
# North South University

Department of Electrical & Computer Engineering

## PROJECT REPORT

**Course Code:** CSE231L

**Course Title:** Digital Logic Design

**Section:** 09

**Project Name:**

***Seven Segment Display***

**Date of Submission:** 01st June, 2024

**Submitted by Group Number:** 02

**Group members:**


| *Name* | *ID* |
|---|---|
| **Sudipto Roy** | **2222756041** |
| **Md. Rayhan Sharif Rafsun** | **2222472042** |
| **Shakil Ahmed** | **2221453042** |
| **Munem Bulbul** | **2221314642** |


**Course Instructor:** Omar Ibne Shahid (OISD)

**Submitted To Lab Instructor:** Pritthika Dhar

**Project Title:** 7 Segment Display

## Objectives:

- To clearly and accurately display numerical digits (0-9) and different letters for various applications such as digital clocks, calculators, and electronic meters.
- To provide good visibility and readability from a reasonable distance and under various lighting conditions.
- To offer a compact and efficient way of displaying information that can be easily integrated into different electronic devices.
- To consume minimal power, making it suitable for battery-operated devices.
- To offer a cost-effective solution for numerical displays in a variety of consumer and industrial electronic devices.
- To be used in various configurations, such as single-digit displays or multi-digit arrays, for displaying more complex information.

## Apparatus:

### *For Basic Gates Implementation:*

1. Breadboard
2. 3 * IC 7408 Quadruple 2-input AND gates (12 gates)
3. 2 * IC 7432 Quadruple 2-input OR gates (6 gates)
4. 1 * IC 7411 Triple 3-input AND gates (1 gate)
5. 1 * IC 7404 Hex Inverters (3 gates)
6. 33 Ohm Resistor
7. 7 Segment Display
8. Wire
9. LED
10. Battery
11. 9 Volt to 5 Volt Converter

### *For NAND Gates Implementation:*

1. Breadboard
2. 6 * IC 7400 Quadruple 2-input NAND gates (22 gates)
3. 1 * IC 7410 Triple 3-input NAND gates (1 gate)
4. 33 Ohm Resistor
5. 7 Segment Display
6. Wire

7. LED
8. Battery
9. 9 Volt to 5 Volt Converter

### *For NOR Gates Imlementation:*

1. Breadboard
2. 6 * IC 7402 Quadruple 2-input NOR gates (24 gates)
3. 1 * IC 7427 Triple 3-input NOR gates (2 gate)
4. 33 Ohm Resistor
5. 7 Segment Display
6. Wire
7. LED
8. Battery
9. 9 Volt to 5 Volt Converter

### *For MUX Implementation:*

1. Breadboard
2. 4 * IC 74LS153 4:1 Multiplexer
3. 1 * IC 7404 Hex Inverters (1 gate)
4. 33 Ohm Resistor
5. 7 Segment Display
6. Wire
7. LED
8. Battery
9. 9 Volt to 5 Volt Converter

### *For Decoder Implementation:*

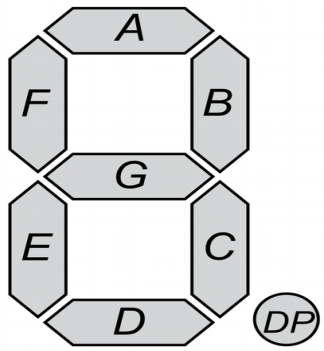
1. Breadboard
2. 1 * 74HC238 3 to 8 Line Decoder
3. 2 *  IC 7432 Quadruple 2-input OR gates (5 gates)
4. 2 *  IC 4075 Triple 3-input OR gates (6 gates)
5. 33 Ohm Resistor
6. 7 Segment Display
7. Wire
8. LED
9. 9 Volt to 5 Volt Converter

## Theory:

A seven segment display is widely used electric component that can represent numeric digits and some alphabetic characters. It consists of seven individual LED segments arranged in a specific pattern, allowing the display of various combinations to form the desired characters. Each segments represents a particular part of the character, and by selectively activating or deactivating these segments, different digits or characters can be displayed.

**Segments and Mapping**: The seven segment of typical seven-segment display are labeled as A, B, C, D, E, F and G. These segments are usually arranged in a pattern similar to the number "8". With segment 'G' forming the horizontal line in the middle and 'A' to 'F' forming the remaining vertical and diagonal lines. By activating or deactivating specific segments, different characters can be formed.

**Numeric Mapping:** The segments of a seven segment display can be controlled individually to represent the ten decimal (0-9) numbers. The following picture demonstrates the mapping of each segment to its corresponding digit:




**Character Mapping:** In addition to numeric digits, some alphabetic characters can also be displayed on a seven segment display. By utilizing the combinations of segments, the display can represent letters such as A, b, C, d, E, F, G, H, J, L, O, P, U, S etc.

**Controlling the display:** To display a specific digit or character, the segments must be activated or deactivated accordingly. This is typically achieved by providing appropriate control signals to the display circuitry.

There are several methods to control seven segment display, such as;

- ***Basic Logic Gates:*** Basic Logic Gates such as AND, OR and NOT gates can be used to control the segments based on input values. By designing circuits using the gates, the segments can be deactivated as needed.

- ***Universal Gates:*** Universal Gates such as; NAND or NOR gates offer flexibility in designing the display circuit. These gates can perform any logic function including controlling the segments of the display.

- ***Multiplexing:*** Multiplexing involves using a multiplexer to select the desired segments for each digit. The segments are sequentially activated and refreshed at the high rate to create illusion of continuous display.

- ***Decoders:*** Decoders are used to convert binary inputs to their corresponding segment activation. By providing appropriate input to decoder, the desired digit or character.

## Circuit Diagram:

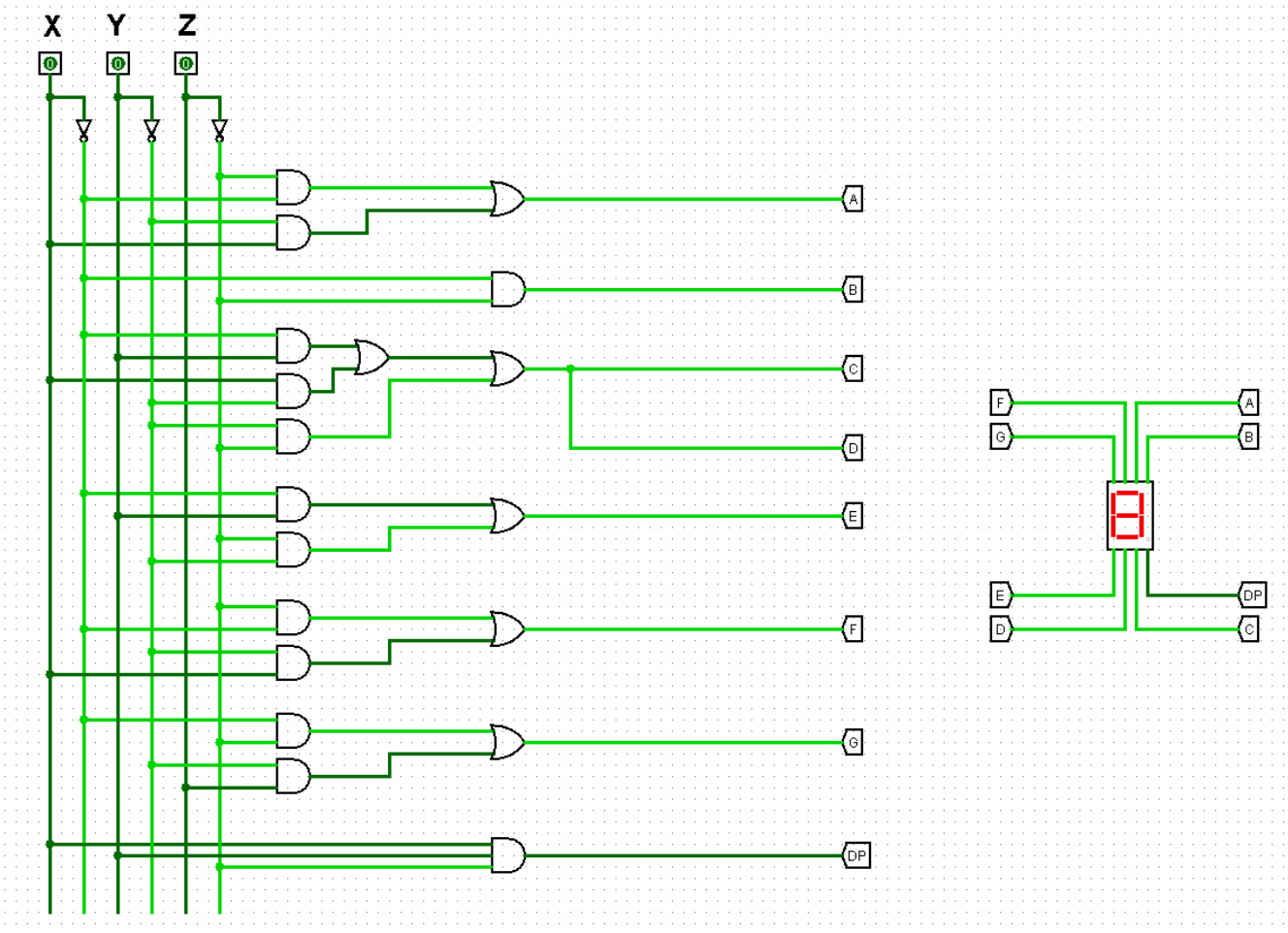



Figure 01 - Using Basic Gates

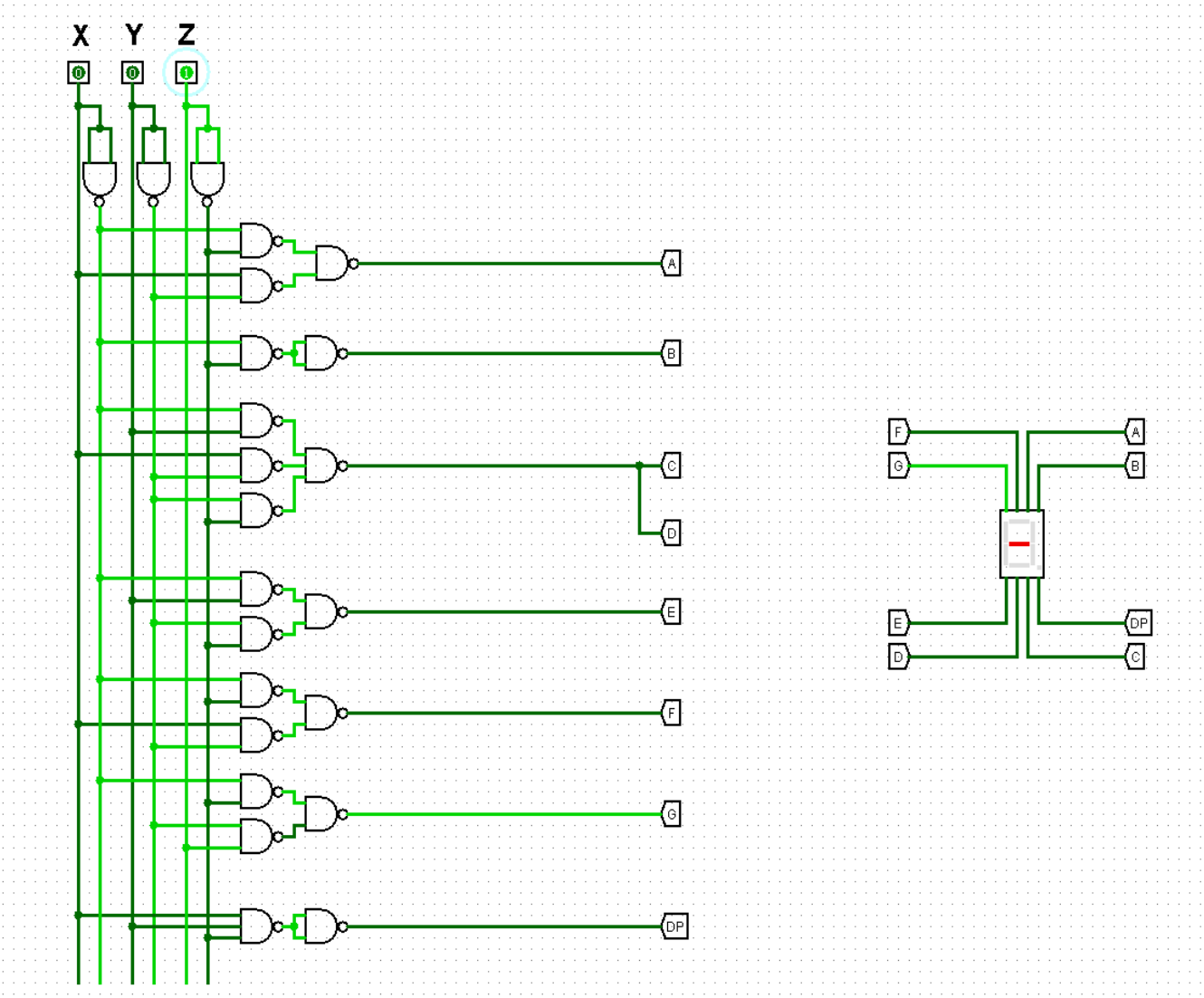



Figure 02 – Using NAND Gates

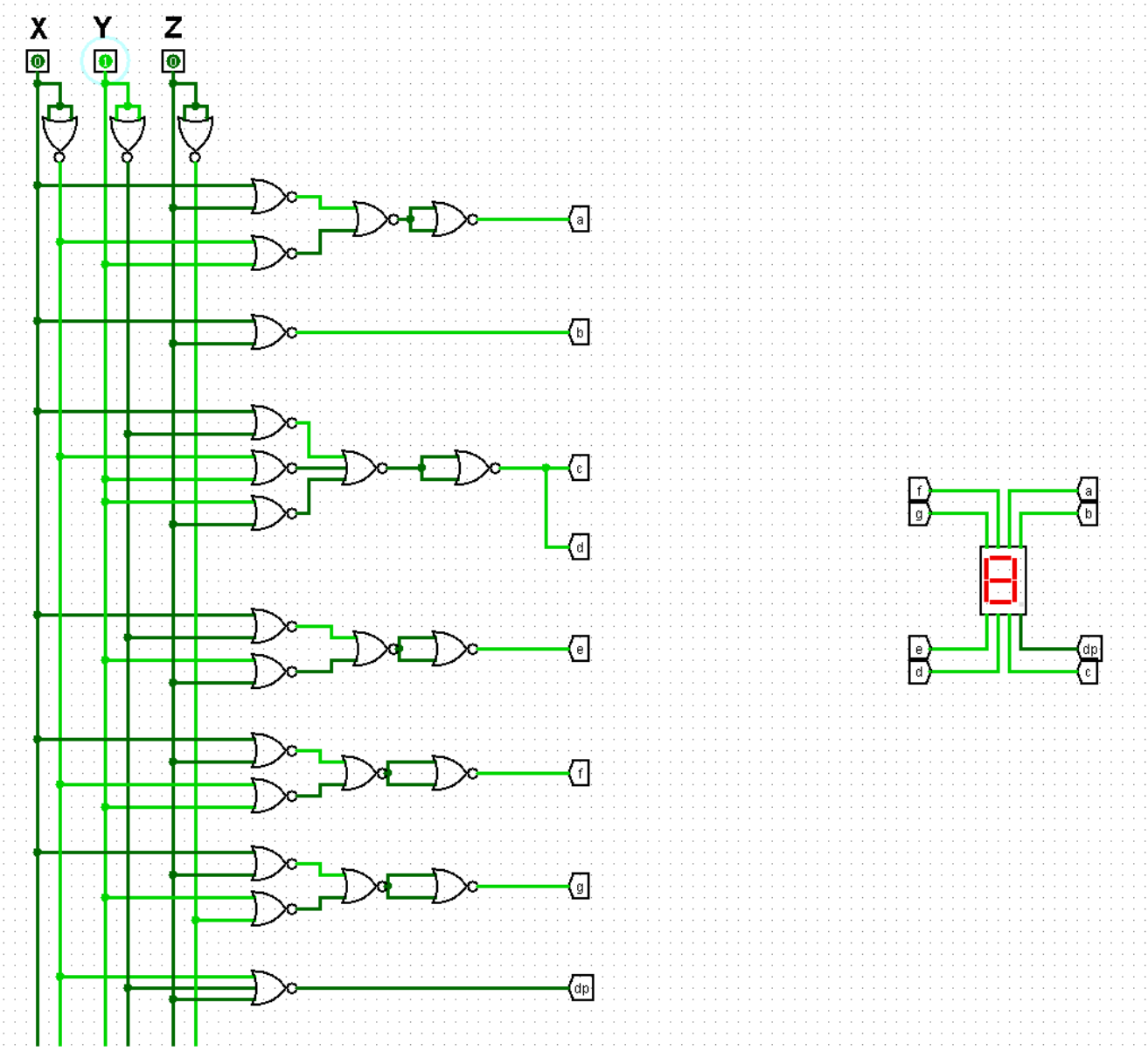



Figure 03 – Using NOR Gates

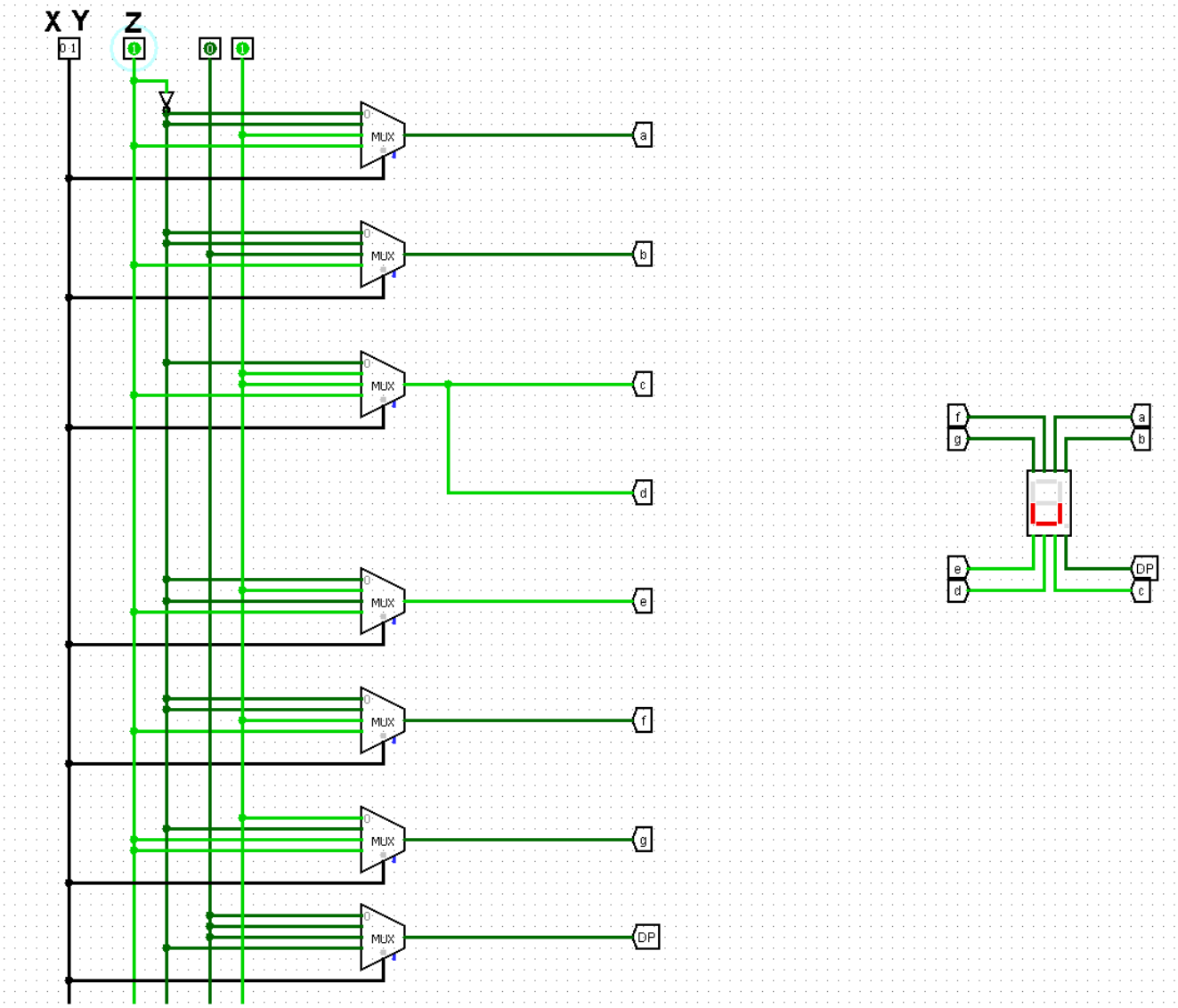



Figure 04 – Using MUX

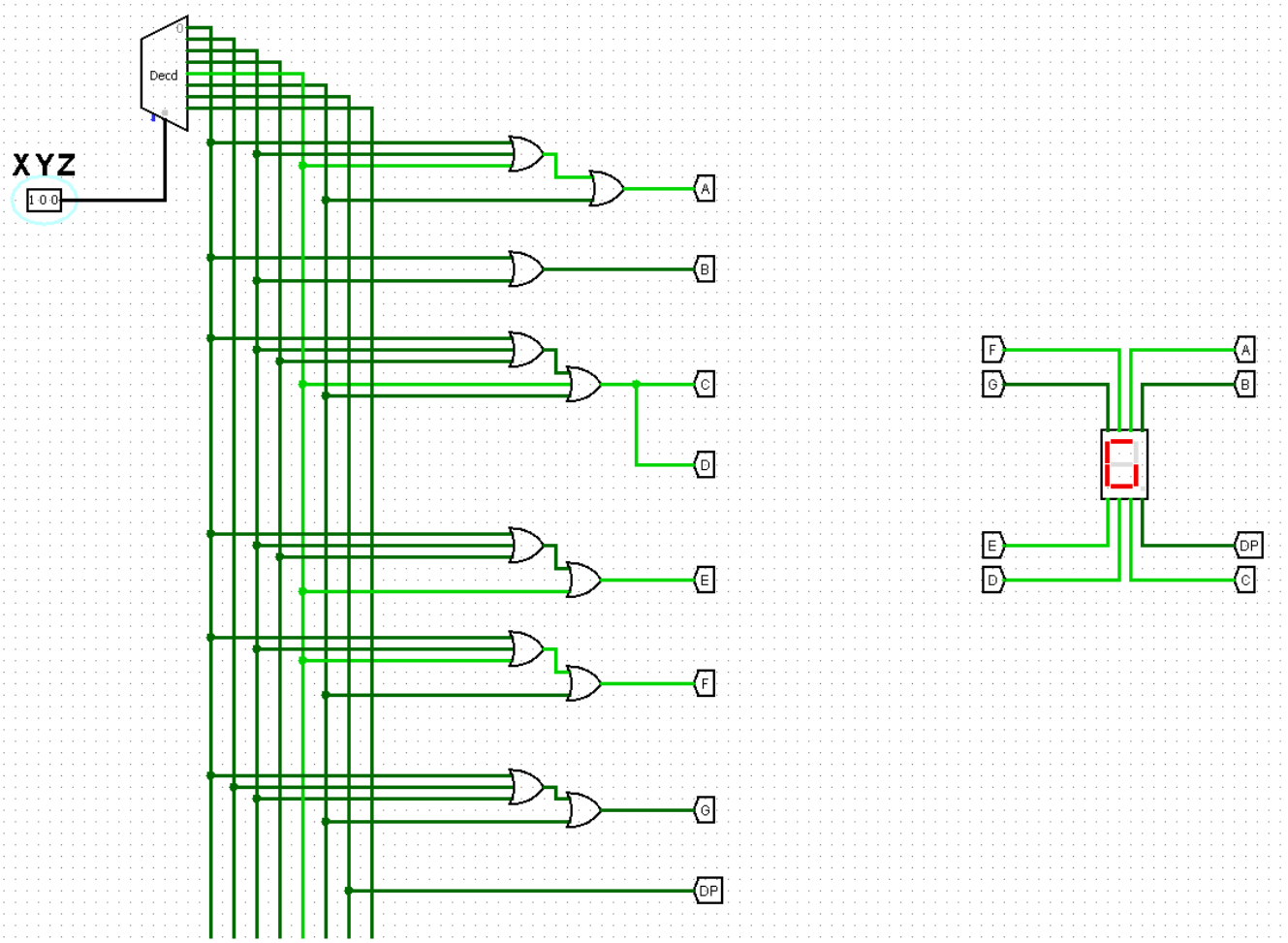



Figure 04 – Using Decoder

## Experimental Procedure:

***<u>Using Basic Gates</u>***

To create seven segment display using basic gates, we will need to use logic gates like AND, OR and NOT gates. Our three inputs labeled as; X, Y and Z represents the three bits of binary number. Here are the steps:

***1. Create Truth Table:*** We started by creating a truth table that maps the binary input values to the desired segments that should be lit up for each number. For a seven segment display, we have segments labeled A, B, C, D, E, F and G. Each of these segment corresponds to a specific input combination. For example, for the number 0, all the segments except G lighted up. So, we created a truth table mapping the binary inputs X, Y, Z to the segments (A, B, C, D, E, F and G).

***2. Implementing segment A:*** Using truth table we implemented the logic for segment A. This segment is typically controlled by inputs of X, Y and Z. And then we used AND, OR and NOT gates to achieve the desired logic expression for segment A.

***3. Implementing segments B, C, D, E, F and G:*** We repeated the same process of step-02 to implement segments B, C, D, E, F and G.

***4. Implementing the DP:*** We repeated the same process to implement the DP.

***5. Connect Output to the Seven Segment Display:*** Connect outputs of each segments logic to the corresponding segments on the seven segment display.

***6. Test the display:*** Verify that the implemented logic functions correctly by providing different binary inputs and observing the corresponding segments on the Seven Segment Display accordingly. Repeat this process for each segments (B, C, D, E, F, G and DP) using appropriate logic expressions. And by connecting outputs to the corresponding segments on the Seven Segment Display.

Cost Analysis:

| Components | Quantity | Unit Price | Total |
|---|---|---|---|
| IC 7408 Quadruple 2-input AND gates | 3 | 25 | 75 |
| IC 7432 Quadruple 2-input OR gates | 2 | 25 | 50 |
| IC 7411 Triple 3-input AND gates | 1 | 35 | 35 |
| IC 7404 Hex Inverters | 1 | 20 | 20 |
| Others | | | 530 |
| | | | Total= 710 |

***Using NAND gates***

1. Creating Seven Segment Display using universal gates is an another alternative approach. Firstly, we placed all the IC's in appropriate position.

2. We placed wires according to the circuitry now. And, placed all the NAND gates in the IC.

3. Now, we placed the outputs to the seven segment display. And placed 2 resistor in seven segment display also.

4. Now, we supplied Ground and VCC(+5V) connection from battery to all the ICs in respectively in pin number 7 and pin number 14.

5. We checked our desired output in the seven segment by switching all the combination.

Cost Analysis:

| Components | Quantity | Unit Price | Total |
|---|---|---|---|
| IC 7400 Quadruple 2-input NAND gates | 6 | 25 | 150 |
| IC 7410 Triple 3-input NAND gates | 1 | 25 | 25 |
| Others | | | 530 |
| | | | Total= 705 |

***Using NOR gates***

1. Creating Seven Segment Display using universal gates is an another alternative approach. Firstly, we placed all the IC's in appropriate position.

2. We placed wires according to the circuitry now. And, placed all the NOR gates in the IC.

3. Now, we placed the outputs to the seven segment display. And placed 2 resistor in seven segment display also.

4. Now, we supplied Ground and VCC(+5V) connection from battery to all the ICs in respectively in pin number 7 and pin number 14.

5. We checked our desired output in the seven segment by switching all the combination.

Cost Analysis:

| Components | Quantity | Unit Price | Total |
|---|---|---|---|
| IC 7402 Quadruple 2-input NOR gates | 6 | 25 | 150 |
| IC 7427 Triple 3-input NOR gates | 1 | 25 | 25 |
| Others | | | 530 |
| | | | Total= 705 |

It requires one more gates than NAND gates implementation.

***<u>Using MUX</u>***

1. We gathered the required components multiplexer chip, a seven segment display, resistors, power supply and connecting wires.

2. Understood the pin configuration of the multiplexer. Here we used 4:1 MUX, IC-74153

3. Determined the logic input combinations for each letters. We provided each data input line for the MUX, using X, Y as selection inputs, S1 and S0 respectively. Noted the values in Data input column by creating truth table to determine the combination of inputs required to display.

4. Connected input pins to the logic input combinations based on the truth table. Connected input pins of the MUX chips to appropriate control signals.

5. We connected VCC(+5 v) to the pin no-16 and ground to pin no-8.

| | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **DP** |
|---|---|---|---|---|---|---|---|---|
| **i1** | Z’ | Z’ | Z’ | Z’ | Z’ | Z’ | 1 | 0 |
| **i2** | Z’ | Z’ | 1 | 1 | 1 | Z’ | Z’ | 0 |
| **i3** | 1 | 0 | 1 | 1 | Z’ | 1 | Z | 0 |
| **i4** | Z | Z | Z | Z | Z | Z | Z | Z’ |

6. Now according to the upper table we inserted input in our MUX
7. Then, we tested all the combination accroring to our experimental data table.

Cost Analysis:

| Components | Quantity | Unit Price | Total |
|---|---|---|---|
| IC 74LS153 4:1 Multiplexer | 4 | 42 | 168 |
| IC 7404 Hex Inverters | 1 | 20 | 20 |
| Others | | | 530 |
| | | | Total= 718 |

***Using Decoder***

To create seven segment display using decoders, we need a decoder with at least three inputs. So we use a 3:8 decoder. To represent the our number and letters using a 3:8 decoder with a seven segment display, we need to follow these steps -

1. Wire up the IC 74183 .

2. Set the Enable inputs to the appropriate values. G1 should be set to High and both G2A and G2B should be set to Low.
3. The 3 select inputs (X Y Z) should be connected to 3 binary switches and the 8 outputs should be connected to Seven Segment Display.

4.Now, we check the output on Seven Segment Display for all combinations.

Cost Analysis:

| Components | Quantity | Unit Price | Total |
|---|---|---|---|
| 74HC238 3 to 8 Line Decoder | 1 | 80 | 80 |
| IC 7432 Quadruple 2-input OR gates | 2 | 25 | 50 |
| IC 4075 Triple 3-input OR gates | 2 | 25 | 50 |
| Others | | | 530 |
| | | | Total= 710 |

## Experimental Data Table:

Characters to print: **8-BuGS.**

| Character | Input | | | A | B | C | D | E | F | G | DP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | X | Y | Z | | | | | | | | |
| **8** | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| **-** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **B** | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| **u** | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| **G** | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| **S** | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| **.** | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

***Boolean Algebra:***

A = X'Y'Z' + X'YZ' + XY'Z' + XY'Z
B = X'Y'Z' +X'YZ'
C = X'Y'Z' + X'YZ' + X'YZ + XY'Z' + XY'Z
D = X'Y'Z' + X'YZ' + X'YZ + XY'Z' + XY'Z
E = X'Y'Z' + X'YZ' + X'YZ + XY'Z'
F = X'Y'Z' + X'YZ' + XY'Z' + XY'Z
G = X'Y'Z' + X'Y'Z + X'YZ' + XY'Z
DP = XYZ'

K-map:

For A:

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | 0 |

A = X'Z' + XY'

For B:

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 0 | 0 | X | 0 |

B = X'Z'

For C:

| | | | |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 1 | 1 | X | 0 |

C = Y’Z’ + XY’ +X’Y

For D:

| | | | |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 1 | 1 | X | 0 |

D = Y’Z’ + XY’ +X’Y

For E:

| | | | |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 1 | 0 | X | 0 |

E = Y’Z’ + X’Y

For F:

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 1 | 1 | X | 0 |

F = X’Z’ + XY’

For G:

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
| 0 | 1 | X | 0 |

G = X’Z’ + Y’Z

For DP:

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | X | 1 |

DP = XYZ’

## Using NAND Gates

For A : $\bar{X}\bar{Z} + X\bar{Y}$

$$= \overline{\overline{\bar{X}\bar{Z} + X\bar{Y}}}$$

$$= \overline{\overline{(\bar{X}\bar{Z})} \cdot \overline{(X\bar{Y})}}$$

For B : $\bar{X}\bar{Z}$

$$= \overline{\overline{\bar{X}\bar{Z}}}$$

For C and D : $\bar{X}Y + \bar{Y}\bar{Z} + X\bar{Y}$

$$= \overline{\overline{\bar{X}Y + \bar{Y}\bar{Z} + X\bar{Y}}}$$

$$= \overline{\overline{\bar{X}Y} \cdot \overline{\bar{Y}\bar{Z}} \cdot \overline{X\bar{Y}}}$$

For E : $\bar{X}Y + \bar{Y}\bar{Z}$

$$= \overline{\overline{\bar{X}Y + \bar{Y}\bar{Z}}}$$

$$= \overline{\overline{\bar{X}Y} \cdot \overline{\bar{Y}\bar{Z}}}$$

For F : $\bar{X}\bar{Z} + X\bar{Y}$

$$= \overline{\overline{\bar{X}\bar{Z} + X\bar{Y}}}$$

$$= \overline{\overline{\bar{X}\bar{Z}} \cdot \overline{X\bar{Y}}}$$

For G : $\bar{X}\bar{Z} + \bar{Y}Z$

$$= \overline{\overline{\bar{X}\bar{Z} + \bar{Y}Z}}$$

$$= \overline{\overline{\bar{X}\bar{Z}} \cdot \overline{\bar{Y}Z}}$$

For DP: $XY\bar{Z}$

$= \overline{\overline{XY\bar{Z}}}$

## Using NOR Gates

For A : $\bar{X}\bar{Z} + X\bar{Y}$

$= \overline{\overline{\bar{X}\bar{Z}}} + \overline{\overline{X\bar{Y}}}$

$= \overline{\overline{\bar{X}} + \overline{\bar{Z}}} + \overline{\overline{X} + \overline{\bar{Y}}}$

$= \overline{X + Z} + \overline{X + \bar{Y}}$

$= \overline{\overline{X + Z} + \overline{X + \bar{Y}}}$

For B : $\bar{X}\bar{Z}$

$= \overline{\overline{\bar{X}\bar{Z}}}$

$= \overline{\overline{\bar{X}} + \overline{\bar{Z}}}$

$= \overline{X + Z}$

For C and D : $\bar{X}Y + \bar{Y}\bar{Z} + X\bar{Y}$

$= \overline{\overline{\bar{X}Y}} + \overline{\overline{\bar{Y}\bar{Z}}} + \overline{\overline{X\bar{Y}}}$

$= \overline{\overline{\bar{X}} + \overline{Y}} + \overline{\overline{\bar{Y}} + \overline{\bar{Z}}} + \overline{\overline{X} + \overline{\bar{Y}}}$

$= \overline{X + \bar{Y}} + \overline{Y + Z} + \overline{X + \bar{Y}}$

$= \overline{\overline{X + \bar{Y}} + \overline{Y + Z} + \overline{X + \bar{Y}}}$

For E : $\bar{X}Y + \bar{Y}\bar{Z}$

$= \overline{\overline{\bar{X}Y} + \overline{\bar{Y}\bar{Z}}}$

$= \overline{\overline{X+\bar{Y}} + \overline{Y+Z}}$

$= \overline{X+\bar{Y}+Y+Z}$

$= \overline{X+1+Z}$

For F : $\bar{X}\bar{Z} + X\bar{Y}$

$= \overline{\overline{\bar{X}\bar{Z}} + \overline{X\bar{Y}}}$

$= \overline{X+Z + \bar{X}+Y}$

$= \overline{X+Z+\bar{X}+Y}$

For G : $\bar{X}\bar{Z} + \bar{Y}Z$

$= \overline{\overline{\bar{X}\bar{Z}} + \overline{\bar{Y}Z}}$

$= \overline{X+Z + Y+\bar{Z}}$

$= \overline{X+Z+Y+\bar{Z}}$

For DP : $XY\bar{Z}$

$= \overline{\overline{XY\bar{Z}}}$

$= \overline{\bar{X} + \bar{Y} + Z}$

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

Using SOP Form

For A : $\bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + X\bar{Y}Z$

$= \sum m\,(0, 2, 4, 5)$

For B : $\bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z}$

$= \sum m\,(0, 2)$

For C and D : $\bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + \bar{X}YZ + X\bar{Y}\bar{Z} + X\bar{Y}Z$

$= \sum m\,(0, 2, 3, 4, 5)$

For E : $\bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + \bar{X}YZ + X\bar{Y}\bar{Z}$

$= \sum m\,(0, 2, 3, 4)$

For F : $\bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + X\bar{Y}Z$

$= \sum m\,(0, 2, 4, 5)$

For $G_1$ : $\bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}Z$

$= \sum m\,(0, 1, 2, 5)$

For DP : $XY\bar{Z}$

$= \sum m\,(6)$

## Using POS Form

For A : $(\bar{X} + \bar{Y} + Z)(\bar{X} + Y + Z)(X + Y + \bar{Z})$

$= \Pi M(1,3,6)$

For B : $(\bar{X} + \bar{Y} + Z)(\bar{X} + Y + Z)(X + \bar{Y} + \bar{Z})(X + \bar{Y} + Z)(X + Y + \bar{Z})$

$= \Pi M(1,3,4,5,6)$

For C and D : $(\bar{X} + \bar{Y} + Z)(X + Y + \bar{Z})$

$= \Pi M(1,6)$

For E : $(\bar{X} + \bar{Y} + Z)(X + \bar{Y} + Z)(X + Y + \bar{Z})$

$= \Pi M(1,5,6)$

For F : $(\bar{X} + \bar{Y} + Z)(\bar{X} + Y + Z)(X + Y + \bar{Z})$

$= \Pi M(1,3,6)$

For G : $(\bar{X} + Y + Z)(X + \bar{Y} + \bar{Z})(X + Y + \bar{Z})$

$= \Pi M(3,4,6)$

For DP : $(\bar{X} + \bar{Y} + \bar{Z})(\bar{X} + \bar{Y} + Z)(\bar{X} + Y + \bar{Z})(\bar{X} + Y + Z)(X + \bar{Y} + \bar{Z})$
$(X + \bar{Y} + Z)$

$= \Pi M(0,1,2,3,4,5)$

**Result:** After connecting all the wires to the IC's and Seven Segment Display, we checked all the combination according to our Truth Table. And we got all the outputs according our truth table.

**Discussion:** In this project, we made a Seven Segment Display using five methods. Here is a discussion given below -

1. ***Basic Logic Gates Method:*** Using basic logic gates AND, OR and NOT provided the flexibility in terms of customizing the circuit and allowed for fine gained control over display.

Pros: As we had specific requirements it was a good option.

Cons: Designing circuit using basic gates was more time consuming and complex because of so much wires and gates.

2. ***NAND Gates Method:*** Using NAND gates to perform any logic function, including controlling the segments of the display.

Pros: It provided high level of flexibility and versatility in designing circuit. In our case this approach was most cost efficient.

Cons: Compared to other methods, we faced less problem and cons in this NAND gates implementation. It requires deep understanding of universal gates behavior and combination.

3. ***NOR Gates Implementation:*** Using NOR gates to perform any logic function, including controlling the segments of the display.

Pros: It provided high level of flexibility and versatility in designing circuit. In our case this approach was most cost efficient.

Cons: Compared to NAND gates method, NOR gates implementation requires more gates. It requires deep understanding of universal gates behavior and combination.

4. ***Multiplexer (MUX) Method Implementation:*** The MUX method involves using a MUX to select the appropriate combination of segments to display each digit. WE had to use 7 MUX (4:1) to display all the letters properly.

Pros: The MUX method allowed dynamic control of the display. Since it consists selectivity it requires less components than the basic gates method.

Cons: In our case, this MUX approach cost more money on component than the NAND gate approach. Beside this, implementing a multiplexer-based display required addition circuitry for digit selection.

5. ***Decoder Method Implementation:*** In this method, we had to use 1 decoder IC (3:8) but it requires extra 6 - 3 input OR gates (2 - IC 4075) and 5 – 2 input OR gates (2 – IC 7432).

Pros: Using decoder simplified the design process as it directly mapped the input valyues to the appropriate segments. It was relatively straight forward to implement.

Cons: Using this method costs more and less sufficient for our project.

## Determining the best method:

We have used NAND gate implementation for its efficiency and low cost build.

Cost Analysis:

| Components | Quantity | Unit Price | Total |
|---|---|---|---|
| IC 7400 Quadruple 2-input NAND gates | 6 | 25 | 150 |
| IC 7410 Triple 3-input NAND gates | 1 | 25 | 25 |
| Breadboard | 2 | 155 | 310 |
| Battery (9 Volt) | 1 | 85 | 85 |
| 9 Volt to 5 Volt Converter | 1 | 1 | 15 |
| Wires | | | 100 |
| Resistor | | | 5 |
| Seven Segment Display | 1 | 15 | 15 |
| | | | Total= 705 |

# Sequential Part:

**Objectives:** The purpose of this theory is to propose a sequential component integration approach for the development of a Seven Segment Display Project. By organizing the project task and activities in a sequential manner, we aim to optimize resource allocation, streamline development process and ensure the successful implementation of a functional Seven Segment Display.

## Apparatus:

1. Breadboard
2. 2 * IC 7432 Quadruple 2-input OR gates (2 gates)
3. 1 * IC 7408 Quadruple 2-input AND gates (1 gate)
4. 2 * IC 7476 J-K Flip-Flop
5. 1 * 555 Timer IC
6. Wire
7. Battery
8. 9 Volt to 5 Volt Converter

## Theory:

Flip-flops are fundamental building blocks in digital electronics, primarily used for storage, synchronization, and sequencing of digital data. The JK, D, and T flip-flops are variations of basic flip-flops, each with unique characteristics and operational behaviors. Here's a detailed explanation of each:

### JK Flip-Flop

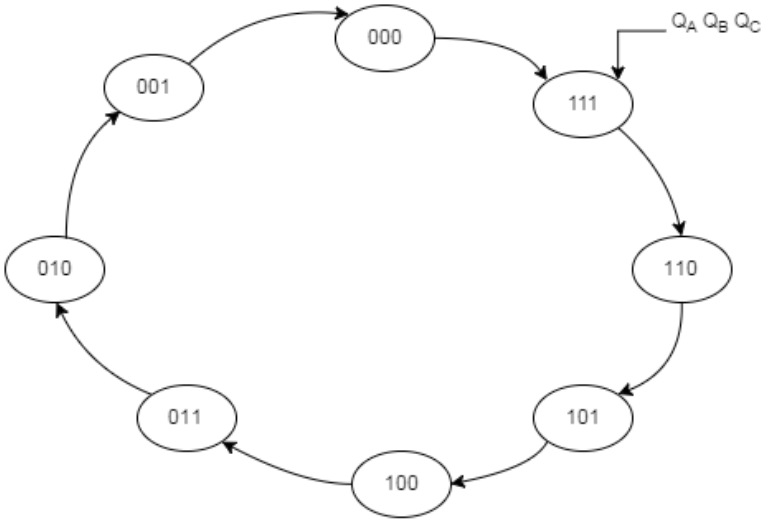
The JK flip-flop is a versatile and widely used type of flip-flop. It has two inputs labeled J and K, and two outputs, Q and $\bar{Q}$.

### D Flip-Flop

The D (Data or Delay) flip-flop is simpler and more commonly used in digital systems for its straightforward behavior. It has a single input labeled D and two outputs, Q and $\bar{Q}$.

### T Flip-Flop

The T (Toggle) flip-flop is derived from the JK flip-flop and is used mainly for toggling operations. It has a single input labeled T and two outputs, Q and $\bar{Q}$.




state diagram of 3 bit ripple down counter

## State Table:

**Table 01: (J-k Flip-flop)**

| Present state (A,B,C) | Next state | $J_A$ | $K_A$ | $J_B$ | $K_B$ | $J_C$ | $k_C$ |
|---|---|---|---|---|---|---|---|
| **000** | **001** | **0** | **x** | **0** | **x** | **1** | **x** |
| **001** | **010** | **0** | **x** | **1** | **x** | **x** | **1** |
| **010** | **011** | **0** | **x** | **x** | **0** | **1** | **x** |
| **011** | **100** | **1** | **x** | **x** | **1** | **x** | **1** |
| **100** | **101** | **x** | **0** | **0** | **x** | **1** | **x** |
| **101** | **110** | **x** | **0** | **1** | **x** | **x** | **1** |
| **110** | **111** | **x** | **0** | **x** | **0** | **1** | **x** |
| **111** | **000** | **x** | **1** | **x** | **1** | **x** | **1** |

## K-map:

| A’B’C’ | A’B’C | A’BC | A’BC’ |
|---|---|---|---|
| AB’C’ | AB’C | ABC | ABC’ |

For $J_A$:

| 0 | 0 | 1 | 0 |
|---|---|---|---|
| X | X | X | X |

$J_A = BC$

For $K_A$:

| X | X | X | X |
|---|---|---|---|
| 0 | 0 | X | 1 |

$K_A = B$

For $J_B$:

| 0 | 1 | X | 0 |
|---|---|---|---|
| 0 | 1 | X | 0 |

$J_B = C$

For $K_B$:

| X | X | 1 | 0 |
|---|---|---|---|
| X | X | X | 1 |

$K_B = A+C$

For Jc:

| 1 | X | X | 1 |
|---|---|---|---|
| 1 | X | X | 0 |

$J_C = A’ + B’$

For Kc:

| | | | |
|---|---|---|---|
| X | X | 1 | X |
| X | 1 | X | X |

$K_C = 1$

Cost Analysis:

| Components | Quantity | Unit Price | Total |
|---|---|---|---|
| IC 7476 JK Flip-Flop | 2 | 40 | 80 |
| IC 7432 Quadruple 2-input OR gates | 1 | 25 | 25 |
| IC 7408 Quadruple 2-input AND gates | 1 | 25 | 25 |
| | | | Total= 130 |

**Table 02: (D Flip-flop)**

| Present state (A,B,C) | Next state | $D_A$ | $D_B$ | $D_C$ |
|---|---|---|---|---|
| **000** | **001** | **0** | **0** | **1** |
| **001** | **010** | **0** | **1** | **0** |
| **010** | **011** | **0** | **1** | **1** |
| **011** | **100** | **1** | **0** | **0** |
| **100** | **101** | **1** | **0** | **1** |
| **101** | **110** | **1** | **1** | **0** |
| **110** | **111** | **1** | **1** | **1** |
| **111** | **000** | **X** | **X** | **X** |

For $D_A$:

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 1 | 1 | X | 1 |

$D_A = AB' + BC$

For $D_B$:

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 1 | X | 0 |

$D_B = B'C + A'BC'$

For $D_C$:

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 1 | 0 | X | 0 |

$D_C = B'C' + A'C'$

Cost Analysis:

| Components | Quantity | Unit Price | Total |
|---|---|---|---|
| IC 7474 D Flip-Flop | 2 | 40 | 80 |
| IC 7432 Quadruple 2-input OR gates | 2 | 25 | 50 |
| IC 7408 Quadruple 2-input AND gates | 2 | 25 | 50 |
| | | | Total= 180 |

**Table 03: (T flip-flop)**

| Present state (A,B,C) | Next state | $T_A$ | $T_B$ | $T_C$ |
|---|---|---|---|---|
| **000** | **001** | **0** | **0** | **1** |
| **001** | **010** | **0** | **1** | **1** |
| **010** | **011** | **0** | **0** | **1** |
| **011** | **100** | **1** | **1** | **1** |
| **100** | **101** | **0** | **0** | **1** |
| **101** | **110** | **0** | **1** | **1** |
| **110** | **111** | **1** | **1** | **0** |
| **111** | **000** | **X** | **X** | **X** |

For $T_A$:

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 0 | X | 1 |

$T_A = BC + AB$

For $T_B$:

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
| 0 | 1 | X | 1 |

$T_B = C+AB$

For $T_C$:

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | X | 0 |

$T_C = A'+B'$

Cost Analysis:

| Components | Quantity | Unit Price | Total |
|---|---|---|---|
| IC 7476 JK Flip-Flop | 2 | 40 | 80 |
| IC 7432 Quadruple 2-input OR gates | 1 | 25 | 25 |
| IC 7408 Quadruple 2-input AND gates | 1 | 25 | 25 |
| | | | Total= 130 |

## Circuit Diagram:

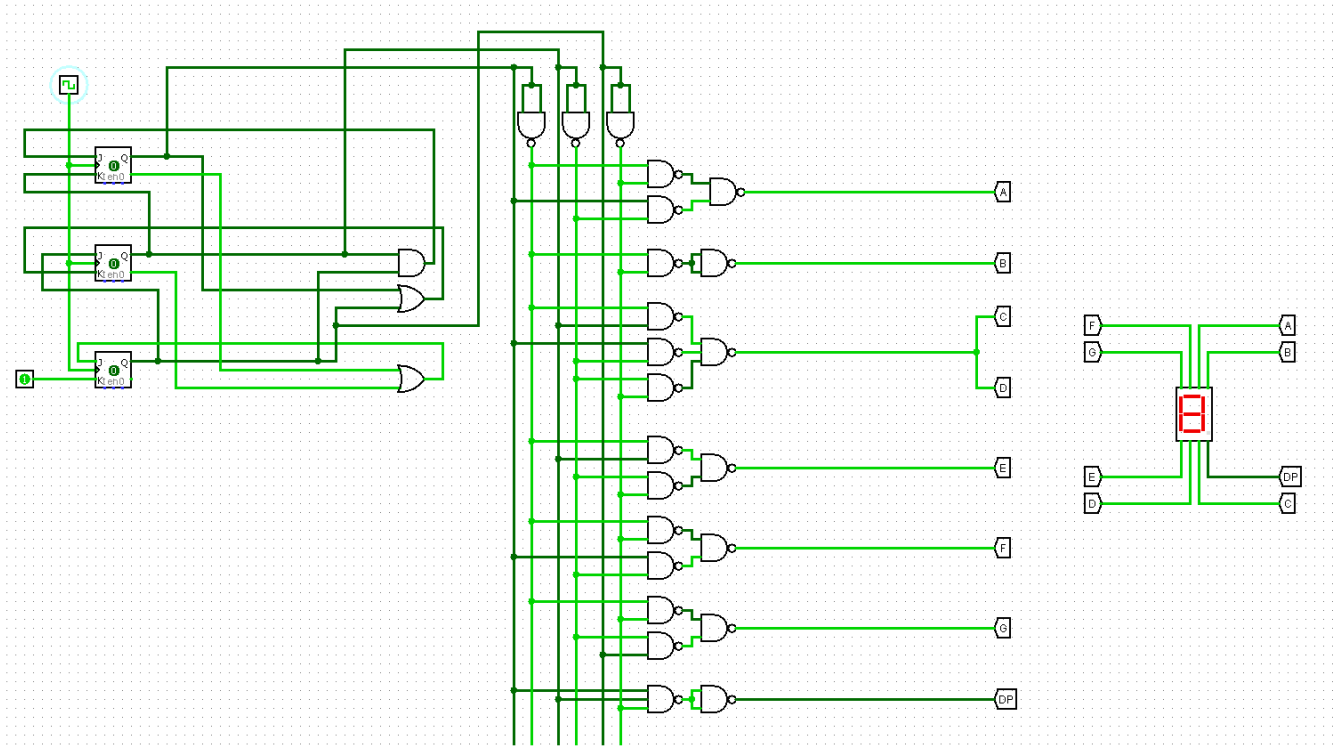



Figure – 06 : Sequential Part using JK Flip Flop

## Experimental Procedure:

1. From the K-map we found our equation. Then connect to the $J_A$, $J_B$, $J_C$ and $K_A$, $K_B$, $K_C$ as like the equation.

2. Here in 7476 IC we connected VCC in 14 no. pin and ground in 7 no. pin. We found our output in 3 and 5 no. pin, and it connects to computational part.

3. 13 and 10 no. pin are CLR, it connects to the VCC.

4. 12 and 9 no. pin are clock, we connected it to the 555 time IC's output 3 no. pin.

5. In 555 Timer IC, pin no 4,8 connected to VCC and 1 no pin connected to the ground.
Then 2 and 6 should be short. Then, 6 connect to a capacitor. And 6 to 7 connect a resistor. After that 7 connect to a resistor with VCC. And Output comes from 3 no pin.

**Result:** After connect to the all wires in 7476 J-K Flip-Flop and 555 timer IC, we cleared the button through 1 to 0. Then, it sequentially show 000 to 001 and so on. We got our output according to our experimental data table.

**Discussion:** In our seven-segment display project, we utilized JK flip-flops to achieve a stable and accurate numeric output. The JK flip-flops were configured to count in binary and control state transitions, allowing for precise and synchronous toggling. This binary count was then decoded to drive the appropriate segments on the display. The robustness and versatility of JK flip-flops ensured reliable performance, preventing glitches and timing issues, thus enabling the correct and consistent presentation of numbers on the seven-segment display.