

Sistemes Digitals

Examen Final. 10 de juny del 2016

Temps per a la resolució: 2 hores.

1 Problema: Detector d'una seqüència en l'ordre de pulsació (50%)

Es vol dissenyar un sistema en VHDL que sigui capaç de detectar una seqüència predefinida en l'ordre en que es premen tres pulsadors. Com a entrades en el sistema tindrem els tres pulsadors anomenats: $B1$, $B2$ i $B3$, l'entrada clk que serà el rellotge d'1 kHz i una entrada de reset: $reset$. Com a sortides tindrem:

- Un vector anomenat *intents* que mostrarà les pulsacions que s'han fet fins al moment.
- Un senyal *inici* que indicarà a l'usuari que està al punt inicial de la seqüència.
- Un senyal *encertada* que s'activarà quan s'hagi encertat la seqüència bona.

Per simplificar el problema es tindrà en compte que:

- La seqüència patró sempre serà la mateixa: $B2 - B1 - B3$.
- Cada cop que es prem un dels pulsadors, s'ha d'incrementar la sortida *intents*. Com a màxim voldrem comptar 15 intents.
- Si polsem un botó erroni sempre tornarem a l'inici. Això vol dir que, per exemple, si hem premut $B2 - B1 - B2$, tornarem a l'inici i haurem de tornar a prémer $B2$ per iniciar la seqüència correcta.

El senyal *reset* posarà a 0 la sortida *intents* i ens farà tornar a l'inici de la seqüència.

Per resoldre el problema respongueu les següents preguntes:

1. Dibuixeu un diagrama de blocs *detallat* corresponent al sistema complet. Atenció: Feu un disseny completament síncron. (1 punt)
2. Escriviu l'*entity* i l'*architecture* del sistema. (3 punts)
3. Dibuixeu un cronograma detallat, amb els senyals interns importants, que verifiqui el correcte funcionament del vostre disseny. (1 punt)

2 Qüestions (50%)

Cada pregunta encertada suma 0.5 punts, excepte una que suma 1 punt (està indicada). Els errors en preguntes tipus test resten 1/6.

1. Què és el temps de setup d'un flip-flop?
 - a) El temps que ha d'estar estable un senyal a l'entrada després del flanc de rellotge.
 - b) El temps que tarda l'entrada en propagar-se cap a la sortida.
 - c) El temps que ha d'estar estable un senyal a l'entrada abans del flanc de rellotge.
 - d) El temps que tarda la sortida del flip-flop a estabilitzar-se després del flanc de rellotge.
2. Quina afirmació és correcta sobre l'arquitectura Harvard?:
 - a) La memòria de dades i la memòria de programa comparteixen bus.
 - b) La memòria de dades i la memòria de programa no comparteixen bus.
 - c) L'amplada de les dades ha de ser igual a l'amplada de les instruccions.
 - d) No es pot accedir al mateix temps a la memòria de programa i a la memòria de les dades.
3. Observeu les dues descripcions en VHDL que hi ha a continuació. Tenint en compte que els senyals *a* i *b* només poden valdre '0' o '1' (són de tipus bit) encercleu l'afirmació correcta:

```
circuit1: process(a,b) | -- circuit2:
begin |
  if a = '1' and b = '1' then | s2 <= a and b;
    s1 <= '1'; |
  else |
    s1 <= '0'; |
  end if; |
end process circuit1; |
```

- a) Es generarà un latch en el cas de *s1* ja que l'assignació està dins d'un process.
 - b) Es generarà un latch en el cas de *s2* ja que falta l'**else**.
 - c) Es generaran dos circuits diferents ja que *s1* té una assignació seqüencial.
 - d) Es generarà el mateix circuit combinacional per *s1* que per *s2*.
4. Els flip-flops que es sintetitzin en un process que tingui (*clk*, *reset*) a la llista de sensibilitat tindran reset síncron o asíncron? Justifiqueu la resposta.

5. Quina afirmació és correcta sobre el senyal *k_jump*?
 - a) És el valor que es sumarà (o restarà) a la posició actual de la ROM.
 - b) És la posició absoluta de la ROM a on es vol saltar.
 - c) És una constant que serveix per definir la instrucció RJMP.
 - d) És un senyal de 16 bits.

6. Quina de les afirmacions següents és correcta sobre la línia següent?

```
nx_pc <= std_logic_vector(unsigned(pr_pc) + 1);
```

- a) S'ha d'eliminar ja que aquesta assignació és concurrent i quan ampliem l'AVR ho fem seqüencial.
 - b) S'ha de mantenir ja que en cap moment voldrem fer salts de més d'una posició en el `pr_pc`.
 - c) S'ha d'eliminar quan ampliem l'AVR per afegir les instruccions dels salts; utilitzem un altre codi pel càlcul del comptador de programa.
 - d) S'ha de mantenir en totes les ampliacions de l'AVR ja que volem seguir sumant +1 en el comptador de programa fins i tot en els casos que vulguem saltar posicions.
7. Quina de les afirmacions següents és correcta sobre el bloc de control?
- a) L'entrada és l'opcode de 8 bits.
 - b) L'entrada és el `pr_pc`.
 - c) És un bloc síncron.
 - d) És un bloc combinacional.
8. Quina de les següents afirmacions és falsa?
- a) La instrucció MOV copia el valor d'un registre a un altre.
 - b) La instrucció OUT treu un valor contingut a l'opcode a un port de sortida.
 - c) La instrucció LD Rd, X carrega el valor contingut a la posició X de la RAM al registre Rd.
 - d) La instrucció LDI carrega un valor contingut a l'opcode a un registre.
9. Escriu els passos necessaris per engegar el Timer i fer que el programa de la ROM es quedi esperant fins que el Timer acabi de comptar. (1 punt)