American University of Beirut
Maroun Semaan Faculty of Engineering and Architecture
Department of Electrical and Computer Engineering

EECE 320: Digital Systems Design
Verilog Project

Elevator Controller

As a new engineer at State Machines, Inc., your first assignment is to design an elevator controller for a three-story building. The controller will be based on a finite state machine (FSM) that operates according to the following specifications:

Inputs:

1. `clk`: Clock signal.
2. `rst`: Asynchronous, active-high, reset signal.
3. `f_req[3:0]`: A 4-bit input vector where each bit represents a request from a particular floor.
   - `f_req[0]`: Request from Ground Floor (GF)
   - `f_req[1]`: Request from Floor 1
   - `f_req[2]`: Request from Floor 2
   - `f_req[3]`: Request from Floor 3
4. `c_req[3:0]`: A 4-bit input vector where each bit represents a request for a particular floor from inside the elevator cabin.
   - `c_req[0]`: Request for Ground Floor (GF)
   - `c_req[1]`: Request for Floor 1
   - `c_req[2]`: Request for Floor 2
   - `c_req[3]`: Request for Floor 3

Outputs:

1. `request[3:0]`: A 4-bit output corresponding to pending floor requests.
   - `request[0]`: Pending request for Ground Floor
   - `request[1]`: Pending request for Floor 1
   - `request[2]`: Pending request for Floor 2
   - `request[3]`: Pending request for Floor 3

2. `current_floor[1:0]`: A 2-bit output corresponding to the current floor the elevator is at.
    - 00: Ground Floor
    - 01: Floor 1
    - 10: Floor 2
    - 11: Floor 3
3. `direction`: 1-bit output indicating elevator's current direction of movement.
    - 0: Moving down
    - 1: Moving up
4. `door_open`: 1-bit output indicating whether the elevator doors are open (1) or closed (0).

Behavior:

1. When the reset signal is asserted, the elevator is assumed to be at the Ground Floor with `direction` = 1 and `door_open` = 0.
2. Assume each floor or cabin request is a clock-synchronized pulse with a duration of one clock cycle. When a floor or cabin request is asserted, the corresponding bit in the output `request` bit-vector should become asserted until the request is fulfilled. Once fulfilled, the corresponding `request` bit should be cleared.
3. Assume floor or cabin requests never occur simultaneously.
4. When a floor request is received, the elevator should move to the requested floor if it is not already there.
5. The elevator can only move one floor every clock cycle.
6. Once the elevator reaches the requested floor, the doors should open (`door_open` = 1) for one clock cycle and then close (`door_open` = 0).
7. If there are multiple pending floor requests, the elevator should prioritize the requests in the following order:
    - Requests in the current direction of movement should be serviced first.
    - Requests in the opposite direction should be serviced only after servicing the current direction's requests.
8. If there are no requests, the elevator should remain stopped at the current floor with the doors closed.

Project Deliverables

1. **State Diagram:**
    - Implement your elevator controller as a Moore FSM.
    - Draw the FSM state diagram, labeling all states and transitions clearly.
2. **Verilog Implementation:**
    - Develop a Vrilog model to implement the FSM for the elevator controller.
    - Use meaningful comments to explain your code.
3. **Testbench:**
    - Develop a Verilog testbench to validate your FSM design.
    - Simulate multiple scenarios, including:
        - Single floor request.
        - Multiple requests in the same direction.
        - Requests in both directions.
        - No requests (idle state).
4. **Simulation:**
    - Use EDA Playground to simulate and test your design.
5. **Report:**
    - Submit a short report (3-4 pages) that includes:
        - Team member names and ID numbers. A team may consist of one or two members only.
        - The state diagram.
        - An explanation of your design.
        - Screenshots of simulation waveforms.

Grading Criteria

- State Diagram and FSM Design (20%)
- Correctness of Verilog Code (30%)
- Testbench Completeness (20%)
- Simulation Results (20%)
- Report Quality (10%)

Submission, Due Date, and Late Penalty

- Please submit the following on Moodle:
  - Verilog source code for the FSM.
  - Verilog testbench.
  - Report (please convert to PDF).
- The project is due by **11:59 PM on Thursday, December 12th, 2024**.
- Starting 12:00 midnight, late submissions will receive a **rolling 10-point deduction every 24 hours**. No submissions will be accepted beyond 00:00 AM on Monday, December 16th, 2024 (midnight Sunday evening).