

COGNOMS:**GRUP:****NOM:****EXAMEN FINAL D'EC****13 de gener de 2016**

L'examen consta de 8 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 duració de l'examen és de 180 minuts. Les notes i la solució es publicaran al Racó el dia 21 de gener. La revisió es farà el 22 de gener a les 17:00h.

Pregunta 1. (1,20 punts)

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

```

                .data
a:      .asciiz      "ABCD"          # codi ASCII 'A' = 0x41
b:      .half        127
c:      .byte        -128
d:      .word        a, a+2
e:      .dword       1

```

- a) **(0,30 punts)** Completa la següent taula amb el contingut de la memòria, indicant el valor de cada byte EN HEXADECIMAL, i deixant EN BLANC les posicions no ocupades per cap dada.

@Memòria	Dada	@Memòria	Dada	@Memòria	Dada	@Memòria	Dada
0x10010000		0x10010010		0x10010020		0x10010030	
0x10010001		0x10010011		0x10010021		0x10010031	
0x10010002		0x10010012		0x10010022		0x10010032	
0x10010003		0x10010013		0x10010023		0x10010033	
0x10010004		0x10010014		0x10010024		0x10010034	
0x10010005		0x10010015		0x10010025		0x10010035	
0x10010006		0x10010016		0x10010026		0x10010036	
0x10010007		0x10010017		0x10010027		0x10010037	
0x10010008		0x10010018		0x10010028		0x10010038	
0x10010009		0x10010019		0x10010029		0x10010039	
0x1001000A		0x1001001A		0x1001002A		0x1001003A	
0x1001000B		0x1001001B		0x1001002B		0x1001003B	
0x1001000C		0x1001001C		0x1001002C		0x1001003C	
0x1001000D		0x1001001D		0x1001002D		0x1001003D	
0x1001000E		0x1001001E		0x1001002E		0x1001003E	
0x1001000F		0x1001001F		0x1001002F		0x1001003F	

- b) (0,30 punts)** Indica quin serà el contingut final de \$t0 en hexadecimal després d'executar el següent codi:

```
la    $t0, a
lb    $t0, 1($t0)
ori    $t0, $t0, 0xAB00
andi    $t0, $t0, 0xFF0
```

\$t0 =

- c) (0,30 punts)** Si la variable b correspon a la traducció de la següent declaració en alt nivell
unsigned short b;

Tradueix la següent sentència a codi MIPS:

```
b = b*256;
```

- d) (0,30 punts)** Si la variable e correspon a la traducció de la següent declaració en alt nivell
long long e;

Completa el codi MIPS que hi ha a continuació que correspon a la traducció de
e++;

```
la    $t0, e
lw    $t1, 0($t0)
lw    $t2, 4($t0)
```

```
sw    $t1, 0($t0)
sw    $t2, 4($t0)
```

GRUP:

Pregunta 2. (1,25 punts)

Quina és la mantissa (en binari) i l'exponent (en decimal) d'aquests dos nombres?

	mantissa (binari)	exponent (decimal)
\$f2:	<div style="border: 1px solid black; display: inline-block; width: 18px; height: 18px; position: relative;"><div style="position: absolute; left: -6px; top: 50%; transform: translateY(-50%);">.</div></div> <div style="border: 1px solid black; display: inline-block; width: 170px; height: 18px;"></div>	<div style="border: 1px solid black; display: inline-block; width: 40px; height: 18px;"></div>
\$f4:	<div style="border: 1px solid black; display: inline-block; width: 18px; height: 18px; position: relative;"><div style="position: absolute; left: -6px; top: 50%; transform: translateY(-50%);">.</div></div> <div style="border: 1px solid black; display: inline-block; width: 170px; height: 18px;"></div>	<div style="border: 1px solid black; display: inline-block; width: 40px; height: 18px;"></div>

op		G	R	S

estat després de re-normalitzar (si cal):

	G	R	S	exponent (decimal)

[illegible]

\$f0 = 0x

\$f_0\$ (decimal) =

Pregunta 3. (0,80 punts)

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'estudiat a classe. Cada resposta correcta suma 0,08 punts; les respostes no contestades no es tenen en compte; cada resposta incorrecta resta 0.08 punts; i la puntuació total mínima és 0.

	Afirmació	V	F
1.-	La diferència entre les instruccions MIPS <code>add</code> / <code>addu</code> radica en què la instrucció <code>add</code> pot generar una excepció per sobreiximent (overflow) a la suma d'enters mentre que la instrucció <code>addu</code> mai pot generar cap excepció.		
2.-	En un sistema computador basat en el MIPS qualsevol programa pot inhibir les interrupcions, posant a 1 el bit <code>ExL</code> (bit 1) del registre <code>Status</code> (registre \$12 del coprocessador 0) amb el codi: <pre> mfc0 \$t0, \$12 ori \$t0, 2 mtc0 \$t0, \$12 </pre>		
3.-	En un sistema computador basat en el MIPS pot passar que al final de l'execució d'algunes instruccions el bit <code>D</code> del TLB per a alguna VPN no sigui coherent amb el bit <code>D</code> de la Taula de Pàgines associat a la mateixa VPN.		
4.-	La instrucció <code>tlbwr</code> del MIPS escriu el contingut d'una entrada de la Taula de Pàgines al TLB, en aquella entrada del TLB que fa més temps que no s'accedeix en aplicació de l'algorisme LRU.		
5.-	La rutina <code>TLBmiss</code> estudiada al tema 8 s'executa quan hi ha una fallada de pàgina.		
6.-	L'excepció per accés no alineat a memòria pot ser inhibida a través del camp <code>Interrupt Mask</code> .		
7.-	En el MIPS, una mateixa instrucció pot causar durant la seva execució 2 fallades de pàgina.		
8.-	La RSE estudiada al tema 8 per al processador MIPS s'executa com a tractament de totes les excepcions.		
9.-	El temps mitjà d'accés a memòria de les instruccions <code>load</code> i <code>store</code> en un computador que disposa de memòria virtual amb TLB i memòria cache, és inferior al temps d'accés de la memòria principal.		
10.-	4. Al processador MIPS hi ha certes adreces que si s'accedeixen en mode usuari (<code>ExL=0</code>) provoquen una excepció mentre que són accessibles normalment en mode sistema (<code>ExL=1</code>).		

COGNOMS:**GRUP:****NOM:****Pregunta 4. (0,75 punts)**

Un processador disposa de 4 tipus d'instruccions diferents: A, B, C i D. La següent taula mostra quin és el número d'instruccions executades per a un programa sota consideració i el CPI de cada tipus d'instrucció. El processador té un rellotge a 2GHz.

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

a) (0,25 punts) Calcula el CPI mitjà del programa sota consideració.

CPI =

b) (0,25 punts) Indica quin és el temps d'execució (en segons) del programa sota consideració.

t_{exe} =

c) (0,25 punts) Indica quin seria el guany (speed-up) obtingut si s'aconseguís reduir el CPI de les instruccions de **tipus B** a 1 cicle.

guany =

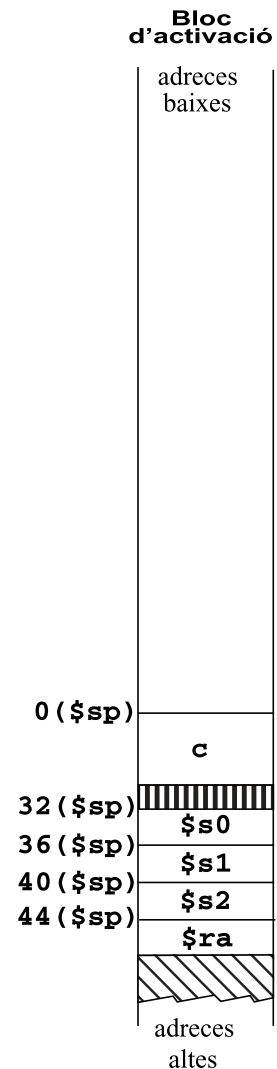
Pregunta 5. (1,75 punts)

Donades les següents declaracions en C:

```
short exam2(short *x, short *y);
void exam1(short a[], int t) {
    short c[15];
    int i;
    for (i=0; i<t; i++) {
        c[i] = exam2(a, &c[t-i]);
    }
}
```

Seguint les regles de l'ABI estudiades, determina quines variables, paràmetres o càlculs intermedis de la funció `exam1` cal guardar obligatòriament en registres segurs (\$s) per garantir que no siguin alterats per les crides a `exam2`:

Tradueix a MIPS la subrutina `exam1`, i dibuixa el bloc d'activació, especificant-hi la posició on apunta el registre `$sp` un cop reservat l'espai corresponent a la pila, així com el nom de cada un dels elements que conté, i la seva posició (desplaçament relatiu al `$sp`).



COGNOMS:**GRUP:****NOM:****Pregunta 6. (1 punt)**

Sigui el següent fragment de programa en C, on hi ha una sentència **E** desconeguda:

```

short M[20][10];
main() {
    int i;                /* es guarda en $t0 */
    ...
    for (i=0; i<10; i=i+2)
        E ;
}

```

Hem traduït el bucle anterior a MIPS, usant la tècnica d'accés seqüencial i hem obtingut el següent codi, on hi falten els requadres **A** i **B**, ja que depenen de quina sigui la sentència **E**:

```

        .data
M:      .space 400
        .text
main:   . . .
        la      $t2, A           # inicialitzem el punter
        li      $t0, 0           # i=0
        li      $t1, 10
for:
        bge     $t0, $t1, fi
        sh      $zero, 0($t2)    # sentència E
        addiu   $t2, $t2, B
        addiu   $t0, $t0, 2      # i = i + 2
        b       for
fi:     . . .

```

Omple la taula següent indicant els operands que cal escriure als requadres **A** i **B** perquè el programa en MIPS sigui la traducció correcta del programa en C, per a cada una de les possibles sentències d'**E** (o, si no és possible aplicar accés seqüencial, escriu "N/A"):

	A	B
M[9-i][1] = 0		
M[i][i] = 0		
M[2*i][1] = 0		
M[1][i*i] = 0		

Pregunta 7. (1,50 punts)

Disposen d'un computador que gestiona memòria virtual paginada amb pàgines de 1MB. El processador permet adreçar fins a 4 GB però només pot tenir 64MB de memòria física. El sistema operatiu limita a 4 el nombre de marcs de pàgina disponible per cada procés i la seva política de reemplaçament és LRU.

- a) **(0,50 punts)** Quants bytes ocuparà la taula de pàgines tenint en compte que per cada entrada cal que tingui un bit de presència (P) i un bit de pàgina modificada (D) a part del número de pàgina física?

Durant l'execució d'un cert programa la seva taula de pàgines és en el següent estat:

VPN (hex)	PPN (hex)	P	D
...			
078	2E	1	0
...			
15B	3F	1	1
...			
183	25	1	0
...			
1F6	20	1	1
...			

Els darrers accessos que ha realitzat aquest programa han estat en el següent ordre de VPNs: 0x078, 0x1F6, 0x15B i 0x183.

- b) **(1 punt)** Omple la següent taula per a cada referència a memòria que es realitza posteriorment a l'estat indicat anteriorment.

adr. lògica (hex)	VPN (hex)	fallada de pàgina? (SI/NO)	lectura de disc (SI/NO)	escrip-tura a disc (SI/NO)	VPN pàgina reemplaçada (hex)	PPN (hex)
E 0x07712345						
E 0x16C43232						
L 0x16C23246						
L 0x18321221						
L 0x07812345						
E 0x07743232						

COGNOMS:

GRUP:

NOM:

Pregunta 8. (1,75 punts)

Disposen d'un processador de 32 bits (tant per adreces com per a dades) amb memòria principal i dues memòries cache associatives per conjunts, una per a instruccions (MCI) i una altra per a dades (MCD), amb els següents paràmetres:

- Freqüència del rellotge: 100 MHz
- Mida de l'espai d'adreçament: 4 GB
- Capacitat de la MCD: 256 KB
- Capacitat de la MCI: 512 KB
- Blocs per conjunt, tant a MCI com a MCD: 4 blocs
- Mida de bloc: 16 paraules
- Temps d'accés a memòria cache (MCI i MCD) en cas d'encert: 2 cicles
- Temps d'accés a memòria principal per llegir/escriure blocs: 25 cicles
- Temps d'accés a memòria principal per llegir/escriure paraules: 10 cicles
- MCD gestionada amb una política d'escriptura retardada amb assignació

Quan s'executen un conjunt de programes representatius (benchmark) observem que:

- CPI_{ideal} : 2,2
- Taxa d'encerts en MCI: 90%
- Taxa d'encerts en MCD: 80%
- De cada 4 blocs reemplaçats a MCD se n'ha modificat 1
- En mitjana, les instruccions que s'executen són
 - aritmètico-lògiques: 35%
 - moviment de dades (immediats i registres): 20%
 - salt: 15%
 - load: 15%
 - comparació: 10%
 - store: 5%

a) (0,25 punts) Quants bits ocuparà l'etiqueta (tag) dels blocs emmagatzemats a MCD?

b) (0,50 punts) En mitjana, quants cicles per instrucció de penalització/retard introdueixen les fallades a la MCI?

- c) **(0,50 punts)** En mitjana, quants cicles per instrucció de penalització/retard introdueixen les fallades a la MCD?

- d) **(0,25 punts)** Quin és el CPI d'aquest sistema processador-memòria obtingut en executar el benchmark?

- e) **(0,25 punts)** Si ara la cache de dades MCD es gestionés amb una política d'escriptura immediata sense assignació, quin seria el temps mitjà d'accés a dades en memòria, en nanosegons?