EL203

ELEVATOR CONTROLLER
GROUP - 5 (LAB GROUP - 1)

NAME	ID	CONTRIBUTION
202201040	MAVDIYA SUJAL P.	Report, Hardware
202201041	DHRITI M GOENKA	Code, Hardware
202201048	JOSHI SNEH VIPULKUMAR	Report, Hardware
202201049	ADITYA SABLE	Code, Hardware
202201053	SANIA PATEL	Code, Hardware
202201054	ASHUTOSH SINGARWAL	Report, Hardware

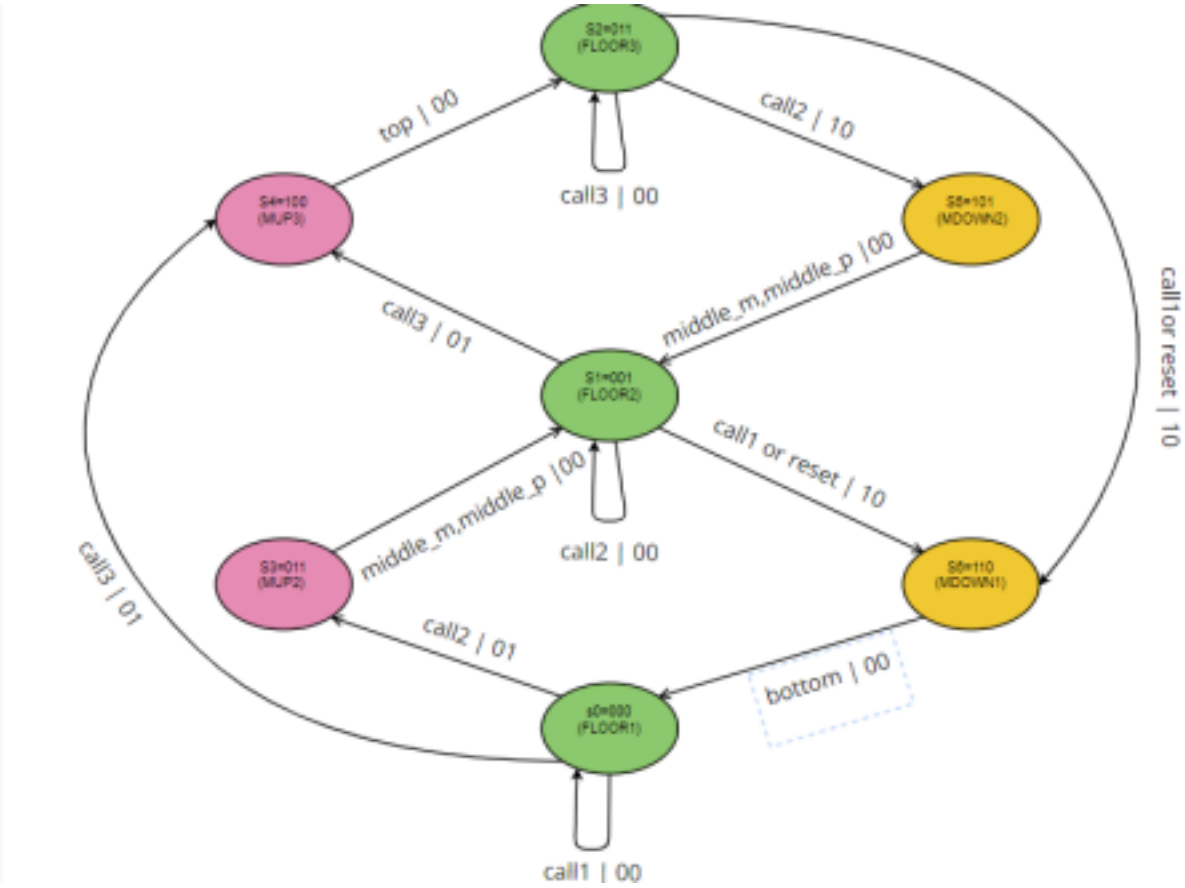
Professors :
Prof. Bishwjit Mishra
Prof. Yash Agarwal

PROBLEM STATEMENT:

The lift is a customized structure representing a lift shaft to which are fixed three “call” buttons and associated “lift coming” indicators, plus a rack-and-pinion gear system and motor whereby a simple platform representing the lift cage can be driven up and down. The position of platform is detected by four more switches- one each for “top”, “middle-plus”, “middle-minus” and “bottom”. The two switches which detect the platform in the middle position are operated by the rack cage, rather than an isolated point on it – only when both ”middleswitches” are closed is the platform properly aligned. The top and bottom position sensing arrangements

don"t need this refinement because any “up” movement with the “top” switch active is illegal, as is “down” movement with the “bottom” switch active. The model can be powered using +5V and +12V supplies using a 7 pin DIN connector from an outlet. Simple electronics on a printed circuit board in the model base convert and condition input and outputs, such that each switch provides a logic „HI" when open and „LO" when made (but see A.3 below). The platform motor is driven in response to the UP/DOWN “direction” control line when the “/ENABLE” control line is taken „LO". An integrated motor drive IC is used to provide appropriate connection of the +12V and 0V supplies to the motor, and to limit the current. Crude speed control is possible by Pulse Width modulation of the enable control line. Each floor indicator is driven by an active LO-control line.

STATE DIAGRAM:



ENCODING:

- Seven states are depicted in state diagram by the letters FLOOR1, FLOOR2, FLOOR3, MUP2, MUP3, MDOWN2, and MDOWN1.

State FLOOR1 = The lift is stationary on floor 1. It represents state 000.
State FLOOR2 = The lift is stationary on floor 2. It represents state 001.
State FLOOR3 = The lift is stationary on floor 3. It represents state 010.
State MUP2: The lift is moving up to Floor 2. It represents state 011.

State MUP3: The lift is moving up to Floor 3. It represents state 100.
State MDOWN2: The lift is moving down to Floor 2. It represents state 101.
State MDOWN1: The lift is moving down to Floor 1. It represents state 110.
State M = 00: Represents that the motor is off.
Output state M = 01: Represents the upward motion of the elevator.
Output state M = 10: Represents the downward motion of the elevator.

- bottom, middle_p, middle_m,top stand for bottom, middle plus, middle minus, and top sensor inputs, respectively. The lift is considered to be in a traveling condition until a sensor is off, indicating it has reached its destination. bottom: Bottom Floor Sensor - Activation indicates the elevator has reached the bottom middle_p: Middle Plus Sensor - Activates to align the lift's ceiling for a middle floor. middle_m: Middle Minus Sensor - Activates to align the lift's floor for a middle floor. top: Top Floor Sensor - Activation indicates the elevator has reached the top floor.

- The call buttons for floor 1, 2 and 3 are represented by the push buttons call1, call2 and call3 respectively
- Pressing any button activates the elevator, except if it's already on that floor.
- middle_m and middle_p "ON" means the elevator is stationary on floor 2. ❖ Pressing "RESET" moves the elevator to first floor from any position.

Verilog Code Snippets:

```
always @(posedge clk or posedge reset) begin
    if (reset)
        call1 = 1'b1;
    else
        curr_state <= n_state;
end
```

If the reset button is pressed, the call1 button will be activated; otherwise, the next state will be assigned.

```
always @(curr_state) begin
    case (curr_state)
        FLOOR1, FLOOR2, FLOOR3: out_motor = 2'b00;  // motor off
        MUP2, MUP3: out_motor = 2'b01;    // motor upwards
        MDOWN1, MDOWN2: out_motor = 2'b10;    //motor downwards
        default: out_motor = 2'b00;     // motor off
    endcase
end
```

This code assigns the motor state based on the current state:

- If the current state indicates that the elevator is stationary on floors 0, 1, or 2, the motor state is set to 00, indicating that the motor is off.
- If the current state indicates that the elevator is moving upward or downward, the motor state is set to 01 or 10, respectively, indicating that the motor is on.

```
always @(*) begin
    case (curr_state)
        FLOOR1: begin
            if (call2) n_state = MUP2;
            else if (call3) n_state = MUP3;
            else n_state = FLOOR1;
        end
```

This code segment pertains to operations on floor 1:

- When call1 is pressed, the next state will transition to moving up to floor 2 (MUP2).
- When call2 is pressed, the next state will transition to moving up to floor 3 (MUP3).

Similar conditions apply for floor 2:

- If call1 is pressed, the next state will transition to moving down to floor 1 (MDOWN1).
- If call2 is pressed, the next state will transition to moving up to floor 3 (MUP3).

Regarding floor 3:

- Upon pressing call1, the next state will transition to moving down to floor 1 (MDOWN1).
- If call2 is pressed, the next state will transition to moving down to floor 2

(MDOWN2).

```
MUP2: begin
    if (middle_m && middle_p) n_state = FLOOR2;
    else n_state = MUP2;
end
MUP3: begin
    if (top) n_state = FLOOR3;
    else n_state = MUP3;
end
```

This code excerpt handles the moving states MUP2 and MUP3.

- When both middle_m and middle_p are activated, the elevator remains stationary on floor 2.
- If top is activated, it indicates that the elevator has reached floor 3.

```
MDOWN2: begin
    if (middle_m && middle_p) n_state = FLOOR2;
    else n_state = MDOWN2;
end
MDOWN1: begin
    if (bottom) n_state = FLOOR1;
    else n_state = MDOWN1;
end
default: n_state = FLOOR1;
```

This code excerpt handles the moving states MDOWN2 and MDOWN1.

- When both middle_m and middle_p are activated, the elevator remains stationary on floor 2.
- If bottom is activated, it indicates that the elevator has reached floor 1.

UCF FILE:

In this file, we have assigned the IO pins to be used by all inputs and outputs. The input pins for sensors are on the Ja and Jb blocks. The call buttons are controlled by the push buttons on the FPGA board. The motor outputs are connected to the JC block.The lift reset button is taken as the reset provided on the FPGA. All the input-output ports

are defined to have an IOSTANDARD value equal to LVCMOS33.