2.5

Website: contoureducation.com.au | **Phone:** 1800 888 300
Email: hello@contoureducation.com.au

VCE Specialist Mathematics ½
Logic & Algorithms II [2.5]
Workbook

Outline:

Learning Objectives:

- [ ] **SM12 [2.5.1]** - Understand The basics of logic and propositional statements
- [ ] **SM12 [2.5.2]** - Construct truth tables and recognise equivalent logical expressions
- [ ] **SM12 [2.5.3]** - Represent logical expressions using switching circuits and logic gates
- [ ] **SM12 [2.5.4]** - Simplify and evaluate Boolean algebra expressions using algebraic identities and Karnaugh maps

Section A: Logic

Sub-Section: Introduction to Logic

What is logic used for in maths?

Logic

$$\textit{Is it true}(1)\textit{ or false}(0)?$$

- Logic is the way we determine what is true in maths, and the conditions for these things to be true.

Space for Personal Notes

Question 1

State if the following statements are true or false.

a. Milk is white.

True

b. Milk is white and water is red.

True and false } false

Space for Personal Notes

Sub-Section: Connectives

Active Recall

- Logic is a way we determine what is true in maths.

How do we connect multiple propositions?

Connectives

$$(P \wedge Q) = \text{"}P \textit{ and } Q\text{"}$$

$$(P \vee Q) = \text{"}P \textit{ or } Q\text{"}$$

$$P \rightarrow Q = \text{"}\textit{if } P\text{, } \textit{then } Q\text{"}$$

$$P \leftrightarrow Q = \text{"}P \textit{ if and only if } Q\text{"}$$

- The symbols are called connectives, as their name suggests, they help us connect different propositions together.

Space for Personal Notes

Question 2 Walkthrough.

Translate the following into English:

P = "I cheat", R = "I will write an exam", Q = "I will get caught", S = "I will fail".

$$(R \wedge P) \to (Q \wedge S)$$

If I will write an exam AND I cheat, then I will get caught AND I will fail

Space for Personal Notes

Question 3

Translate the following into English:

P = "I get 20 RAW in further", R = "I fail English Exam", Q = "Parents are mad", S = "Have no dinner".

$$R \vee P \rightarrow Q \wedge S$$

If I fail an English exam OR I get a 20 raw, then parents are mad AND I will have no dinner

Space for Personal Notes

Question 4

Translate into propositional logic using the correct syntax.

If David does not die, then Mary will not get any money and David's family will be happy.

P = David dies

Q = Mary gets money

R = David's family would be happy.

Space for Personal Notes

$\neg P \rightarrow \neg Q \wedge R$

Sub-Section: Truth Tables

Is there a way to visualise the logic?

Truth Tables

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

- Is a way to visualise the logic for connectives.
- Instead of true or false, we can write 1 and 0 respectively.

Let's consider all cases for truth tables.

Case 1: Negations $(\neg, \sim)$

- $\text{Neg } p = 1 - P$

p	$\neg p$
1	0
0	1

Case 2: Conjunction ($\wedge$, &)

- $p \wedge q = \min(p, q)$

p	q	$p \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

- **Example**: Your mum says that she will buy you a PS5 if you get raw 50 in both Specialist Maths and Maths Methods. We can summarise the situations and outcomes in a diagram comparable to a truth table.

You get raw 50 in MM	You get raw 50 in SM	You get a PS5
Yes	Yes	Yes
Yes	No	No
No	Yes	No
No	No	No

Case 3: Disjunction $(\vee, +)$

- $p \vee q = \max\ (p, q)$

p	q	$p \vee q$
1	1	1
1	0	1
0	1	1
0	0	0

- **Example**: Your mum says that she will give you $300 if you get raw 50 in either Specialist Maths or Maths Methods. We can summarise the situations and outcomes in a diagram comparable to a truth table.

You get raw 50 in MM	You get raw 50 in SM	You get a $300
Yes	Yes	Yes
Yes	No	Yes
No	Yes	Yes
No	No	No

Case 4: Conditional $(\rightarrow)$

p	q	$p \rightarrow q$
1	1	1
1	0	0
0	1	1
0	0	1

- This can be a bit confusing to wrap our heads around, but consider the following statement:
  - If it is raining, I will wear a raincoat.
- If it is raining, and I am, in fact wearing a raincoat, I am not lying to you.
- If it is NOT raining outside, and I am not wearing a raincoat, I am also not lying to you.

0 0

- If it is NOT raining outside, and I am wearing a raincoat, I am also not lying to you.

0 1

- If it is raining, and I am, in fact NOT wearing a raincoat, I AM lying to you.
- $p \rightarrow q$ is only false if p is true and q is false.

$$p \rightarrow q = \begin{cases} 1, if\ p \leq q \\ 0, otherwise \end{cases}$$

Space for Personal Notes

Case 5: Biconditional ($\leftrightarrow, \equiv$) If and only if

p	q	$p \leftrightarrow q$
1	1	1
1	0	0
0	1	0
0	0	1

$$p \leftrightarrow q = \begin{cases} 1, if\ p = q \\ 0, otherwise \end{cases}$$

Case 6: Exclusive-Or ($\underline{\vee}$ or $\oplus$)

p	q	$p \underline{\vee} q$
1	1	0
1	0	1
0	1	1
0	0	0

- The "exclusive or" function is written XOR in some programming languages.

Space for Personal Notes

Question 5 Walkthrough.

Using the truth table, show that ($p \vee \neg p$) is always true.

p	$\neg p$	$p \vee \neg p$
1	0	1
0	1	1

Space for Personal Notes

NOTE: What we have just described above is a tautology, a statement that is true by necessity or just by how it is formed logically.

Question 6

Using the truth table, show that $(p \wedge \neg p)$ is always false.

p	$\neg p$	$p \wedge \neg p$
1	0	0
0	1	0

Space for Personal Notes

NOTE: What we have just described above is a contradiction, a statement that is false by necessity or just by how it is formed logically.

Sub-Section: Equivalence

Equivalence

$$A \equiv B$$

- **Definition:**
  - Equivalence is when two statements are the same.

Useful Equivalences

- Equivalence Law:
  - $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
- Implication Law:
  - $p \rightarrow q \equiv (\neg p) \vee q$
- Double Negation Law:
  - $\neg\neg p \equiv p$
- Idempotent Laws:
  - $p \wedge p \equiv p$
  - $p \vee p \equiv p$
- Commutative Laws:
  - $p \wedge q \equiv q \wedge p$
  - $p \vee q \equiv q \vee p$
- Associative Laws:
  - $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$
  - $p \vee (q \vee r) \equiv (p \vee q) \vee r$

- Distributive Laws:
  - $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
  - $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
- De Morgan's Laws:
  - $\neg(p \wedge q) \equiv (\neg p) \vee (\neg q)$
  - $\neg(p \vee q) \equiv (\neg p) \wedge (\neg q)$
- Identity Laws:
  - $p \wedge T \equiv p$
  - $p \vee F \equiv p$
- Inverse Laws:
  - $p \wedge (\neg p) \equiv F$
  - $p \vee (\neg p) \equiv T$

Space for Personal Notes

Question 7

Use a truth table for the following questions. NO

a. Is $(p \wedge q)$ is logically equivalent to $\neg(p \vee q)$?

NOTE: "Logical equivalence" means that the combination of statements carries the same truth values.

p	q	$p \wedge q$	$p \vee q$	$\neg(p \vee q)$
1	1	1	1	0
1	0	0	1	0
0	1	0	1	0
0	0	0	0	1

b. Is $\neg(p \wedge q)$ logically equivalent to $(\neg p \vee \neg q)$?

p	q	$p \wedge q$	$\neg(p \wedge q)$	$\neg p$	$\neg q$	$\neg p \vee \neg q$
1	1	1	0	0	0	0
1	0	0	1	0	1	1
0	1	0	1	1	0	1
0	0	0	1	1	1	1

Space for Personal Notes

Logical equivalence

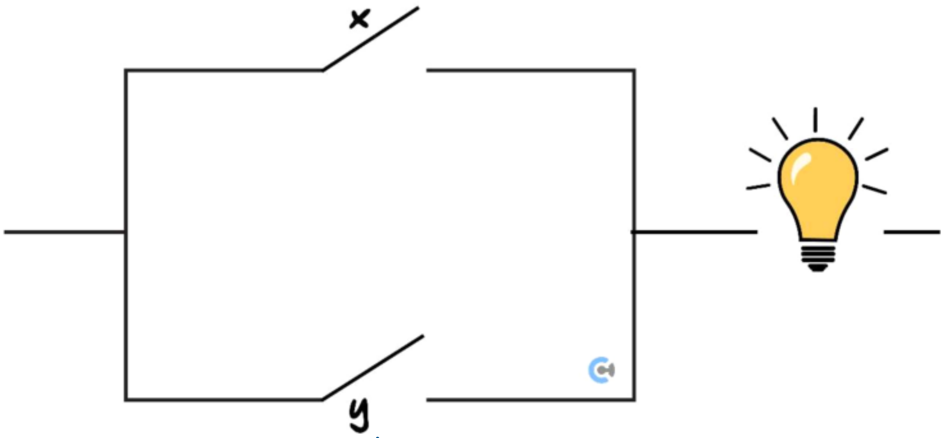
Sub-Section: Circuit Representation

Discussion: How could we use circuits to represent true or false?

'On' → True
'Off' → False

Switching Circuits

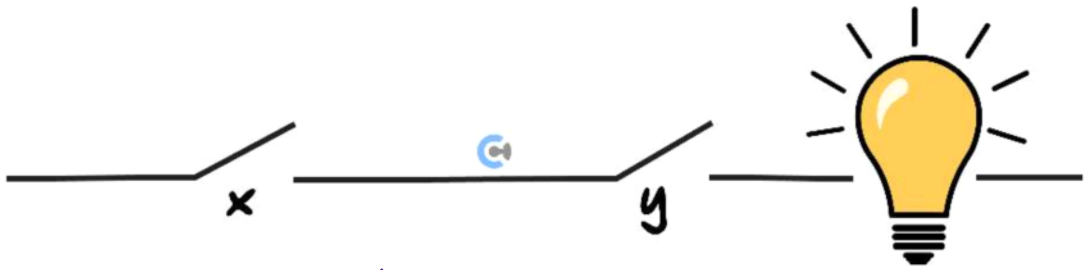
- Is a visual way of representing logic.
- **Disjunction** ($\vee$, +)



Switches x and y in parallel.

x	y	State of System
Open 0	Open 0	Open 0
Open 0	Closed 1	Closed 1
Closed 1	Open 0	Closed 1
Closed 1	Closed 1	Closed 1

Conjunction ($\wedge$, &)



Switches x and y in series.

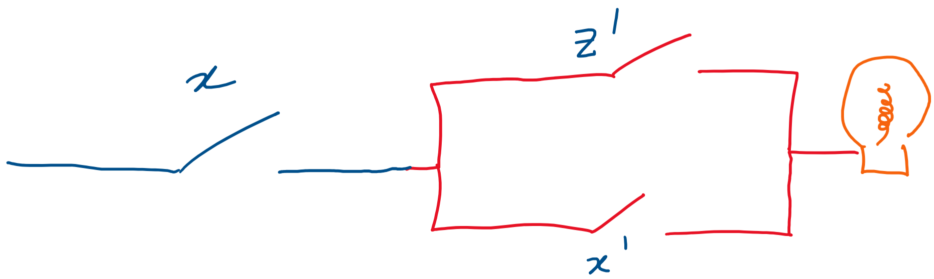
x	y	State of System
Open 0	Open 0	Open 0
Open 0	Closed 1	Open 0
Closed 1	Open 0	Open 0
Closed 1	Closed 1	Closed 1

Space for Personal Notes

Question 8 Walkthrough.

Draw a switching circuit that is represented by the following expression.

$$\overbrace{x \wedge \underbrace{(z' \vee x')}_{\text{parallel}}}^{\text{series}}$$



Space for Personal Notes

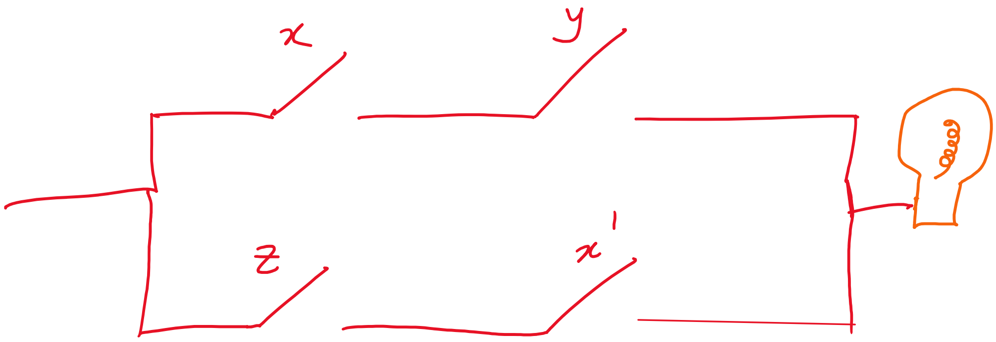
Question 9

parallel.

Draw the switching circuit that is represented by the following expression.

$$(x \wedge y) \vee (z \wedge x')$$

series



Space for Personal Notes

Section B: Boolean Algebra

Sub-Section: Introduction to Boolean Algebra

What is Boolean Algebra?

Boolean Algebra

$$1 = True$$

$$0 = False$$

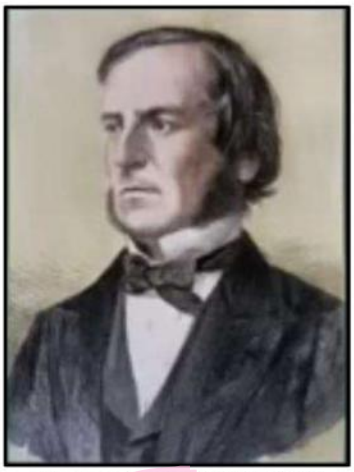
- Algebra of trues and falses (1s and 0s)
- The set of rules used to simplify logical expressions without changing their functionality.

T = 0	$\bar{x}$	not	$\neg P$
x ∨ y	$x + y$	or	$P \vee Q$
x ∧ y	xy	and	$P \wedge Q$
	$x \to y / \bar{x} + y$	implication	$P \to Q$
	$x \equiv y$	equivalence	$P \leftrightarrow Q$

Space for Personal Notes

Context: George Boole

- English mathematician George Boole came up with Boolean Algebra.



George Boole (1815-1864)

- His goal was to find a set of mathematical axioms that could reproduce the classical results of logic.

Space for Personal Notes

Question 10

Evaluate the following using Boolean algebra.

a. $1 + 0$ $=1$

$1 \vee 0 = 1$

b. $1 + 1$ $= 1$

$1 \vee 1 = 1$

Space for Personal Notes

NOTE: 1 and 0 represent true and false respectively.

Let's summarise!

Fundamental Laws and Theorems of Boolean Algebra

- $\overline{\mathbf{1}} = \mathbf{0}$
- $\overline{\mathbf{0}} = \mathbf{1}$
- And, just like how we had the useful laws for equivalences involving connectives, we have the following Laws of Boolean Algebra:

1. $X + 0 = X$
2. $X + 1 = 1$
3. $X + X = X$
4. $X + \bar{X} = 1$
5. $X \cdot 0 = 0$
6. $X \cdot 1 = X$
7. $X \cdot X = X$
8. $X \cdot \bar{X} = 0$
9. $\bar{\bar{X}} = X$
10. $X + Y = Y + X$
11. $XY = YX$
12. $(X + Y) + Z = X + (Y + Z)$
13. $(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$
14. $X(Y + Z) = XY + XZ$
15. $X + Y \cdot Z = (X + Y) \cdot (X + Z)$
16. $\overline{X + Y} = \bar{X} \cdot \bar{Y}$
17. $\overline{X \cdot Y} = \bar{X} + \bar{Y}$

Question 11 Walkthrough.

Simplify the following Boolean expression, stating which of the laws and/or theorems of Boolean Algebra you used in your working.

$$(X+Y)(X+\bar{Y})(\bar{X}+Z)$$

$X \cdot X = X$

$Y \cdot \bar{Y} = 0$

$X \cdot X + X \cdot \bar{Y} + Y \cdot X + Y \cdot \bar{Y}$

$X + X\bar{Y} + XY$

$X + X(\underbrace{\bar{Y} + Y}_{=1})$

$X + X = X$

$X(\bar{X} + Z)$

$X \cdot \bar{X} + XZ$

$X \wedge \bar{X} = 0$

$\boxed{XZ}$

Space for Personal Notes

Question 12

Simplify the following Boolean expressions, stating which of the laws and/or theorems of Boolean Algebra you used in your working.

a. $X = A\,B\,C + \bar{A}\,B + A\,B\,\bar{C}$

$$X = ABC + AB\bar{C} + \bar{A}B$$

$$= AB\underbrace{(C+\bar{C})}_{=1} + \bar{A}B$$

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

$$= \underbrace{(A+\bar{A})}_{=1}B$$

$$= B$$

b. $X\,Y\,Z + X\,\bar{Y}\,Z + X\,Y\,\bar{Z}$

$$= XZ\overbrace{(Y+\bar{Y})}^{=1} + XY\bar{Z}$$

$$= XZ + XY\bar{Z}$$

$$= X(Z + Y\bar{Z})$$

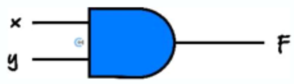
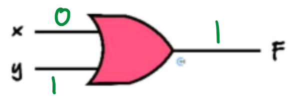
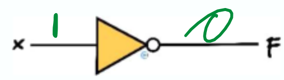
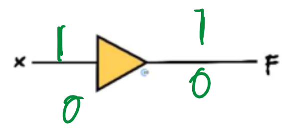
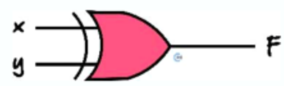
$$= X((Z+Y)(Z+\bar{Z}))$$

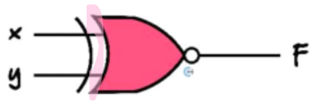
$$= X(Z+Y)$$

Sub-Section: Logic Gate Representation

Logic Gate

- Boolean algebra is generally used in the creation and simplification of electronic circuits.
- We design models for these electronic circuits by making each of our logical operations a type of gate: a logic gate.

Name	Graphic Symbol	Algebraic Function	Truth Table
AND $x \wedge y$		$F = xy$	x: 0, 0, 1, 1 y: 0, 1, 0, 1 F: 0, 0, 0, 1
OR $x \vee y$		$F = x + y$	x: 0, 0, 1, 1 y: 0, 1, 0, 1 F: 0, 1, 1, 1
Inverter		$F = x'$	x: 0, 1 F: 1, 0
Buffer		$F = x$	x: 0, 1 F: 0, 1
Exclusive-OR(XOR)		$F = xy' + x'y$ $= x \oplus y$	x: 0, 0, 1, 1 y: 0, 1, 0, 1 F: 0, 1, 1, 0

<table>
<tr>
<td>Exclusive-NOR or equivalence</td>
<td>
</td>
<td>$F = xy + x'y'$
$= (x \oplus y)'$</td>
<td>
<table>
<tr><th>x</th><th>y</th><th>F</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
</table>

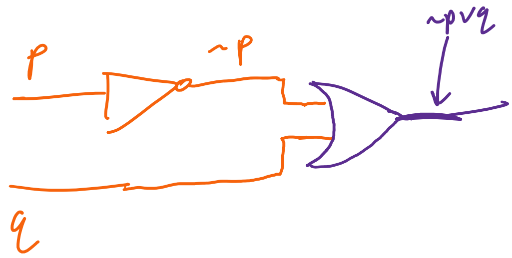
Space for Personal Notes

Question 13 Walkthrough.

Use logic gates to represent the following expressions and draw the corresponding truth tables:

$$\sim p \vee q$$

p	q	$\sim p$	$\sim p \vee q$
1	1	0	1
1	0	0	0
0	1	1	1
0	0	1	1



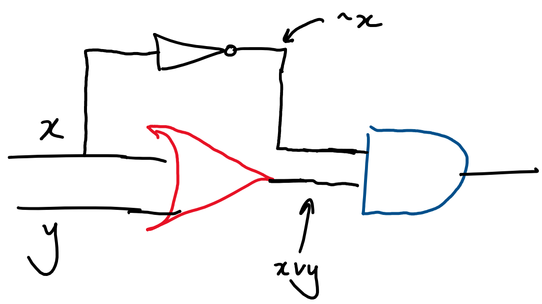
Space for Personal Notes

Question 14

Use logic gates to represent the following expressions and draw the corresponding truth tables:

$$(x \vee y) \wedge {\sim}x$$

x	y	$x \vee y$	$\sim x$	$(x \vee y) \wedge \sim x$
1	1	1	0	0
1	0	1	0	0
0	1	1	1	1
0	0	0	1	0



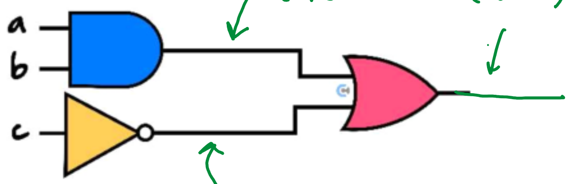
Space for Personal Notes

Question 15 Walkthrough.

Write down the Boolean expression for the circuits below.

$a \wedge b$

$(a \wedge b) \vee \neg c$



$\neg c$

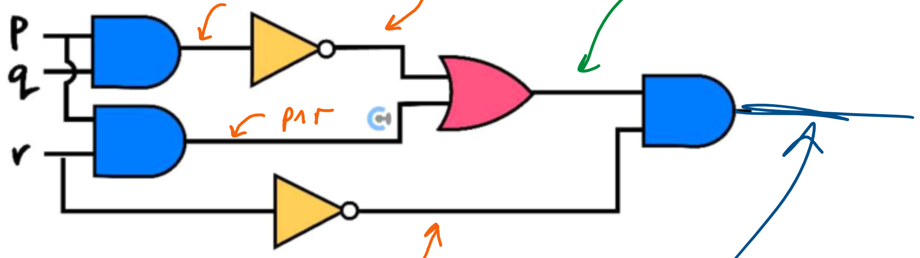
$ab + \overline{c}$

Space for Personal Notes

Question 16

Write down the Boolean expression for the circuits below. $\neg(p \wedge q)$ $\quad \neg(p \wedge q) \vee (p \wedge r)$

$p \wedge q$



$p \wedge r$

$\neg r$

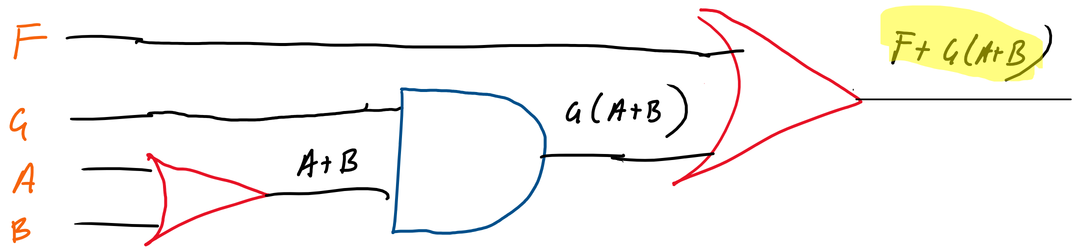
$\left(\neg(p \wedge q) \vee (p \wedge r)\right) \wedge \neg r$

$\left(\overline{pq} + pr\right)\bar{r}$

Space for Personal Notes

Question 17

Draw the network using logic gates representing $F + G(A + B)$.



Space for Personal Notes

Section C: Karnaugh Maps

Creating Karnaugh Maps (K-Maps)

- Let's look at how to use a K-Map using an example.
- Consider the following Boolean expression and the associated truth table:

$A \vee B$

$$F = A + B \text{ (i)}$$

OR

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

- The corresponding K-Map is constructed by the following conventions:
  - Draw a grid, with one square for each row in the truth table.
  - The possible values of A are written as column headings on the top.
  - The possible values of B are written as row levels along the LHS.

Karnaugh Map

$B \backslash A$	0	1
0	0	1
1	1	1

F

  - The input values will act as coordinates for the output values.

- Once the K-Map is filled in it will look like this:

B \ A	0	1
0	0	1
1	1	1

F

$F = A + B$

$\vee$

Space for Personal Notes

Question 18 Walkthrough.

Create Karnaugh Maps for the AND function.

B \ A	0	1
0	0	0
1	0	1

Space for Personal Notes

Question 19

Create Karnaugh Maps for the XOR, and Equivalence functions.

XOR

B \ A	0	1
0	0	1
1	1	0

$\equiv$

Equivalence

B \ A	0	1
0	1	0
1	0	1

Space for Personal Notes

Okay... what is it useful for?

Deriving Expressions from a Karnaugh Map

- A K-Map can be used to derive the simplest possible Boolean expression.
- Consider the following (different) truth table and the corresponding K-Map:

A	B	F
0	0	0
0	1	0
1	0	1
1	1	1

Karnaugh Map

B \ A	0	1
0	0	1
1	0	1

F

- To derive the simplist possible boolean expression from the K-Map, we are looking for the largest possible groupings. of 1.
- In our K-Map, the largest possible grouping of ones is as follows:

B \ A	0	1
0	0	1
1	0	1

F

- Here, we can see that:
  - Whenever there is a 1 in the group, the value of input A is 1.
  - The topmost 1 corresponds to a value of 0 for input B.
  - But, the bottom 1 corresponds to the value of 1 for input B.
- As such, the output is independent of the input B, which makes B a redundant input.
- As such, $F = A$
- Consider this third truth table and the corresponding K-Map:

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

B \ A	0	1
0	0	1
1	1	1

F

  - **Rule: A single group of 1s CANNOT be "L-Shaped".**

- As such, here we have two grouping of 1s.

$B \backslash A$	0	1
0	0	1
1	1	1

F

- The 1s in the vertical group always occur with input <u>A=1</u>, therefore they can be matched to the expression A.
- The 1s in the horizontal group always occur with input <u>B=1</u>, therefore they can be matched to the expression B.
- As such, <u>$F = A \vee B$</u>

Space for Personal Notes

Question 20

Construct a K-Map corresponding to the following truth table and hence determine a Boolean Expression for F in terms of A and B.

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

B \ A	0	1
0	1	1
1	1	0

$$F = \neg A \vee \neg B$$

Space for Personal Notes

Karnaugh Maps With 3 Variables

- When there are 3 variables, it is often best practice to:
  - place one variable and its possibles values as column headings, and
  - the possible combinations of values for the remaining variables as row levels.

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

BC \ A	0	1
00	0	1
01	0	1
11	1	1
10	1	1

- When we construct our groups, we find that the Boolean expression for F is $F =$ $A \vee B$

C \ AB	00	01	11	10
0	0	1	1	1
1	0	1	1	1

WRONG!

C \ AB	00	01	11	10
0	0	1	1	1
1	0	1	1	1

CORRECT!

Space for Personal Notes

Question 21

Construct a K-Map corresponding to the following truth table and hence determine a Boolean Expression for F in terms of A, B, and C.

A	***B***	***C***	***F***
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

BC \ A	0	1
00	1	1
01	0	1
11	0	1
10	1	1

$F = A \vee \neg C$

Space for Personal Notes

Contour Checklist

Learning Objective: [2.5.1] – Understand the Basics of Logic and Propositional Statements

Key Takeaways

- **Connectives**

$(P \wedge Q) =$ "P and Q"

$(P \vee Q) =$ "P or Q"

$P \to Q =$ "if P, then Q"

$P \leftrightarrow Q =$ "P if and only if Q"

  - The symbols are called connectives, as their name suggests, they help us connect different propositions together.

Learning Objective: [2.5.2] – Construct Truth Tables and Recognise Equivalent Logical Expressions

Key Takeaways

- **Case 1: Negations** $(\neg, \sim)$

  - Neg $p = 1 - P$

p	$\neg p$
1	0

- **Case 2: Conjunction** $(\wedge, \&)$
  - $p \wedge q = \min(p, q)$

p	q	$p \wedge q$
1	1	1
1	0	0
0	1	0
0	0	0

- **Case 3: Disjunction** $(\vee, +)$
  - $p \vee q = \max(p, q)$

p	q	$p \vee q$
1	1	1
1	0	1
0	1	1
0	0	0

Case 4: Conditional $(\rightarrow)$

p	q	$p \rightarrow q$
1	1	1
1	0	0
0	1	1
0	0	1

- $p \rightarrow q$ is only false if p is true and q is false.

$$p \rightarrow q = \begin{cases} 1, if\ p \leq q \\ 0, otherwise \end{cases}$$

Case 5: Biconditional $(\leftrightarrow, \equiv)$

p	q	$p \leftrightarrow q$
1	1	1
1	0	0
0	1	0
0	0	1

$$p \leftrightarrow q = \begin{cases} 1, if\ p = q \\ 0, otherwise \end{cases}$$

- **Case 6: Exclusive-Or** ($\underline{\vee}$ or $\oplus$)

p	q	$p \underline{\vee} q$
1	1	0
1	0	1
0	1	1
0	0	0

  - The "exclusive or" function is written XOR in some programming languages.

- **Equivalence**

$$A \equiv B$$

  - **Definition:**
    - Equivalence is when two statements are the same.

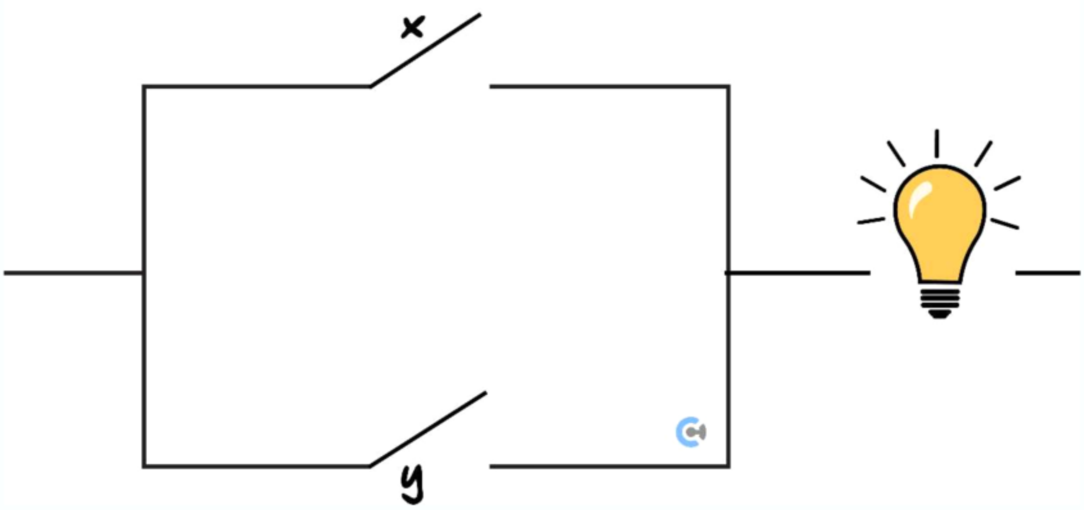
- **Useful Equivalences**
  - Equivalence Law
    - $p \leftrightarrow q \equiv$ $(p \to q) \wedge (q \to p)$
  - Implication Law
    - $p \to q \equiv$ $\neg p \vee q$
  - Double Negation Law
    - $\neg\neg p \equiv$ p
  - Idempotent Laws
    - $p \wedge p \equiv$ p
    - $p \vee p \equiv$ p

- Commutative Laws
  - $p \wedge q \equiv$ $q \wedge p$
  - $p \vee q \equiv$ $q \vee p$
- Associative Laws
  - $p \wedge (q \wedge r) \equiv$ $(p \wedge q) \wedge r$
  - $p \vee (q \vee r) \equiv$ $(p \vee q) \vee r$
- Distributive Laws
  - $p \wedge (q \vee r) \equiv$ $(p \wedge q) \vee (p \wedge r)$
  - $p \vee (q \wedge r) \equiv$ $(p \vee q) \wedge (p \vee r)$
- De Morgan's Laws
  - $\neg(p \wedge q) \equiv$ $\neg p \vee \neg q$
  - $\neg(p \vee q) \equiv$ $\neg p \wedge \neg q$
- Identity Laws
  - $p \wedge \mathrm{T} \equiv$ p
  - $p \vee \mathrm{F} \equiv$ p
- Inverse Laws
  - $p \wedge (\neg p) \equiv$ F
  - $p \vee (\neg p) \equiv$ T

Learning Objective: [2.5.3] – Represent Logical Expressions Using Switching Circuits and Logic Gates

Key Takeaways

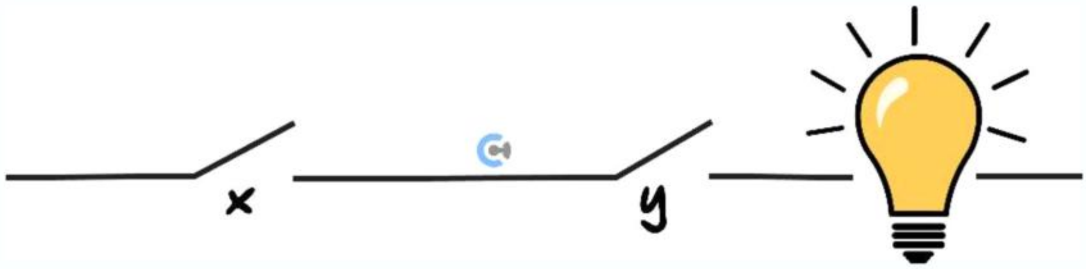
- **Switching Circuits**
  - Is a visual way of representing logic.
  - **Disjunction** $(\vee, +)$



- Switches x and y in parallel.

x	y	State of System
Open	Open	Open
Open	Closed	Closed
Closed	Open	Closed
Closed	Closed	Closed

- **Conjunction** ($\wedge$, &)

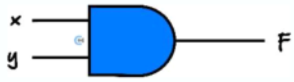
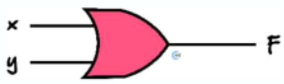
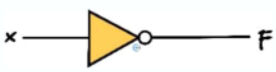
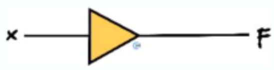
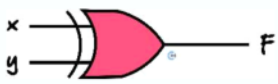
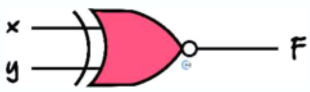


- Switches x and y in series.

x	y	State of System
Open	Open	Open
Open	Closed	Open
Closed	Open	Open
Closed	Closed	Closed

Logic Gate

- Boolean algebra is generally used in the creation and simplification of electronic circuits.
- We design models for these electronic circuits by making each of our logical operations a type of gate: a logic gates.

Name	Graphic Symbol	Algebraic Function	Truth Table
AND		$F = xy$	x y F: 0 0 0; 0 1 0; 1 0 0; 1 1 1
OR		$F = x + y$	x y F: 0 0 0; 0 1 1; 1 0 1; 1 1 1
Inverter		$F = x'$	x F: 0 1; 1 0
Buffer		$F = x$	x F: 0 0; 1 1
Exclusive-OR(XOR)		$F = xy' + x'y$ $= x\oplus y$	x y F: 0 0 0; 0 1 1; 1 0 1; 1 1 0
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x\oplus y)'$	x y F: 0 0 1; 0 1 0; 1 0 0; 1 1 1

Learning Objective: [2.5.4] – Simplify and Evaluate Boolean Algebra Expressions Using Algebraic Identities and Karnaugh Maps

Key Takeaways

Fundamental Laws and Theorems of Boolean Algebra

- $\overline{\mathbf{1}} = \mathbf{0}$
- $\overline{\mathbf{0}} = \mathbf{1}$
- And, just like how we had the useful laws for equivalences involving connectives, we have the following Fundamental Laws and Theorems of Boolean Algebra

1. $X + 0 = X$
2. $X + 1 = 1$
3. $X + X = X$
4. $X + \bar{X} = 1$
5. $X \cdot 0 = 0$
6. $X \cdot 1 = X$
7. $X \cdot X = X$
8. $X \cdot \bar{X} = 0$
9. $\bar{\bar{X}} = X$
10. $X + Y = Y + X$
11. $XY = YX$
12. $(X + Y) + Z = X + (Y + Z)$
13. $(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$
14. $X(Y + Z) = XY + XZ$
15. $X + Y \cdot Z = (X + Y) \cdot (X + Z)$

16. $\overline{X+Y} = \bar{X} \cdot \bar{Y}$

17. $\overline{X \cdot Y} = \bar{X} + \bar{Y}$

- **Creating Karnaugh Maps (K-Maps)**
  - To create a Karnaugh map:
    - Draw a grid, with one square for each row in the truth table.
    - The possible values of A are written as column headings on the top.
    - The possible values of B are written as row levels along the LHS.
  - To derive the simplist possible Boolean expression from the K-Map, we are looking for the largest possible groupings of ones.
  - When there are 3 variables, it is often best practice to:
    - place one variable and its possible values as column headings, and
    - the possible combinations of values for the remaining variables as row levels.

CONTOUREDUCATION

Website: contoureducation.com.au | **Phone:** 1800 888 300 | **Email:** hello@contoureducation.com.au

VCE Specialist Mathematics ½

Free 1-on-1 Consults

What are 1-on-1 Consults?

- **Who Runs Them?** Experienced Contour tutors (45 + raw scores and 99 + ATARs).
- **Who Can Join?** Fully enrolled Contour students.
- **When Are They?** 30 –minute 1-on-1 help sessions, after-school weekdays, and all-day weekends.
- **What To Do?** Join on time, ask questions, re-learn concepts, or extend yourself!
- **Price?** Completely free!
- **One Active Booking Per Subject:** Must attend your current consultation before scheduling the next. :)

SAVE THE LINK, AND MAKE THE MOST OF THIS (FREE) SERVICE!

Booking Link

bit.ly/contour-specialist-consult-2025