

Exercise walk-through

15.2

Boolean Algebra and Logic Circuits

A-Level Computer Science

Laws you must be able to use

Boolean algebra simplifies expressions. + is OR, · is AND, an overbar is

	Law	Form
NOT.	identity	$A + 0 = A, \quad A \cdot 1 = A$
	null	$A + 1 = 1, \quad A \cdot 0 = 0$
	idempotent	$A + A = A$
	inverse	$A + \overline{A} = 1, \quad A \cdot \overline{A} = 0$
	De Morgan	$(A + B)' = A' \cdot B', \quad (A \cdot B)' = A' + B'$
	absorption	$A + AB = A$

Example: $Z = AB + A\overline{B} = A(B + \overline{B}) = A$. Fewer terms $\Rightarrow$ fewer gates.

Boolean algebra, and where it runs

Truth tables

Truth tables list every input combination with its output – the definitive way to prove two expressions equivalent.

De Morgan's laws

De Morgan's laws : $\overline{A \cdot B} = \overline{A} + \overline{B}$ and $\overline{A + B} = \overline{A} \cdot \overline{B}$ – break the bar and swap the operator.

The ALU

The **ALU** is where these gates end up: it performs the arithmetic and logic the instruction set exposes.

Just-in-time compilation

Just-in-time compilation compiles bytecode to native code *while* the program runs, so hot paths get compiled and the rest is interpreted.

Simplify with the laws, then check with a truth table. The check costs four rows and catches every algebra slip.

Karnaugh maps – group the 1s

A **Karnaugh map** simplifies by grouping adjacent 1s.

- rows and columns in **Gray code** (00, 01, 11, 10)
- groups are rectangles whose sides are 1, 2, 4, 8
- the map **wraps around** its edges
- a group of 2 drops one variable; of 4 drops two

1s in $\bar{A}B$ and AB

The two 1s share the $B = 1$ column. A changes, so it disappears:

$$X = B$$

Two gates fewer than $\bar{A}B + AB$.

Exercises 2.1-2.3

2.1 Simplify $A + AB$.

[1]

2.2 Use **De Morgan's law** to rewrite $\overline{A \cdot B}$.

[1]

2.3 State the **sum** and **carry** outputs of a half adder as Boolean expressions.

[2]

Exercises 2.1-2.3

2.1 Simplify $A + AB$. [1]

2.1

- $A + AB = A$ (absorption law)

2.2 Use **De Morgan's law** to rewrite $\overline{A \cdot B}$. [1]

2.2

- $\overline{A \cdot B} = \overline{A} + \overline{B}$

2.3 State the **sum** and **carry** outputs of a half adder as Boolean expressions. [2]

2.3

- $S = A \oplus B$ (A XOR B)
- $C = A \cdot B$ (A AND B)

Exercises 2.4·3.2

2.4 State what a **flip-flop** stores.

[1]

3.1 Simplify $Z = AB + A\overline{B}$ using Boolean algebra. Show your steps.

[2]

3.2 A K-map has four 1s forming a group where $A = 0$ and $C = 0$ throughout. Write the simplified term, and state how many variables a **group of 4** removes.

[2]

Exercises 2.4·3.2

2.4 State what a **flip-flop** stores.

[1]

2.4

- **one bit** (it is bistable – it remembers its state)

3.1 Simplify $Z = AB + A\overline{B}$ using Boolean algebra. Show your steps.

[2]

3.1

- $Z = AB + A\overline{B} = A(B + \overline{B})$
- $= A \cdot 1 = A$

3.2 A K-map has four 1s forming a group where $A = 0$ and $C = 0$ throughout. Write the simplified term, and state how many variables a **group of 4** removes.

[2]

3.2

- term = $\overline{A}\overline{C}$
- a group of 4 removes **2** variables

Exercise 3.3

3.3 Complete the truth table for a **half adder**.

[3]

A	B	S	C
0	0		
0	1		
1	0		
1	1		

Exercise 3.3

3.3 Complete the truth table for a **half adder**.

[3]

A	B	S	C
0	0		
0	1		
1	0		
1	1		

3.3

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- S column = $A \text{ XOR } B$; C column = $A \text{ AND } B$; all four rows correct

Exercises 3.4-3.5

3.4 State **one** difference between an **SR** flip-flop and a **JK** flip-flop. [2]

3.5 A two-input logic circuit gives output 1 only on the rows $A=0, B=1$ and $A=1, B=1$. Write the Boolean expression as a **sum-of-products**, then **simplify** it. [3]

Exercises 3.4-3.5

3.4 State **one** difference between an **SR** flip-flop and a **JK** flip-flop. [2]

3.4

- in an SR flip-flop the input $S=1$, $R=1$ is **invalid**
- a JK flip-flop uses $J=1$, $K=1$ to **toggle** (flip) the output – so JK suits counters

3.5 A two-input logic circuit gives output 1 only on the rows $A=0$, $B=1$ and $A=1$, $B=1$. Write the Boolean expression as a **sum-of-products**, then **simplify** it. [3]

3.5

- SOP: $Z = \overline{A} \cdot B + A \cdot B$
- $= B(\overline{A} + A) = B \cdot 1 = B$

Exercise 3.6 (a)

3.6 A logic circuit has three inputs A , B and C , and one output X , given by

$$X = \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

(a) **Complete** the truth table.

[4]

A	B	C	X
0	0	0	_____
0	0	1	_____
0	1	0	_____
0	1	1	_____
1	0	0	_____
1	0	1	_____
1	1	0	_____
1	1	1	_____

Exercise 3.6 (a)

3.6 A logic circuit has three inputs A , B and C , and one output X , given by

$$X = \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

(a) **Complete** the truth table.

[4]

A	B	C	X
0	0	0	_____
0	0	1	_____
0	1	0	_____
0	1	1	_____
1	0	0	_____
1	0	1	_____
1	1	0	_____
1	1	1	_____

(a) $X = 0, 0, 1, 1, 0, 1, 0, 1$ for the eight rows in the order given

Exercise 3.6 (b)

3.6 A logic circuit has three inputs A , B and C , and one output X , given by

$$X = \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

(b) **Complete** the Karnaugh map for X .

[2]

$AB \backslash C$	0	1
00	_____	_____
01	_____	_____
11	_____	_____
10	_____	_____

Exercise 3.6 (b)

3.6 A logic circuit has three inputs A , B and C , and one output X , given by

$$X = \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

(b) **Complete** the Karnaugh map for X .

[2]

$AB \backslash C$	0	1
00	_____	_____
01	_____	_____
11	_____	_____
10	_____	_____

(b) reading the same values into the map: row 00 gives 0,0; row 01 gives 1,1; row 11 gives 0,1; row 10 gives 0,1

Exercise 3.6 (c)

3.6 A logic circuit has three inputs A , B and C , and one output X , given by

$$X = \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

(c) **Draw** loops around the optimal groups on your map, and **write** the simplified sum-of-products expression. [4]

Exercise 3.6 (c)

3.6 A logic circuit has three inputs A , B and C , and one output X , given by

$$X = \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

(c) **Draw** loops around the optimal groups on your map, and **write** the simplified sum-of-products expression. [4]

(c) two groups: the whole $AB = 01$ row, which is $\bar{A}B$

- and the vertical pair in the $C = 1$ column covering $AB = 11$ and $AB = 10$, which is AC
- so $X = \bar{A}B + AC$

Exercise 3.6 (d)

3.6 A logic circuit has three inputs A , B and C , and one output X , given by

$$X = \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

(d) **Explain** why the row order of the map is 00, 01, 11, 10 and not 00, 01, 10, 11. [2]

Exercise 3.6 (d)

3.6 A logic circuit has three inputs A , B and C , and one output X , given by

$$X = \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

- (d) **Explain** why the row order of the map is 00, 01, 11, 10 and not 00, 01, 10, 11. [2]
- (d) consecutive rows must differ in **exactly one variable** – that is Gray-code order
- it is what makes an adjacent pair of 1s correspond to a term with one variable eliminated, so grouping works at all

Exercise 3.6 (e)

3.6 A logic circuit has three inputs A , B and C , and one output X , given by

$$X = \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

(e) **State** how many two-input gates the simplified expression needs, compared with the original. [2]

Exercise 3.6 (e)

3.6 A logic circuit has three inputs A , B and C , and one output X , given by

$$X = \bar{A} \cdot B + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

(e) **State** how many two-input gates the simplified expression needs, compared with the original. [2]

(e) simplified: one NOT, two ANDs and one OR – **four** gates

- the original has three AND terms, two of them three-input, plus two NOTs and two ORs: at least **eight** two-input gates