# Assignment 2

Fundamentals of Digital System Design

Due Date: November 9, 2024

---

This assignment is based on the traffic light system explained below. In this assignment, you have to do the following,

1. State with reason which type of FSM (Moore or Mealy machine) is more suitable for the traffic light controller module (Fig. 2) in the description. (**Hint**: if the external timer will count k amount of time after **amber_timer_en** is high, we have to make sure that the system is in the amber state for exactly k time. Hence the timing of when the **amber_timer_en** signal becomes high is important.)

2. Make a **state diagram** and a **state transition table** of the FSM.

The completed assignment should be submitted as a .pdf file **on or before 11.59pm, November 9, 2024**.
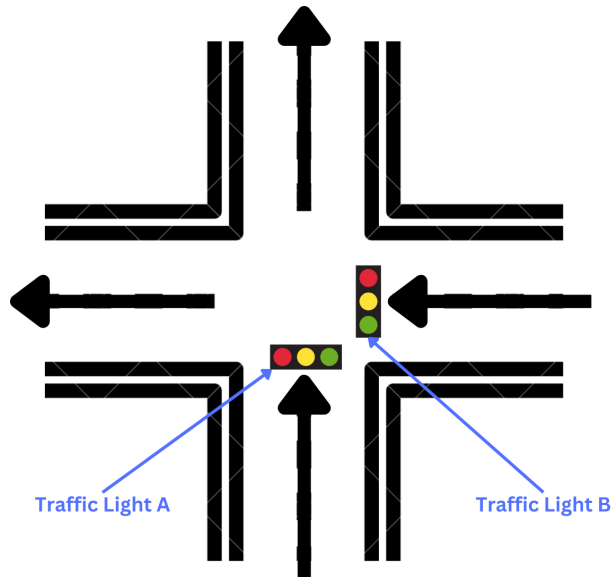
## Traffic Light System Description



Figure 1: Traffic Light Configuration

The Figure 1 shows the traffic lights in this traffic light system. It contains two traffic lights for the two roads. In the roads, the vehicles move in the

direction shown by the arrows. Both the traffic lights are equipped with a camera that evaluate whether there is traffic behind it. The traffic light controller will control both lights considering the level of traffic in the two roads. An external timer module is present to ensure that the traffic light shows the "Amber" colour for a required amount of time. The traffic light A and traffic light B in the Figure 1 cannot be both "Green" at the same time. The traffic light A has more priority over traffic light B.

The Figure 2 shows the I/O of the traffic light controller module for this scenario. The Table 1 gives the description of each I/O signal.
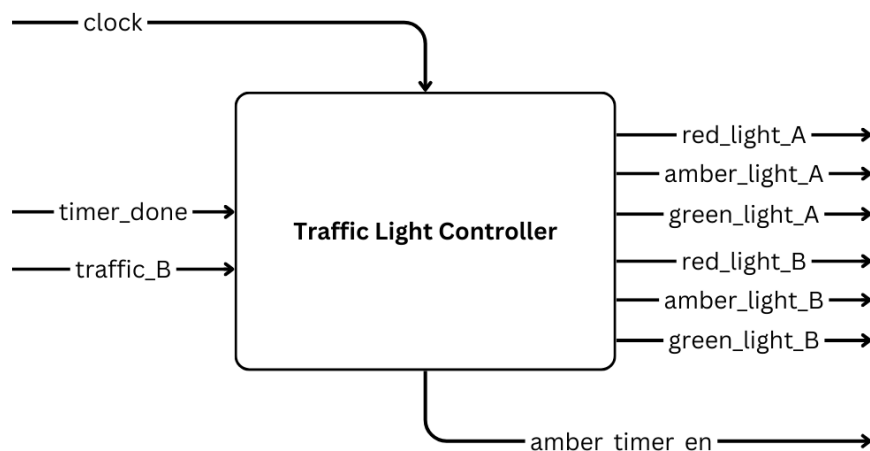
Figure 2: Traffic Light Module I/O

| Signal | Type | Description |
|---|---|---|
| clock | input | clock signal |
| traffic_B | input | 1: traffic in B **and** not traffic in A, 0: traffic in A **or** not traffic in B |
| timer_done | input | 1: timer is done, 0: timer is on going |
| amber_timer_en | output | 1: enable the amber timer in the timer module |
| red_light_A | output | on/off red light in traffic light A |
| amber_light_A | output | on/off amber light in traffic light A |
| green_light_A | output | on/off green light in traffic light A |
| red_light_B | output | on/off red light in traffic light B |
| amber_light_B | output | on/off amber light in traffic light B |
| green_light_B | output | on/off green light in traffic light B |

Table 1: Traffic Light Controller I/O Description

At the start of the system, the traffic light A is **Green** while the traffic light B is **Red** since traffic light A has priority. It will stay in that manner

until **traffic_B** signal becomes high indicating that there is traffic in B and not in A. When the **traffic_B** is high, the traffic light A should go from **Green** to **Amber**. When it is **Amber**, it should enable (make high) the **amber_timer_en** to start a timer in the external timer module. Once the **timer_done** signal becomes high (when traffic light A is showing Amber), traffic light A should become **Red** while the traffic light B should become **Green**. The two traffic lights would keep displaying these colours until the **traffic_B** signal becomes low. When it becomes low, the traffic light B should become **Amber**. While in this state the **amber_timer_en** should become high. Once the **timer_done** signal becomes high, the traffic light B should become **Red** and the traffic light A should become **Green**.

Please note that the **traffic_B** signal would not have any effect when the traffic light is showing Amber.
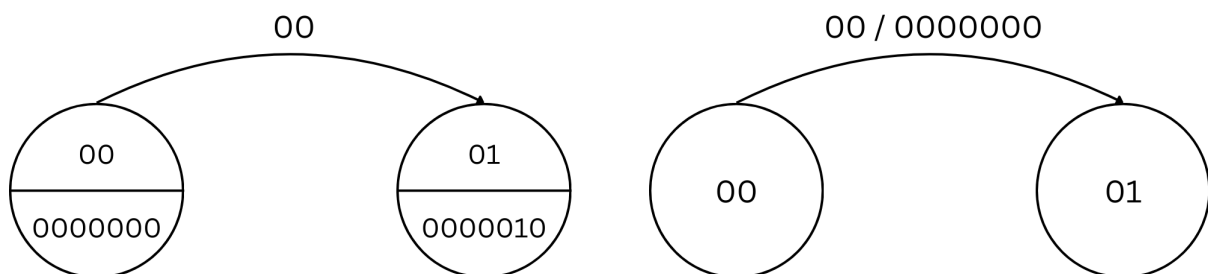
Please use the states in the Table 2 when making the FSM.

| State | Bit Pattern | Description |
|-------|-------------|-------------|
| $G_A R_B$ | 00 | Traffic light A is Green and traffic light B is Red |
| $A_A R_B$ | 01 | Traffic light A is Amber and traffic light B is Red |
| $R_A G_B$ | 10 | Traffic light A is Red and traffic light B is Green |
| $R_A A_B$ | 11 | Traffic light A is Red and traffic light B is Amber |

Table 2: States of the FSM

## Additional Information

The state diagram of an FSM with 2 inputs and 7 outputs can be drawn as shown below.

Please use the following bit-indexing scheme for inputs and outputs when making the state diagram and the state transition table.

## OO
[1] [0]

| Index | Description |
|-------|-------------|
| 0 | traffic_B |
| 1 | timer_done |

## OOOOOOO
[6] [5] [4] [3] [2] [1] [0]

| Index | Description |
|-------|-------------|
| 0 | red_light_A |
| 1 | amber_light_A |
| 2 | green_light_A |
| 3 | red_light_B |
| 4 | amber_light_B |
| 5 | green_light_B |
| 6 | amber_timer_en |