Dhirubhai Ambani Institute of Information and Communication Technology

Embedded Hardware Design (EL203)

Prof. Biswajit Mishra & Prof. Yash Agrawal

Lab Group 6

Student Id	Name	Contribution	Percentage
202201462	Lunagariya Darpan	Code, Hardware	18
202201463	Popatiya Harsh	State Diagram, Report	16
202201467	Patel Dharmi	State Diagram, Report	16
202201469	Tandel Divya	Code, Report	16
202201470	Asodariya Priyank	Code, Report	16
202201472	Prajapati Kaushik	Code, Hardware	18

Problem Statement:

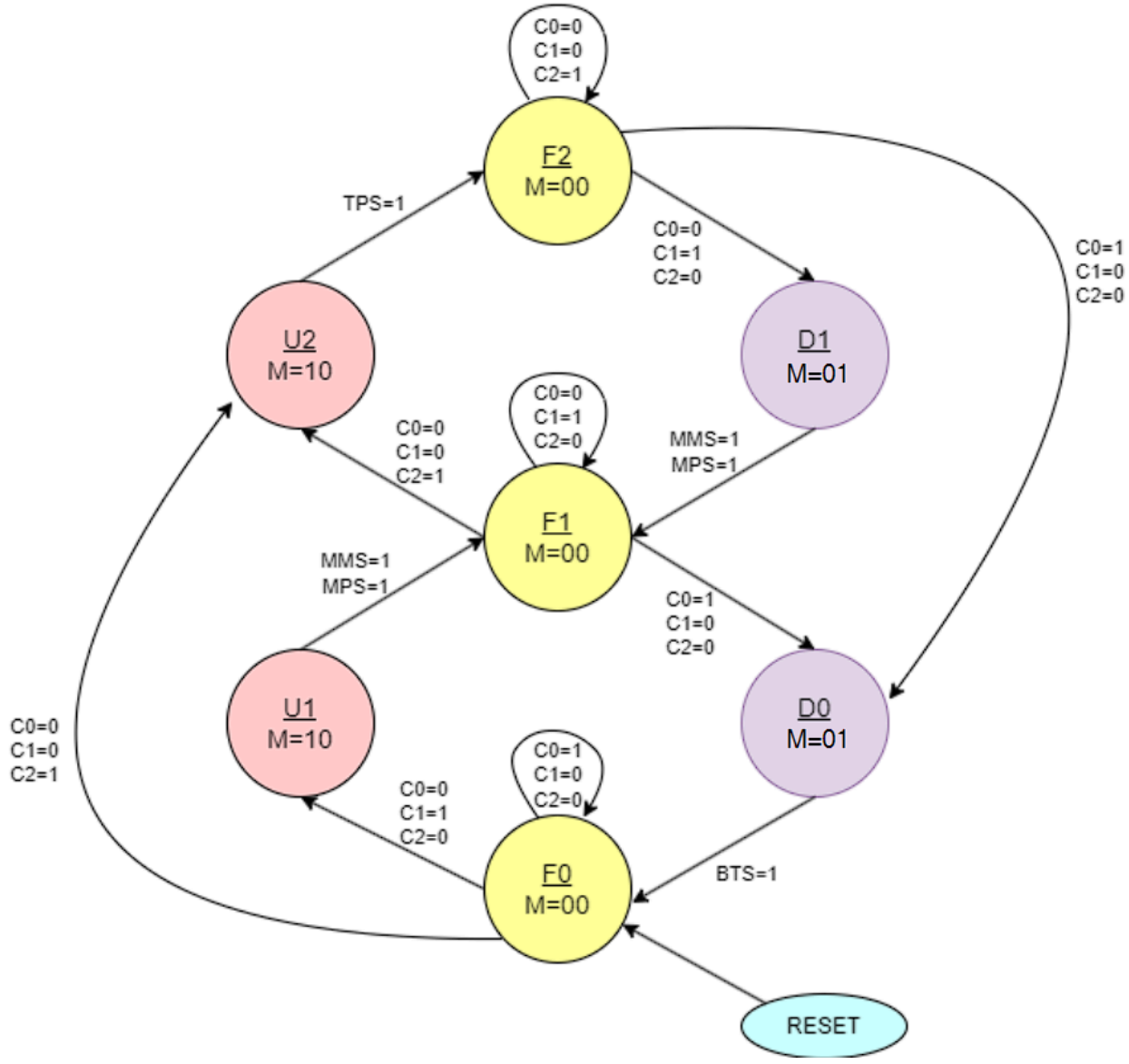
We are given a building model having three floors and we are asked to design an elevator system. The building consists of three floors and each floor has a call button to call the elevator to that particular floor. Additionally, on each floor, there is an LED that illuminates when the call button is pressed and indicates that the elevator is arriving at that floor. The elevator system includes four IR sensors, one at the ground floor, one at the top floor, and two positioned at intermediate levels between the ground and first floor, and between the first and second floor respectively. The motion of the elevator is governed by a DC motor, which is controlled by the L293DA motor driver. Also, the elevator system has a global reset button. If pressed, the lift comes to the ground floor regardless of the call buttons pressed.

Component List:

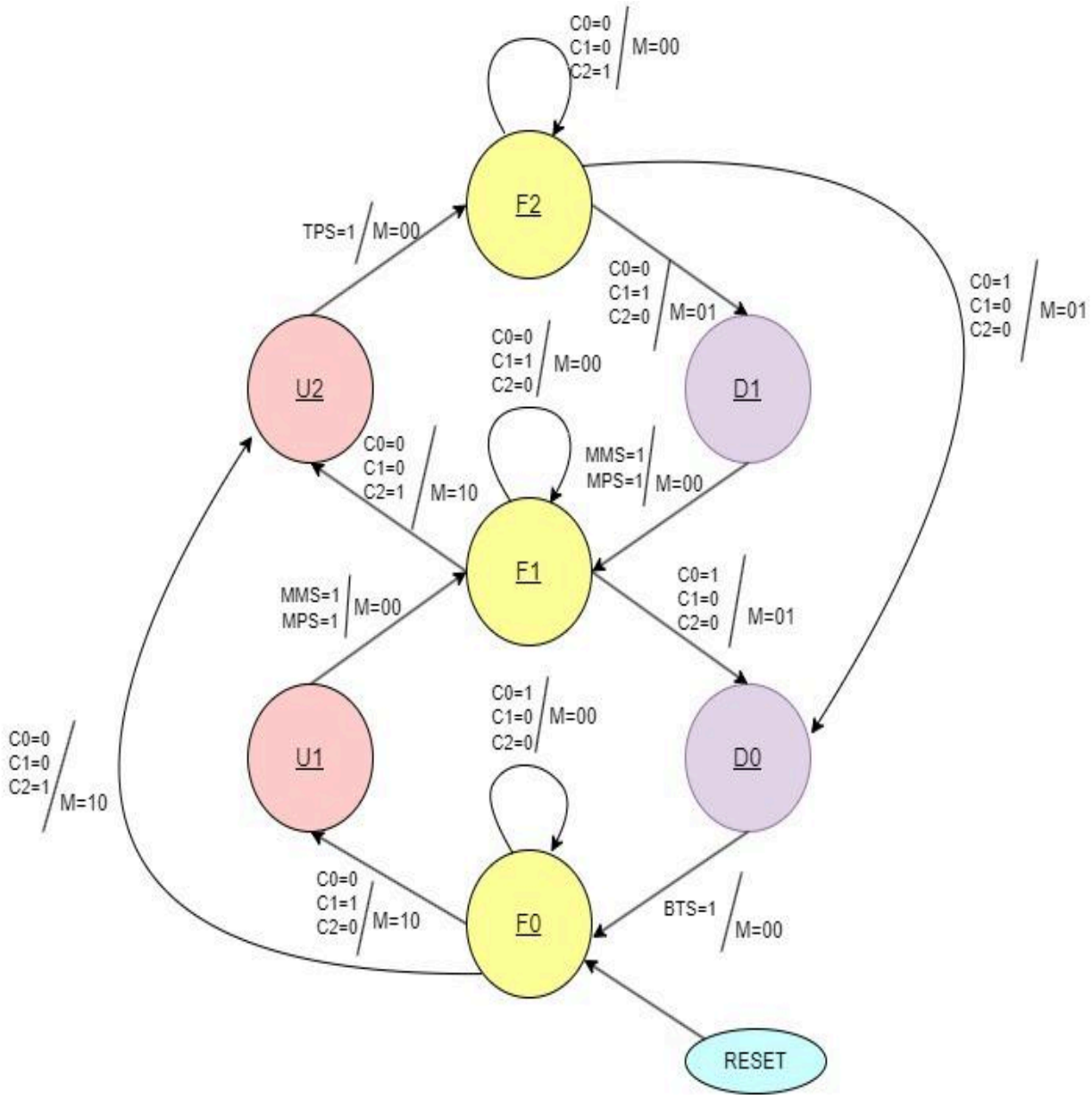
1. Artix-7 FPGA kit
2. Elevator model made
3. 12 V DC motor
4. Four IR sensor module
5. Three LEDs
6. L293DA motor driver board
7. 12V DC power supply
8. Jumper Wires

State Diagram:

(a) Moore Machine:



(b) Mealy Machine:



Description for the state diagram:

The elevator system is designed for a building with three floors, equipped with an elevator capable of vertical traversal. The system employs seven distinct states, each represented by a unique code: F0, F1, F2, U1, U2, D1, and D0.

- F0, F1, and F2: Indicate that the elevator is stationary at the ground, first, or second floor, respectively.
- U1 and U2: Signify the elevator's upward motion towards the first or second floor.
- D1 and D0: Represent the elevator's downward motion towards the first or ground floor.

The states are encoded as follows:

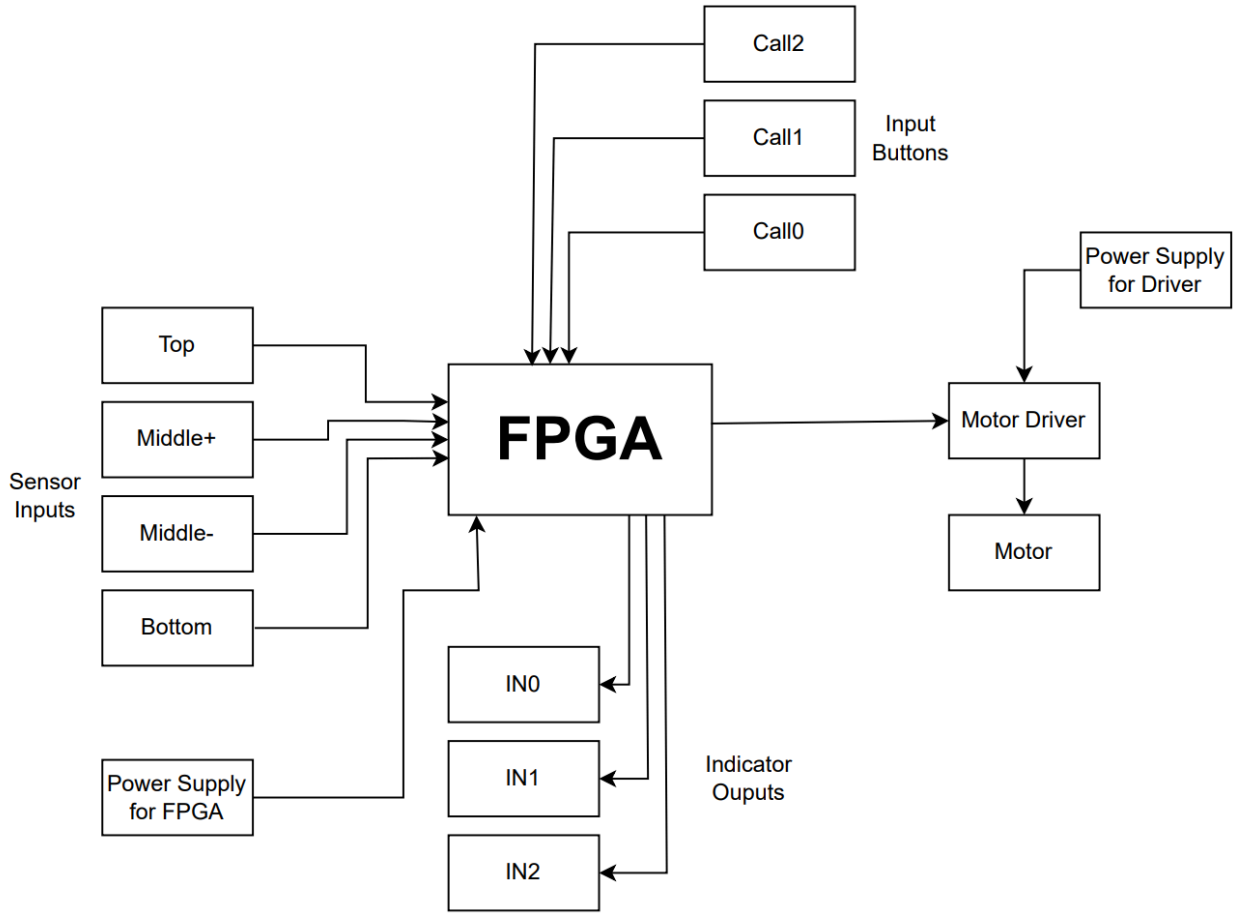
- F0 = 3'b000 (Ground Floor - Initial State)
- F1 = 3'b001 (First Floor)
- F2 = 3'b010 (Second Floor)
- U1 = 3'b011 (Moving Up to First Floor)
- U2 = 3'b100 (Moving Up to Second Floor)
- D1 = 3'b101 (Moving Down to First Floor)
- D0 = 3'b110 (Moving Down to Ground Floor)

The system incorporates three call buttons, named CALL0, CALL1, and CALL2, which act as IR sensors positioned at the respective floors to summon the elevator to the ground, first, and second floors, respectively.

Initiating from the ground floor (state F0), pressing the CALL2 button signals a request for the elevator to ascend to the second floor. Subsequently, LED2 illuminates to indicate the lift's destination. The system transitions to state U2, prompting motor activation to initiate upward movement. The elevator ascends until the top sensor detects its presence at the second floor. Upon arrival, LED2 ceases flashing, and the motors halt, awaiting further commands.

Similarly, if a user presses the CALL1 button, signaling the elevator to halt at the first floor, LED1 lights up to indicate the lift's approach. Motors engage to facilitate upward motion, transitioning the system to state U1. Upon detection by both middle plus and minus sensors, indicating the elevator's arrival at the first floor, the motors deactivate, and LED1 extinguishes.
This process continues iteratively, responding to calls from any floor, executing the appropriate transitions, and ensuring efficient elevator operation within the building.

Basic Block Diagram for FPGA:



VHDL Code Snippets:

elevator_mealy.v : The process “elevator_mealy” is created in which there are three call buttons, four input sensors, and the reset button.

How are the call buttons handled?

```
F0:
    if(CALL0 == 0 && CALL1 == 1 && CALL2 == 0) begin
        MOTOR_DIR = 2'b10;
        next_state = U1;
        LED0 = 1;
        LED1 = 0;
        LED2 = 0;
    end
    else if(CALL0 == 0 && CALL1 == 0 && CALL2 == 1) begin
        MOTOR_DIR = 2'b10;
        next_state = U2;
        LED0 = 0;
        LED1 = 1;
        LED2 = 0;
    end
    else begin
        MOTOR_DIR = 2'b00;
        next_state = F0;
    end
F1:
    if(CALL0 == 1 && CALL1 == 0 && CALL2 == 0) begin
        MOTOR_DIR = 2'b01;
        next_state = D0;
        LED0 = 1;
        LED1 = 0;
        LED2 = 0;
    end
    else if(CALL0 == 0 && CALL1 == 0 && CALL2 == 1) begin
        MOTOR_DIR = 2'b10;
        next_state = U2;
        LED0 = 0;
        LED1 = 0;
        LED2 = 1;
    end
    else begin
        MOTOR_DIR = 2'b00;
        next_state = F1;
    end
F2:
    if(CALL0 == 1 && CALL1 == 0 && CALL2 == 0) begin
        MOTOR_DIR = 2'b01;
        next_state = D0;
        LED0 = 1;
        LED1 = 0;
        LED2 = 0;
    end
    else if(CALL0 == 0 && CALL1 == 1 && CALL2 == 0) begin
        MOTOR_DIR = 2'b01;
        next_state = D1;
        LED0 = 0;
        LED1 = 1;
        LED2 = 0;
    end
    else begin
        MOTOR_DIR = 2'b00;
        next_state = F2;
    end
```

How are the sensor inputs handled?

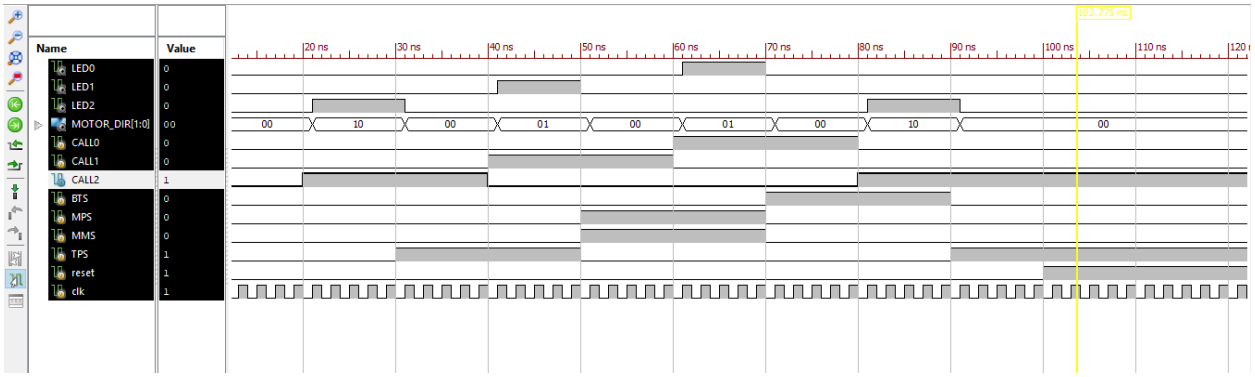
```
U1:
    begin
        if(MPS == 1 && MMS == 1) begin
            MOTOR_DIR = 2'b00;
            next_state = F1;
            LED0 = 0;
            LED1 = 0;
            LED2 = 0;
        end
        else begin
            MOTOR_DIR = 2'b10;
            next_state = U1;
        end
    end
U2:
    begin
        if(TPS == 1) begin
            MOTOR_DIR = 2'b00;
            next_state = F2;
            LED0 = 0;
            LED1 = 0;
            LED2 = 0;
        end
        else begin
            MOTOR_DIR = 2'b10;
            next_state = U2;
        end
    end
D1:
    begin
        if(MPS == 1 && MMS == 1) begin
            MOTOR_DIR = 2'b00;
            next_state = F1;
            LED0 = 0;
            LED1 = 0;
            LED2 = 0;
        end
        else begin
            MOTOR_DIR = 2'b01;
            next_state = D1;
        end
    end
D0:
    begin
        MOTOR_DIR = 2'b01;
        if(BTS == 1) begin
            MOTOR_DIR = 2'b00;
            next_state = F0;
            LED0 = 0;
            LED1 = 0;
            LED2 = 0;
        end
        else begin
            MOTOR_DIR = 2'b01;
            next_state = D0;
        end
    end
```

elev.ucf

In this file, we have mapped the FPGA to our code. Here, the call buttons are connected to input pins. The input pins for sensor input, push buttons, and reset button were defined, whereas output pins for the 3 LEDs, and 2 pins for motor output were defined. All the pins to connect the jumper wires are defined on the 4 digital J-ports. The lift reset button is taken as the push button provided on the FPGA.

Test Simulation Outputs:

(a) Mealy Machine



(b) Moore Machine