

**EXAMEN FINAL D'EC**  
**17 de juny de 2021**

- L'examen consta de 10 preguntes, que s'han de contestar als mateixos fulls de l'enunciat. No oblidis posar el teu nom i cognoms a tots els fulls.
- La durada de l'examen és de 3:00 hores (180 minuts)
- Les notes, la solució i el procediment de revisió es publicaran al Racó el dia 25 de juny.

**Pregunta 1 (1 punt)**

Un processador d'última generació disposa de 4 tipus d'instruccions diferents: A, B, C i D. La següent taula mostra quin és el nombre d'instruccions executades per un programa sota consideració i el CPI de cada tipus d'instrucció. La freqüència de rellotge del processador és de 2GHz i la potència dissipada és de 251W.

Tipus d'instrucció	#instr.	CPI
A	$4 \cdot 10^9$	1
B	$3 \cdot 10^9$	3
C	$2 \cdot 10^9$	1
D	$1 \cdot 10^9$	3

Calcula el CPI mitjà del programa sota consideració

CPI =

**1,8**

Indica quin és el temps d'execució en segons del programa

 $T_{\text{exe}}$  =**9**

s

Calcula l'energia consumida en Joules durant l'execució del del programa

E =

**2259**

J

Indica quin seria el guany (speed-up) que s'obtingria si s'aconseguís reduir el CPI de les instruccions de tipus B a 1 cicle.

guany =

**1,5****Pregunta 2 (0.5 punts)**

Quin serà el contingut final de `$t0` en hexadecimal després d'executar el següent codi?

```
li    $t0, -4
sra   $t1, $t0, 31
xor    $t0, $t0, $t1
subu   $t0, $t0, $t1
sll    $t1, $t1, 31
or     $t0, $t0, $t1
```

`$t0` =**0x80000004**

### Pregunta 3 (1 punt)

Donades les següents declaracions de variables globals en ensamblador MIPS, que s'ubiquen a memòria a partir de l'adreça 0x10010000:

```
.data
a: .ascii "cba"           # el codi ASCII de la 'a' és 0x61
b: .half -8
c: .word a
d: .byte 0x05
   .align 1
e: .byte 0xA5, 0xA4, 0xA3, 0xA2, 0xA1
f: .float 1.25
```

Omple la següent taula amb el contingut de memòria **en hexadecimal**. Les variables s'emmagatzemen a partir de l'adreça 0x10010000. Deixa EN BLANC les posicions no ocupades per cap dada.

@Memòria	Dada	@Memòria	Dada	@Memòria	Dada	@Memòria	Dada
0x10010000	63	0x10010008	00	0x10010010	A3	0x10010018	
0x10010001	62	0x10010009	00	0x10010011	A2	0x10010019	
0x10010002	61	0x1001000A	01	0x10010012	A1	0x1001001A	
0x10010003		0x1001000B	10	0x10010013		0x1001001B	
0x10010004	F8	0x1001000C	05	0x10010014	00	0x1001001C	
0x10010005	FF	0x1001000D		0x10010015	00	0x1001001D	
0x10010006		0x1001000E	A5	0x10010016	A0	0x1001001E	
0x10010007		0x1001000F	A4	0x10010017	3F	0x1001001F	

### Pregunta 4 (1 punt)

Posa una X al costat de cada una de les següents afirmacions (a la columna V si és Verdadera o a la columna F si és Falsa). Suposem en tots els casos que es fa referència a un processador MIPS com l'estudiad a classe. Cada resposta correcta suma 0,1 punts; les respostes no contestades no es tenen en compte; cada resposta incorrecta resta 0,1 punts; i la puntuació total mínima és 0.

Afirmació	V	F
La traducció d'adreces virtuals a físiques en una instrucció <code>lw</code> pot arribar a produir fins a quatre excepcions.	X	
En un computador que disposa de memòria virtual amb TLB sempre cal fer dos accessos a memòria principal per cada referència: un a la Taula de Pàgines i l'altre a l'adreça de la referència.		X
Les fallades de TLB causen una excepció i es tracten per software. No obstant, les fallades de TLB invoquen la rutina <code>TLBmiss</code> , molt més curta d'executar que la rutina d'excepcions genèrica RSE.	X	
El TLB funciona com una memòria cache de la Taula de Pàgines, les etiquetes de la qual consisteixen en el número de pàgina virtual (VPN).	X	
En un processador amb adreces de 32 bits, una memòria cache associativa de 4 vies, de 32KB i blocs de 32 bytes, s'han de dedicar 20 bits a l'etiqueta, 7 al número de conjunt i 5 al desplaçament.		X
La taula de pàgines es pot modificar quan el bit EXL, del registre Status, val 0.		X
Després de servir una interrupció, de vegades, tornem a reexecutar la mateixa instrucció durant la qual ha arribat aquesta interrupció.		X
Podem desactivar les excepcions per <i>overflow</i> posant a 0 el bit corresponent a aquestes excepcions del camp IM del registre Status.		X
Les interrupcions es gestionen com un tipus concret d'excepció, i permeten al sistema operatiu interaccionar amb els dispositius externs de forma asíncrona.	X	
La instrucció <code>tlbwr</code> implementa la política de reemplaçament LRU pel TLB.		X

**Pregunta 5 (0,75 punts)**

Donat el següent fragment de codi en llenguatge C

```
unsigned int x,y;

if (( x > y ) && ( y >= 10 )) {
    y++;
    x = x / 4;
} else {
    x = y;
}
```

L'hem traduït a ensamblador MIPS sense fer servir macros o pseudoinstruccions. Completa els següents requadres amb els corresponents mnemònics i operands a fi que la traducció sigui correcta.

```

    sltu      $t3, $t1, $t0
    beq      $t3, $zero, else
    addiu     $t7, $zero, 10
    sltu      $t3, $t1 , $t7
    bne       $t3 , $zero , else
then:
    addiu     $t1, $t1, 1
    srl       $t0, $t0, 2
    beq       $zero, $zero, endif
else:
    addu      $t0, $t1, $zero
endif:
```

**Pregunta 6 (1 punt)**

Escriu un codi en ensamblador de MIPS (màxim 4 instruccions) que multipliqui els valors de \$t0 i \$t1 (ambdós naturals) de forma que \$t3 tingui el resultat de la multiplicació i \$t7 valgui 0 si hi ha hagut sobreiximent, i 1 altrament.

```

multu $t1, $t0
mflo  $t3
mfhi  $t7
sltiu $t7, $t7, 1
```

### Pregunta 7 (1 punt)

Donat el següent fragment de codi en llenguatge C

```
int    vec[N];
short ind[N][N];

int i, sum = 0;
for (i=0; i<N; i++)
    sum += vec[ ind[i][i] ] + vec[ ind[N-i-1][2] ];
```

L'hem traduït a ensamblador MIPS. Completa els següents requadres amb valors (en funció d' $N$  si cal) a fi que la traducció sigui correcta.

```
li      $t0, 0          # i
li      $t7, N
li      $v0, 0          # sum

la      $t1, vec
la      $t2, ind
addiu   $t3, $t2, 

loop:
    beq      $t0, $t7, end_loop

    lh      $t5, 0($t2)
    sll     $t5, $t5, 
    addu    $t5, $t5, $t1
    lw      $t5, 0($t5)
    addu    $v0, $v0, $t5

    lh      $t5, 0($t3)
    sll     $t5, $t5, 
    addu    $t5, $t5, $t1
    lw      $t5, 0($t5)
    addu    $v0, $v0, $t5

    addiu   $t2, $t2, 
    addiu   $t3, $t3, 
    addiu   $t0, $t0, 
    b       loop

end_loop:
```

**Pregunta 8 (1.25 punts)**

Tenim un processador MIPS amb un co-processador de coma flotant, on els registres \$f4 i \$f6 contenen els valors 0x4136DB6D i 0xBFB6DB6B, respectivament. En aquest processador, s'executa la instrucció MIPS add.s \$f0, \$f4, \$f6. La unitat de suma i resta utilitza els tres bits GRS de guarda i arrodoneix al més pròxim. Contesta les següents preguntes:

Quina és la mantissa (en binari) i l'exponent (en decimal) dels nombres que hi ha a \$f4 i \$f6?

	mantissa (binari)	exponent (decimal)
\$f4:	1, 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1	3
\$f6:	1, 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1	0

Omple les següents caselles mostrant l'operació (+/-), els nombres a operar, els bits de guarda i el resultat:

	mantissa (binari)	G R S
op	1, 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1	0 0 0
-	0, 0 0 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1	0 1 1
	1, 0 0 1	1 0 1

Omple el resultat després de re-normalitzar i arrodonir:

	mantissa (binari)	exponent (decimal)
	1, 0 1 0	3

Quin és el contingut de \$f0 en hexadecimal després d'executar la instrucció?

\$f0 = 0x41200000

Quin valor decimal representa \$f0 després d'executar la instrucció?

\$f0 = 10,0

**Pregunta 9 (1.25 punts)**

Considera un processador amb adreces de 32 bits i una memòria cache de dades amb les següents característiques:

- Capacitat: 1kB
- Mida del bloc: 64 bytes
- Correspondència associativa per conjunts
- Blocs per conjunt: 2
- Política de reemplaçament LRU
- Escriptura retardada amb assignació (write-back, write-allocate)

Suposem que la memòria cache és inicialment buida, i executem el següent programa al nostre processador:

```
int V[128];
int M[128][16];

void main() {
    int i,j,tmp;           // a $t0, $t1 i $t2 respectivament
    for (i=0; i<128; i++)
        for (j=0; j<16; j++) {
            tmp = M[i][j];
            V[i] = V[i] + tmp;
        }
}
```

Considera addicionalment que el vector  $V$  comença a partir de l'adreça 0x10010000 i que tant el vector  $V$  com la matriu  $M$  estan convenientment inicialitzats, tot i que no s'ha de tenir en compte aquesta inicialització en la resolució de l'exercici.

Quin serà el número del bloc de MP on es troba  $M[0][0]$ , en hexadecimal?

Bloc MP  $M[0][0]$  = **0x0400408**

Quantes referències a memòria per dades es realitzaran durant l'execució del programa, i quantes fallades de cache es produiran?

Referències = **6144**

Fallades = **136**

Decidim experimentar canviant aquesta cache per una altra amb les mateixes característiques (1kB, blocs 64 bytes) però amb una política de correspondència directa.

Quina serà ara la quantitat de fallades que s'obtindrà en l'execució d'aquest codi?

Fallades = **384**

### Pregunta 10 (1,25 punts)

Considera un processador MIPS amb un sistema de memòria virtual paginada. Les pàgines són de 4kB de mida i la política de reemplaçament és LRU. El sistema operatiu determina que la memòria física només mantindrà 4 pàgines per programa.

Per simplificar el problema només es consideren els accessos a dades, descartant la gestió dels accessos a instruccions.

L'estat de la taula de pàgines en un moment donat de l'execució d'un programa és el següent (només mostrem les entrades amb  $P=1$ ):

VPN	PPN	P	D
0x10010	0x0	1	1
0x10020	0x3	1	1
0x1007F	0x2	1	0
0x10080	0x1	1	0

El sistema està complementat per un TLB completament associatiu de dues entrades, amb reemplaçament LRU. L'ordre de les últimes referències a memòria és el següent: 0x10020 (accés més llunyà), 0x1007F, 0x10080, 0x10010 (accés més recent).

Considera la seqüència d'accessos a memòria de la taula següent. Per a cada accés, on s'indica l'adreça de l'accés en hexadecimal i si és Lectura (L) o Escriptura (E), omple el seu número de pàgina lògica (VPN), si provoca una fallada de TLB, si provoca una fallada de pàgina, si s'ha d'escriure a disc, quina pàgina lògica es reemplaça i quin és el PPN resultat de la traducció d'adreces. A les columnes amb preguntes booleanes escriu "SI" o "NO". A les columnes numèriques escriu la dada en hexadecimal.

L/E	Adreça (hex)	VPN (hex)	Fallada TLB?	Fallada Pàgina?	Escript. Disc?	VPN Reempl. (hex)	PPN (hex)
E	10080874	<b>10080</b>	<b>NO</b>	<b>NO</b>	<b>NO</b>	<b>-</b>	<b>1</b>
L	1009077C	<b>10090</b>	<b>SI</b>	<b>SI</b>	<b>SI</b>	<b>10020</b>	<b>3</b>
E	10020AA0	<b>10020</b>	<b>SI</b>	<b>SI</b>	<b>NO</b>	<b>1007F</b>	<b>2</b>
L	1007EFFF	<b>1007E</b>	<b>SI</b>	<b>SI</b>	<b>SI</b>	<b>10010</b>	<b>0</b>
L	10010998	<b>10010</b>	<b>SI</b>	<b>SI</b>	<b>SI</b>	<b>10080</b>	<b>1</b>