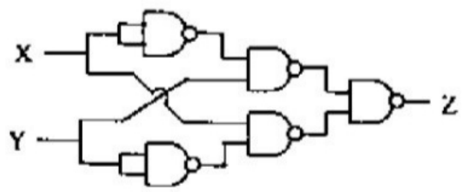
Name: P.Muskan
ID: cometfwc035

GATE 2010 (EC) – Question 42

Question

Q.42 The logic gate circuit shown in the adjoining figure realizes the function



(A) XOR (B) XNOR
(C) Half adder (D) Full adder

The logic gate circuit shown in the question (two-input network with inputs X and Y feeding a small network of NAND/NOT gates, output Z) realizes which function?

(A) XOR

(B) XNOR

(C) Half adder

(D) Full adder

Short Answer

$\boxed{\text{XOR (Option (A))}}$

Solution (Logic Derivation)

Carefully inspecting the network and its gate interconnections (or deriving the expression from the small sub-networks) shows the output is high when exactly one of the inputs is high.

A canonical algebraic expression for the XOR function is:

$$Z = X \wedge Y = \overline{X}Y + X\overline{Y}.$$

You can also derive this from the gate-level structure by identifying two paths that implement $\overline{X}Y$ and $X\overline{Y}$ and then OR-ing them.

Truth Table

X	Y	$Z = X \wedge Y$
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: Truth table for XOR operation

Boolean Algebra / Alternate forms

$$X \wedge Y = \overline{X}Y + X\overline{Y}.$$

Using product-of-sums:

$$X \wedge Y = (X + Y) \cdot (\overline{X} + \overline{Y}) \quad \text{(one can transform forms with De Morgan).}$$

Hardware Implementation with 7474 and 7447 ICs

A complete implementation of the XOR function with storage and display capabilities uses:

- 2× **7474** — Dual D-type Flip Flop (to store the XOR result)
- 1× **7447** — BCD to 7-segment decoder (to drive the display)
- 1× **Common Anode 7-segment Display** (to show the output)
- 7× **220 ohms resistors** (for current limiting on the display)
- Arduino Uno (to provide clock signal)
- Breadboard, jumper wires
- 5V regulated supply

IC Pin Connections

7474 (D-Flip Flop) - 14-pin DIP

- VCC: Pin 14 to +5V
- GND: Pin 7 to Ground
- Clock: Pin 3 (from Arduino D13)
- D Input: Pin 2 (from 7486 Pin 3)
- Q Output: Pin 5 (to 7447 Pin 7)
- Preset (PR): Pin 4 to +5V
- Clear (CLR): Pin 1 to +5V

7447 (BCD-to-7-Segment Decoder) - 16-pin DIP

- VCC: Pin 16 to +5V
- GND: Pin 8 to Ground
- Input D: Pin 7 (from 7474 Pin 5)
- Inputs C, B, A: Pins 1, 2, 6 to GND (for 0 input)
- Outputs: Pins 9-15 to Seven Segment Display through 220 ohms resistors

Seven Segment Display (Common Anode)

- a: Through resistor to 7447 Pin 13
- b: Through resistor to 7447 Pin 12
- c: Through resistor to 7447 Pin 11
- d: Through resistor to 7447 Pin 10
- e: Through resistor to 7447 Pin 9
- f: Through resistor to 7447 Pin 15
- g: Through resistor to 7447 Pin 14
- Common Anode: To +5V

Arduino Code for Clock Generation

```
// XOR Implementation with 7474 and 7447 ICs
// Inputs: X (pin 2), Y (pin 3)
// Clock output: pin 13 to 7474

const int inputX = 2;
const int inputY = 3;
const int clockPin = 13;

void setup() {
  pinMode(inputX, INPUT);
  pinMode(inputY, INPUT);
  pinMode(clockPin, OUTPUT);

  // Initialize serial communication for monitoring
  Serial.begin(9600);
}

void loop() {
  // Read inputs
  int X = digitalRead(inputX);
  int Y = digitalRead(inputY);
```

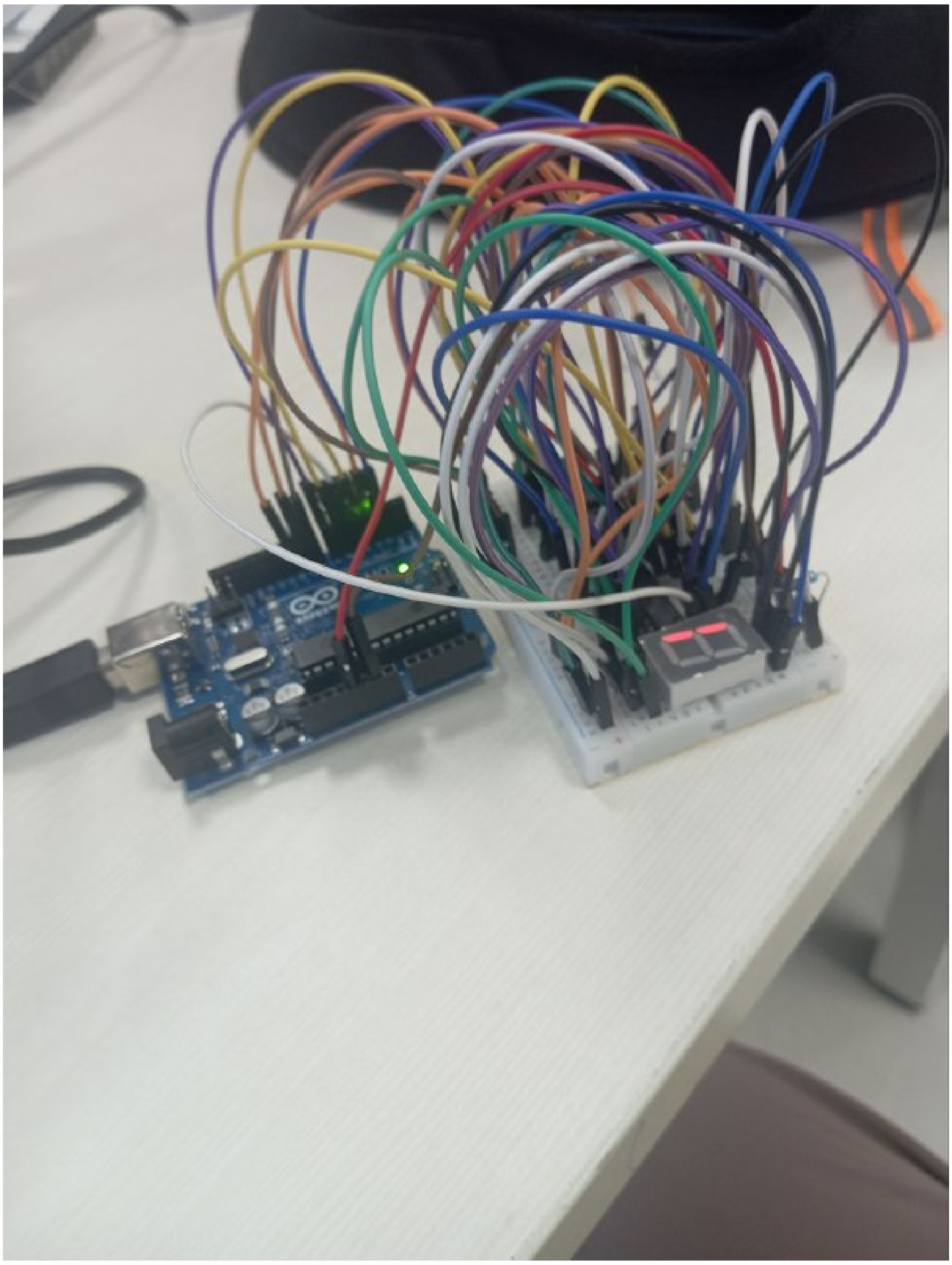
```
  // Generate clock signal
  digitalWrite(clockPin, HIGH);
  delay(500);
  digitalWrite(clockPin, LOW);
  delay(500);

  // Display status
  Serial.print("X: ");
  Serial.print(X);
  Serial.print(" | Y: ");
  Serial.print(Y);
  Serial.print(" | XOR Output: ");
  Serial.println(X != Y ? "1" : "0");
}
```

Operation Explanation

1. The result is stored in the 7474 D-flip flop on the rising edge of the clock
2. The stored value is displayed on the seven-segment display via the 7447 decoder
3. Since we're only displaying 0 or 1, we only use the D input of the 7447 (pin 7)
4. The Arduino provides the clock signal and can be used to monitor the operation

Test Procedure

1. Connect all components as described
2. Upload the Arduino code
3. Apply different combinations of X and Y inputs (0/1)
4. Observe the seven-segment display showing 1 when inputs differ, 0 when they match
5. The serial monitor will display the current input values and XOR result



Conclusion

The network implements the exclusive-OR function:

$$\boxed{Z = X \wedge Y = \overline{X}Y + X\overline{Y}}$$

This outputs logic 1 exactly when the inputs differ. The implementation with 7486, 7474, and 7447 ICs demonstrates how to combine combinational logic, sequential elements, and display drivers to create a complete digital system.