Sistemes Digitals
Control. 10 d'abril de 2014

Temps per a la resolució: 2 hores.

1 Control d'un semàfor (70%)

Imagineu que heu de controlar un semàfor sense tenir en compte els altres semàfors que hi ha a la ciutat ni altres factors que en facin canviar el funcionament. En aquest cas, dissenyareu un sistema que activarà el color verd durant 30 segons, seguidament s'activarà el color groc durant 5 segons i després el vermell durant 40 segons. Aquest cicle es va repetint indefinidament sense cap interrupció.
El sistema tindrà com a entrada un senyal de rellotge `clk` de 200 Hz. I com a sortida: un senyal `verd`, un senyal `groc` i un senyal `vermell` que valdran '1' quan s'hagi d'encendre cada color correponent i '0' en el cas contrari.

Per resoldre el problema respongueu les següents preguntes:

1. Dibuixeu un diagrama de blocs corresponent al *sistema complet*.
2. Escriviu l'`entity` i l'`architecture` del sistema. Atenció: Definiu acuradament cadascun dels senyals interns del vostre disseny; Feu un disseny sintetitzable completament síncron;
3. Dibuixeu un cronograma(incloent els senyals interns) que verifiqui el correcte funcionament del vostre sistema.

2 Qüestions 30%

1. Definiu breument què són els temps de Setup i de Hold d'un sistema digital?

2. Feu les següents operacions treballant amb vectors binaris amb signe i d'amplada de 7 bits. Indiqueu, també, si hi ha hagut Overflow i/o Carry de sortida:
   a) -50 - 20
   b) 62 - 25

3. Tenim un rellotge de 10 MHz d'entrada i volem crear un divisor de freqüència per generar un rellotge de 1 MHz.
   a) És correcte el codi següent tenint en compte el que volem aconseguir? Si és incorrecte corregiu-lo. (Pista: Penseu en si volem simular aquest codi.)
   b) Dibuixeu el cronograma amb els senyals `clk`, `count`, `clk_aux`, `clk_out`.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity divisor is
port (  clk     : in std_logic;
        clk_out : out std_logic);
end divisor;

architecture behav of divisor is
  signal count   : unsigned(3 downto 0);
  signal clk_aux : std_logic;
 begin
  process(clk)
  begin
    if rising_edge(clk) then
      if count = 10 then
        clk_aux <= not(clk_aux);
        count <= (others => '0');
      else
        count <= count + 1;
      end if;
    end if;
  end process;

  clk_out <= clk_aux;

end behav;
```

4. Expliqueu breument la diferència entre una màquina d'estats tipus Moore i una de tipus Mealy.