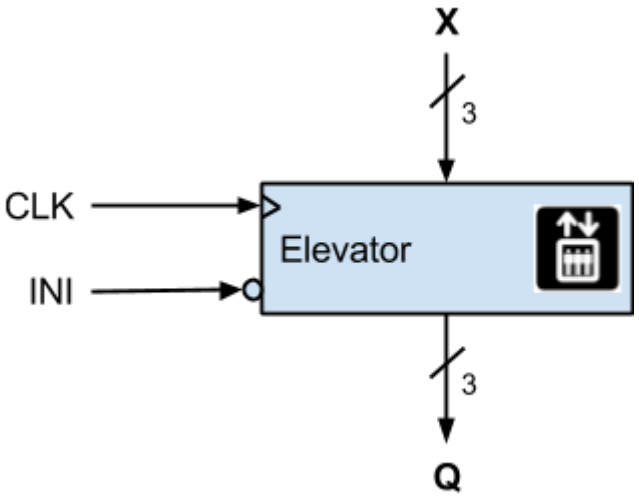
Lab 5: Design and Implementation of an Elevator Circuit Emulator

1 Overview

The goal of this lab is to design and build a circuit that will emulate the controller of an elevator in a building with 7 floors.



The circuit will have the following set of ports:

- A 3-bit output ***Q***, which show the the floor in which the elevator is stopped
- A 3-bit input ***X***, where we must place in binary the desired floor to which we want to go
- A clock input *CLK*
- An input *INI*, of synchronous initialization to (000), active Low.

Both **X** and **Q** encode the floor in binary, i.e., (000) is the ground floor, (001) is the first floor, and so forth until (111), which is the last (seventh) floor.

The output will be (000) in all those cases where *ini* is 0. In those cycles where *ini* is 1, the behavior of the system will be as follows:

- If Q=X the output will keep its value, i.e., we are in the level we wanted.
- If Q>X, the output must be (cycle after cycle) decreasing until Q=X. The elevator must go down because the floor in which it is placed is greater than the commanded floor.
- If Q<X, the output must be increasing until Q=X. In this case, the elevator must go up because the floor in which it is placed is less than the commanded floor.

2 Development

2.1 Design phase

- The circuit design must be completed at home. It must be summarized in the corresponding lab notebook.
- To design the controller a 4-bit modulo 16 up/down counter with parallel load will be used (IC 74169). The three less significant bit of the output are equivalent to the Q output of our system.
- To compare X and Q and determine the direction of the counter, a 4-bit full adder (IC 74283) will be used, as well as inverters and 3-input NAND gates

2.2 Building and debugging phase

- Use the following ICs to mount the circuit:
    - **74169**: A bidirectional modulo 16 synchronous counter (with parallel load signal, active low, which takes precedence over the two signals to enable the count, also active low)
    - **74283**: A 4-bit full adder
    - **7410**: 3 3-input NAND gates
    - **7404**: 6 inverters

- Connect output ***Q*** to the right 7-segment display (that internally includes a binary-to-bcd converter)
- Connect input ***X*** to three switches and also to the left 7-segment display
- Connect input *ini* to an additional switch
- Connect *clk* to a pulse switch
- **Try different combinations of desired floor and current floor** to validate the design. If the system does not work, it must be debugged

2.3 Using the function generator

- Configure the function generator in order to produce a square wave of 1 Hz
- Connect the output of the function generator to the *CLK* input of your circuit
- Connect a pulse switch output to the *ini* input
- When the circuit is correctly working, show your implementation to your lab professor