B. Tech. 3rd Semester

Design of Digital Systems (CS201)

Mini-Project Final Report

On

Advanced Elevator Control System

Akash Ravi
Roll No: 211CS207
Phone no: 7892889801
E-mail: akashravi.211cs207@nitk.edu.in

Navaneet Y V N
Roll no: 211CS234
Phone no: 9886838143
E-mail: navaneetyvn.211cs234@nitk.edu.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,

SURATHKAL, MANGALORE – 575025

January, 2023

1. Abstract

PROBLEM STATEMENT

Multi-storied buildings now-a-days generally have elevator systems with them. These elevator systems need precise control to carry out the task assigned to them. In many multi storeyed buildings, elevators have restricted access(Corporate settings, Hospitals, or some Residential complexes). Sometimes elevators are restricted for the use of the employees of that particular office only. In hospitals, certain lifts are meant to be used by patients only, and some other lifts are meant to be used by the hospital staff to move around equipment, stretchers etc. In some educational institutions, some lifts are meant for faculty use only. But this system is very rarely followed. Anyone and everyone who sees an elevator is naturally inclined to use it. Due to such reasons, when in case of an emergency, it causes delay in the case of a medical emerngency which could very well be the difference between life and death.

It is also observed that the lights and fans in the lift are left running when there is nobody using the lift. This wastes a lot of power and people are either in a hurry and usually forget to switch them off or they are just too lazy to do it and don't care about the power being wasted. In complexes with multiple elevators, not all the elevators are used at the same time. In such cases, if all the elevators are running, a lot of power is wasted in just keeping the elevators switched on.

SOLUTION

Along with the regular functionality of an elevator our idea is to implement the restricted access model in the elevators by implementing elevator access only via a passcode. In cases where time is key, entering the passcode is time consuming and undesirable. For such cases, we shall use an RFID key instead of a passcode so that the time spent in entering a passcode can be eliminated.

In order to save power by the smart use of lights and fans, we shall use a minimum weight condition to keep the lights and fan running. There will also be a mechanism to turn the lights and fan on and off manually. To implement a sleep mode, the lift shall power itself off after a set time of the day.

2. Introduction

The problem at hand is that we have to design an advanced elevator control system that makes sure that the access to the elevator is restricted. In this way, when such an elevator is used in places like hospitals, for example, only people with access shall be allowed to use the elevator and won't have to wait for someone else to use the elevator. This feature can actually be life-saving when someone has to use the elevator in case of an emergency. In places like an educational institutions, some elevators are allowed to be used by the faculty and staff only and not for the students, In such places, this elevator control system shall be found handy.

Almost all the people leave the lights and fan in an elevator running after using the elevator. This is a huge wastage of energy and hence something could be done to prevent this wastage of energy. Thus we shall use a minimum wight condition to detect the presence of a person in a lift and iff the module detects the presence of someone in the elevator, the lights and fan are switched on. When nobody is using the lift, the lights and fans are automatically switched off. This weight check shall also be used to ensure that the lift is not carrying excessive load. The lift shall not move when there is excessive load for safety purposes.

Another reason for wastage of energy is that the lift is kept on when nobody shall be using it for a long time. In such cases we can use something called a sleep mode to disable the lift automatically and save power. For this elevator control system that we are developing, we have decided to put the elevator to sleep mode between 2200 hrs and 0600 hrs which is ideally the time for most of the people to sleep and hence when the lift is not mostly being used.

Coming to how this elevator control system shall work, firstly the user shall have to enter the password to access the lift. We shall use a 7 bit binary number to represent the password for now. Once the password is verified, the user will be able to request a lift call to either go up or down according to the situation.

Once the user is inside the elevator, he/she has to press the close button before the elevator can go anywhere. Only after the doors are closed, the elevator shall start moving. The system takes care of the priority of calls as well. For example, if there is a call for the lift in the 2^{nd} floor to go up and in the 3^{rd} floor to go up, while the lift is initially at the ground floor. In this case, the elevator shall go to the 3^{rd} floor and serve the up call while ignoring the down call in the 2^{nd} floor. Once the call is served and the direction is changed to down, the lift shall come down to second floor and serve the call.

Coming to the other features of the elevator, once a user enters the elevator, the minimum weight condition comes into play. If the user is above 30kg, the lights and fan will automatically turn on. If the weight is above 480kg, the elevator shall show an overweight warning and stop working. Also when the clock strikes 2200, the elevator stops working and it only starts working at 0600 hrs in the morning. But there is also an option to manually disable the sleep mode and use the elevator in emergency situations. The same thing goes with the lights and fan switch as well. If by chance there is a user who fails to meet the minimum weight condition but is still a real user, then he/she can manually switch on the lights and fan.

Let's see how the design of this elevator system looks like.

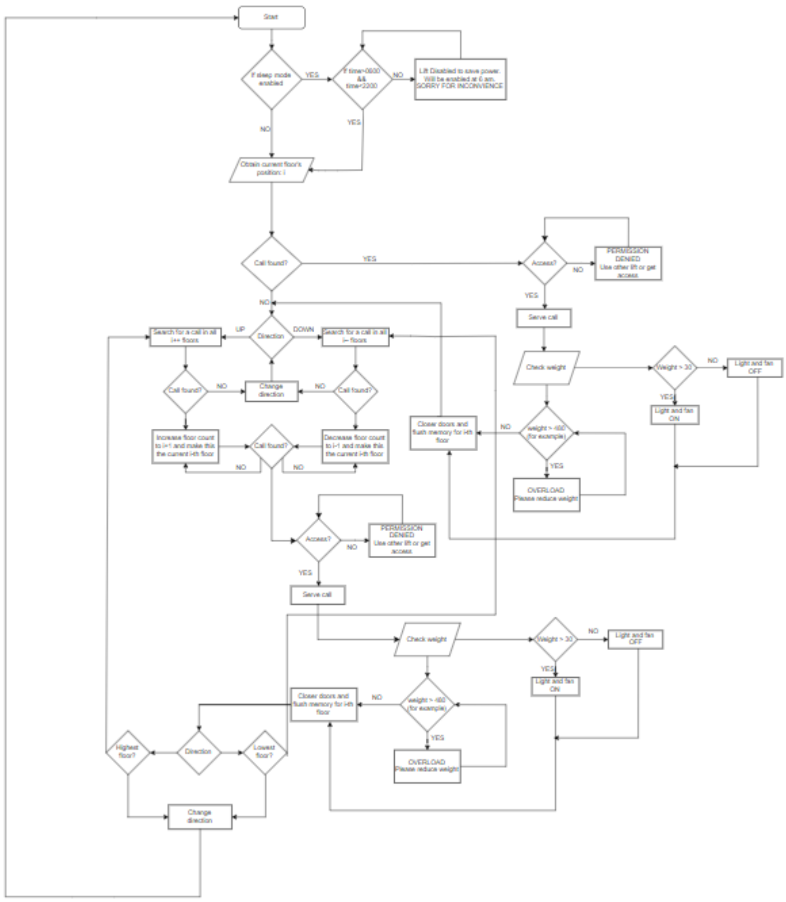
3. Design

WORKING PRINCIPLE

The working of the control system can be described in the following steps:

- Initially, the elevator system should find the current position of the elevator relative to the building structure.
- Next, the system should check for the presence of any call in the current floor.(i). If any call is found, then the elevator should serve the call and erase the call from its memory so that the same call is not served again.
- Then, depending on the current direction of motion of the elevator, the system should search for calls in floors above of below the current floor accordingly.
- If there's a call found in the direction of motion of the elevator, it should either increase or decrease it's floor count by 1 and repeat the 2nd step till the call is served by the elevator.

This above process can also be shown in the form of a flow chart as below:



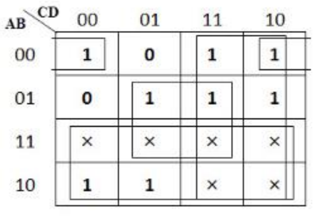
COMPONENTS REQUIRED

- Clock
- Shift registers
- Decoders
- Priority encoder
- JK Flip Flops
- T Flip Flops
- D Flip Flops
- Comparator
- Demultiplexer
- Multiplexer
- 7 segment display
- LEDs
- Basic logic gates

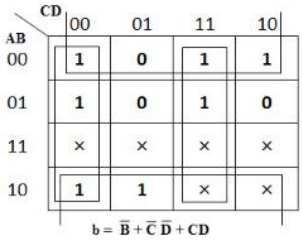
Turth tables and Expressions

- **7 segment display**

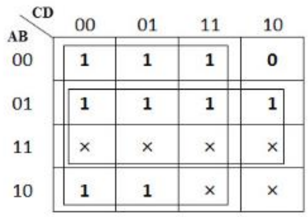
Digit	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0



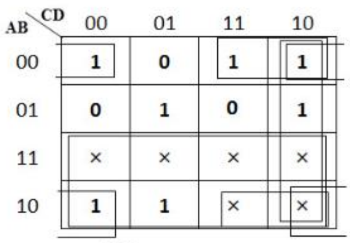
$a = A + C + BD + \overline{B}\overline{D}$



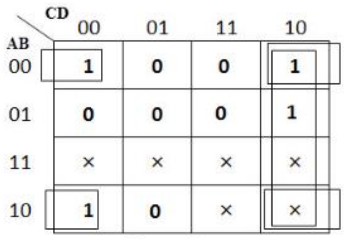
$b = \overline{B} + \overline{C}\,\overline{D} + CD$



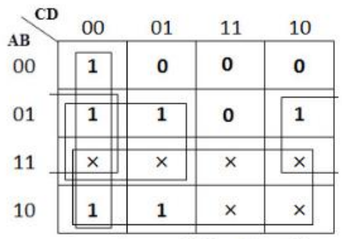
$c = B + \overline{C} + D$



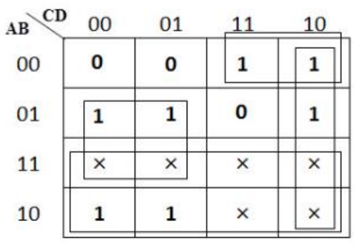
$d = \overline{B}\,\overline{D} + C\overline{D} + B\overline{C}D + \overline{B}C + A$



$e = \overline{B}\,\overline{D} + C\overline{D}$



$f = A + \overline{C}\,\overline{D} + B\overline{C} + B\overline{D}$



$g = \overline{B}C + C\overline{D} + B\overline{C} + B\overline{C} + A$

Expressions

$$a = A + C + BD + \overline{B}\,\overline{D}$$

$$b = \overline{B} + \overline{C}\,\overline{D} + CD$$

$$c = B + \overline{C} + D$$

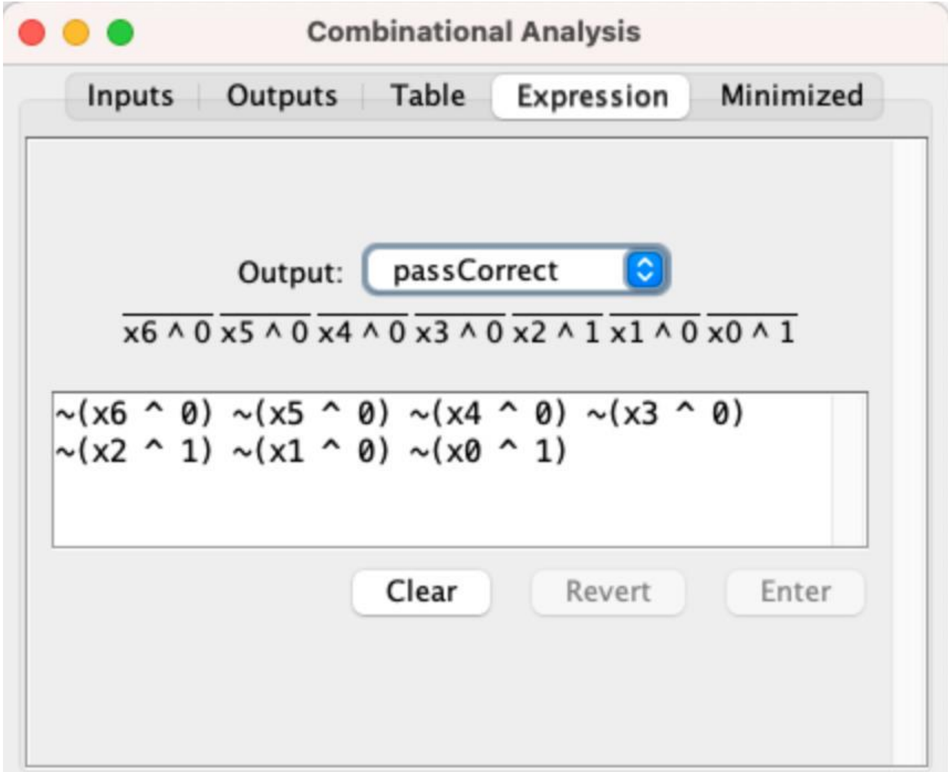
$$d = \overline{B}\,\overline{D} + C\,\overline{D} + B\,\overline{C}\,D + \overline{B}\,C + A$$

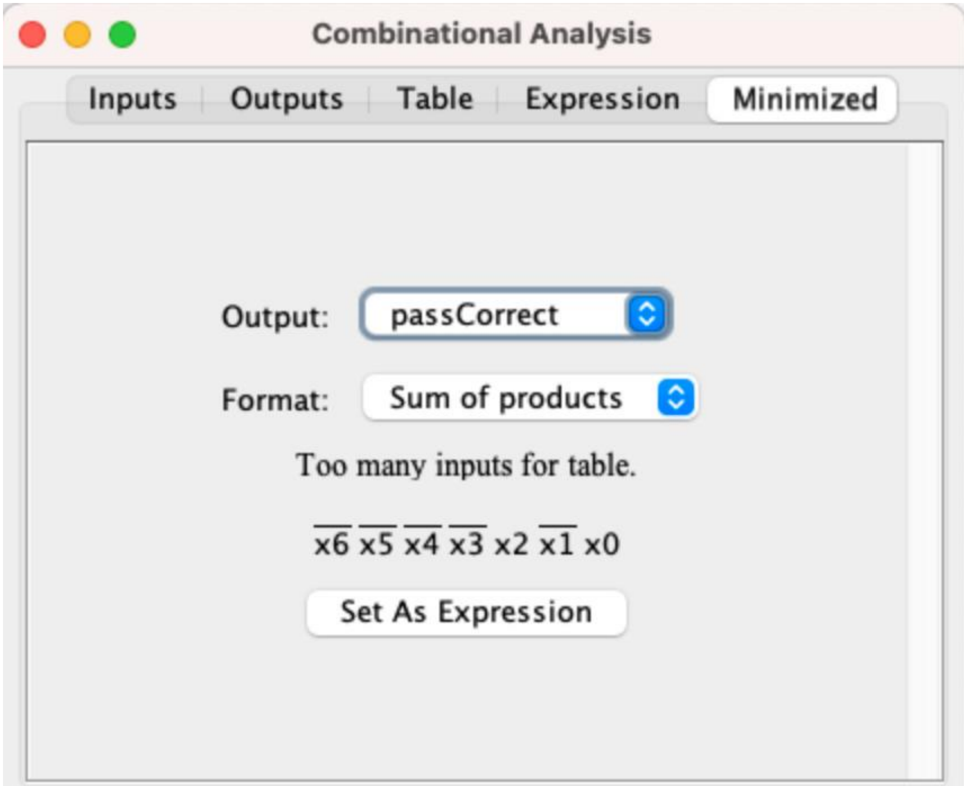
$$e = \overline{B}\,\overline{D} + C\,\overline{D}$$

$$f = A + \overline{C}\,\overline{D} + B\,\overline{C} + B\,\overline{D}$$

$$g = A + B\,\overline{C} + \overline{B}\,C + C\,\overline{D}$$

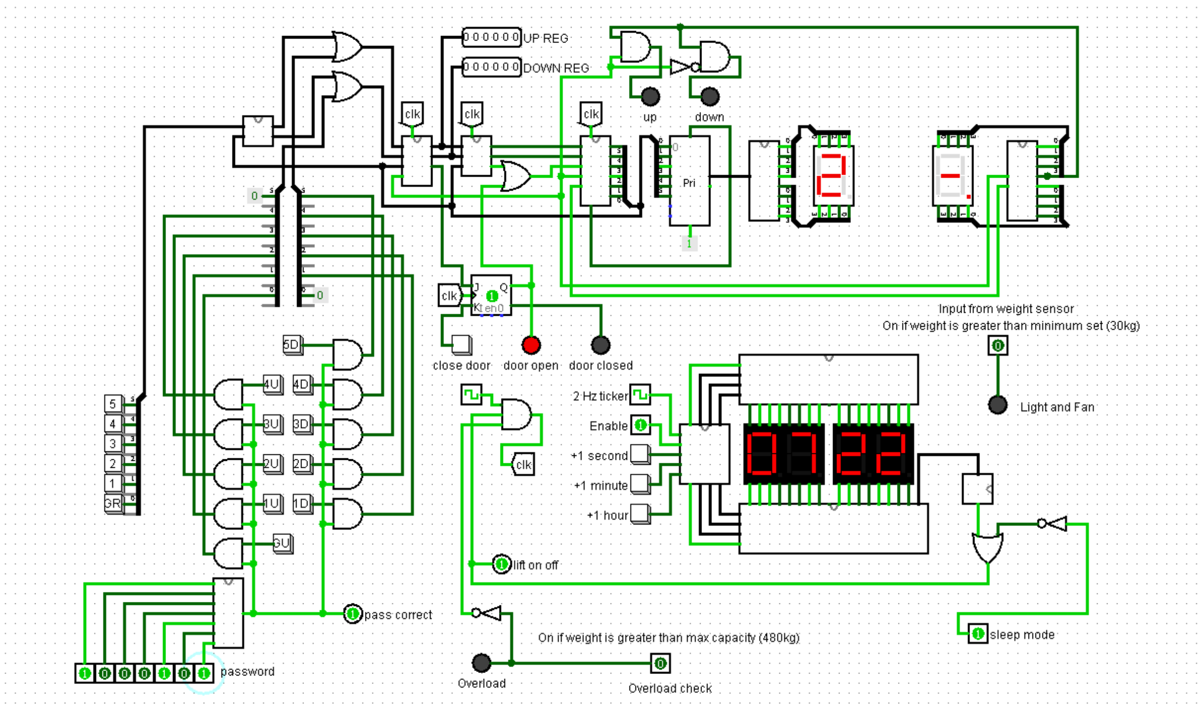
Password Check:





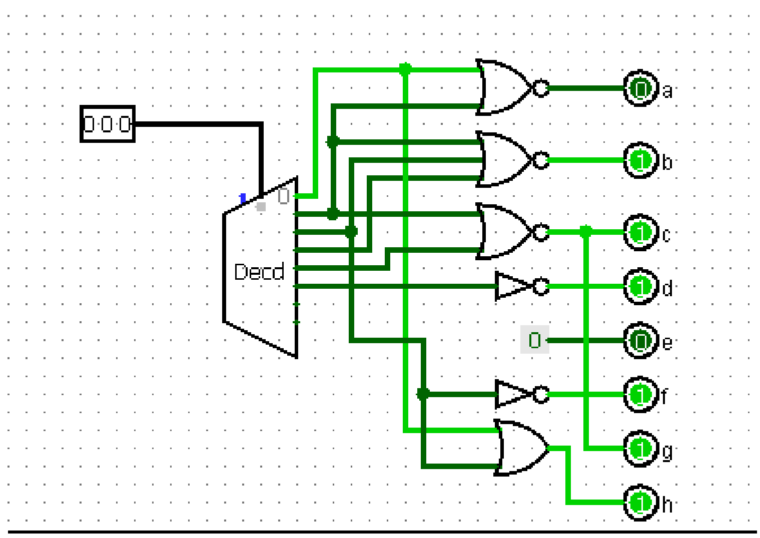
LOGISIM CIRCUIT

- Main Circuit



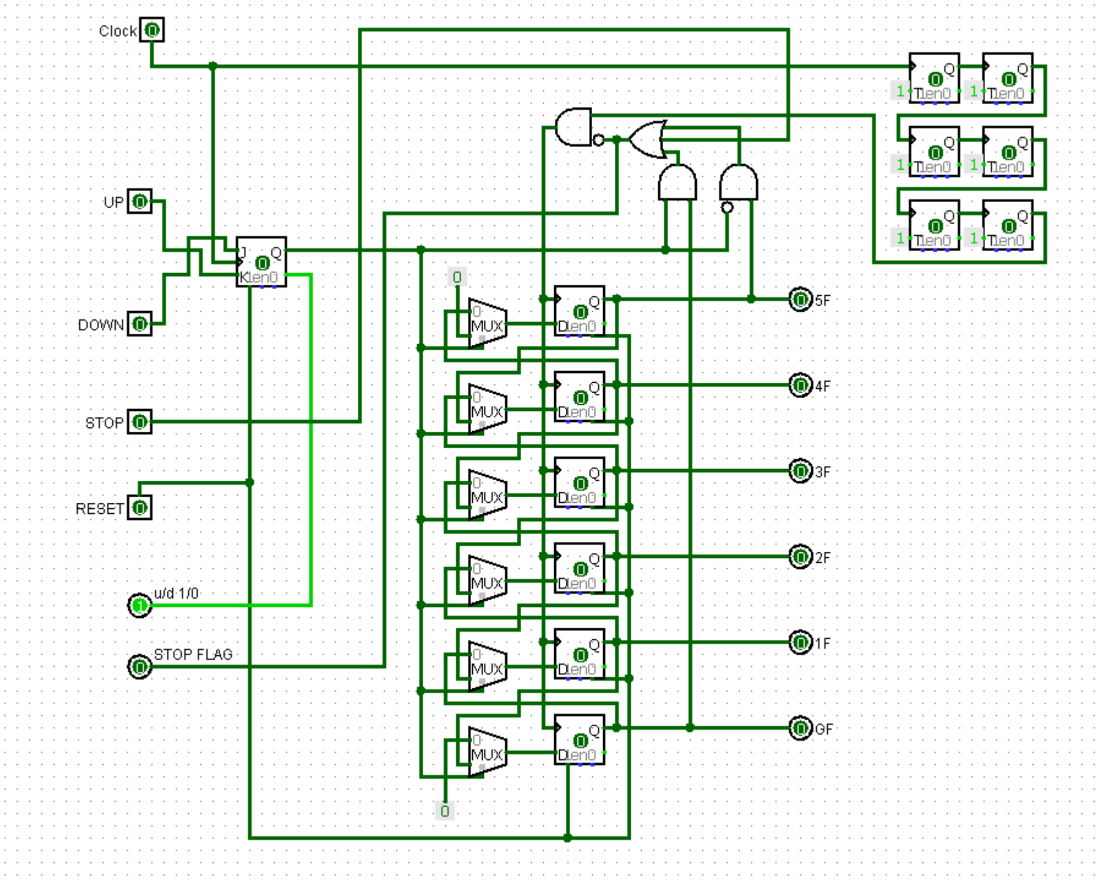
SUB-CIRCUITS :

- Seven Segment Display

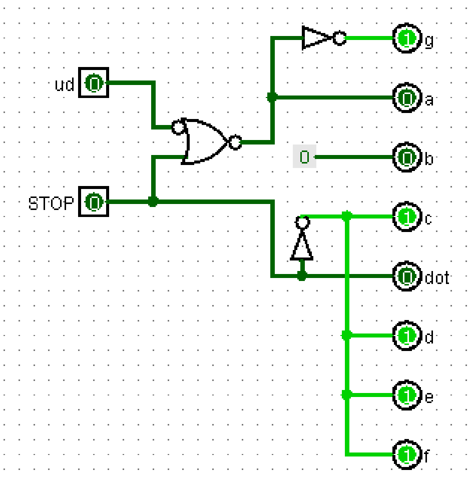


This circuit takes in the floor position as input and enabales the segments of the seven segment display to display the floor number accordingly.

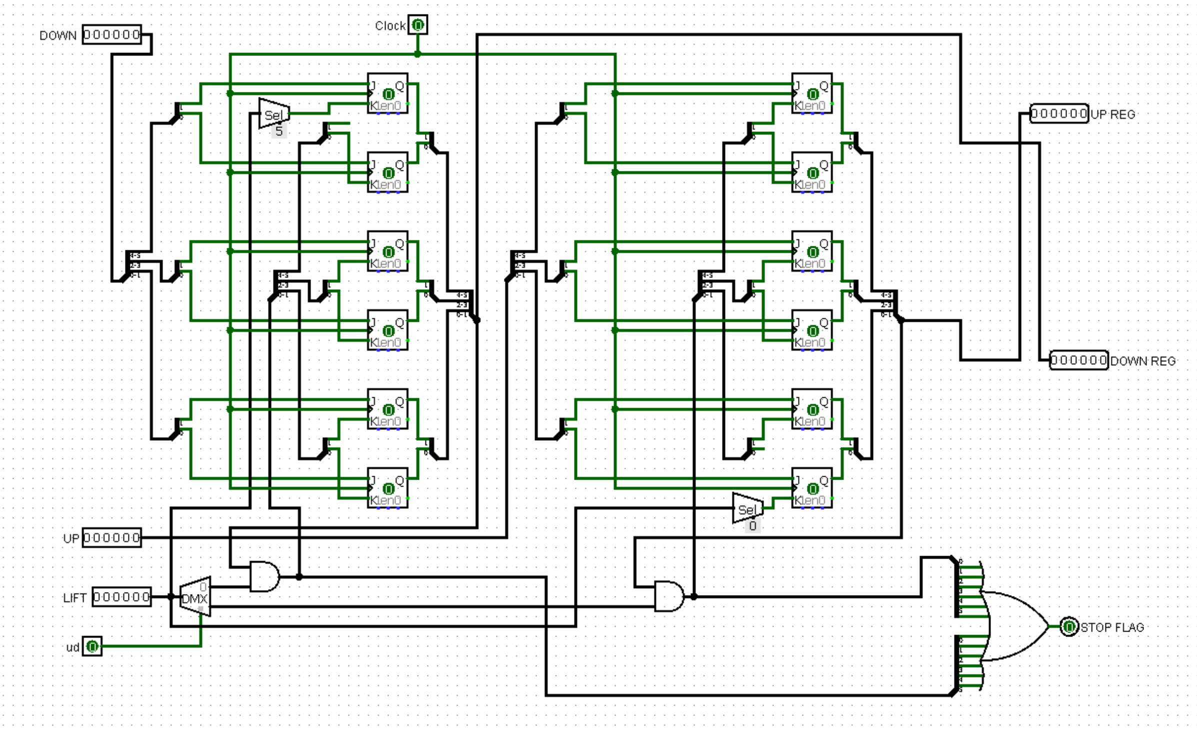
- **Lift Shift register**



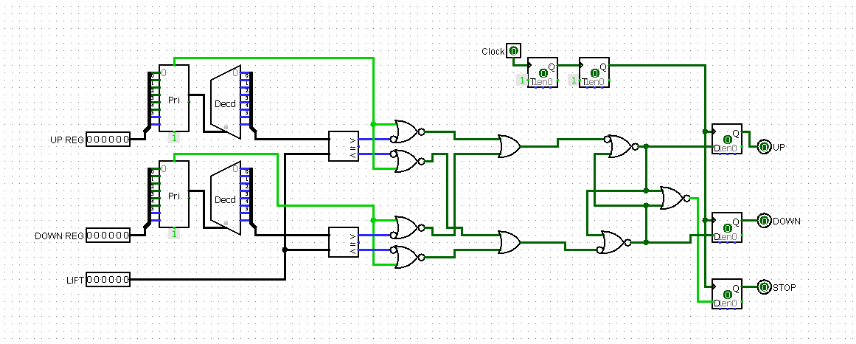
- **Up-Down Encoder to &-segment**



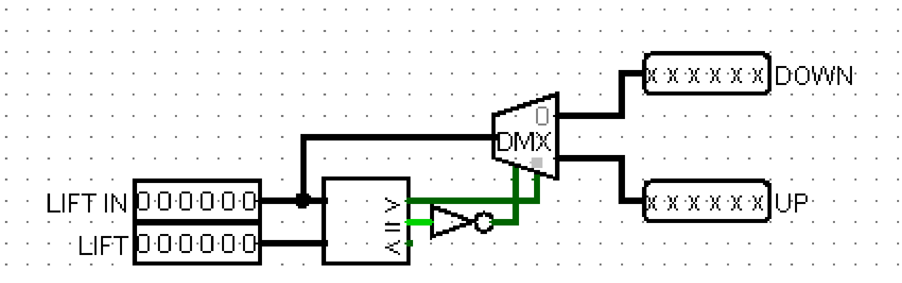
- Call register



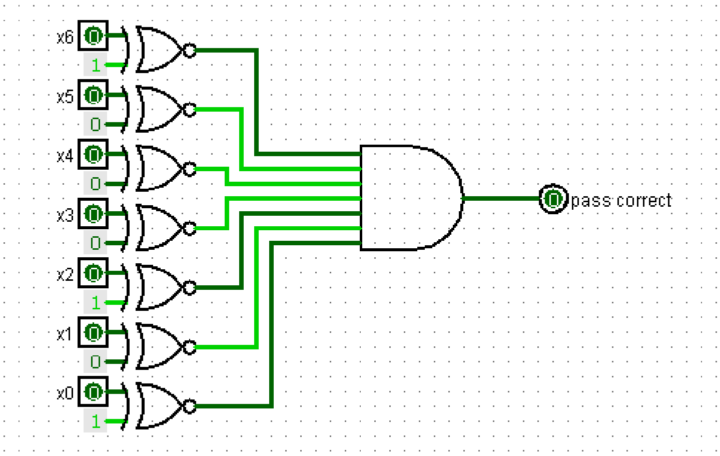
- Up-Down Comparator



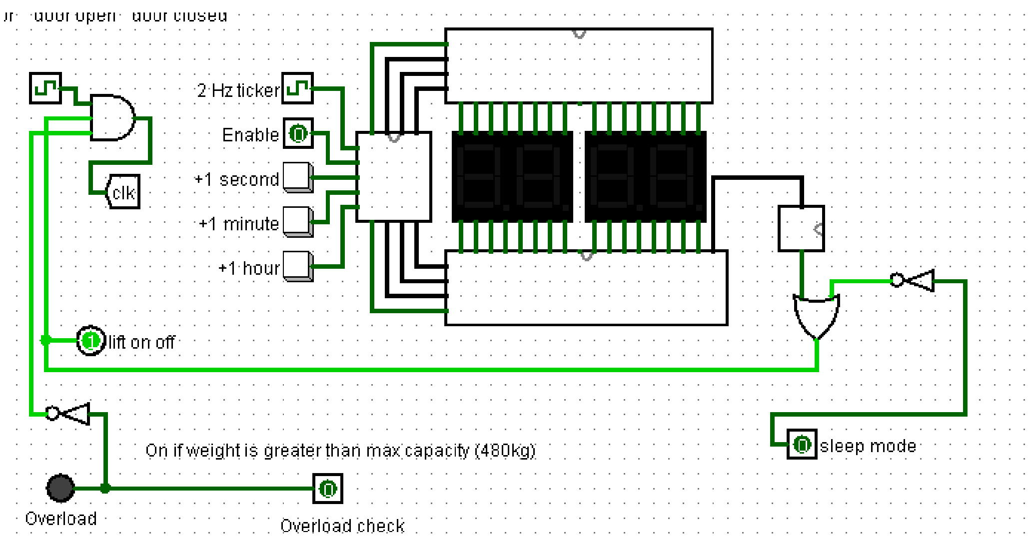
- Relative position Comparator



- Password module



- Sleep-mode module



4. Verilog Code

Method used: Behavioural modelling

```
module elevator(input clk, reset, input [3:0] floor_sel, input[5:0]Time, input
 up, input down, output reg [3:0] floor, input [10:0]weight, input[6:0]password, output reg light_fan, output
reg limit, output reg sleepmode, output reg passwordcheck, output reg [3:0] direction);
  reg [3:0] next_floor;
  integer i;

  always @(posedge clk or negedge reset) begin
    if (reset) begin
      floor <= 4'b0000;
      end
    else begin
      if (next_floor == floor_sel) begin
        direction <= 4'b0000;
      end else if (next_floor > floor_sel) begin
        direction <= 4'b0001;
      end else begin
        direction <= 4'b0010;
      end
    end
  end

  always @(*) begin
    if((Time>=6) && (Time<= 22) ) begin
      sleepmode <= 1'b0;
      if (password == 69) begin
          passwordcheck <= 1'b1;
      floor <= floor_sel;
      end
      else begin
      passwordcheck <= 1'b0;
      end
      if((weight<=480)) begin
        light_fan <= 1'b1;
        limit <= 1'b0;
        if(weight<30)begin
          light_fan <= 1'b0;
          limit <= 1'b1;
        end
        if (reset) begin
          next_floor <= 4'b0000;
        end else if (up) begin
          next_floor <= 4'b0000;
```

```
        for (i = 0; i < 4; i = i + 1) begin
          if (floor_sel[i] == 1'b1) next_floor <= i;
        end
      end else if (down) begin
        next_floor <= 4'b0000;
        for (i = 3; i >= 0; i = i - 1) begin
          if (floor_sel[i] == 1'b1) next_floor <= i;
        end
      end
    end
    else  begin
      light_fan <= 1'b1;
      limit <= 1'b1;
    end
  end

  else begin
    if(password != 69 && ((Time>=6) && (Time<= 22)))begin
      passwordcheck <= 1'b0;
      sleepmode <= 1'b0;
    end
    else if(( password !=69 ) && (Time>22 || Time<6)) begin
      passwordcheck <= 1'b0;
      sleepmode <= 1'b1;
    end
    else begin
      passwordcheck <= 1'b1;
      sleepmode <= 1'b1;
    end
      floor <= 4'b0000;
      end
 end

endmodule

module elevator_testbench;
 reg clk, reset, up, down;
 reg[5:0] Time;
 reg[10:0] weight;
 reg[6:0] password;
 reg [3:0] floor_sel;
 wire [3:0] floor , direction;
 wire light_fan, limit, sleepmode, passwordcheck;

 elevator dut(clk, reset, floor_sel, Time, up, down, floor, weight, password, light_fan, limit, sleepmode,
passwordcheck, direction);

 initial begin
```

```
    clk = 0;
    forever #5 clk = ~clk;
  end

  initial begin
    $dumpfile("elevator.vcd");
    $dumpvars();
    $monitor("\n\nFloor sleceted :%d Time :%02d Weight :%d Password :%b Floor :%d \nLights and fan
:%b Sleep mode :%b Overweight/Underweight: %b Password check :%b ", floor_sel, Time, weight,
password, floor, light_fan, sleepmode, limit, passwordcheck );
    reset = 1;
    Time = 8;
    up = 0;
    down = 0;
    #10 reset = 0;

    floor_sel = 2;
    password = 69;
    weight = 80;
    up = 1;
    #5 up = 0;

    #20 floor_sel = 4'b0100;
    password = 68;
    weight = 80;
    up = 1;
    #5 up = 0;
    Time = 9;

    #20 floor_sel = 4'b0010;
    password = 69;
    weight = 481;
    down = 1;
    #5 down = 0;
    Time = 13;

    #20 floor_sel = 4'b0100;
    password = 69;
    weight = 8;
    up = 1;
    #5 up = 0;
    Time = 23;

    #20 floor_sel = 4'b0001;
    password = 69;
    weight = 80;
    down = 1;
    #5 down = 0;
```

```
    Time = 7;

    #20 floor_sel = 4'b1000;
    password = 69;
    weight = 80;
    up = 1;
    #5 up = 0;
    Time = 9;

    #20 floor_sel = 4'b0100;
    password = 69;
    weight = 80;
    down = 1;
    #5 down = 0;
    Time = 23;

    #20 floor_sel = 4'b0100;
    password = 69;
    weight = 80;
    down = 1;
    #5 down = 0;
    Time = 2;

    #20 floor_sel = 4'b1000;
    password = 69;
    weight = 80;
    up = 1;
    #5 up = 0;
    Time = 9;

    #20 floor_sel = 4'b0010;
    password = 60;
    weight = 80;
    down = 1;
    #5 down = 0;
    Time = 13;
    #20 $finish;
  end
endmodule
```

Output:

```
Floor sleceted : x Time :08 Weight :   x Password :xxxxxxx Floor : x
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 1 Password check :0


Floor sleceted : x Time :08 Weight :   x Password :xxxxxxx Floor : 0
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 1 Password check :0


Floor sleceted : 2 Time :08 Weight :  80 Password :1000101 Floor : 2
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 0 Password check :1


Floor sleceted : 4 Time :08 Weight :  80 Password :1000100 Floor : 2
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 0 Password check :0


Floor sleceted : 4 Time :09 Weight :  80 Password :1000100 Floor : 2
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 0 Password check :0


Floor sleceted : 2 Time :09 Weight : 481 Password :1000101 Floor : 2
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 1 Password check :1


Floor sleceted : 2 Time :13 Weight : 481 Password :1000101 Floor : 2
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 1 Password check :1


Floor sleceted : 4 Time :13 Weight :   8 Password :1000101 Floor : 4
Lights and fan :0 Sleep mode :0 Overweight/Underweight: 1 Password check :1


Floor sleceted : 4 Time :23 Weight :   8 Password :1000101 Floor : 0
Lights and fan :0 Sleep mode :1 Overweight/Underweight: 1 Password check :1


Floor sleceted : 1 Time :23 Weight :  80 Password :1000101 Floor : 0
Lights and fan :0 Sleep mode :1 Overweight/Underweight: 1 Password check :1
```

```
Floor sleceted : 1 Time :07 Weight :  80 Password :1000101 Floor : 1
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 0 Password check :1


Floor sleceted : 8 Time :07 Weight :  80 Password :1000101 Floor : 8
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 0 Password check :1


Floor sleceted : 8 Time :09 Weight :  80 Password :1000101 Floor : 8
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 0 Password check :1


Floor sleceted : 4 Time :09 Weight :  80 Password :1000101 Floor : 4
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 0 Password check :1


Floor sleceted : 4 Time :23 Weight :  80 Password :1000101 Floor : 0
Lights and fan :1 Sleep mode :1 Overweight/Underweight: 0 Password check :1


Floor sleceted : 4 Time :02 Weight :  80 Password :1000101 Floor : 0
Lights and fan :1 Sleep mode :1 Overweight/Underweight: 0 Password check :1


Floor sleceted : 8 Time :02 Weight :  80 Password :1000101 Floor : 0
Lights and fan :1 Sleep mode :1 Overweight/Underweight: 0 Password check :1


Floor sleceted : 8 Time :09 Weight :  80 Password :1000101 Floor : 8
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 0 Password check :1


Floor sleceted : 2 Time :09 Weight :  80 Password :0111100 Floor : 8
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 0 Password check :0


Floor sleceted : 2 Time :13 Weight :  80 Password :0111100 Floor : 8
Lights and fan :1 Sleep mode :0 Overweight/Underweight: 0 Password check :0
elevator_final.v:170: $finish called at 260 (1s)
```

5. Conclusions and Future Work

The present model of the elevator control system is a very basic model with basic implementation of password check, sleep mode and automatic lights and fans. Maybe in the future, we can improve the precision and work on the details of this elevator control system. Maybe we can increse the complexity of the password to increase the security protocol or we can also make the password alphanumeric than just a passcode. A more advanced password check module could be developed where the access is governed thorugh rfid tags and if we go further into modernizing the module, maybe we can ensure access through biometrics.

To make the sleep mode more effective, maybe we can also find a way to integrate this design with a machine learning module to learn the usage pattern of the lift and go into sleep mode when the model puts the lift into sleep mode based on its learnings. To improve the accuracy of the working of the automated lights and fan system, we can make use of a sensor that detects if a person enters or leaves the lift and hence use this to switch the lights and fan on and off. An even more effective way of doing this would be using an artificial intelligence tool to detect the presence of a human and accordingly operate the lights and fan. Using an AI model will also enable us to make the lights and fan smart, depending on the temperature and other factors.

6. References

- A Simulation Study of an Elevator Control System using Digital Logic
- https://www.youtube.com/watch?v=eDzSo_VpeCQ
- https://www.ijettjournal.org/2017/volume-52/number-3/IJETT-V52P222.pdf