Solutions

Each answer shows the steps, then the **result**.

2.1

- it has a **third input**, the carry in C_{in}, so it can add a column that receives a carry from the column to its right
- it is built from **two half adders and an OR gate**

2.2

- Sum $= A \oplus B \oplus C_{\text{in}}$
- $C_{\text{out}} = A \cdot B + C_{\text{in}} \cdot (A \oplus B)$ – any equivalent form is accepted

2.3

- De Morgan on the whole bar: $\overline{\overline{A}} \cdot \overline{B}$
- double negation gives $\mathbf{A} \cdot \overline{\mathbf{B}}$

2.4

- the output **toggles**: it changes to the opposite of its previous value

2.5

- in Gray code, **adjacent cells differ in exactly one variable**
- so a group of adjacent 1s has that one variable taking both values, which lets it be cancelled; in binary counting order adjacent cells could differ in two variables and grouping them would be invalid

3.1

- Sum column correct: 0, 1, 1, 0, 1, 0, 0, 1
- C_{out} column correct: 0, 0, 0, 1, 0, 1, 1, 1

A	B	C_{in}	**Sum**	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- the Sum is 1 on the rows with an **odd** number of 1s; the carry out is 1 whenever at least two inputs are 1

3.2

(a) two **NOR** gates

- **cross-coupled**: the output of each is an input to the other
- S and R labelled as the remaining input of one gate each, and the outputs labelled Q and $\overline{Q}$

(b) $S = 0$, $R = 0$: Q **holds** its previous value

- $S = 0$, $R = 1$: $Q = 0$, reset; $S = 1$, $R = 0$: $Q = 1$, set
- $S = 1$, $R = 1$ is the **invalid** row
- because both outputs would be forced to 0, so Q and $\overline{Q}$ would no longer be complements of each other, and what happens next depends on which input changes first

3.3

- the outputs are **fed back** as inputs, so with both inputs at 0 the circuit holds whatever state it was last put into
- that state persists until it is deliberately set or reset, so one flip-flop stores **one bit**; a register is a row of them

3.4

(a) the four cells are the **corners** of the map, and a K-map **wraps round**, so they form one valid group of four

- across all four, $B = 0$ and $D = 0$; A and C each take both values and cancel
- $\mathbf{F} = \overline{\mathbf{B}} \cdot \overline{\mathbf{D}}$
- four separate groups of one, giving four ANDed terms, earns the first mark only: the question is about grouping

(b) any two: it gives the simplified expression **directly**, without a chain of algebra

- the simplified circuit needs **fewer gates**, so it is cheaper, smaller and faster; and grouping is visual, so it is less error-prone than algebraic manipulation

3.5

- $A + \overline{A}B$: apply the **distributive** law, $A + \overline{A}B = (A + \overline{A})(A + B)$
- $A + \overline{A} = 1$ – the **complement** law
- $1 \cdot (A + B) = \mathbf{A} + \mathbf{B}$ – the identity law
- quoting the absorption result $A + \overline{A}B = A + B$ directly earns the answer mark but not the working marks, and the question says to show each step