

COGNOMS:

NOM:

SIGNATURA:

1) Escriu un programa en *assembler* de SISA-F que es sincronitzi per enquesta i que compti el nombre de bits que valen 1 en un nombre de 16 bits emmagatzemat a memòria a través de la variable *w* (per exemple, $w=0xF0E0$). Mostreu el resultat de la compta per pantalla en la fila i columna que doni el resultat (per exemple, si el resultat es 5, en la fila i columna 5).

(2,5 puntos)

```
.include "macros.s"
.include "crt0.s"
```

```
.data
```

```
    w: .word 0xF0E0
```

```
.text
```

```
main:
```

```
    $MOVEI R0, w           ; posem en R0 l'adreça de memòria w
    LD R1, 0(R0)          ; posem en R1 el contingut de l'adreça de memòria w
    XOR R3, R0, R0        ; fem servir R3 per comptar el número d'uns. L'inicialitzem a 0
    MOVI R4, 0x01         ; posem en R4 una màscara de bits 0000000000000001
    MOVI R5, 16           ; fem servir R5 com a comptador d'iteracions pendents
    MOVI R6, 0xFF        ; fem servir R6 per salvar el valor -1 per a els desplaçaments
```

```
Bucle:
```

```
    AND R2, R1, R4        ; obtenim el bit de menor pes de R1 i el deixem en R2
    ADD R3, R3, R2        ; incrementem el comptador d'uns (R3) si el valor de R2 val 1
    SHL R1, R1, R6        ; desplaçem el contingut de R1 un bit cap la dreta
    ADDI R5, R5, -1       ; disminuïm el comptador d'iteracions pendents (R5)
    BNZ R5, Bucle        ; si encara queden iteracions tornem a l'inici del bucle
    OUT Rfil_pant, R3
    OUT Rcol_pant, R3
    $MOVEI R2, '0
    ADD R3, R3, R2
    OUT Rdat_pant, R3
    $MOVEI R0, 0xF000
    OUT Rcon_pant, R0    ; mostrem el resultat per pantalla
```

```
HALT
```

2) Escriu un programa en *assembler* de SISA-F que es sincronitzi per interrupcions i que, a l'igual que el problema 1), compti el número de bits que son "1" de w (per exemple, w= 0xF0E0), i els mostri en pantalla després de 10 s. A més, quan es premi una tecla, envieu a la Impressora el nombre de bits que valen "0". Escriu el "main" i les dues RSI. No cal fer una RSI per a la impressora. RSI del relox amb nom **clock**. RSI del teclat amb nom **teclat**.

(2,5 puntos)

```
.include "macros.s"
.include "crt0.s"

.data
    w: .word 0xF0E0
    ticks: .word 0 ; variable global para acumulador de ticks
    final: .byte 0 ; variable global indica que se alcanzó los 10 s
           .balign 2
    tecla: .byte 0 ; variable global que guarda la tecla pulsada
           .balign 2

.text
main:
    $MOVEI R0, interrupts_vector
    $MOVEI R1, clock
    ST 0(R0), R1
    MOVI R2,1
    OUT Rcon_rel, R2
    $MOVEI R1, teclat
    ST 2*1(R0), R1 ;codi id del teclat es 1
    MOVI R2, 1
    OUT Rcon_tec, R2
    EI
    $MOVEI R0, final

bucle1:
    LDB R2, 0(R0)
    BZ R2, bucle1

    $MOVEI R0, w ; posem en R0 l'adreça de memòria w
    LD R1, 0(R0) ; posem en R1 el contingut de l'adreça de memòria w
    XOR R3, R0, R0 ; fem servir R3 per comptar el número d'uns. L'inicialitzem a 0
    MOVI R4, 0x01 ; posem en R4 una màscara de bits 0000000000000001
    MOVI R5, 16 ; fem servir R5 com a comptador d'iteracions pendents
    MOVI R6, 0xFF ; fem servir R6 per salvar el valor -1 per a els desplaçaments

bucle:
    AND R2, R1, R4 ; obtenim el bit de menor pes de R1 i el deixem en R2
    ADD R3, R3, R2 ; incrementem el comptador d'uns (R3) si el valor de R2 val 1
    SHL R1, R1, R6 ; desplaçem el contingut de R1 un bit cap la dreta
    ADDI R5, R5, -1 ; disminuïm el comptador d'iteracions pendents (R5)
    BNZ R5, bucle ; si encara queden iteracions tornem a l'inici del bucle
    OUT Rfil_pant, R3
```

```

    OUT Rcol_pant, R3
    $MOVEI R2, '0
    ADD R3,R3,R2
    OUT Rdat_pant, R3
    $MOVEI R0, 0xF000
    OUT Rcon_pant, R0 ; mostrem el resultat per pantalla
    $MOVEI R1, tecla
bucle2:
    LDB R5, 0(R1)
    BZ R5, bucle2
    SUB R3, R3, R2
    MOVI R5, 16
    SUB R3, R5, R3
    $MOVEI R2, '0
    ADD R3,R3,R2
    OUT Rdat_imp, R3
    $MOVEI R0, 0x8001
    OUT Rcon_imp, R0 ; mostrem el resultat imprès
HALT

clock:
    $MOVEI R0, ticks
    LD R1, 0(R0)
    ADDI R1, R1, 1 ;incrementem una unitat de temps 1/10seg ticks++
    ST 0(R0), R1
    MOVI R2, 100
    $CMPGE R2, R1, R2
    BZ R2, fiCLK
    MOVI R5,1
    $MOVEI R0, final
    STB 0(R0), R5 ;final=1, true
fiCLK: JMP R6

teclat:
    IN R3, Rdat_tec
    $MOVEI R4, tecla
    STB 0(R4), R3
    JMP R6

```

3) Donat un processador d'arquitectura de 32 bits en un sistema amb memòria principal de 16 MB, amb una memòria cache directa de 16 KB, i de 4 words per bloc.

a) Donada una **direcció a memòria** originada des del processador ¿quants bits seran de byte offset, word offset, index i tag?

(1 punto)

Word: 32 bits → 4 bytes

Memòria Principal: 16 MB → Bus de d'adreces : 24 bits

Byte offset: 4 bytes/word → 2 bits

Word offset: 4 words/block → 2 bits

Cache 16 KB → Nro Bloques = 16 KBytes/(16 Bytes/Bloque) = 1 KBloques = 1024 Bloques

Index: $\log_2(1024) = 10$ → 10 bits

Tag: $24 - (2 + 2 + 10) = 10$ → 10 bits

b) Calcular el nombre total de bits emmagatzemats en aquesta memòria cache directa.

(1 punto)

$2^n \times (\text{tamaño de bloque} + \text{campo de tag} + \text{campo del bit de validez})$

$$2^{10}(16 \text{ Bytes} \times 8 \text{ bits/Byte} + 10 + 1) = 142336 \text{ bits}$$

$$142336 \text{ bits} = 17792 \text{ Bytes} = 17,375 \text{ KB}$$

c) Donada la següent seqüència de **@ de word** que genera el processador: 8208, 8206, 12303, 8211, 4098, 32769.

Omple les següents taules.

(3 puntos)

@ word	@bloque MP	Index	Hit/Miss
8208	$8208/4=2052$	$2052 \bmod 1024=4$	MISS
8206	$8206/4=2051$	$2051 \bmod 1024=3$	MISS
12303	$12303/4=3075$	$3075 \bmod 1024=3$	MISS
8211	$8211/4=2052$	$2052 \bmod 1024=4$	HIT
4098	$4098/4=1024$	$1024 \bmod 1024=0$	MISS
32769	$32769/4=8192$	$8192 \bmod 1024=0$	MISS
Tasa de Hits			1/6

Memoria Cache:

INDEX	V	TAG	Word3	Word2	Word1	Word0
0000000000	1	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1	M[32771] M[4099]	M[32770] M[4098]	M[32769] M[4097]	M[32768] M[4096]
0000000001	0					
0000000010	0					
0000000011	1	0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0	M[12303] M[8207]	M[12302] M[8206]	M[12301] M[8205]	M[12300] M[8204]
0000000100	1	0 0 0 0 0 0 0 0 1 0	M[8211]	M[8210]	M[8209]	M[8208]
1111111111						

2^24 16M	2^20 1M				2^16								2^10 1K											
@Mem	8388608	4194304	2E+06	1E+06	524288	262144	1E+05	65536	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
@Word	2097152	1048576	524288	262144	131072	65536	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1		
8208	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	
8206	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	
12303	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	1	
8211	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	
4098	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	
32769	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	