UNIVERSITY COMPUTER SCIENCE • EXERCISE SHEET

# 3.2 Logic gates and digital building blocks

Name: _____ Class: _____ Date: _____ **Total: 24 marks**

**KEY WORDS** half adder • multiplexer

## Objective

Build the blocks; minimise with algebra.

## 1 Key knowledge and worked examples

### ■ The blocks

Half adder: sum $= a \oplus b$, carry $= ab$; full adder chains two halves + OR; $n$ full adders make a ripple-carry adder ✓. Mux: out $= \bar{s}a + sb$.

### ■ Worked example: identities at work

$x + x = x$, $x \cdot x = x$, $x + \bar{x} = 1$: expressions with one truth table collapse into one circuit – fewer gates, less delay ✓.

## 2 Practice

**2.1** Give the sum and carry of $1 + 1$ in a half adder. [3]

**2.2** How many 2-to-1 muxes make a 4-to-1? [3]

**2.3** Simplify $x \cdot \bar{x}$. [3]

## 3 Exam-style questions

**3.1** Design a 2-bit ripple-carry adder from half and full adders.

Name the pieces. Write each output as a boolean expression. Trace $11_2 + 01_2$ through your circuit and give the sum and the final carry. [8]

**3.2** Two circuits are claimed to implement the same 2-to-1 mux: $f = \bar{s}a + sb$ and $g = \bar{s}a + sb + s\bar{s}\bar{a}$.

Simplify $g$, showing every step, and prove algebraically whether the claim holds. Build the joint truth table for both outputs and confirm they match row by row. State which circuit you would build and why. [7]

## Solutions

**2.1**

- Sum 0, carry 1 (binary 10) ✓.

**2.2**

- Three ✓.

**2.3**

- 0 – always false ✓.

**3.1**

- Pieces: half adder (bit 0) + full adder (bit 1) ✓.
- $s_0 = a_0 \oplus b_0$; $c_0 = a_0 b_0$ ✓.
- $s_1 = a_1 \oplus b_1 \oplus c_0$; $c_1 = a_1 b_1 + c_0(a_1 \oplus b_1)$ ✓.
- Trace: $a = 11$, $b = 01$: $s_0 = 1 \oplus 1 = 0$, $c_0 = 1$; $s_1 = 1 \oplus 0 \oplus 1 = 0$, $c_1 = 0 + 1 = 1 \to 100$ ✓.

**3.2**

- $g = \bar{s}a + sb + \bar{s}\bar{a}s$; the last term has $s\bar{s} = 0$ ✓.
- So $g = \bar{s}a + sb = f$ – one identity collapses it ✓.
- Joint table: $s$=0 → both output $a$; $s$=1 → both output $b$ ✓.
- Build $f$: fewer gates, less delay – the identity is the optimiser ✓.