Version OBA/2

EGT2
ENGINEERING TRIPOS PART IIA

Friday 1 May 2026 9.30 to 11.10

Module 3B2

INTEGRATED DIGITAL ELECTRONICS

*Answer not more than **three** questions.*

All questions carry the same number of marks.

*The **approximate** percentage of marks allocated to each part of a question is indicated in the right margin.*

*Write your candidate number **not** your name on the cover sheet.*

STATIONERY REQUIREMENTS

Single-sided script paper

SPECIAL REQUIREMENTS TO BE SUPPLIED FOR THIS EXAM

CUED approved calculator allowed

Engineering Data Book

10 minutes reading time is allowed for this paper at the start of the exam.

You may not start to read the questions printed on the subsequent pages of this question paper until instructed to do so.

You may not remove any stationery from the Examination Room.

1 (a) Consider the Verilog code in Fig. 1.

(i) Derive the Boolean expression for the value of `q_out[k]` following a rising clock edge in terms of the inputs `ctrl[k]`, `data_in[k]`, and the current value of `q_out[k]`. [10%]

Solution:
The `if-else` logic inside the loop describes a 2-to-1 multiplexer behaviour where `ctrl[k]` acts as the select signal.

$$\texttt{q\_out[k]} = (\texttt{ctrl} \cdot \overline{\texttt{q\_out[k]}}) + (\overline{\texttt{ctrl}} \cdot \texttt{data\_in})$$

This matches the characteristic equation of a D bistable with a specific combinational input.

(ii) Rewrite the `else` branch (lines 13–18) using a single line of behavioural Verilog statement that utilises bitwise operators instead of the `for` loop and `if` statement. [10%]

Solution:
Since the control signal `ctrl` is a vector, we must use bitwise operators to apply the logic to each bit individually.
Logic Derivation:

$$\texttt{q\_out}^{+} = (\texttt{ctrl}) \cdot (\overline{\texttt{q\_out}}) + (\overline{\texttt{ctrl}}) \cdot (\texttt{data\_in})$$

Verilog Code:

```
q_out <= (ctrl  &  ~q_out)  |  (~ctrl  &  data_in);
```


(cont.

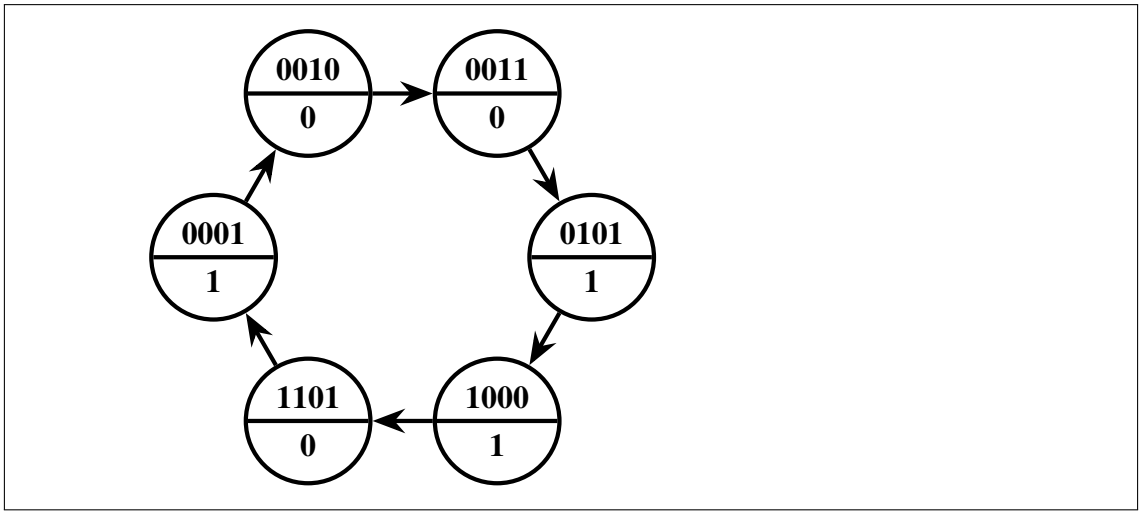
```
1   module multi_reg (
2       input            clk, rst,
3       input [3:0]      data_in,
4       input [3:0]      ctrl,
5       output reg [3:0] q_out
6   );
7   integer k;
8
9   always @(posedge clk or posedge rst) begin
10      if (rst)
11          q_out <= 4'b0000;
12      else begin
13          for (k = 0; k < 4; k = k + 1) begin
14              if (ctrl[k])
15                  q_out[k] <= ~q_out[k];
16              else
17                  q_out[k] <= data_in[k];
18          end
19      end
20  end
21  endmodule
```

Fig. 1

(b) Design a sequential circuit using T bistables that continuously cycles through a sequence of states representing a subset of Fibonacci numbers: {1, 2, 3, 5, 8, 13}. Each state is represented by its 4-bit binary numbers, e.g., 1=0001, 13=1101. The counter transitions to the next state in the sequence on each clock pulse, returning to 1 after reaching 13. The circuit has a single output, Z. $Z = 1$ when the current state is 1, 5, or 8. $Z = 0$ when the current state is 2, 3, or 13. Assume all unused states as "don't care" conditions.

(i) Draw the state diagram for this circuit. [5%]

Solution:



(ii) Build the state table for this circuit. [20%]

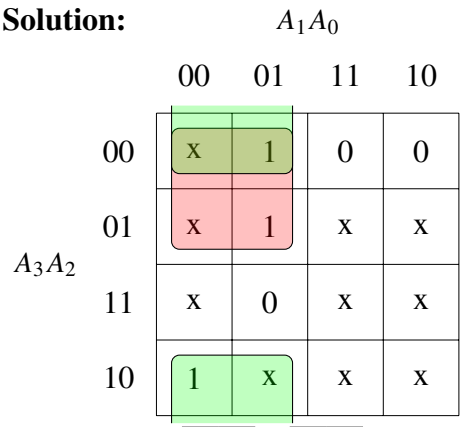
Solution:

Present State				Next State					FF Inputs			
A_3	A_2	A_1	A_0	A_3	A_2	A_1	A_0	z	T_{A_3}	T_{A_2}	T_{A_1}	T_{A_0}
0	0	0	0	x	x	x	x	x	x	x	x	x
0	0	0	1	0	0	1	0	1	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	1	0	0	1	1	0
0	1	0	0	x	x	x	x	x	x	x	x	x
0	1	0	1	1	0	0	0	1	1	1	0	1
0	1	1	0	x	x	x	x	x	x	x	x	x
0	1	1	1	x	x	x	x	x	x	x	x	x
1	0	0	0	1	1	0	1	1	0	1	0	1
1	0	0	1	x	x	x	x	x	x	x	x	x
1	0	1	0	x	x	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x	x	x
1	1	0	1	0	0	0	1	0	1	1	0	0
1	1	1	0	x	x	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x	x	x

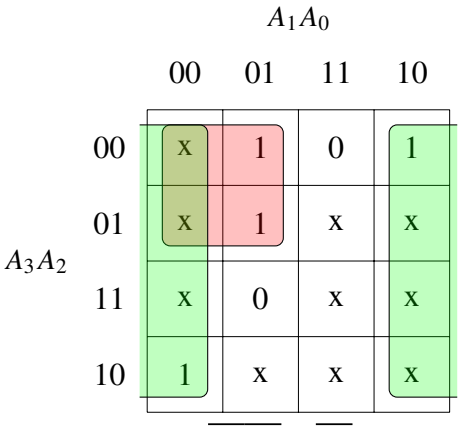
(iii) Derive the minimised input functions for the bistables and minimised output function for the circuit. [20%]

(cont.

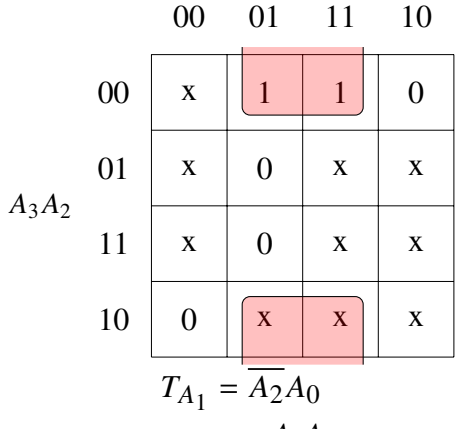
Solution:



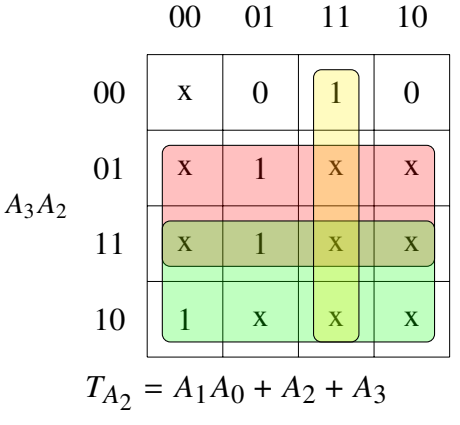
$z = \overline{A_3A_1} + \overline{A_2A_1}$



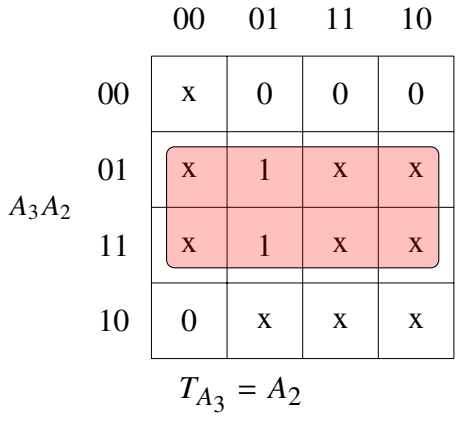
$T_{A_0} = \overline{A_3A_1} + \overline{A_0}$



$T_{A_1} = \overline{A_2}A_0$



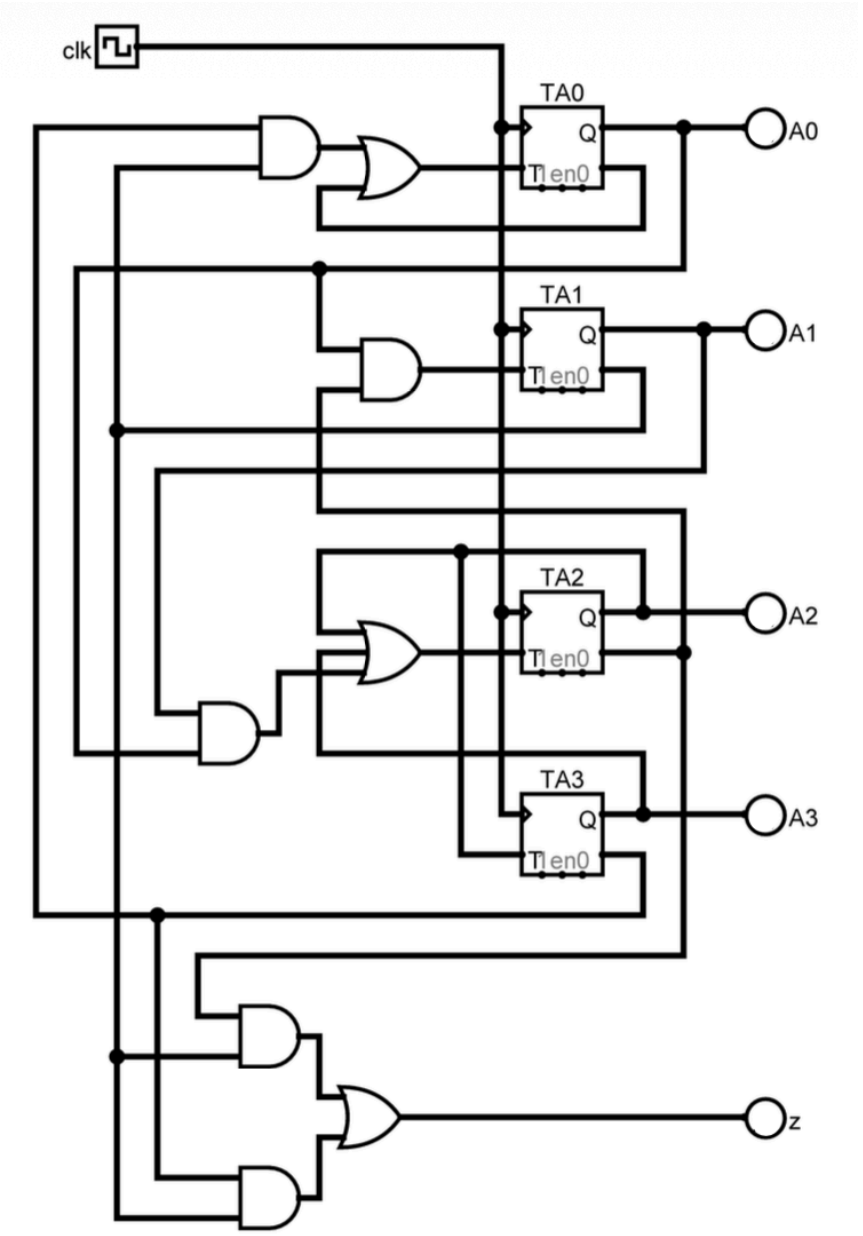
$T_{A_2} = A_1A_0 + A_2 + A_3$



$T_{A_3} = A_2$

(iv) Draw the circuit implementation for this counter using T bistables and logic gates. [20%]

Solution:



(c) Briefly discuss the advantages and disadvantages of implementing logic functions with ROMs, LUTs, and PLAs. [15%]

Solution:

•ROMs (Read-Only Memory):

(cont.

– *Advantage:* Simple design process (direct truth table mapping); guarantees implementation of any function of N inputs; predictable delay.

– *Disadvantage:* Size grows exponentially with the number of inputs (2^N). Inefficient for functions with many inputs but few product terms (sparse logic).

•LUTs (Look-Up Tables):

– *Advantage:* Highly flexible; can implement any function of k inputs (where k is small, typically 4-6); efficient for modern FPGA architectures; reprogrammable.

– *Disadvantage:* Propagation delay increases significantly if the function requires cascading many LUTs (i.e., when inputs $> k$).

•PLAs (Programmable Logic Arrays):

– *Advantage:* Efficient for implementing functions in Sum-of-Products form, especially those with many inputs but few product terms (sparse logic). Both AND and OR planes are programmable.

– *Disadvantage:* Slower than dedicated gates due to high fan-in/fan-out on the planes; interconnects can consume significant power.

2 (a) Consider the circuit in Fig. 2.

(i) Build the state table for this circuit. [10%]

Solution:

By analyzing the logic gate connections in the schematic, we derive the following equations for the Bistable inputs and the circuit output Z:

•Bistable 1 (Q_1):

$$J_1 = \overline{\overline{(X \cdot Q_2')} \cdot \overline{(X' \cdot Q_2)}} = (X \cdot Q_2') + (X' \cdot Q_2) = X \oplus Q_2$$
$$K_1 = \overline{\overline{(X' \cdot Q_2)}} = X' \cdot Q_2$$

•Bistable 2 (Q_2):

$$J_2 = \overline{Q_1 + X'} = Q_1' \cdot X$$
$$K_2 = J_2 = Q_1' \cdot X$$

•Output (Z):

$$Z = \overline{Q_1' + Q_2'} = Q_1 \cdot Q_2$$

Using the characteristic equation $Q^+ = J\overline{Q} + \overline{K}Q$, we calculate the next states.

Present State Q_1Q_2	**Input** X	**Bistable Inputs** J_1K_1	J_2K_2	**Next State** Q_1^+	Q_2^+	**Output** Z
0 0	0	0 0	0 0	0	0	0
0 0	1	1 0	1 1	1	1	0
0 1	0	1 1	0 0	1	1	0
0 1	1	0 0	1 1	0	0	0
1 0	0	0 0	0 0	1	0	0
1 0	1	1 0	0 0	1	0	0
1 1	0	1 1	0 0	0	1	1
1 1	1	0 0	0 0	1	1	1

(ii) Is the circuit a Mealy or Moore circuit? Justify your answer. [10%]

Solution:

This is a Moore circuit because the output Z ($Q_1 \cdot Q_2$) depends only on the

(cont.

current state and not directly on the input X.

(iii) Does the circuit have any unused states? Assume 00 is the initial state. [10%]

Solution:
The state **10** loops to itself regardless of the input and cannot be reached from states 00, 01, or 11. Therefore, 10 is an unused state (assuming the initial state is 00).

(iv) What is the output sequence for the input sequence $X = 01100$ when it is applied in the order of 0 (first), 1, 1, 0, 0 (last)? [10%]

Solution:
Given Input Sequence $X = 0, 1, 1, 0, 0$ and Initial State 00.

Clock Edge	**Current State**	**Input** (X)	**Next State**	**Output** (Z)
–	00	–	–	0
1	00	0	00	0
2	00	1	11	1
3	11	1	11	1
4	11	0	01	0
5	01	0	11	1

Reading the output Z values from the timing trace above (after each clock edge):

Output Sequence: 0, 1, 1, 0, 1

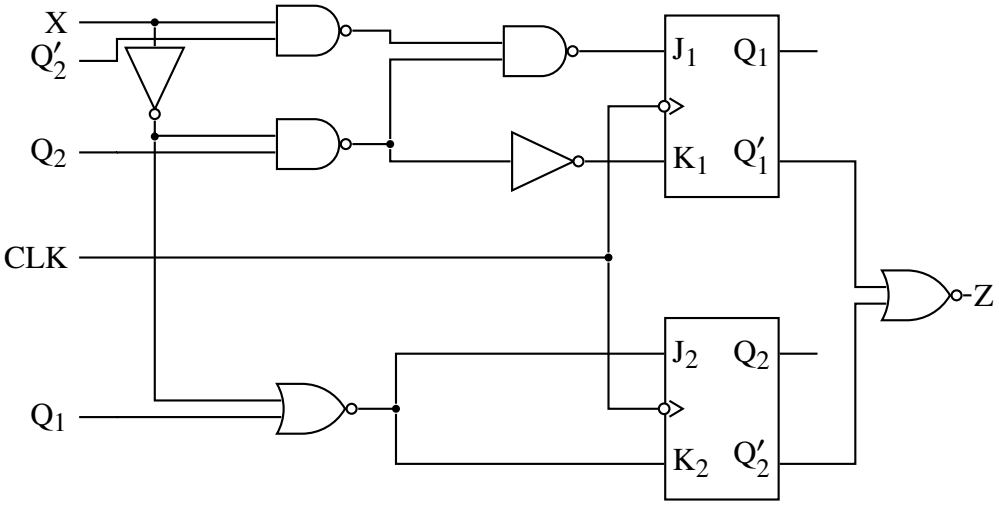


Fig. 2

(b) A 4-input LUT is often used as the basic building block of an FPGA.

(i) Show how a 4-input LUT can be implemented by combining two 3-input LUTs and a 2-to-1 multiplexer. [15%]

Solution:

A 4-input Look-Up Table (LUT) implementing a function $Y(A, B, C, D)$ can be constructed using two 3-input LUTs and a 2:1 multiplexer by applying the Shannon Expansion Theorem. We can decompose the function with respect to one variable, for instance, D:

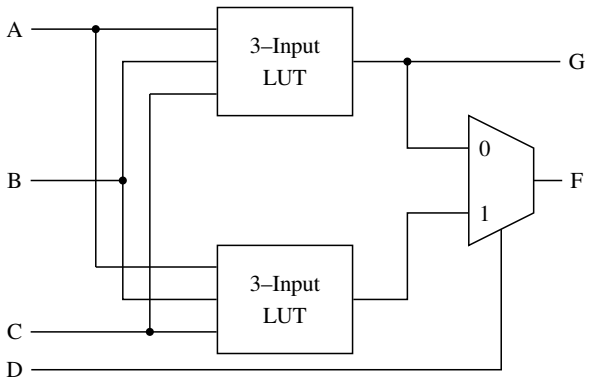
$$Y(A, B, C, D) = \bar{D} \cdot Y(A, B, C, 0) + D \cdot Y(A, B, C, 1)$$

Based on the structure shown in figure below:

- •The variable D is used as the select input for the 2:1 multiplexer.
- •The top 3-input LUT implements the function when $D = 0$. Its inputs are A, B, C, and its content is the truth table for the sub-function $G(A, B, C) = Y(A, B, C, 0)$. The output of this LUT is connected to input 0 of the multiplexer.
- •The bottom 3-input LUT implements the function when $D = 1$. Its inputs are A, B, C, and its content is the truth table for the sub-function $F(A, B, C) = Y(A, B, C, 1)$. The output of this LUT is connected to input 1 of the multiplexer.

Depending on the value of D, the multiplexer selects the appropriate sub-function to produce the correct 4-input output.

(cont.



(ii) Consider the implementation of a comparator circuit in Fig. 3. Determine the logic expressions for F_{01} and F_{23} so that the circuit sets $S_{A=B}$ to 1 when the two binary numbers, $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$, are equal. [15%]

Solution:

For a 4-bit comparator, the equality output $S_{A=B}$ is 1 only when all corresponding bits of A and B are equal. The logic equation for the output $S_{A=B}$ is written as:

$$S_{A=B} = \overline{(A_3 \oplus B_3)}\ \overline{(A_2 \oplus B_2)}\ \overline{(A_1 \oplus B_1)}\ \overline{(A_0 \oplus B_0)}$$

Because the circuit is cascaded such that:

$$S_{A=B} = F_{23} \cdot F_{01}$$

It follows that the lower-stage LUT output must be:

$$F_{01} = \overline{(A_1 \oplus B_1)}\ \overline{(A_0 \oplus B_0)}$$

And the upper-stage LUT output must be:

$$F_{23} = \overline{(A_3 \oplus B_3)}\ \overline{(A_2 \oplus B_2)}$$

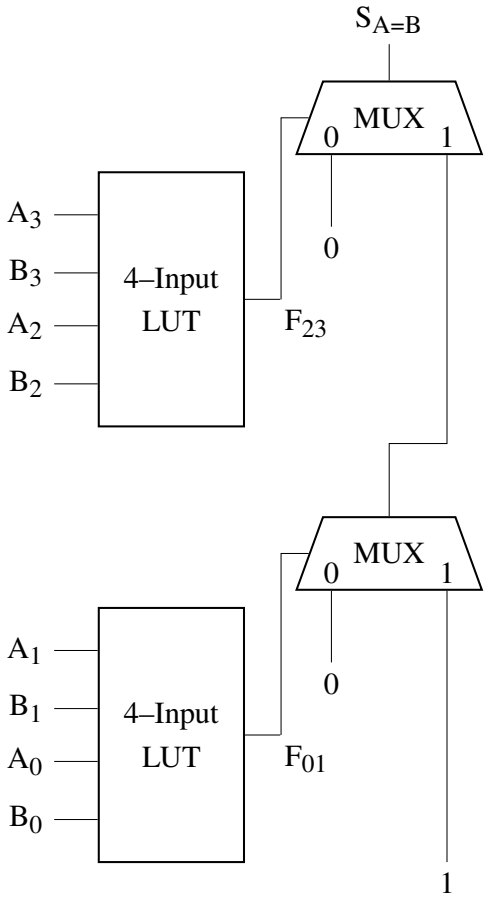


Fig. 3

(c) Design a Mealy-type synchronous sequential controller circuit for a vending machine that dispenses two different meals: A (for £ 1.50) and B (for £ 2.50). The machine has a selection switch that captures the user's choice, and has a single coin slot that accepts 50 p, £ 1, and £ 2, one coin at a time. If the accumulated credit equals or exceeds the price of the selected meal, the machine dispenses corresponding meal and resets. Assume that the machine does not give any change. Clearly state any other assumption that you make in your design.

(i) Build the state table for this circuit. [15%]

Solution:

(cont.

Current State	**Inputs**		**Next State**	**Outputs**	
(Credit Held)	Coin	Sel	(Q_{next})	Dispense A	Dispense B
S_0 (0p)	50p	X	S_1	0	0
S_0 (0p)	£1	X	S_2	0	0
S_0 (0p)	£2	A	S_0	1	0
S_0 (0p)	£2	B	S_4	0	0
S_1 (50p)	50p	X	S_2	0	0
S_1 (50p)	£1	A	S_0	1	0
S_1 (50p)	£1	B	S_3	0	0
S_1 (50p)	£2	A	S_0	1	0
S_1 (50p)	£2	B	S_0	0	1
S_2 (£1.00)	50p	A	S_0	1	0
S_2 (£1.00)	50p	B	S_3	0	0
S_2 (£1.00)	£1	A	S_0	1	0
S_2 (£1.00)	£1	B	S_4	0	0
S_2 (£1.00)	£2	A	S_0	1	0
S_2 (£1.00)	£2	B	S_0	0	1
S_3 (£1.50)	50p	A	S_0	1	0
S_3 (£1.50)	50p	B	S_4	0	0
S_3 (£1.50)	£1	A	S_0	1	0
S_3 (£1.50)	£1	B	S_0	0	1
S_3 (£1.50)	£2	A	S_0	1	0
S_3 (£1.50)	£2	B	S_0	0	1
S_4 (£2.00)	Any	A	S_0	1	0
S_4 (£2.00)	Any	B	S_0	0	1

Note: 'X' in the Selection column indicates the output/state is independent of the selection switch for that coin value.

(ii) Draw the circuit implementation for this controller using ROM. [15%]

Solution:

The large block represents the Read-Only Memory (ROM), which acts as a hardware "Truth Table." Unlike the PAL implementation, which uses logic gates to calculate the result, the ROM simply retrieves a pre-calculated value stored at

a specific memory address.

- **Address Inputs (7 bits):** The address pointer is formed by combining the *Current State* (Q_2, Q_1, Q_0) and the *External Inputs* (Sel, C_{50}, C_1, C_2). This combination tells the ROM exactly "where we are" and "what just happened."

- **Data Outputs (5 bits):** The content stored at that address is the instruction for the machine. It outputs the *Next State* (D_2, D_1, D_0) to be saved, and the *Mealy Outputs* (D_A, D_B) to control the dispenser.

The smaller block on the right labeled **State Reg** consists of D Bistables.

- The ROM is purely combinational, meaning its output changes instantly when the input changes.

- The Register's job is to **synchronise** the system. It captures the *Next State* (D) from the ROM and holds it stable as the *Current State* (Q) for the duration of one clock cycle.

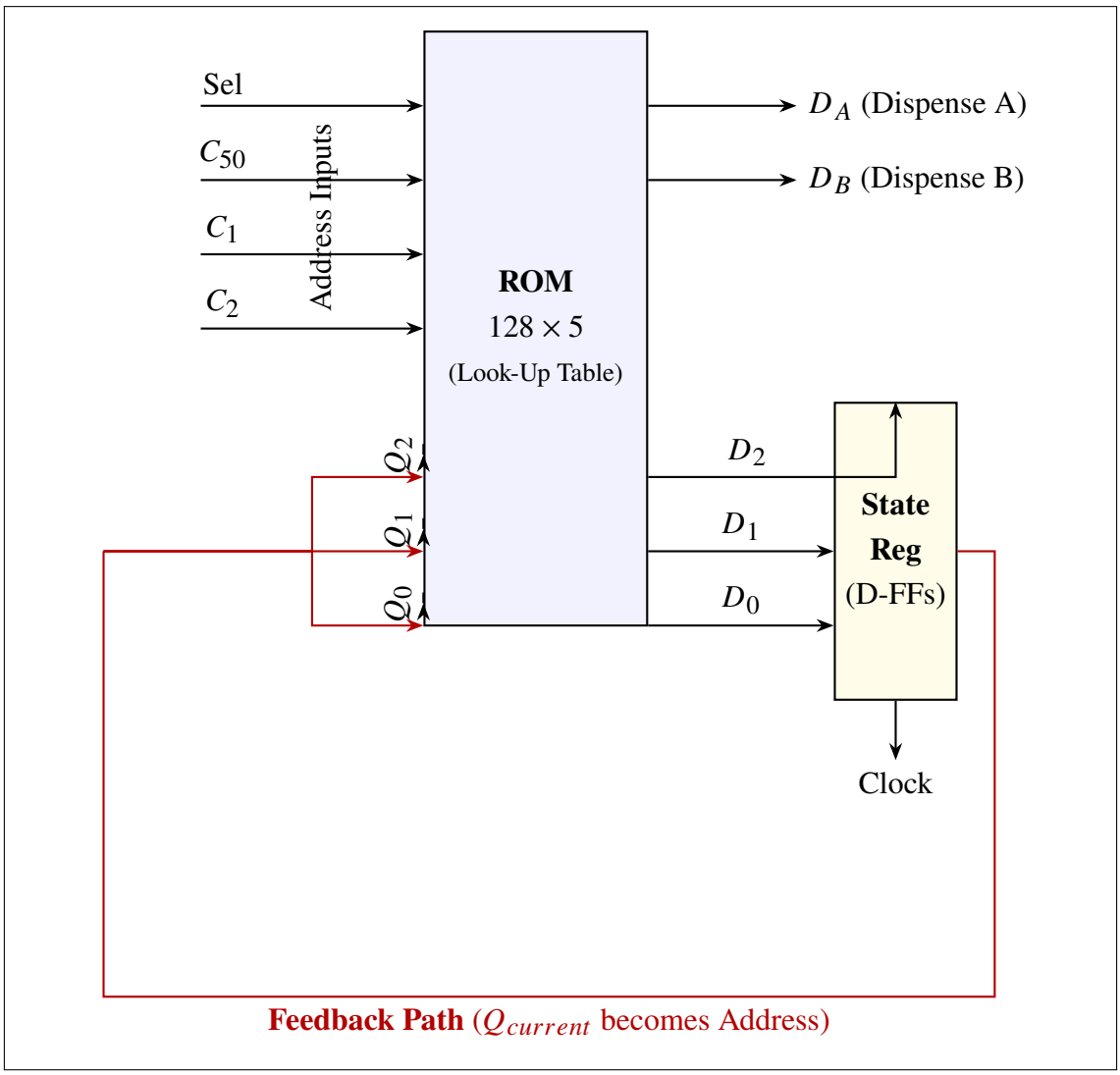
The red line in the diagram illustrates the feedback loop that characterises a Sequential Circuit.

- The *Current State* (Q) is an output of the Register.

- This value is fed back to the address input of the ROM.

- **Logic Flow:** The machine looks up the address for (Current State + Input) → The ROM outputs the Next State → The Register updates on the next clock edge → The cycle repeats.

(cont.


Sel
C_{50}
C_1
C_2
Address Inputs
ROM
128×5
(Look-Up Table)
D_A (Dispense A)
D_B (Dispense B)
Q_2
Q_1
Q_0
D_2
D_1
D_0
State
Reg
(D-FFs)
Clock
Feedback Path ($Q_{current}$ becomes Address)

3 (a) The voltage transfer characteristic of an inverter is shown in Fig. 4. The supply voltage V_{DD} is 5 V.

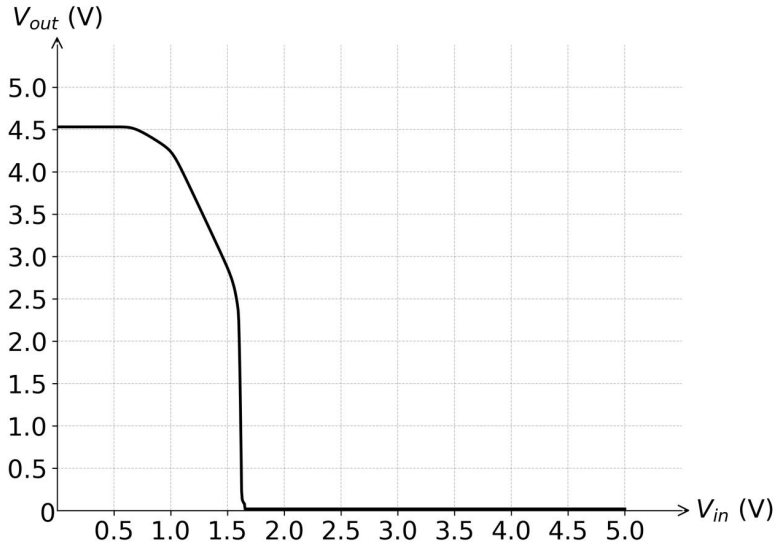


Fig. 4

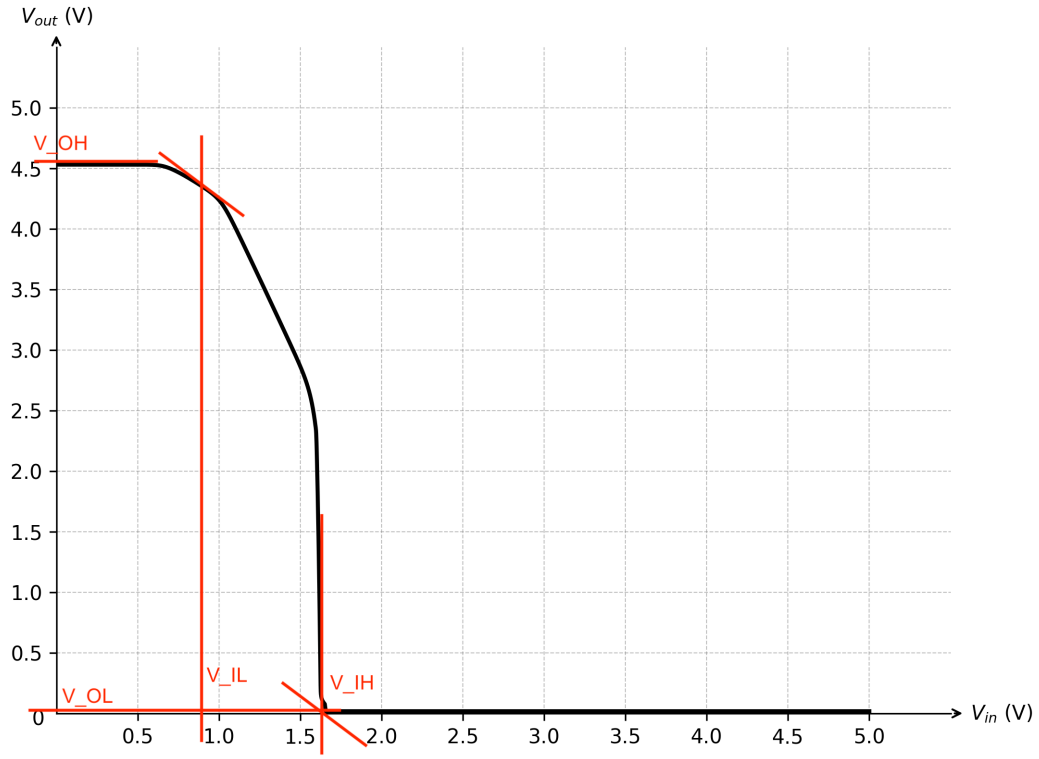
(i) Estimate the high and low noise margins graphically. [12%]

(ii) Explain whether it is likely that the inverter characterised by Fig. 4 is implemented by the CMOS technology or not. [8%]

Solution:

(i) From the annotated graph, $V_{IL} \approx 0.9$ V, $V_{IH} \approx 1.6$ v, $V_{OH} = 4.5$ V, $V_{OL} \approx 0$ V.
So, $NM_H = 4.5 - 1.6 = 2.9$, $NM_L = 0.9$

(cont.



(ii) It is not likely to be a CMOS inverter. This is because CMOS inverter gives a more balanced noise margins. The output high voltage would be as high as V_{DD} too.

(b) A CMOS inverter is constructed with an NMOS transistor and a PMOS transistor, as shown in Fig. 5. Its output is loaded with a capacitor C_L.

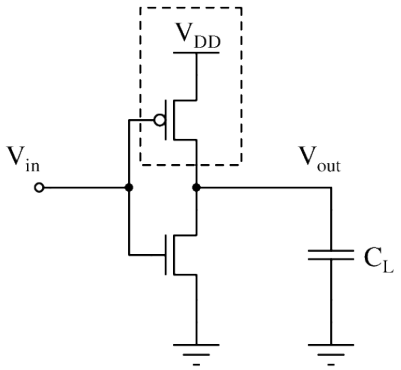


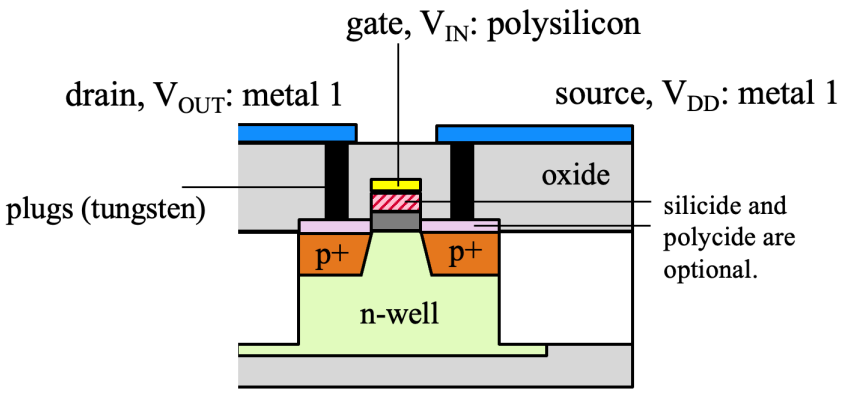
Fig. 5

(i) Sketch the cross-sectional device structure of the PMOS transistor that is inside the dashed region in Fig. 5. Assume the inverter was fabricated in the conventional

twin-well process. Based on your sketch, discuss the mode of operation of the transistor when input $V_{in} = V_{DD}/2$. State your assumption(s). [30%]

Solution:

The cross-sectional structure of the requested transistor, with labels:



The magnitude of the gate voltage $|V_{GSp}|$ of the PMOS transistor is $V_{DD}/2$. **Assume that** $V_{DD} \gg |V_{Tp}|$, the PMOS threshold voltage, then the transistor is in saturation or velocity saturation mode of operation.

(ii) Explain the effect of the ratio between the widths of the PMOS transistor and the NMOS transistor on the voltage transfer characteristics (VTC) of the inverter. [20%]

Solution:

The drain currents grow with the widths of the transistors. At the switching threshold V_M of the inverter, the magnitudes of the currents are the same. So, the ratios between the widths of the transistors affects the switching threshold V_M and therefore the noise margins (and robustness) of the inverter.

Wider, stronger PMOS shifts the V_M to the right while a wider, stronger NMOS shifts it to the left.

(iii) If C_L = 125 pF is initially charged to V_{DD} = 1.8 V, the width of the NMOS transistor W_n = 200 nm, the width of the PMOS transistor W_p = 300 nm and the lengths of both transistors $L_n = L_p$ = 180 nm, calculate the fall time t_f for the output. State your assumptions. [30%]

Solution:

(cont.

Assume that there is a step input (low to high) such that the PMOS is off and the NMOS is ON throughout the transient. The channel of the NMOS is short.

$$E_C L_n = (6 \times 10^6)(180 \times 10^{-9}) = 1.08\,\text{V}$$
$$V_{DSATn} = \frac{(1.8 - 0.5)(1.08)}{(1.8 - 0.5) + 1.08} = 0.59 < V_{DSn}$$

Hence, the NMOS is in velocity saturation. Assume the current is constant throughout, then:

$$I_{DSATn} = (200 \times 10^{-7})(8 \times 10^6)(1 \times 10^{-6})\frac{(1.8 - 0.5)^2}{(1.8 - 0.5) + 1.08} = 113.6\,\mu\text{A}$$
$$R_{eqn} = \frac{3}{4}\frac{V_{DD}}{I_{DSATn}} = 11.9\,\text{k}\Omega$$
$$t_f = \ln(9) R_{eqn} C_L$$
$$= (2.2)(11.9 \times 10^3)(125 \times 10^{-12}) = 3.27\,\mu\text{s}$$

Note: mindful with the conversion between cm and m

You may find the constants in Table 1 and equations below useful.

	NMOS	**PMOS**
Critical field (E_C)	6×10^4 V cm^{-1}	24×10^4 V cm^{-1}
Effective Mobility (μ_e)	270 cm^2 V^{-1} s^{-1}	70 cm^2 V^{-1} s^{-1}
Threshold Voltage (V_T)	0.5 V	−0.5 V
Oxide Capacitance (C_{ox})	1.0 μF cm^{-2}	
Carrier Saturation Velocity (υ_{sat})	8×10^6 cm s^{-1}	

Table 1

$$V_{DSAT} = \frac{(V_{GS} - V_T)E_C L}{(V_{GS} - V_T) + E_C L}$$

$$I_{DS} = \frac{W}{L}\frac{\mu_e C_{ox}}{\left(1 + \frac{V_{DS}}{E_C L}\right)}\left(V_{GS} - V_T - \frac{V_{DS}}{2}\right)V_{DS}, \qquad \text{for} \quad V_{DS} < V_{DSAT}$$

$$I_{DS} = W\upsilon_{sat}C_{ox}\frac{(V_{GS} - V_T)^2}{(V_{GS} - V_T) + E_C L}, \qquad \text{for} \quad V_{DS} \geq V_{DSAT}$$

$$R_{eq} = \frac{3}{4}\frac{V_{DD}}{I_{Dsat}}$$

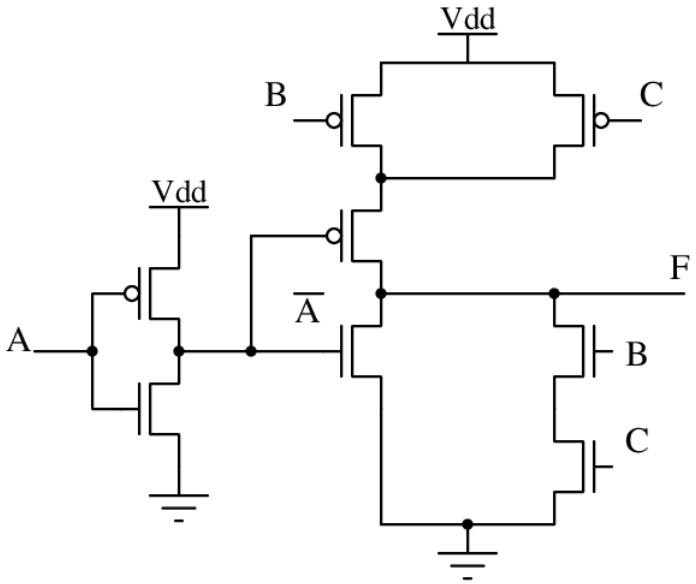
4 (a) You are given a Boolean function $F = A\overline{B} + A\overline{C}$.

(i) Design a CMOS gate for F with as few transistors as possible. Draw the transistor schematic. Use only positive input phases A, B and C. [20%]

(ii) Identify and explain a particular input combination that gives the longest propagation delay of your design in part (i). You may assume that all three inputs switch at the same time. [15%]

Solution:

(i) $\overline{F} = \overline{A} + BC$. Pull-down network (PDN) implements $\overline{F}$, while pull-up network (PUN) is a dual of the PDN. An inverter is needed to generate A



(ii) As A runs through an extra inverter in the gate design and that PMOS transistors are usally slower than the NMOS, the input combination $A = 1, B = 0, C = 1$ (or $A = 1, B = 1, C = 0$) gives the longest delays through a pull-down NMOS transistor (inverter, first level) and two pull-up PMOS transistors in series (PUN, second level).

(b) Analyse the CMOS implementation of a clocked J-K bistable in Fig. 6 and explain its working principle. Give a reason for the unpopularity of such design in modern integrated electronics. The excitation table for the J-K bistable is given in Table 2. [35%]

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

Table 2

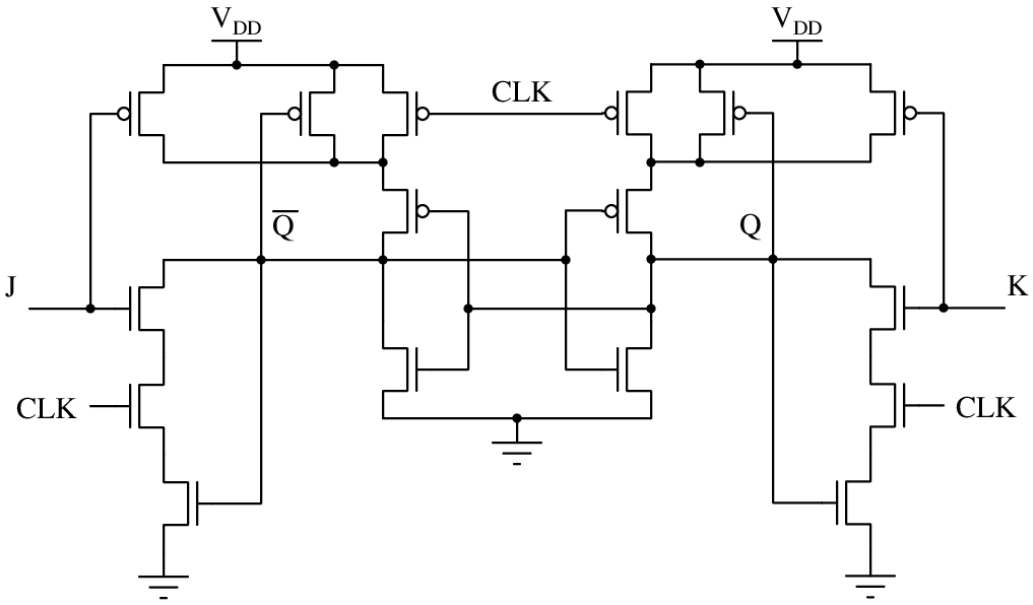


Fig. 6

Solution:

First note that when $CLK = 0$, a virtual supply is provided to the cross-coupled inverters that keeps the output values Q and $\overline{Q}$. For the following, take $CLK = 1$ so the output would change according to the characteristic table.

Case 1: $J = K = 0$ The virtual supply is continued through PMOS transistors connected to J and K. Q and $\overline{Q}$ could not be pulled down, so the cross-coupled inverters keep the outputs unchanged.

Case 2: $J = 0, K = 1$, assume $Q = 1$ (otherwise no change happens) The pull-down network is turned ON for Q, and a virtual supply is given to the inverter gated by Q and pulls up $\overline{Q}$ to 1. Once $Q = 0$ and $\overline{Q} = 1$, the cross-coupled inverters are back holding the static state.

Case 3: $J = 1, K = 0$. This is similar to case 2 but the role of J and K is reversed.

(cont.

Case 4: $J = K = 1$, assume $Q = 1$ and $\overline{Q} = 0$.

Then the pull-down network is turned ON for Q, and a virtual supply is given to inverter gated by Q. So $\overline{Q}$ starts to swing from 0 to 1. Note that the toggling will continue for a long enough high time of CLK.

The large number of transistors in the design requires large silicon area. This increases the fabricated cost so it is not popular in modern designs.

(c) Fig. 7 shows the typical implementation of the static random access memory (SRAM) cell.

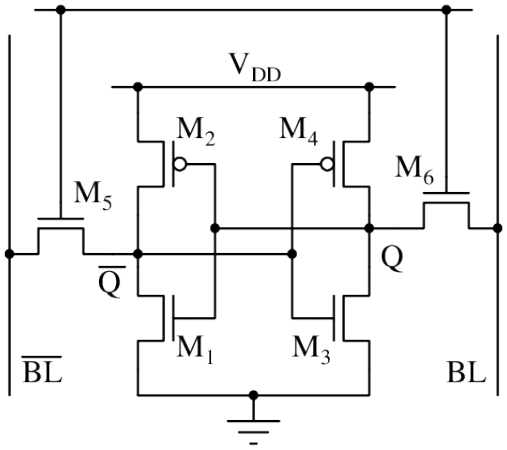


Fig. 7

(i) Explain the procedure to write logic '1' to the cell with reference to Fig. 7. [15%]

(ii) Transistor scaling leads to higher power dissipation. Suggest how FinFET could mitigate this problem. [15%]

Solution:

(i) The writing of '1' starts with setting BL to 1 and $\overline{BL}$ to 0. Then the access transistors M_5 and M_6 are turned on (asserted) to allow BL and $\overline{BL}$ to drive Q and $\overline{Q}$ respectively, potentially switching the metastable point of the cross-coupled inverter. After that, M_5 and M_6 are turned off to disconnect Q from BL.

(ii) The SRAM cell will benefit from replacing M_1, M_2, M_3, M_4 with FinFETs. This is because the static power of the SRAM cell comes from the subthreshold currents of

the transistors across the diffusions, even when two out of four transistors are turned off. The structure of FinFET provides stronger control by surrounding the channels from three sides. Therefore its leakage current is smaller compared to the planar FET used conventionally.

END OF PAPER