Traffic Light Controller FSM Design in SystemVerilog

Purpose: To design and implement a digital traffic light controller using SystemVerilog.

Functionality: Manages traffic flow at a two-way intersection (main road vs. minor road).

Key Feature: Sensor-actuated minor road green light.

Technology: Hardware Description Language (HDL) for FPGA/ASIC implementation.

Design Overview

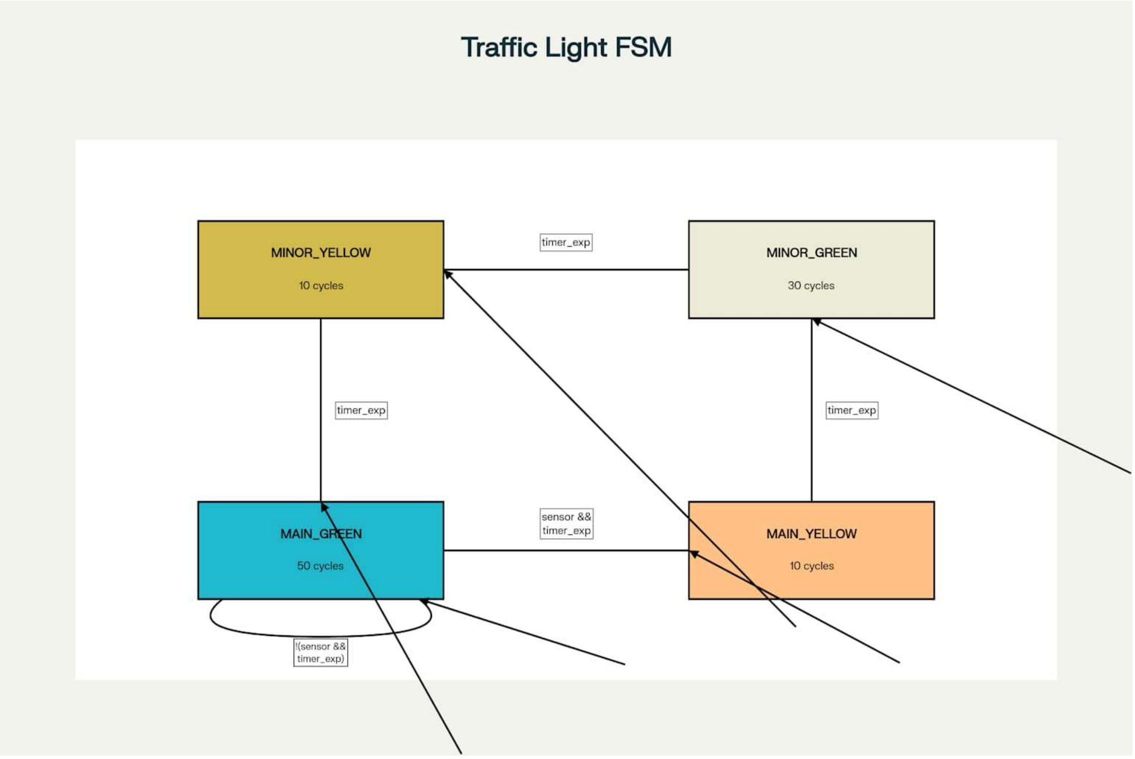
The traffic light controller manages a typical intersection with a **main road** (high-priority) and a **minor road** (low-priority). The system uses a sensor-based approach where the main road maintains priority unless vehicles are detected on the minor road.

The FSM implements a **Moore state machine** where outputs depend only on the current state, ensuring predictable and safe operation. This approach provides better timing control and eliminates glitches that might occur in Mealy machines.

FSM States and Transitions

The controller uses **four distinct states** to manage the traffic flow:

- **MAIN_GREEN**: Main Road has green light, minor road has red light
- **MAIN_YELLOW**: Main Road transitions to yellow, minor road remains red
- **MINOR_GREEN**: Main Road has red light, minor road has green light
- **MINOR_YELLOW**: Main Road remains red, minor road transitions to yellow



Traffic Light FSM State Diagram showing the four states and their transitions with timing information

The state transitions are controlled by:

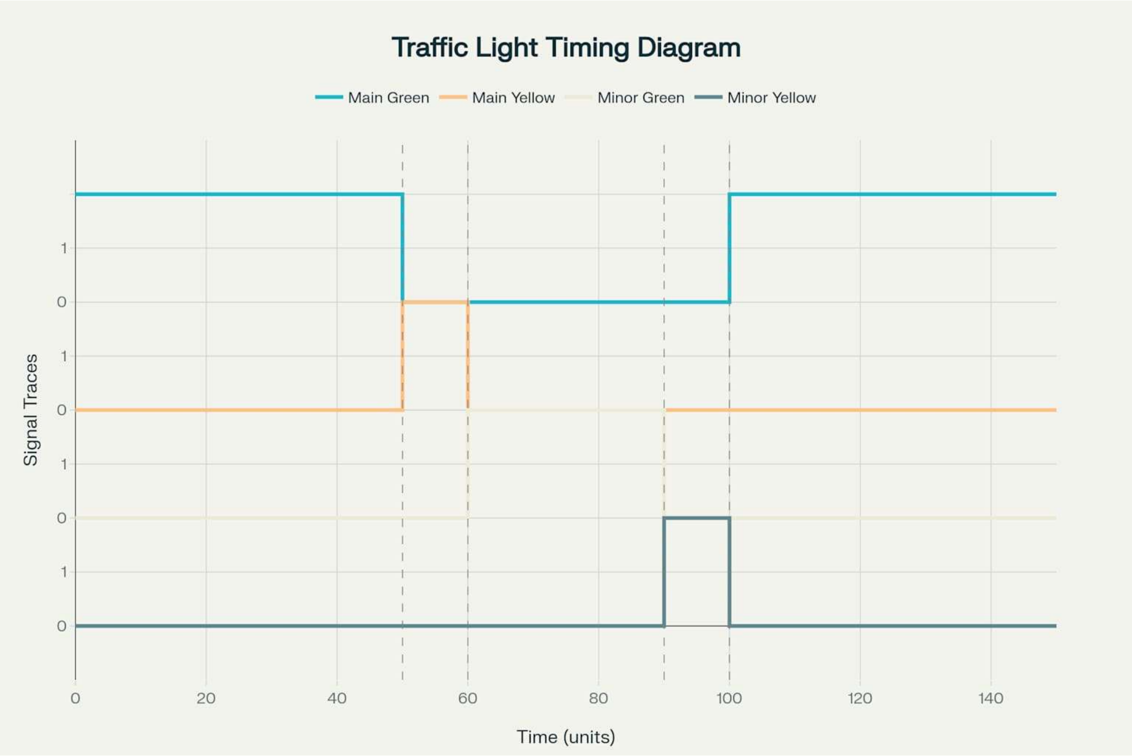
- **Timer expiration**: Ensures minimum phase durations
- **Sensor input**: Detects vehicles on minor road
- **Safety logic**: Prevents conflicting green lights

Timing Sequences and Phase Control

The controller implements **precise timing control** for each phase to ensure safe traffic flow:

Phase	**Duration**	**Purpose**
Main Green	50 cycles	Primary traffic flow on main road

Phase	Duration	Purpose
Main Yellow	10 cycles	Warning before switching to minor road
Minor Green	30 cycles	Allow minor road traffic to proceed
Minor Yellow	10 cycles	Warning before returning to main road



Traffic Light Timing Diagram showing the sequence of light phases over time

The timing sequence ensures that:

- **Yellow phases** provide adequate warning time
- **Green phases** allow sufficient traffic flow
- **Sensor activation** only triggers changes when timing permits

Key Design Features

Modern System Verilog Constructs

The implementation leverages **SystemVerilog best practices**:

- **typedef enum**: Creates readable state definitions

- **always_ff**: Explicitly indicates sequential logic blocks
- **always_comb**: Clearly marks combinational logic blocks
- **logic**: Modern data type replacing reg/wire distinctions
- **localparam**: Parameterized timing values for easy modification

Safety Mechanisms

The design incorporates multiple **safety features**:

- **Default red lights**: All outputs default to red for fail-safe operation
- **Mutex operation**: Only one direction gets green light at any time
- **Proper sequencing**: Yellow always transitions between green and red
- **Timer-based control**: Prevents premature state changes

Testing and Verification

The design includes a comprehensive **testbench** that validates:

- **Reset behaviour**: Proper initialization to safe state
- **Normal operation**: Cycling through all states with correct timing
- **Sensor response**: Appropriate reaction to vehicle detection
- **Timing accuracy**: Verification of all phase durations

The testbench generates stimulus patterns and monitors outputs to ensure correct FSM behaviour under various operating conditions.

Benefits and Applications

This FSM design offers several advantages:

1. **Simplicity**: Easy to understand and modify
2. **Safety**: Built-in fail-safe mechanisms
3. **Efficiency**: Optimized for minimal hardware resources
4. **Scalability**: Can be extended for more complex intersections

   The controller is suitable for **simple intersections** where one road has clear priority over another, such as highway-farm road intersections or main street-side street configurations.