CS303 TERM PROJECT

Ahmet Alperen Güngör 28847 – Anıl Arslan 29468

In the project; the top module, clock divider and debouncer as well as seven segment display is given to us, we implemented elevator module using these modules. In the elevator module, we created 4 parameters: 3 for representing the 3 states “IDLE”, “UP” and “DOWN” which is necessary for our state machine; and one other parameter to keep the track of our time which is named “limit”. We created 8 registers: 2 dummy registers called “dummy” and “dummy2” for the unused else and default statements, one reg to keep track of the current time called “timer”, another reg to keep track of the current floor of elevator called “current\_floor\_info”, 2 regs to use while changing states, which are called “STATE” and “NEXT\_STATE”. 2 more regs, where “queue” keeps track of requested floors while elevator moves, and the other reg called “wanted\_floor\_info” that keeps the track of the first requested floor information.

Text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generatedIn the initial begin part, we initialized all of our outputs to 0 so that it does not show as X in the simulation. After that part, we used an always block to achieve current state and next state transition, which is triggered by the posedge of our clk\_50hz clock. If reset, state is IDLE. If not, next state will be assigned to the current state.

Next, we used another always block for our state machine to accurately perform state transitions by changing and assigning necessary values to our variables. Before checking and updating the values, we checked the reset situation, and if it is reset then timer goes back to 0 as well as current floor. If not, our always block continues. If 5 seconds have passed, we check the current state and do the updates accordingly. If the led is busy after 5 seconds and we are at the final destination, then state will change to IDLE and busy led will be turned off. If we are not in the final destination, elevator continues working. We turn off the inside and outside led according to current floor info when elevator stops in this floor.

In the next part inside our always block, we update our variables “wanted\_floor\_info” (which was the final destination floor) and “queue” (which is the other accepted destination floors) according to buttons that were pushed from inside and outside – but while doing so necessary condition checks are performed. If a requested floor is not in the range of the elevators current movement direction and range, then the request is ignored.

In the third always block, we updated the outputs values of a,b,c,d,e,f,g,p of seven segment display according to current state and floor of the elevator.

A picture containing diagram

Description automatically generated Below is an example simulation, where user calls from outside second floor, makes the state UP and reaches the second floor. Later, user calls from the inside third floor, again reaches the destinated floor. Finally, user calls from the outside first floor. Since it was in IDLE state after reaching the third floor, it makes the state DOWN and goes to the first floor.