolean algebra expression is DeMorgan's Theorem. It states:

1 $(a + b)' = a'b'$ 2 $(ab)' = a' + b'$

- Complement the expression $(a(b + z(x + a')))'$ and simplify.

$$(a(b+z(x + a')))' = a' + (b + z(x + a'))'$$
$$= a' + b'(z(x + a'))'$$
$$= a' + b'(z' + (x + a')')$$

DeMorgan's Theorem

- A key theorem in simplifying Boolean algebra expression is DeMorgan's Theorem. It states:

1 $(a + b)' = a'b'$ 2 $(ab)' = a' + b'$

- Complement the expression $(a(b + z(x + a')))'$ and simplify.

$(a(b+z(x + a')))'$ $= a' + (b + z(x + a'))'$

$= a' + b'(z(x + a'))'$

$= a' + b'(z' + (x + a')')$

$= a' + b'(z' + x'a'')$

DeMorgan's Theorem

- A key theorem in simplifying Boolean algebra expression is DeMorgan's Theorem. It states:

1 $(a + b)' = a'b'$ 2 $(ab)' = a' + b'$

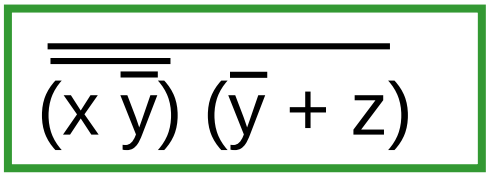
- Complement the expression $(a(b + z(x + a')))'$ and simplify.

$$
\begin{aligned}
(a(b+z(x + a')))' &= a' + (b + z(x + a'))' \\
&= a' + b'(z(x + a'))' \\
&= a' + b'(z' + (x + a')') \\
&= a' + b'(z' + x'a'') \\
&= a' + b'(z' + x'a)
\end{aligned}
$$

DeMorgan's Theorem - Exercise

Simplify the expression:

((x y')'(y'+z))'



DeMorgan's Theorem - Exercise

Simplify the expression:

((x y')'(y'+z))'

$$\overline{(x\,\bar{y})\,(\bar{y}+z)}$$

1 (a + b)' = a'b'

2 (ab)' = a' + b'

5 minutes .....

DeMorgan's Theorem - Exercise

Simplify the expression:

((x y')'(y'+z))'

$$\overline{\overline{(x\,\overline{y})}\;(\overline{y}+z)}$$

$$= \overline{\overline{\overline{(X \cdot \overline{Y})}} \cdot (\overline{Y} + Z)}$$

$$= (\overline{\overline{(X \cdot \overline{Y})}} + \overline{(\overline{Y} + Z)})$$

$$= (X \cdot \overline{Y}) + (\overline{\overline{Y}} \cdot \overline{Z})$$

$$= (X \cdot \overline{Y}) + (Y \cdot \overline{Z})$$

$$= X\overline{Y} + Y\overline{Z}$$

Week 3 Lecture 5b

- NOR and NAND gates
- Boolean algebra
  - Rules
  - DeMorgan's Theorems

- Course web page:

https://ecs.wgtn.ac.nz/Courses/XMUT101_2021T1/

- kerese@ecs.vuw.ac.nz