

Sistemes Digitals

Examen Final. 28 de juny del 2013

Temps per a la resolució: 2 hores. Publicació de qualificacions: 1 de Juliol del 2013.

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

Utilitzant la freqüència de rellotge principal(50MHz) de la placa de la FPGA que utilitzem al laboratori, volem dissenyar un PWM de 25kHz amb un cicle de treball variable que podrà ser d'un 25%, un 50% o un 75%. Aquest cicle de treball serà escollit amb un sol botó de la placa. Cada cop que es premi el botó, el cicle de treball s'haurà d'incrementar un 25%. Quan s'arribi al 75% de cicle de treball i es torni a premer el botó, es tornarà a un 25%.

Per resoldre el problema respongueu les següents preguntes:

1. Dibuixeu un diagrama de blocs *detallat* corresponent al sistema complet.
2. Escriviu l'**entity** i l'**architecture** del sistema(podeu fer-ho tot en un sol bloc o separar-ho). Atenció: Feu un disseny completament síncron.
3. 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: -0.2 punts)

1. Una vegada sintetitzat el següent procés(encercleu l'afirmació que creieu correcta):

```
process(a,b,c,d)
begin
  if (a = '1') then
    if (b = '1' and c = '1' and d = '1') then
      o <= '1';
    else
      o <= '0';
    end if;
  end if;
end process;
```

- a) El senyal o serà la funció **and** dels senyals a,b,c,d
 - b) El senyal o quedarà a un valor indefinit quan a='0'
 - c) El senyal o es generarà a partir d'un latch amb el senyal a com a senyal de rellotge
 - d) El senyal o es generarà a partir d'un flip-flop amb els senyals b,c i d com a senyals de rellotge
2. Els flip-flops que es sintetitzin en un procés que tingui (clk,rst) a la llista de sensibilitat tindran reset síncron o asíncron? Justifiqueu la resposta.

3. Una vegada sintetitzat el següent procés(encerclau l'afirmació que creieu correcta):

```
process(a,b,c,d,sel)
begin
  q <= a;
  if (sel(0) = '0') then
    q <= b;
  end if;
  if (sel(0) = '1') then
    q <= c;
  end if;
  if (sel(1) = '0') then
    q <= d;
  end if;
end process;
```

- a) El valor per defecte del senyal q és a
- b) El senyal sel actuarà com a rellotge del senyal q
- c) L'assignació amb més prioritats és la corresponent a sel(1)='0'
- d) Es generarà un latch controlat pel senyal sel

4. 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

5. Completa la següent frase:

El comptador de programa apunta a una adreça de, on hi ha guardats els

6. 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.

7. En el miniAVR s'ha definit una instrucció, la qual té la següent sintaxi: LD Rd, x. On la x és:

- a) El valor a carregar en el registre Rd.
- b) L'adreça del registre on s'agafarà el valor per carregar-lo a Rd.
- c) L'adreça del port extern on s'agafarà el valor per carregar-lo a Rd.

- d) L'adreça de la memòria RAM on s'agafarà el valor per carregar-lo a Rd.
8. El prescaler del Timer serveix per:
- Dividir el rellotge principal i tenir una unitat més lenta per poder comptar.
 - Definir el valor màxim fins el qual comptarà el Timer.
 - Saber si el Timer està comptant o ha acabat.
 - Definir el número de port on estarà connectat el Timer.
9. Quina de les següents afirmacions és correcta?
- En el nostre miniAVR, l'ALU només realitza les operacions de AND, OR, EOR i ADC.
 - S'utilitza el salt relatiu BRNE sempre per fer bucles en un programa.
 - La instrucció NOP només serveix per perdre el primer cicle de rellotge ja que normalment allà hi fem un reset.
 - La instrucció RJMP -1 la fem servir per deixar el programa aturat en aquella instrucció.
10. Escriu els passos concrets per fer un bucle en un programa que duri fins que el Timer acabi de comptar el temps que es vol.