Lab 4A: Traffic Light Controller

1. Objective

The objective of this lab is to design and implement a traffic light controller for a four-way intersection. The controller should manage traffic signals for both North-South and East-West directions, handle pedestrian crossing requests, and provide an emergency override mechanism. A timer module is used to enforce correct timing of signal phases.

2. Specification

The traffic light controller must satisfy the following requirements:

- **Intersection Control**
    - North-South (NS) and East-West (EW) traffic lights.
    - Normal cycle: Green (30s) → Yellow (5s) → Red.
- **Emergency Override**
    - During emergency, all lights should turn red with flashing.
    - Normal operation resumes when emergency is cleared.
- **Pedestrian Crossing**
    - Pedestrian request triggers a dedicated crossing state.
    - Vehicles see red lights, and pedestrian walk signal is active for 10s.
- **Startup Condition**
    - Controller begins in a flashing red state before normal operation starts.
- **Clock Input**
    - System runs on a **1 Hz clock** (1-second resolution).

- Students design a timer for time delays.

3. Design Methodology

1. **FSM State Design**
    - The controller was modeled as a Finite State Machine (FSM).
    - States were defined for each light phase, pedestrian crossing, and emergency mode.
    - Transition conditions were carefully derived based on timer values and input signals.
2. **States Used**
    - **STARTUP_FLASH**: Initial blinking red lights.
    - **NS_GREEN_EW_RED**: NS traffic green, EW red.
    - **NS_YELLOW_EW_RED**: NS yellow, EW red.
    - **NS_RED_EW_GREEN**: NS red, EW green.
    - **NS_RED_EW_YELLOW**: NS red, EW yellow.
    - **PEDESTRIAN_CROSSING**: Both NS and EW red, pedestrian walk enabled.
    - **EMERGENCY_ALL_RED**: Emergency mode, all red flashing.
3. **Timer Design**
    - A 6-bit timer was implemented to count seconds (0–63 range).
    - Timer resets at the beginning of each state to ensure accurate phase timing.
4. **Pedestrian Request Handling**
    - A **pedestrian latch** captures button presses even if the FSM is busy.
    - Pedestrian crossing is served after the current green/yellow cycle finishes.

5. **Emergency Handling**
    - Immediate transition to **EMERGENCY_ALL_RED** upon emergency signal.
    - Lights flash red until the emergency signal is cleared.
    - On clearing, controller restarts from the startup flash state.

---

4. State Diagram

The state diagram was designed to include all operational modes:

- **Normal Cycle**:
    - `NS_GREEN_EW_RED → NS_YELLOW_EW_RED → NS_RED_EW_GREEN → NS_RED_EW_YELLOW → NS_GREEN_EW_RED`
- **Pedestrian Handling**:
    - During a yellow phase, if a pedestrian request is latched, transition goes to **PEDESTRIAN_CROSSING** before resuming the next normal cycle.
- **Emergency Handling**:
    - Any state transitions immediately to **EMERGENCY_ALL_RED** when emergency input is high.
- **Startup**:
    - System begins with **STARTUP_FLASH** for 5s, then transitions to normal operation.

---

5. Simulation Scenarios

The design was verified through a testbench that simulated different traffic and pedestrian events. Key scenarios include:

1. **Startup Behavior**
    - Controller begins with flashing red for 5 seconds.
    - Then transitions to normal operation with NS green.
2. **Normal Operation**
    - NS green for 30s → NS yellow for 5s → EW green for 30s → EW yellow for 5s.
    - This cycle repeats continuously.
3. **Pedestrian Crossing**
    - A pedestrian request during NS or EW yellow is latched.
    - After yellow, FSM transitions to **PEDESTRIAN_CROSSING** for 10s.
    - Then it returns to the correct traffic state to maintain fairness.
4. **Emergency Mode**
    - When emergency is triggered, FSM immediately transitions to **EMERGENCY_ALL_RED**.
    - Lights flash red until emergency is cleared.
    - After clearing, system restarts with startup flash and resumes normal operation.

---

6. Results

- The FSM correctly managed all signal transitions with accurate timing.
- Pedestrian requests were served without disrupting the traffic fairness cycle.
- Emergency override worked immediately and reliably.
- The startup flash ensured safety before normal operation began.

7. Conclusion

In this lab, a traffic light controller was successfully designed and verified. The system effectively handled normal traffic flow, pedestrian requests, and emergency overrides. The modular design with a separate timer and FSM approach made the system robust and easy to understand.

This project demonstrates how digital design techniques can be applied to real-world control systems like traffic management.