

3.3 Combinational building blocks

Name: _____ Class: _____ Date: _____

Total: 21 marks

KEY WORDS multiplexer • decoder • full adder • hazard

Objective

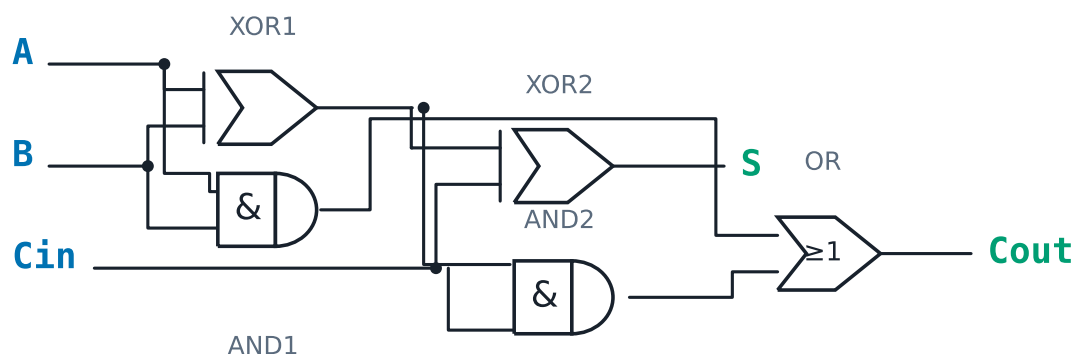
Design with multiplexers, decoders and adders; read a select code to a data line; explain hazards and why redundant grouping covers them – every block answer verified by evaluating all input rows.

1 Key knowledge and worked examples

■ Three blocks cover most designs

- **Multiplexer** (MUX): 2^n data inputs, n select lines; $Y = D_i$ where i is the select code's value. A 4-to-1 MUX with $S_1S_0 = 10$ outputs D_2 .
- **Decoder**: n inputs, 2^n outputs, exactly one high – output i is the minterm i . Any SOP function = OR of decoder outputs (plus NOT for the variable complements is unnecessary – the decoder generates all minterms).
- **Full adder**: $S = A \oplus B \oplus C_{in}$, $C_{out} = AB + C_{in} \cdot (A \oplus B)$. Ripple n of them for an n -bit adder.

The full adder: five gates, three inputs, two outputs



■ Worked example: full adder, all eight rows

At $A = 1, B = 1, Cin = 1$: $S = 1 \oplus 1 \oplus 1 = 1$; $Cout = 1 \cdot 1 + 1 \cdot (1 \oplus 1) = 1$. Three ones in $\rightarrow$ binary 11 out: $2 \cdot 1 + 1 = 3 \checkmark$. The invariant $2 \cdot Cout + S = A + B + Cin$ holds on all eight rows – that is the exhaustive verification, and it is how the HDL testbench of unit 3.7 checks the same block.

■ Hazards: unequal paths glitch

A **hazard** is a momentary false output when one input change reaches the output through paths of different delays. The classic: $F = AB + A'C$ – when $B = C = 1$ and A changes $1 \rightarrow 0$, both terms can be momentarily 0. The remedy is the redundant **consensus** term BC (adding it changes no truth-table value but holds the output during the switch). Hazards matter because the glitch can downstream be captured as a clock edge – the reason synchronous designs sample on edges, not levels.

2 Practice

2.1 An 8-to-1 MUX has data inputs $D0..D7$ and selects $S2S1S0 = 101$. Which input reaches the output? What is $S2S1S0$ to select $D6$? [3]

2.2 A 3-to-8 decoder has inputs $A2A1A0 = 011$. Which output is high? Which minterm does it represent? [3]

2.3 Compute a full adder at $A = 0, B = 1, Cin = 1$: give S and $Cout$, and verify with the invariant. [3]

3 Exam-style questions

3.1 Design a 4-bit ripple adder for $A + B$ where $A = 1011$ and $B = 0110$.

Work each full adder stage (LSB first), giving S and C_{out} per stage, the final 4-bit sum and the carry out. Verify in decimal, and state the largest pair of 4-bit unsigned values that does NOT produce a carry out. [6]

3.2 A circuit must output 1 when a 3-bit unsigned input is greater than 5.

Realise F as an OR of 3-to-8 decoder outputs (name the minterms), then minimise on a Karnaugh map and give the minimal SOP. Verify by evaluating your expression on all 8 inputs, and state which single-variable don't-care condition could NOT be used here (and why there are no don't-cares at all). [6]

Solutions

2.1

- Select value $101_2 = 5 \rightarrow D5$ reaches the output.
- To select D6: $S2S1S0 = 110$.

2.2

- $011_2 = 3 \rightarrow$ output Y3 is high (all others low).
- It represents minterm 3: $A2' \cdot A1 \cdot A0$.

2.3

- $S = 0 \oplus 1 \oplus 1 = 0$.
- $Cout = 0 \cdot 1 + 1 \cdot (0 \oplus 1) = 1$.
- Invariant: $2 \cdot 1 + 0 = 2 = 0 + 1 + 1 \checkmark$.

3.1

- Stage 0: $1+0+0$: $S0=1$, $C1=0$.
- Stage 1: $1+1+0$: $S1=0$, $C2=1$.
- Stage 2: $0+1+1$: $S2=0$, $C3=1$.
- Stage 3: $1+0+1$: $S3=0$, $Cout=1$.
- Sum $S = 0001$, Cout 1 $\rightarrow$ the 5-bit result $10001 = 17 = 11 + 6 \checkmark$.
- No carry out requires $A + B \leq 15$; the largest pair is any with sum 15, e.g. $15 + 0$ or $1111 + 0000$ (equivalently $1000 + 0111$).

3.2

- Greater than 5 (of 0–7): minterms 6, 7 $\rightarrow F = Y6 + Y7$.
- Karnaugh map: cells 6 (110), 7 (111) adjacent $\rightarrow$ group = $A \cdot B$ (C eliminated).
- Minimal SOP: $F = AB$.
- Verify: $AB = 1$ exactly on inputs 6, 7 $\checkmark$ (all eight rows).
- No don't-cares exist because every 3-bit pattern is a real, specified input – the function's value matters on all 8 rows; don't-cares require impossible or unspecified combinations, and none are.