Exercise walk-through

Boolean Algebra and Logic Circuits ▪ 15.2

eXercise

You must be able to

- simplify **Boolean** expressions (De Morgan, absorption)
- use a **Karnaugh map** to simplify a truth table
- give the truth tables of a **half/full adder** and describe **SR/JK flip-flops**

Method — Boolean algebra

$+$ = OR, $\cdot$ = AND, overbar = NOT. Useful laws:

- **De Morgan:** $\overline{A+B} = \overline{A} \cdot \overline{B}$ and $\overline{A \cdot B} = \overline{A} + \overline{B}$
- **absorption:** $A + AB = A$
- $A + \overline{A} = 1$, so $Z = AB + A\overline{B} = A(B + \overline{B}) = A$.

Fewer terms $\rightarrow$ **fewer gates**.

Method — Sum-of-products from a truth table

For **each row** where the output is 1, AND the inputs (a variable as itself if it is 1, or NOT it if 0); then OR these product terms together. If the 1s are at rows $A\overline{B}C$ and ABC:

$$Z = A\overline{B}C + ABC = AC(\overline{B} + B) = AC.$$

This is the **sum-of-products (SOP)** form; simplify it with Boolean algebra or a Karnaugh map.

Method — Karnaugh map

Place each truth-table 1 in the map (Gray-code order), then **group adjacent 1s** in blocks of 1, 2, 4 or 8. A larger group = a simpler term — variables that **change** within the group disappear:
A group of 2 drops 1 variable; a group of 4 drops **2**; a group of 8 drops 3.

Method — Karnaugh map

CD \ AB	**00**	**01**	**11**	**10**
00	1	1	0	0
01	1	1	0	0
11	0	0	0	0
10	0	0	0	0

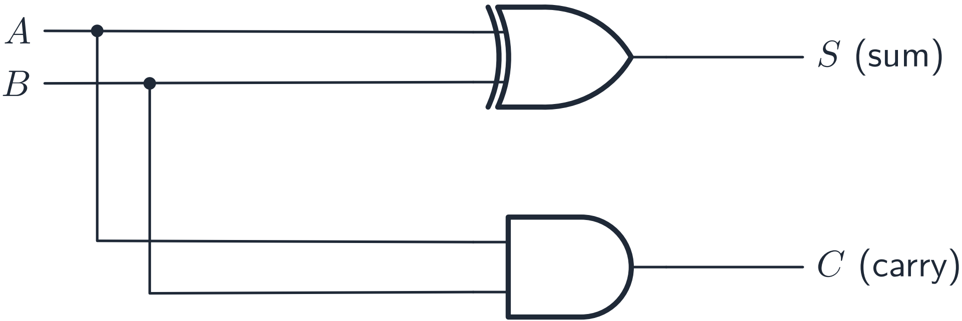
group of 4

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

Method — Half adder and flip-flops

A **half adder** gives $S = A \oplus B$ and $C = A \cdot B$:
A **full adder** also adds a carry-in. A **flip-flop** stores **one bit**: an **SR** flip-flop sets/resets/holds (`S=1,R=1` is invalid); a **JK** uses `1,1` to **toggle**, so it suits counters.

Method — Half adder and flip-flops



Practice 2.1 ▪ 1 mark

Simplify $A + AB$.

Solution 2.1

Worked steps

$A + AB = A$ (absorption law) **B1**.

Practice 2.2 ▪ 1 mark

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

Solution 2.2

Method: Boolean algebra

Worked steps

$\overline{A \cdot B} = \overline{A} + \overline{B}$ **B1**.

Practice 2.3 ▪ 2 marks

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

Solution 2.3

Method: Half adder and flip-flops

Worked steps

$S = A \oplus B$ (A XOR B); $C = A \cdot B$ (A AND B) **B1** **B1**.

Practice 2.4 ▪ 1 mark

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

Solution 2.4

Method: Half adder and flip-flops

Worked steps

One bit (it is bistable — it remembers its state) B1.

Exam-style 3.1 ▪ 2 marks

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

Solution 3.1

Method: Sum-of-products from a truth table

Worked steps

$Z = AB + A\overline{B} = A(B + \overline{B})$ **M1** *factor out* $A = A \cdot 1 = A$ **A1** *using* $B + \overline{B} = 1$.

Exam-style 3.2 ▪ 2 marks

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.

Solution 3.2

Method: Karnaugh map

Worked steps

Term $= \overline{A}\,\overline{C}$ **B1**; a group of 4 removes **2** variables **B1**.

Exam-style 3.3 ▪ 3 marks

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

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

Solution 3.3

Worked steps

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

Solution 3.3

B1 **B1** **B1** *the ‘S‘ column = A XOR B, the ‘C‘ column = A AND B, all four rows correct*

Exam-style 3.4 ▪ 2 marks

State **one** difference between an **SR** flip-flop and a **JK** flip-flop.

Solution 3.4

Method: Half adder and flip-flops

Worked steps

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 **B1** **B1**.

Exam-style 3.5 ▪ 3 marks

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.

Solution 3.5

Method: Sum-of-products from a truth table

Worked steps

SOP: $Z = \overline{A} \cdot B + A \cdot B$ **M1**; $= B(\overline{A} + A) = B \cdot 1 = B$ **M1** **A1** *factor out B, simplified to* $Z = B$.

Exam-style 3.6 ▪ 4 marks

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.

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	______

Exam-style 3.6 ▪ 4 marks

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

$AB\backslash C$	**0**	**1**
00	________	________
01	________	________
11	________	________
10	________	________

Exam-style 3.6 ▪ 4 marks

(c) **Draw** loops around the optimal groups on your map, and **write** the simplified sum-of-products expression.
(d) **Explain** why the row order of the map is 00, 01, 11, 10 and not 00, 01, 10, 11.
(e) **State** how many two-input gates the simplified expression needs, compared with the original.

Solution 3.6

Method: Sum-of-products from a truth table

Worked steps

(a) $X = 0, 0, 1, 1, 0, 1, 0, 1$ for the eight rows in the order given **B1** *for each pair of correct rows, to a maximum of 4*.

(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 $0, 1$ **B1** **B1**.

(c) Two groups: the whole $AB = 01$ row, which is $\bar{A}B$ **B1**; and the vertical pair in the $C = 1$ column covering $AB = 11$ and $AB = 10$, which is AC **B1**. So $X = \bar{A}B + AC$ **B1** **B1**.

(d) Consecutive rows must differ in **exactly one variable** — that is Gray-code order

Solution 3.6

B1; it is what makes an adjacent pair of 1s correspond to a term with one variable eliminated, so grouping works at all. With 01 next to 10 two variables change and the grouping would be wrong **B1**.

(e) Simplified: one NOT, two ANDs and one OR — **four** gates **B1**. The original has three AND terms, two of them three-input, plus two NOTs and two ORs: at least **eight** two-input gates **B1**.