ECGR 3183 Spring 2018
Dr. Mihail Cutitaru
Project 1 (12%) – Draft

Assigned: Wednesday, January 31
**Due: by Wednesday, March 14, 5PM**

**Project Overview**

This projects will implement an elevator control system using any choice of a programming language, although VHDL is highly recommended. The intermediate milestone is optional and requires a high-level simulation of this system, but the simulation will have to be present in the final submission. Work will be done in teams of 3.

**Detailed Description**

The elevator can be used on a building with any number of floors, but for the sake of model simplicity, you can think of only one floor at the beginning and then account for other cases. The controller has to have the following minimum capabilities:

- A user can call the elevator from any given floor (a total of 4 floors – Ground through 3rd) to go either up or down, but special cases have to be considered (i.e. Ground and 3rd). The elevator is called to go to the given direction by pressing the appropriate button.
- The elevator will be sent a *GoUp* or *GoDown* signal to go to the appropriate direction and signal the elevator to move. When none of these two signals are High, the elevator is in a *Wait* state until a call to go to a given floor is made.
- Once the elevator reaches the desired floor, it stops and waits for 1s until it has latched onto the exit doors. A sound will be made (signal *Sound* = 1) when the elevator has reached the correct floor but before the doors open (at the end of latching). The exit doors will open when a control signal *DoorOpen* is sent from the controller after latching. Closing and opening the doors takes 1 second.
- The elevator will move at the speed of 5 seconds per floor. All floors will have two light indicators (store this in register X19 bits 7 downto 4 for UP and 3 downto 0 for DOWN for floors 3 through Ground, respectively) that will be "illuminated" when the elevator is at a given floor and the direction the elevator is going. Update this value as needed during movement. Use masking techniques to read/modify individual bits.
- An IR (signal *IRon*) beam will be used to determine when to close the door and go to user's desired floor. The IR beam will be ON after the doors are open and turn off (reset) with every person moving into the elevator. Once it detects that no more movement is present, it will stay ON for 2 seconds. If there's no additional movement during this time (i.e. IR is ON), then the controller will send a signal *DoorClose* to close the doors.
- Once a user inside the elevator chooses a floor, the floor button selection will be recorded in a memory location (assume 0x1000) by latching it with a *MemWrite* signal when the user presses the given button. The index of the current write address will be stored in register X20. The system will be able to take a new or additional floor selection at any point when a floor selection button is pushed (i.e. when there is at least one person in the elevator). The new floor destination(s) will be stored at sequential incrementing locations from the base address. The

controller has to go through all floor entries in floor order (not necessarily the order that floor selections were pressed). After the requests are fulfilled and the elevator is empty (no user input for 30 seconds), the elevator will go to 1st floor and wait for new user input if the time is 0800 – 1400. The elevator's default floor will be 2nd floor if the time is 1400 – 1800.

- The system should also allow firefighters access by having a special key that allows for all the features an elevator performs in normal operating mode, but also an additional feature: if an elevator is stopped at a given floor, the door is not automatically open – the firefighter has to push the "Door Open" button. The door will stay open, i.e. IR is disabled, as long as the firefighter does not push the "Door Close" button. The firefighters can only insert the key from the outside of the elevator on any floor. When a key is inserted, the elevator will go directly to the floor with the firefighter call regardless of current operation. Once the firefighter key is removed, the elevator will go to the default floor and clear all other stored requests.

**Intermediate milestone is due by Friday, February 16 (additional +5 points of extra credit if submitted)**. It should include the following:

- A complete Mealy Finite State Machine (FSM) detailing the operation of the elevator controller
- A barebones high-level simulation in either C/C++/Java/VHDL/etc implementing the FSM (timing simulation is not required for this milestone, just correct operation)

**Final submission and demo is due by Wednesday, March 14, at 5PM**. Please schedule your demo as early as possible. Final submission will include the following (in addition to content submitted as part of the Intermediate milestone):

- An updated FSM and high-level simulation (if incomplete at Intermediate milestone), including a list of ALL LEGv8 ASM instructions that would be needed to implement this system (include all possible addressing modes needed)
- An extensive discussion on what parts (and how) would need to change in your current project in order to extend this system to one with *n* floors (variable) and *m* elevators (variable)
- An accurate timing simulation covering all possible inputs from any number of users of the elevator
- A report **(hard- and soft-copy)** of minimum 3 pages of actual content, double-spaced, 11-pt Calibri or other sans serif font. Describe your approach (basic design process for the system), problems encountered and solutions designed, and any results obtained and errors.
- A **soft-copy** of the peer- and self-evaluation (on Canvas under *Files/Assignments*) detailing the workload distribution and any comments (one individual submission per team member, these will remain anonymous)
- A 5-10 minute demo of your project by 5PM on Wednesday March 14. Any projects demoed after this time will be deducted 10 points per business day. A test file will be provided.