

Nom i cognoms:.....

PROVA DE PROBLEMES

Temps: 2h/3h. Obtenint 3 o més punts sobre 10, la prova puntuarà el 50% del total de l'examen. La puntuació està posada al costat de cada exercici. Les notes es publicaran a partir del dia 23 a la pàgina web de l'assignatura. La revisió de l'examen es farà el dia 27 a les 12:00h.

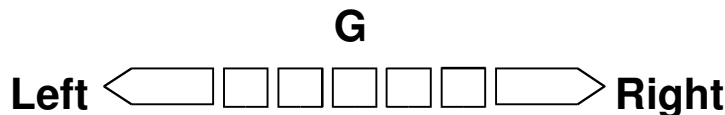
Els alumnes que tenen una nota de més de 4.5 del primer examen parcial, només han de respondre els problemes 3, 4 i 5, excepte els que vulguin pujar nota.

1.- FÓRMULA G - (Sistemes Combinacionals) (30': 4 Punts)

Realitzeu el disseny del circuit controlador de l'element visual basat en barres led, per la representació gràfica del valor de l'acceleració lateral en els girs d'un cotxe de F1.

Funcionament del sistema:

L'element visual que s'ha de controlar consisteix en una barra de 7 led, com la que es mostra a la següent figura:



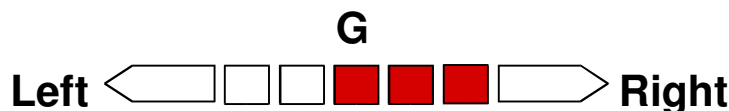
Per realitzar el disseny, cal saber que el sistema de detecció de l'acceleració lateral del cotxe està format per dos sensors (esquerre i dret) que mesuren, cada un d'ells, el valor de l'acceleració en cada una de les dues direccions laterals del cotxe (esquerra i dreta).

Cada un d'aquests sensors proporciona un valor d'acceleració lateral de dos bits, segons la següent taula:

Valor de l'acceleració lateral (dreta o esquerra) en G	Valor binari que proporciona el sensor
0 G	00
1 G	01
2 G	10
3 G	11

En la representació gràfica, s'ha de proporcionar una indicació de l'acceleració lateral del cotxe, tenint en compte **la diferència** entre els valors proporcionats pels dos sensors (esquerra i dreta).

Per exemple: si el sensor d'acceleració cap a l'esquerra indica 1 G (01_b) i el sensor d'acceleració cap a la dreta indica 3 G (11_b), aleshores el valor d'acceleració lateral és de 2 G a la dreta. En aquest cas, l'indicador lluminós hauria de proporcionar la següent mesura:



a. Quina és la taula de la veritat del circuit que recull el senyal dels dos sensors d'acceleració lateral, i proporciona els senyals per l'indicador lluminós (1,5 Punts)?

b. Realitzeu, de forma justificada, el disseny i la implementació d'aquest circuit utilitzant els blocs lògics que siguin més adients (2,5 Punts).

NOTA: En aquest apartat també es valorarà l'elecció, per part de l'estudiant, dels blocs lògics (elements de disseny lògic com MUX, ROM, NAND, etc...) més adequats (1 punt màxim del total de l'apartat).

2.- COMPARADOR DE 1 BIT - (Sistemes Seqüencials) (35': 6 Punts)

Un circuit comparador té una entrada d'un bit (X) sincronitzat amb un senyal de CK i dues sortides (Z1, Z2). L'entrada X representa un nombre binari d'un bit (N). Si el valor actual d'N és més gran que el valor anterior, aleshores $Z1 = 1$. Si el valor actual d'N és inferior al valor anterior, aleshores $Z2 = 1$. En cas contrari, Z1 i Z2 són 0. Quan es rep el 1^{er} valor de l'entrada, no existeix cap valor previ d'N, aleshores, com que no es pot realitzar la comparació, es posaran les sortides Z1 i Z2 a 0.

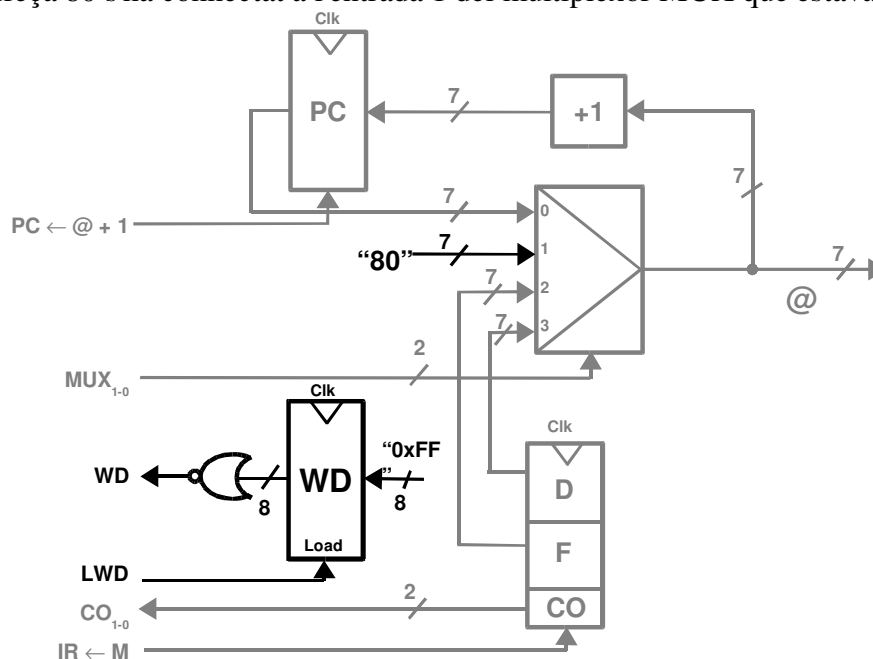
El comparador s'ha d'implementar utilitzant biestables de tipus D activats per flanc de pujada.

a. Obteniu el diagrama d'estats simplificat del sistema i la taula d'estats (2 Punts).

b. Obteniu la taula de veritat d'excitacions i les funcions de sortida. A continuació, obteniu el circuit lògic utilitzant els biestables tipus T i una memòria ROM (4 Punts).

3.- WATCHDOG - (Màquina Senzilla Hardware) (30': 3 Punts)

Un sistema **watchdog** té la finalitat d'impedir que una màquina pugui quedar bloquejada indefinidament. Per aconseguir-ho, posa l'adreça de la rutina d'inici del sistema al comptador de programa si, durant el seu temps d'activitat (2ⁿ cicles de rellotge), no rep un senyal de refresc. S'ha implementat un sistema de **Wachdog** a la MS1 posant **un descomptador WD** de 8 bits a la UP de manera que, si no s'ha refrescat, el descomptador **WD** arriba a zero i la màquina executa el codi de l'adreça 80 on hi ha la rutina d'inici del sistema. L'element (un registre, per exemple) que conté l'adreça 80 s'ha connectat a l'entrada 1 del multiplexor MUX que estava disponible.



S'implementarà una nova instrucció **REFRESH** que no té cap paràmetre. Aquesta instrucció és l'encarregada de reinicialitzar el descomptador cada vegada que es crida. Aquesta instrucció s'haurà d'anar intercalant entre les instruccions dels programes a fi de tenir el descomptador **WD** sempre actiu (que no arribi mai a 0).

NOTA: *Refrescar* significa que es torna a posar el descomptador a 0xFF per tornar a començar el descomptatge.

a. Codifiqueu la nova instrucció. Si cal, recodifiqueu les instruccions originals de la MS1. (0.75 Punts).

b. Feu el diagrama d'estats per la instrucció **ADD** de forma que es pugui interrompre en qualsevol moment pel senyal **WD** (0.5 Punts).

c. Feu el diagrama d'estats de la instrucció **REFRESH** i escriviu els valors del vector de sortides de cada un dels estats de la nova instrucció (1 Punt).

d. Feu una estimació aproximada d'on i cada quantes instruccions s'haurà d'incloure, al menys, una instrucció **REFRESH**. Justifica-ho (0.75 Punts).

4.- TEMPORITZADOR - (Màquina Senzilla Software) (30': 2.5 Punts)

a. Suposant que el rellotge de la Maquina Senzilla fa un milió d'impulsos per segon, quin valor s'hauria de posar a la constant K d'aquest programa perquè tardés un temps, el màxim aproximat possible, a **1 milisegon** en executar-se. Justifica la resposta (**1.5 Punts**).

0:	mov 100,103	100:	0	# Constant 0
1:	cmp 103,102	101:	1	# Constant 1
2:	beq fi:	102:	K	# Constant K
3:	add 101,103	103:		# Variable i
4:	cmp 100,100			
5:	beq 1			
fi:				

b. Aquest codi es podia haver fet més eficient (amb menys instruccions) encara. Recodifica el programa i les dades i calcula el nou valor de K (**1 Punt**).

NOTA: Cal fer servir la versió simplificada de la MS1 on calen 4 cicles per les instruccions ADD i CMP, 3 cicles per MOV i 2 cicles per BEQ.

5.- BINARITZACIÓ - (MIPS) (55: 4.5 Punts)

En el camp del tractament digital d'imatge, existeix una operació anomenada **segmentació**. Aquesta operació s'utilitza, normalment, per separar els objectes d'interès del fons en una determinada escena. Una de les tècniques utilitzades per realitzar una **segmentació** s'anomena **binarització**.

Una **imatge digital** es pot representar com una **matriu quadrada** d'unes determinades mides. Cada *pixel* (punt) de la imatge cal representar-lo amb un valor. Una **imatge d'intensitat** és una imatge on, cada *pixel*, representa una quantificació de la de llum que s'ha rebut de l'escena adquirida. Aquesta quantitat de llum es pot representar amb un valor entre 0 i 255 per cada *pixel*. Un 0 indica que no s'ha rebut gens de llum per aquest determinat punt de l'escena. Un valor 255 indica que s'ha rebut la màxima quantitat de llum representable per aquest determinat punt.

Per tant, una **imatge d'intensitat** es pot representar amb una **matriu bidimensional de bytes**.

L'**operació de binarització** es pot realitzar simplement escollint un **threshold** (llindar) per tal d'obtenir **una nova imatge en blanc i negre** on només es necessiten dos valors (0 i qualsevol altre) per indicar negre i blanc respectivament. Caldrà representar en negre (0) tots els *pixels* que, a la imatge original, tenien un valor d'intensitat més petit o igual que el **llindar** i blanc (diferent de 0) els altres.

En la següent parella d'imatges es pot veure un exemple del resultat de realitzar una binarització.



Figura 1. Imatge d'intensitat en escala de grisos. **Figura 2.** Imatge binaritzada (amb threshold 70).

Es demana:

a. Implementar, en assembleador de MIPS, una rutina per realitzar una binarització. Es pot utilitzar, de forma orientativa, la següent signatura en *pseudocodi* (**2 Punts**):

`acció binary (SourceGrayImg : Adreça, DestBinImg : Adreça, Width : Natural, Height : Natural, Threshold : Natural);`

On *SourceGrayImg* i *DestBinImg* són les adreces de memòria on es troben les dues imatges, *Width* és l'amplada de les imatges, *Height* és l'altura de les imatges i *Threshold* és el llindar de binarització.

b. Implementar, en assembleador d'SPIM, el segment de dades amb les definicions necessàries per emmagatzemar la imatge original en escala de grisos i la imatge binària destí. Les imatges són de 192x144 *pixels* i el *threshold* és 70 (**1 Punt**).

c. Suposant que es té implementada correctament la funció de l'apartat **a**, realitzeu el programa principal (*main*) en assembleador de MIPS, per realitzar una binarització amb les dades definides a l'apartat **b** (**1.5 Punts**).

Recordatori

- Els **Naturals** per *Width* i *Height* són de **32 bits** i el **Natural** per *Threshold* és de **8 bits**.
- Cal seguir el conveni MIPS pel pas de paràmetres ja que altres implementadors hauràn d'utilitzar les vostres rutines: cal passar els 4 primers paràmetres per registre (a0, ..., a3) i els altres per la pila.
- També cal seguir el conveni MIPS en la utilització dels registres, és a dir, que podeu utilitzar t0, ..., t9 dins una rutina sense preocupar-vos de guardar-los (són *caller save*). Si utilitzeu un dels registres s0, ..., s7 dins una rutina, cal salvar els seu valor i restaurar-lo abans de retornar (són *callee save*).
- Per passar els paràmetres considereu l'ordre d'esquerra a dreta. El primer paràmetre és el de més a l'esquerra i, el darrer, el de més a la dreta de la signatura especificada.