Digital Logic Design (2EC303) Special assignment

Design of Elevator Controller Using Verilog HDL

SUBMITTED BY: 21BEC030 DHRUV SHAH

21BEC035 DHRUV GOBBI

Principle:

- An Elevator is a type of vertical transport equipment that efficiently moves people or goods between floors (levels, decks) of a building, vessel or other structure. The elevator car has a pair of control buttons (up/down) for moving the elevator up and down. The following principles have been applied during the design of elevator controller: The floors are defined as first floor and second etc.

1.A floor call is serviced using the elevator.

2.Upon arrival at a floor, the doors open immediately.

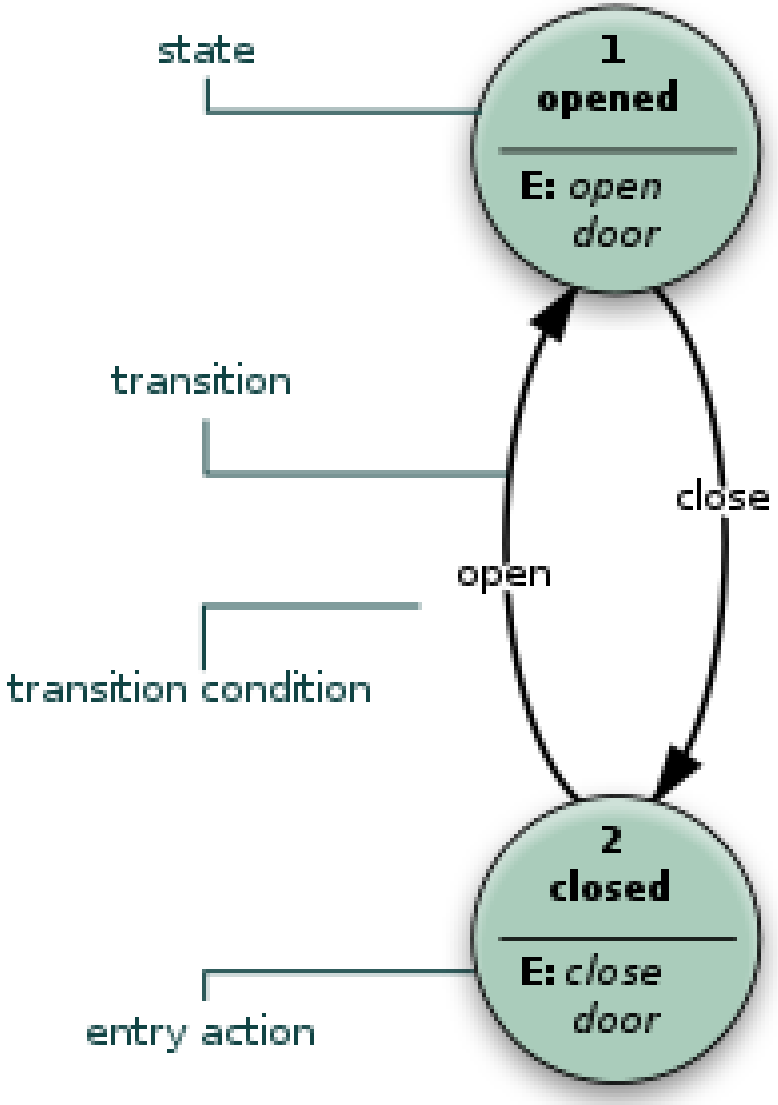
3.Doors remain open before closure.

4.Elevator Up/Down buttons are connected to elevator units.

Finite State Machine:

- It is a model of behavior made up of a finite number of states, transitions between them, and actions. The finite-state machine is similar to a flow graph in which one can inspect the way logic runs when certain conditions are met.

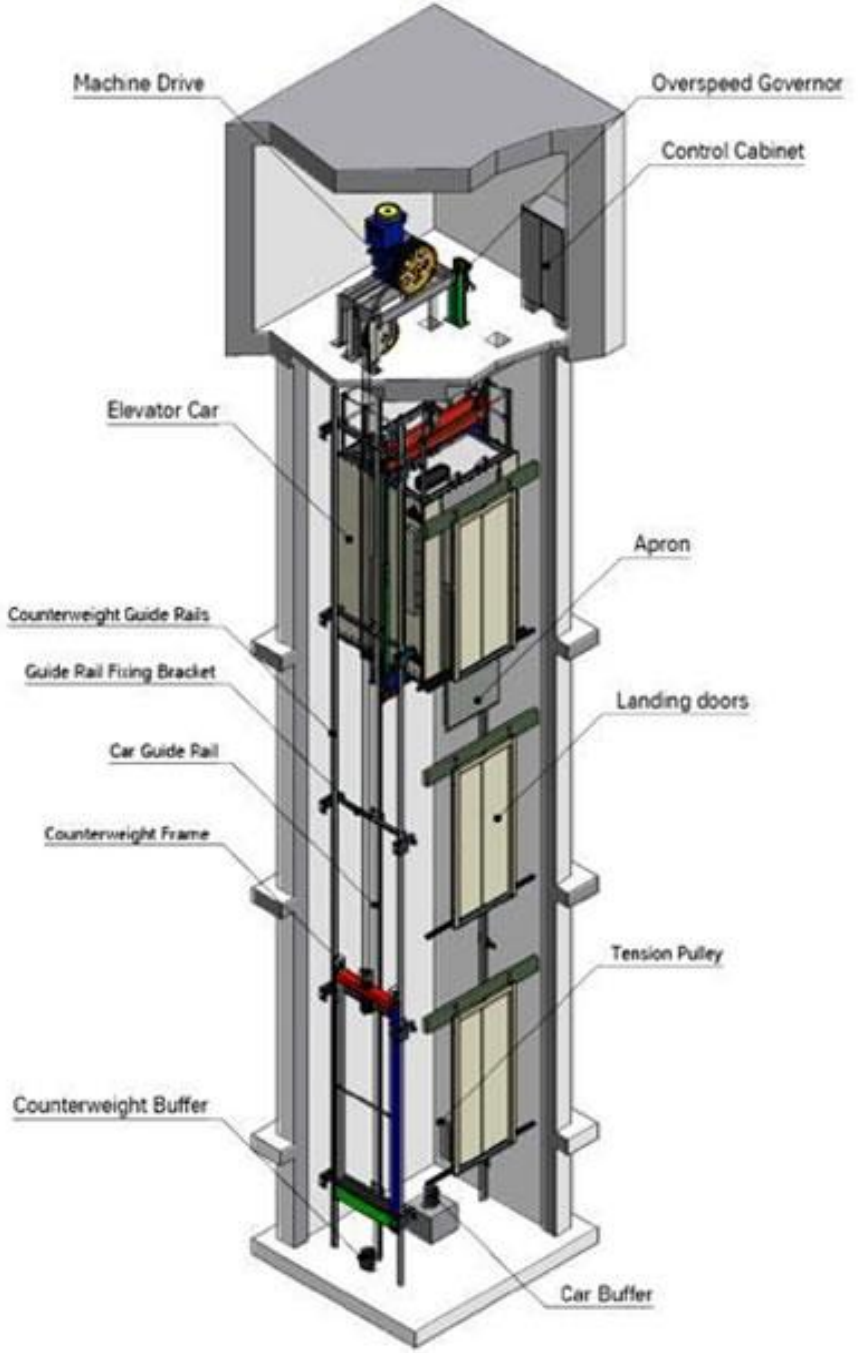
- It has finite internal memory, an input feature (reading symbols in a sequence, one at a time without going backward), and an output feature, which may be in the form of a user interface, once the model is implemented.



- The elevator control system is based on the finite state machine technology idea. The elevator process may be specified using several states according to FSM technology.

- In the FSM technology there is change from one state to another state likewise in the elevator there will be a change from one floor to another.

- The complete programme is built up such that desirable switches are available on each floor, as well as within the elevator, to regulate user commands.



- If we push any of the floor switches while the elevator is still on the ground level, the elevator will proceed higher until it reaches the specified floor and the door will open.
- The doors remain open for a short length of time before closing and the elevator moving to the next floor. The elevator will move in the direction indicated by the user's desired input.

Elevator Algorithm :

1.Consider 2 lifts L1 and L2, nearest one would be called depending upon the requested floor.

2.Keep going in the same path as long as there are demands in that direction.

3.If there are no further requests in that direction, then stop and become idle, or change direction if there are requests in the opposite direction.

Verilog Code:

- module LLiftC(clk,reset,req_floor,stop,door,Up,Down,y1,y2);
- input clk,reset;
- input [6:0] req_floor;
- output reg[1:0] door;
- output reg[1:0] Up;
- output reg[1:0] Down;
- output reg[1:0] stop;
- output [6:0] y1;
- output [6:0] y2;
- reg [6:0] cf;
- reg [6:0] l1 = 6'd0;
- reg [6:0] l2 = 6'd0;

```
always @ (posedge clk)
begin
   if(((l2 > l1) && (l1 > req_floor)) || ((req_floor > l1) && (l1 > l2)))
  cf = l1;
else if(((l1 > l2) && (l2 > req_floor)) || ((req_floor > l2) && (l2 > l1)))
  cf = l2;
else if((l1 > req_floor) && (req_floor > l2))
begin
  if((l1 - req_floor) > (req_floor - l2))
  cf = l2;
else
  cf = l1;
end
else if((l2 > req_floor) && (req_floor > l1))
begin
  if((l2 - req_floor) > (req_floor - l1))
  cf = l1;
else
  cf = l2;
end
else if(l1 == l2)
  cf = l1;
```

```
if(reset)
  begin
    cf=6'd0;
    stop=6'd1;
    door = 1'd1;
    Up=1'd0;
    Down=1'd0;
  end
  else
  begin
    if(req_floor < 6'd61)
    begin
      if(req_floor < cf )
      begin
        cf=cf-1;
        door=1'd0;
        stop=6'd0;
        Up=1'd0;
        Down=1'd1;
      end
```

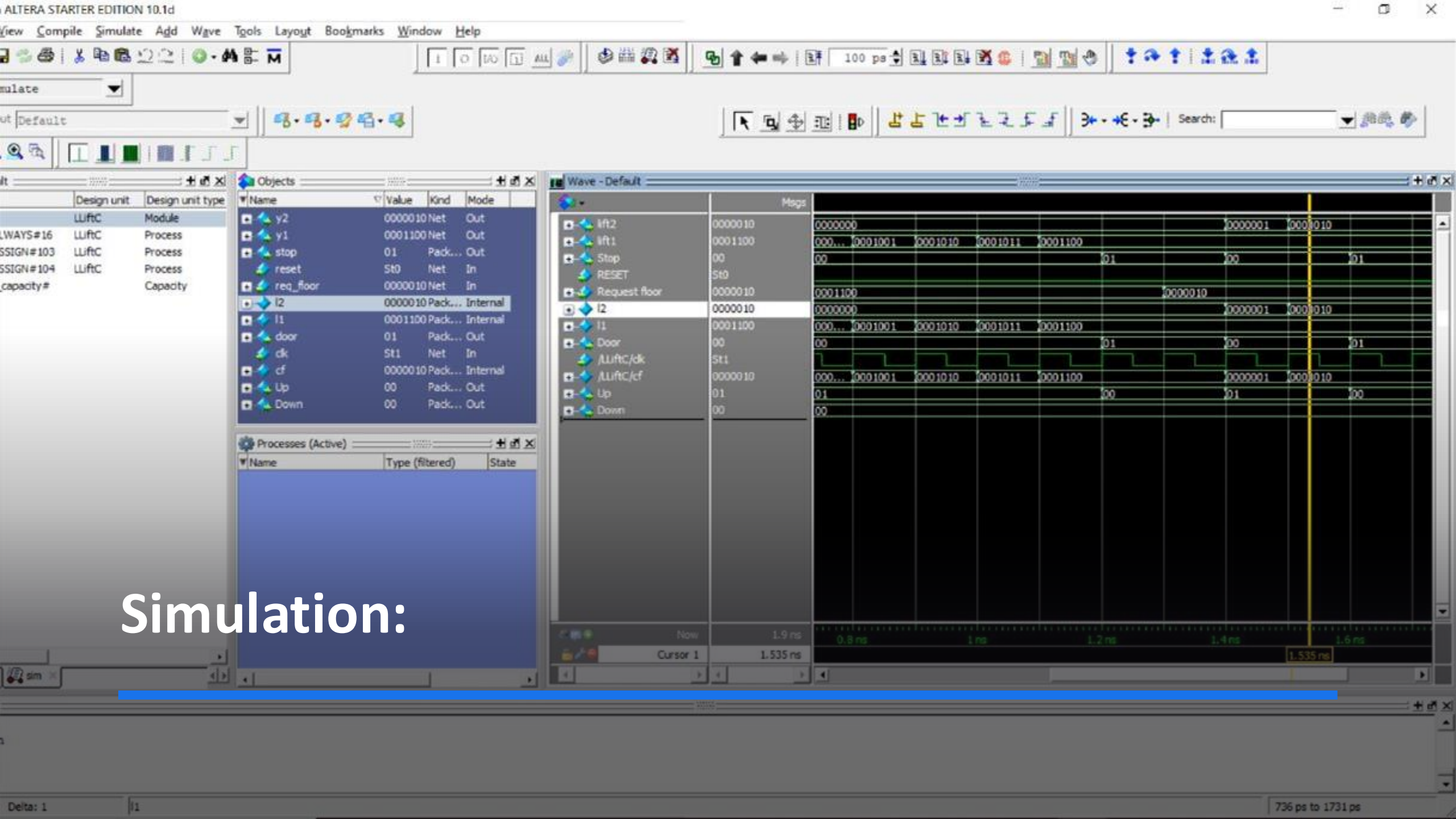
```
else if (req_floor > cf)
        begin
          cf = cf+1;
          door=1'd0;
          stop=6'd0;
          Up=1'd1;
          Down=1'd0;
        end

        else if(req_floor == cf )
        begin
          cf = req_floor;
          door=1'd1;
          stop=6'd1;
          Up=1'd0;
          Down=1'd0;
        end
     end
  end
```

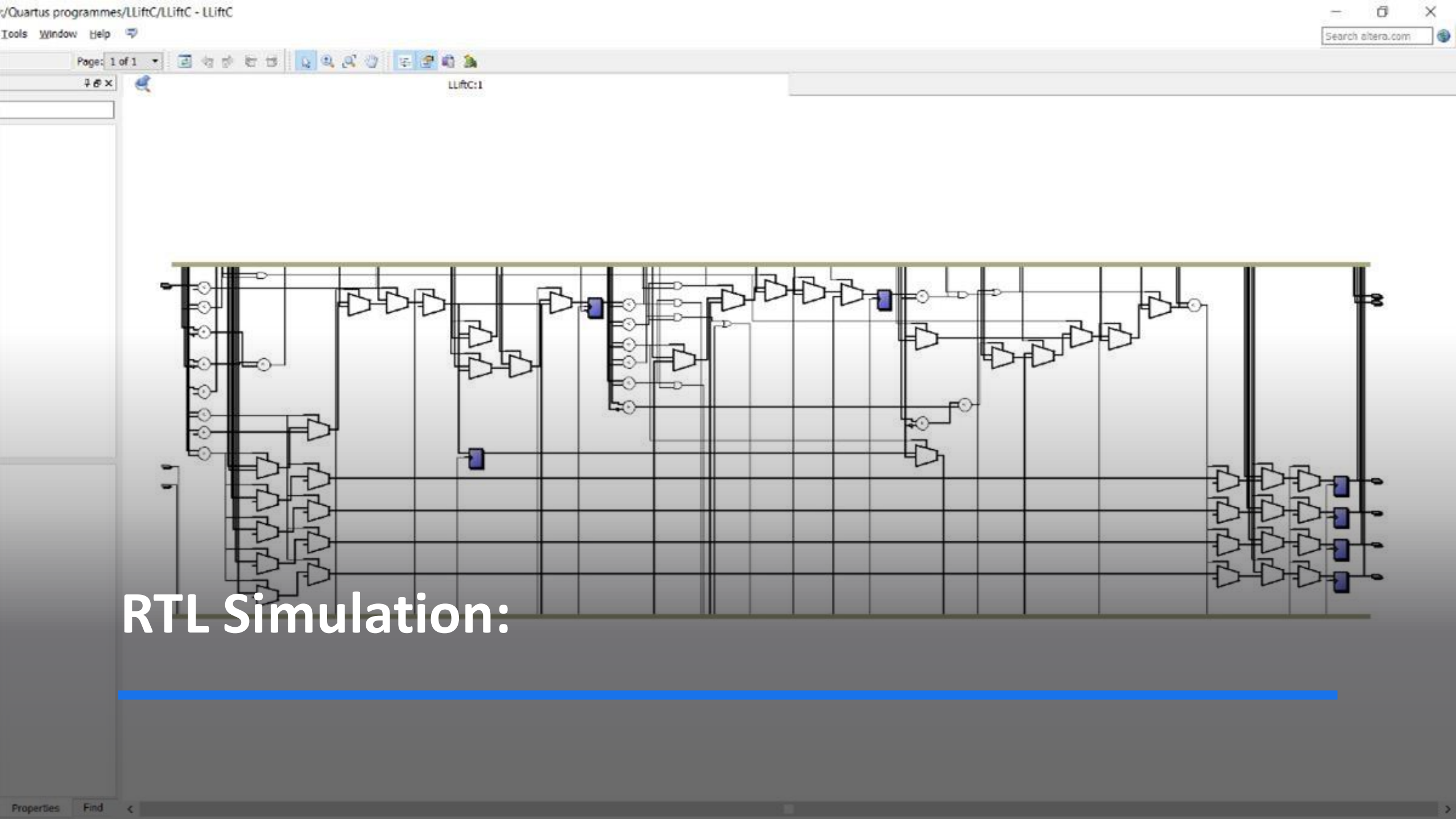
```
if(((l2 > l1) && (l1 > req_floor)) || ((req_floor > l1) && (l1 > l2)))
  l1 = cf;
else if(((l1 > l2) && (l2 > req_floor)) || ((req_floor > l2) && (l2 > l1)))
  l2 = cf;
else if((l1 > req_floor) && (req_floor > l2))
begin
  if((l1 - req_floor) > (req_floor - l2))
  l2 = cf;
else
  l1 = cf;
end
else if((l2 > req_floor) && (req_floor> l1))
begin
  if((l2 - req_floor) > (req_floor - l1))
  l1 = cf;
else
  l1 = cf;
end
else if(l1 == l2)
      l1 = cf;
end
```

```
assign y1 = l1;   // Final position of both the lifts
assign y2 = l2;

endmodule
```


ALTERA STARTER EDITION 10.1d
View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Objects
Name Value Kind Mode
y2 0000010 Net Out
y1 0001100 Net Out
stop 01 Pack... Out
reset St0 Net In
req_floor 0000010 Net In
l2 0000010 Pack... Internal
l1 0001100 Pack... Internal
door 01 Pack... Out
clk St1 Net In
cf 0000010 Pack... Internal
Up 00 Pack... Out
Down 00 Pack... Out
Processes (Active)
Name Type (filtered) State
Wave - Default
Msgs
lift2
lift1
Stop
RESET
Request floor
l2
l1
Door
/LLiftC/clk
/LLiftC/cf
Up
Down
Now
Cursor 1
1.535 ns
Delta: 1
736 ps to 1731 ps

Simulation:



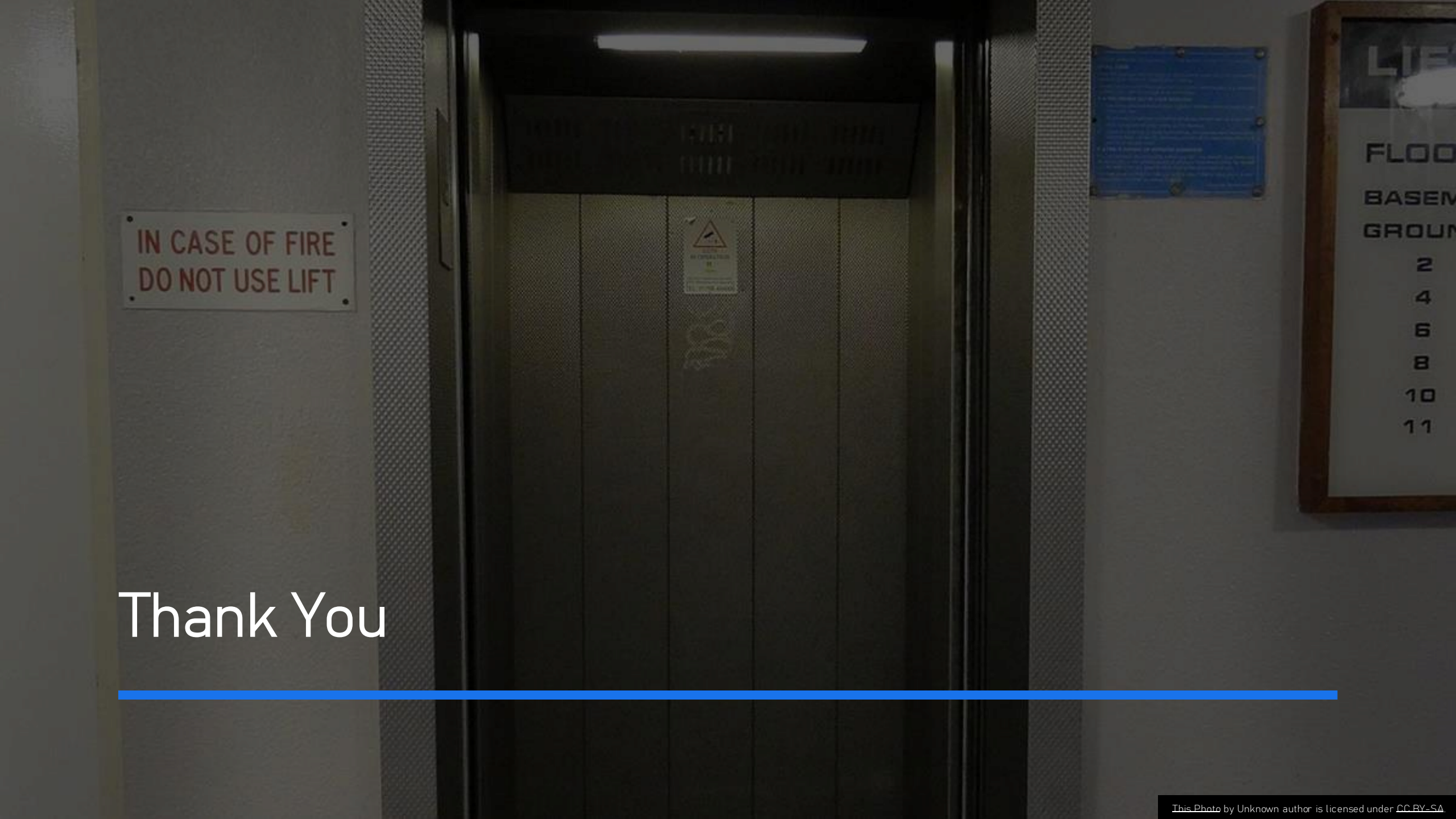
RTL Simulation:

References:

https://www.slideshare.net/visheshsingh19/design-of-elevator-controller-using-verilog-hdl

https://iopscience.iop.org/article/10.1088/1757-899X/225/1/012137/pdf

https://www.youtube.com/watch?v=34q6AIcZvhk&ab_channel=Mr.SunilKumarG.R



Thank You