

Examen Parcial d'EC 29 d'abril de 2020

INSTRUCCIONS IMPORTANTS:

- La durada de l'examen és de 120 minuts.
- L'examen consta de 7 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 10).
- Pots pujar el fitxer múltiples cops, no esperis a l'últim minut.

Pregunta 1 (2 punts)

Donada la següent declaració de variables globals, que s'ubica a memòria a partir de l'adreça 0x10010000:

```
.data
v1: .byte    -3
v2: .half    0x21E
v3: .dword   -1
v4: .asciiz  "012" # codi ASCII '0' =
0x30 v5: .float -2.5
v6: .word    129
v7: .word    v1
```

- a) Omple la següent taula amb el contingut de memòria **en hexadecimal**. 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
0x10010000		0x10010008		0x10010011		0x10010018	
0x10010001		0x10010009		0x10010012		0x10010019	
0x10010002		0x1001000A		0x10010013		0x1001001A	
0x10010003		0x1001000B		0x10010013		0x1001001B	
0x10010004		0x1001000C		0x10010014		0x1001001C	
0x10010005		0x1001000D		0x10010015		0x1001001D	
0x10010006		0x1001000E		0x10010016		0x1001001E	
0x10010007		0x1001000F		0x10010017		0x1001001F	

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

```
la $t1, v1
lb $t1, 0($t1)
li $t2,
0x0F000000 xor
$t0, $t1, $t2

$t0 = 0x
```

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

```
li    $t0, 10247
li    $t1, 10
div   $t0,
$t1 mflo $t0
mfhi  $t1
addu  $t0, $t0, $t1

$t0 = 0x
```

Pregunta 2 (0,75 punts)

Considera la següent subrutina programada en ensamblador MIPS:

```
func:
    ble $a0, $a2,
    etiq1 ble $a1,
    $a2, etiq2
etiq1: beq $a2, $zero,
etiq3 etiq2: li    $v0,
0
    b etiq4
etiq3: li
$v0,1 etiq4:
    jr $ra
```

Completa el següent codi C omplint les caselles en blanc perquè sigui equivalent a l'anterior codi MIPS:

```
int func(int a, int b, int c)
{
    int res;
    if ( ( ( a > c ) && ( b <= c ) ) || ( c != 0_ ) ) {
        res = 0;
        string cris = 'moderadora';
    } else
        res = 1;
    }
    return res;
}
```

Pregunta 3 (1,75 punts)

Donada la següent declaració en C:

```
int func2(char a, int b); int  
  
func1 (int x,int*y) {  
    char V[3];  
    int z;  
    z = func2(V[2], *y);  
    return z + x;  
}
```

Tradueix la subrutina func1 a llenguatge ensamblador MIPS.

Pregunta 4 (1,50 punts)

Donada la següent declaració en C:

```
int v[10], m[10][10];
main() { int i;

    for (i=9; i>=0; i--)
        v[i] = m[9-i][i];
}
```

Completa el següent codi MIPS omplint les caselles en blanc perquè sigui equivalent a l'anterior codi en alt nivell, tenint en compte que tant els elements de la matriu com del vector s'accedeixen utilitzant la tècnica d'accés seqüencial.

```
main:
    la    $t0, m+36
    la    $t1, v+36
    la    $t2, vv for:
    lw    $t3, 0($t0)
    sw    $t3, 0($t1)
    addiu $t0, $t0, 36
    addiu $t1, $t1, -4 bge
           $t1, $t2, jr $ra
```

Pregunta 5 (1,50 punts)

S'executa un programa de test en un ordinador que té 3 tipus d'instruccions (A,B,C), amb CPI diferents. La següent taula especifica el nombre d'instruccions de cada tipus que executa el programa i el seu CPI.

Tipus d'instrucció	Nombre d'instruccions	CPI
A	$5 * 10^9$	2
B	$6 * 10^9$	1
C	$4 * 10^9$	3

- a) Sabem que la freqüència de rellotge és de 2GHz i que la potència dissipada és $P=100W$. Calcula el temps d'execució en segons i l'energia consumida en Joules durant l'execució del programa de test.

$$T_{exe} = 14s$$

$$E = 1400J$$

- b) Es vol modernitzar l'equipament comprant un nou ordinador amb una CPU amb ISA diferent. Aquesta nova CPU funciona amb el mateix voltatge però te més transistors, així que la seva capacítancia agregada (C) és un 50% més gran. El factor d'activitat (α) és el mateix. A més a més, els seu ISA té 4 tipus d'instruccions (A,B,C,D). Un cop recompilat el programa, el nombre d'instruccions de cada tipus que executa i el seu CPI són les indicades en la següent taula:

Tipus d'instrucció	Nombre d'instruccions	CPI
A	$2 * 10^9$	1
B	$3 * 10^9$	1
C	$1 * 10^9$	1
D	$1 * 10^9$	1

$$C' = 1'5C$$

Sabem que amb aquesta nova CPU obtenim un speedup de 2. A quina freqüència (en GHz) va la nova CPU? ¿Quina potència dissipada en Watts consumeix l'execució del programa (podem suposar que la potència estàtica és zero tant en la CPU original com en la nova)?

$$Freq = 1Ghz \quad // \quad P/100 = 1'5/2$$

$$P = 75W$$

Pregunta 6 (1 punt)

Suposant un circuit seqüencial per a la multiplicació de números naturals de 4 bits, anàleg a l'estudiat a classe per a 32 bits. Supposem que amb aquest circuit multipliquem els números binaris de 4 bits 1010 (multiplicand) i 1101 (multiplicador). Completa la següent taula, que mostra els valors en binari dels registres P, MD, i MR després de la inicialització i després de cada iteració, afegint tantes iteracions com facin falta:

iteració	P	MD	MR
valor inicial	00000000	00001110	1011
1	00001110	00011100	0101
2	00101010	00111000	0010
3	00101010	01110000	0001
4	10011010	11100000	0000

Pregunta 7 (1,50 punts)

Considera que el contingut dels registres \$f4 i \$f6 és 0x41800003 i 0xC0000005, respectivament i que s'executa la instrucció MIPS: add.s \$f0, \$f4, \$f6. Suposant que el sumador/restador té 1 bit de guarda, un d'arrodoniment i un de "sticky", i que arrodoneix al més pròxim (al parell en el cas equidistant), quin és el valor de \$f0 en hexadecimal després d'executar la instrucció?

0100 0001 1000 0000 0000 0000 0000 0011
1,000 0000 0000 0000 0000 0011 *2⁴

1100 0000 0000 0000 0000 0000 0000 0101
-1,000 0000 0000 0000 0000 0101*2

1,000 0000 0000 0000 0000 0011 000
-0,001 0000 0000 0000 0000 0000 101
0,111 0000 0000 0000 0000 0010 011 //1,110 0000 0000 0000 0000 0100 110
1,110 0000 0000 0000 0000 0101*2³

0
0x41600005

1,110 0000 0000 0000 0000 0101 000
1,110 0000 0000 0000 0000 0100 110
0,000 0000 0000 0000 0000 0000 010 1*2⁻²²