IIITB COMET FOUNDATION

Name: HEMANTH.T
Batch: COMETFWC033
Date: 04 September 2025

GATE 2018 ph, Question 13 — Detailed Analysis

Question 13

A 2-to-1 multiplexer selects between inputs A_0 and A_1 using select line C. The output is $X = A_0$ when $C = 0$ and $X = A_1$ when $C = 1$. Which gate-level implementation corresponds to this behavior?

Short Answer

The canonical implementation is:

$$\boxed{X = \overline{C}\,A_0 + C\,A_1}$$

(That is, invert C; AND $\overline{C}$ with A_0; AND C with A_1; OR the results.)

Detailed Analysis and Derivation

1. Reasoning from the specification:

- Required behavior:

- If $C = 0$ then $X = A_0$.
- If $C = 1$ then $X = A_1$.

- A standard way to encode a select is:

$$X = (\text{select for } A_0) \cdot A_0 \; + \; (\text{select for } A_1) \cdot A_1.$$

Here the select for A_0 is $\overline{C}$ and for A_1 is C. Hence:

$$X = \overline{C}A_0 + CA_1.$$

2. Check by cases:

$$C = 0 \implies X = \overline{0}A_0 + 0 \cdot A_1 = 1 \cdot A_0 + 0 = A_0,$$
$$C = 1 \implies X = \overline{1}A_0 + 1 \cdot A_1 = 0 \cdot A_0 + A_1 = A_1.$$

Hence the expression is correct.

Truth Table

Here is the full truth table listing all combinations of A_0, A_1, C and resulting X:

A_0	A_1	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Explanation of rows: - Rows where $C = 0$: $X = A_0$ (rows 1,3,5,7). - Rows where $C = 1$: $X = A_1$ (rows 2,4,6,8).

Sum-of-Minterms Form (optional)

If you want minterms (3 variables A_0, A_1, C ordered as $[A_0\, A_1\, C]$), $X = 1$ for combinations: - [0 1 1] decimal 3, - [1 0 0] decimal 4, - [1 1 0] decimal 6, - [1 1 1] decimal 7.

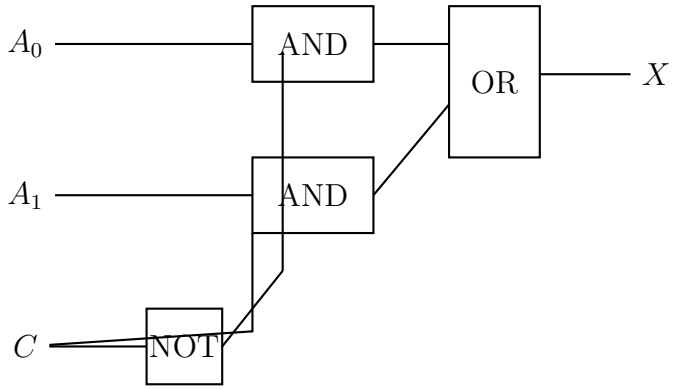
So one can write:

$$X = \sum m(3, 4, 6, 7),$$

which simplifies back to $X = \overline{C}A_0 + CA_1$.

Gate-Level Implementation (diagram)

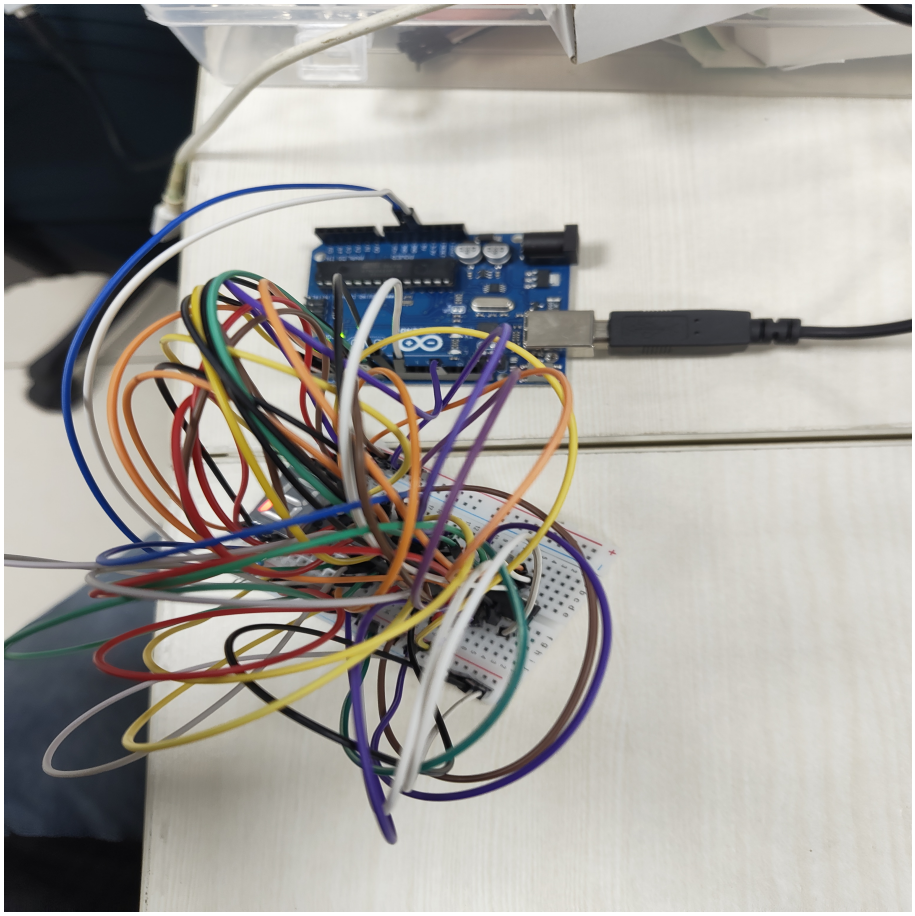
Below is a simple gate-level schematic drawn using basic TikZ shapes (no special gate library required). (Notation: rectangles labeled AND/OR/NOT are used to represent gates.)



Implementation Steps (practical)

If you were to implement this on breadboard or in a simple digital trainer:

1. Use one inverter (NOT) for the select line C to obtain $\overline{C}$.
2. Use two 2-input AND gates:
   - AND1: inputs $\overline{C}$ and $A_0 \rightarrow$ output T_0.
   - AND2: inputs C and $A_1 \rightarrow$ output T_1.
3. Use one 2-input OR gate: OR inputs $T_0, T_1 \rightarrow X$.



Conclusion

- The multiplexer equation $X = \overline{C}A_0 + CA_1$ correctly implements the required selection behavior.
- The full truth table above confirms the mapping of inputs to output for all input combinations.
- The gate-level implementation uses one inverter, two ANDs and one OR (standard, minimal realization).