Dynamic Elevator Controller

2024/2025

Dr. Mohamed Elbably

Team Members

SN	Student Name	Section Number
1	Rawan Gamal Abdullah Hassanien	1
2	Mariam Ahmed Ahmed Hashad	3
3	Manar Ahmed Ahmed Hashad	3
4	Nada Mohamed Rashid	3
5	Shorouk Mostafa Mohamed Hassan	2
6	Shimaa Samer Ahmed Ibrahim	2

Sequence of Controller:

1. **System Initialization**:
   The elevator waits for a valid floor request while ensuring the system is idle with the motor stopped and doors closed.

2. **Receiving a Floor Request**:
   When a floor request is made (`Request_Button`), the system checks the current floor (`Current_Floor`) using the sensor.

3. **Determine Movement Direction**:
   The system compares the requested floor with the current floor:
   - **Move Up** if the requested floor is higher.
   - **Move Down** if the requested floor is lower.
   - **Stay Still** if the requested floor is the same.

4. **Moving Upward**:
   The motor is activated to move the elevator up, incrementing the current floor by one with each step until the requested floor is reached.

5. **Moving Downward**:
   The motor is activated to move the elevator down, decrementing the current floor by one with each step until the requested floor is reached.

6. **Stopping at the Requested Floor**:
   When the current floor matches the requested floor:
   - The motor stops.
   - The door opens to allow passengers to enter or exit.

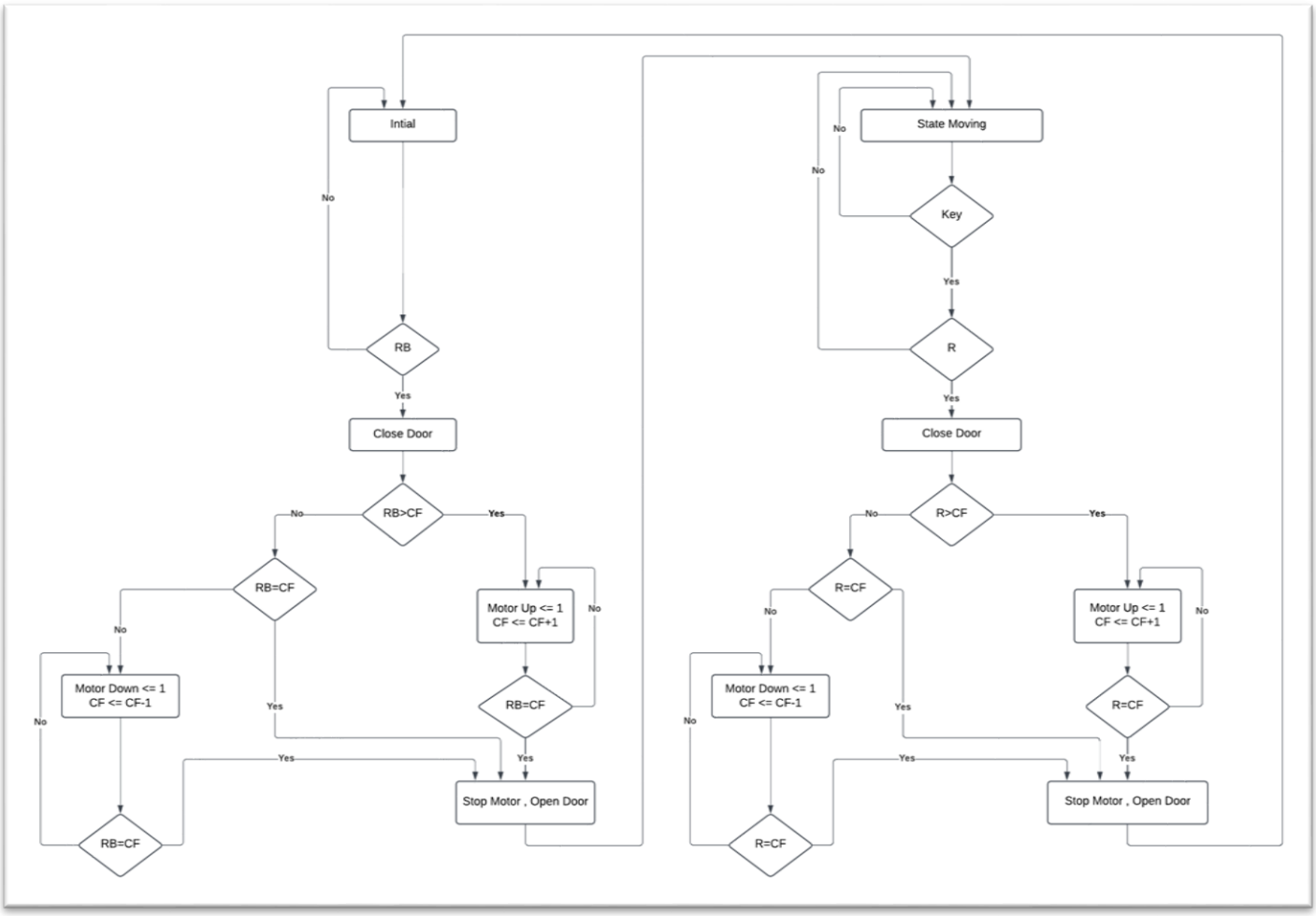
7. **Wait for a Destination Request**:
   After completing the first request, the system waits for a destination floor input (`Request_Floor`).

8. **Determine Destination Direction**:
   The system compares the destination floor with the current floor to decide:
   - **Move Up** if the destination floor is higher.
   - **Move Down** if the destination floor is lower.
   - **Stay Still** if the destination floor is the same.

9. **Moving to the Destination Floor**:
   - The motor activates to move up or down until the destination floor is reached.
   - The current floor is updated with each step.

10. **Arriving at the Destination Floor**:
    When the destination floor is reached:
    - The motor stops.
    - The door opens to allow passengers to exit.
    - The system resets, preparing for the next operation.

ASM Chart:



CSE

VHDL Code:

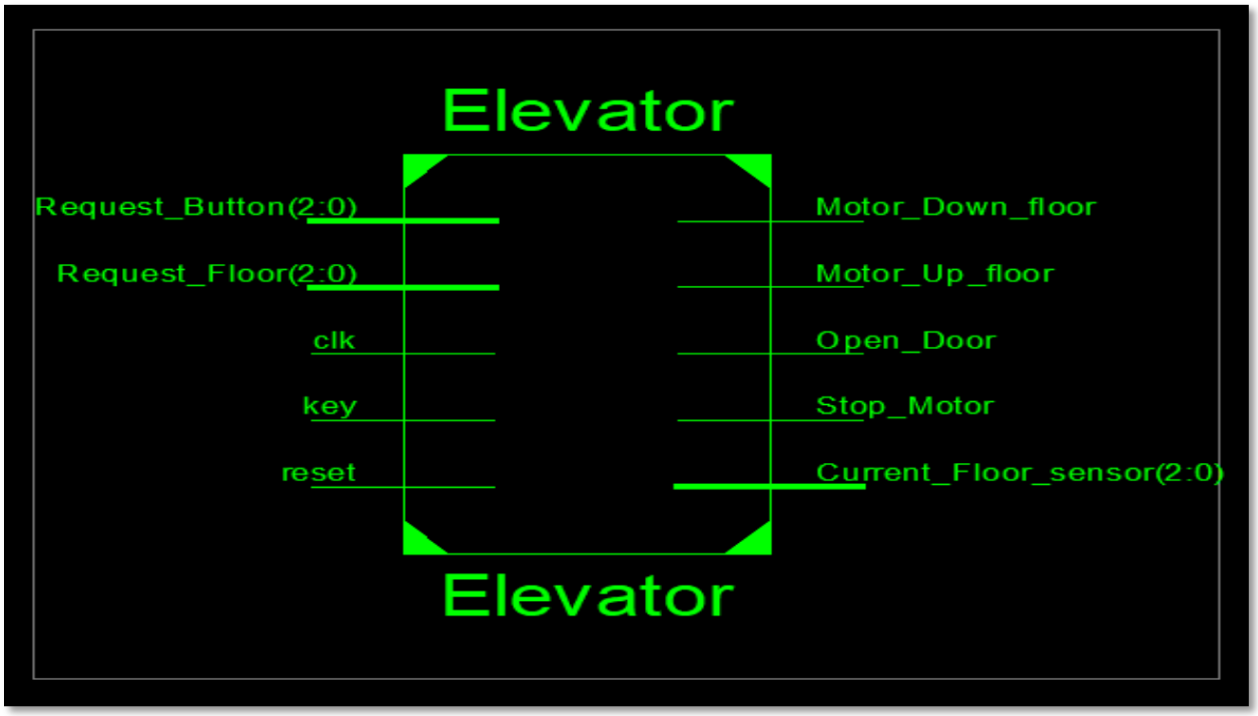
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Elevator_1 is
    Port (
        clk, reset : in STD_LOGIC;
        key: in STD_LOGIC;
        Request_Button, Request_Floor : in STD_LOGIC_VECTOR (2 downto 0);
        Current_Floor_sensor : inout STD_LOGIC_VECTOR (2 downto 0);
        Motor_Up_floor, Motor_Down_floor, Stop_Motor, Open_Door : out STD_LOGIC
    );
end Elevator_1;

architecture Behavioral of Elevator_1 is
    type State is (T1, T2, T3, T4, T5, T6, T7, T8, T9, T10); -- Define states
    signal pr, nxt: State; -- Current and next state
    signal Current_Floor: STD_LOGIC_VECTOR(2 downto 0); -- Current floor as vector

begin

-- Sequential Process: Update current state and synchronize floor transitions
seq: process (clk)
begin
    if rising_edge(clk) then
        if reset = '1' then
            pr <= T1; -- Reset state
            Current_Floor <= "000"; -- Reset to ground floor
        else
            pr <= nxt; -- Move to the next state

            -- Update current floor during transitions
            if nxt = T3  or nxt = T8 then
                Current_Floor <= Current_Floor + "001"; -- Move up
            elsif nxt = T5 or nxt = T10 then
                Current_Floor <= Current_Floor - "001"; -- Move down
            end if;
        end if;
    end if;
```

```
end process seq;


-- Combinational Process: Determine next state and control signals
comb: process (pr, Request_Button, Request_Floor, key, Current_Floor)
begin

    nxt <= pr; -- Default: remain in current state

    case pr is
        -- Idle state
        when T1 =>
                                        -- Default outputs
                                        Motor_Up_floor <= '0';
                                        Motor_Down_floor <= '0';
                                        Stop_Motor <= '1';
                                        Open_Door <= '1';

            if Request_Button >= "000" then
                nxt <= T2; -- Go to request evaluation
            end if;

        -- Evaluate button request
        when T2 =>
            if Request_Button > Current_Floor then
                nxt <= T3; -- Move up
            elsif Request_Button = Current_Floor then
                nxt <= T4; -- Stop and open door
            else
                nxt <= T5; -- Move down
            end if;

        -- Moving up
        when T3 =>
             Motor_Up_floor <= '1'; -- Elevator moves
                                        Motor_Down_floor <= '0';
                                        Stop_Motor <= '0';
                                        Open_Door <= '0';

            if Request_Button = Current_Floor then
                nxt <= T4; -- Stop when target is reached
            end if;
```

```
-- Stop and open door
when T4 =>
     Motor_Up_floor <= '0';
                                  Motor_Down_floor <= '0';
    Stop_Motor <= '1';
    Open_Door <= '1';
    nxt <= T6; -- Wait for key input to move again

-- Moving down
when T5 =>
    Motor_Down_floor <= '1'; -- Elevator moves down
                                 Motor_Up_floor <= '0';
    Stop_Motor <= '0';
    Open_Door <= '0';
    if Request_Button = Current_Floor then
        nxt <= T4; -- Stop when target is reached
    end if;

-- Wait for key input
when T6 =>
    if key = '1' then
        if Request_Floor >= "000" then
            nxt <= T7; -- Evaluate floor request
        end if;
    end if;

-- Evaluate floor request
when T7 =>
    if Request_Floor > Current_Floor then
        nxt <= T8; -- Move up
    elsif Request_Floor = Current_Floor then
        nxt <= T9; -- Open door at the requested floor
    else
        nxt <= T10; -- Move down
    end if;

-- Moving up to the requested floor
when T8 =>
    Motor_Up_floor <= '1';
                                 Motor_Down_floor <= '0';
    Stop_Motor <= '0';
    Open_Door <= '0';
    if Request_Floor = Current_Floor then
```

```
            nxt <= T9; -- Stop when target is reached
        end if;

    -- Open door at the requested floor
    when T9 =>
                    Motor_Up_floor <= '0';
                                Motor_Down_floor <= '0';
        Stop_Motor <= '1';
        Open_Door <= '1';
        nxt <= T1; -- Go back to idle state

    -- Moving down to the requested floor
    when T10 =>
        Motor_Down_floor <= '1';
                                Motor_Up_floor <= '0';
        Stop_Motor <= '0';
        Open_Door <= '0';
        if Request_Floor = Current_Floor then
            nxt <= T9; -- Stop when target is reached
        end if;

   end case;
end process comb;


-- Output current floor to the sensor
Current_Floor_sensor <= Current_Floor;

end Behavioral;
```

ASIC Chip:



YouTube Video:

https://youtu.be/JfYo_V256fk?si=8RFi9781_8G1_cah