

Sistemes Digitals

Examen Final. 17 de juny del 2014

Temps per a la resolució: 2 hores.

1 Problema: Disseny d'un freqüencímetre (50%)

Volem dissenyar un freqüencímetre digital per mesurar la freqüència d'un senyal d'entrada. Per simplificar el problema, aquest freqüencímetre només serà capaç de detectar tres freqüències diferents: 250kHz, 500kHz i 750kHz.

El disseny tindrà com a entrades un rellotge, *clk*, de 100 MHz i el senyal *rf_input* del que es vol mesurar la freqüència. Com a sortides tindrà *banda250k*, *banda500k* i *banda750k* que s'activaran al mesurar cada freqüència corresponent.

La tècnica que utilitzarem per dissenyar aquest freqüencímetre consisteix en mesurar quants cicles del senyal d'entrada (per ex. flancs de pujada) es compten en un interval de temps. Aquest interval serà de 16 μs . Per prendre la decisió de la freqüència mesurada, es tindrà en compte un error de ± 1 cicle. Per exemple, si per mesurar una certa freqüència es comptarien 8 cicles idealment, si comptem 7 o 9 cicles decidirem que la freqüència mesurada és la mateixa.

Per resoldre el problema respongueu les següents preguntes:

1. Calculeu quants cicles es comptaran en l'interval de temps definit en cada cas: 250kHz, 500kHz i 750kHz.
2. Dibuixeu un diagrama de blocs *detallat* corresponent al sistema complet.
3. Escriviu l'*entity* i l'*architecture* del sistema. Atenció: Feu un disseny completament síncron.
4. Dibuixeu un cronograma detallat, amb els senyals interns importants, que verifiqui el correcte funcionament del vostre disseny.

2 Test (50%)

(Errors en preguntes tipus test: -0.1 punts)

1. Quina de les següents afirmacions és correcta respecte a un *process*?
 - a) Sempre descriu un circuit seqüencial
 - b) Sempre descriu un circuit combinacional
 - c) No pot incloure primitives *case*
 - d) No pot incloure primitives *with*
2. La síntesi del procés següent (encercleu l'afirmació correcta):

```
process(clk)
begin
  if rising_edge(clk) then
    rA <= A;
    rB <= B;
    rC <= C;
    suma <= rA + rB + rC;
  end if;
end process;
```

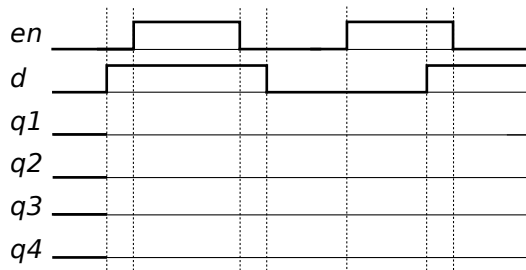
- a) Generarà quatre registres
 - b) Generarà tres registres
 - c) Generarà un registre
 - d) No generarà cap registre
3. Quan sintetitzem el codi següent amb el Quartus II, apareixerà algun *warning*? En cas afirmatiu, quin/s? i per quin motiu apareix/en?
(Suposeu que tots els senyals estan declarats correctament.)

```

process(b)
begin
  case b is
    when '1' => sortida <= entrada;
    when others => null;
  end case;
end process;

```

4. Completeu el cronograma tenint en compte els diferents processos que es descriuen a continuació:



```

fig1: process(d,en)
begin
  if en='1' then
    q1<=d;
  end if;
end process fig1;

fig2: process(d)
begin
  if en='1' then
    q2<=d;
  end if;
end process fig2;

fig3: process(en)
begin
  if en='1' then
    q3<=d;
  end if;
end process fig3;

fig4: process(en)
begin
  q4<=d;
end process fig4;

```

5. Sobre el càlcul del comptador de programa en el nostre mini AVR, quina de les següents afirmacions és correcta?

```
tmp_pc := std_logic_vector(signed(pr_pc & '1') +
    signed(k_jump & '1'));
nx_pc <= tmp_pc(8 downto 1);
```

- a) El '1' que concatenem a cada operant serveix per propagar el carry de l'operació anterior.
 - b) El '1' que concatenem a cada operant serveix per incrementar sempre una unitat el comptador de programa.
 - c) El '1' que concatenem a cada operador serveix per igualar les longituds dels vectors.
 - d) El '1' que concatenem a cada operador serveix perquè la suma es faci correctament ja que treballem amb complement a dos.
6. Quina de les afirmacions següents és correcta sobre el banc de registres implementat en el mini AVR presentat durant el curs?
- a) La lectura del valor de qualsevol registre és síncrona.
 - b) L'escriptura d'un valor a qualsevol registre és síncrona.
 - c) Hi ha disponibles 32 registres per llegir-hi i escriure-hi.
 - d) L'amplada de cada registre és de 16 bits.
7. Quina diferència funcional tenen les instruccions RJMP i el BREQ?
8. Quina de les següents afirmacions és falsa?
- a) Per saber l'estat del timer, utilitzem la instrucció IN per llegir el valor del timer_state.
 - b) El límit del prescaler i el límit del timer es modifiquen a través de la instrucció OUT.
 - c) El timer representa un component extern del nucli del mini AVR.
 - d) Quan el timer està comptant i arriba al límit definit prèviament, per defecte, quedarà aturat.
9. El següent codi correspon a l'opció 1 presentada a classe per implementar la RAM al mini AVR. Perquè no és bona aquesta opció?

```
RAM_process : process(clk)
begin
    if rising_edge(clk) then
        if mem_we = '1' then
            RAM(to_integer(unsigned(mem_adr(9 downto 0)))) <=
                regs(to_integer(unsigned(r_reg)));
        end if;
        ram_q <= RAM(to_integer(unsigned(mem_adr(9 downto 0))));
    end if;
end process;
```