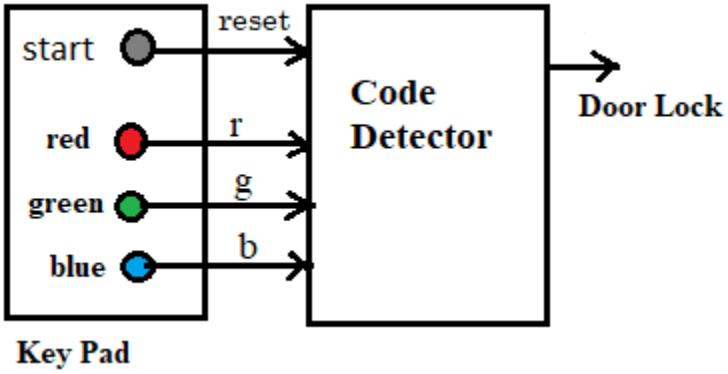
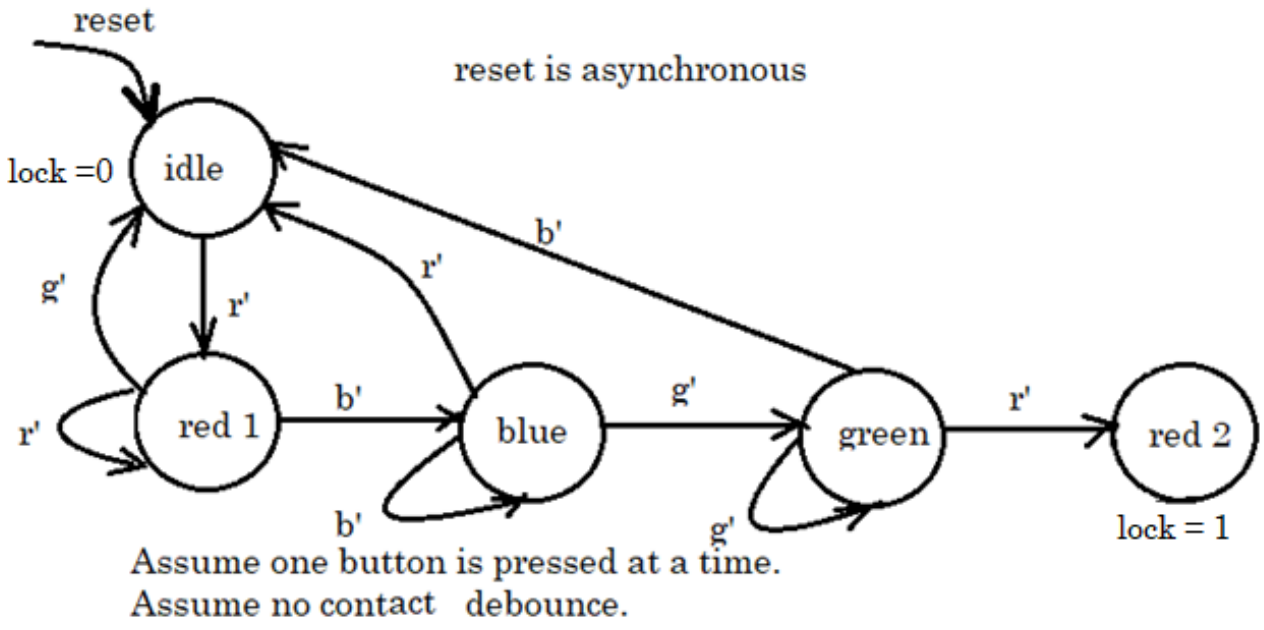
Pre-assignment Task

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:



```
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity CodeDetector is
    Port ( clk : in  STD_LOGIC;
           reset : in  STD_LOGIC;
           r : in  STD_LOGIC;
           g : in  STD_LOGIC;
           b : in  STD_LOGIC;
           lock : out  STD_LOGIC;
           d:  out STD_LOGIC);
end CodeDetector;

architecture Behavioral of CodeDetector is
type statetype is (idle, red1, green, blue, red2);
signal state: statetype;
begin
----state register
   process(clk, reset, r, g, b) begin
   if (reset ='0') then state <= idle;
   elsif rising_edge(clk) then
   case (state) is
      when idle =>
         if (r='0') then state <=red1;
         end if;
      when red1 =>
         if (b='0') then state <= blue;
         elsif(r='0') then state<= red1;
         elsif(g='0') then state <= idle;
         end if;
      when blue =>
         if (g='0') then state <= green;
         elsif(b='0') then state<= blue;
         elsif(r='0') then state <= idle;
         end if;
      when green =>
         if (r='0') then state <= red2;
         elsif(g='0') then state<= green;
         elsif(b='0') then state <= idle;
         end if;
      when red2 =>
         state <= red2;
      when others =>
         state <= idle;
   end case;
   end if;
   end process;
----Output logic
   lock <= '1' when state = red2 else '0'; --One LED used to show locked
   d <= '1' when state = idle else '0';  -- one LED used to show unlocked

end Behavioral;
```

User Constraint file

```
###Clock
   NET "clk"   LOC = P129 | IOSTANDARD = LVTTL | PERIOD = 12 MHz;

###########output LED
   NET "lock"  LOC = P46 | IOSTANDARD = LVTTL | DRIVE = 8  | SLEW = FAST;
   NET "d"     LOC = P50 | IOSTANDARD = LVTTL | DRIVE = 8  | SLEW = FAST;


############Input push buttons
   NET "reset" LOC = P80 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;

   NET "r"  LOC = P79 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;
   NET "b"  LOC = P78 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;
   NET "g"  LOC = P77 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;
```

EEL 4740 – Embedded Computing Systems **Assignment 4**

Points possible: 8 points

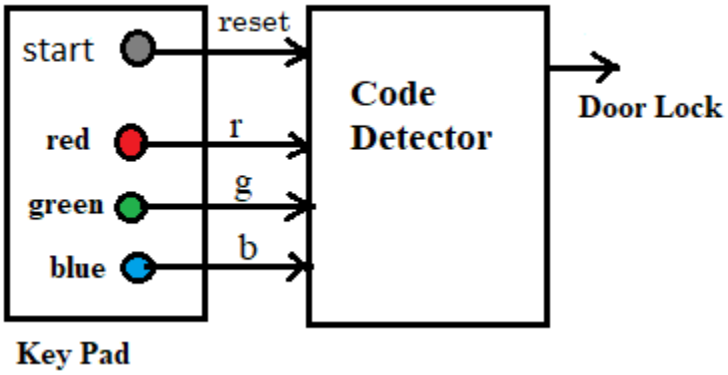
Grading Criteria:

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 3 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.

1. 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.



1. 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.