

COGNOM, NOM: **SOLUCIONS**

**PROBLEMA 1** (4 punts) Les persianes d'un edifici estan regulades amb l'ajut d'un sistema de control que té la interfície mostrada en la figura 1. El sistema inclou sensors de pluja i de llum i un motor que obre i tanca una persiana. Dos sensors de contacte addicionals indiquen si la finestra està oberta (A) o tancada (B). Els sensors A i B mai no poden estar actius simultàniament. Quan tots dos estan inactius, la persiana pot estar bé obrint-se o tancant-se.

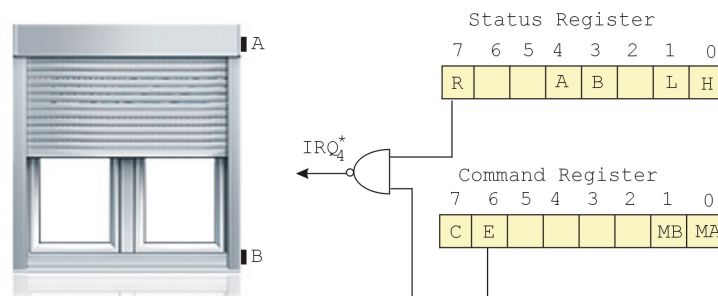


Figura 1: Registres de la interfície del controlador de la persiana.

El controlador de les persianes està ubicat en l'adreça base 0xFF000020 d'un computador basat en un MIPS. Tots dos registres fan servir la mateixa adreça i tenen l'estructura que es descriu tot seguit:

**Status register (Read only):**

- **H:** Sensor de pluja. Es fa 1 quan plou i 0 en altre cas.
- **L:** Sensor de llum. El hardware el fa 1 durant el dia i 0 per la nit.
- **B:** Sensor de persiana tancada. A 1 indica que la persiana està completament tancada. Quan val 0, indica que la persiana està oberta o en moviment.
- **A:** Sensor de persiana oberta. Quan val 1 indica que la persiana està oberta, si val 0 indica que la persiana està tancada o en moviment.
- **R:** Bit de preparat. El hardware el fa 1 quan hi ha algun canvi en els bits H o L.

**Command register (Write only):**

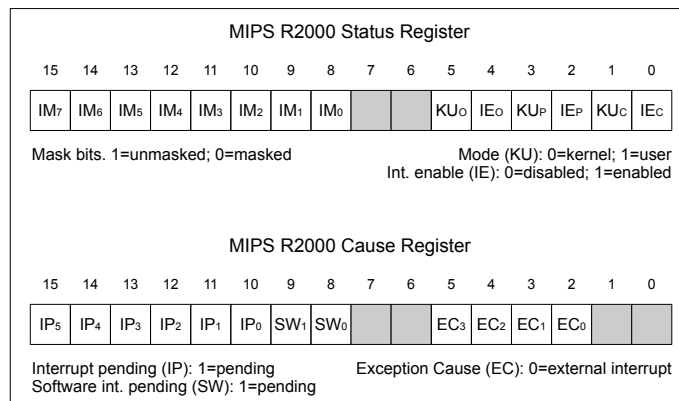
- **MA:** Quan val 1, la persiana puja fins quedar-se oberta. El sistema atura el motor automàticament quan la persiana està oberta del tot.
- **MB:** Quan val 1, la persiana baixa fins tancar-se. El sistema atura el motor automàticament quan la persiana està tancada del tot.
- **E:** Quan val 1, el hardware envia una interrupció  $irq_4$  cada vegada que el bit de preparat passa a 1.
- **C:** Bit de cancel·lació. Quan s'hi escriu 1, el bit (R) de preparat del registre Status passa a valdre 0.

1. (0.25 punts) Escriviu les instruccions d'inicialització que habiliten les interrupcions en la interfície del controlador de la persiana.

**SOLUCIÓ:**

```
la $t0, 0xFF000020
li $t1, 0x40 # fa bit E = 1
sb $t1, 0($t0)
```

2. (0.25 punts) Escriviu el codi d'inicialització que desemmascara la interrupció  $irq_4$  en el MIPS i deixa el processador en mode usuari, amb les interrupcions habilitades en general. No podeu modificar la resta de bits del registre d'estat del processador. Vegeu la informació sobre els registres `$Status` i `$Cause` del MIPS.



### SOLUCIÓ:

```

mfc0 $t0, $12          # llig el registre d'estat
ori $t0, $t0, 0x1003    # hi canvia IM4=KUC=IEC=1
mtc0 $t0, $12          # escriu en registre d'estat

```

3. (1 punt) Escriviu un fragment de codi que implementa el comportament descrit més avall fent servir **consulta de l'estat**:

```

quan R val 1:
  if és de dia then
    if no plou then
      obre persiana
    else
      tanca persiana
  else
    tanca persiana
  end if

```

### SOLUCIÓ: Primera versió, més ajustada a l'especificació:

```

Finestra:
  la $t0, 0xFF000020
loop:  lb $t1, 0($t0)          # llig l'estat
      andi $t2, $t1, 0x80     # aïlla bit R
      beqz $t2, loop          # mentre R = 0, llig estat
      andi $t2, $t1, 2        # aïlla bit L
      beqz $t2, else          # si L = 0, baixar persiana
then:  andi $t2, $t1, 1        # aïlla bit H
      bnez $t2, else          # si H = 1, baixar persiana
then2: li $t1, 0x81           # prepara C = MA = 1
      sb $t1, 0($t0)          # apuja persiana i cancel·la
      jr $ra                  # fi
else:  li $t1, 0x82           # prepara C = MB = 1
      sb $t1, 0($t0)          # baixa persiana i cancel·la
      jr $ra

```

### Segona opció, amb codi més compacte:

```

Finestra: la $t0, 0xFF000020
loop:     lb $t1, 0($t0)      # llig l'estat
          andi $t2, $t1, 0x80 # aïlla bit R
          beqz $t2, loop
          andi $t1, $t1, 0x18 # aïlla bits A i B
          srl $v0, $t1, 0x3   # desplaça tres bits a la dreta
          b retexc

```

4. (1 punt) Implementeu la funció de sistema especificada més avall. Noteu que el registre *Status* del controlador manté l'estat de la persiana sempre actualitzat. Escriviu només el codi de la funció, suposant sempre que comença en l'etiqueta *Blind.Status*.

Funció	Index	Arguments	Resultats
<b>blind_status</b>	\$v0 = 30	—	\$v0 = 0 (en moviment) \$v0 = 1 (tancada) \$v0 = 2 (oberta)

### SOLUCIÓ:

```

Blind_Status:
    la $t0, 0xFF000020
    lb $t1, 0($t0)      # llig l'estat
    andi $t0, $t1, 0x10  # aïlla el bit A
    beqz $t0, VeureB     # si A=0, va a veure el bit B
    li $v0, 2            # (A=1): retorna persiana pujada
    b retexc
VeureB:
    andi $t0, $t1, 0x08  # aïlla el bit B
    beqz $t0, Almig      # si B=0, la persiana està al mig
    li $v0, 1            # (B=1): retorna persiana baixada
    b retexc
Almig:
    li $v0, 0            # (A=0, B=0): retorna persiana al mig
    b retexc

```

5. (1.5 punts) Considereu el fragment de codi següent, que implementa certa funció del sistema:

```

unknown_function:
    la $t0, 0xFF000020
    lb $t1, 0($t0)
    andi $t1, $t1, 1
    bnez $t1, retexc
    li $t1, 0x40
    sb $t1, 0($t0)
    jal suspend_this_process    # Suspèn el procés que crida
    b retexc

```

- (a) (0.5 punts) Indiqueu si les frases següents són veritables o falses (V/F):

La rutina suspen el procés invocant si no està plovent	
La rutina no sempre suspèn el procés invocant	
La rutina és errònia perquè no sincronitza	
La rutina alça la persiana i suspèn el procés invocant	

### SOLUCIÓ:

#### Anàlisi del codi de la funció

la \$t0, 0xFF000020	Llig el registre d'estat de la persiana
lb \$t1, 0(\$t0)	Aïlla el bit H
andi \$t1, \$t1, 1	Si està plovent, acaba
bnez \$t1, retexc	(Si no plou:)
li \$t1, 0x40	Habilita les interrupcions de la persiana
sb \$t1, 0(\$t0)	Suspèn el procés que ha cridat
jal suspend_this_process	
b retexc	

Aquest codi comprova si plou. Si no és el cas, habilita les interrupcions i suspèn el procés que la va cridar. Per tant, correspon a una funció que fa que el procés espere a que ploga.

Per tant,

La rutina suspèn el procés invocant si no està plovent	V
La rutina no sempre suspèn el procés invocant	V
La rutina és errònia perquè no sincronitza	F
La rutina alça la persiana i suspèn el procés invocant	F

- (b) (1 punt) Implementeu la rutina de servei d'interrupció per a `irq4`. Aquesta rutina ha de deixar preparat el procés suspès per la funció `unknown_function` anterior. Afegiu els comentaris adients al vostre codi. Supposeu que hi ha una subrutina de nom `activate_waiting_proc` que gestiona l'estat i les cues de processos. Noteu que el procés només s'ha d'activar quan la condició que el va suspendre ja no s'acompleix.

**SOLUCIÓ:** Com que la funció del sistema suspenia el procés que la cridava si no pluvia, ara caldrà despertar-lo (és a dir, deixar-lo en estat de preparat) quan ploua.

Per tant, el tractament de la interrupció 4 haurà de consultar el bit *H* i reactivar el procés si plou; és a dir, si *H* = 1.

Cal gestionar correctament la interrupció de la persiana: en reactivar el procés, caldrà inhibir-la; però en qualsevol cas caldrà cancel·lar-la

```
int4:    la $t0, 0xFF000020
         lb $t1, 0($t0)           # llig l'estat
         andi $t1, $t1, 1         # filtre el bit H
         beqz $t1, fi             # Si H = 0, salte a fi
         li $t1, 0x80             # C=1, E=0 Cancel·lar i inhibir interrpt.
         sb $t1, 0($t0)
         jal activate_waiting_proc # Activa el proceso
         b retexc

fi:      li $t1, 0xC0             # C=E=1, cancel·lar mantenint interrupció
         sb $t1, 0($t0)
         b retexc
```

**PROBLEMA 2** (3 punts) La figura 2 mostra l'adaptador per a cert perifèric de blocs amb ADM incorporat, que està connectat a un computador basat en un MIPS R2000.

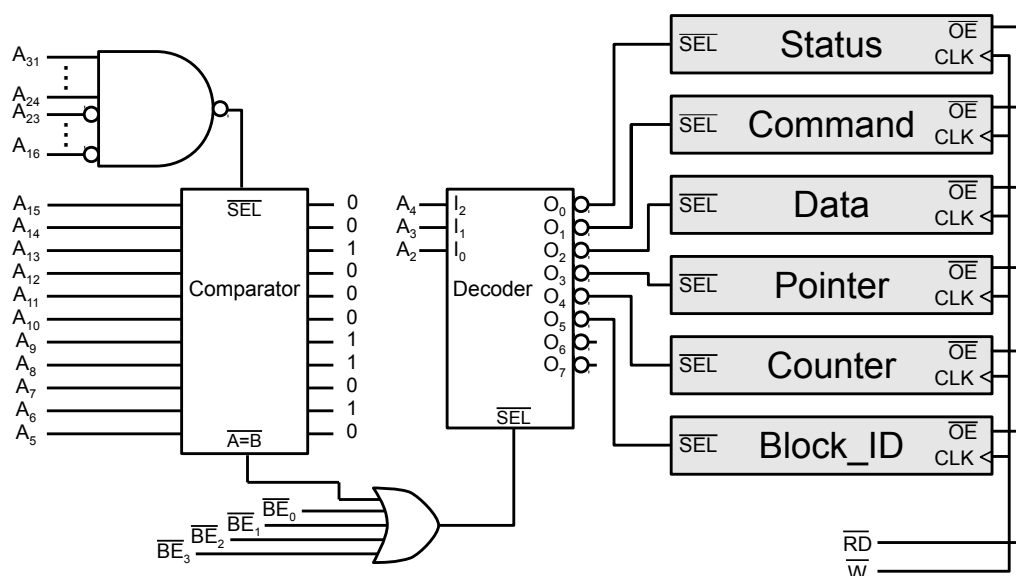


Figura 2: Adaptador per a un perifèric de blocs amb ADM

Noteu que les línies d'adreça des de  $A_{31}$  fins a  $A_{24}$  estan connectades directament a la porta NAND mentre que les línies de  $A_{23}$  a  $A_{16}$  estan connectades a entrades inversores. Els sis registres de la interfície són de 32-bits.

- (1 punt) Quina és l'adreça base (AB) d'aquest dispositiu?

**SOLUCIÓ:**  $AB = 0xFF002340$

- (0.6 punts) Escriviu l'adreça de cadascun dels sis registres de la forma  $\langle AB + offset \rangle$ , amb *offset* expressat en hexadecimal.

**SOLUCIÓ:**

Status: $AB$	Pointer: $AB + 0x0C$
Command: $AB + 4$	Counter: $AB + 0x10$
Data: $AB + 8$	Block_ID: $AB + 0x14$

3. (0.4 punts) Suposant que el registre \$t0 conté l'adreça del registre d'estat de la interfície, indiqueu si les següents frases relatives a l'execució de la instrucció **lb \$t1, 0(\$t0)** són veritables o falses (V/F):

La instrucció carrega \$t1 amb tot el contingut del registre <i>Status</i>	
La instrucció carrega en \$t1 el byte menys significatiu del registre <i>Status</i>	
La instrucció no accedeix al registre <i>Status</i>	
La instrucció carrega \$t1 amb l'adreça del registre <i>Status</i>	

**SOLUCIÓ:** Atés que per a seleccionar els registres de la interfície totes les línies  $BE_i^*$  han de valdre zero, només és possible operar amb una les instruccions `lw` i `sw`.

La instrucció `lb` només activarà la línia  $BE_0^*$ , per tant:

La instrucció carrega \$t1 amb tot el contingut del registre <i>Status</i>	<i>F</i>
La instrucció carrega en \$t1 el byte menys significatiu del registre <i>Status</i>	<i>F</i>
La instrucció no accedeix al registre <i>Status</i>	<i>V</i>
La instrucció carrega \$t1 amb l'adreça del registre <i>Status</i>	<i>F</i>

4. (1 punt) Escriviu el codi que programa una transferència de lectura amb ADM des del perifèric a la memòria. L'operació de lectura comença quan el software escriu un 1 en els bits 0 i 3 de *Command* i un 0 en la resta de bits. Heu de llegir el bloc amb identificador 0xABCD1234. Els blocs són de 512 bytes però l'amplada del bus de dades és 32-bit. Per tant, cada cicle ADM transfereix una paraula completa de 32-bit i decrementa en 1 el registre Counter. Heu d'emmagatzemar el bloc a partir de l'adreça etiquetada com *Mem\_Block* de la memòria principal.

**SOLUCIÓ:**

```
la $t0, 0xFF002340
li $t1, 0xABCD1234    # Block_ID
sw $t1, 8($t0)
la $t1, Mem_Block     # Pointer
sw $t1, 0x10($t0)
li $t1, 128           # Count = 512 B/block / 4 B/cycle = 128 cycles/block
sw $t1, 0xC($t0)
li $t1, 9              # Command: set bits 0 and 3 only
sw $t1, 4($t0)
```

**PROBLEMA 3** (1 punt) Considereu un disc magnètic amb els paràmetres següents:

**Velocitat de rotació:** 6000 RPM

**Densitat lineal:** 5000 tpi

**Nombre de cares:** 4

**Radi intern:** 1"

**Radi extern:** 3"

**Capacitat d'un sector:** 512 Bytes

**Format:** ZCAV amb dues zones:

Zona	Limits	Sectors/track
0	2--3"	500
1	1--2"	300

1. (0.5 punts) Calculeu la velocitat de transferència (en MB/s) del disc en accedir a la zona 0.

**SOLUCIÓ:** El disc gira a 6000 RPM, és a dir, a

$$\frac{6000 \text{ RPM}}{60 \text{ segons/minut}} = 100 \text{ rotacions per segon}$$

i per tant el temps per fer una volta és

$$\frac{1}{100} = 10 \text{ ms/rotació}$$

En aquest temps, el disc pot transferir una pista completa, que en la zona 0 correspon a 500 sectors de 512 Bytes. Podem obtenir la velocitat de transferència:

$$\frac{500 \text{ sectors} \times 512 \text{ bytes/sector}}{10 \text{ ms}} = 25.6 \text{ MB/s}$$

2. (0.5 punts) Calculeu la capacitat total d'aquest disc en MB ( $1M = 10^6$ ).

**SOLUCIÓ:** L'amplària de qualsevol zona és 1", per tant cadascuna conté 5000 pistes per cara. La capacitat total de la zona 0 és

$$5000 \text{ pistes} \times 500 \text{ sectors/pista} \times 4 \text{ cares} \times 512 \text{ B/sector} = 5,120,000,000 \text{ Bytes} = 5120 \text{ MB}$$

En la zona 1 hi ha

$$5000 \text{ pistes} \times 300 \text{ sectors/pista} \times 4 \text{ cares} \times 512 \text{ B/sector} = 3,072,000,000 \text{ Bytes} = 3072 \text{ MB}$$

En sumar la capacitat de les zones 0 i 1, tenim

$$5120 + 3072 = 8192 \text{ MB}$$

**PROBLEMA 4** (2 punts) La figura 3 mostra l'estructura del bus i els perifèrics d'un computador donat. Totes les amplades de banda especificades són efectives, excepte per als busos SATA i PCIe 1x, que fan servir 10 bits per a codificar cada byte. El disc dur HD2 conté un arxiu amb un pel·lícula de dos minuts de durada comprimida amb la codificació H265, és a dir, a 30 Mbps. La pel·lícula descomprimida està formada per quadres de video de  $1920 \times 1200 \times 24$  bits a 30 fps i audio 5.1 at 24 bits per canal mostrejat a 96 KHz.

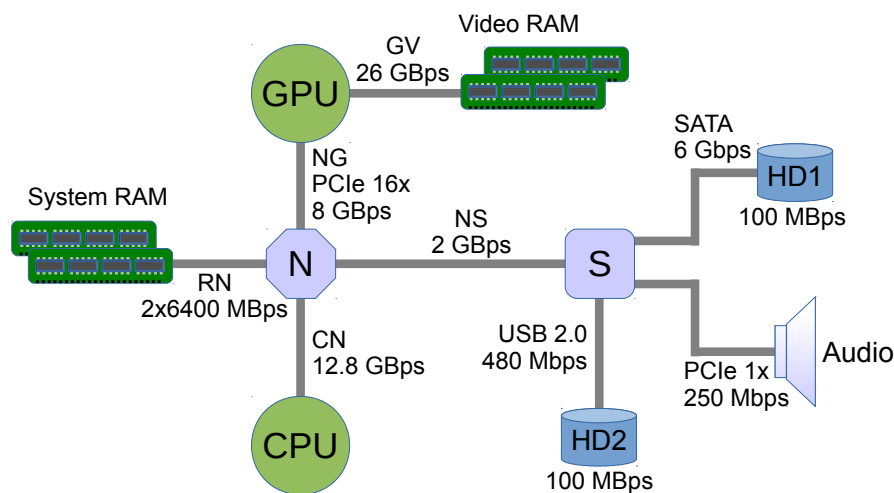


Figura 3: Estructura dels busos i perifèrics.

1. (0.5 punts) Calculeu el llarg de l'arxiu de la pel·lícula comprimida.

**SOLUCIÓ:**

$$2 \text{ minuts} \times 60 \text{ segons/minut} \times \frac{30 \cdot 10^6 \text{ bits/segon}}{8 \text{ bits/byte}} = 450 \cdot 10^6 \text{ bytes} = 450 \text{ MB}$$

2. (0.5 punts) Calculeu el temps que costarà copiar l'arxiu de la pel·lícula comprimida des de HD2 a HD1. La còpia consisteix a llegir l'arxiu des de HD2 a la RAM del sistema, i transportar-lo des de la RAM del sistema a HD1. Ambdues transferències fan servir ADM. Supposeu que les dues transferències es fan concurrentment.

**SOLUCIÓ:** La velocitat efectiva màxima de transferència de cadascun dels busos implicats, en MBps, serà:

**Bus RN** = 12800 MB/s

**Bus NS** = 2000 MB/s

**Bus USB** (no codificat) = 60 MB/s

**Bus SATA** (codificat a 8/10) =  $6 / 10 \text{ GB/s} = 600 \text{ MB/s}$

Per tant,

- La transferència d'1 GB de la càmera a la memòria, està limitada pels 60 MBps bus USB1 i necessita  $1 \text{ GB} / 60 \text{ MBps} = 16.7 \text{ seconds}$ .

- La transferència de memòria al disc està limitada pel propi disc (120 MBps), que demana 1 GB / 120 MBps = 8.3 seconds
- La còpia completa vindrà limitada pel bus USB1.
- El temps mínim per a fer la còpia serà:

$$\text{temps} = \frac{450 \text{ MB}}{60 \text{ MB/segon}} = 7,5 \text{ s.}$$

3. (0.5 punts) Calculeu el percentatge d'ús de tots els busos implicats en la transferència anterior.

**SOLUCIÓ:**

$$\text{RN Bus: } \frac{60 + 60 \text{ MBps}}{12800 \text{ MBps}} \simeq 1 \%$$

$$\text{NS Bus: } \frac{60 + 60 \text{ MBps}}{2000 \text{ MBps}} = 6 \%$$

$$\text{USB Bus: } \frac{60 \text{ MBps}}{60 \text{ MBps}} = 100 \%$$

$$\text{Bus SATA: } \frac{60 \text{ MBps}}{600 \text{ MBps}} = 10 \%$$

4. (0.5 punts) Supposeu ara que la pel·lícula està descomprimida en la memòria RAM. Justifiqueu si l'amplada de banda del bus NG serà suficient per a reproduir el vídeo descomprimit.

**SOLUCIÓ:** La reproducció demana l'amplada de banda  $B$ :

$$B = 30 \text{ frame/s} \times 1920 \times 1200 \text{ píxels/frame} \times 24 \text{ bits/píxel} = 1658880000 \text{ bps} = 207,36 \text{ MBps}$$

Si el bus NG suporta 8000 MBps, aquest consum suposa un  $207,36/8000 \simeq 2,6\%$  del màxim. Per tant, és prou per a visualitzar el vídeo descomprimit.