Elevator Control System

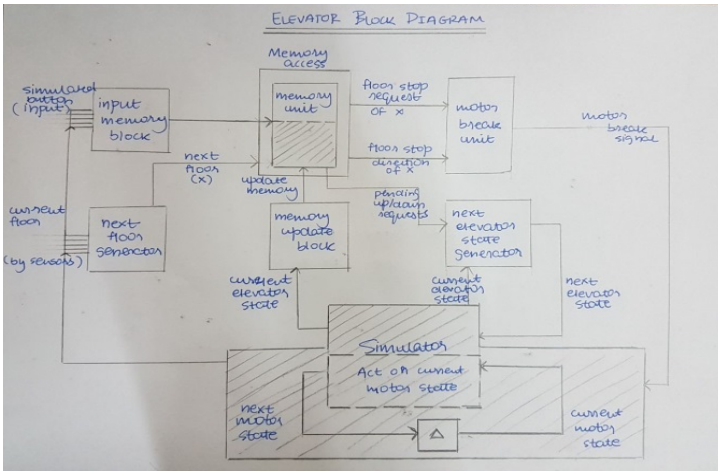
Nitish Shah (160030002, *nitish.shah1@iitgoa.ac.in*),
Piyush Doke (160010006, *piyush.doke.16001@iitgoa.ac.in*),
Shivansh Aadittya(160020021, shivansh.aadittya.16002*@iitgoa.ac.in*),
P Shanmuga Srinivas (160020012 *shanmuga.srinivas.16002@iitgoa.ac.in*)

Introduction:

The digital elevator system controls the movement of the elevator and minimizes the distance traveled by the elevator. It works similar to the lift system that can usually be seen in buildings. The algorithm is simple: if the elevator is making an up or down journey, then complete all the requests in that direction, change the direction and repeat (a request for a floor says that the elevator has to stop at that floor). This ensures to a certain degree that the elevator doesn't make unnecessary journeys or that the elevator doesn't force a person to go up, when he/she wants to go down of vice versa (this can be done because there are two buttons on each floor, one labeled up and one down. The person has to choose which way he/she wants to go and presses only the corresponding button).

System Overview and Implementation Details:

A block diagram of the system:



The elevator system: The elevator can be called on a floor from the floor itself (each floor has two buttons, one for up and one for down) or from inside the elevator. An array stores which floors the elevator has to stop on and another array stores if the elevator has to stop on that floor on the up or down journey. Elevator next state indicates if the elevator is making its up or down journey or is idle. Motor state will indicate if the motor is going up, down or is stationary. A break signal will turn to high when the lift has to stop at the next floor. The pending up/down requests will be checked in order depending upon the motor state. The lift changes its direction if there are no pending up requests during up motion and vice versa.

The elevator has three jobs:

Storing button inputs: When a button is pressed, the floor number is stored and whether the elevator has to stop on the up journey or the down is decided as follows:

If the button is on a floor, then the direction stores is the same as the direction the person wants to go in, if the button is pressed inside the elevator, then the direction is based on if the floor is above or below the current elevator floor.

Stopping on the floor: The elevator checks if has to stop on the next floor and if the direction is the same. If it is, it sends a 'break' signal to the simulator which stops the elevator on the next floor.

Decide which way to go next: Every time the elevator stops on a floor, it checks if there are pending requests (if it has to pick or drop someone on a floor) in the same direction it was going. if there are then it keeps going in the same direction, if not, then it checks if there are requests to stop in the other direction. If there are, then it changes its
direction.

The Simulator: The elevator just decides the next way to go based on the current inputs. To actually use the system, a simulator is needed, which takes the next decision made by the elevator, simulates it, and feeds back the new situation so the elevator can decide what to do next.

Output: An LED for each floor is kept to show which floor the elevator is on. There are 2 buttons on each floor to call the elevator (except ground and top floor, which have only one button), and a button for each floor inside the elevator.

Result:

The elevator works as expected.

Conclusion:

We developed a simple model of an elevator system. With multi elevator support and appropriate physical components, this model can be scaled to fit real life scenarios.