

# Entregable tema 8º Part 1

## Problema 1º

Adreça = 0xFFFF0000 palanca.

R → 1 es desplaça a A o a B

R → 0 cancel·la

Perilla L = 0xFFFF1000

Perilla Z = 0xFFFF2000

$$C = 1 \rightarrow R = 0$$

$$L = 1 \rightarrow \text{s'encen}$$

$$AB = 10 \rightarrow A$$

$$L = 0 \rightarrow \text{s'apaga}.$$

$$AB = 01 \rightarrow B$$

$$AB = 00 \rightarrow \text{Repòs}$$

- ① la \$t0, 0xFFFF0000 } Carga l'adreça  
de la palanca.  
la \$t1, 0xFFFF1000 } Carga les  
adreces de les  
peretes  
la \$t2, 0xFFFF2000 } Apaga les perilles  
sb \$zero, 0(\$t1) }  
sb \$zero, 0(\$t2)}

sync: lb \$t3, 0(\$t0) } Espera a que  
bgez \$t3, sync } R=1

clear: li \$t3, 8 } Col·loca  
sb \$t3, 4(\$t0) } C=1 → R=0

read: lb \$t3, 8(\$t0) } Comprova quina  
andi: \$t3, \$t3, 3 } pereta hem  
seleccionat

switch: li \$t4, 1

si es 01 }  
beq \$t3, \$t4, case 1 } activem B

li \$t4, 2 } si es 10 }  
begz \$t3, \$t4, case 2 } activem A

case0: sb \$zero, 0(\$t1) } Repòs  
sb \$zero, 0(\$t2)}

j sync

case1: li \$t3, 2 }  
sb \$t3, 0(\$t2) } Activem B

j sync

case2: li \$t3, 2 }  
sb \$t3, 0(\$t1) } Activem A

j sync

② la \$t0, 0xFFFF0000

beq \$t3, \$t4, case 1

li \$t4, 2

begz \$t3, \$t4, case 2

case0: j sync

case1: lb \$t3, 0(\$t2)

xori \$t3, \$t3, 2

sb \$t3, 0(\$t2)

j sync

case2: lb \$t3, 0(\$t1)

xori \$t3, \$t3, 2

sb \$t3, 0(\$t1)

j sync

clear: li \$t3, 8

sb \$t3, 4(\$t0)

read: lb \$t3, 8(\$t0)

andi: \$t3, \$t3, 3

switch: li \$t4, 1

## Problema 2

(1)

```

        .Ktext 0x80000080
        sw $at, 0($K1)
        sw $t0, 4($K1) } guarda el
        sw $t1, 8($K1) context
        mfc0 $K0,$13
        andi $t0,$K0,0x003C } Comprueba
        bnez $t0,retexc el tipo
        andi $t0,$K0,0x0800 } de interrupción
        beqz $t0,retexc
        clear: la $t0 0xFFFF0000
        li $t1,8
        sb $t1,4($t0) } col.loca
        c=1 (cancella)
        l=0
        read: lb $t1,8($t0) } Comprueba
        andi $t1,$t1,3 la perota
        switch: li $t0,1 seleccionada
        beq $t1,$t0,case1 } Si es B
    
```

```

        li $t0,2
        beqz $t1,$t0,case2 } Si es A
        case0: la $t0, 0xFFFF1000
        sb $zero, 0($t1) } Apaga
        la $t0, 0xFFFF2000 les peretas
        sb $zero, 0($t1) } j retexc
        case1: la $t0, 0xFFFF2000
        li $t1,2
        sb $t1,0($t0) } Encen
        j retexc B
        case2: la $t0, 0xFFFF1000
        li $t1,2
        sb $t1,0($t0) } Encen
        j retexc A
        retexc: lw $at, 0($K1)
        lw $t0, 4($K1)
        lw $t1,8($K1)
        mfc0 $K0,$14
        rfe
        jr $K0
    
```

(2)

```

        .Ktext 0x80000080
        sw $at, 0($K1)
        sw $t0, 4($K1)
        sw $t1, 8($K1)
        mfc0 $K0,$13
        andi $t0,$K0,0x003C
        bnez $t0,retexc
        andi $t0,$K0,0x0800
        beqz $t0,retexc
        clear: la $t0 0xFFFF0000
        li $t1,8
        sb $t1,4($t0)
        read: lb $t1,8($t0)
        andi $t1,$t1,3
        switch: li $t0,1
        beq $t1,$t0,case1
        li $t0,2
        beqz $t1,$t0,case2
    
```

```

        case0: j retexc
        case1: la $t0 0xFFFF2000
        lb $t1,0($t0)
        xor $t1,$t1,0x02
        sb $t1,0($t0)
        j retexc
        case2: la $t0,0xFFFF2000
        lb $t1,0($t0)
        xor $t1,$t1,0x02
        sb $t1,0($t0)
        j retexc
        retexc: lw $at, 0($K1)
        lw $t0, 4($K1)
        lw $t1,8($K1)
        mfc0 $K0,$14
        rfe
        jr $K0
    
```

# Entregable Tema 8 :

## Part 2:

### Problema 3:

① li \$v0,30 } 0  
 syscall  
 li \$v0,33 } 1  
 syscall  
 li \$v0,33 } 2  
 syscall  
 li \$v0,33 } 3  
 syscall  
 li \$v0,32 } 3  
 syscall  
 #Punt(1) → \$v0 = 3

li \$v0,31 }  
 li \$a0,100 } 100  
 syscall  
 li \$v0,33 } 101  
 syscall  
 li \$v0,32 } 101  
 syscall  
 #Punt(2) → \$v0 = 101

### ②

Nom	\$v0	arguments	return	descripció
Reset(Sum30)	30	—	—	posa "var" a 0
Set (Sum31)	31	\$a0 = valor nou de var	—	posa el valor de "\$a0" en "var"
get (Sum32)	32	—	\$v0 = "var"	retorna el valor de var
increment(Sum33)	33	—	—	incrementa "var" en 1

③ add\_to\_counter: la \$t0, var  
 lw \$t1,0(\$t0)  
 add \$v0,\$t1,\$a0  
 sw \$v0,0(\$t0)  
 j retexc

.ktext  
 li \$t1,34  
 beq \$v0,\$t1,add\_to\_counter  
 j retexc

### Problema 4:

④ .ktext 0x80000080

sw \$at,0(\$k1)  
 sw \$t0,\$(\$k1)  
 sw \$t1,8(\$k1)

} Salvar el context

mfc0 \$k0,\$13 → carregar el registre causa  
 andi \$t0,\$k0,0x003C} agarra els bits de excepcíó  
 beqz \$t0,Interrupcions → si es 0, son interrupcions  
 li \$t1,0x20 → carrega la mascara de syscall  
 beq \$t0,\$t1,Circles → va a circles si es syscall  
 j retexc → ix del anijnejador

Interrupcions: andi \$t0, \$KO, 0x0800 } Mira si és la interrupció 1  
begz \$t0, retexc } Si es 0 torna a retexc

clear: li \$t3, 0x48 } Cancel·lar i habilitar les interrupcions  
sb \$t3, 4(\$t0) }  
j retexc

Errides: li \$t1, 87 } Comprova quina funció  
beg \$t1, \$v0, SumX } s'ha cridat per la interrupció;  
li \$t1, 88 } si no es cap ix a retexc.  
beg \$t1, \$v0, SumY }  
j retexc

SumX: li \$t1, 2 } Encen  
la \$t0, 0xFFFF1000 } amb dues peretes  
sb \$t1, 0(\$t0) } i ix a retexc  
la \$t0, 0xFFFF2000 }  
sb \$t1, 0(\$t0) }  
j retexc

SumY: la \$t0, 0xFFFF100 } reforma el el valor actual  
lb \$v0, 4(\$t0) } de la peret A:B i  
la \$t0, 0xFFFF200 } suma els seus valors.  
lb \$t1, 4(\$t0) } Desplaça en 1 posició el  
add \$v0, \$v0, \$t1 } resultat a la dreta.  
srl \$v0, \$v0, 1 }

retexc: lw \$af, 0(\$K1) } restaura el  
lw \$t0, 4(\$K1) } context i  
lw \$t1, 8(\$K1) } torna al  
mfco \$KO, \$14 } mode usuari  
rfe }  
jr \$KO }

Nom	\$v0	arguments	reform	descripció
SumX	87	—	—	Encen ambdues peretes
SumY	88	—	\$v0 = valor de les peretes	suma els valors de les peretes i desplaça 1 posició a la dreta

② GetHandle: la \$t0, 0xFFFF0000  
 lw \$t1, 8(\$t0)  
 andi \$t1, \$t1, 0x3  
 beqz \$t1, palanca\_centre  
 li \$t0, 0x1  
 beq \$t1, \$t0, posición\_B  
 posiciónA: li \$v0, 0x02  
 j retexc  
 posiciónB: li \$v0, 0x01  
 j retexc  
 palanca\_centre: li \$v0, \$zero  
 j retexc

SetLightOn: addi \$t1, \$a0, 0x01  
 beq \$t1, perillaB  
 addi \$t1, \$a0, 0x02  
 beq \$t1, perillaA  
 j retexc

GetLight: addi \$t1, \$a0, 0x01  
 beq \$t1, estatB  
 addi \$t1, \$a0, 0x02  
 beq \$t1, estatA  
 j retexc

apagar: la \$t0, 0xFFFF1000 ## Funció per a la activitat 4  
 sw \$zero, 0(\$t0) ## El codi de la funció és el 91.  
 la \$t0, 0xFFFF2000  
 sw \$zero, 0(\$t0)  
 j retexc

Crides: :  
 li \$t1, 90  
 beq \$t1, \$v0, GetHandle  
 li \$t1, 85  
 beq \$t1, \$v0, SetLightOn  
 li \$t1, 86  
 beq \$t1, \$v0, GetLight  
 li \$t1, 91  
 beq \$t1, \$v0, apaga  
 j retexc

perillaA: li \$t1, 0x02  
 la \$t0, 0xFFFF1000  
 sw \$t1, 0(\$t0)  
 j retexc  
 perillaB: li \$t1, 0x02  
 la \$t0, 0xFFFF2000  
 sw \$t1, 0(\$t0)  
 j retexc

estatA: la \$t0, 0xFFFF1000  
 lw \$v0, 0(\$t0)  
 srl \$v0, \$v0, 1  
 j retexc  
 estatB: la \$t0, 0xFFFF2000  
 lw \$v0, 0(\$t0)  
 srl \$v0, \$v0, 1  
 j retexc

(3) bucle: li \$v0, 90  
 syscall  
 li \$t1, 0x1  
 beq \$t1, \$v0, esB  
 li \$tL, 0x2  
 beq \$t1, \$v0, esA  
 li \$v0, 91 ## apagar 2 perilles  
 syscall

esA: li \$v0, 85  
 li \$a0, 0x02  
 syscall  
 j bucle

esB: li \$v0, 85  
 li \$a0, 0x01  
 syscall  
 j bucle

---

(4) .Ktext 0x80000080  
 sw \$at, 0(\$K1)  
 sw \$t0, 4(\$K1)  
 sw \$t1, 8(\$K1)  
 mfco \$K0, \$13  
 andi \$t0, \$K0, 0x0003C  
 beqz \$t0 Interrupcions  
 li \$tL, 0x20  
 beq \$t0, \$t1 Crides  
 j refexc

Interrupcions: andi \$t0, \$K0, 0x0800  
 beqz \$t0, refexc

clear: li \$tL, 8  
 sb \$tL, 4(\$t0)

read: lb \$tL, 8(\$t0)

and i \$t1, \$tL, 1  
 beqz \$tL, refexc  
 jal activa\_proceso\_espera  
 j refexc

Crides: li \$t1, 89  
 beq \$t1, \$v0, Sunz  
 j refexc

Sunz: jal suspen\_aquest\_proc

refexc: jal fixa\_context  
 lw \$at, 0(\$K1)  
 lw \$t0, 4(\$K1)  
 lw \$t1, 4(\$K1)  
 mfco \$K0, \$14  
 rfe  
 jr \$K0

La funció "Sunz" crida a la funció de suspender el proces, que suspendrà el proces interromput. La funció està en bucle sense utilitzar variables i quan hi haja una interrupció tornara a comprovar la interrupció: