

# **VLSI DESIGN LAB. ASSIGNMENT -5**

**SAURAV GUPTA**

**ROLL No. – 153070046**

**Dept.- Electrical Engineering**

**Specialisation – Microelectronics (TA)**

**Batch - 2015-17**

**Teacher concerned – VIRENDRA SINGH**

## **PROBLEM STATEMENT:**

Design a coordinated lift group control system (LGCS), which controls 3 lifts to carry passengers in a 6-floor building (i.e, there are 6 floors to visit and 3 lifts).

Each floor has two hall call buttons (except the ground and top floors where they have only one), an up button to request transport to a higher floor and a down button to request transport to a lower floor. These buttons illuminate when pressed. The illumination is cancelled when lift visits the floor and is either moving in the desired direction or has no outstanding requests. In the latter case, if both floor buttons are pressed, only one should be cancelled.

Each lift has a set of buttons (car call button), one for each floor. These illuminate when pressed and cause the lift to visit the corresponding floor. The illumination is cancelled when the corresponding floor is visited by the lift.

When lift has no requests to service, it should remain at its final destination with its doors closed and await further requests.

The lift control system has a set of sensors to detect the floor it is visiting which is communicating lift asynchronously.

Each lift has an emergency button that, when pressed, causes a warning signal to be sent to the site manager. The lift is then deemed "out of service". Each lift has a mechanism to cancel its "out of service" status.

The LGCS should satisfy the following conditions:

- A upward traveling lift should not change its direction at any floor when it has passengers wishing to go to higher floor, and vice-versa for downward traveling lift.

- Any request (hall call, and car call) should eventually be serviced

Your design should either minimize the average waiting time or the energy consumption. Please write it clearly which optimization your system is using.

Average waiting time (AWT) is the time until the service lift arrives at the floor after a passenger presses a hall call button. AWT is the average of all waiting times in a unit time.

Run count (RC) is the number of lift moves in a unit time and is used to estimate the power consumption of the system since most energy is consumed by starting or stopping the lift.

Write a synthesizable behavioral description of the above circuit in Verilog.

**Sol:**

Each Lift has its individual controller whose FSM is given in Fig.1. There is a master controller of these 3 Lift controller which assigns the Hall Call Up or Hall Call Down request to any of these three lift depending upon the Algorithm given below.

### **Algorithm of Master Controller**

1. Find out which Lift is nearest to the present floor from which Hall Call Up/Down request has come.
2. If emergency button of any Lift is pressed, that Lift will not be assigned any request.
3. If Hall Call Up button is pressed from Ground Floor then
  - assign the request to Lift 1 if it is nearest and it is moving down or it is in stop state.
  - assign the request to Lift 3 if it is nearest and it is moving down or it is in stop state.
  - else assign to Lift 2
4. If Hall Call Up/Down button is pressed from First Floor then
  - assign the request to Lift 1 if it is nearest and it is moving down from 5th,4th,3rd,2nd floor/Stop state in any of these floor or moving up from Ground Floor/in stop state.
  - assign the request to Lift 3 if it is nearest and it is moving down from 5th,4th,3rd,2nd floor/Stop state in any of these floor or moving up from Ground Floor/in stop state.
  - else assign to Lift 2.
5. If Hall Call Up/Down button is pressed from Second Floor then
  - assign the request to Lift 1 if it is nearest and it is moving down from 5th,4th,3rd floor/Stop state in any of these floor or moving up from Ground or 1st Floor/Stop state in any of these floor.

- assign the request to Lift 3 if it is nearest and it is moving down from 5th,4th,3rd floor/Stop state in any of these floor or moving up from Ground or 1st Floor/Stop state in any of these floor.
- else assign to Lift 2.

6. If Hall Call Up/Down button is pressed from Third Floor then

- assign the request to Lift 1 if it is nearest and it is moving down from 5th,4th floor/Stop state in any of these floor or moving up from Ground,1st,2nd Floor/Stop state in any of these floor.
- assign the request to Lift 3 if it is nearest and it is moving down from 5th,4th floor/Stop state in any of these floor or moving up from Ground,1st,2nd Floor/Stop state in any of these floor.
- else assign to Lift 2.

7. If Hall Call Up/Down button is pressed from Fourth Floor then

- assign the request to Lift 1 if it is nearest and it is moving down from 5th floor/in stop state or moving up from Ground,1st,2nd,3rd Floor/Stop state in any of these floor.
- assign the request to Lift 3 if it is nearest and it is moving down from 5th floor/in stop state or moving up from Ground,1st,2nd,3rd Floor/Stop state in any of these floor.
- else assign to Lift 2.

8. If Hall Call Down button is pressed from Fifth Floor then

- assign the request to Lift 1 if it is nearest and it is moving up/it is in stop state.
- assign the request to Lift 3 if it is nearest and it is moving up/ it is in stop state.
- else assign to Lift 2.

**Note:** Master controller only takes care of assignment of Hall Call request to any of the three lift, servicing the request is the headache of individual Lift controller. Also the Car call requests are handled by individual Lifts.

## Controller FSM of single LIFT

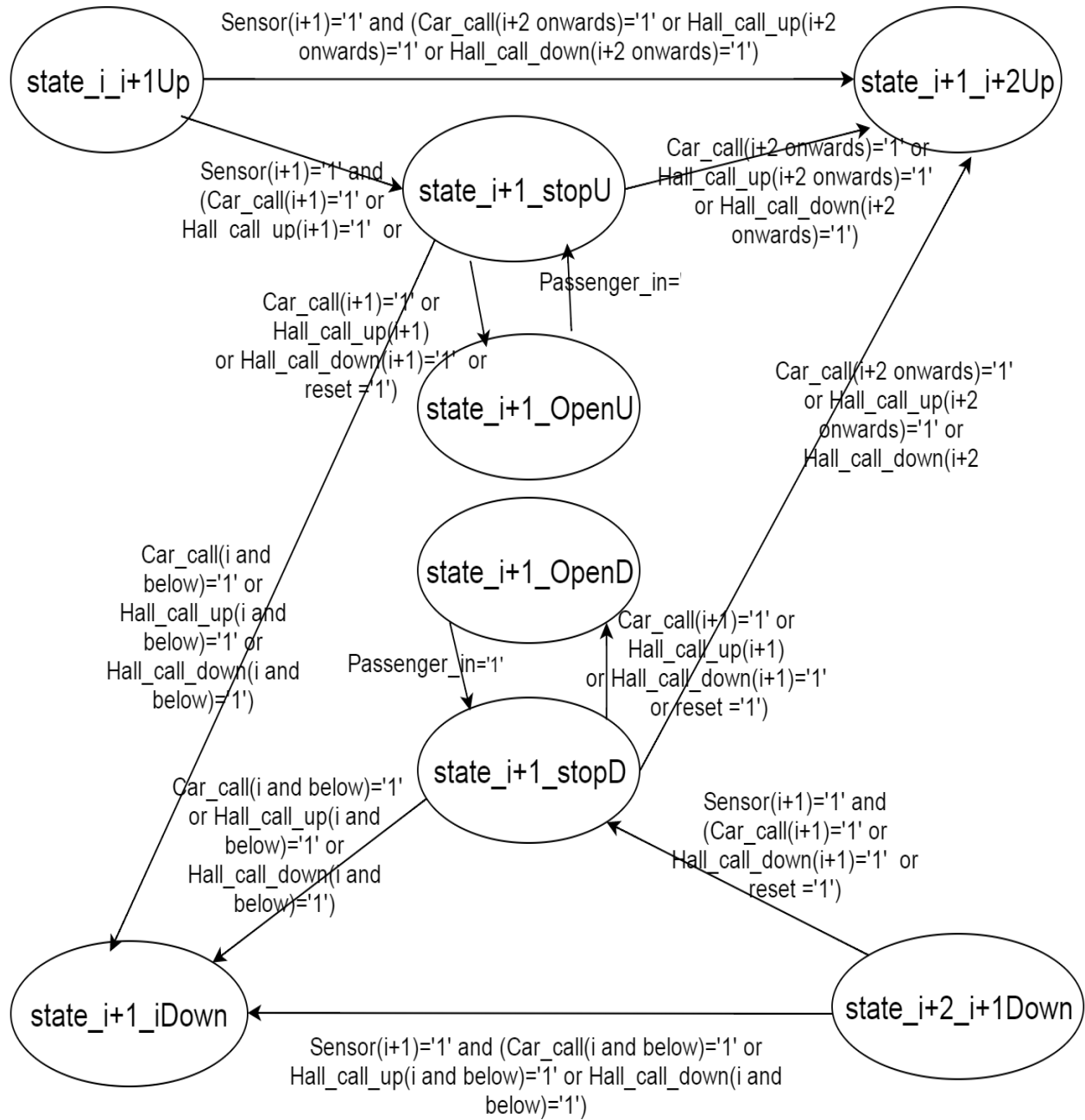
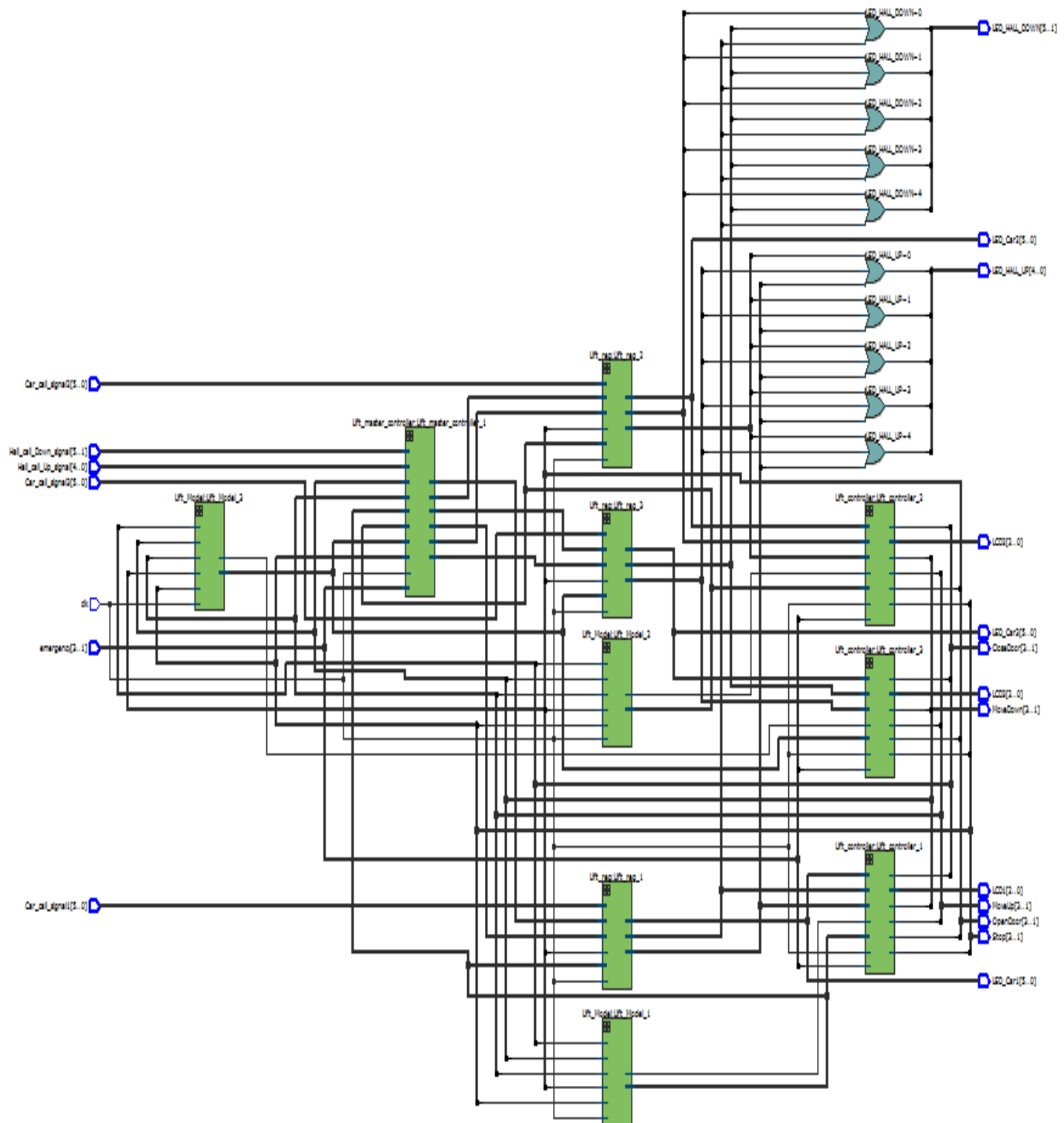


Fig. 1 Controller FSM of single Lift

## Circuit Diagram



*Fig. 2 Circuit Diagram*

## 1. Lift overall

## 2. Lift overall testbench

## Individual Components

## 3. Lift Model

- Lift
- counter

## 4. Lift Register

## 5. Lift Controller

## 6. Master Lift Controller

## CODE

### 1. Lift overall

```
module
Lift_overall(clk,emergency,Car_call_signal1,Car_call_signal2,Car_call_
signal3,Hall_call_Up_signal,Hall_call_Down_signal,MoveUp,MoveDown,Open
Door,CloseDoor,Stop,LCD1,LCD2,LCD3,LED_HALL_UP,LED_HALL_DOWN,LED_Car1,
LED_Car2,LED_Car3);
    input clk;
    input [3:1] emergency;
    input [5:0] Car_call_signal1,Car_call_signal2,Car_call_signal3;
    input [4:0] Hall_call_Up_signal;
    input [5:1] Hall_call_Down_signal;

    output [3:1] MoveUp,MoveDown,OpenDoor,CloseDoor,Stop;
    output [4:0] LED_HALL_UP;
    output [5:1] LED_HALL_DOWN;
    output [5:0] LED_Car1,LED_Car2,LED_Car3;
    output [2:0] LCD1,LCD2,LCD3;

    wire [5:0] Car_call_temp1,Car_call_temp2,Car_call_temp3;
```

```

        wire [4:0]
Hall_call_Up_temp1,Hall_call_Up_temp2,Hall_call_Up_temp3;
        wire [5:1]
Hall_call_Down_temp1,Hall_call_Down_temp2,Hall_call_Down_temp3;
        wire [3:1] OpenDoor_temp,
CloseDoor_temp,Stop_temp,MoveDown_temp,MoveUp_temp,Passenger_in_temp;
        wire [5:0] Sensor1, Sensor2, Sensor3;
        wire
one_is_nearest_temp,two_is_nearest_temp,three_is_nearest_temp;
        wire [4:0]
Hall_call_Up_signalout1_temp,Hall_call_Up_signalout2_temp,Hall_call_Up
_signalout3_temp;
        wire [5:1]
Hall_call_Down_signalout1_temp,Hall_call_Down_signalout2_temp,
Hall_call_Down_signalout3_temp;
        wire [2:0] LCD_temp1,LCD_temp2,LCD_temp3;

        Lift_master_controller
Lift_master_controller_1(clk,emergency,Sensor1,Sensor2,Sensor3,MoveUp_
temp,MoveDown_temp,Stop_temp,Hall_call_Up_signal,Hall_call_Down_signal
,Hall_call_Up_signalout1_temp,Hall_call_Up_signalout2_temp,Hall_call_U
p_signalout3_temp,Hall_call_Down_signalout1_temp,Hall_call_Down_signal
out2_temp, Hall_call_Down_signalout3_temp);
        Lift_controller Lift_controller_1(clk,
emergency[1],Passenger_in_temp[1],Sensor1,Car_call_temp1,Hall_call_Up_
temp1,Hall_call_Down_temp1,MoveUp_temp[1],MoveDown_temp[1],OpenDoor_te
mp[1],CloseDoor_temp[1],Stop_temp[1],LCD_temp1);
        Lift_controller Lift_controller_2(clk,
emergency[2],Passenger_in_temp[2],Sensor2,Car_call_temp2,Hall_call_Up_
temp2,Hall_call_Down_temp2,MoveUp_temp[2],MoveDown_temp[2],OpenDoor_te
mp[2],CloseDoor_temp[2],Stop_temp[2],LCD_temp2);
        Lift_controller Lift_controller_3(clk,
emergency[3],Passenger_in_temp[3],Sensor3,Car_call_temp3,Hall_call_Up_
temp3,Hall_call_Down_temp3,MoveUp_temp[3],MoveDown_temp[3],OpenDoor_te
mp[3],CloseDoor_temp[3],Stop_temp[3],LCD_temp3);
        Lift_reg
Lift_reg_1(clk,Car_call_signal1,Hall_call_Up_signalout1_temp,Hall_call
_Down_signalout1_temp,Sensor1,OpenDoor_temp[1],Hall_call_Up_temp1,Hall
_call_Down_temp1,Car_call_temp1);
        Lift_reg
Lift_reg_2(clk,Car_call_signal2,Hall_call_Up_signalout2_temp,Hall_call
_Down_signalout2_temp,Sensor2,OpenDoor_temp[2],Hall_call_Up_temp2,Hall
_call_Down_temp2,Car_call_temp2);
        Lift_reg
Lift_reg_3(clk,Car_call_signal3,Hall_call_Up_signalout3_temp,Hall_call
_Down_signalout3_temp,Sensor3,OpenDoor_temp[3],Hall_call_Up_temp3,Hall
_call_Down_temp3,Car_call_temp3);
        Lift_Model
Lift_Model_1(clk,MoveUp_temp[1],MoveDown_temp[1],OpenDoor_temp[1],Clos
eDoor_temp[1],Stop_temp[1],Sensor1,Passenger_in_temp[1]);

```



```

    Lift_Model
    Lift_Model_2(clk,MoveUp_temp[2],MoveDown_temp[2],OpenDoor_temp[2],CloseDoor_temp[2],Stop_temp[2],Sensor2,Passenger_in_temp[2]);
    Lift_Model
    Lift_Model_3(clk,MoveUp_temp[3],MoveDown_temp[3],OpenDoor_temp[3],CloseDoor_temp[3],Stop_temp[3],Sensor3,Passenger_in_temp[3]);
    assign OpenDoor = OpenDoor_temp;
    assign CloseDoor = CloseDoor_temp;
    assign MoveUp = MoveUp_temp;
    assign MoveDown = MoveDown_temp;
    assign Stop = Stop_temp;
    assign LED_HALL_UP = Hall_call_Up_temp1| Hall_call_Up_temp2|
Hall_call_Up_temp3;
    assign LED_HALL_DOWN =
Hall_call_Down_temp1|Hall_call_Down_temp2|Hall_call_Down_temp3;
    assign LED_Car1 = Car_call_temp1;
    assign LED_Car2 = Car_call_temp2;
    assign LED_Car3 = Car_call_temp3;
    assign LCD1 = LCD_temp1;
    assign LCD2 = LCD_temp2;
    assign LCD3 = LCD_temp3;
endmodule

```

## 2. Lift overall testbench

```

`timescale 1ms/100us
module Lift_overall_tb;
    wire [3:1] MoveUp,MoveDown,OpenDoor,CloseDoor,Stop;
    wire [4:0] LED_HALL_UP;
    wire [5:1] LED_HALL_DOWN;
    wire [5:0] LED_Car1,LED_Car2,LED_Car3;
    wire [2:0] LCD1,LCD2,LCD3;
    reg clk;
    reg [3:1] emergency;
    reg [5:0] Car_call_signal1=6'b000000;
    reg [5:0] Car_call_signal2=6'b000000;
    reg [5:0] Car_call_signal3=6'b000000;
    reg [4:0] Hall_call_Up_signal=6'b000000;
    reg [5:1] Hall_call_Down_signal=6'b000000;

    Lift_overall
    DUT(clk,emergency,Car_call_signal1,Car_call_signal2,Car_call_signal3,Hall_call_Up_signal,Hall_call_Down_signal,MoveUp,MoveDown,OpenDoor,CloseDoor,Stop,LCD1,LCD2,LCD3,LED_HALL_UP,LED_HALL_DOWN,LED_Car1,LED_Car2,LED_Car3);

    initial
    begin
        $dumpfile("run.vcd");
    end

```

```

    $dumpvars(0,Lift_overall_tb);
    clk=0;
    emergency=0;
    #100
    Hall_call_Up_signal[2]=1; #60 Hall_call_Up_signal[2]=0;
    #2000
    Car_call_signal2[4]=1; #60 Car_call_signal2[4]=0;
    #800
    Hall_call_Down_signal[5]=1;#60 Hall_call_Down_signal[5]=0;
    #3000
    Hall_call_Up_signal[1]=1;#60 Hall_call_Up_signal[1]=0;
    #700
    Car_call_signal2[0]=1;#60 Car_call_signal2[0]=0;
    #400
    Car_call_signal1[3]=1;#60 Car_call_signal1[3]=0;
    #100
    Hall_call_Up_signal[0]=1;#60 Hall_call_Up_signal[0]=0;
    #300
    Car_call_signal3[3]=1;#60 Car_call_signal3[3]=0;
    #10000 $finish;
end

always
begin
    clk = #10 ~clk;
end

always
begin
    #12000 emergency[1]=1;
end

endmodule

```

### 3. Lift Model

```

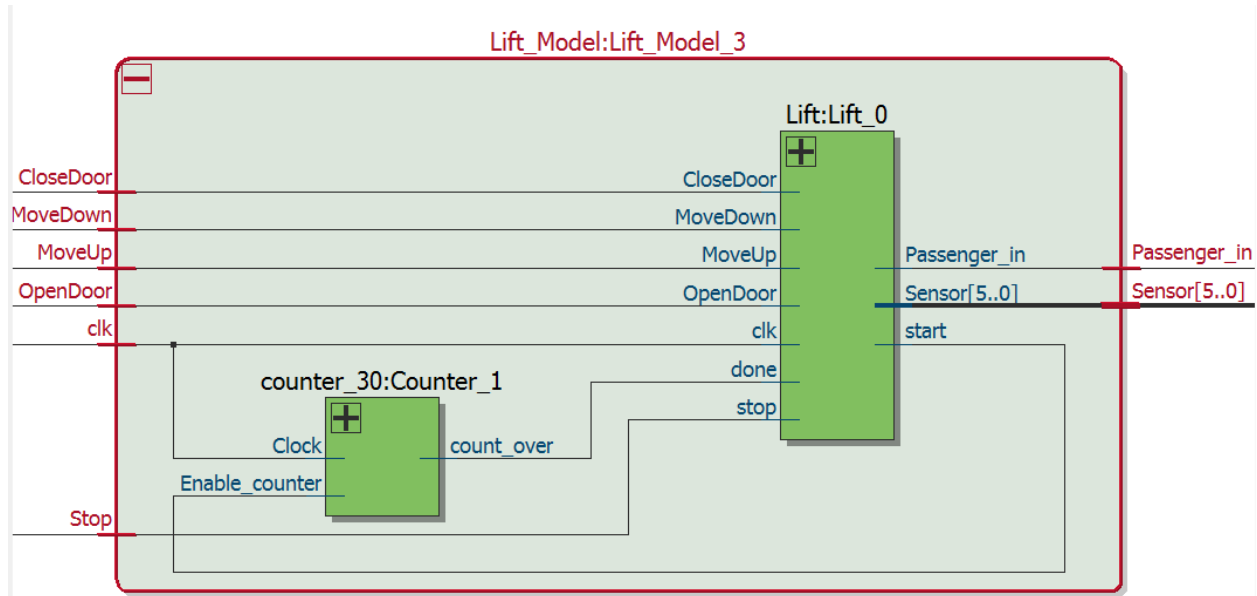
module
Lift_Model (clk,MoveUp,MoveDown,OpenDoor,CloseDoor,Stop,Sensor,Passenge
r_in);
    input clk,MoveUp,MoveDown,OpenDoor,CloseDoor,Stop;
    output [5:0] Sensor;
    output Passenger_in;

    wire start;
    wire done;

    Lift
    Lift_0 (clk,MoveUp,MoveDown,OpenDoor,CloseDoor,Stop,start,done,Sensor,P
assenger_in);
    counter_30 Counter_1 (clk,start,done);

```

```
endmodule
```



### 3.a Lift

```
module
Lift (clk, MoveUp, MoveDown, OpenDoor, CloseDoor, stop, start, done, Sensor, Pas
senger_in);
    input clk, MoveUp, MoveDown, OpenDoor, CloseDoor, stop, done;
    output [5:0] Sensor;
    output Passenger_in, start;

    reg Passenger_in_temp=1'b0, start_temp=1'b0;
    reg [5:0] Sensor_temp = 6'b000001;

    always @(posedge clk)
    begin
        if (clk==1'b1) begin
            if (MoveUp ==1'b1 && stop==1'b0) begin
                start_temp<=1;
                if (done==1'b1) begin
                    start_temp<=0;
                    Sensor_temp<= Sensor_temp<<1;
                end
            end

            else if (MoveDown ==1'b1 && stop==1'b0) begin
                start_temp<=1;
                if (done==1'b1) begin
                    start_temp<=0;
                    Sensor_temp<= Sensor_temp>>1;
                end
            end
        end
    end
```

```

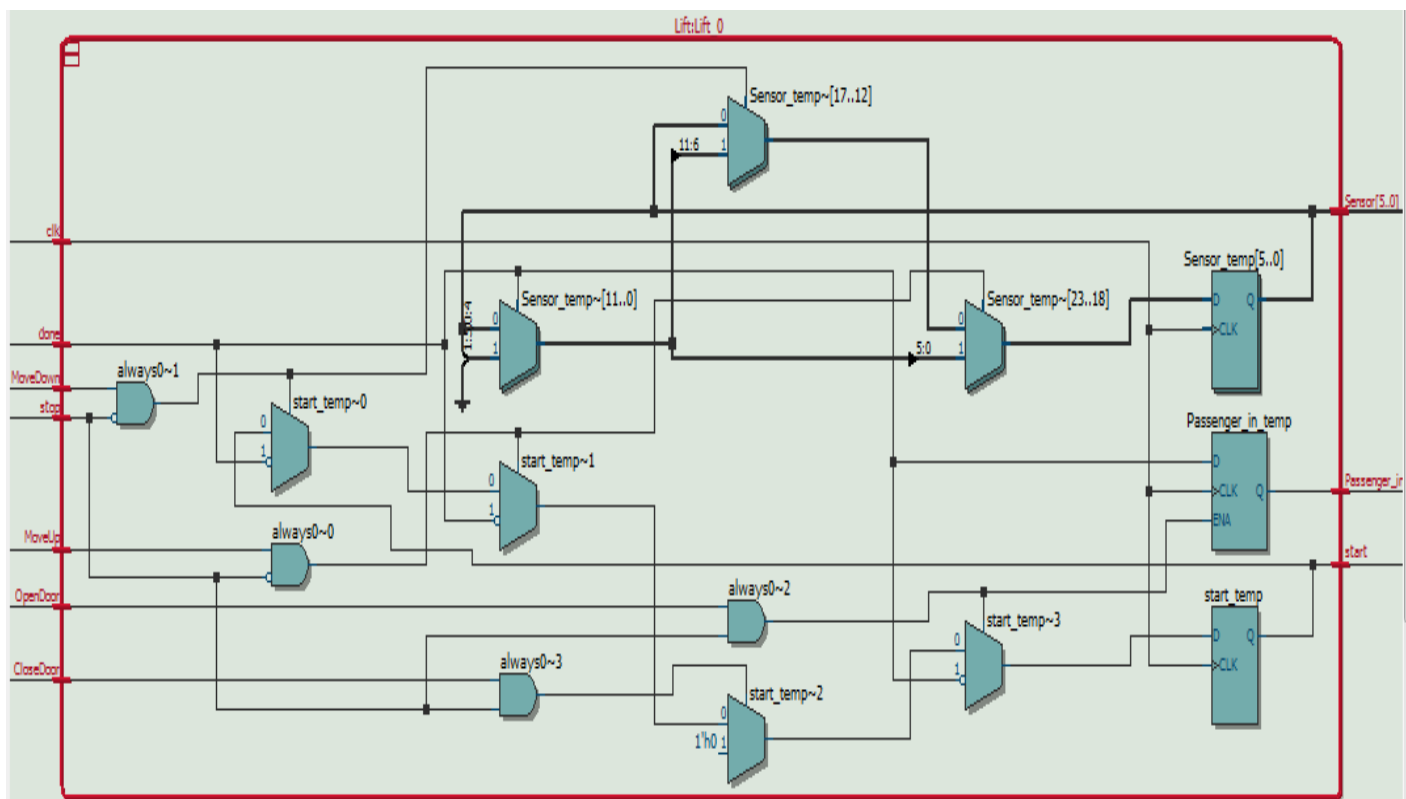
        if(OpenDoor==1'b1 && stop==1'b1) begin
            start_temp<=1; Passenger_in_temp<=0;
            if(done==1'b1) begin
                start_temp<=0;
                Passenger_in_temp<=1;
            end
        end
    end

    else if(CloseDoor==1'b1 && stop==1'b1) begin
        start_temp<=0;
    end

end
end
assign Sensor = Sensor_temp;
assign Passenger_in = Passenger_in_temp;
assign start = start_temp;

endmodule

```



### 3 b. Counter

```

module counter_30(Clock,Enable_counter,count_over);
    input Clock,Enable_counter;
    output reg count_over;

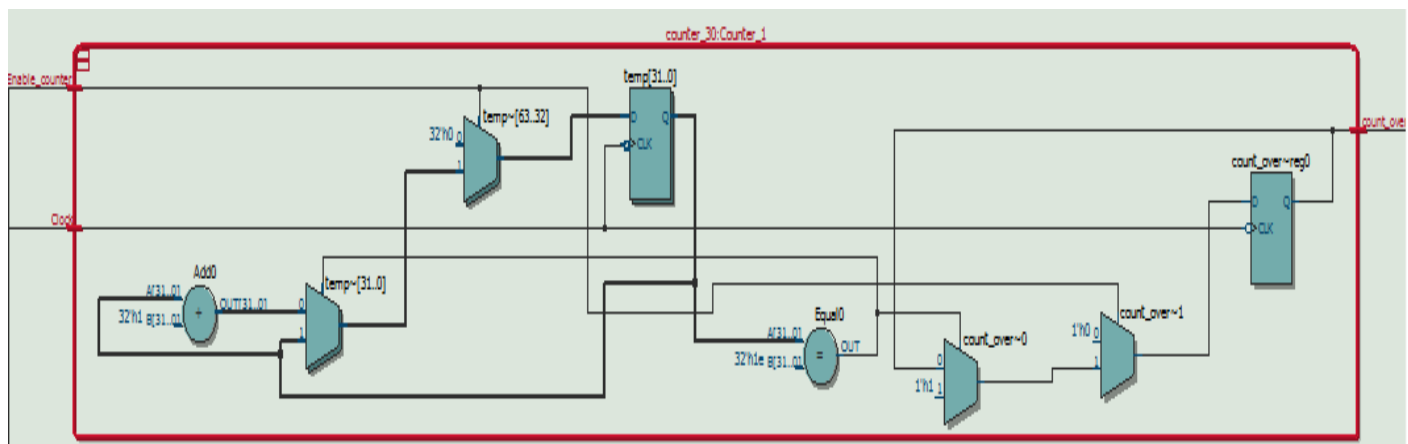
```

```

integer temp =0;

always @(negedge Clock)
begin
    if (Enable_counter ==1'b1) begin
        if (Clock ==1'b0) begin
            if (temp==30) begin
                count_over <= 1'b1 ;
            end
            else begin
                temp <= temp+1;
            end
        end
    end
    else begin
        temp<=0;
        count_over<=1'b0;
    end
end
endmodule

```



## 4. Lift Register

```

module Lift_reg
(clk,Car_call_signal,Hall_call_Up_signal,Hall_call_Down_signal,Sensor,
OpenDoor,Hall_call_Up,Hall_call_Down,Car_call);
    input clk,OpenDoor;
    input [5:0] Car_call_signal,Sensor;
    input [4:0] Hall_call_Up_signal;
    input [5:1] Hall_call_Down_signal;

    output [4:0] Hall_call_Up;
    output [5:1] Hall_call_Down;
    output [5:0] Car_call;

    reg [5:1] Hall_call_Down_temp =5'b000000;

```

```

reg [4:0] Hall_call_Up_temp=5'b00000;
reg [5:0] Car_call_temp=6'b000000;

always @ (posedge clk)
begin
    if (clk) begin
        Hall_call_Up_temp <= Hall_call_Up_signal |
Hall_call_Up_temp;
        Hall_call_Down_temp <= Hall_call_Down_signal |
Hall_call_Down_temp;
        Car_call_temp <= Car_call_signal | Car_call_temp;
        if(OpenDoor == 1'b1) begin
            if(Sensor[0] == 1'b1) begin
                Hall_call_Up_temp[0]<=1'b0;
                Car_call_temp[0]<=1'b0; end
            else if(Sensor[1] == 1'b1) begin
                Hall_call_Up_temp[1]<=1'b0;
                Hall_call_Down_temp[1]<=1'b0;
                Car_call_temp[1]<=1'b0; end
            else if(Sensor[2] == 1'b1) begin
                Hall_call_Up_temp[2]<=1'b0;
                Hall_call_Down_temp[2]<=1'b0;
                Car_call_temp[2]<=1'b0; end
            else if(Sensor[3] == 1'b1) begin
                Hall_call_Up_temp[3]<=1'b0;
                Hall_call_Down_temp[3]<=1'b0;
                Car_call_temp[3]<=1'b0; end
            else if(Sensor[4] == 1'b1) begin
                Hall_call_Up_temp[4]<=1'b0;
                Hall_call_Down_temp[4]<=1'b0;
                Car_call_temp[4]<=1'b0; end
            else if(Sensor[5] == 1'b1) begin
                Hall_call_Down_temp[5]<=1'b0;
                Car_call_temp[5]<=1'b0; end
        end
    end
end
assign Hall_call_Down = Hall_call_Down_temp;
assign Hall_call_Up = Hall_call_Up_temp;
assign Car_call = Car_call_temp;
endmodule

```

## 5. Lift Controller

```

module
Lift_controller(clk,emergency,Passenger_in,Sensor,Car_call,Hall_call_U
p,Hall_call_Down,MoveUp,MoveDown,OpenDoor,CloseDoor,Stop,LCD);
    input clk,emergency,Passenger_in;
    input [5:0] Sensor,Car_call;
    input [4:0] Hall_call_Up;

```

```

    input [5:1] Hall_call_Down;
    output reg
MoveUp=1'b0,MoveDown=1'b0,OpenDoor=1'b0,CloseDoor=1'b0,Stop=1'b0;
    output reg [2:0] LCD=3'b000;

    parameter s_0_stop = 0,s_0_open
=1,s_01_Up=2,s_10_Down=3,s_1_stopU =4,s_1_stopD =5,s_1_openU
=6,s_1_openD=7,s_12_Up =8,s_21_Down =9,s_2_stopU =10,s_2_stopD
=11,s_2_openU=12,s_2_openD =13,s_23_Up =14,s_32_Down =15,s_3_stopU
=16,s_3_stopD =17,s_3_openU =18,s_3_openD =19,s_34_Up =20,s_43_Down
=21,s_4_stopU =22,s_4_stopD =23,s_4_openU =24,s_4_openD= 25,s_45_Up
=26,s_54_Down =27,s_5_stop =28,s_5_open =29;
    reg[0:4] state_signal, next_state_var;

    always
@ (emergency,Sensor,Hall_call_Up,Hall_call_Down,Car_call,Passenger_in,s
tate_signal)
    begin
        case (state_signal)
            s_0_stop: begin
                if (emergency ==1'b1 || Hall_call_Up[0]==1'b1 ||
Car_call[0]==1'b1)
                    next_state_var = s_0_open;
                else if (Hall_call_Down[1]==1'b1 ||
Hall_call_Up[1]==1'b1 || Car_call[1]==1'b1 || Hall_call_Down[2]==1'b1
|| Hall_call_Up[2]==1'b1 || Car_call[2]==1'b1 ||
Hall_call_Down[3]==1'b1 || Hall_call_Up[3]==1'b1 || Car_call[3]==1'b1
|| Hall_call_Down[4]==1'b1 || Hall_call_Up[4]==1'b1 ||
Car_call[4]==1'b1 || Hall_call_Down[5]==1'b1 || Car_call[5]==1'b1)
                    next_state_var = s_01_Up;
                end

            s_0_open: begin
                if (emergency ==1'b1)
                    next_state_var = s_0_open;
                else
                    if (Passenger_in ==1'b1)
                        next_state_var = s_0_stop;
                    end

            s_01_Up: begin
                if (Sensor[1]==1'b1 && (Car_call[1]==1'b1 ||
Hall_call_Up[1]==1'b1 || emergency==1'b1))
                    next_state_var = s_1_stopU;
                else if (Sensor[1]==1'b1 && (Hall_call_Down[2]==1'b1 ||
Hall_call_Up[2]==1'b1 || Car_call[2]==1'b1 ||
Hall_call_Down[3]==1'b1 || Hall_call_Up[3]==1'b1 || Car_call[3]==1'b1
|| Hall_call_Down[4]==1'b1 || Hall_call_Up[4]==1'b1 ||
Car_call[4]==1'b1 || Hall_call_Down[5]==1'b1 || Car_call[5]==1'b1 ))
                    next_state_var = s_12_Up;
                else if (Sensor[1]==1'b1)
                    next_state_var = s_1_stopU;
            end
        endcase
    end

```

```

        end
    s_10_Down: begin
        if(Sensor[0]==1'b1)
            next_state_var = s_0_stop;
        end

    s_1_stopU: begin
        if(emergency ==1'b1 || Car_call[1]==1'b1 ||
Hall_call_Down[1]==1'b1 || Hall_call_Up[1]==1'b1)
            next_state_var = s_1_openU;
        else if (Hall_call_Down[2]==1'b1 ||
Hall_call_Up[2]==1'b1 || Car_call[2]==1'b1 ||
Hall_call_Down[3]==1'b1|| Hall_call_Up[3]==1'b1 || Car_call[3]==1'b1
|| Hall_call_Down[4]==1'b1|| Hall_call_Up[4]==1'b1 ||
Car_call[4]==1'b1|| Hall_call_Down[5]==1'b1 || Car_call[5]==1'b1)
            next_state_var = s_12_Up;
        else if (Hall_call_Up[0]==1'b1)
            next_state_var = s_10_Down;
        end

    s_1_stopD: begin
        if(emergency ==1'b1 || Car_call[1]==1'b1 ||
Hall_call_Down[1]==1'b1 || Hall_call_Up[1]==1'b1)
            next_state_var = s_1_openD;
        else if (Hall_call_Up[0]==1'b1)
            next_state_var = s_10_Down;
        else if (Hall_call_Down[2]==1'b1 ||
Hall_call_Up[2]==1'b1 || Car_call[2]==1'b1 ||
Hall_call_Down[3]==1'b1|| Hall_call_Up[3]==1'b1 || Car_call[3]==1'b1
|| Hall_call_Down[4]==1'b1|| Hall_call_Up[4]==1'b1 ||
Car_call[4]==1'b1|| Hall_call_Down[5]==1'b1 || Car_call[5]==1'b1)
            next_state_var = s_12_Up;
        end

    s_1_openU: begin
        if(emergency ==1'b1)
            next_state_var = s_1_openU;
        else
            if(Passenger_in ==1'b1)
                next_state_var = s_1_stopU;
            end

    s_1_openD: begin
        if(emergency ==1'b1)
            next_state_var = s_1_openD;
        else
            if(Passenger_in ==1'b1)
                next_state_var = s_1_stopD;
            end

    s_12_Up: begin

```



```

        if(Sensor[2]==1'b1 && (Hall_call_Up[2]==1'b1 ||
Car_call[2]==1'b1 || emergency==1'b1))
            next_state_var = s_2_stopU;
        else if(Sensor[2]==1'b1 && (Hall_call_Down[3]==1'b1||
Hall_call_Up[3]==1'b1 || Car_call[3]==1'b1 ||
Hall_call_Down[4]==1'b1|| Hall_call_Up[4]==1'b1 || Car_call[4]==1'b1
|| Hall_call_Down[5]==1'b1 || Car_call[5]==1'b1 ))
            next_state_var = s_23_Up;
        else if(Sensor[2]==1'b1)
            next_state_var = s_2_stopU;
        end

s_21_Down: begin
    if(Sensor[1]==1'b1 && (Hall_call_Down[1]==1'b1 ||
Car_call[1]==1'b1 || emergency==1'b1))
        next_state_var = s_1_stopD;
    else if(Sensor[1]==1'b1 && (Hall_call_Up[0]==1'b1 ||
Car_call[0]==1'b1))
        next_state_var = s_10_Down;
    else if(Sensor[1]==1'b1)
        next_state_var = s_1_stopD;
    end

s_2_stopU: begin
    if(emergency ==1'b1 || Hall_call_Down[2]==1'b1 ||
Hall_call_Up[2]==1'b1 || Car_call[2]==1'b1)
        next_state_var = s_2_openU;
    else if (Hall_call_Down[3]==1'b1||
Hall_call_Up[3]==1'b1 || Car_call[3]==1'b1 ||
Hall_call_Down[4]==1'b1|| Hall_call_Up[4]==1'b1 || Car_call[4]==1'b1
|| Hall_call_Down[5]==1'b1 || Car_call[5]==1'b1)
        next_state_var = s_23_Up;
    else if (Hall_call_Up[0]==1'b1 || Car_call[0]==1'b1 ||
Hall_call_Down[1]==1'b1|| Hall_call_Up[1]==1'b1 || Car_call[1]==1'b1)
        next_state_var = s_21_Down;
    end

s_2_stopD: begin
    if(emergency ==1'b1 || Hall_call_Down[2]==1'b1 ||
Hall_call_Up[2]==1'b1 || Car_call[2]==1'b1)
        next_state_var = s_2_openD;
    else if (Hall_call_Up[0]==1'b1 || Car_call[0]==1'b1 ||
Hall_call_Down[1]==1'b1|| Hall_call_Up[1]==1'b1 || Car_call[1]==1'b1)
        next_state_var = s_21_Down;
    else if (Hall_call_Down[3]==1'b1||
Hall_call_Up[3]==1'b1 || Car_call[3]==1'b1 ||
Hall_call_Down[4]==1'b1|| Hall_call_Up[4]==1'b1 || Car_call[4]==1'b1
|| Hall_call_Down[5]==1'b1 || Car_call[5]==1'b1)
        next_state_var = s_23_Up;
    end

s_2_openU: begin

```

```

        if(emergency ==1'b1)
            next_state_var = s_2_openU;
        else
            if(Passenger_in ==1'b1)
                next_state_var = s_2_stopU;
            end
        end

s_2_openD: begin
    if(emergency ==1'b1)
        next_state_var = s_2_openD;
    else
        if(Passenger_in ==1'b1)
            next_state_var = s_2_stopD;
        end
    end

s_23_Up: begin
    if(Sensor[3]==1'b1 && (Hall_call_Up[3]==1'b1 ||
Car_call[3]==1'b1 || emergency==1'b1))
        next_state_var = s_3_stopU;
    else if(Sensor[3]==1'b1 && (Hall_call_Down[4]==1'b1||
Hall_call_Up[4]==1'b1 || Car_call[4]==1'b1 || Hall_call_Down[5]==1'b1
|| Car_call[5]==1'b1 ))
        next_state_var = s_34_Up;
    else if(Sensor[3]==1'b1)
        next_state_var = s_3_stopU;
    end

s_32_Down: begin
    if(Sensor[2]==1'b1 && (Hall_call_Down[2]==1'b1 ||
Car_call[2]==1'b1 || emergency==1'b1))
        next_state_var = s_2_stopD;
    else if(Sensor[2]==1'b1 &&(Hall_call_Up[0]==1'b1 ||
Car_call[0]==1'b1 || Hall_call_Down[1]==1'b1|| Hall_call_Up[1]==1'b1
|| Car_call[1]==1'b1))
        next_state_var = s_21_Down;
    else if(Sensor[2]==1'b1)
        next_state_var = s_2_stopD;
    end

s_3_stopU: begin
    if(emergency ==1'b1 || Hall_call_Down[3]==1'b1 ||
Hall_call_Up[3]==1'b1 || Car_call[3]==1'b1)
        next_state_var = s_3_openU;
    else if (Hall_call_Down[4]==1'b1||
Hall_call_Up[4]==1'b1 || Car_call[4]==1'b1 || Hall_call_Down[5]==1'b1
|| Car_call[5]==1'b1)
        next_state_var = s_34_Up;
    else if (Hall_call_Up[0]==1'b1 || Car_call[0]==1'b1 ||
Hall_call_Down[1]==1'b1 || Hall_call_Up[1]==1'b1 || Car_call[1]==1'b1
|| Hall_call_Down[2]==1'b1 || Hall_call_Up[2]==1'b1 ||
Car_call[2]==1'b1)
        next_state_var = s_32_Down;

```

```

        end

        s_3_stopD: begin
            if(emergency ==1'b1 || Hall_call_Down[3]==1'b1 ||
Hall_call_Up[3]==1'b1 || Car_call[3]==1'b1)
                next_state_var = s_3_openU;
            else if (Hall_call_Up[0]==1'b1 || Car_call[0]==1'b1 ||
Hall_call_Down[1]==1'b1 || Hall_call_Up[1]==1'b1 || Car_call[1]==1'b1
|| Hall_call_Down[2]==1'b1 || Hall_call_Up[2]==1'b1 ||
Car_call[2]==1'b1)
                next_state_var = s_32_Down;
            else if (Hall_call_Down[4]==1'b1||
Hall_call_Up[4]==1'b1 || Car_call[4]==1'b1 || Hall_call_Down[5]==1'b1
|| Car_call[5]==1'b1)
                next_state_var = s_34_Up;
            end

        s_3_openU: begin
            if(emergency ==1'b1)
                next_state_var = s_3_openU;
            else
                if(Passenger_in ==1'b1)
                    next_state_var = s_3_stopU;
                end

        s_3_openD: begin
            if(emergency ==1'b1)
                next_state_var = s_3_openD;
            else
                if(Passenger_in ==1'b1)
                    next_state_var = s_3_stopD;
                end

        s_34_Up: begin
            if(Sensor[4]==1'b1 && (Hall_call_Up[4]==1'b1 ||
Car_call[4]==1'b1 || emergency==1'b1))
                next_state_var = s_4_stopU;
            else if(Sensor[4]==1'b1 && (Hall_call_Down[5]==1'b1 ||
Car_call[5]==1'b1))
                next_state_var = s_45_Up;
            else if(Sensor[4]==1'b1)
                next_state_var = s_4_stopU;
            end

        s_43_Down: begin
            if(Sensor[3]==1'b1 && (Hall_call_Down[3]==1'b1 ||
Car_call[3]==1'b1 || emergency==1'b1))
                next_state_var = s_3_stopD;
            else if(Sensor[3]==1'b1 && (Hall_call_Up[0]==1'b1 ||
Car_call[0]==1'b1 || Hall_call_Down[1]==1'b1|| Hall_call_Up[1]==1'b1
|| Car_call[1]==1'b1 || Hall_call_Down[2]==1'b1 ||
Hall_call_Up[2]==1'b1 || Car_call[2]==1'b1))

```

```

        next_state_var = s_32_Down;
    else if (Sensor[3]==1'b1)
        next_state_var = s_3_stopD;
    end

    s_4_stopU: begin
        if(emergency ==1'b1 || Hall_call_Down[4]==1'b1 ||
Hall_call_Up[4]==1'b1 || Car_call[4]==1'b1)
            next_state_var = s_4_openU;
        else if (Hall_call_Down[5]==1'b1 || Car_call[5]==1'b1)
            next_state_var = s_45_Up;
        else if (Hall_call_Up[0]==1'b1 || Car_call[0]==1'b1||
Hall_call_Down[1]==1'b1|| Hall_call_Up[1]==1'b1 || Car_call[1]==1'b1
|| Hall_call_Down[2]==1'b1 || Hall_call_Up[2]==1'b1 ||
Car_call[2]==1'b1|| Hall_call_Down[3]==1'b1 || Hall_call_Up[3]==1'b1
|| Car_call[3]==1'b1)
            next_state_var = s_43_Down;
        end

    s_4_stopD: begin
        if(emergency ==1'b1 || Hall_call_Down[4]==1'b1 ||
Hall_call_Up[4]==1'b1 || Car_call[4]==1'b1)
            next_state_var = s_4_openD;
        else if (Hall_call_Up[0]==1'b1 || Car_call[0]==1'b1||
Hall_call_Down[1]==1'b1|| Hall_call_Up[1]==1'b1 || Car_call[1]==1'b1
|| Hall_call_Down[2]==1'b1 || Hall_call_Up[2]==1'b1 ||
Car_call[2]==1'b1|| Hall_call_Down[3]==1'b1 || Hall_call_Up[3]==1'b1
|| Car_call[3]==1'b1)
            next_state_var = s_43_Down;
        else if (Hall_call_Down[5]==1'b1 || Car_call[5]==1'b1)
            next_state_var = s_45_Up;
        end

    s_4_openU: begin
        if(emergency ==1'b1)
            next_state_var = s_4_openU;
        else
            if(Passenger_in ==1'b1)
                next_state_var = s_4_stopU;
            end

    s_4_openD: begin
        if(emergency ==1'b1)
            next_state_var = s_4_openD;
        else
            if(Passenger_in ==1'b1)
                next_state_var = s_4_stopD;
            end

    s_45_Up: begin
        if (Sensor[5]==1'b1)
            next_state_var = s_5_stop;

```

```

        end

        s_54_Down: begin
            if(Sensor[4]==1'b1 && (Hall_call_Down[4]==1'b1 ||
Car_call[4]==1'b1 || emergency==1'b1))
                next_state_var = s_4_stopD;
            else if(Sensor[4]==1'b1 && (Hall_call_Up[0]==1'b1 ||
Car_call[0]==1'b1 || Hall_call_Down[1]==1'b1|| Hall_call_Up[1]==1'b1
|| Car_call[1]==1'b1 || Hall_call_Down[2]==1'b1 ||
Hall_call_Up[2]==1'b1 || Car_call[2]==1'b1 || Hall_call_Down[3]==1'b1
|| Hall_call_Up[3]==1'b1 || Car_call[3]==1'b1 ))
                next_state_var = s_43_Down;
            else if(Sensor[4]==1'b1)
                next_state_var = s_4_stopD;
            end

        s_5_stop: begin
            if(emergency ==1'b1 || Hall_call_Down[5]==1'b1 ||
Car_call[5]==1'b1)
                next_state_var = s_5_open;
            else if (Hall_call_Up[0]==1'b1 || Car_call[0]==1'b1 ||
Hall_call_Down[1]==1'b1|| Hall_call_Up[1]==1'b1 || Car_call[1]==1'b1
|| Hall_call_Down[2]==1'b1 || Hall_call_Up[2]==1'b1 ||
Car_call[2]==1'b1 || Hall_call_Down[3]==1'b1 || Hall_call_Up[3]==1'b1
|| Car_call[3]==1'b1 || Hall_call_Down[4]==1'b1 ||
Hall_call_Up[4]==1'b1 || Car_call[4]==1'b1)
                next_state_var = s_54_Down;
            end

        s_5_open: begin
            if(emergency ==1'b1)
                next_state_var = s_5_open;
            else
                if(Passenger_in ==1'b1)
                    next_state_var = s_5_stop;
                end
            end

        default : begin
            next_state_var = s_0_stop;
            end
    endcase
end

always @(posedge clk)
    state_signal <= next_state_var;

always @(state_signal)
begin
    case(state_signal)
    s_0_stop: begin
        MoveUp <= 1'b1;
    end
    end
end

```

```

        MoveDown <= 1'b0;
        OpenDoor <= 1'b0;
        CloseDoor <=1'b1;
        Stop <=1'b1;
        LCD <= 3'b000;
    end

s_0_open: begin
    MoveUp <= 1'b1;
    MoveDown <= 1'b0;
    OpenDoor <=1'b1;
    CloseDoor <= 1'b0;
    Stop <=1'b1;
    LCD<=3'b000;
end

s_01_Up: begin
    MoveUp <=1'b1;
    MoveDown <= 1'b0;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b0;
    LCD<= 3'b000;
end

s_10_Down: begin
    MoveUp <= 1'b0;
    MoveDown <=1'b1;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b0;
    LCD<= 3'b001;
end

s_1_stopU: begin
    MoveUp <= 1'b1;
    MoveDown <= 1'b0;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b1;
    LCD<=3'b001;
end

s_1_stopD: begin
    MoveUp <= 1'b0;
    MoveDown <= 1'b1;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b1;
    LCD<= 3'b001;
end

```

```

s_1_openU: begin
    MoveUp <= 1'b1;
    MoveDown <= 1'b0;
    OpenDoor <=1'b1;
    CloseDoor <= 1'b0;
    Stop <=1'b1;
    LCD<= 3'b001;
end

s_1_openD: begin
    MoveUp <= 1'b0;
    MoveDown <= 1'b1;
    OpenDoor <=1'b1;
    CloseDoor <= 1'b0;
    Stop <=1'b1;
    LCD<=3'b001;
end

s_12_Up: begin
    MoveUp <=1'b1;
    MoveDown <= 1'b0;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b0;
    LCD<= 3'b001;
end

s_21_Down: begin
    MoveUp <= 1'b0;
    MoveDown <=1'b1;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b0;
    LCD<= 3'b010;
end

s_2_stopU: begin
    MoveUp <= 1'b1;
    MoveDown <= 1'b0;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b1;
    LCD<= 3'b010;
end

s_2_stopD: begin
    MoveUp <= 1'b0;
    MoveDown <= 1'b1;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b1;
    LCD<= 3'b010;
end

```

```

        end

s_2_openU: begin
    MoveUp <= 1'b1;
    MoveDown <= 1'b0;
    OpenDoor <=1'b1;
    CloseDoor <= 1'b0;
    Stop <=1'b1;
    LCD<= 3'b010;
end

s_2_openD: begin
    MoveUp <= 1'b0;
    MoveDown <= 1'b1;
    OpenDoor <=1'b1;
    CloseDoor <= 1'b0;
    Stop <=1'b1;
    LCD<= 3'b010;
end

s_23_Up: begin
    MoveUp <=1'b1;
    MoveDown <= 1'b0;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b0;
    LCD<= 3'b010;
end

s_32_Down: begin
    MoveUp <= 1'b0;
    MoveDown <=1'b1;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b0;
    LCD<=3'b011;
end

s_3_stopU: begin
    MoveUp <= 1'b1;
    MoveDown <= 1'b0;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b1;
    LCD<= 3'b011;
end

s_3_stopD: begin
    MoveUp <= 1'b0;
    MoveDown <= 1'b1;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;

```



```

        Stop <=1'b1;
        LCD<= 3'b011;
    end

s_3_openU: begin
    MoveUp <= 1'b1;
    MoveDown <= 1'b0;
    OpenDoor <=1'b1;
    CloseDoor <= 1'b0;
    Stop <=1'b1;
    LCD<= 3'b011;
end

s_3_openD: begin
    MoveUp <= 1'b0;
    MoveDown <= 1'b1;
    OpenDoor <=1'b1;
    CloseDoor <= 1'b0;
    Stop <=1'b1;
    LCD<= 3'b011;
end

s_34_Up: begin
    MoveUp <=1'b1;
    MoveDown <= 1'b0;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b0;
    LCD<= 3'b011;
end

s_43_Down: begin
    MoveUp <= 1'b0;
    MoveDown <=1'b1;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b0;
    LCD<=3'b100;
end

s_4_stopU: begin
    MoveUp <= 1'b1;
    MoveDown <= 1'b0;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b1;
    LCD<= 3'b100;
end

s_4_stopD: begin
    MoveUp <= 1'b0;
    MoveDown <= 1'b1;

```

```

        OpenDoor <= 1'b0;
        CloseDoor <=1'b1;
        Stop <=1'b1;
        LCD<= 3'b100;
    end

s_4_openU: begin
    MoveUp <= 1'b1;
    MoveDown <= 1'b0;
    OpenDoor <=1'b1;
    CloseDoor <= 1'b0;
    Stop <=1'b1;
    LCD<= 3'b100;
end

s_4_openD: begin
    MoveUp <= 1'b0;
    MoveDown <= 1'b1;
    OpenDoor <=1'b1;
    CloseDoor <= 1'b0;
    Stop <=1'b1;
    LCD<= 3'b100;
end

s_45_Up: begin
    MoveUp <=1'b1;
    MoveDown <= 1'b0;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b0;
    LCD<= 3'b100;
end

s_54_Down: begin
    MoveUp <= 1'b0;
    MoveDown <=1'b1;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b0;
    LCD<= 3'b101;
end

s_5_stop: begin
    MoveUp <= 1'b0;
    MoveDown <= 1'b1;
    OpenDoor <= 1'b0;
    CloseDoor <=1'b1;
    Stop <=1'b1;
    LCD<= 3'b101;
end

s_5_open: begin

```

```

        MoveUp <= 1'b0;
        MoveDown <= 1'b1;
        OpenDoor <=1'b1;
        CloseDoor <= 1'b0;
        Stop <=1'b1;
        LCD<= 3'b101;
    end

    default: begin
        MoveUp <= 1'b0;
        MoveDown <= 1'b0;
        OpenDoor <= 1'b0;
        CloseDoor <=1'b1;
        Stop <=1'b1;
        LCD<= 3'b000;
    end

endcase
end
endmodule

```

## 6. Lift Master Controller

```

module
Lift_master_controller(clk,emergency,Sensor1,Sensor2,Sensor3,Move_Up,M
ove_Down,Stop,Hall_call_Up_signal,Hall_call_Down_signal,Hall_call_Up_s
ignalout1,Hall_call_Up_signalout2,Hall_call_Up_signalout3,Hall_call_Do
wn_signalout1,Hall_call_Down_signalout2, Hall_call_Down_signalout3);
    input clk;
    input [3:1] emergency,Move_Up,Move_Down,Stop;
    input [5:0] Sensor1,Sensor2,Sensor3;
    input [4:0] Hall_call_Up_signal;
    input [5:1] Hall_call_Down_signal;
    output reg[4:0]
Hall_call_Up_signalout1=5'b00000,Hall_call_Up_signalout2=5'b00000,Hall
_call_Up_signalout3=5'b00000;
    output reg [5:1] Hall_call_Down_signalout1=5'b00000,
Hall_call_Down_signalout2=5'b00000,
Hall_call_Down_signalout3=5'b00000;

    reg [5:0] a=6'b000000;
    integer b,c,d;
    reg one_near_than_two;
    reg two_near_than_three;
    reg three_near_than_one;
    reg one_is_nearest,two_is_nearest,three_is_nearest;

    always @
    (clk,emergency,Move_Up,Move_Down,Stop,Sensor1,Sensor2,Sensor3,Hall_cal
l_Up_signal,Hall_call_Down_signal) begin

        if(clk ==1'b0) begin

```

```

a =
{1'b0,Hall_call_Up_signal}||{Hall_call_Down_signal,1'b0};
    if((Sensor1|a)%9==1'b0)
        b = 3;
    else if((Sensor1|a)%33==1'b0)
        b = 5;
    else if((Sensor1|a)%3==1'b0)
        b = 1;
    else if((Sensor1|a)%5==1'b0)
        b = 2;
    else if((Sensor1|a)%17==1'b0)
        b = 4;
    else
        b = 0;

    if((Sensor2|a)%9==1'b0)
        c = 3;
    else if((Sensor2|a)%33==1'b0)
        c = 5;
    else if((Sensor2|a)%3==1'b0)
        c = 1;
    else if((Sensor2|a)%5==1'b0)
        c = 2;
    else if((Sensor2|a)%17==1'b0)
        c = 4;
    else
        c = 0;

    if((Sensor3|a)%9==1'b0)
        d = 3;
    else if((Sensor3|a)%33==1'b0)
        d = 5;
    else if((Sensor3|a)%3==1'b0)
        d = 1;
    else if((Sensor3|a)%5==1'b0)
        d = 2;
    else if((Sensor3|a)%17==1'b0)
        d = 4;
    else
        d = 0;

    if(b<c)
        one_near_than_two=1'b1;
    else
        one_near_than_two=1'b0;

    if(d<c)
        two_near_than_three=1'b0;
    else
        two_near_than_three=1'b1;

```

```

        if(d<b)
            three_near_than_one=1'b1;
        else
            three_near_than_one=1'b0;

        if(emergency==3'b000)begin
            if((one_near_than_two==1'b1 &&
two_near_than_three==1'b1 &&
three_near_than_one==1'b0)|| (one_near_than_two==1'b1 &&
two_near_than_three==1'b0 && three_near_than_one==1'b0))begin
                one_is_nearest = 1'b1;two_is_nearest =
1'b0;three_is_nearest = 1'b0;end
            else if((one_near_than_two==1'b1 &&
two_near_than_three==1'b0 &&
three_near_than_one==1'b1)|| (one_near_than_two==1'b0 &&
two_near_than_three==1'b0 && three_near_than_one==1'b1))begin
                one_is_nearest = 1'b0;two_is_nearest =
1'b0;three_is_nearest = 1'b1;end
            else begin
                one_is_nearest = 1'b0;two_is_nearest =
1'b1;three_is_nearest = 1'b0;end
            end

        else if(emergency[1]==1'b1)begin
            two_is_nearest = two_near_than_three;
            three_is_nearest = ~ two_near_than_three;
        end

        else if(emergency[2]==1'b1)begin
            one_is_nearest = ~ three_near_than_one;
            three_is_nearest = three_near_than_one;
        end

        else if(emergency[3]==1'b1)begin
            one_is_nearest = one_near_than_two;
            two_is_nearest = ~(one_near_than_two);
        end

    end

    if(clk ==1'b1) begin
        if(Hall_call_Up_signal[0]==1'b1)begin
            if((Move_Down[1]==1'b1||Stop[1]==1'b1) &&
one_is_nearest==1'b1 && emergency[1]==1'b0)
                Hall_call_Up_signalout1[0]
=Hall_call_Up_signal[0];
            else if((Move_Down[3]==1'b1||Stop[3]==1'b1) &&
three_is_nearest==1'b1 && emergency[3]==1'b0)
                Hall_call_Up_signalout3[0]
=Hall_call_Up_signal[0];
            else if(emergency[2]==1'b0)

```

```

        Hall_call_Up_signalout2[0]
=Hall_call_Up_signal[0];
    end
    else begin
        Hall_call_Up_signalout1[0] =Hall_call_Up_signal[0];
        Hall_call_Up_signalout3[0] =Hall_call_Up_signal[0];
        Hall_call_Up_signalout2[0] =Hall_call_Up_signal[0];
    end

    if(Hall_call_Up_signal[1]==1'b1)begin
        if(((Move_Down[1]==1'b1 && (Sensor1[5]==1'b1 ||
Sensor1[4]==1'b1|| Sensor1[3]==1'b1|| Sensor1[2]==1'b1)) ||
(Move_Up[1]==1'b1 && Sensor1[0]==1'b1 ))&& one_is_nearest==1'b1 &&
emergency[1]==1'b0)

Hall_call_Up_signalout1[1]=Hall_call_Up_signal[1];
        else if(((Move_Down[3]==1'b1 && (Sensor3[5]==1'b1 ||
Sensor3[4]==1'b1|| Sensor3[3]==1'b1|| Sensor3[2]==1'b1)) ||
(Move_Up[3]==1'b1 && Sensor3[0]==1'b1 ))&& three_is_nearest==1'b1 &&
emergency[3]==1'b0)

Hall_call_Up_signalout3[1]=Hall_call_Up_signal[1];
        else if(emergency[2]==1'b0)

Hall_call_Up_signalout2[1]=Hall_call_Up_signal[1];
    end
    else begin
        Hall_call_Up_signalout1[1] =Hall_call_Up_signal[1];
        Hall_call_Up_signalout3[1] =Hall_call_Up_signal[1];
        Hall_call_Up_signalout2[1] =Hall_call_Up_signal[1];
    end

    if(Hall_call_Down_signal[1]==1'b1)begin
        if(((Move_Down[1]==1'b1 && (Sensor1[5]==1'b1 ||
Sensor1[4]==1'b1|| Sensor1[3]==1'b1|| Sensor1[2]==1'b1)) ||
(Move_Up[1]==1'b1 && Sensor1[0]==1'b1 ))&& one_is_nearest==1'b1 &&
emergency[1]==1'b0)

Hall_call_Down_signalout1[1]=Hall_call_Down_signal[1];
        else if(((Move_Down[3]==1'b1 && (Sensor3[5]==1'b1 ||
Sensor3[4]==1'b1|| Sensor3[3]==1'b1|| Sensor3[2]==1'b1)) ||
(Move_Up[3]==1'b1 && Sensor3[0]==1'b1 ))&& three_is_nearest==1'b1 &&
emergency[3]==1'b0)

Hall_call_Down_signalout3[1]=Hall_call_Down_signal[1];
        else if(emergency[2]==1'b0)

Hall_call_Down_signalout2[1]=Hall_call_Down_signal[1];
    end
    else begin
        Hall_call_Down_signalout1[1]=Hall_call_Down_signal[1];
        Hall_call_Down_signalout3[1]=Hall_call_Down_signal[1];

```

```

        Hall_call_Down_signalout2[1]=Hall_call_Down_signal[1];
    end

    if(Hall_call_Up_signal[2]==1'b1)begin
        if(((Move_Down[1]==1'b1 && (Sensor1[5]==1'b1 ||
Sensor1[4]==1'b1 || Sensor1[3]==1'b1)) || (Move_Up[1]==1'b1 &&
(Sensor1[1]==1'b1 || Sensor1[0]==1'b1)))&& one_is_nearest==1'b1 &&
emergency[1]==1'b0)

Hall_call_Up_signalout1[2]=Hall_call_Up_signal[2];
        else if(((Move_Down[3]==1'b1 && (Sensor3[5]==1'b1 ||
Sensor3[4]==1'b1 || Sensor3[3]==1'b1)) || (Move_Up[3]==1'b1 &&
(Sensor3[1]==1'b1 || Sensor3[0]==1'b1)))&& three_is_nearest==1'b1 &&
emergency[3]==1'b0)

Hall_call_Up_signalout3[2]=Hall_call_Up_signal[2];
        else if(emergency[2]==1'b0)

Hall_call_Up_signalout2[2]=Hall_call_Up_signal[2];
        end
        else begin
            Hall_call_Up_signalout1[2]=Hall_call_Up_signal[2];
            Hall_call_Up_signalout3[2]=Hall_call_Up_signal[2];
            Hall_call_Up_signalout2[2]=Hall_call_Up_signal[2];
        end
    end

    if(Hall_call_Down_signal[2]==1'b1)begin
        if(((Move_Down[1]==1'b1 && (Sensor1[5]==1'b1 ||
Sensor1[4]==1'b1 || Sensor1[3]==1'b1)) || (Move_Up[1]==1'b1 &&
(Sensor1[1]==1'b1 || Sensor1[0]==1'b1)))&& one_is_nearest==1'b1 &&
emergency[1]==1'b0)

Hall_call_Down_signalout1[2]=Hall_call_Down_signal[2];
        else if(((Move_Down[3]==1'b1 && (Sensor3[5]==1'b1 ||
Sensor3[4]==1'b1 || Sensor3[3]==1'b1)) || (Move_Up[3]==1'b1 &&
(Sensor3[1]==1'b1 || Sensor3[0]==1'b1)))&& three_is_nearest==1'b1 &&
emergency[3]==1'b0)

Hall_call_Down_signalout3[2]=Hall_call_Down_signal[2];
        else if(emergency[2]==1'b0)

Hall_call_Down_signalout2[2]=Hall_call_Down_signal[2];
        end
        else begin
            Hall_call_Down_signalout1[2]=Hall_call_Down_signal[2];
            Hall_call_Down_signalout3[2]=Hall_call_Down_signal[2];
            Hall_call_Down_signalout2[2]=Hall_call_Down_signal[2];
        end
    end

    if(Hall_call_Up_signal[3]==1'b1)begin

```

```

        if((Move_Down[1]==1'b1 && (Sensor1[5]==1'b1 ||
Sensor1[4]==1'b1)) || (Move_Up[1]==1'b1 && (Sensor1[2]==1'b1 ||
Sensor1[1]==1'b1|| Sensor1[0]==1'b1)))&& one_is_nearest==1'b1 &&
emergency[1]==1'b0)

Hall_call_Up_signalout1[3]=Hall_call_Up_signal[3];
        else if((Move_Down[3]==1'b1 && (Sensor3[5]==1'b1 ||
Sensor3[4]==1'b1)) || (Move_Up[3]==1'b1 && (Sensor3[2]==1'b1 ||
Sensor3[1]==1'b1|| Sensor3[0]==1'b1)))&& three_is_nearest==1'b1 &&
emergency[3]==1'b0)

Hall_call_Up_signalout3[3]=Hall_call_Up_signal[3];
        else if(emergency[2]==1'b0)

Hall_call_Up_signalout2[3]=Hall_call_Up_signal[3];
        end
        else begin
            Hall_call_Up_signalout1[3]=Hall_call_Up_signal[3];
            Hall_call_Up_signalout3[3]=Hall_call_Up_signal[3];
            Hall_call_Up_signalout2[3]=Hall_call_Up_signal[3];
        end

        if(Hall_call_Down_signal[3]==1'b1)begin
            if((Move_Down[1]==1'b1 && (Sensor1[5]==1'b1 ||
Sensor1[4]==1'b1)) || (Move_Up[1]==1'b1 && (Sensor1[2]==1'b1 ||
Sensor1[1]==1'b1|| Sensor1[0]==1'b1)))&& one_is_nearest==1'b1 &&
emergency[1]==1'b0)

Hall_call_Down_signalout1[3]=Hall_call_Down_signal[3];
            else if((Move_Down[3]==1'b1 && (Sensor3[5]==1'b1 ||
Sensor3[4]==1'b1)) || (Move_Up[3]==1'b1 && (Sensor3[2]==1'b1 ||
Sensor3[1]==1'b1|| Sensor3[0]==1'b1)))&& three_is_nearest==1'b1 &&
emergency[3]==1'b0)

Hall_call_Down_signalout3[3]=Hall_call_Down_signal[3];
            else if(emergency[2]==1'b0)

Hall_call_Down_signalout2[3]=Hall_call_Down_signal[3];
            end
            else begin
                Hall_call_Down_signalout1[3]=Hall_call_Down_signal[3];
                Hall_call_Down_signalout3[3]=Hall_call_Down_signal[3];
                Hall_call_Down_signalout2[3]=Hall_call_Down_signal[3];
            end

            if(Hall_call_Up_signal[4]==1'b1)begin
                if((Move_Down[1]==1'b1 && Sensor1[5]==1'b1 ) ||
(Move_Up[1]==1'b1 && (Sensor1[3]==1'b1 || Sensor1[2]==1'b1 ||
Sensor1[1]==1'b1|| Sensor1[0]==1'b1)))&& one_is_nearest==1'b1 &&
emergency[1]==1'b0)

Hall_call_Up_signalout1[4]=Hall_call_Up_signal[4];

```



```

        else if(((Move_Down[3]==1'b1 && Sensor3[5]==1'b1 ) ||
(Move_Up[3]==1'b1 && (Sensor3[3]==1'b1 || Sensor3[2]==1'b1 ||
Sensor3[1]==1'b1|| Sensor3[0]==1'b1)))&& three_is_nearest==1'b1 &&
emergency[3]==1'b0)

Hall_call_Up_signalout3[4]=Hall_call_Up_signal[4];
        else if(emergency[2]==1'b0)

Hall_call_Up_signalout2[4]=Hall_call_Up_signal[4];
        end
        else begin
            Hall_call_Up_signalout1[4]=Hall_call_Up_signal[4];
            Hall_call_Up_signalout3[4]=Hall_call_Up_signal[4];
            Hall_call_Up_signalout2[4]=Hall_call_Up_signal[4];
        end

        if(Hall_call_Down_signal[4]==1'b1)begin
            if(((Move_Down[1]==1'b1 && Sensor1[5]==1'b1 ) ||
(Move_Up[1]==1'b1 && (Sensor1[3]==1'b1 || Sensor1[2]==1'b1 ||
Sensor1[1]==1'b1|| Sensor1[0]==1'b1)))&& one_is_nearest==1'b1 &&
emergency[1]==1'b0)

Hall_call_Down_signalout1[4]=Hall_call_Down_signal[4];
            else if(((Move_Down[3]==1'b1 && Sensor3[5]==1'b1 ) ||
(Move_Up[3]==1'b1 && (Sensor3[3]==1'b1 || Sensor3[2]==1'b1 ||
Sensor3[1]==1'b1|| Sensor3[0]==1'b1)))&& three_is_nearest==1'b1 &&
emergency[3]==1'b0)

Hall_call_Down_signalout3[4]=Hall_call_Down_signal[4];
            else if(emergency[2]==1'b0)

Hall_call_Down_signalout2[4]=Hall_call_Down_signal[4];
            end
            else begin
                Hall_call_Down_signalout1[4]=Hall_call_Down_signal[4];
                Hall_call_Down_signalout3[4]=Hall_call_Down_signal[4];
                Hall_call_Down_signalout2[4]=Hall_call_Down_signal[4];
            end

            if(Hall_call_Down_signal[5]==1'b1)begin
                if(Move_Up[1]==1'b1 && one_is_nearest==1'b1 &&
emergency[1]==1'b0)

Hall_call_Down_signalout1[5]=Hall_call_Down_signal[5];
                else if(Move_Up[3]==1'b1 && three_is_nearest==1'b1 &&
emergency[3]==1'b0)

Hall_call_Down_signalout3[5]=Hall_call_Down_signal[5];
                else if(emergency[2]==1'b0)

Hall_call_Down_signalout2[5]=Hall_call_Down_signal[5];
            end

```

```
        else begin
            Hall_call_Down_signalout1[5]=Hall_call_Down_signal[5];
            Hall_call_Down_signalout3[5]=Hall_call_Down_signal[5];
            Hall_call_Down_signalout2[5]=Hall_call_Down_signal[5];
        end
    end
end
endmodule
```