# Finite state machine: DSD ID1340 report

CS18BTECH11039 : Raj Patil

2nd October 2018

# Contents

# Chapter 1

# Pre-Implementation Discussion

## 1.1 Introduction

*This reports intends to completely record the procedure by which the state machine was implemented and the variables involved.*

**Student details**
name : Raj Deepaknath Patil
roll no. : CS18BTECH11039
course : Digital system design

## 1.2 INPUT Variables

As the number of states involved in the finite state machine are 8 in number which are shown in a table that follows, we will need 3 binary variables as inputs to assign a separate binary combination to each state.

Furthermore, as there are 4 desired operations we wish to implement, namely left, right, hazard and off; we will need further 2 more select inputs which will again be just one bit each.

hence the input variables are listed as follows:-

1. A (one of 3 variables used for state definition)
2. B (one of 3 variables used for state definition)
3. C (one of 3 variables used for state definition)
4. S0 (one of the 2 select inputs)
5. S1 (one of the 2 select inputs)

***NOTE:***the PQR inputs which are listed in the below table are just the previous state values of ABC (They are the same variables.)

## 1.3 Defining the states

We will need to define 8 states as listed below in the table:-

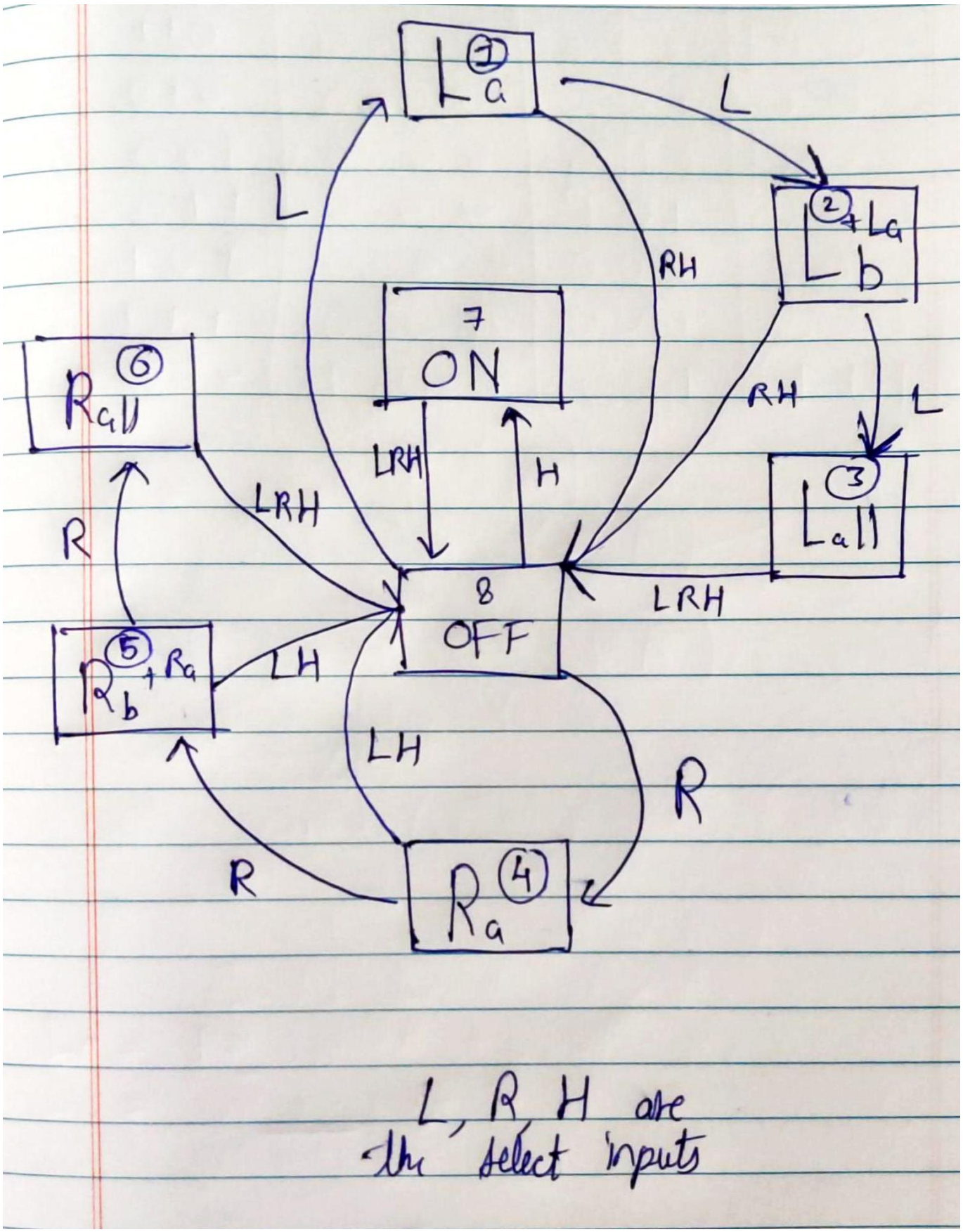
| sr no. | state description | P | Q | R |
|---|---|---|---|---|
| 1 | La lit | 1 | 0 | 0 |
| 2 | La+Lb lit | 1 | 1 | 0 |
| 3 | La+Lb+Lc lit | 0 | 1 | 0 |
| 4 | Ra lit | 0 | 0 | 1 |
| 5 | Ra+Rb lit | 0 | 1 | 1 |
| 6 | Ra+Rb+Rc lit | 1 | 0 | 1 |
| 7 | all lights lit | 1 | 1 | 1 |
| 8 | all lights off | 0 | 0 | 0 |

also assigning the names for the selection situations now for further reference:

| S1 | S0 | select variable |
|---|---|---|
| 1 | 0 | Left :- L |
| 0 | 1 | Right :- R |
| 1 | 1 | hazard :- H |

## 1.4 The state transition diagram

*Here the select inputs are the variable which will decide what transition takes place from a particular state, the initial state is the 8th one i.e. all the lights are off initially*


1
La
L
2
La
Lb
RH
L
7
ON
RH
L
6
Ra||
LRH
H
LRH
R
3
La||
8
OFF
LRH
5
Rb
Ra
LH
LH
R
R
4
Ra
L, R, H are
the select inputs

## 1.5 Hardware required

**As we have 8 states under consideration, we will need only *3* flip-flops.**
**We will be using D flip-flops.**

Other basic hardware required for the implementation of the circuit is :-

- 6 LEDs
- logic gates or a micro-controller to provide the combinational logic and the clock as well.
- resistors

# Chapter 2

# implementation and combinational logic

## 2.1 Transition table

| State no<br>–<br>– | previous state<br>–<br>PQR | if(Left)<br>S1= 1, S0=0<br>ABC | if(Right)<br>S1=0, S1=1<br>ABC | if(Hazard)<br>S1=1, S0=1<br>ABC |
|---|---|---|---|---|
| 1 | 100 | 110 | 000 | 000 |
| 2 | 110 | 010 | 000 | 000 |
| 3 | 010 | 000 | 000 | 000 |
| 4 | 001 | 000 | 011 | 000 |
| 5 | 011 | 000 | 101 | 000 |
| 6 | 101 | 000 | 000 | 000 |
| 7 | 111 | 000 | 000 | 000 |
| 8 | 000 | 100 | 001 | 111 |

**NOTE:**:- here A, B, C are just the next states of P, Q ,R respectively and are not logically new variables ,

so they will just be the outputs to the flip-flops to which PQR will inputs.

## 2.2 Transition equations

The transition equations will be as follows :-

1. A = S1!S0(!Q!R) + !S1S0(!PQR) + S1S0!P!Q!R
2. B = S1!S0(P!R) + !S1S0(!P!QR) + S1S0!P!Q!R
3. C = !S1S0(!PR + !P!Q!R) + S1S0!P!Q!R

   **NOTE**: the ! symbol signifies a negation.

## 2.3 The final implementation

Before we can produce a circuit, we need to create outputs regarding the LEDs i.e for La to Rc

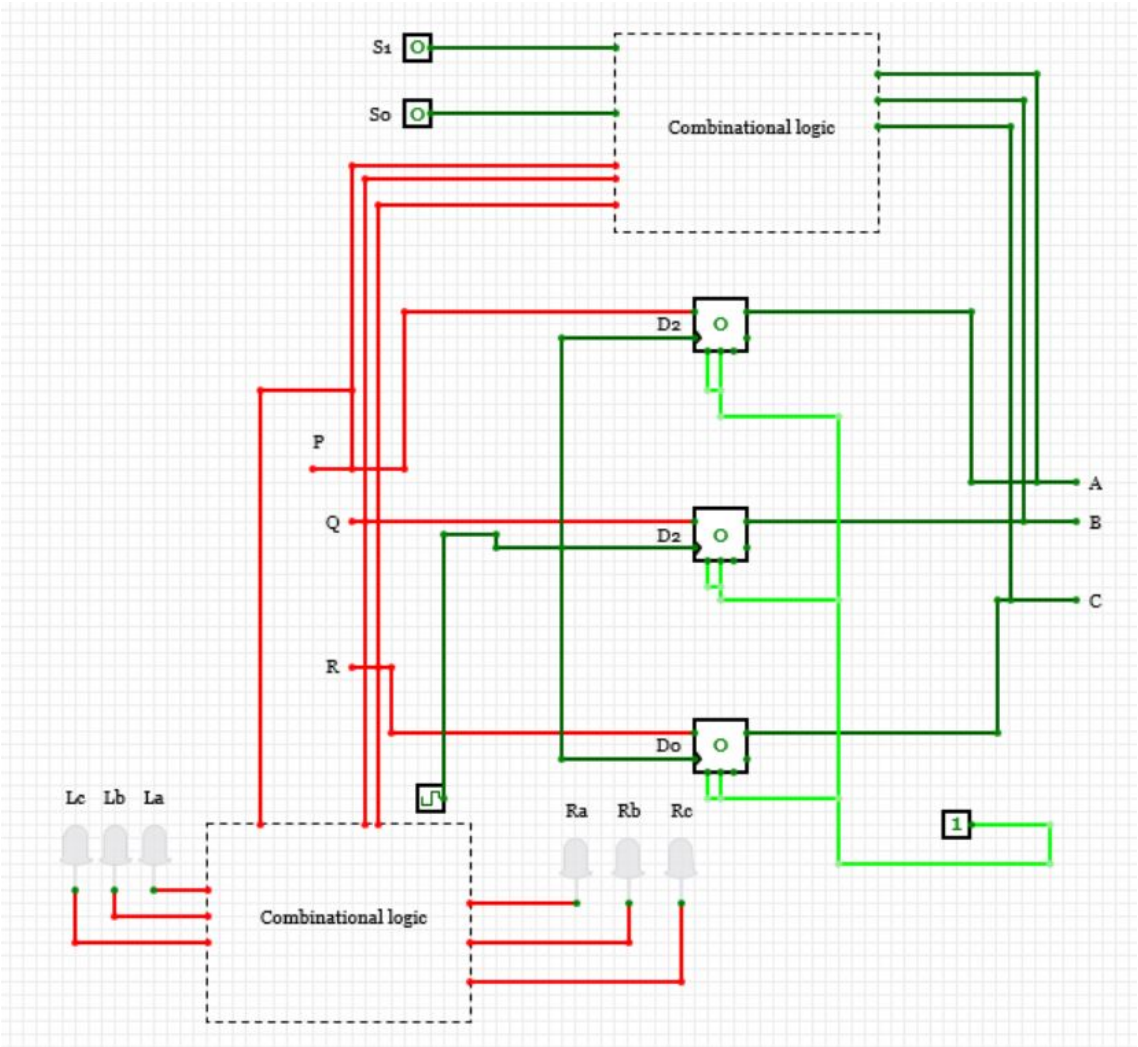
These outputs in terms of P, Q and R are listed below :-

- La = P!R + !PQ!R + PQR

- Lb = Q!R + PQR
- Lc = Q!(P!R + !PR)
- Ra = R
- Rb = R(P!Q + !PR) + PQR
- Rc = PR

These outputs will be connected to the final LEDs...
**the final implemented circuit is on next page**

it was designed using an online circuit simulator ("CircuitVerse") and is solely created by me and does not involve any sort of plagiarism.




***NOTES***:

1. The flip-flops have been set to appropriate preset and clear and enable inputs.
2. The combinational logic is implemented in the rectangles and is the same as the one provided in section 2.3 for the LEDs and 2.2 for the transitional equations.
3. The Select inputs will decide where the transition takes place from a specific state.
4. The clock and the logic was implemented using an arduino genuino uno micro-controller

## 2.4 answers to the questions

*This section provides the reference to the sections in which the required questions have been answered for making the evaluation explicitly easy.*

1. Q1) section 1.2
2. Q2) section 1.3
3. Q3) section 1.4
4. Q4) section 1.5(3 flip-flops required)
5. Q5) section 2.1
6. Q6) section 2.2
7. Q7) section 2.3