**A VLSI implementation of Elevator control based on Finite state machine using Verilog**

Ayush A. Pawar

Under Dr. Babita Jajodia

*VLSI project, Department of ECE,*

*Indian Institute of Information Technology, Guwahati*

*Opp. IT Park Street, Bongora, Guwahati, Assam, India. 781015.*

*Email:* [*pawar.anjuman@iiitg.ac.in*](mailto:pawar.anjuman@iiitg.ac.in)

**Abstract:**

In today’s world, lifts are the basic need of any building, whether it be any office or a shopping mall. It encourages the quicker development of individuals and gear between floors. Normally lifts are intended for a particular building considering the fundamental factors, for example, the tallness of the building, the quantity of individuals venturing out to each floor and the normal times of high utilization. The lift framework is composed with various control procedures. This usage depends on FPGA, which can be utilized for a working with any number of floors, with the predetermined sources of info and yields. This controller can be executed for a lift with the required number of floors by just changing a control variable in the code. This approach depends on a calculation which decreases the measure of calculation required, by concentrating just on the pertinent guidelines that enhances the execution of the gathering of lift framework.

1. **Introduction & Objective**

An elevator or lift is transport equipment which is located vertically in the building to take luggage or passenger from one floor to another. Mostly elevators are power-driven by electric motors. Elevators plays vital role in both commercial and residential locations. The main operational principal of elevators is the conversion of electrical energy into mechanical energy. The early elevators were considered as cabs and driven by hands or by animals or by rope. The creation of screw drive based system was the significant step in elevator’s designs, which finally led to the design of latest modern elevators

Traditional elevator control system is based on Relay logic, PLC and Microcontroller etc. but these systems have reduced number of inputs and outputs. With the advancement of technology electronics devices are becoming much smaller, flexible, efficient and easy to use. These electronic devices are capable of implementing various functions. These electronic devices have various properties and are helpful to man in day to day life. Moore's law stated that the transistors double in size every year. With the increase in number of the transistors the size of electronics device shrinks. With the decrease in size of devices and increase in number of transistors. FPGA technology is the best replacement of Relay logic, PLC and Microcontroller due to its flexibility, lower cost, efficiency, security, operational speed and parallel processing. For the implementation of different hardware architectures, the option of reconfiguring the FPGAs by software makes it better option.

A state diagram is simply used to explain the state machine automation graphically. State Machines are those devices whose output is not only based on the output from recent inputs but also from previous inputs. The elevator control system is basically finite state machine (FSM). FSM is a digital sequential circuit that consists on different defined states that are controlled by inputs. The advantage of state machine is that they can be used to determine the behavior of the system based on the control variable. These are classified into two types Mealy State Machine and Moore State machine.

In Mealy state machine present state depends upon both present state outputs and previous state outputs where as in case of Moore state machine present state depends only on recent input. The given *elevator control system is based on Mealy machine.*

In this report we basically focus on *designing the lift controller for a 40-stored building by using state machine concept*.

1. **Algorithms**

Our Basic Logic is based on Mealy state machine. Here, we

* At any floor when input button is pressed to reach any desired floor the state will be changed.
* According to this state the motor moves the elevator car upward or downward.
* As car reaches the desired floor, sensor generates an input to change the state.
* At this state output “01” for stop will be generated which will stop the car.
* According to next instruction (input) the same procedure repeats.

1. **Description of Elevator Control system**

This Ecosystem has been designed for 40 floors. Digital designs can be best described using Finite State Machine (FSM). . It has been used to model the design in which the floors are represented by parameters S00, S01, S02…..S39, S40 representing states as ground floor, first floor, second….. Fortieth floor respectively. Upward /downward motion as well as opening and closing of the lift door is controlled by motors for which the sensors represented by variables Up/Down and Door respectively have been used in the Verilog code.

The controller contains a Reset button that is used to bring the elevator immediately to last state (current floor) when the reset input is high, also enabling reset opens the door i.e. Door =1 , and stop the motion of lift, i.e. Up = 0 and Down = 0; also Stop = 1;

The basic Logic consists of starting an initial state of lift. I.e. basically keeping lift at ground floor at rest.

Then with positive edge of clock or positive edge of reset (to stop immediately) our lift will work

If reset is pressed, then lift will remain at last state i.e. last floor, opening all the doors and stopping the motion of lift.

If reset is not pressed, then our lift will work for normal conditions, so now user will input any query (request). Now we will have three conditions:

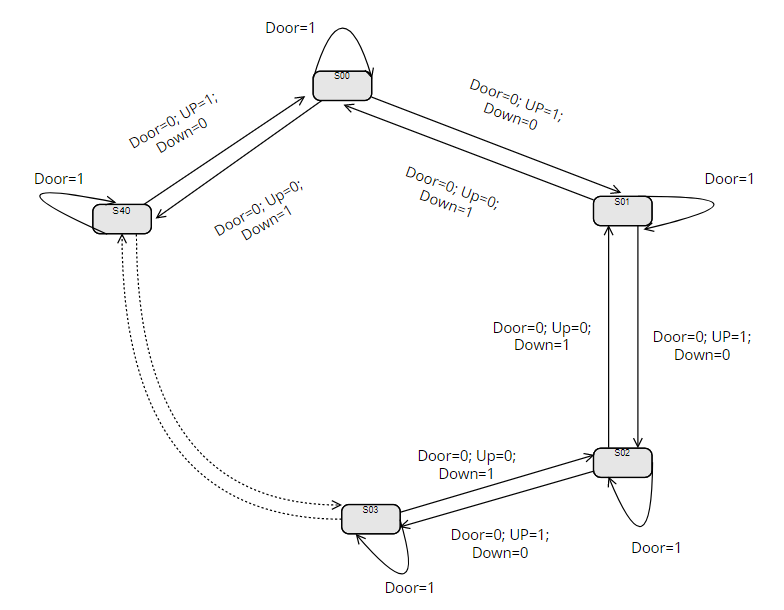
first if our goto floor or requested floor is greater than our current floor(goto>cf)

second, if our goto floor is less than requested floor (goto<cf)

third, if our goto floor is equal to our requested floor(goto = cf)

Depending upon all three conditions our lift will go up/down or remain stopped wherever it is and update floors as per logic given in code.

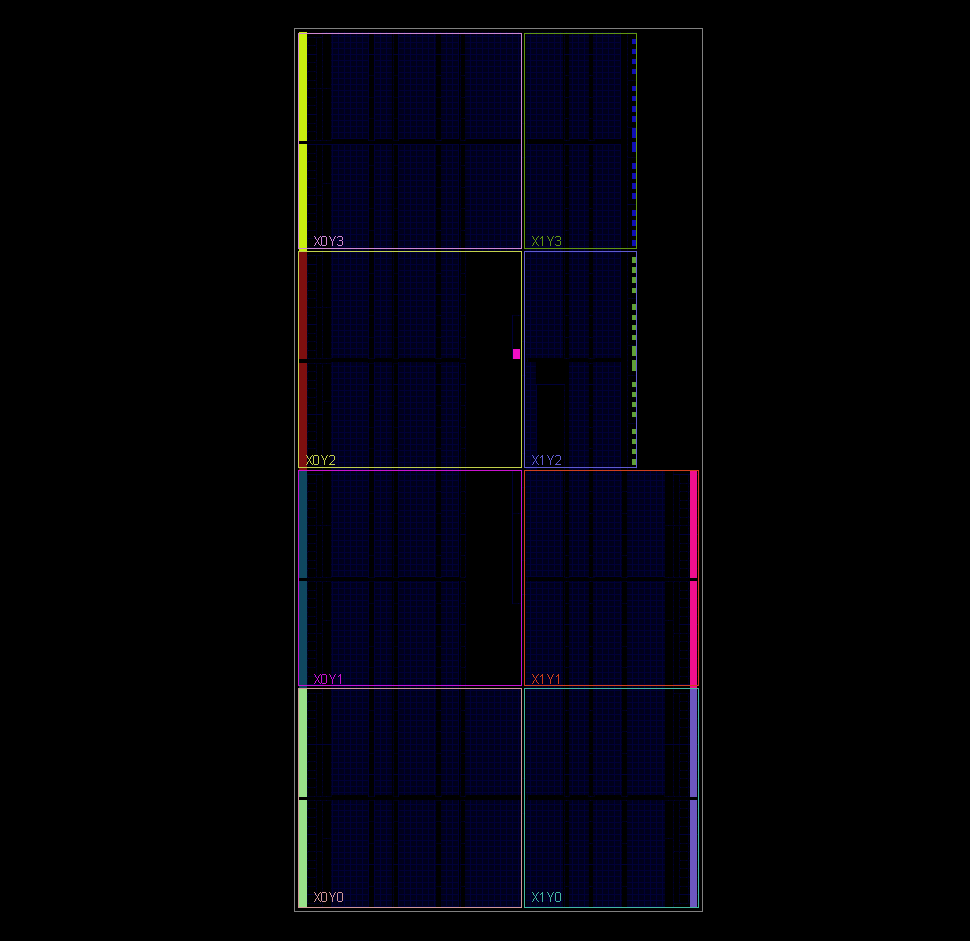
Also after updating temp current floor, we update our result in our output result, y.



**Fig 1:** State Machine Representation for Elevator

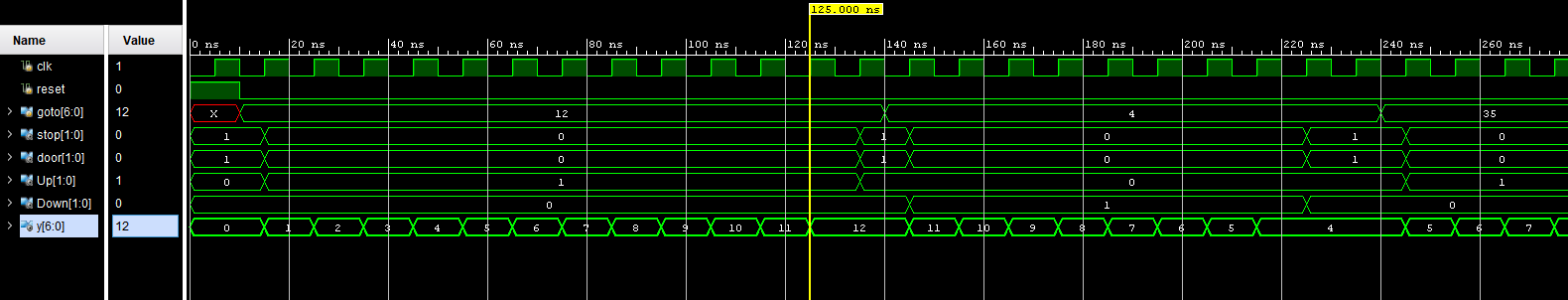
\***Note:** The following diagram has been implemented using Visual Paradigm Online software.

1. **Block Diagram/Circuit Diagram**

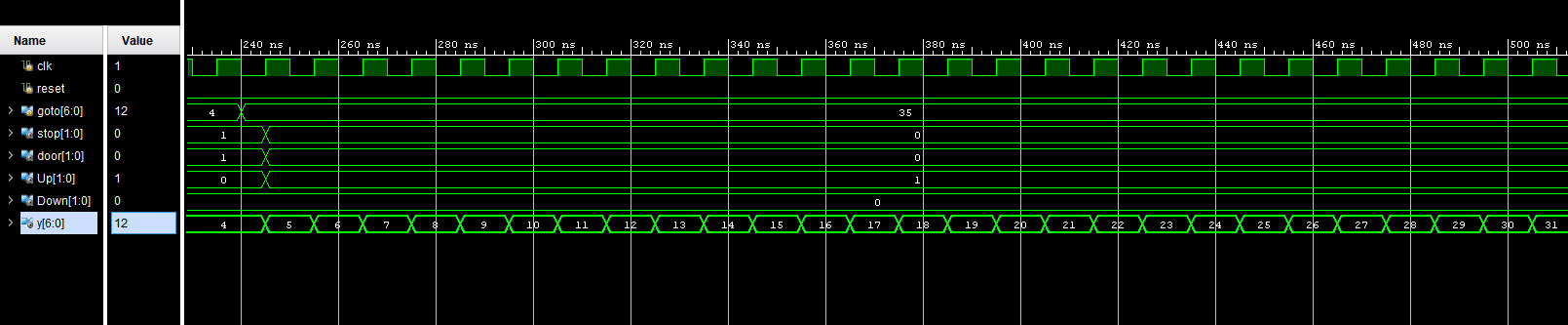


**Fig 2: Schematic Diagram for implementation**

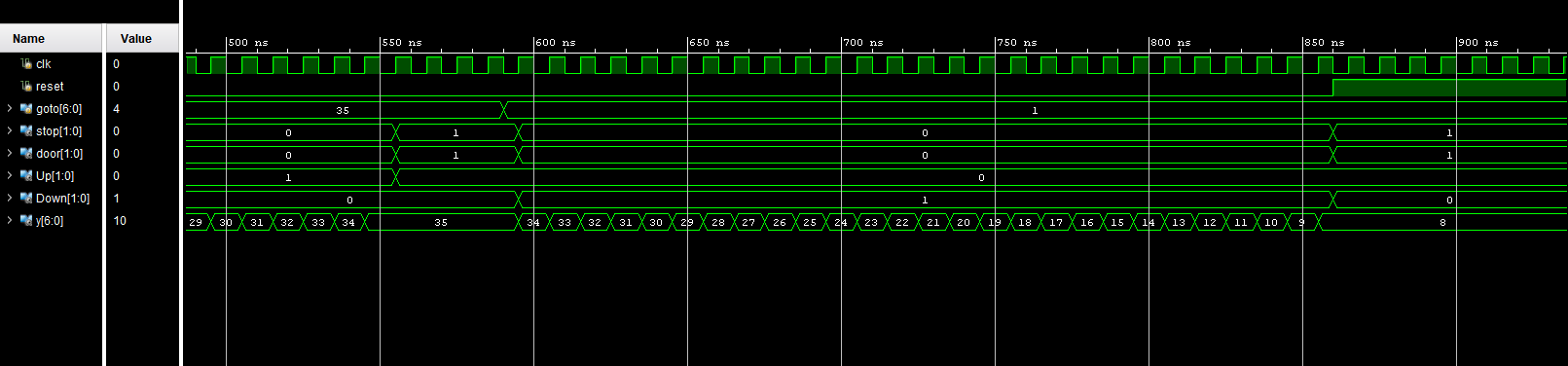
1. **Simulation Result(through Testbench)**



**Fig (3).** Represents lift moving from ground to twelfth floor and back to fourth floor



**Fig (4).** Represents lift moving to 35th floor after user inputs goto=35

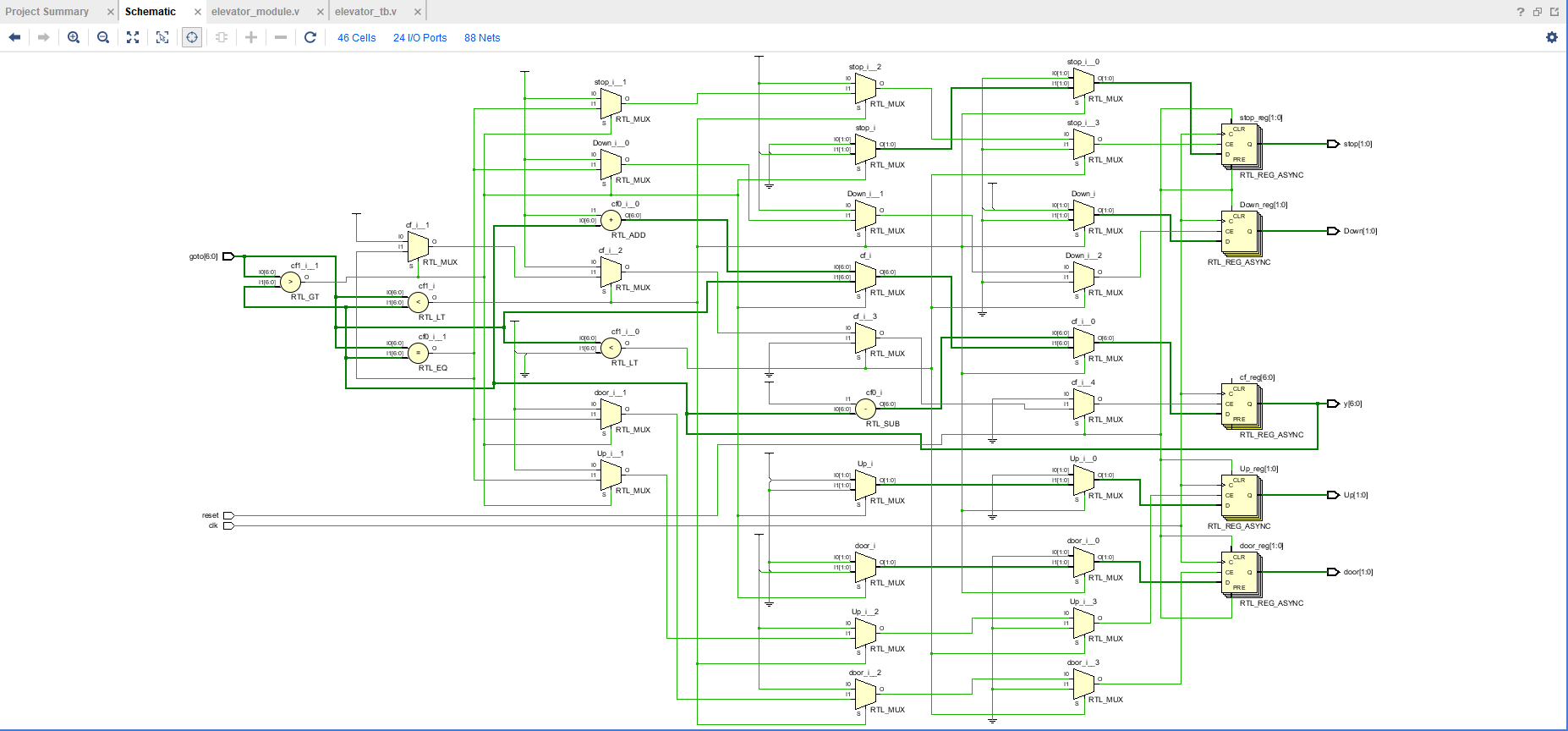


**Fig (5).** Represents lift as Idle at 35th floor for few time and lift going to 1st floor after user inputs goto=1;

But lift stops at current floor=8 as reset is set to high

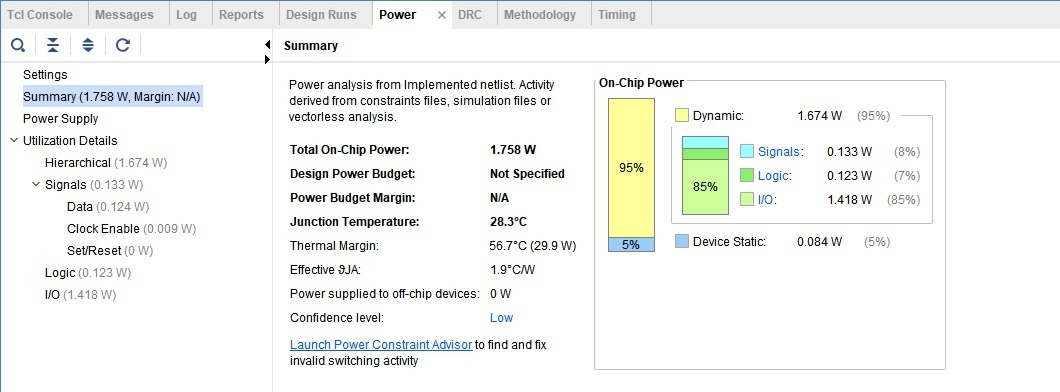
**\*Note:** It can also be seen that the lift stops at 8th floor as user press reset button at 860ns.

1. **Schematic Diagram RTL**

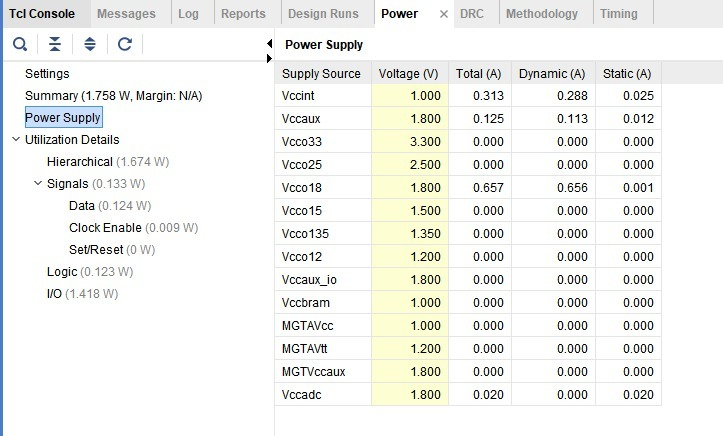


**Fig (6).** RTL Schematic Design

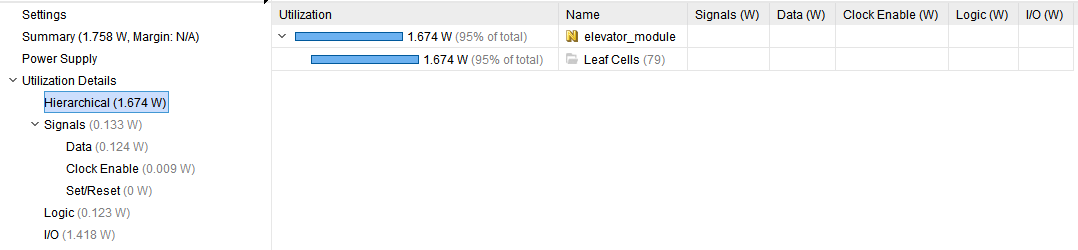
1. **Hardware Implementation Result**



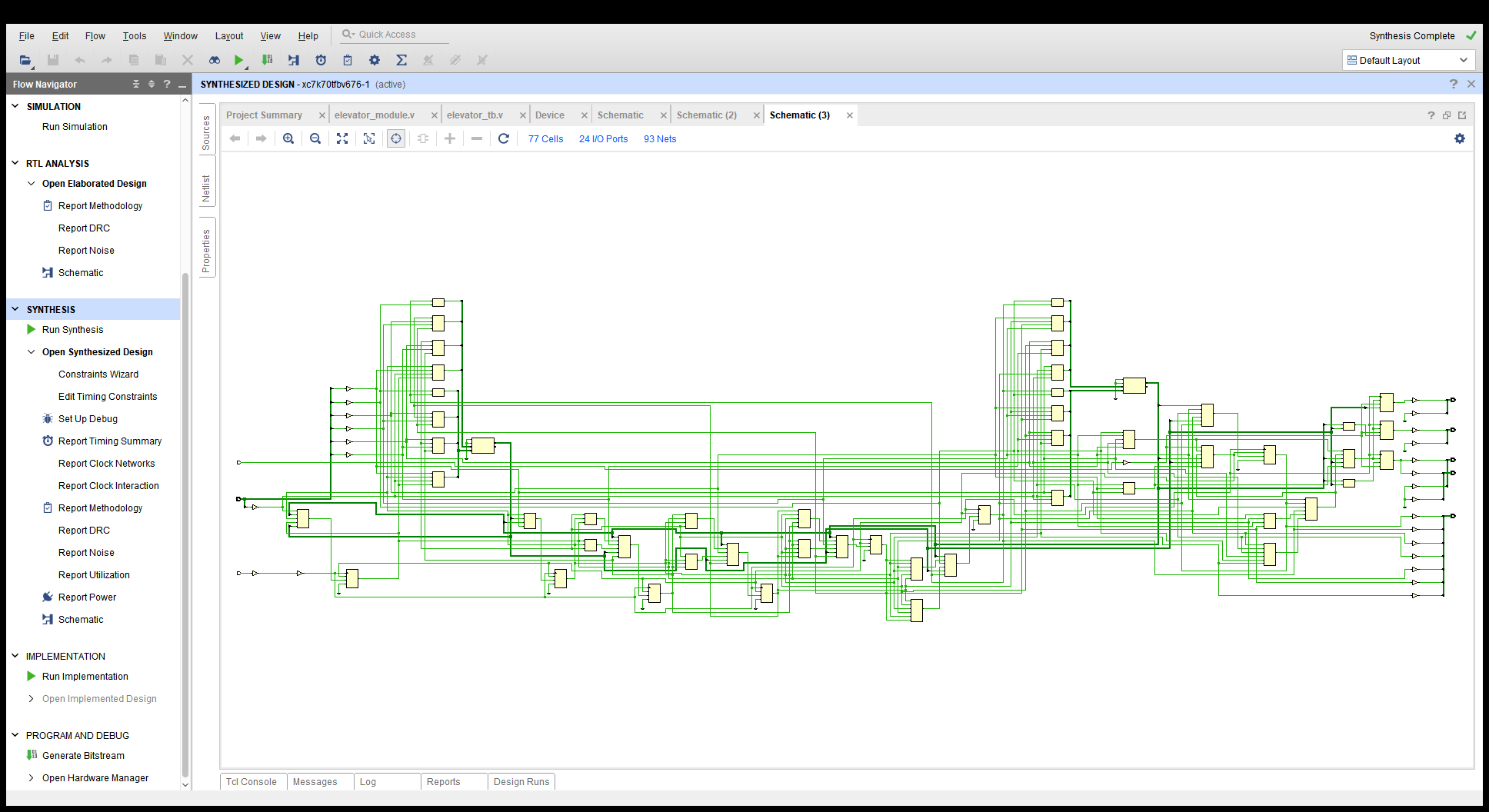
**Fig (7).** Power Summary



**Fig (8).** Power Supply



**Fig (9).** Hierarchical Utilization Details

****

**Fig (10).** Post Synthesis Schematic Diagram

1. **Conclusion**

A forty floor lift controller based on Verilog has been successfully proposed and simulated in this paper.

1. **Annexure**
2. **Code for Module:**

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company: Indian Institute of Information Technology, Guwahati

// Engineer: Ayush Anjuman Pawar

//

// Create Date: 02/18/2022 02:13:13 PM

// Design Name: Elevator

// Module Name: elevator\_module

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module elevator\_module(clk,reset,goto,stop,door,Up,Down,y);

input clk,reset;

input [6:0] goto; // 7 bit goto input (7 bits taken

output reg[1:0] door; // door open(1) or close(0)

output reg[1:0] Up; // lift going up(1)

output reg[1:0] Down; // lift going down(1)

output reg[1:0] stop; // represents (1) if lift is stopped

output [6:0] y; // y is output floor

reg [6:0] cf ; // represents current floor

initial begin // iinitial conditions

cf= 6'd0; // represents ground floor

stop= 6'd1; // stop = 1(lift stopped)

door = 1'd1; // door = 1(open)

Up=1'd0; // up\_movement = 0 (no up movement)

Down=1'd0; // down\_movement = 0(no down movement)

end

always @ (posedge clk or posedge reset) // operations to work @posedge of clock or if reset button is pushed

begin

if(reset) // if reset is 'ON'

begin // Lift should stop wherever it is and open the gates.

cf=cf; // represents lift remains at surrent floor if reset is "ON"

stop= 6'd1; // stop = 1(lift stopped)

door = 1'd1; // door = 1(open)

Up=1'd0; // up\_movement = 0 (no up movement)

Down=1'd0; // down\_movement = 0(no down movement)

end

else // if reset is 'OFF'

begin

if(goto < 6'd41) //As Max value of req floor is 40

begin

if(goto < cf ) // If GOTO floor is UP

begin // Lift will go up one by one until goto floor is reached

cf=cf-1; // represents sequential movement of lift one floor at a time

door = 1'd0; // door is closed

stop = 6'd0; // car is not stopped, i.e. car is moving

Up=1'd0; // elevator car moving up

Down=1'd1; // down is not glowing

end

else if (goto > cf) // If

begin

cf = cf+1;

door=1'd0;

stop=6'd0;

Up=1'd1;

Down=1'd0;

end

else if(goto == cf )

begin

cf = goto;

door=1'd1;

stop=6'd1;

Up=1'd0;

Down=1'd0;

end

end

end

end

assign y = cf;

endmodule

1. **Code for Test Bench Module:**

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company: Indian Institute of Information Technology, Guwahati

// Engineer: Ayush Anjuman Pawar

//

// Create Date: 02/18/2022 02:16:28 PM

// Design Name: Elevator

// Module Name: elevator\_tb

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module elevator\_tb();

reg clk,reset;

reg [6:0] goto;

wire [1:0] stop,door,Up,Down;

wire [6:0] y;

elevator\_module dut(.clk(clk),

.reset(reset),

.goto(goto),

.stop(stop),

.door(door),

.Up(Up),

.Down(Down),

.y(y)

);

initial begin

clk=0;

forever #5 clk =~ clk; // clock period = 10ns

end

initial begin

reset = 1;

#10;

reset=0;

goto=6'd12;

#130

goto=6'd4;

#100

goto=6'd35;

#350

goto=6'd01;

#270

reset=1;

end

endmodule

**Abbreviations:**

* FPGA: Field Programmable gate array.
* PLC: Programmable Logic Controller
* FSM: Finite State Machine
* S01: State-01
* GOTO: User input for requested floor
* CF: CURRENT FLOOR
* RTL: Register Transfer Level

Submitted By:

Pawar Ayush Anjuman

1901137

SG-32