

# Sistemes Digitals

## Examen Final. 13 de juny del 2023

Temps per a la resolució: 2 hores.

### 1 Problema: Dispositiu làser (50%)

Des d'una empresa que es dedica a fabricar radars i dispositius làsers per detectar la velocitat de vehicles, se'ns demana que dissenyem en VHDL un dispositiu làser. El seu funcionament és molt simple, aquests dispositius consten de dos feixos làsers separats 40 cm entre ells. Quan un vehicle creua el primer feix, es comença a cronometrar el temps que tarda fins a creuar el segon feix. D'aquesta forma es sap quant de temps ha tardat en fer 40 cm i, per tant, la velocitat. Se'ns demana poder mesurar velocitats entre 3 km/h i 300 km/h.

Com a entrades del sistema tindrem els dos feixos làser: *laser1* i *laser2* i el senyal de rellotge *clk* que serà de 100 MHz. Les entrades corresponents als feixos làser seran de tipus `std_logic` i estaran sempre a '1' i només es posaran a '0' en el moment que passa un vehicle. Com a sortides tindrem un vector anomenat *temps* que mostrarà el temps, en microsegons, que ha tardat en fer els 40 cm. Aquest senyal de sortida només s'ha d'actualitzar un cop s'ha creuat el segon feix làser.

Per simplificar el problema es tindrà en compte que:

- No passaran mai dos vehicles alhora.
- Els vehicles sempre creuaran els dos feixos làser.
- Un vehicle no anirà a menys de 3 km/h ni a més de 300 km/h.
- No hi haurà altres objectes que passin per davant dels feixos làser.

Per resoldre el problema respongueu les següents preguntes:

1. Calculeu quant de temps tarda un vehicle a fer 40 cm a 3 km/h i a 300 km/h. Tanmateix, tenint en compte que volem tenir una resolució d'1  $\mu\text{s}$ , dimensioneu el cronòmetre correctament. (0,5 punts)
2. Dibuixeu un diagrama de blocs *detallat* corresponent al sistema complet. Atenció: Feu un disseny completament síncron. (0,5 punts)
3. Escriviu l'*entity* i l'*architecture* del sistema. (3 punts)
4. Dibuixeu un cronograma detallat, amb els senyals interns importants, que verifiqui el correcte funcionament del vostre disseny. (1 punt)

## 2 Qüestions (50%)

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

1. Quina de les següents afirmacions és certa respecte l'execució del següent codi ensamblador:

```
LDI r16, xAA
LDI r17, x55
AND r17, r16
BREQ +1
RJMP -3
...
```

- a) Es realitza un bucle infinit.
- b) La instrucció RJMP -3 no s'executa.
- c) El contingut del registre r16 acaba siguent 0.
- d) El contingut del registre r17 acaba siguent xFF.

2. Indiqueu quina de les següents afirmacions és certa respecte el codi VHDL de sota:

```
architecture behav of cronos is
    signal flag1, flag2 : std_logic := '1';
    signal Q : unsigned(1 downto 0) := "00";

    process (clk) is
    begin
        if rising_edge(clk) then
            Q <= "00";
            Q <= Q + 1;
            flag1 <= '1';
            if Q = 4 then
                Q <= "00";
                flag1 <= '0';
            end if;
        end if;
    end process;
    flag2 <= '1' when (Q > "10") else '0';
end behav;
```

- a) flag2 valdrà sempre '0'
- b) Sempre que flag2 està actiu '1', flag1 està a '0'
- c) El comptador Q es reinicia cada 5 cicles de clk
- d) flag1 valdrà sempre '1'

3. Quina afirmació és correcta sobre la memòria RAM?

- a) Per adreçar-la cal concatenar dos registres.
- b) El valor a carregar ve donat per la concatenació de dos registres.
- c) La sortida va connectada cap a un port de sortida.
- d) Per adreçar-la cal concatenar el d\_reg i el r\_reg.

4. Indiqueu quina de les següents afirmacions és falsa respecte el Mini AVR:

- a) La unitat de control s'encarrega de descodificar les instruccions i activar els senyals de control cap a la resta de mòduls.
  - b) El registre d'estat s'utilitza per emmagatzemar l'estat de l'última operació de la ALU.
  - c) L'escriptura als registres és síncrona.
  - d) La unitat de control conté les instruccions que ha d'executar el Mini AVR.
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 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.
7. Indiqueu quina de les següents afirmacions és certa respecte les instruccions del Mini AVR:
- a) La instrucció LDI permet carregar un valor `k` a una adreça de la memòria RAM.
  - b) La instrucció LD permet carregar el contingut d'una adreça de la memòria RAM a un registre.
  - c) La instrucció OUT permet carregar un valor `k` a una adreça dels ports d'entrada / sortida.
  - d) La instrucció IN permet carregar el contingut d'un registre a una adreça de la memòria RAM.
8. 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 quan volem ampliar l'AVR, per permetre salts condicionals.
  - b) S'ha d'eliminar quan volem ampliar l'AVR, per permetre tot tipus de salts.
  - c) 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.
  - d) S'ha de mantenir ja que és l'única forma d'incrementar el `pr_pc`.
9. Què cal fer per engegar el timer del miniAVR? (1 punt)
- a) Escriure el limit que vulguem en el port de sortida número 2.
  - b) Escriure el limit que vulguem en el port de sortida número 10.
  - c) Escriure un `x01` en el `timer_state` en el port de sortida número 2.
  - d) Escriure un `x01` en el `timer_state` en el port de sortida número 10.