12/14/16

**Project Overview**

The elevator controller is parameterizable on the fields of floors and elevators. It can have up to 6 elevators and up to 16 floors. The 6 elevator limit is because there are only 6 7-segment displays on the DE1-SoC board. We max out at 16 floors because that’s the highest number we can display with 1 digit in hexadecimal. When a single elevator is moving up or down it will show the floor that it just moved to. For floors above 9 it will show the hexadecimal value for that floor. When an elevator is in the loading state it will show an “L” to signal that the elevator’s doors are open and passengers can enter or exit. When an elevator is in idle it will simply display “-”.

We use some specific terminology when describing our elevator system either in the code or in documentation. A floor call is a request to be picked up by an elevator. A floor call can either be to go up or down; the elevator will handle these requests differently depending on the situation. A destination represents the floor buttons inside of the elevator. Once a passenger enters the elevator they choose where they want to go, we call this request a destination for the elevator. Within the logic of the state machine we use a floor\_call bit and a destination bit to represent if there are any floor calls or destinations. We also use one bit to represent the elevator’s direction 1 means up and 0 means down. The state machine has 4 states: idle, loading, up and down. When an elevator is idle it is not moving and its direction is whatever direction it was heading in previously. When an elevator is loading its doors are open and passengers can enter or exit. The up and down states mean the elevator is in motion and will accordingly increment or decrement the current floor of the elevator.

To enter inputs to the elevator you use the switches and button “KEY\_0”. Switch 5 determines if the input is a floor call for 0 or a destination for 1. Switch 4 determines the direction that the request wants to go. Switches 3 down to 0 represent the floor that corresponds to either the floor call or the destination. Since destination requests happen from within an elevator we need a way to specify which elevator when we enter any destination request. We use switches 8 down to 6 to specify which elevator the destination request is coming from. The elevator runs off two clocks, a high precision input clock and a slow clock that describes how long the elevator stays on any given state. The high precision clock runs at 50 MHz so the system can quickly latch any inputs. The slow clock runs at 0.33 Hz (3 seconds per state). We make this clock so slow so it is easy to enter inputs and clearly see what the elevator is doing.

For a single elevator we store floor calls and destinations in vectors that are the length of the number of floors. There are two vectors for floor calls, one for up calls and one for down calls. There is just one vector for destination requests. When a request comes in we set the bit at the corresponding floor to 1. Each elevator state machine has a corresponding floor control component. This component is responsible for acknowledging when the elevator has arrived at a floor call or a destination and turning off the bit in its vector.

There is also a master control component that gives this system intelligence. It takes in the inputs from the board and determines which elevator should receive that request. There are many logic cases that we wrote to determine the best elevator to give each request to. If the master control still can’t resolve which elevator should receive the request it gets sent to the first elevator by default. There are still many cases in this component that we can improve, but ran out of time at the end of the semester.

Overall we feel like we accomplished all of the requirements for this project. We finished the following stages:

1. Develop a system with a single elevator in a building with a fixed number of floors.
   1. We described use cases for both single and multiple elevators
   2. Block Diagram and State Diagram
   3. Test Plan, Test Cases, and Test Bench Simulation
2. Make the number of floors parameterizable for the single elevator design.
3. Make the number of independent elevators parameterizable.
4. Make the system smart by coordinating the actions of the system.