# CE 1911

# Homework 8

1 – Create the VHDL for the following state transition tables using enumerated types (rstb=0 puts the FSM into the red state) – code + RTL 50pts

| State | Din | Next State |
|---|---|---|
| red | 0 | red |
| red | 1 | green |
| green | 0 | yellow |
| green | 1 | green |
| yellow | 0 | red |
| yellow | 1 | orange |
| orange | 0 | blue |
| orange | 1 | green |
| blue | 0 | red |
| blue | 1 | Yellow |

| State | Detect |
|---|---|
| red | 0 |
| green | 0 |
| yellow | 1 |
| orange | 0 |
| blue | 1 |

```
----------------------------------------
--
-- color_fsm.vhdl
--
-- created 4/9/18
-- tj
--
-- rev 0
----------------------------------------
--
-- color FSM HW
----------------------------------------
--
-- Inputs: rstb, clk, din
-- Outputs: detect
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity color_fsm is
   port (
         i_clk          :  in std_logic;
         i_rstb         :  in std_logic;
         i_din          :  in std_logic;

         o_detect       :  out std_logic
    );
end entity;
```

2 – Use the following testbench to simulate your design – provide a plot with the state signal added to the waveforms – 1000 ns duration 50pts

```
----------------------------------------
--
-- color_fsm_tb.vhdl
--
-- created 4/9/18
-- tj
--
-- rev 0
----------------------------------------
--
-- testbench for color FSM HW
-- color_fsm.vhdl
----------------------------------------
--
-- Brute force
--
----------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity color_fsm_tb is
   -- testbench - no entity
end entity;

architecture behavioral of color_fsm_tb is
   signal CLK:    std_logic;
   signal RSTB:   std_logic;
   signal DIN:    std_logic;
   signal DETECT: std_logic;

   constant PER:  time  := 50 ns;

   --
   -- Component Prototype
   --
   component color_fsm is
      port (
            i_clk          :  in std_logic;
            i_rstb         :  in std_logic;
            i_din          :  in std_logic;

            o_detect       :  out std_logic
      );
   end component;
```

```
begin
   --
   -- device under test
   --
   dut: color_fsm
   port map(
            i_clk    => CLK,
            i_rstb   => RSTB,
            i_din    => DIN,
            o_detect => DETECT
   );

   ----------------------------------------
   -- Test processes
   ----------------------------------------

   -- Clock process
   clock: process    -- note - no sensitivity list
      begin
         CLK <= '0';
         wait for PER/2;
         infinite: loop
            CLK <= not CLK; wait for PER/2;
         end loop;
   end process clock;

   -- Reset process
   reset: process    -- note - no sensitivity list
   begin
      RSTB <= '0'; wait for 2.5*PER;
      RSTB <= '1'; wait;
   end process reset;
```

```
   -- Run process
   run: process      -- note - no sensitivity list
      begin
         ------------------------
         -- Initialize  input
         DIN <= '0';

         -- wait for reset
         wait for 3*PER;

         -- variations on input
         DIN <= '0';       -- to red
         wait for 3*PER;
         DIN <= '1';       -- to green
         wait for 5*PER;
         DIN <= '0';       -- to yellow
         wait for 1*PER;
         DIN <= '1';
         wait for 1*PER;   -- to orange
         DIN <= '0';
         wait for 1*PER;   -- to blue
         DIN <= '1';
         wait for 1*PER;   -- to yellow
      end process run;

   ----------------------------------------
   -- End test processes
   ----------------------------------------
end architecture;
```