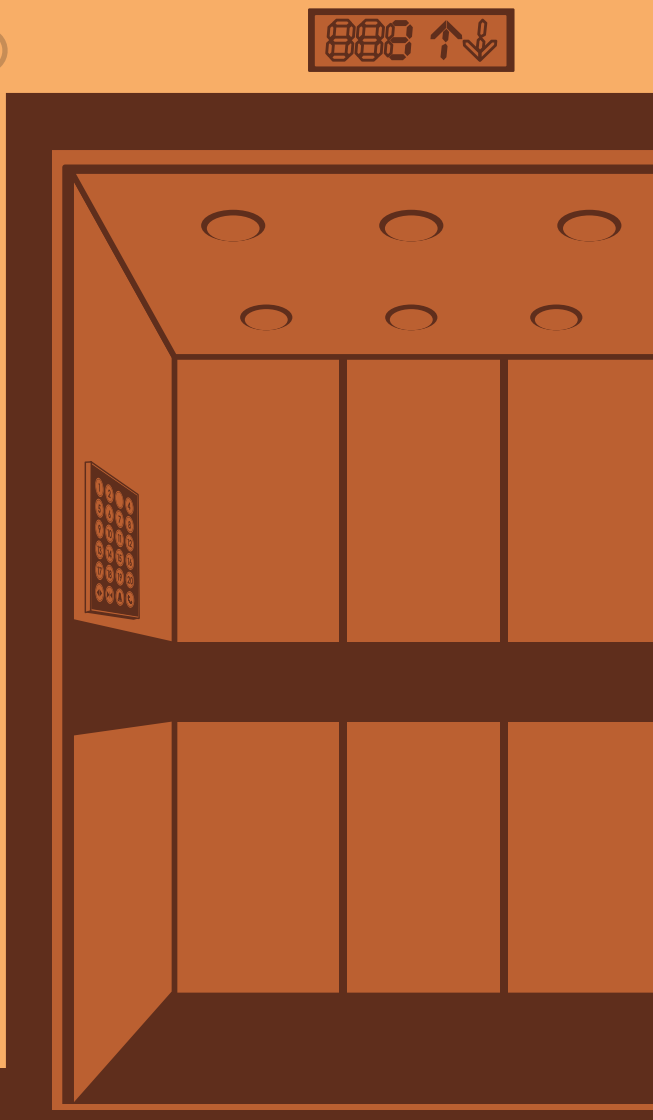
Sabancı Üniversitesi

CS303 TERM PROJECT
FALL 2022

ELEVATOR DESIGN

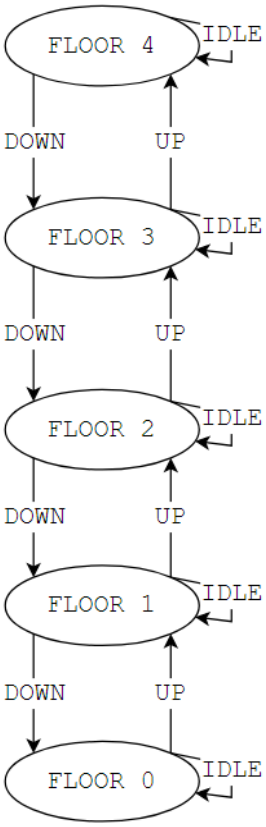


Elif Cemre Durgut - 26493
İbrahim Gürkan Yeşilbağ - 28386

Table of Contents

Project Code Explanation

In this project, we implemented a simplified elevator project in Verilog HDL as defined in the CS303 Term Project documentation.



In the elevator.v file, we have 2 different types of states, one shows the current floor and the other shows the current status.

The possible floors are

- FLOOR0,
- FLOOR1,
- FLOOR2,
- FLOOR3 and
- FLOOR4.

The possible status states are

- IDLE,
- UP and
- DOWN.

We also keep a 5-bit *requests* array which shows the requested floors to stop.
Ex: [10010] means that the elevator should stop at floors 1 and 4.

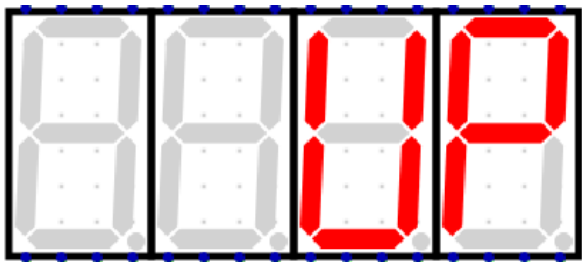
In the first always block, we do the necessary state transitions to update *curr_floor* and *curr_status* using the *next_floor* and *next_status* variables. Also, the *led_busy* is updated for synchronizing the led with the floor change.

In the second always block, we take requests from the users, evaluate them and open/close the corresponding outside and inside LEDs. We used nested cases to decide the validity of the requests.

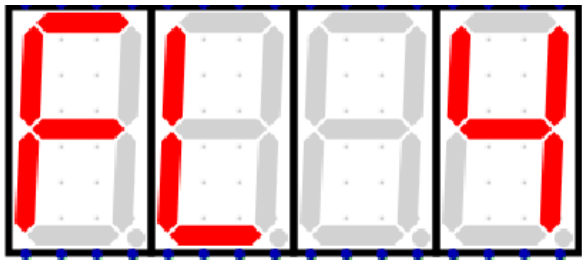
Simplified pseudocode can be found in the next page:

```
case(curr_status)
IDLE
	case(curr_floor)
		FLOOR0:
			Accept any request
			Requests[i] = 1
			Turn on led_in/outside_i
			Go up
			Next floor: FLOOR1
			If there is a request at FLOOR0:
				Turn off the leds
				Reset the requests[0]
				Go IDLE if no further requests
		FLOOR1:
			If request is from FLOOR0:
				Requests[0] = 1
				Turn on led_in/outside_i
				Go down
				Next floor: FLOOR0
			Else:
				Requests[i] = 1
				Turn on led_in/outside_i
			Go up
			Next floor: FLOOR2
		If there is a request at FLOOR1:
				Turn off the leds
				Reset the requests[1]
				Go IDLE if no further requests
		FLOOR2: ...
		FLOOR3: ...
		FLOOR4: ...
UP
	case(curr_floor)
		FLOOR0: ...
		FLOOR1: ...
		FLOOR2: ...
		FLOOR3: ...
		FLOOR4: ...
DOWN
	case(curr_floor)
		FLOOR0: ...
		FLOOR1: ...
		FLOOR2: ...
		FLOOR3: ...
		FLOOR4: ...
```

The third always block determines the output for the seven-segment display LEDs.
An example:



```
a[7:4] <= 4'b1110;
b[7:4] <= 4'b1100;
c[7:4] <= 4'b1101;
d[7:4] <= 4'b1101;
e[7:4] <= 4'b1100;
f[7:4] <= 4'b1100;
g[7:4] <= 4'b1110;
p[7:4] <= 4'b1111;
```

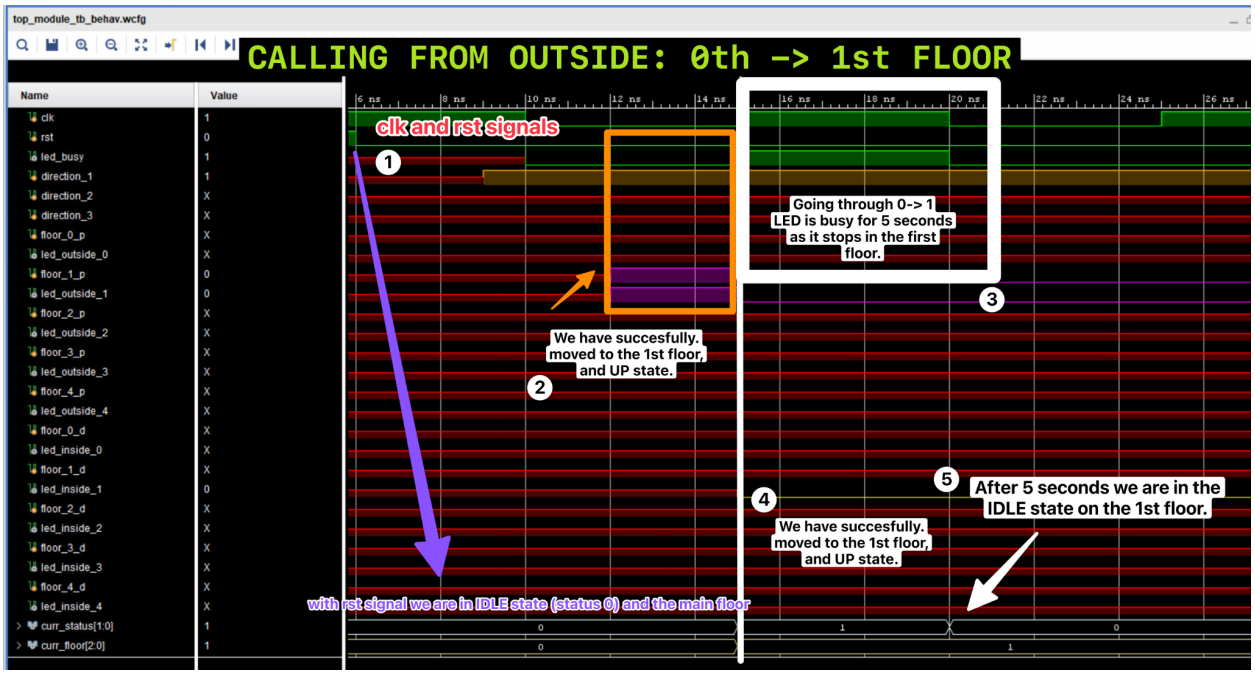


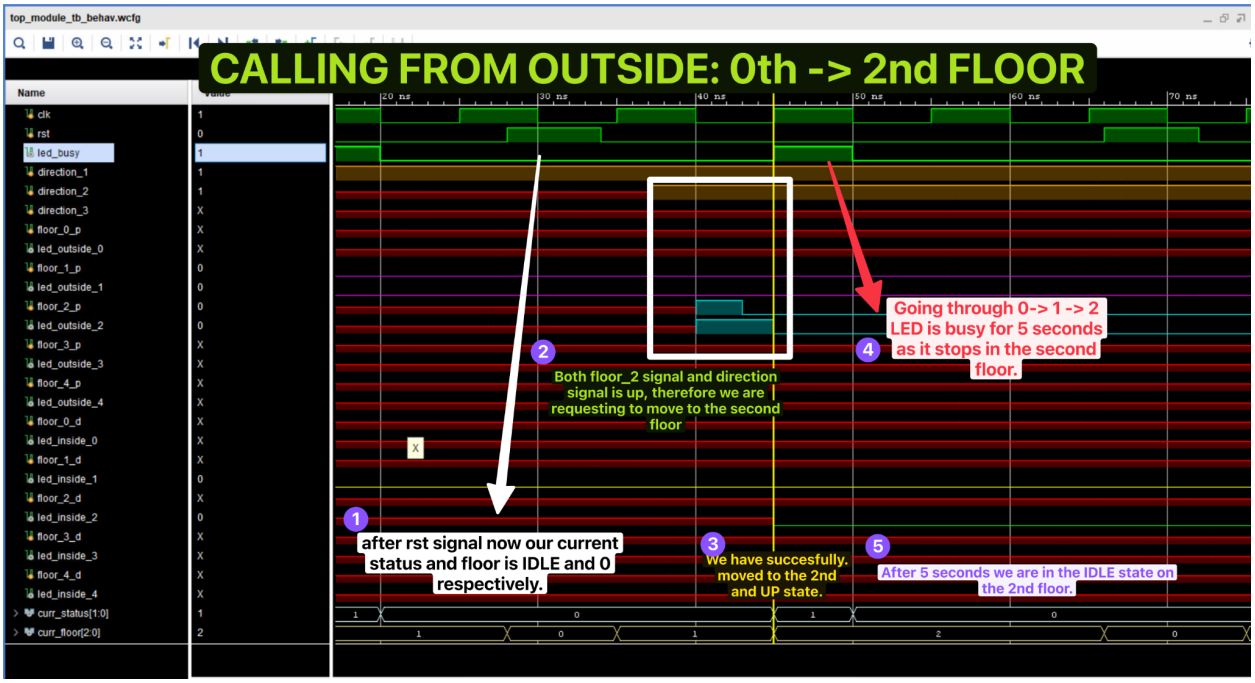
```
a[3:0] <= 4'b0111;
b[3:0] <= 4'b1110;
c[3:0] <= 4'b1110;
d[3:0] <= 4'b1111;
e[3:0] <= 4'b0111;
f[3:0] <= 4'b0110;
g[3:0] <= 4'b0110;
p[3:0] <= 4'b1111;
```

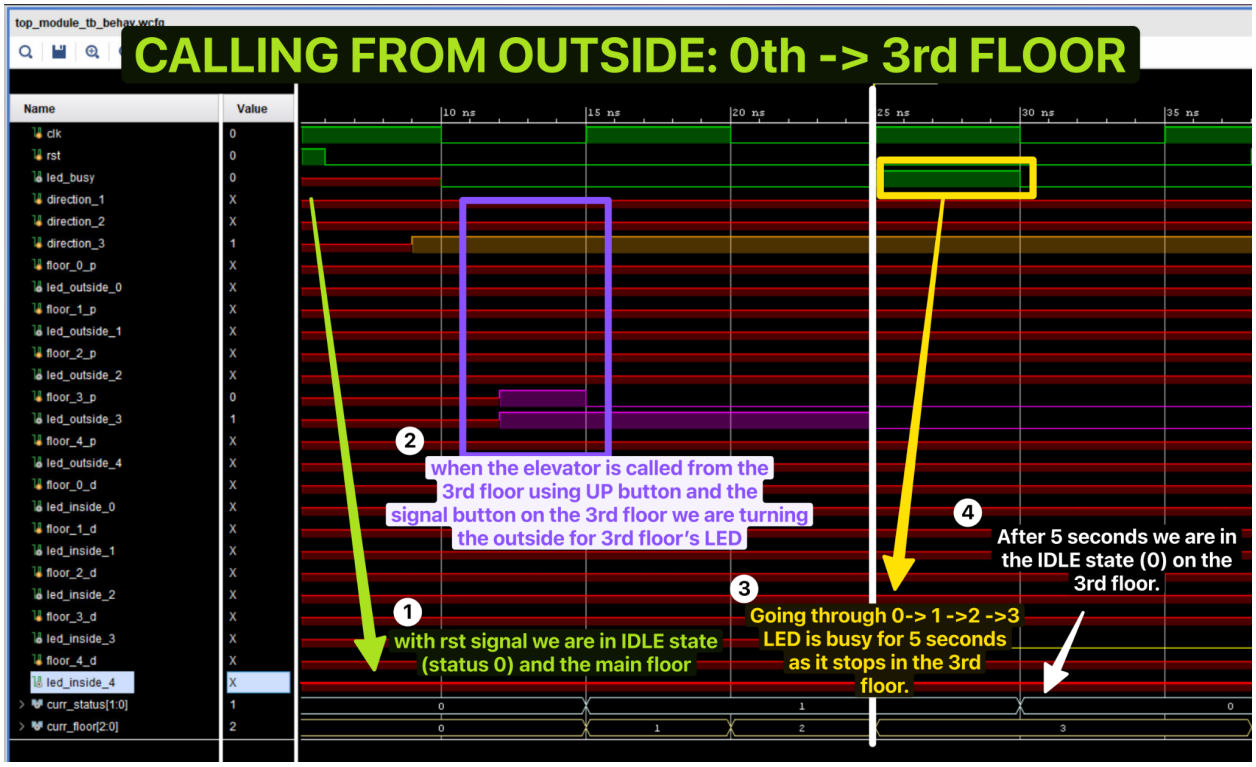
And the last always block controls the counter to keep the clock cycle period during states.

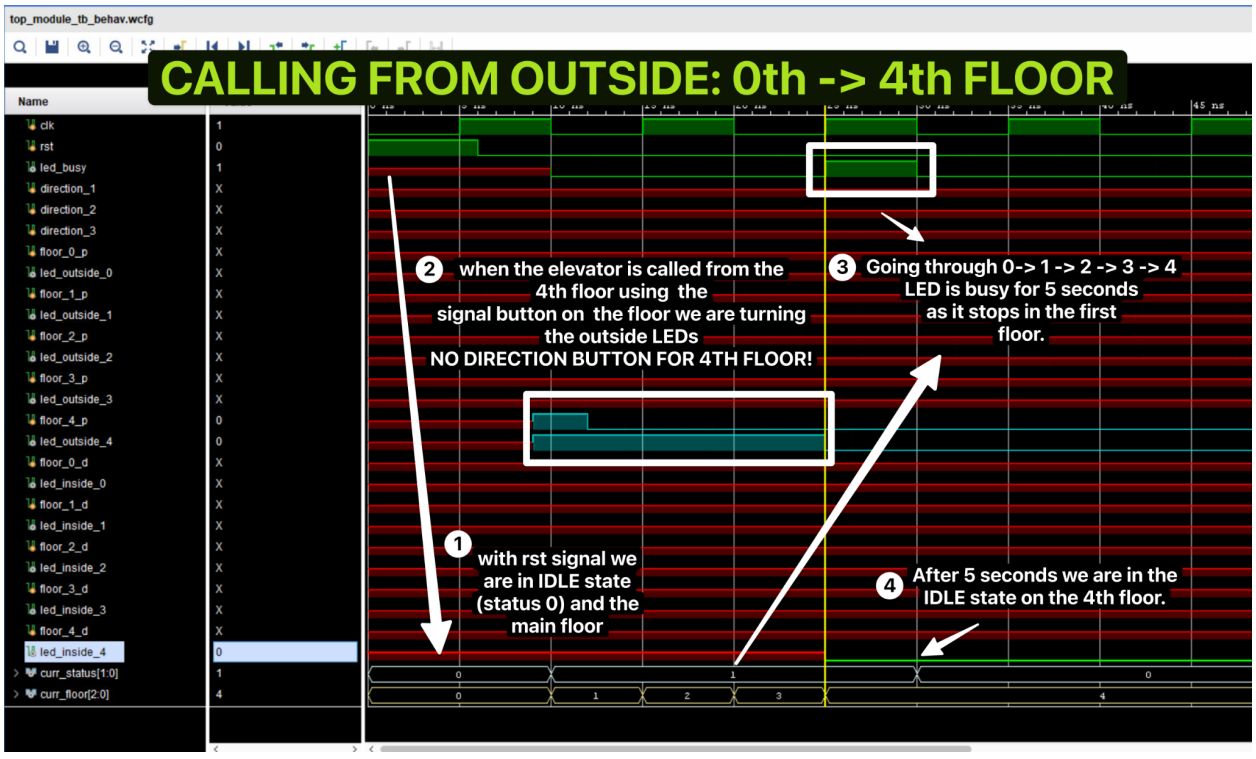
Test Cases

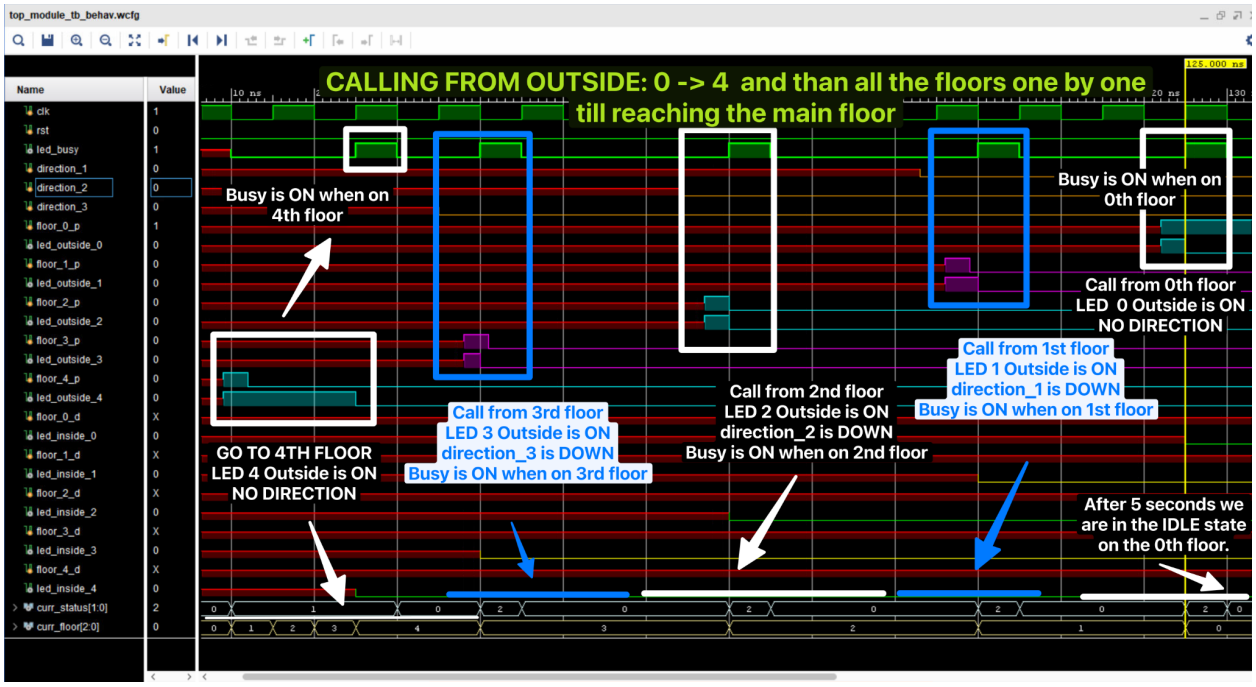
1) Calling only from outside

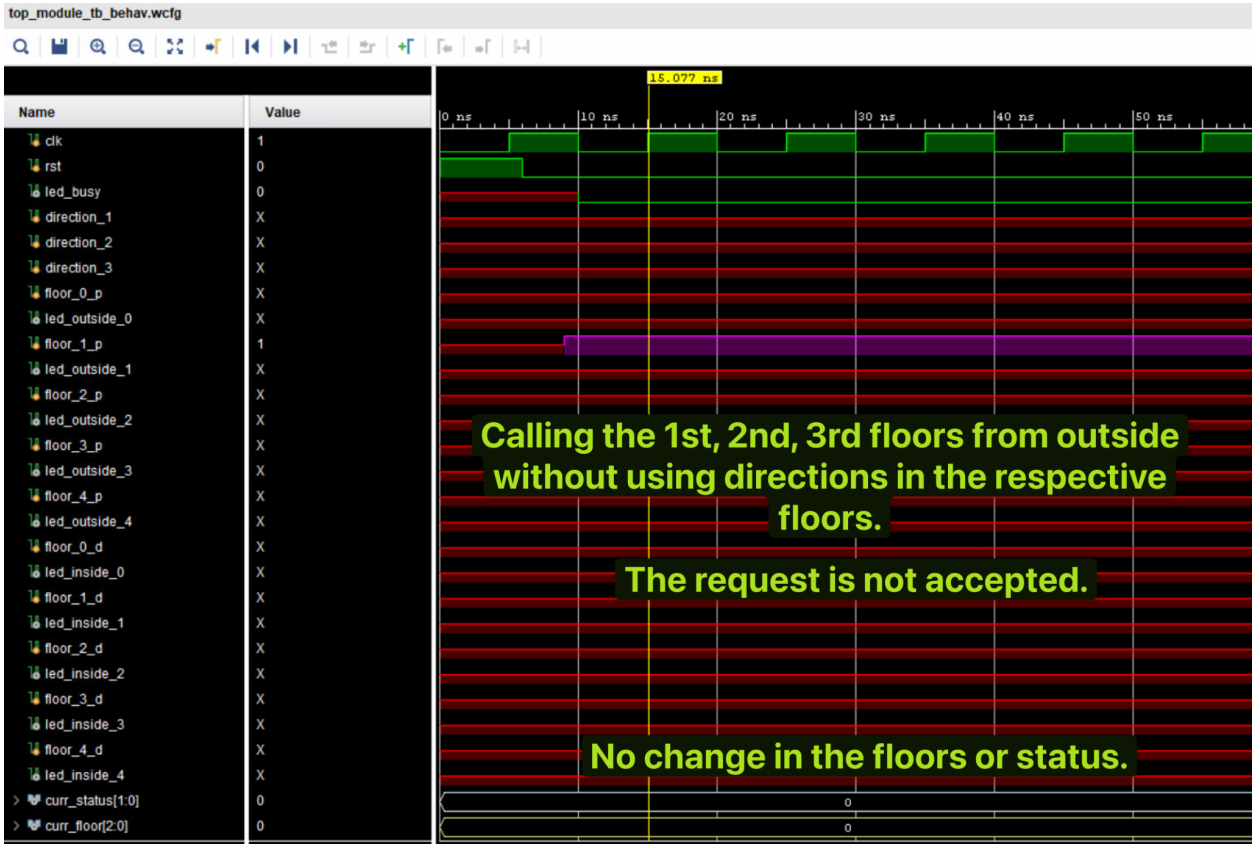




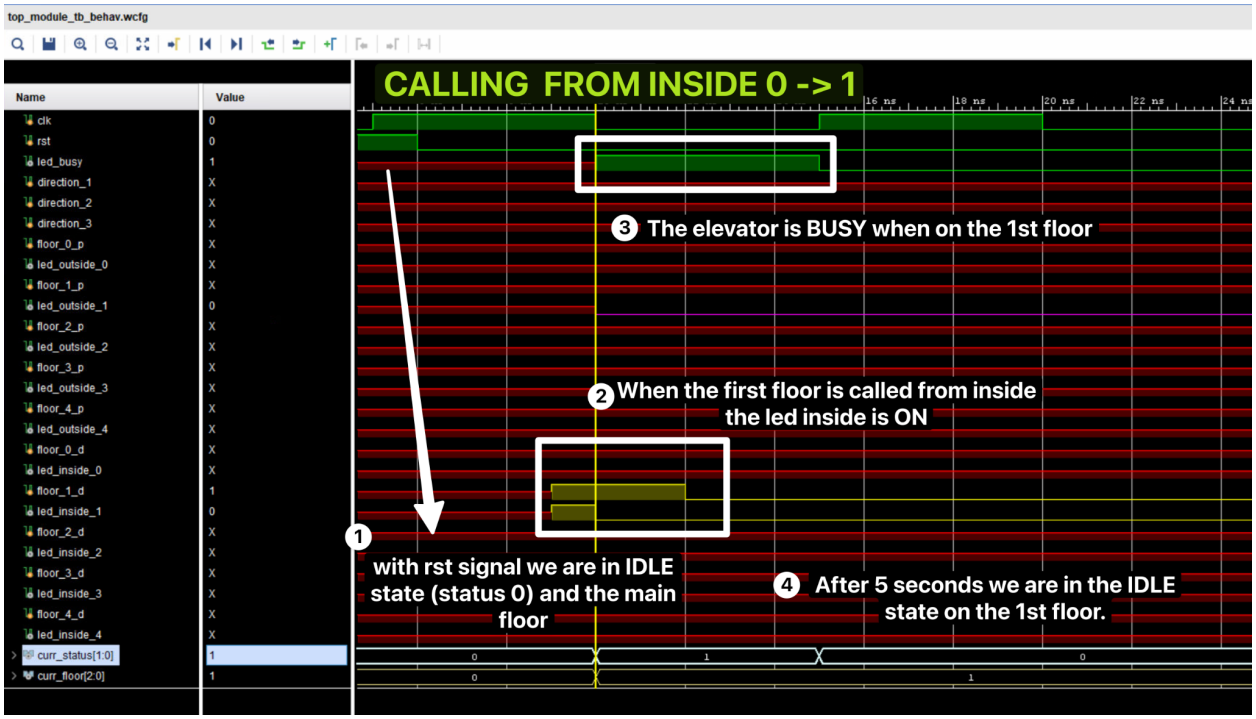

CALLING FROM OUTSIDE: 0th -> 3rd FLOOR
top_module_tb_behav.wcfg
Name
Value
clk
rst
led_busy
direction_1
direction_2
direction_3
floor_0_p
led_outside_0
floor_1_p
led_outside_1
floor_2_p
led_outside_2
floor_3_p
led_outside_3
floor_4_p
led_outside_4
floor_0_d
led_inside_0
floor_1_d
led_inside_1
floor_2_d
led_inside_2
floor_3_d
led_inside_3
floor_4_d
led_inside_4
curr_status[1:0]
curr_floor[2:0]
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
2
when the elevator is called from the 3rd floor using UP button and the signal button on the 3rd floor we are turning the outside for 3rd floor's LED
4
After 5 seconds we are in the IDLE state (0) on the 3rd floor.
1
with rst signal we are in IDLE state (status 0) and the main floor
3
Going through 0-> 1 ->2 ->3 LED is busy for 5 seconds as it stops in the 3rd floor.

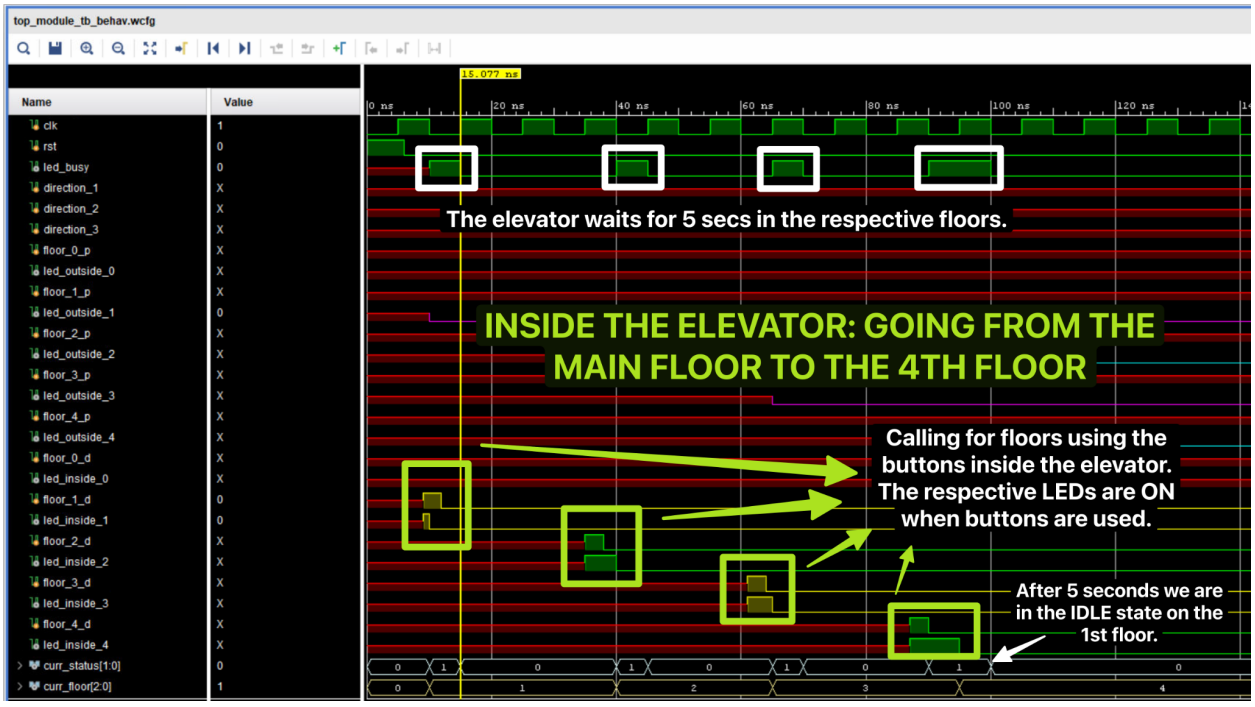

CALLING FROM OUTSIDE: 0th -> 4th FLOOR
top_module_tb_behav.wcfg
Name
clk
rst
led_busy
direction_1
direction_2
direction_3
floor_0_p
led_outside_0
floor_1_p
led_outside_1
floor_2_p
led_outside_2
floor_3_p
led_outside_3
floor_4_p
led_outside_4
floor_0_d
led_inside_0
floor_1_d
led_inside_1
floor_2_d
led_inside_2
floor_3_d
led_inside_3
floor_4_d
led_inside_4
curr_status[1:0]
curr_floor[2:0]
45 ns
2
when the elevator is called from the 4th floor using the signal button on the floor we are turning the outside LEDs
NO DIRECTION BUTTON FOR 4TH FLOOR!
3
Going through 0-> 1 -> 2 -> 3 -> 4 LED is busy for 5 seconds as it stops in the first floor.
1
with rst signal we are in IDLE state (status 0) and the main floor
4
After 5 seconds we are in the IDLE state on the 4th floor.

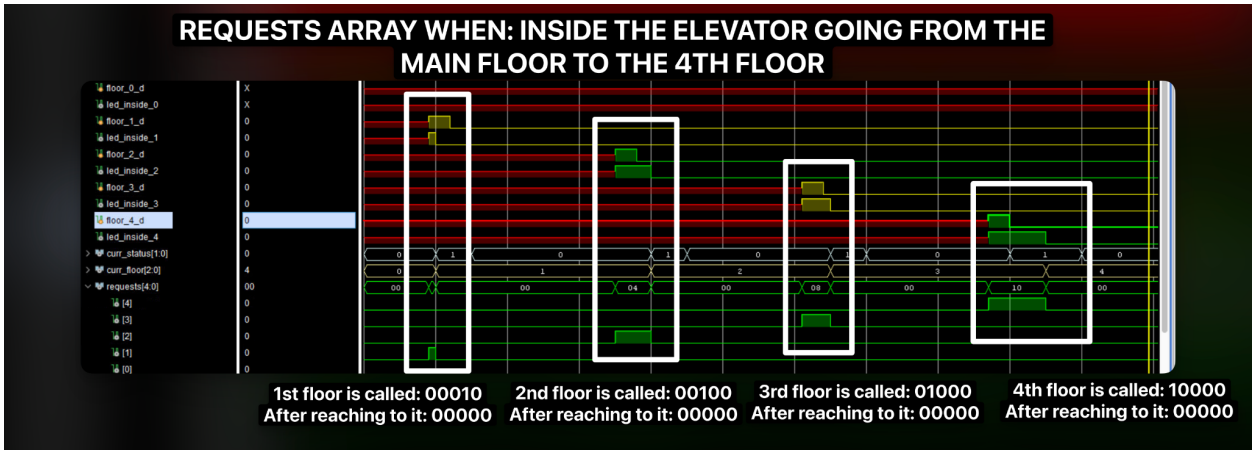

top_module_tb_behav.wcfg
CALLING FROM OUTSIDE: 0 -> 4 and than all the floors one by one till reaching the main floor
Busy is ON when on 4th floor
Busy is ON when on 0th floor
Call from 0th floor
LED 0 Outside is ON
NO DIRECTION
Call from 1st floor
LED 1 Outside is ON
direction_1 is DOWN
Busy is ON when on 1st floor
Call from 2nd floor
LED 2 Outside is ON
direction_2 is DOWN
Busy is ON when on 2nd floor
Call from 3rd floor
LED 3 Outside is ON
direction_3 is DOWN
Busy is ON when on 3rd floor
GO TO 4TH FLOOR
LED 4 Outside is ON
NO DIRECTION
After 5 seconds we are in the IDLE state on the 0th floor.

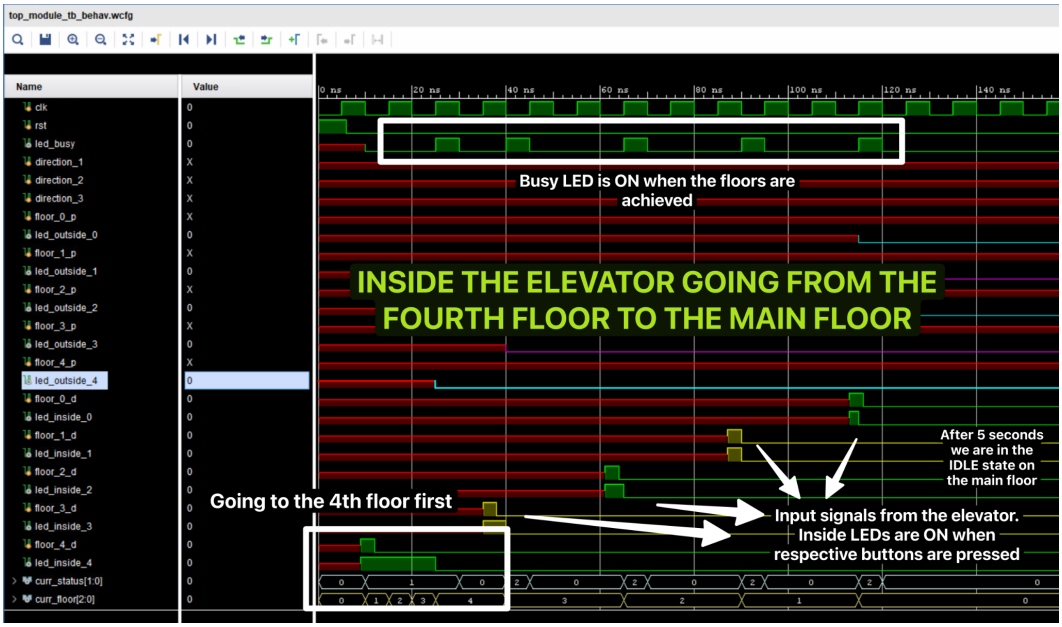

top_module_tb_behav.wcfg
Calling the 1st, 2nd, 3rd floors from outside without using directions in the respective floors.
The request is not accepted.
No change in the floors or status.

2) Calling only from inside



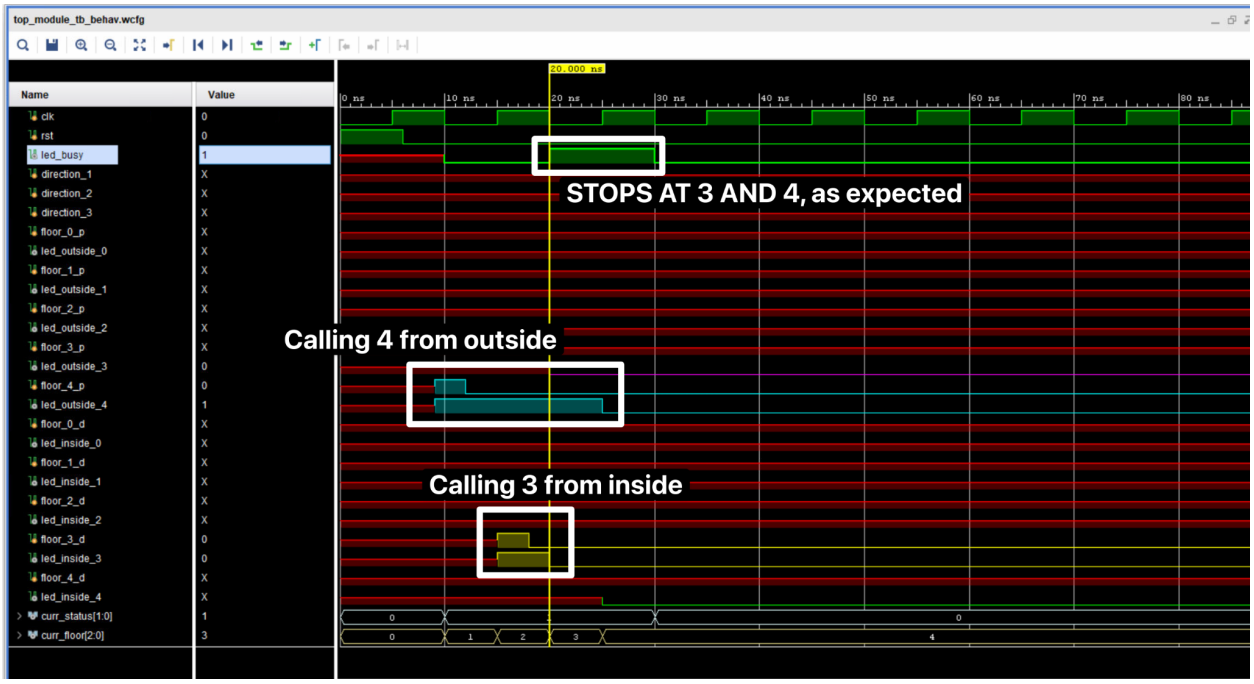


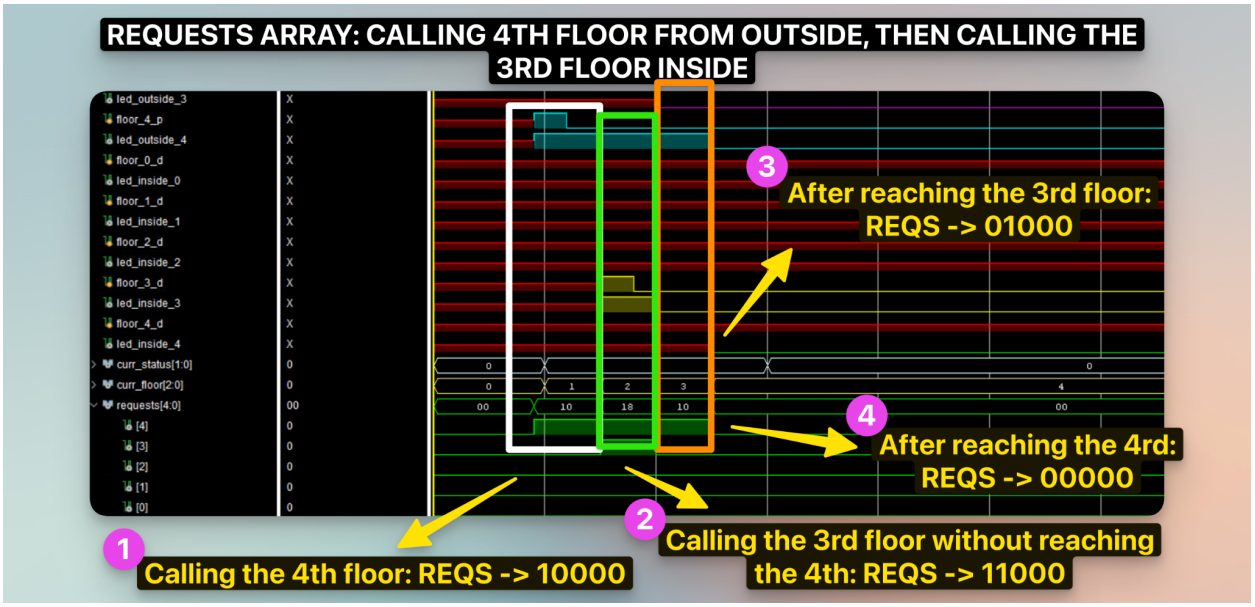

REQUESTS ARRAY WHEN: INSIDE THE ELEVATOR GOING FROM THE MAIN FLOOR TO THE 4TH FLOOR
1st floor is called: 00010
After reaching to it: 00000
2nd floor is called: 00100
After reaching to it: 00000
3rd floor is called: 01000
After reaching to it: 00000
4th floor is called: 10000
After reaching to it: 00000

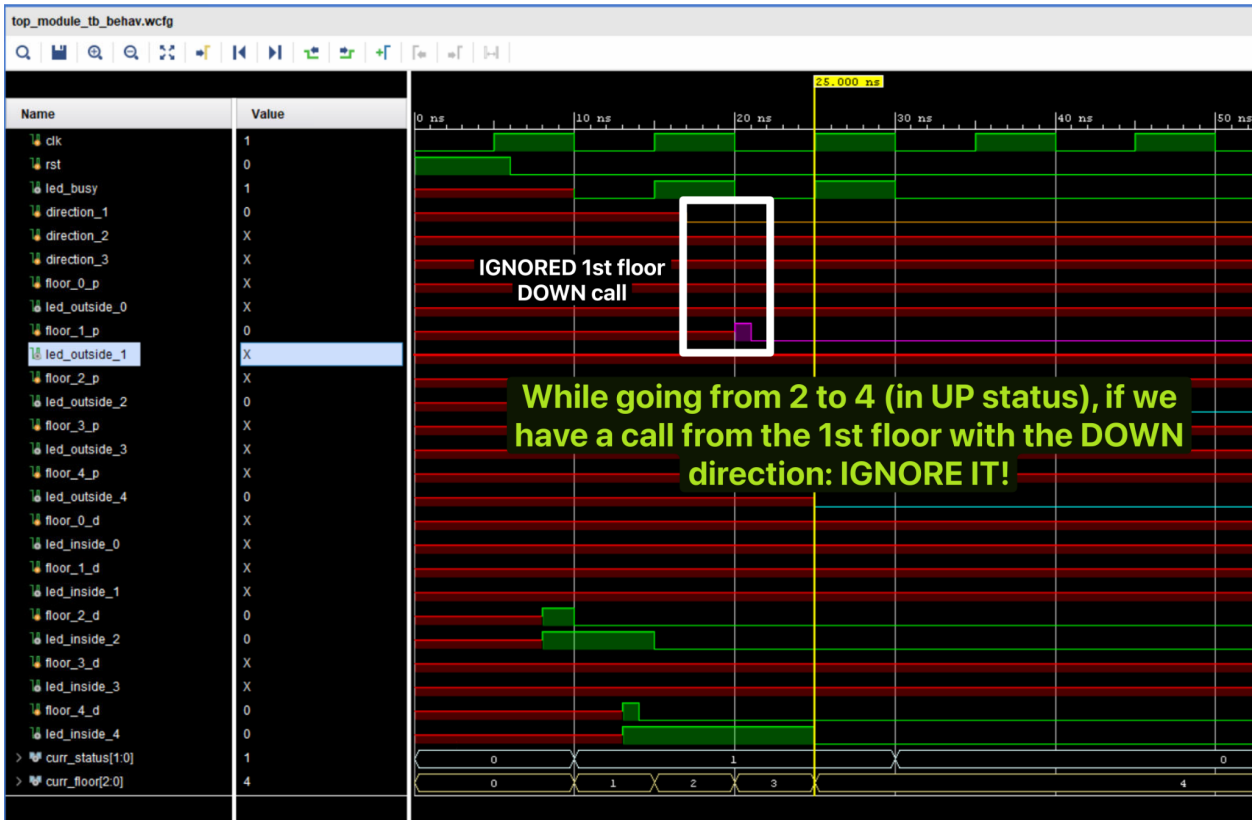

top_module_tb_behav.wcfg
Busy LED is ON when the floors are achieved
INSIDE THE ELEVATOR GOING FROM THE FOURTH FLOOR TO THE MAIN FLOOR
Going to the 4th floor first
Input signals from the elevator. Inside LEDs are ON when respective buttons are pressed
After 5 seconds we are in the IDLE state on the main floor


top_module_tb_behav.wcfg
INSIDE OF ELEVATOR:
0 -> 1 -> 2 -> 3 -> 4 -> 3 -> 2 -> 1 -> 0

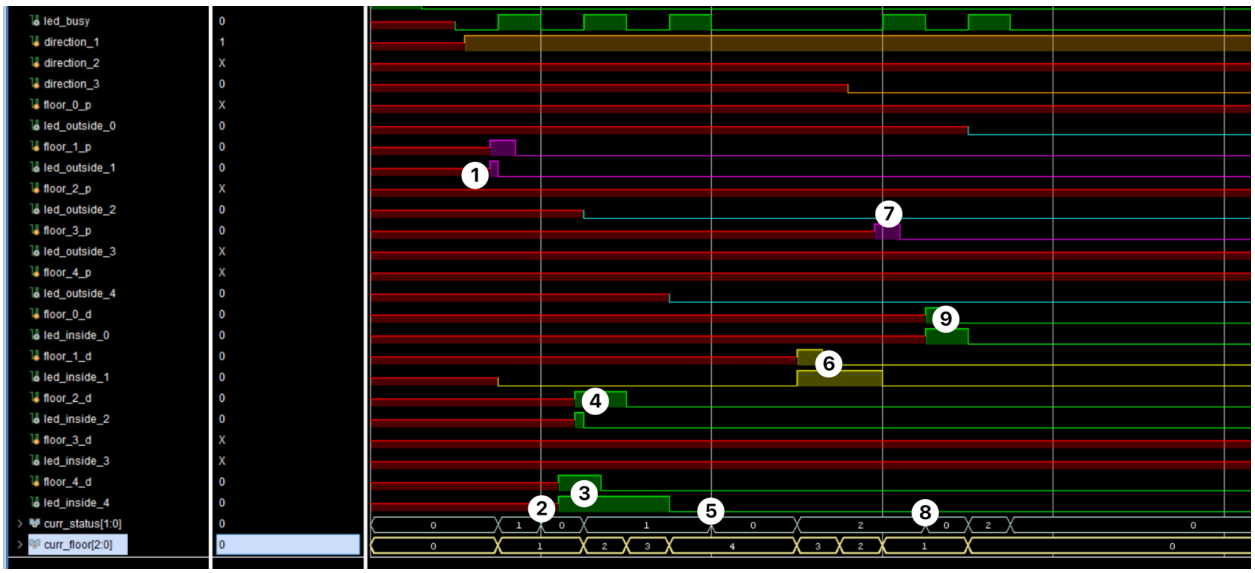
3) Calling from outside and inside






top_module_tb_behav.wcfg
Name
Value
clk
rst
led_busy
direction_1
direction_2
direction_3
floor_0_p
led_outside_0
floor_1_p
led_outside_1
floor_2_p
led_outside_2
floor_3_p
led_outside_3
floor_4_p
led_outside_4
floor_0_d
led_inside_0
floor_1_d
led_inside_1
floor_2_d
led_inside_2
floor_3_d
led_inside_3
floor_4_d
led_inside_4
curr_status[1:0]
curr_floor[2:0]
25.000 ns
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
IGNORED 1st floor DOWN call
While going from 2 to 4 (in UP status), if we have a call from the 1st floor with the DOWN direction: IGNORE IT!

A Comprehensive Test Case



1) Request from the outside of floor 1 (accepted).
2) The elevator becomes busy on floor 1, then it becomes idle since there is no request for 5 seconds.
3) Request to go to floor 4 from the inside (accepted).
4) While the elevator is going up to floor 4, another call to floor 2 from the inside (accepted). The elevator stops at floor 2.
5) The elevator reaches floor 4 and stops. Since there is no request for 5 sec, it becomes idle.
6) Request to go to floor 1 from the inside (accepted).
7) Request from the outside of floor 3 to go down (rejected since floor 3 has been passed).
8) The elevator stops at floor 1, becomes busy, and then idle.
9) Request to go to floor 0 from the inside (accepted). The elevator goes to floor 0, stops, and becomes idle.