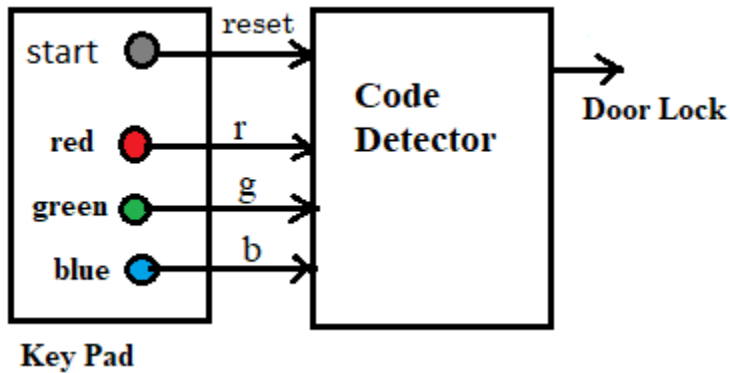


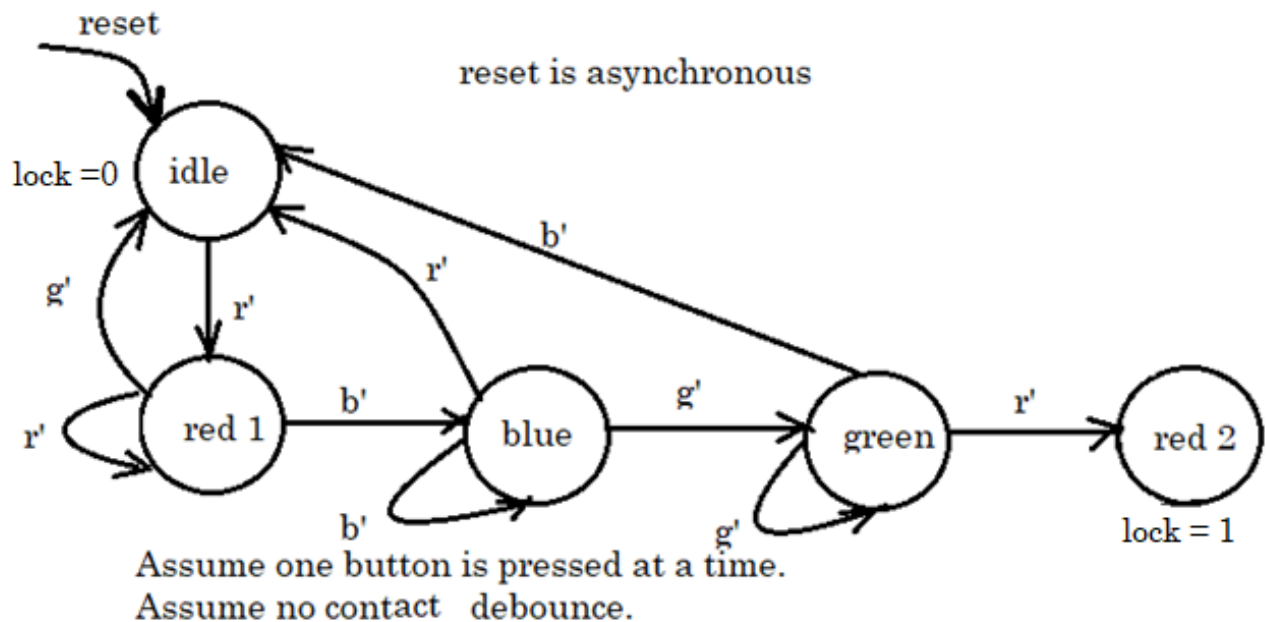
Pre-assignment Task

To prepare for this assignment enter the following code and run it on you FPGA board.

This is the VHDL model for a code detector as shown in the Figure. The keypad is used to unlock a door. Pressing the start button followed by the sequence red-blue-green-red unlocks the door, no other sequence can open the door. Assume the clock is slowed down and each pressing of a button is detected once. For example when red is pressed it is only detected as pressed for one clock cycle.



This is the state diagram for the code detector:



```

19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 entity CodeDetector is
24     Port ( clk : in  STD_LOGIC;
25           reset : in  STD_LOGIC;
26           r : in  STD_LOGIC;
27           g : in  STD_LOGIC;
28           b : in  STD_LOGIC;
29           lock : out STD_LOGIC;
30           d:  out STD_LOGIC);
31 end CodeDetector;
32
33 architecture Behavioral of CodeDetector is
34     type statetype is (idle, red1, green, blue, red2);
35     signal state: statetype;
36     begin
37         ----state register
38         process(clk, reset, r, g, b) begin
39             if (reset = '0') then state <= idle;
40             elsif rising_edge(clk) then
41                 case (state) is
42                     when idle =>
43                         if (r='0') then state <= red1;
44                         end if;
45                     when red1 =>
46                         if (b='0') then state <= blue;
47                         elsif(r='0') then state<= red1;
48                         elsif(g='0') then state <= idle;
49                         end if;
50                     when blue =>
51                         if (g='0') then state <= green;
52                         elsif(b='0') then state<= blue;
53                         elsif(r='0') then state <= idle;
54                         end if;
55                     when green =>
56                         if (r='0') then state <= red2;
57                         elsif(g='0') then state<= green;
58                         elsif(b='0') then state <= idle;
59                         end if;
60                     when red2 =>
61                         state <= red2;
62                     when others =>
63                         state <= idle;
64                 end case;
65             end if;
66         end process;
67         ----Output logic
68         lock <= '1' when state = red2 else '0'; --One LED used to show locked
69         d <= '1' when state = idle else '0';  -- one LED used to show unlocked
70
71     end Behavioral;

```

User Constraint file

```
1  ###Clock
2  NET "clk" LOC = P129 | IOSTANDARD = LVTTTL | PERIOD = 12 MHz;
3
4  #####output LED
5  NET "lock" LOC = P46 | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST;
6  NET "d" LOC = P50 | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST;
7
8
9  #####Input push buttons
10 NET "reset" LOC = P80 | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST | PULLUP;
11
12 NET "r" LOC = P79 | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST | PULLUP;
13 NET "b" LOC = P78 | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST | PULLUP;
14 NET "g" LOC = P77 | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = FAST | PULLUP;
```

Points possible: 10 points

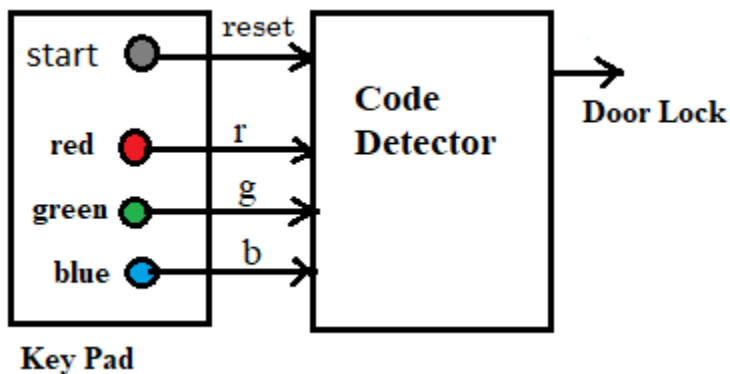
Grading Criteria:

Exercise 1 - Submit a screen shot of the **state diagram** and your **VHDL code** and also a short video of the program running on your FPGA board. The video must show student face the code on the computer and on the FPGA board. You will receive 5 points if your solution is complete and correct. Otherwise you can receive partial credit.

Exercise 2 – Scan your solution, or take pictures of your solution and submit on canvas. You will receive 5 points if your solution is complete and correct. Otherwise you can receive partial credit.

Write the following program to be executed on the FPGA board.

- Write a VHDL model for a code detector as shown in the Figure. The keypad is used to unlock a door. Pressing the start button followed by the sequence red-green-red-blue unlocks the door, no other sequence can open the door. Assume the clock is slowed down and each pressing of a button is detected once. For example when red is pressed it is only detected as pressed for one clock cycle.



- Four lights are connected to a decoder. Design a controller that will blink the lights in the following order: 0, 2, 1, 3, 0, 2, the controller output is the binary code for the light number. For example for turning on light 0, $s1 = 0$, $s0 = 0$. To turn on light 2, $s1 = 1$, $s0 = 0$. Start from a state diagram, draw the state table, minimize the logic, and draw final logic circuit at gate level.

