

# DIGITAL ELECTRONIC INNOVATIVE ASSIGNMENT REPORT



Elevator Circuit Logic

SUBJECT: DIGITAL ELECTRONICS

SUBJECT CODE: 2CS507CC23

SUBMITTED TO: Prof. Kruti Lavingia

PREPARED BY:

NAME: **Harsh Patel**

ROLL NO: **25BCE521**

NAME: **Om Vaghela**

ROLL NO: **25BCE533**

Table of Contents

- 1. Objective ..... 3
- 2. Components Used ..... 4
- 3. Design and Operation ..... 5
  - 3.1 System Overview ..... 5
  - 3.2 State Machine Design ..... 6
  - 3.3 Circuit Implementation ..... 7
- 4. Working Procedure ..... 8
- 5. Circuit Schematics ..... 9
  - 5.1 Main Circuit ..... 9
  - 5.2 Sub-Circuit: Floor Traversal FSM ..... 10
  - 5.3 Sub-Circuit: 7-Segment Driver ..... 10
- 6. Applications ..... 11
- 7. Advantages ..... 11
- 8. Conclusion ..... 12

## 1. Objective

The objective of this project is to design and simulate a digital controller for a 4-floor elevator system (Ground, 1, 2, and 3) using Logisim. The system is designed to accept a floor call from one of four input buttons.

The core of the project is a sequential logic circuit, designed as a finite state machine (FSM), which manages the elevator's current floor state. As specified in the design brief, the elevator must proceed to any called floor by "reading through intermediate floors". This means the controller must sequentially increment or decrement its current floor state, one floor at a time, until it matches the destination floor.

The system provides real-time visual feedback by displaying the elevator's current floor number on a 7-segment display. It also includes logic to output the elevator's current status, indicating whether it is moving up, moving down, or has arrived at the correct floor.

## 2. Components Used

The circuit is constructed using several standard digital logic components and custom sub-circuits within Logisim.

Priority Encoder:

Used to manage the four floor-call input buttons. It takes four inputs and outputs a 2-bit binary number representing the highest-priority input (the highest floor called).

J-K Flip-Flops:

Two J-K flip-flops are used in the floor traversal sub-circuit. These are the memory elements of the sequential circuit, storing the current state (the 2-bit code for the current floor).

Combinational Logic Gates (AND, OR, NOT, XOR):

These gates are used extensively in both the main circuit and sub-circuits to build the required logic. They are used to implement the next-state logic for the J-K flip-flops (based on the derived equations) and to create the BCD-to-7-segment decoder.

Comparator:

A 2-bit comparator is used to continuously compare the 2-bit current floor (from the FSM) with the 2-bit destination floor (from the priority encoder).

7-Segment Display:

A standard 7-segment display component is used to visually show the current floor number (0, 1, 2, or 3).

Floor Traversal FSM (Sub-circuit):

This custom sub-circuit contains the J-K flip-flops and the combinational logic to implement the state machine's floor-to-floor movement.

7-Segment Driver (Sub-circuit):

This custom sub-circuit functions as a BCD-to-7-segment decoder.

Input Pins:

Four input pins are used to simulate the elevator call buttons for floors 0, 1, 2, and 3.

Output Pins:

Three output pins are connected to the comparator's outputs ( $A > B$ ,  $A = B$ ,  $A < B$ ) to indicate the elevator's status (Moving Down, At Floor, Moving Up).

Clock:

A clock component provides the synchronizing pulses required for the sequential FSM to transition between states.

### 3. Design and Operation

The circuit's functionality is based on a finite state machine (FSM) that controls the elevator's state, combined with peripheral components for input-processing and output display.

#### 3.1 System Overview

The overall data flow of the circuit is as follows:

- **Input:** The user presses one of the four **Input Pins** (floor call buttons).
- **Encoding:** These four inputs are fed into a **Priority Encoder**. The encoder outputs a 2-bit binary number ( $F_1F_0$ ) representing the selected destination floor.
- **State Logic:** This 2-bit destination ( $F_1F_0$ ) is sent to two components simultaneously: the **Comparator** and the **Floor Traversal FSM** (sub-circuit).
- **State Storage:** The FSM sub-circuit contains the sequential logic. It takes the destination ( $F_1F_0$ ), its own current state ( $Q_1Q_0$ ), and a **Clock** signal. On each clock pulse, it calculates its next state, effectively incrementing or decrementing the current floor by one until it reaches the destination. Its 2-bit output is the current floor ( $Q_1Q_0$ ).
- **Comparison:** The **Comparator** receives the destination ( $F_1F_0$ ) from the encoder and the current floor ( $Q_1Q_0$ ) from the FSM. It continuously compares them and lights up one of the three **Output Pins**:
  - $A < B$ : Destination is higher (Moving Up).
  - $A = B$ : Destination reached (At Floor).
  - $A > B$ : Destination is lower (Moving Down).
- **Display:** The 2-bit current floor ( $Q_1Q_0$ ) from the FSM is also sent to the **7-Segment Driver** (sub-circuit). This driver converts the binary code (00, 01, 10, 11) into the corresponding signals for the **7-Segment Display**, which then shows '0', '1', '2', or '3'.

#### 3.2 State Machine Design

The core of the controller is the sequential circuit, which was designed based on the state tables and K-maps provided in the project's planning documents.

The system has four states, representing the four floors (G, 1, 2, 3). These states are encoded using two state variables (and two J-K flip-flops),  $Q_1$  and  $Q_0$ . The inputs to the FSM are the 2-bit destination floor,  $F_1$  and  $F_0$ .

A state table was created to define the "Earliest Next State" based on the "Present State" ( $Q_1Q_0$ ) and the inputs ( $F_1F_0$ ). This table maps out all possible combinations and determines the required next state to ensure the elevator moves one floor at a time.

From this state table, K-maps were used to derive the simplified Boolean equations for the J and K inputs of each flip-flop. The derived equations for the  $Q_0$  flip-flop are:

- $J_0 = F_0 + Q_1 \oplus F_1$
- $K_0 = F_0' + Q_1 \oplus F_1$

Similarly, equations for  $J_1$  and  $K_1$  were derived from their respective K-maps. This logic ensures that if the current floor is '1' ( $Q_1Q_0=01$ ) and the destination is '3' ( $F_1F_0=11$ ), the next state on the clock pulse will be '2' ( $Q_1Q_0=10$ ), not '3' directly. This enforces the "traverse through intermediate floors" requirement.

### 3.3 Circuit Implementation

The theoretical state machine design is implemented in Logisim using distinct modules.

Floor Traversal Module:

This sub-circuit is the direct hardware implementation of the FSM. It contains the two J-K Flip-Flops that store the state  $Q_1Q_0$ . The inputs ( $F_1F_0$ ) and the current state outputs ( $Q_1Q_0$ ) are fed into a network of AND, OR, NOT, and XOR gates. This combinational logic implements the exact Boolean equations (like  $J_0 = F_0 + Q_1 \oplus F_1$ ) derived from the K-maps. The outputs of this logic are connected to the J and K inputs of the flip-flops, ensuring the correct next state is loaded on every clock pulse.

7-Segment Driver Module:

This sub-circuit is a combinational BCD-to-7-segment decoder. It takes a 4-bit input ( $W, X, Y, Z$ ). In the main circuit, the 2-bit current floor ( $Q_1Q_0$ ) is wired to the Y and Z inputs, while the W and X inputs are tied to a 0 constant. This correctly maps the floor codes (0000, 0001, 0010, 0011) to display the digits 0, 1, 2, and 3.

Main Circuit:

This is the top-level circuit that integrates all the components. It wires the four input pins to the priority encoder. The encoder's 2-bit output (destination) is fanned out to the floor traversal FSM and the comparator. The FSM's 2-bit output (current floor) is also fanned out to the comparator and the 7-segment driver. Finally, the comparator's three outputs are connected to the status indicator pins.

## 4. Working Procedure

The following steps describe the operation of the circuit from the user's perspective.

1. **Initial State:** The circuit powers on, and the FSM is in its default state,  $Q_1Q_0 = 00$  (Ground Floor). The 7-segment display shows '0'. The comparator sees Current (0) = Destination (0), so the 'At Floor' indicator is active.

2. **Floor Call:** The user presses the input button for Floor 3. The **Priority Encoder** detects this input and outputs the binary code '11' ( $F_1F_0$ ) as the new destination.
3. **Status Update:** Instantly, the **Comparator** receives the new destination '11' (F) and compares it to the current floor '00' (Q). Since  $F > Q$ , the 'Moving Up' output pin becomes active.
4. **Sequential Traversal:**
  - **Pulse 1:** On the next clock pulse, the **floor traversal FSM** evaluates its logic. With  $Q=00$  and  $F=11$ , the logic (e.g.,  $J_0, K_0, J_1, K_1$ ) calculates the next state as '01'. The flip-flops update, and the FSM's output becomes  $Q_1Q_0 = 01$  (Floor 1).
  - **Display Update:** The **7-Segment Display** immediately updates to show '1'. The comparator now sees  $F=11$  and  $Q=01$ , so the 'Moving Up' status remains active.
  - **Pulse 2:** On the second clock pulse, the FSM re-evaluates. With  $Q=01$  and  $F=11$ , the logic calculates the next state as '10'. The FSM output becomes  $Q_1Q_0 = 10$  (Floor 2). The display updates to '2'.
  - **Pulse 3:** On the third clock pulse, the FSM re-evaluates. With  $Q=10$  and  $F=11$ , the logic calculates the next state as '11'. The FSM output becomes  $Q_1Q_0 = 11$  (Floor 3). The display updates to '3'.
5. **Reaching Destination:**
  - The FSM's output is now  $Q=11$ . The **Comparator** sees that the current floor '11' equals the destination floor '11'.
  - The 'Moving Up' pin deactivates, and the 'At Floor' pin becomes active.
  - The FSM's internal logic, seeing  $Q=F$ , now calculates the next state to be the same as the present state. The elevator will remain at Floor 3 ( $Q=11$ ) on all subsequent clock pulses until a new destination is selected.

## 5. Circuit Schematics

### 5.1 Main Circuit

The main circuit diagram below shows the interconnection of all primary components and sub-circuits.

*Figure 1: The main elevator controller schematic, showing the priority encoder (left), FSM and 7-segment driver (center), and comparator with status outputs (right).*

### 5.2 Sub-Circuit: Floor Traversal FSM

This circuit implements the core sequential logic using J-K flip-flops and combinational gates to generate the next state.

*Figure 2: The floor traversal FSM, showing the two J-K flip-flops (right) and the combinational logic (left) that implements the derived state equations.*

### 5.3 Sub-Circuit: 7-Segment Driver

This circuit is a standard BCD-to-7-segment decoder built from basic logic gates.

*Figure 3: The 7-segment driver sub-circuit, a custom decoder that converts a 4-bit BCD input into 7 output signals for the display.*

## 6. Applications

While this project is a simulation, the design principles are fundamental to many real-world systems:

- **Elevator Controllers:** The primary application is as the logic core for a simple, real-world elevator. The design can be expanded with more states (floors) and more complex input logic.
- **Sequential Control Systems:** The FSM design is universally applicable. It can be adapted to control any process requiring ordered steps, such as traffic light systems, automatic car washes, vending machine dispensing logic, or industrial assembly line controls.
- **Digital Counters:** The circuit is a form of specialized synchronous counter. This concept can be used in any device that needs to count up or down to a specific target value.
- **Educational Tool:** This project is an excellent academic tool for teaching digital electronics. It clearly demonstrates the complete design process of a sequential circuit, from the initial state diagram and K-maps to a final, functional hardware simulation.

## 7. Advantages

This hardware-based digital design offers several benefits over a software-based (microcontroller) solution:

- **High Speed and Reliability:** The circuit operates at the speed of the logic gates. Its behavior is deterministic, predictable, and not subject to software bugs, memory leaks, or operating system crashes, making it highly reliable for a critical system like an elevator.
- **Real-Time Feedback:** The system provides instant feedback. The 7-segment display and status lights reflect the FSM's state in real-time with no processing lag.
- **Modular Design:** The project is well-structured into logical sub-circuits (FSM, Display Driver). This modularity makes the system easier to design, test, and debug, as each part can be verified independently.
- **Efficient Input Handling:** The use of a **Priority Encoder** is a simple and efficient hardware solution for managing multiple inputs (floor calls) and selecting the highest-priority request.

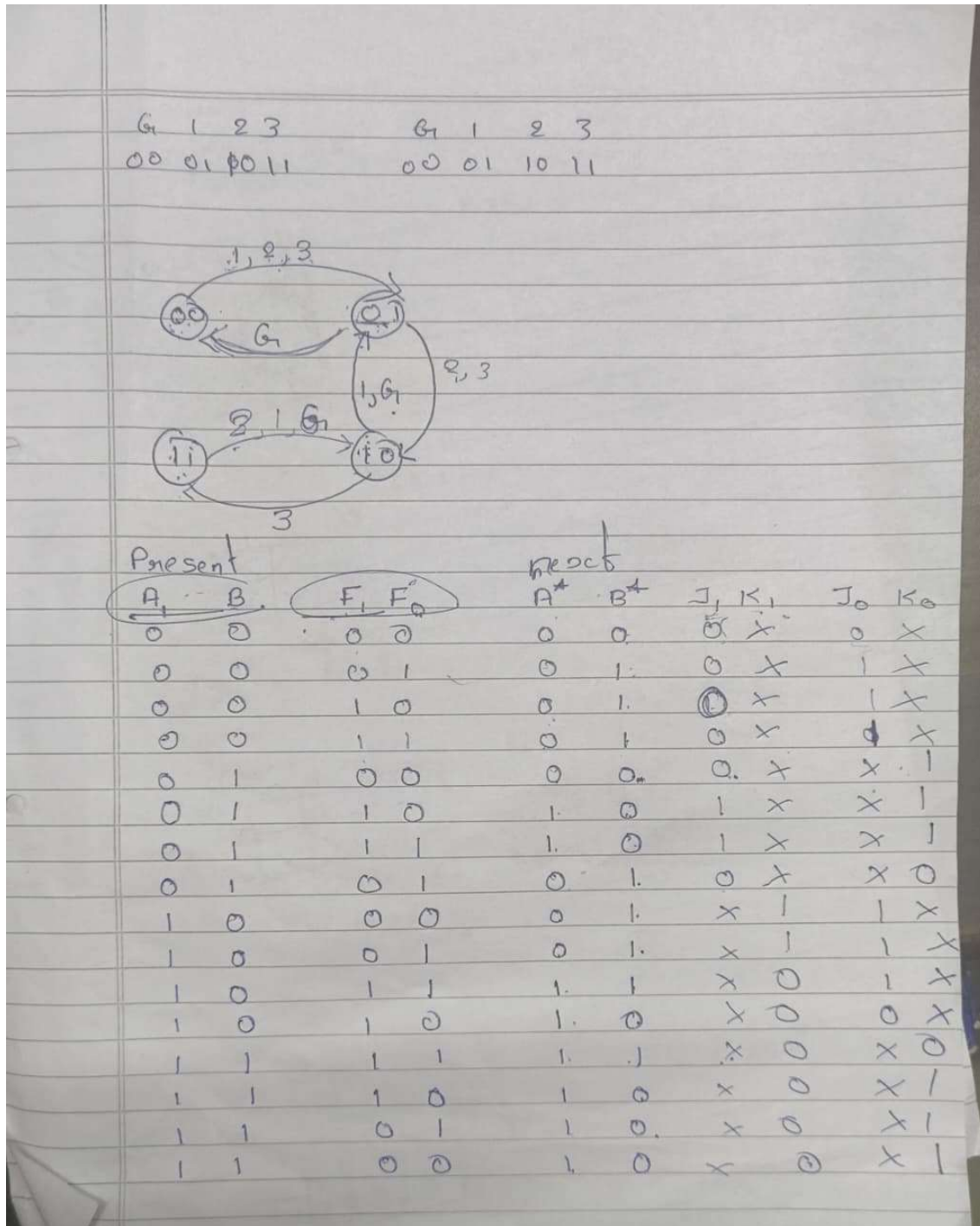
## 8. Conclusion

This project successfully demonstrates the design and simulation of a 4-floor elevator controller using fundamental digital logic principles. By translating a conceptual state diagram and state tables into a functional sequential circuit, the project provides a tangible and practical application of

finite state machine (FSM) theory.

The integration of combinational components, such as the priority encoder and comparator, with sequential memory elements (the J-K flip-flops) results in a complete and robust system. The circuit correctly manages user inputs, processes state transitions, and generates clear, appropriate outputs for both display and status.

The project fully achieves its objective by simulating the required floor-by-floor traversal and providing unambiguous visual feedback. It serves as a comprehensive example of how combinational and sequential logic are combined to build complex, useful, and reliable digital systems.



$$J_1 = \cancel{F_0 F_1} + B F_1$$

		F <sub>0</sub>			
		00	01	11	10
A \ B	00	0			1
	01				1
	11	X	X	X	X
	10	X	X	X	X

$$B \cdot F_1$$

$$K_1 = B' F_1'$$

		F <sub>0</sub>			
		00	01	11	10
A \ B	00	X	X	X	X
	01	X	X	X	X
	11				
	10				

		F <sub>0</sub>			
		00	01	11	10
A \ B	00		1	1	1
	01	X	X	X	X
	11	X	X	X	X
	10	X	1	1	

$$F_0 + AB'F_1 + A'F_1$$

$$F_0 + A'F_1 + A'F_1$$

$$F_0 + A \oplus F_1$$

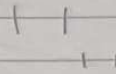
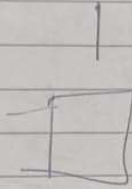
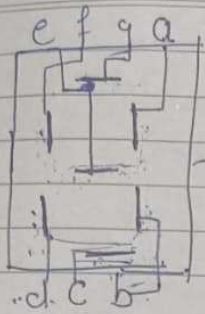
		F <sub>0</sub>			
		00	01	11	10
A \ B	00	X	X	X	X
	01	1		1	1
	11	1	1		1
	10	X	X	X	X

$$F_0' + F_1 A' + A F_1'$$

$$F_0' + F_1 \oplus A$$

$J_0$	
$J_1$	$4 \times 2$
$J_2$	Perim
$J_3$	Circ

$F_0$	$F_1$	$F_2$	$GND$		
$J_0$	$J_1$	$J_2$	$J_3$	$\gamma_1$	$\gamma_0$
0	0	0	0	x	x
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1



	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	0	1	1
1	0	0	0	1	1	1	0	0	0	0	0
2	0	0	1	0	1	0	1	1	1	0	1
3	0	0	1	1	1	1	1	0	1	0	1
4	0	1	0	0	1	1	0	0	1	1	0
5	0	1	0	1	0	1	1	0	1	1	1
6	0	1	1	0	0	1	1	1	1	1	1
7	0	1	1	1	1	1	0	0	0	0	1
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	1	1	1

a

