VLSI DESIGN LAB. ASSIGNMENT -4

SAURAV GUPTA

ROLL No. – 153070046

Dept.- Electrical Engineering

Specialisation – Microelectronics (TA)

Batch - 2015-17

Teacher concerned – VIRENDRA SINGH

PROBLEM STATEMENT:

Design a lift controller, which controls a lift that services passengers in a 6 floor building.

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.

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.

The controller 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 Write a synthesizable behavioural description of the above circuit in VHDL.

Sol:

Controller FSM

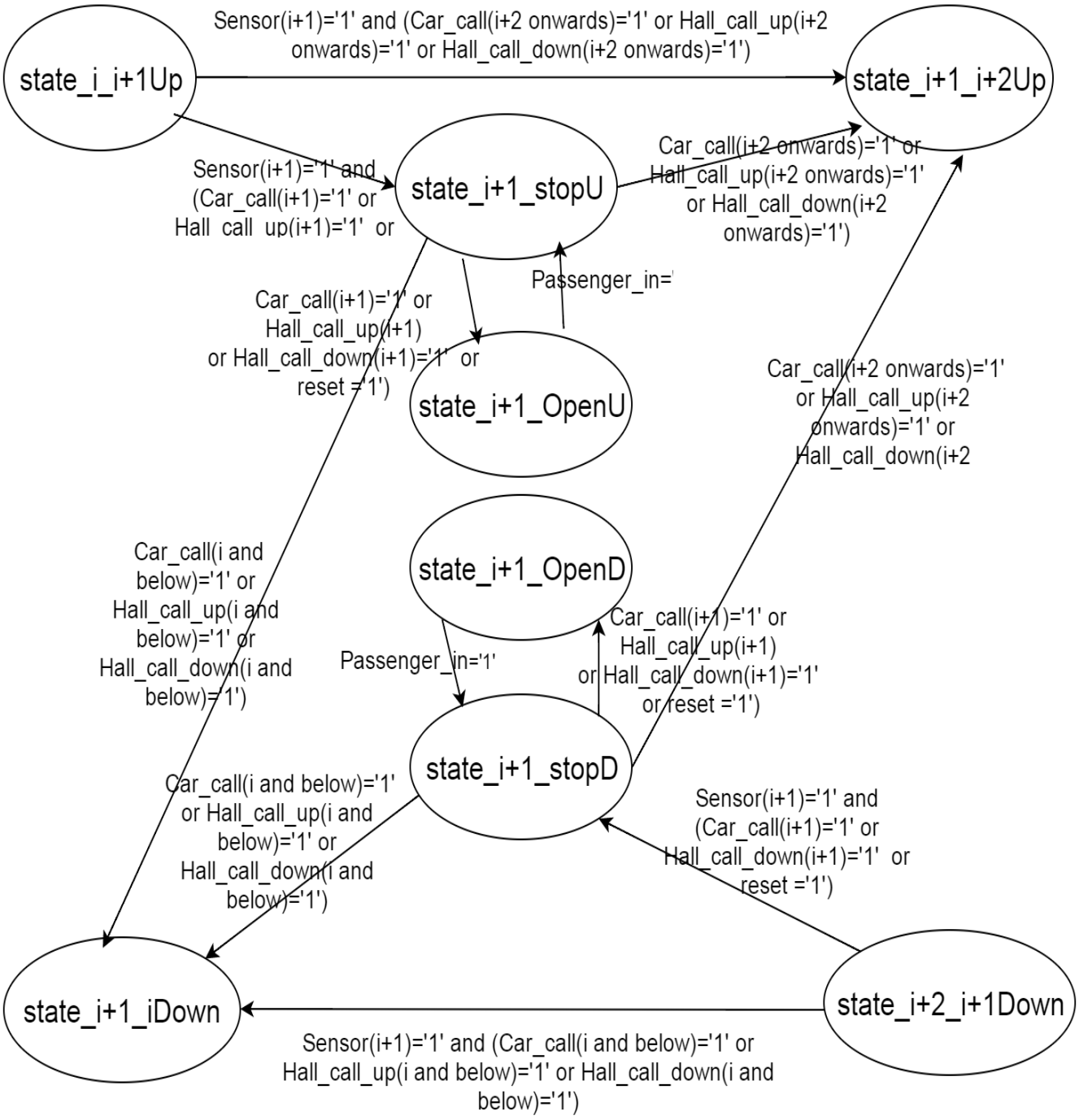


Fig. 1 Controller FSM for any state i

Circuit Diagram

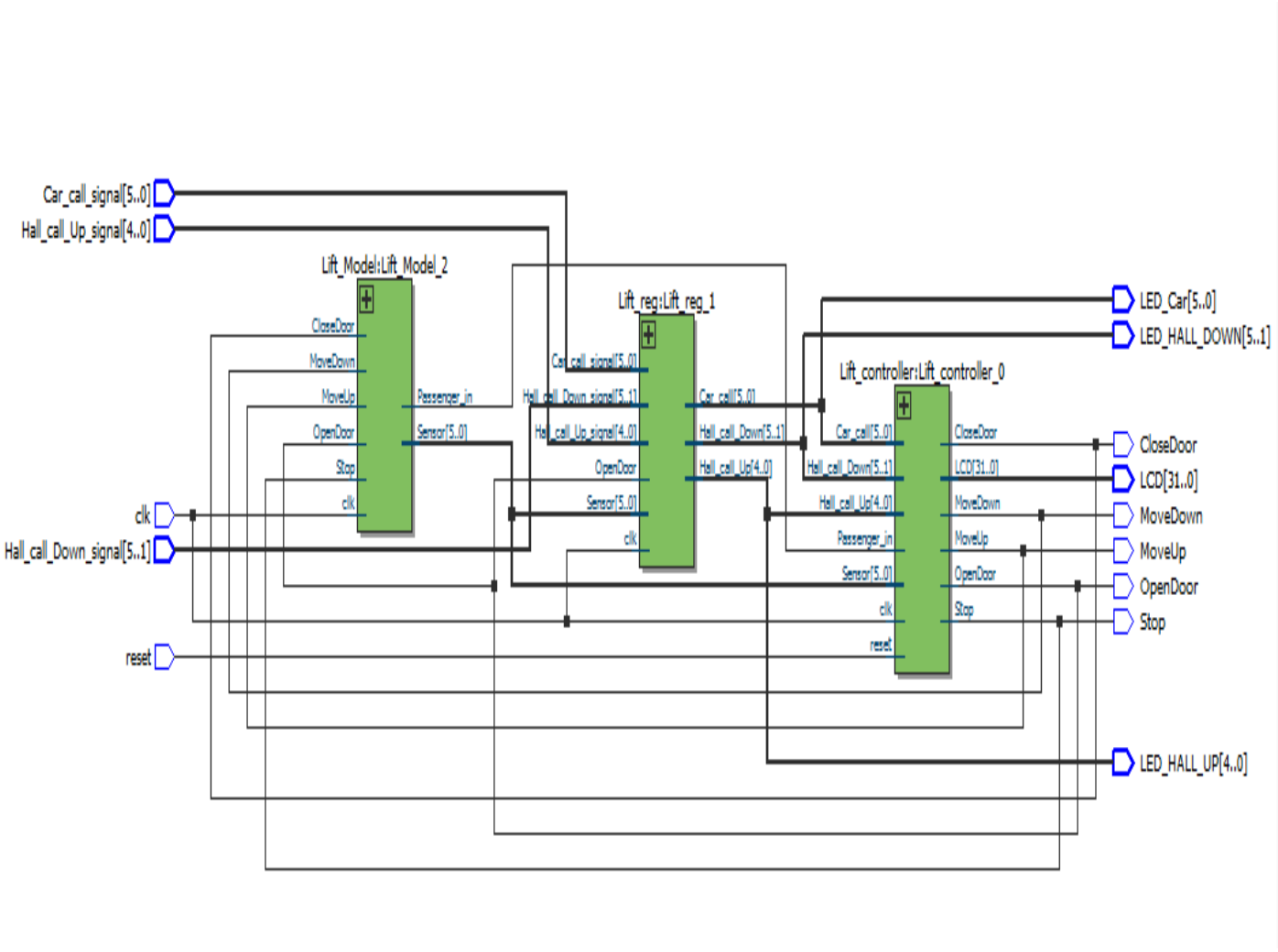


Fig. 2 Circuit Diagram

Individual Components

1. Lift Model

- **Lift**
- **counter**

2. Lift Register

3. Lift Controller

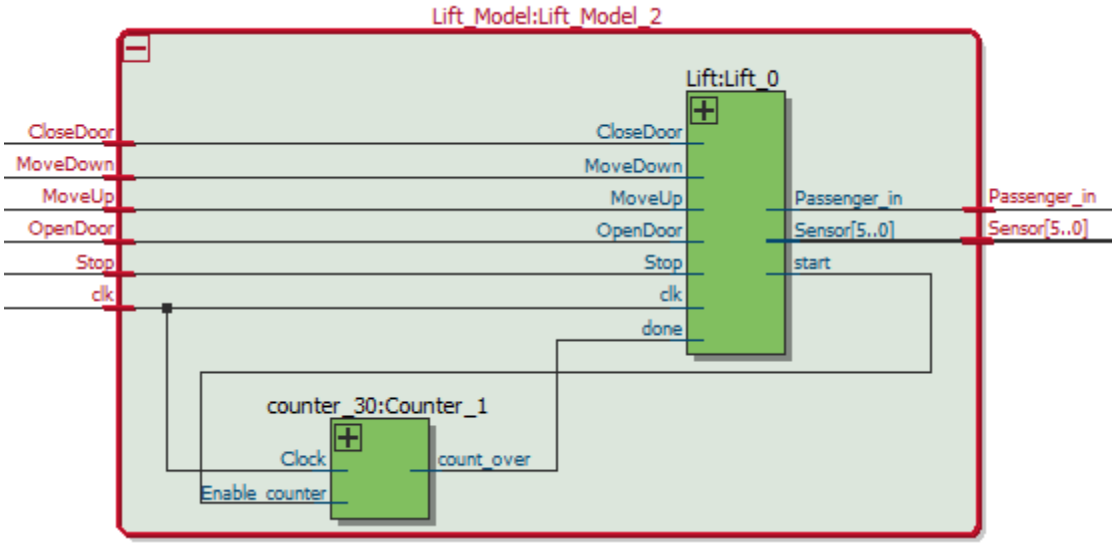
1. Lift Model

--This Model acts as a Lift.

```
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
```



1.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 reg Passenger_in,start;

      reg [5:0] Sensor_temp = 6'b000001;

      always @(posedge clk)
      begin
      if(clk==1'b1) begin
            if(MoveUp ==1'b1) begin
                  start<=1;
                  if(done==1'b1) begin
                        start<=0;
                        Sensor_temp<= Sensor_temp<<1;
                  end
            end

            else if(MoveDown ==1'b1) begin
                  start<=1;
                  if(done==1'b1) begin
                        start<=0;
                        Sensor_temp<= Sensor_temp>>1;
                  end
            end

            else if(OpenDoor==1'b1 && stop==1'b1) begin
                  start<=1;Passenger_in<=0;
                  if(done==1'b1) begin
                        start<=0;
                        Passenger_in<=1;
                  end
            end

            else if(CloseDoor==1'b1 && stop==1'b1) begin
                  start<=0;
            end
      end
      end
      assign Sensor = Sensor_temp;
endmodule
```

1. b Counter_30

--counts 30 clock cycle to provide delay for moving from one floor to other and also for time of closing door.

```
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
```

Lift Controller

```
module
Lift_controller(clk,reset,Passenger_in,Sensor,Car_call,Hall_call_Up,Ha
ll_call_Down,MoveUp,MoveDown,OpenDoor,CloseDoor,Stop,LCD);
      input clk,reset,Passenger_in;
      input [5:0] Sensor,Car_call;
      input [4:0] Hall_call_Up;
      input [5:1] Hall_call_Down;
      output reg MoveUp,MoveDown,OpenDoor,CloseDoor,Stop;
      output reg [2:0] LCD;

      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[0:5] state_signal, next_state_var;

      always
@(reset,Sensor,Hall_call_Up,Hall_call_Down,Car_call,Passenger_in,state
_signal)
      begin
            case(state_signal)
            s_0_stop: begin
                  if(reset ==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(reset ==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 || reset==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
          s_10_Down: begin
                if(Sensor[0]==1'b1)
                     next_state_var = s_0_stop;
                end

          s_1_stopU: begin
                if(reset ==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(reset ==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(reset ==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(reset ==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 || reset==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 || reset==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(reset ==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(reset ==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(reset ==1'b1)
                      next_state_var = s_2_openU;
                else
                      if(Passenger_in ==1'b1)
                            next_state_var = s_2_stopU;
                end

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

            s_23_Up: begin
                if(Sensor[3]==1'b1 && (Hall_call_Up[3]==1'b1 ||
Car_call[3]==1'b1 || reset==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 || reset==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(reset ==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[1]==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(reset ==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[1]==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(reset ==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(reset ==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 || reset==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 || reset==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(reset ==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(reset ==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(reset ==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(reset ==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 || reset==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(reset ==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(reset ==1'b1)
                     next_state_var = s_5_open;
                else
                     if(Passenger_in ==1'b1)
                          next_state_var = s_5_stop;
                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'b0;
                MoveDown <= 1'b0;
                OpenDoor <= 1'b0;
                CloseDoor <=1'b1;
                Stop <=1'b1;
                LCD <= 3'b000;
          end

        s_0_open: begin
                MoveUp <= 1'b0;
                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'b0;
           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'b0;
           OpenDoor <= 1'b0;
           CloseDoor <=1'b1;
           Stop <=1'b1;
           LCD<= 3'b001;
      end

    s_1_openU: begin
           MoveUp <= 1'b0;
           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'b0;
           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'b0;
            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'b0;
            OpenDoor <= 1'b0;
            CloseDoor <=1'b1;
            Stop <=1'b1;
            LCD<= 3'b010;
      end

    s_2_openU: begin
            MoveUp <= 1'b0;
            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'b0;
            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'b0;
            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'b0;
            OpenDoor <= 1'b0;
            CloseDoor <=1'b1;
            Stop <=1'b1;
            LCD<= 3'b011;
      end

    s_3_openU: begin
            MoveUp <= 1'b0;
            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'b0;
            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'b0;
            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'b0;
            OpenDoor <= 1'b0;
            CloseDoor <=1'b1;
            Stop <=1'b1;
            LCD<= 3'b100;
      end

    s_4_openU: begin
            MoveUp <= 1'b0;
            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'b0;
            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'b0;
                OpenDoor <= 1'b0;
                CloseDoor <=1'b1;
                Stop <=1'b1;
                LCD<= 3'b101;
          end

          s_5_open: begin
                MoveUp <= 1'b0;
                MoveDown <= 1'b0;
                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
```

Lift Register

-- Register users request

```
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'b00000;
      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
```

Overall Circuit

--final circuit

```
module
Lift_overall(clk,reset,Car_call_signal,Hall_call_Up_signal,Hall_call_D
own_signal,MoveUp,MoveDown,OpenDoor,CloseDoor,Stop,LCD,LED_HALL_UP,LED
_HALL_DOWN,LED_Car);
      input clk,reset;
      input [5:0] Car_call_signal;
      input [4:0] Hall_call_Up_signal;
      input [5:1] Hall_call_Down_signal;
      output MoveUp,MoveDown,OpenDoor,CloseDoor,Stop;
      output [2:0] LCD;
      output [4:0] LED_HALL_UP;
      output [5:1] LED_HALL_DOWN;
      output [5:0] LED_Car;

      wire [5:0] Car_call_temp;
      wire [4:0] Hall_call_Up_temp;
      wire [5:1] Hall_call_Down_temp;
      wire OpenDoor_temp,
CloseDoor_temp,Stop_temp,MoveDown_temp,MoveUp_temp,Passenger_in_temp;
      wire [5:0] Sensor;
      wire [2:0] LCD_temp;

      Lift_controller Lift_controller_0(clk,
reset,Passenger_in_temp,Sensor,Car_call_temp,Hall_call_Up_temp,Hall_ca
ll_Down_temp,MoveUp_temp,MoveDown_temp,OpenDoor_temp,CloseDoor_temp,St
op_temp,LCD_temp);
      Lift_reg
Lift_reg_1(clk,Car_call_signal,Hall_call_Up_signal,Hall_call_Down_sign
al,Sensor,OpenDoor_temp,Hall_call_Up_temp,Hall_call_Down_temp,Car_call
_temp);
      Lift_Model
Lift_Model_2(clk,MoveUp_temp,MoveDown_temp,OpenDoor_temp,CloseDoor_tem
p,Stop_temp,Sensor,Passenger_in_temp);
      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_temp;
```

```
      assign LED_HALL_DOWN = Hall_call_Down_temp;
      assign LED_Car = Car_call_temp;
      assign LCD = LCD_temp;
 endmodule
```

Overall circuit's test bench

```
`timescale 1ms/100us
module Lift_overall_tb;
      wire MoveUp,MoveDown,OpenDoor,CloseDoor,Stop;
      wire [4:0] LED_HALL_UP;
      wire [5:1] LED_HALL_DOWN;
      wire [5:0] LED_Car;
      wire [2:0] LCD;
      reg clk,reset;
      reg [5:0] Car_call_signal=6'b000000;
      reg [4:0] Hall_call_Up_signal=6'b00000;
      reg [5:1] Hall_call_Down_signal=6'b00000;

      Lift_overall
DUT(clk,reset,Car_call_signal,Hall_call_Up_signal,Hall_call_Down_signa
l,MoveUp,MoveDown,OpenDoor,CloseDoor,Stop,LCD,LED_HALL_UP,LED_HALL_DOW
N,LED_Car);

      initial
      begin
            $dumpfile("run.vcd");
            $dumpvars(0,Lift_overall_tb);
            clk=0;
            reset=0;
            Hall_call_Up_signal[2]=1; #600 Hall_call_Up_signal[2]=0;
            #2000
            Car_call_signal[4]=1; #600 Car_call_signal[4]=0;
            #800
            Hall_call_Down_signal[5]=1;#600 Hall_call_Down_signal[5]=0;
            #3000
            Car_call_signal[0]=1;#600 Car_call_signal[0]=0;
            #400
            Hall_call_Up_signal[3]=1;#600 Hall_call_Up_signal[3]=0;
            #10000 $finish;
      end

      always
      begin
            clk = #10 ~clk;
      end
      always
      begin
            #19000 reset=1;
      end
  endmodule
```