Introduction to Digital Logic
EECS/CSE 31L

Homework 3
Finite State Machine

course instructor: Pooria M.Yaghini
Henry Samueli School of Engineering
University of California, Irvine

Nov 22, 2014

Due on Saturday 12/01/2014 , 11:30 pm

1 Designing a state machine (points 50)

This question evaluates your ability to write a FSM in VHDL. Based on following entity and Figure1 write the ARCHITECTURE part for this mealy FSM.

```
--------------------------------
library ieee;
use IEEE.std_logic_1164.all;
--------------------------------
ENTITY mealy is
port (clk : IN std_logic;
      reset : IN std_logic;
      input : IN std_logic;
      output : OUT std_logic
  );
END mealy;
--------------------------------
```

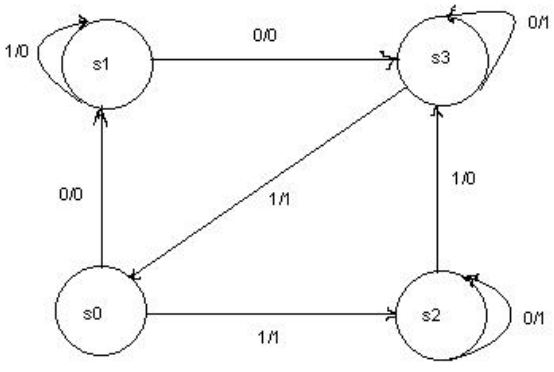

Figure 1: FSM model

2 Traffic light (points 50)

In this question, we complete the traffic light controller example at the end of lecture 6. Please answer the following questions.

I) How many modes of operation are there?

II) Fill the blanks (A, B, C, D, E and F).

III) Which states(RG, RY ,GR, YR, YY) belong to each mode of operation?

```
------------------------------
library ieee;
use IEEE.std_logic_1164.all;
------------------------------
ENTITY tlc IS
 (???A???)(
  timeRG: POSITIVE := 1800;
  timeRY: POSTIVE := 300;
  timeGR: POSITIVE := 2700;
  timeYR: POSITIVE := 300;
  timeTEST: POSITIVE := 60;
  timeMAX: POSITIVE := 2700);
 PORT (
  clk,stby ,test,: IN STD_LOGIC;
  r1,r2,y1,y2,g1,g2: OUT STD_LOGIC);
END tlc;
------------------------------
ARCHITECTURE fsm of tlc IS
 TYPE state IS ( RG, RY ,GR, YR, YY):
 SIGNAL pr_state, nx_state: state;
 (???B???) timer: INTEGER RANGE 0 TO timeMAX;
BEGIN
 ---Lower section of FSM---
 PROCESS ( clk, stby)
  VARIABLE count : INTEGER RANGE 0 to timeMAX;
 BEGIN
  IF (stby='1') THEN
   pr_state <= (???C???);
   count :=0;
  ELSIF (clk'EVENT AND clk ='1') THEN
   count := count +1;
   IF ( count >= timer) THEN
    pr_state <= nx_state;
    count := 0;
   END IF;
  END IF;
 END PROCESS:
 ---Upper section of FSM---
 PROCESS ((???D???))
 BEGIN
  CASE pr_state IS
   WHEN RG =>
    r1<='1'; y1<='0'; g1<'0';
    r2<='0'; y2<='0'; g2<='1';
    nx_state <= RY;
    IF (test='0') THEN
     timer <= timeRG;
    ELSE
     timer <= timeTEST;
    END IF;
   WHEN RY =>
    r1<='1'; y1<='0'; g1<'0';
    r2<='0'; y2<='1'; g2<='0';
    nx_state <= GR;
    IF (test ='0') THEN
     timer <= timeRY;
    ELSE
     timer <= timeTEST;
    END IF;
   WHEN GR =>
    r1<='0'; y1<='0'; g1<'1';
    r2<='1'; y2<='0'; g2<='0';
    nx_state <= TR;
    IF ( test = '0') THEN
     timer <= timeGR;
    ELSE
     timer <= timerTEST;
    END IF;
   WHEN YR =>
    r1<='0'; y1<='1'; g1<'0';
```

```
    (???E???)
    nx_state <= RG;
    IF (test ='0') THEN
     timer <= timeYR;
    ELSE
     timer <= timeTEST;
    END IF;
   WHEN YY =>
    r1<='0'; y1<='1'; g1<'0';
    r2<='0'; y2<='1'; g2<='0';
    timer <= timeTEST;
    nx_state <= RY;
  (???F???);
 END PROCESS;
END fem;
--------------------------------
```