Working of the Traffic light controller

The traffic light controller is implemented using **behavioral Verilog** and is based on a **Finite State Machine (FSM)** approach. The design controls a four-way traffic intersection and ensures that traffic flows safely by allowing only one direction to receive a green signal at any given time. The controller operates synchronously with a clock signal and uses a reset input to initialize the system.

At the core of the design is a **state register**, which holds the current state of the controller. Each state represents a specific traffic condition, such as one direction having a green or yellow signal while all other directions remain red. The FSM cycles through a fixed sequence of states corresponding to each direction in turn. The state transitions occur on the **positive edge of the clock**, ensuring synchronized and predictable operation.

When the **reset** signal is asserted, the controller immediately enters the initial state, typically the **North-Green** state. At this point, the counter is cleared and the traffic signal for the north direction is set to green, while all other directions are set to red. This ensures a known and safe starting condition for the system.

The FSM uses a **counter-based timing mechanism** to control how long each state remains active. For each state, a predefined duration is assigned (for example, a longer time for green and a shorter time for yellow). The counter increments on every clock cycle. As long as the counter value is less than the specified duration for the current state, the FSM remains in that state. Once the counter reaches the defined limit, it is reset, and the FSM transitions to the next state in the sequence.

The **next-state logic** is implemented using a combinational `case` statement that examines the current state and determines the appropriate next state. For example, after the North-Green state completes its duration, the FSM transitions to the North-Yellow state. After the yellow duration expires, control moves to the East-Green state, and the sequence continues through East-Yellow, South-Green, South-Yellow, West-Green, and West-Yellow before returning to North-Green. This cyclic behavior allows continuous and orderly traffic flow.

The **output logic** of the controller is also state-based. For each FSM state, the Verilog code assigns values to the traffic light outputs such that only the active direction shows green or yellow, while all others remain red. Because the outputs depend only on the current state and not directly on inputs, the controller behaves as a **Moore FSM**, which simplifies timing and improves reliability.

Overall, the Verilog traffic light controller demonstrates a clean separation between **state storage**, **state transition logic**, **timing control**, and **output generation**. By using synchronous logic, counters for timing, and a structured FSM approach, the design ensures safe, predictable, and easily extensible traffic signal control suitable for simulation and synthesis.