

EXAMEN PARCIAL D'EC
29 de abril de 2020

INSTRUCCIONS IMPORTANTS:

- La durada de l'examen és de 120 minuts.
- L'examen consta de 8 preguntes, que s'han de contestar al fitxer de respostes (el fitxer de text respostes.txt que acompanya a aquest enunciat).
- A la capçalera del fitxer de respostes posa el teu nom i cognoms. No canviïs el nom del fitxer de respostes.
- El fitxer de respostes s'ha de pujar al Racó, al mateix lloc des d'on has descarregat aquest enunciat (dins l'apartat de Pràctiques i l'apartat habilitat per a l'examen parcial del grup 50).
- Pots pujar el fitxer múltiples cops, no esperis a l'últim minut.

Pregunta 1. (1,5 punts)

El processador Intel de última generació disposa de 5 tipus d'instruccions diferents: Comparació, Aritmètico-lògiques, Moviment de dades, Memòria i Salt.

Després de realitzar un benchmark (conjunt d'instruccions representatives), s'han obtingut els següents percentatges d'instruccions executades de cada tipus i el CPI associat a cadascuna d'elles.

Tipus d' instrucció	% de execució	CPI
Comparació	15%	1
Aritmètico-lògiques,	30%	4
Moviment de dades	30%	25
Memòria	20%	5
Salt	5%	2

a) (0,5 punts) Calcula el CPI mitjà d'aquest processador per al conjunt d'instruccions donat.

b) (1 punt) Es pot aconseguir un guany (speed-up) d'1,5 en el temps d'execució del conjunt representatiu de programes (benchmark) a partir de la millora de les instruccions aritmètico-lògiques y de salt? Raona la resposta. En cas afirmatiu, indica quin ha de ser el nou CPI d'aquest tipus d'instrucció.

Pregunta 2. (1,25 punts)

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

```
.data
w:  .half    -17, -2, 5
x:  .word     w
y:  .half    0x0A34, 0xBFA7, 0x12C9
z:  .asciiz   "zyxw"           # El codi ascii de la 'z' és 0x7A
```

a) (0,25 punts) Omple 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
0x10010000	
0x10010001	
0x10010002	
0x10010003	
0x10010004	
0x10010005	
0x10010006	
0x10010007	
0x10010008	
0x10010009	
0x1001000A	

@Memòria	Dada
0x1001000B	
0x1001000C	
0x1001000D	
0x1001000E	
0x1001000F	
0x10010010	
0x10010011	
0x10010012	
0x10010013	
0x10010014	
0x10010015	

b) (0,50 punts) Quin és el valor final de \$t0 en hexadecimal després d'executar el següent fragment de codi?

```
la      $t0, y
lw      $t1, 0($t0)
sb      $t1, 3($t0)
lh      $t0, 2($t0)
```

\$t0 =

c) (0,50 punts) Quin és el valor final de \$t0 en hexadecimal després d'executar el següent fragment de codi?

```
la      $t0, x
lb      $t0, 3($t0)
sll     $t1, $t0, 4
and     $t0, $t0, $t1
addu    $t0, $t0, $t1
```

\$t0 =

Pregunta 3. (0,5 punts)

a) (0,5 punts) Utilitzant únicament 3 instruccions, escriu un programa en MIPS que vagi intercanviar els 4 bits de menys pes amb els 4 bits de més pes de l registre \$ s5

Pregunta 4. (2.25 punts)

Sigui el següent fragment de programa en C:

```
int mat[500][500];
main() {
    int i;        /* guardat al registre $t0 */
    int j;        /* guardat al registre $t1 */
    . . . * sentències dels apartats *
}
```

a) (0,75 punts) Tradueix a ensamblador MIPS la següent sentència, suposant que i ocupa el registre \$t0 i j el registre \$t1 :

```
if (((j>0) && (j+17<500)) || (i>=0))
    j = 1;
else
    j = -100;
```

b) (0,75 punts) Tradueix a ensamblador MIPS la següent sentència amb el menor nombre possible d'instruccions (es pot fer en 6 línies), suposant que i ocupa el registre \$t0 i j el registre \$t1 .

```
j = mat[i+2][499-i];
```

c) (0,75 punt) Tradueix a ensamblador MIPS la següent sentència. **Utilitza la tècnica d'accés seqüencial** per accedir a la matriu. Suposa que i ocupa el registre \$t0 i j el registre \$t1 .

```
for (i=4; i<j; i++)
    mat[i*2][j] = i;
```

Pregunta 5. (2,25 punts)

Donada la següent declaració de funcions en C:

```
int *shift(int x, int y);

int my_subroutine(int *r)
{
    int i;
    int *pointer;

    pointer = r+5;

    for (i=3; i>=0; i--)
    {
        *r = *pointer;
        pointer = shift(i, *r);
    }
    return *pointer;
}
```

a) (0,25 punts) Examina el codi de la subrutina *my_subroutine* i determina el número mínim de registres que s'hauran de salvar a la pila durant l'execució.

Número de registres

b) (2 punts) Tradueix a ensamblador MIPS la rutina *my_subroutine*

Pregunta 6. (0,5 punts)

Donada la següent sentència escrita en ensamblador MIPS:

a: float 43.00008

a) (0,5 punt) Aproxima el valor de la variable "a" com a nombre binari normalitzat en notació científica amb 8 bits significatius.

Pregunta 7. (1 punts)

a) (1 punts) Considereu dos números flotants normalitzats en coma flotant float1 i float2 que es representen com a flotants de precisió IEEE de 32 bits (simple precisió). Això defineix dues strings de 32 bits. Suppose float_1 < float_2. Si les dues cadenes de 32 bits fossin tractades como unsigned integers (denominats Uint1 i Uint2). Dit això ¿Es podria concloure que Uint1 < Uint2? Raona la teva resposta

Pregunta 8 (0,75 punts) L'estàndard de coma flotant IEEE-754 de precisió simple (32 bits) ens permet representar menys de 2^{32} nombres diferents. D'aquests nombres possibles:

a) (0,25 punts) Quants són estrictament entre 2^{-5} i 2^{-4} ?

b) (0,25 punts) Quants són estrictament entre 2^{13} i 2^{14} ?

c) (0,25 punts) Quants són estrictament entre 2^{47} i 2^{48} ?