DIGITAL DESIGN : CS-221

Practical examples of FSMs

Date: 29th October, 2013

Submitted by:

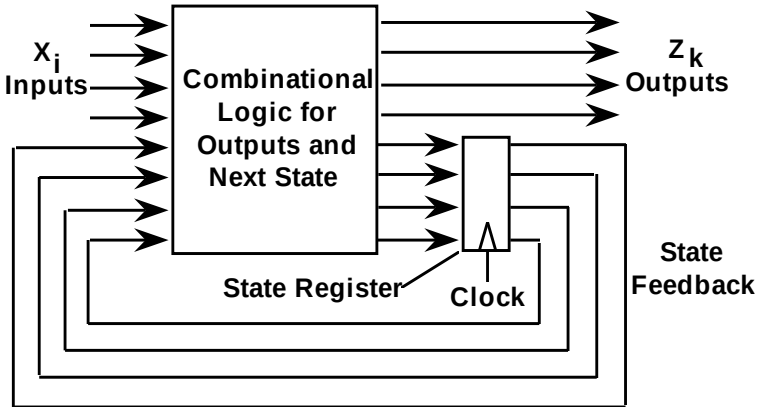
Radhika Patodiya – 120101054

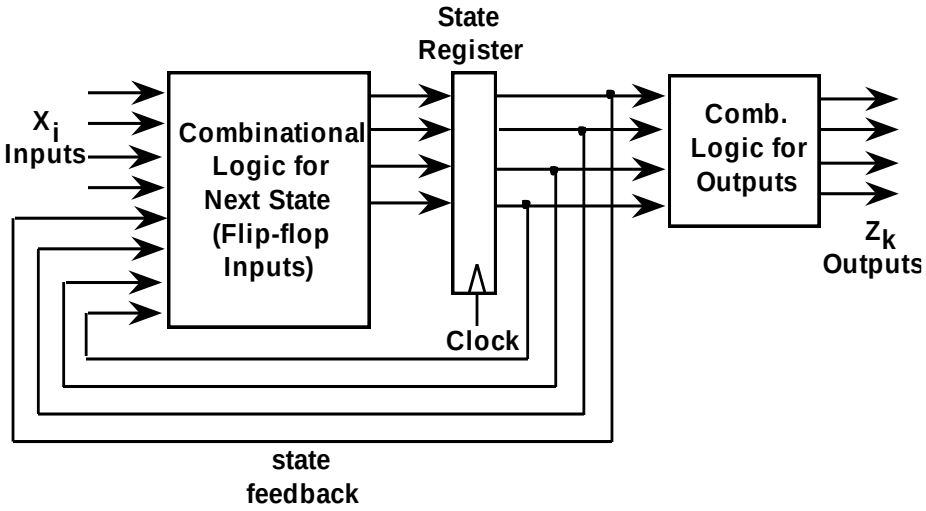
Anmol Kalia – 120123006

Soumya Wadhwa – 120123040

Introduction

Moore machine and Mealy machine





Moore Machine

- Outputs are functions solely of the current state.
- Outputs change synchronously with state changes.

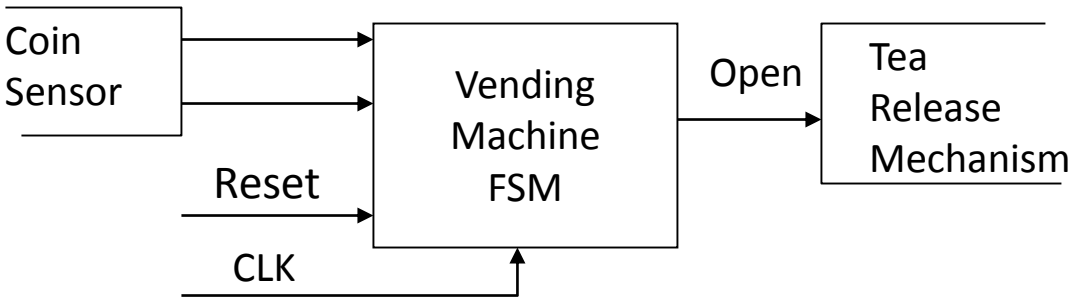
Mealy Machine

- Outputs depend on the current state as well as inputs.
- Input change causes an immediate output change.
- Asynchronous signals.

Simple Vending Machine

General machine concept

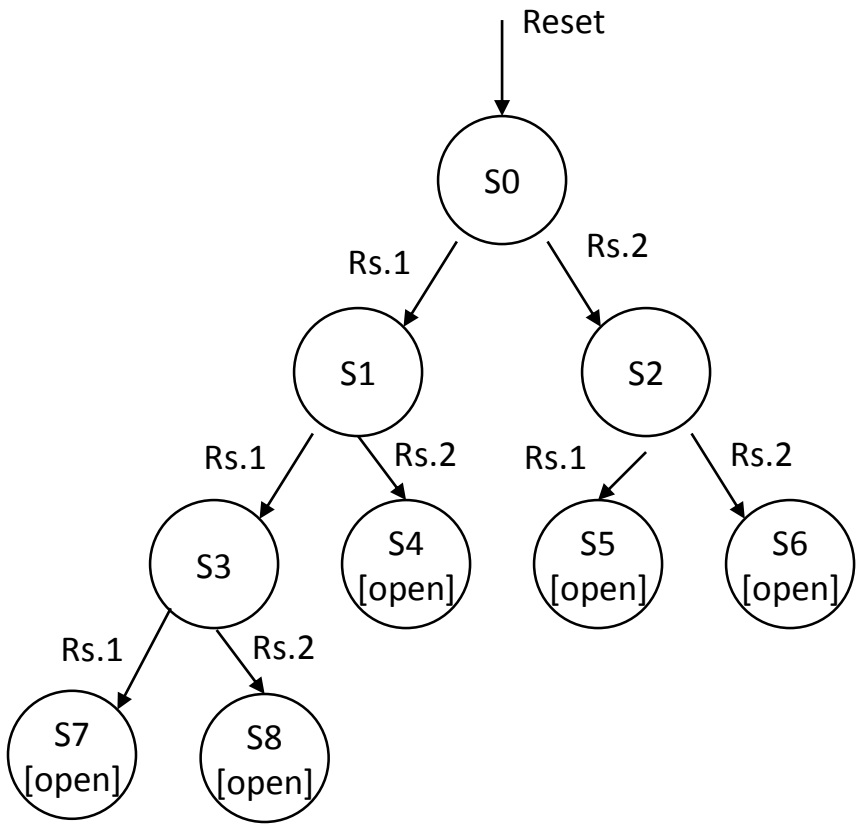
- Deliver tea after Rs.3 have been deposited.
- Single slot for one coin of Rs.1 or Rs.2 at a time.
- No further change after vending tea, even on adding coins, unless the machine is reset.



Vending Machine block diagram

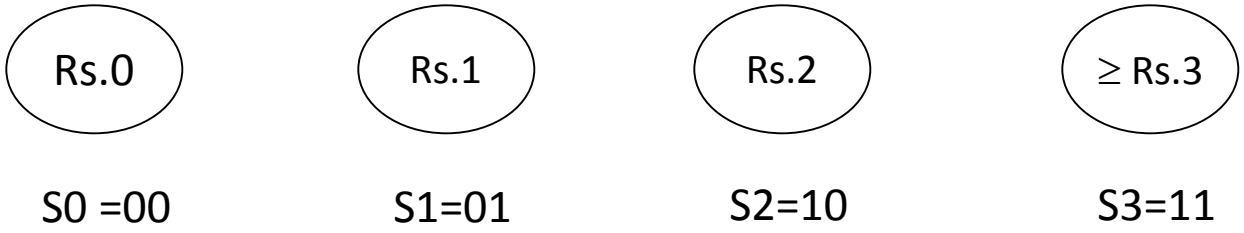
Suitable Abstract Representation

- Tabulate typical input sequences:
  - Three Rs.1 coins
  - One Rs.1 coin, then one Rs.2 coin
  - One Rs.2 coin, then one Rs.1 coin
  - Two Rs.2 coins
- Draw state diagram:
  Inputs: Rs.1, Rs.2, reset
  Output: open chute
- Assumptions:
  - Assume Rs.1 and Rs.2 asserted for one cycle
  - Each state has a self loop for no coin
  - Only one coin is inserted at a time

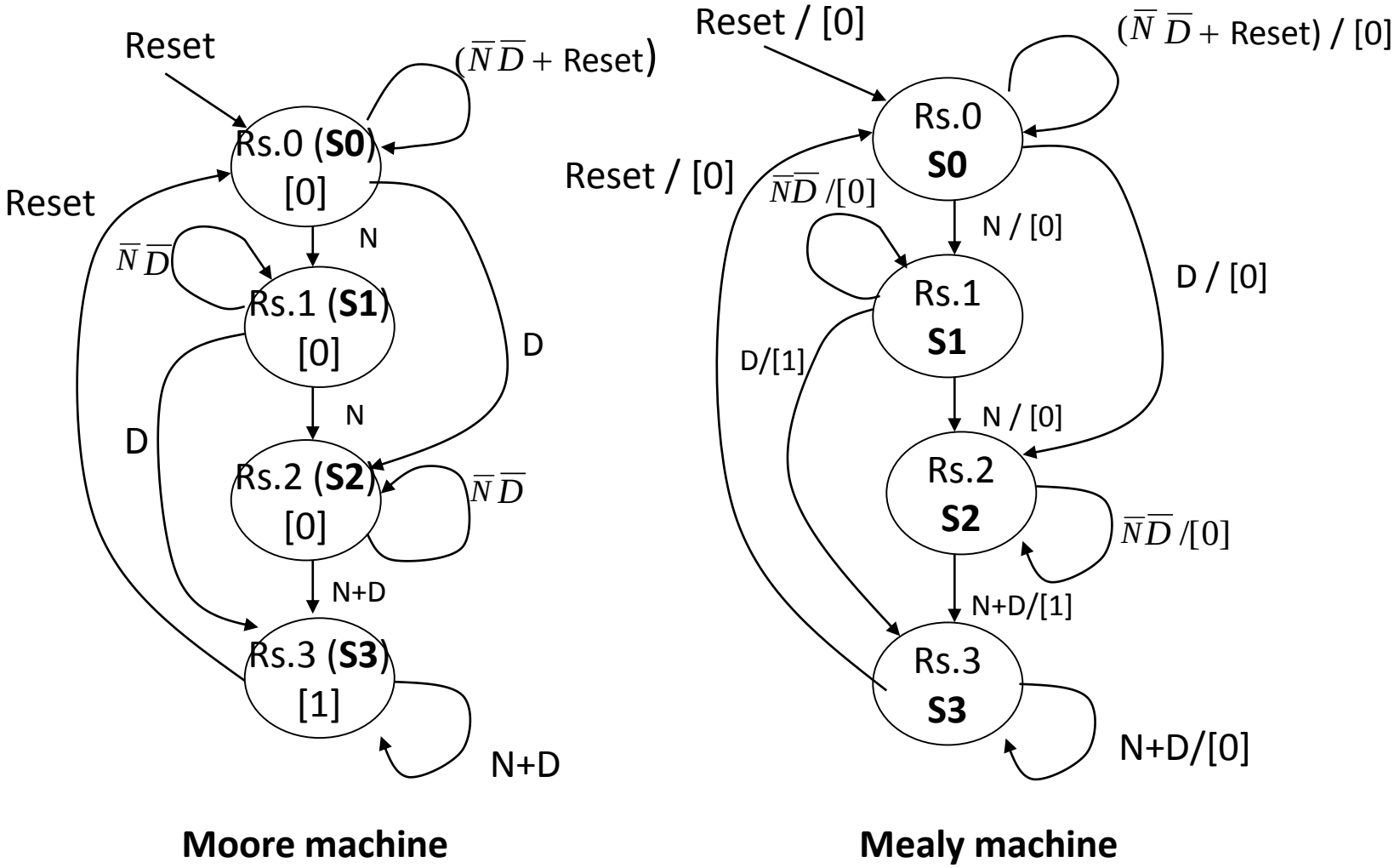


In order to draw **Moore** and **Mealy** Finite State Diagrams, we need to minimize number of states and also reuse states wherever possible.

States after minimisation



Let's denote Rs.1 by N and Rs.2 by D.
[0]=No output
[1]=Tea output



Moore machine **Mealy machine**

Moore and Mealy machine state diagrams for the vending machine FSM

NOTE: In the Moore Machine, the state S3 poses a problem since the self loop indicates a repeated tea output without the resetting of the machine, which is not desirable. Hence, we can use an indicator circuit which inhibits the release of the tea in the state S3 till it is reached again after a reset.

In such a case, we can also switch to another state S4 where there is no tea output.

Therefore, a simple tea vending machine can be realized using sequential logic circuits.

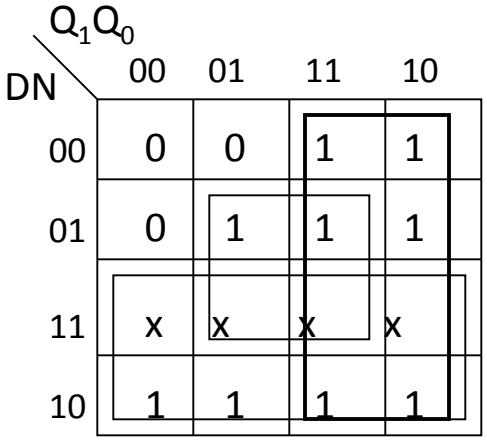
State transition table for Moore and Mealy machines (Next state gives D f/f excitation too)

Present State		Inputs		Next State		Moore Output	Mealy Output
Q_1	Q_2	D	N	Q_1^+	Q_2^+	Open	Open
0	0	0	0	0	0	0	0
		0	1	0	1	0	0
		1	0	1	0	0	0
		1	1	x	x	x	x
0	1	0	0	0	1	0	0
		0	1	1	0	0	0
		1	0	1	1	0	1
		1	1	x	x	x	x
1	0	0	0	1	0	0	0
		0	1	1	1	0	1
		1	0	1	1	0	1
		1	1	x	x	x	x
1	1	0	0	1	1	1	1
		0	1	1	1	1	1
		1	0	1	1	1	1
		1	1	x	x	x	x

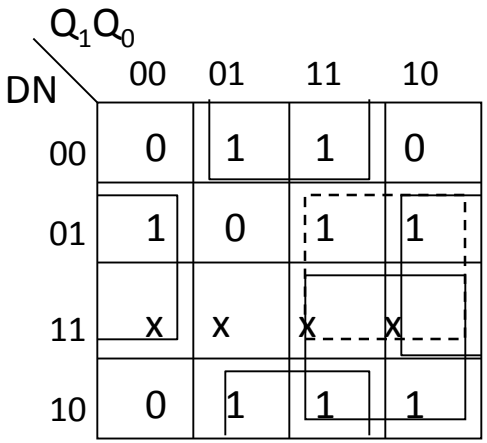
$Q^+ = D$

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

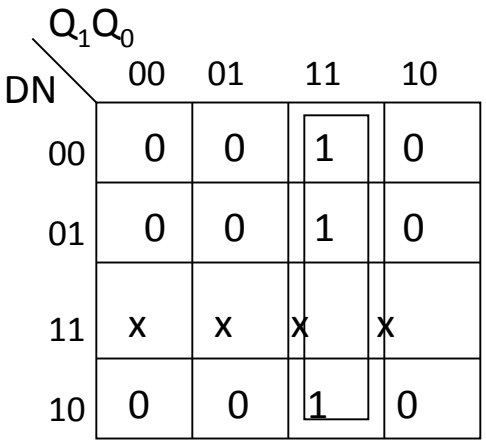
Implementation using D-flip/flop



K-map for D_1

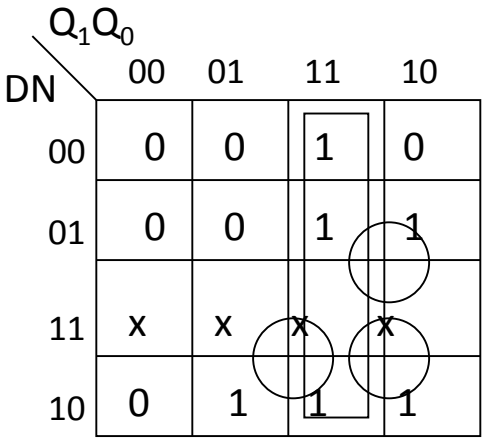


K-map for D_0



K-map for Open (Moore)

- $D_1 = Q_1 + D + Q_0 \cdot N$
- $D_0 = N \cdot \overline{Q_0} + Q_0 \cdot \overline{N} + Q_1 \cdot N + Q_1 \cdot D$
- Moore OPEN = $Q_1 \cdot Q_0$
- Mealy OPEN = $Q1 \cdot Q_0 + D \cdot Q_0 + D \cdot Q_1 + N \cdot Q_1$



K-map for Open (Mealy)

Traffic Light Controller

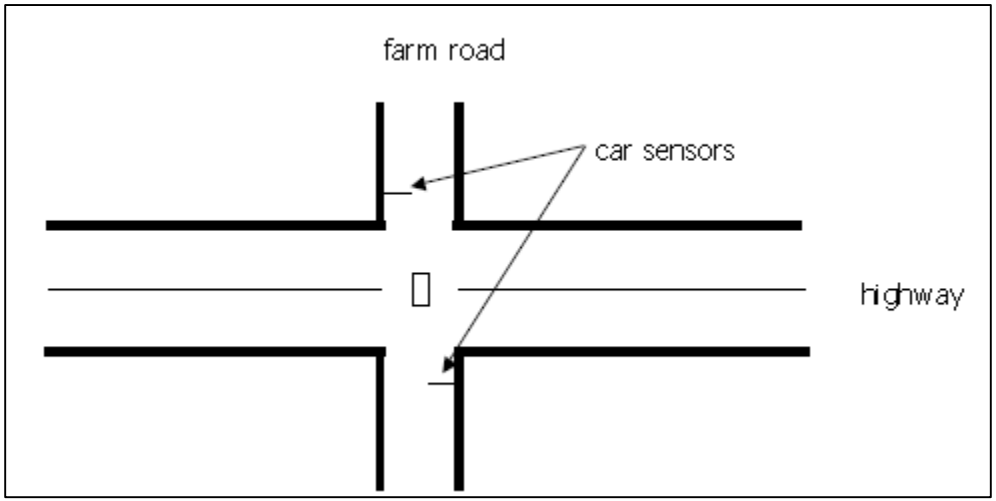
General machine concept

- A busy highway is intersected by a little used farm road.
- Detectors C sense the presence of vehicles waiting on the farm road.
- With no vehicle on the farm road, the lights remain green in the highway direction.
- If there is a vehicle on the farm road, the highway lights go from green to yellow to red, allowing the farm road lights to become green.
- These stay green only as long as a vehicle is detected on the farm road but never longer than a set interval.
- When these conditions are not met, the farm lights transition from green to yellow to red, allowing the highway lights to return to green.
- Even if the farm road vehicles are waiting, the highway lights remain green atleast for a set interval.
- Assume you have an interval timer that generates:

  a *short time pulse (TS)* and
  a *long time pulse (TL)*,
  in response to a *set (ST)* signal.
  TS is to be used for timing yellow lights and TL for green lights.

TRAFFIC LIGHT CONTROLLER:

The farm road and highway intersection

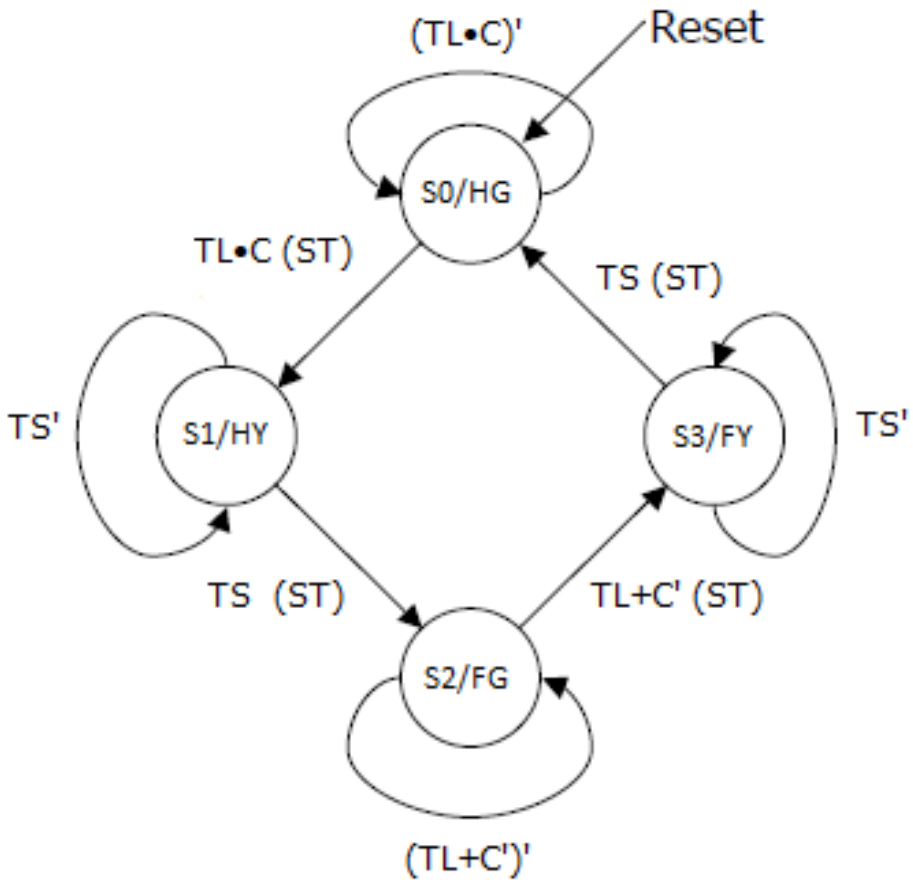


Input	Description
Reset	Place FSM in initial state
C	Detect vehicle on the farm road
TS	Short time interval expired
TL	Long time interval expired

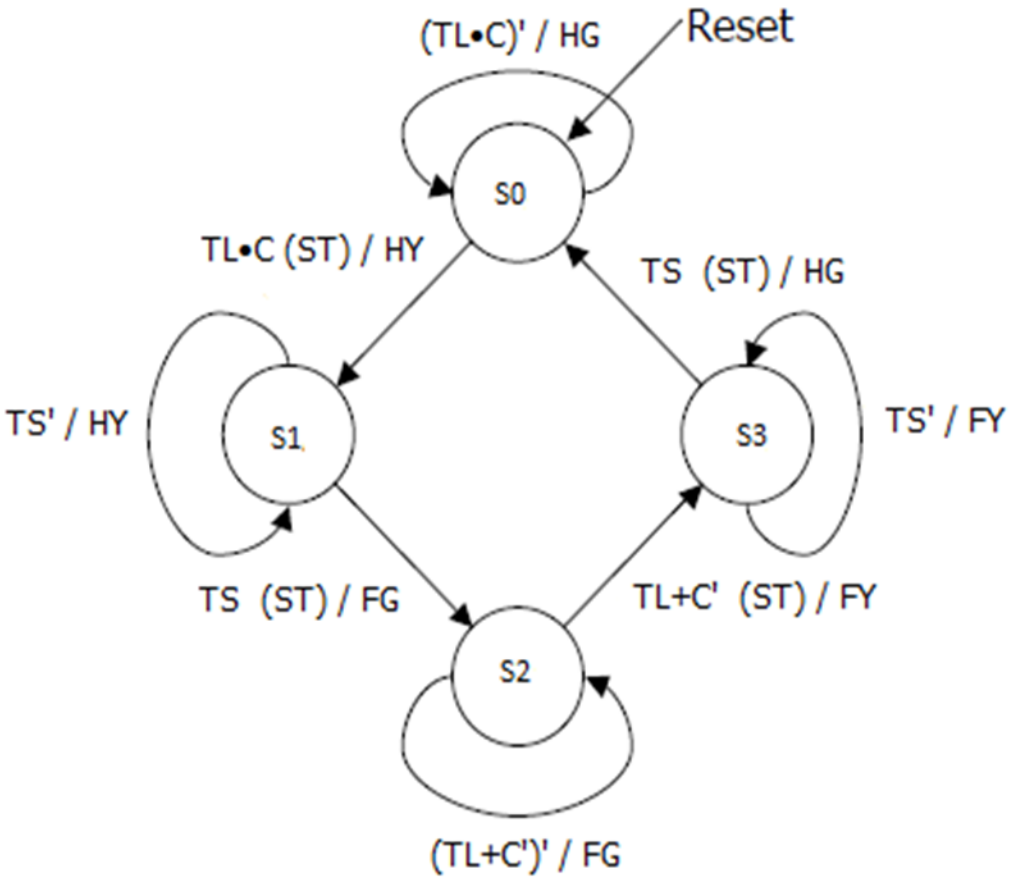
Output	Description
HG, HY, HR	Assert green/yellow/red highway lights
FG, FY, FR	Assert green/yellow/red farm road lights
ST	Start timing a long or short interval

State	Description
S0	Highway green (Farm road red)
S1	Highway yellow (Farm road red)
S2	Farm road green (Highway red)
S3	Farm road yellow (Highway red)

Note: Some light configurations imply others (mentioned in the brackets).



Moore Machine



Mealy machine

In both, the Moore and Mealy machines, the ST (set) variable mentioned in the brackets means that the timer is set and starts timing the short or long interval required for the yellow or green lights respectively.

Therefore, a simple traffic light controller can be realized using sequential logic circuits. The sensors and timers make this implementation more efficient.