

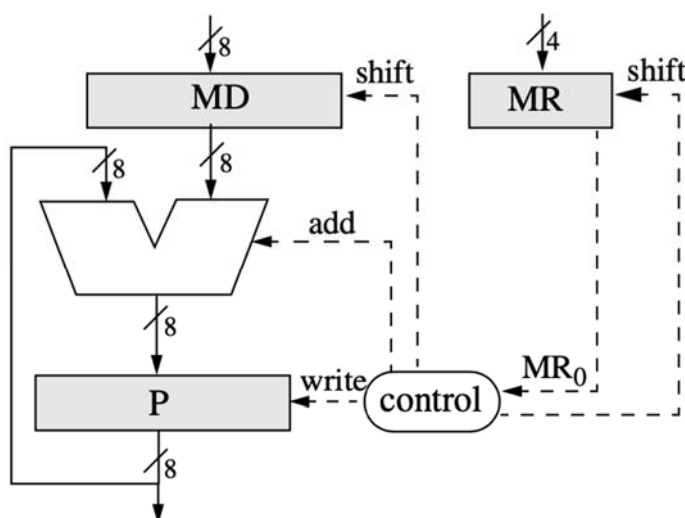
EXAMEN PARCIAL D'EC

3 de novembre de 2022

- L'examen consta de 5 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 14 de novembre. La revisió es farà presencialment el 15 de novembre. De 8 a 9h (preguntes 1 i 2), i de 9 a 10h (preguntes 3, 4 i 5).

Pregunta 1 (1,25 punts)

Segui el següent diagrama del multiplicador seqüencial de números naturals de 4 bits, anàleg a l'estudiat al curs, el qual calcula el producte en 8 bits:



Suposem que amb aquest circuit multipliquem els números binaris de 4 bits 1100 (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:

Iter.	P (Producte)								MD (Multiplicand)								MR (Multiplicador)			
Init.																				
1																				
2																				
3																				
4																				
5																				
6																				
7																				
8																				

Pregunta 2 (2,50 punts)

Donat el següent codi en llenguatge C que utilitza accés seqüencial

```
void quadrat(int M[200][200]) {  
    int i;  
    for (i=0; i < 200; i+=4)  
        M[i][200-i-1] = i*i;  
}
```

Completa el codi MIPS que apareix a continuació:

```
quadrat:  
    move    $t0, $zero  
for:  
    slti    $t1, $t0, 200  
    beq     $t1, $zero, return  
    mult    $t0, $t0  
    mflo    $t1  
  
    sw      $t1,  ($a0)  
    addiu     
    addiu   $t0, $t0, 4  
    b       for  
return:  
    jr      $ra
```

Considerant que:

- Les instruccions de memòria (lw/sw) tarden 4 cicles
- Els salts, si no salten, tarden 1 cicle
- Els salts, si salten, tarden 2 cicles
- La resta d'instruccions tarden 1 cicle
- La freqüència de rellotge del processador (f) és 2Ghz
- La potència dissipada per la CPU (P) és de 40W

Quantes instruccions s'executen en aquesta funció?

Quants cicles de processador es consumeixen en aquesta funció?

Calcula el temps d'execució (t_{exe}) d'aquesta funció en nanosegons
(pots deixar l'expressió que el calcula sense avaluar)

Calcula el CPI promig de la funció
(pots deixar l'expressió que el calcula sense avaluar)

Indica la fórmula que utilitzaries a partir de les dades proporcionades per
calcular l'energia (E) consumida durant l'execució d'aquest programa.

Pregunta 3 (2,50 punts)

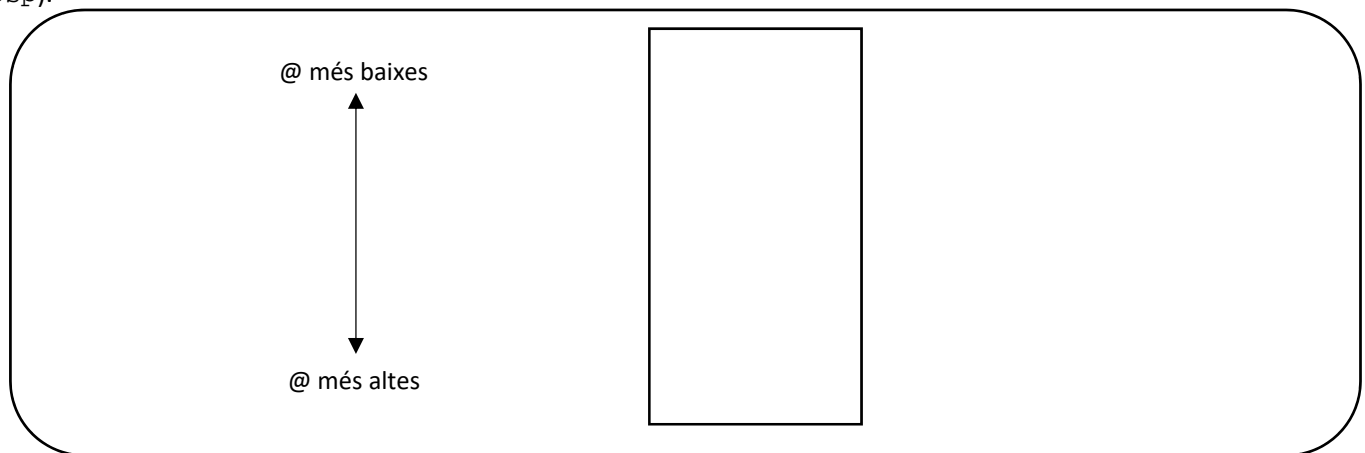
Donades les següents declaracions, en C:

```
int g(int p1, char *p2, int *p3);
int f(int par, char dat){
    char vec[10];
    int s;
    do {
        par = s + g(par, vec, &s);
    } while (vec[par] != dat);
    return par;
}
```

Seguint les regles de l'ABI de MIPS estudiades, quins elements de la funció `f` (variables locals, paràmetres, o càlculs intermedis) s'han d'emmagatzemar necessàriament en registres de tipus segur `$s`?

<u>Element de <code>f</code> (en C)</u>	<u>Registre <code>\$s</code></u>

Dibuixa un esquema del bloc d'activació de `f`, especificant-hi la posició on apunta el registre `$sp` un cop reservat l'espai corresponent a la pila, així com el nom de cada registre i/o variable, i la seva posició (desplaçament relatiu a `$sp`).



Tradueix a MIPS la següent sentència del cos de la subrutina `f`:

```
par = s + g(par, vec, &s);
```

Pregunta 4 (2,50 punts)

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

```
char a = '3';
short b[4] = {-19, 7, -55, 1};
int c = 5;
short *d = b;
long long e;
char f[] = "2022";
short g[100];
```

Tradueix-la a llenguatge ensamblador MIPS.

.data

Completa la següent taula amb el contingut de memòria en hexadecimal (sense el prefix 0x), dels primers 48 bytes. Tingues en compte que el codi ASCII del '0' és el 0x30. 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
0x10010000		0x10010010		0x10010020	
0x10010001		0x10010011		0x10010021	
0x10010002		0x10010012		0x10010022	
0x10010003		0x10010013		0x10010023	
0x10010004		0x10010014		0x10010024	
0x10010005		0x10010015		0x10010025	
0x10010006		0x10010016		0x10010026	
0x10010007		0x10010017		0x10010027	
0x10010008		0x10010018		0x10010028	
0x10010009		0x10010019		0x10010029	
0x1001000A		0x1001001A		0x1001002A	
0x1001000B		0x1001001B		0x1001002B	
0x1001000C		0x1001001C		0x1001002C	
0x1001000D		0x1001001D		0x1001002D	
0x1001000E		0x1001001E		0x1001002E	
0x1001000F		0x1001001F		0x1001002F	

Donat el següent codi en ensamblador MIPS, indica quin és el valor final en hexadecimal del registre \$t0:

```
la      $t1, b
lh      $t2, 0($t1)
lh      $t3, 2($t1)
div     $t2, $t3
mfhi    $t0
```

\$t0 = 0x

Cognoms: Nom:
 DNI:

Donat el següent codi en assembleador MIPS, indica quin és el valor final en hexadecimal del registre \$t0:

```
lui      $t1, 0x1001
addiu    $t1, $t1, 6
lh       $t2, 0($t1)
sra      $t0, $t2, 4
```

\$t0 = 0x

Tradueix a llenguatge assembleador del MIPS la següent sentència en C:

```
*(d+1) = *d + 5;
```

Pregunta 5 (1,25 punts)

Donada la següent funció escrita en C:

```
int f(int x, int y)
{
    if (x>y && (x%16)==0)
        return x;
    else
        return -1;
}
```

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

f:	<input type="text"/>	\$a0, \$a1,	<input type="text"/>
et1:	<input type="text"/>	\$t0, \$a0, 15	
et2:	<input type="text"/>	\$t0, \$zero,	<input type="text"/>
et3:	move	\$v0, \$a0	
et4:	b	<input type="text"/>	
et5:	li	\$v0, -1	
et6:	jr	\$ra	