**Lab Report**

**Experiment No:01**

**Experiment Title:**

**Design and Implementation of 2-to-4 Line Decoder**

**Objective:**

* To design and implement a **2-to-4 line decoder** using basic logic gates.
* To derive the Boolean expressions for each output.
* To verify decoder operation with the truth table.

**Apparatus Required:**

* Breadboard
* IC 7404 (NOT gates), IC 7408 (AND gates)
* 2 toggle switches (for inputs A and B)
* 4 LEDs (for outputs Y0 to Y3)
* Resistors
* Connecting wires
* 5V Power supply

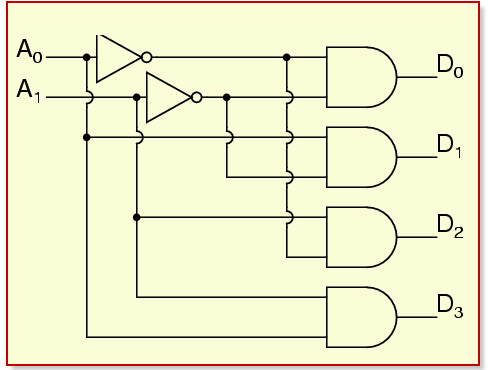
**Theory:**

A **Decoder** is a combinational circuit that decodes a binary input into a unique output line.

A **2-to-4 decoder** has:

* **2 input lines**: A1 (MSB), A0 (LSB)
* **4 output lines**: Y0, Y1, Y2, Y3
* Only **one output** is active (HIGH = 1) for any valid input.

**Circuit diagram :**

****

**Truth Table (2-to-4 Decoder):**

| **A1** | **A0** | **Y0** | **Y1** | **Y2** | **Y3** |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

Each output line represents a unique binary input. The active line is 1; all others are 0.

**Boolean Equations for Outputs:**

Let inputs be **A1 (MSB)** and **A0 (LSB)**.  
Let A1' and A0' be the complements of A1 and A0.

* **Y0 = A1' · A0'**
* **Y1 = A1' · A0**
* **Y2 = A1 · A0'**
* **Y3 = A1 · A0**

**Logic Circuit Diagram Description:**

1. **Inputs:** A1 and A0 from switches

**Inverters:**

* + Invert A1 → A1'
  + Invert A0 → A0'

**AND Gates (2-input):**

* + Y0 = A1' AND A0'
  + Y1 = A1' AND A0
  + Y2 = A1 AND A0'
  + Y3 = A1 AND A0

1. **Outputs:** Connect each AND gate output to an LED to indicate activation.

**Procedure:**

1. Place the ICs (7404 and 7408) on the breadboard.
2. Connect switches for inputs A1 and A0.
3. Use NOT gates to create A1' and A0'.
4. Feed proper combinations of inputs to each AND gate.
5. Connect AND gate outputs to LEDs via resistors.
6. Apply different combinations to inputs and verify which output is active.
7. Compare results with the truth table.

**Observation Table:**

| **A1** | **A0** | **Expected Output** | **Observed Output** |
| --- | --- | --- | --- |
| 0 | 0 | Y0 = 1 | Y0 LED ON |
| 0 | 1 | Y1 = 1 | Y1 LED ON |
| 1 | 0 | Y2 = 1 | Y2 LED ON |
| 1 | 1 | Y3 = 1 | Y3 LED ON |

**Result:**

* The 2-to-4 decoder circuit was successfully implemented.
* For each input combination, exactly one corresponding output line was active, matching the truth table.

**Conclusion:**

In this experiment, we designed and built a **2-to-4 line decoder** using **NOT and AND gates**. The decoder activated exactly one output based on the binary input values. This demonstrates the essential role of decoders in digital logic and address decoding.

**Experiment No:02**

**Experiment Title:**

**Design and Implementation of Encoder Logic Circuit**

**Objective:**

* To design and implement a **4-to-2 line encoder** using basic logic gates.
* To derive the Boolean equations for the encoder outputs.
* To verify the encoder operation using a truth table.

**Apparatus Required:**

* Breadboard
* Logic gate ICs (7404 - NOT, 7432 - OR)
* 4 Push Buttons (Inputs D0–D3)
* 2 LEDs (for outputs A1 and A0)
* Resistors
* Connecting wires
* Power supply (5V)

**Theory:**

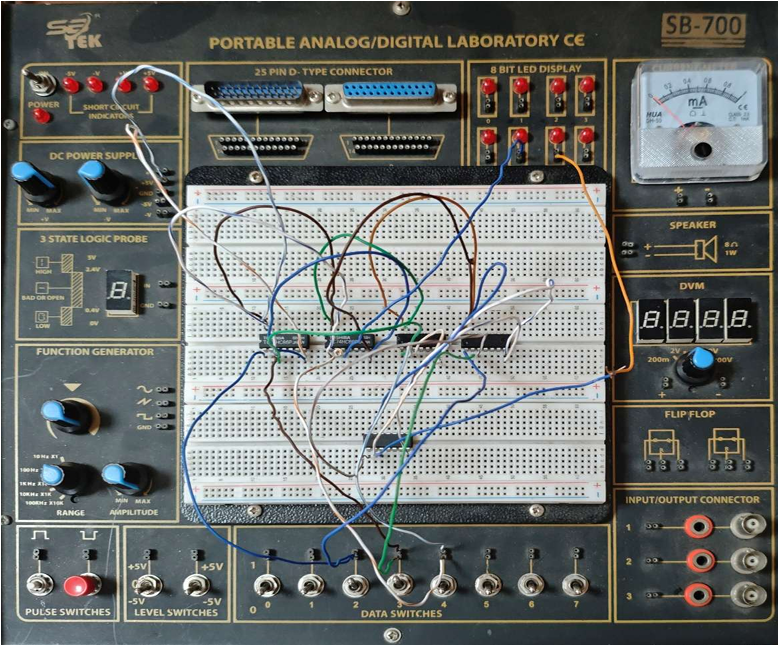
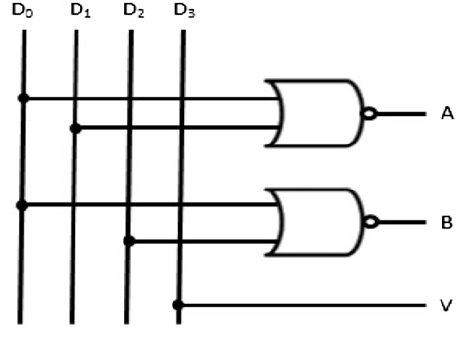
An **Encoder** is a combinational logic circuit that converts **2ⁿ input lines into an n-bit binary output**. It performs the **reverse operation of a decoder**. In a **4-to-2 encoder**, only **one input** should be active (HIGH) at a time. The circuit generates a binary code based on the position of the active input.

**Input and Output Mapping (4-to-2 Encoder):**

| **Input Lines** | **A1** | **A0** |
| --- | --- | --- |
| D0 = 1 | 0 | 0 |
| D1 = 1 | 0 | 1 |
| D2 = 1 | 1 | 0 |
| D3 = 1 | 1 | 1 |

Only one input (D0–D3) is active at a time. If multiple inputs are active, a **priority encoder** is used (not discussed here).

**Circuit diagram**



**Truth Table (4-to-2 Encoder):**

| **D3** | **D2** | **D1** | **D0** | **A1** | **A0** |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |

**Boolean Expressions for Outputs:**

From the truth table, we derive:

* **A1 = D2 + D3**
* **A0 = D1 + D3**

These are simple **OR operations** based on which input is active.

**Logic Circuit Diagram Description:**

1. **Inputs**: D0, D1, D2, D3 (active in HIGH)
2. **Outputs**: A1 and A0
3. **OR Gates**:
   * A1 = D2 OR D3
   * A0 = D1 OR D3
4. **Connections**:
   * Connect D2 and D3 to one OR gate → Output = A1
   * Connect D1 and D3 to another OR gate → Output = A0
   * Connect A1 and A0 to LEDs

**Procedure:**

1. Connect 4 push buttons as D0 to D3 on the breadboard.
2. Use 7432 OR gate IC to implement the output logic.
3. Connect outputs A1 and A0 to LEDs via resistors.
4. Activate each input one by one (only one at a time).
5. Observe the corresponding binary output.
6. Compare the observed output with the truth table.

**Observation Table:**

| **Input** | **Expected Output (A1 A0)** | **Observed Output** |
| --- | --- | --- |
| D0 | 00 | A1=0, A0=0 |
| D1 | 01 | A1=0, A0=1 |
| D2 | 10 | A1=1, A0=0 |
| D3 | 11 | A1=1, A0=1 |

**Result:**

* The encoder circuit correctly converted each active input to the corresponding binary code.
* The outputs matched the expected binary values based on the input position.

**Conclusion:**

In this experiment, a **4-to-2 encoder** was successfully designed using **basic OR gates**. The encoder converts a single active input into a unique binary code, confirming its theoretical design and practical application.