

EXAMEN PARCIAL D'EC
5 d'abril de 2022

- L'examen consta de 6 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 1:30 hores (90 minuts)
- Les notes i la solució es publicaran al Racó el dia 19 d'abril. La revisió es farà presencialment el 21 d'abril a les 9:00h.

Pregunta 1 (1,5 punts)

Un processador ha estat dissenyat per poder funcionar correctament a les següents combinacions de freqüències i voltatges d'alimentació:

	Voltatge (V)	Freqüència (GHz)
A	2,0	1,25
B	2,3	2

Sabent que la potència dinàmica de la combinació A és de $P_A = 80W$, es demana que contestis les següents preguntes:

Quina és la potència dinàmica dissipada per la combinació B en watts?

$$P = \alpha \cdot C \cdot V^2 \cdot f = 80 / (2^2 \cdot 1,25 \cdot 10^9) \cdot 2,3^2 \cdot 2 \cdot 10^9 = 169,28 \text{ W}$$

Quin és el guany de rendiment (o speedup) que s'obté amb la combinació B respecte de la combinació A, executant el mateix programa?

$$\text{Guany} = t_{\text{exeA}} / t_{\text{exeB}} = f_B / f_A = 1,6$$

Amb la combinació A, quin és el temps d'execució (en segons) d'un programa que executa $1,2 \cdot 10^{10}$ instruccions i que té un CPI promig de 4?

$$t_{\text{exeA}} = n \cdot \text{CPI} / f_A = 38,4 \text{ s}$$

Pregunta 2 (1,5 punts)

Donada la següent declaració de variables globals d'un programa escrit en llenguatge C:

```
short a[3] = {-6, 8, 3};
unsigned int b;
long long c = -8;
char d[] = {0xC, 0xA, 0xF, 0xE};
char e[] = "CAFE";
short *f = &a[1];
```

Tradueix-la a llenguatge ensamblador MIPS

```
.data
a: .half -6, 8, 3
b: .word 0
c: .dword -8
d: .byte 0xC, 0xA, 0xF, 0xE
e: .asciiz "CAFE"
f: .word a+2
```

Omple la següent taula amb el contingut de memòria **en hexadecimal** (sense el prefix 0x). Tingues en compte que el codi ASCII de la 'A' és el 0x41. Les variables s'emmagatzemen a partir de l'adreça 0x10010000. Les posicions de memòria sense inicialitzar es deixen en blanc.

@Memòria	Dada	@Memòria	Dada	@Memòria	Dada	@Memòria	Dada	@Memòria	Dada
0x10010000	FA	0x10010008	00	0x10010010	F8	0x10010018	0C	0x10010020	00
0x10010001	FF	0x10010009	00	0x10010011	FF	0x10010019	0A	0x10010021	
0x10010002	08	0x1001000A	00	0x10010012	FF	0x1001001A	0F	0x10010022	
0x10010003	00	0x1001000B	00	0x10010013	FF	0x1001001B	0E	0x10010023	
0x10010004	03	0x1001000C		0x10010014	FF	0x1001001C	43	0x10010024	02
0x10010005	00	0x1001000D		0x10010015	FF	0x1001001D	41	0x10010025	00
0x10010006		0x1001000E		0x10010016	FF	0x1001001E	46	0x10010026	01
0x10010007		0x1001000F		0x10010017	FF	0x1001001F	45	0x10010027	10

Calcula el valor final del registre \$t0 en hexadecimal després d'executar el següent codi.

```
la    $t1, d
lh    $t1, 2($t1)
sra   $t1, $t1, 8
addiu $t1, $t1, -12
lui   $t2, 0x1001
or    $t2, $t2, $t1
lb    $t0, -1($t2)
```

\$t0 = **0xFFFFFFFF**

Pregunta 3 (1,5 punts)

Donada la següent funció escrita en alt nivell en C:

```
int func(int x, unsigned int y) {
    if ((y > 0) && (x!=0))
        x = 0;
    else if ((~x ^ 0x1111) != 0)
        x++;
    return x;
}
```

Completa el següent fragment de codi en MIPS, que tradueix l'anterior funció, escrivint en cada calaix un mnemònic d'instrucció o macro, etiqueta, registre o immediat.

	bleu / beq	\$a1, \$zero,	et4
et1:	beq	\$a0, \$zero,	et4
et2:	move	\$a0, \$zero	
et3:	b	et8	
et4:	nor	\$t2, \$a0,	\$zero / \$a0
et5:	xori	\$t2, \$t2,	0x1111
et6:	beq	\$t2, \$zero, et8	
et7:	addiu	\$a0, \$a0, 1	
et8:	move	\$v0, \$a0	

Pregunta 4 (2 punts)

Donat el següent programa en llenguatge C:

```
int A[10][10];
int B[100][8];

main() {
    int i;                                /* ocupa el registre $t0 */
    for (i=0; i<10; i=i+1){
        B[i*i][5]=A[9-i][i];
    }
}
```

Considerant que les variables globals ja estan declarades, completa el següent codi MIPS omplint les caselles en blanc perquè sigui equivalent a l'anterior codi en alt nivell. Tingues en compte que els elements de la matriu A s'accedeixen utilitzant la tècnica d'accés seqüencial.

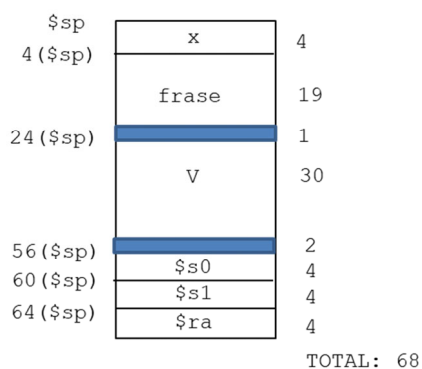
```
main: li    $t0, 0                      # i = 0
      la    $t1, A + 
      la    $t2, B
      li    $t3, 10
      b     cond
do:   lw    $t4, 0($t1)
      mult  $t0, $t0
      mflo  $t5
      sll   $t5, $t5, 
      addu  $t5, $t2, $t5
      sw    $t4, ($t5)
      addiu $t1, $t1, 
      addiu $t0, $t0, 1
cond: blt   $t0, $t3, do
      jr    $ra
```

Pregunta 5 (2 punts)

Donat el codi següent:

```
int examen(char c, char W[], int k) {
    int i, *j, x;
    char frase[19];
    short V[15]
    ...
}
```

Dibuixa el bloc d'activació de la rutina examen, indicant clarament la mida i el desplaçament necessari per accedir a cada element, sabent que utilitzarem els registres segurs \$s0, \$s1 i \$ra i que durant l'execució de la rutina s'utilitzarà &x.



Escriu les 4 primeres instruccions de la rutina examen.

```
examen:    addiu    $sp, $sp, -68
           sw       $s0, 56($sp)
           sw       $s1, 60($sp)
           sw       $ra, 64($sp)
```

Tradueix la següent sentència:

```
frase[*j] = W[k];
```

suposant que està dins la rutina examen, i que les variables locals i i j estan als registres \$s0 i \$s1 respectivament.

```
addu      $t0, $a1, $a2
lb        $t0, 0($t0)
lw        $t1, 0($s1)
addu      $t1, $t1, $sp
sb        $t0, 4($t1)
```

Pregunta 6 (1,5 punts)

Donat el següent fragment de codi, en C

```
int func() {  
    long long x, y;  
    ...  
    return (x < y);  
}
```

Ompler els quadres per tal que el següent fragment de codi MIPS sigui la traducció de la sentència visible de la funció `func()`.

Suposem que:

`x` està guardat en `$t1` (part alta) i `$t0` (part baixa)

`y` està guardat en `$t3` (part alta) i `$t2` (part baixa)

`slt`

`$v0,`

`$t1`

`,`

`$t3`

`bne`

`$v0, $zero, fi`

`bgt`

`$t1, $t3, fi`

`sltu`

`$v0, $t0, $t2`

`fi:`