Design FSMs using Verilog or VHDL in the VLSI software environment. Write testbenches to test different states and transitions of the FSM. Simulate the FSM behavior and verify its correctness.//in giva a code in python//

Input program

class TrafficLightFSM:

# Define the states

RED = 'RED'

GREEN = 'GREEN'

YELLOW = 'YELLOW'

def \_\_init\_\_(self):

# Initialize the FSM with the default state

self.state = TrafficLightFSM.RED

self.timer = 0

def next\_state(self):

"""Transition to the next state based on the current state."""

if self.state == TrafficLightFSM.RED:

if self.timer >= 5: # 5 time units in RED state

self.state = TrafficLightFSM.GREEN

self.timer = 0

else:

self.timer += 1

elif self.state == TrafficLightFSM.GREEN:

if self.timer >= 5: # 5 time units in GREEN state

self.state = TrafficLightFSM.YELLOW

self.timer = 0

else:

self.timer += 1

elif self.state == TrafficLightFSM.YELLOW:

if self.timer >= 3: # 3 time units in YELLOW state

self.state = TrafficLightFSM.RED

self.timer = 0

else:

self.timer += 1

def get\_state(self):

"""Returns the current state of the traffic light."""

return self.state

# Testing the FSM

if \_\_name\_\_ == "\_\_main\_\_":

fsm = TrafficLightFSM()

# Simulate the FSM for 20 clock cycles

for i in range(20):

print(f"Time: {i}, Traffic Light: {fsm.get\_state()}")

fsm.next\_state(

outputprogram

Time: 0, Traffic Light: RED

Time: 1, Traffic Light: RED

Time: 2, Traffic Light: RED

Time: 3, Traffic Light: RED

Time: 4, Traffic Light: RED

Time: 5, Traffic Light: RED

Time: 6, Traffic Light: GREEN

Time: 7, Traffic Light: GREEN

Time: 8, Traffic Light: GREEN

Time: 9, Traffic Light: GREEN

Time: 10, Traffic Light: GREEN

Time: 11, Traffic Light: GREEN

Time: 12, Traffic Light: YELLOW

Time: 13, Traffic Light: YELLOW

Time: 14, Traffic Light: YELLOW

Time: 15, Traffic Light: YELLOW

Time: 16, Traffic Light: RED

Time: 17, Traffic Light: RED

Time: 18, Traffic Light: RED

Time: 19, Traffic Light: RED