VLSI System Design

ELE301P

LAB - 9 - Report

Praveen B R

COE19B007

Submission Date: 13/11/2021

INDEX

- ❖ Q1) Lift Design
  - ➢ Objective
  - ➢ Theory
  - ➢ Code
  - ➢ Output/Waveform
  - ➢ Conclusion

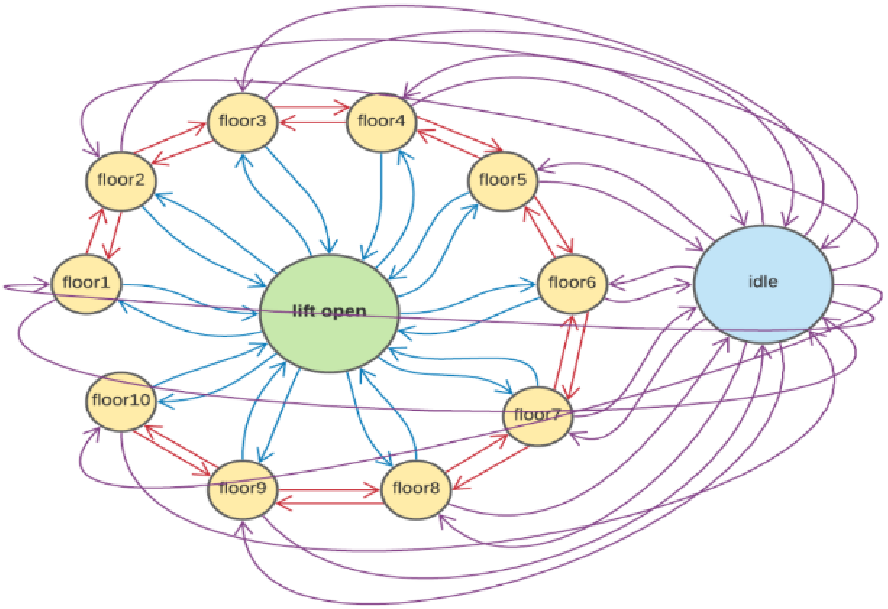
- ❖ Miscellaneous Questions
  - ➢ Question 1
  - ➢ Question 2
  - ➢ Question 3

Q1) Lift Design

Objective:

To design a lift group system consisting of 4 lifts which serves for a 10 story building such that waiting time is minimum.

Theory:



Here, I need to design a lift system consisting of 4 lifts which will be used in a building of 10 floods in such a way that that waiting time is minimum.

My logic is that waiting time is calculated for each floor and to achieve the minimum average waiting time I decided to place lifts on floors 1,4,7,10.

Floor	Waiting Time
1	0
2	1
3	1
4	0
5	1
6	1
7	0
8	1
9	1
10	0

Average Waiting Time = 0.6 units

Lift 1 is on floor 1

Lift 2 is on floor 4
Lift 3 is on floor 7
Lift 4 is on floor 10

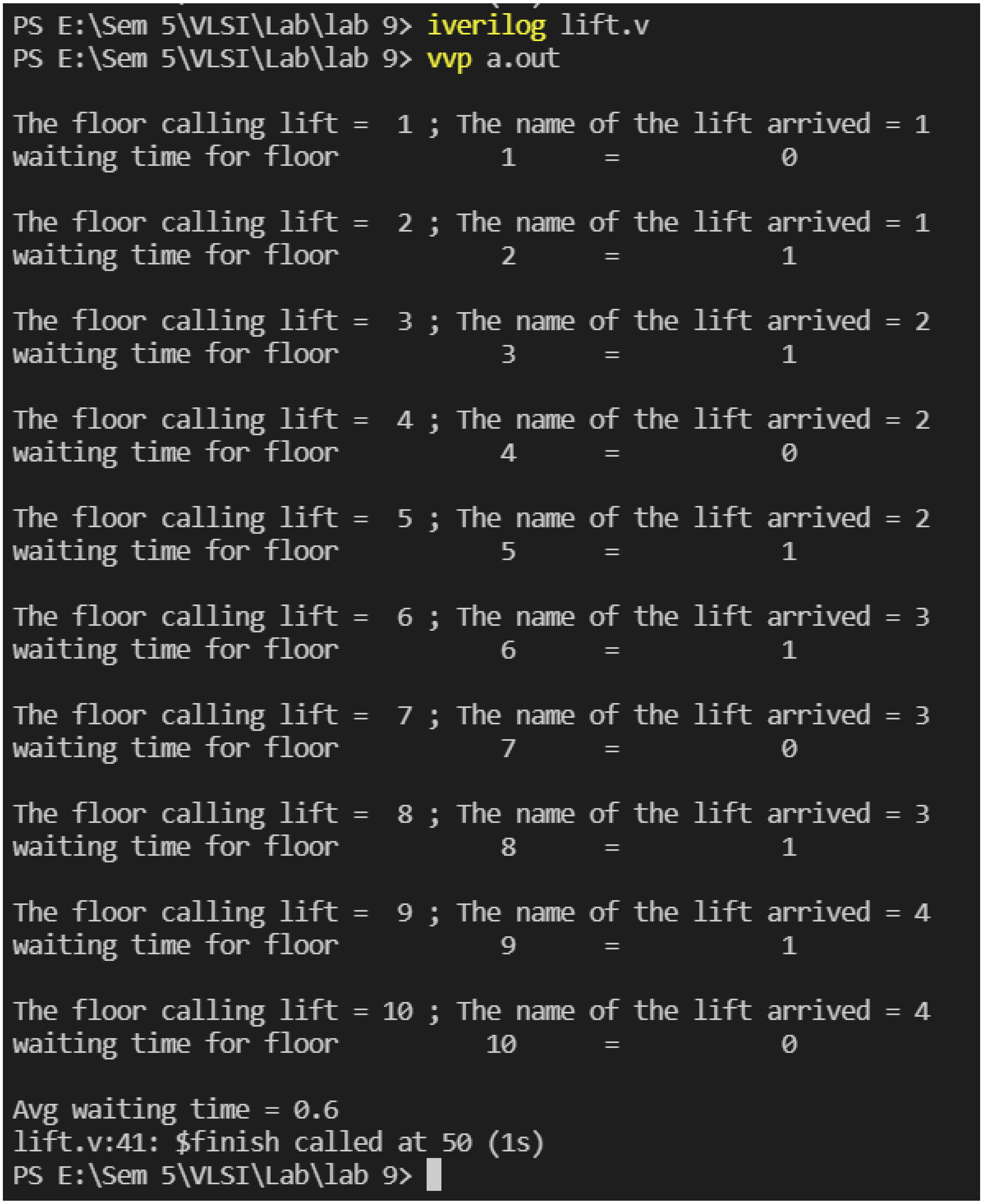
Implementing it in verilog

Code:

```
module Lift_Controller(floor,lift_number);
    input wire [3:0]floor;
    output reg [2:0]lift_number;
    integer count=0, wt=0;

    always @(floor) begin
        if(floor<3) lift_number = 1;
        else if(floor>2 && floor<6) lift_number=2;
        else if(floor>5 && floor<9) lift_number=3;
        else if(floor>8 && floor<11) lift_number=4;


    end

    always @(floor) begin
        count=count+1;
        if(floor==1 || floor==4 || floor==7 || floor== 10)
        wt = 0;
        else
        wt = 1;
        $strobe("waiting time for floor %d      = %d",count,wt);
    end
endmodule

module lift_tb;
    reg [3:0]floor;
    wire [2:0]lift_number;

    initial begin
        floor=1; #5  // ground floor
        floor=2; #5
        floor=3; #5
        floor=4; #5
        floor=5; #5
        floor=6; #5
        floor=7; #5
        floor=8; #5
        floor=9; #5
        floor=10; #5  // top floor
        $display("\nAvg waiting time = 0.6");
    $finish;
    end

    Lift_Controller LC1 (floor,lift_number);
    initial
    begin
        $dumpfile("lift.vcd");
        $dumpvars(0,lift_tb);
    end
    initial
    begin
        $monitor("\nThe floor calling lift = %d ; The name of the lift
arrived = %d",floor,lift_number);
    end

endmodule
```

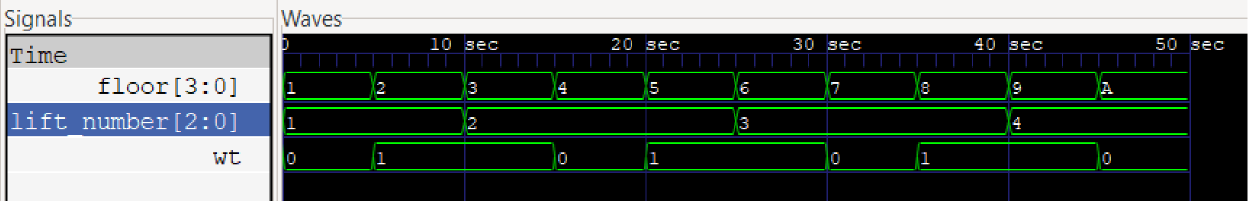
Output/Waveform:

Terminal



```
PS E:\Sem 5\VLSI\Lab\lab 9> iverilog lift.v
PS E:\Sem 5\VLSI\Lab\lab 9> vvp a.out

The floor calling lift =  1 ; The name of the lift arrived = 1
waiting time for floor          1       =            0

The floor calling lift =  2 ; The name of the lift arrived = 1
waiting time for floor          2       =            1

The floor calling lift =  3 ; The name of the lift arrived = 2
waiting time for floor          3       =            1

The floor calling lift =  4 ; The name of the lift arrived = 2
waiting time for floor          4       =            0

The floor calling lift =  5 ; The name of the lift arrived = 2
waiting time for floor          5       =            1

The floor calling lift =  6 ; The name of the lift arrived = 3
waiting time for floor          6       =            1

The floor calling lift =  7 ; The name of the lift arrived = 3
waiting time for floor          7       =            0

The floor calling lift =  8 ; The name of the lift arrived = 3
waiting time for floor          8       =            1

The floor calling lift =  9 ; The name of the lift arrived = 4
waiting time for floor          9       =            1

The floor calling lift = 10 ; The name of the lift arrived = 4
waiting time for floor         10       =            0

Avg waiting time = 0.6
lift.v:41: $finish called at 50 (1s)
PS E:\Sem 5\VLSI\Lab\lab 9>
```

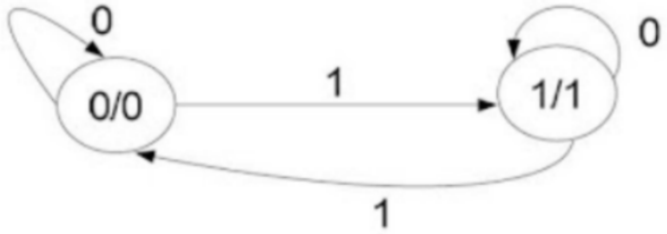
Waveform



Conclusion:

Thus, a lift group system consisting of 4 lifts which serves for a 10 story building such that waiting time is minimum has been successfully implemented.

Miscellaneous Questions



Question 1 What is the output sequence for the given input data sequence 001010110110110111?

Let us name the state 0/0 as A and 1/1 as B.

Since the start state is not mentioned in the above figure, we will find the output sequence for both A and B as the start states.

If we consider A as the start state , then for the input sequence 001010110110110111 we get output as 001100100100100101.

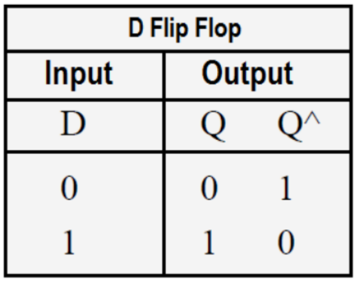
If we consider B as the start state , then for the input sequence 001010110110110111 we get output as 110011011011011010

Question 2 What is the behaviour of the above Finite state machine?

Clearly,outputs of the FSM are given **inside** the states so we can say that it only depends on the current state. So the given FSM is of Moore type.

Question 3 Decide the Flip-flop which can be used by avoiding external logic gates.

We can represent the given FSM by avoiding external logic by using D flip flop as it gives high out when input is high and low out when input is low



D Flip Flop		
Input	Output	
D	Q	Q^
0	0	1
1	1	0