

UNIVERSIDAD CARLOS III DE MADRID
DEPARTAMENTO DE INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA. ESTRUCTURA DE COMPUTADORES

Para la realización del presente examen se dispondrá de **2 horas**. **NO** se podrán utilizar libros, apuntes **ni** calculadoras de ningún tipo.

Ejercicio 1. Dado el siguiente fragmento de programa en ensamblador.

```
.text
                                .globl main
main:
                                li    $a0, 5
                                jal   funcion
                                move  $a0, $v0
                                li    $v0, 1
                                syscall

                                li    $v0, 10
                                syscall

funcion:
                                move  $t0, $a0
                                li    $t1, 0
bucle: beq  $t0, 0, fin
                                add  $t1, $t1, $t0
                                sub, $t0, $t0, 1
                                b     bucle
fin:   move $v0, $t1
                                jr   $ra
```

Se pide :

- a) Indicar de forma razonada el valor que se imprime por pantalla (primera llamada al sistema del código anterior).
- b) Si en el registro `$a0`, que se utiliza para el paso de parámetros a la función, el valor que se almacena se representa en complemento a uno, ¿qué rango de números podrían pasarse a la función?

Ejercicio 2. Considere una función denominada `Vocales`. A esta función se le pasa como parámetro la dirección de inicio de una cadena de caracteres. La función calcula el número de veces que aparece el carácter 'a' (en minúscula) en la cadena. En caso de pasar la cadena nula la función devuelve el valor -1. En caso de que la cadena no tenga ninguna 'a', la función devuelve 0. Se pide:

- a) Programar utilizando el ensamblador del MIPS32 el código de la función `Vocales`.
- b) Indique en qué registro se ha de pasar el argumento a la función y en qué registro se debe recoger el resultado.
- c) Dado el siguiente fragmento de programa:

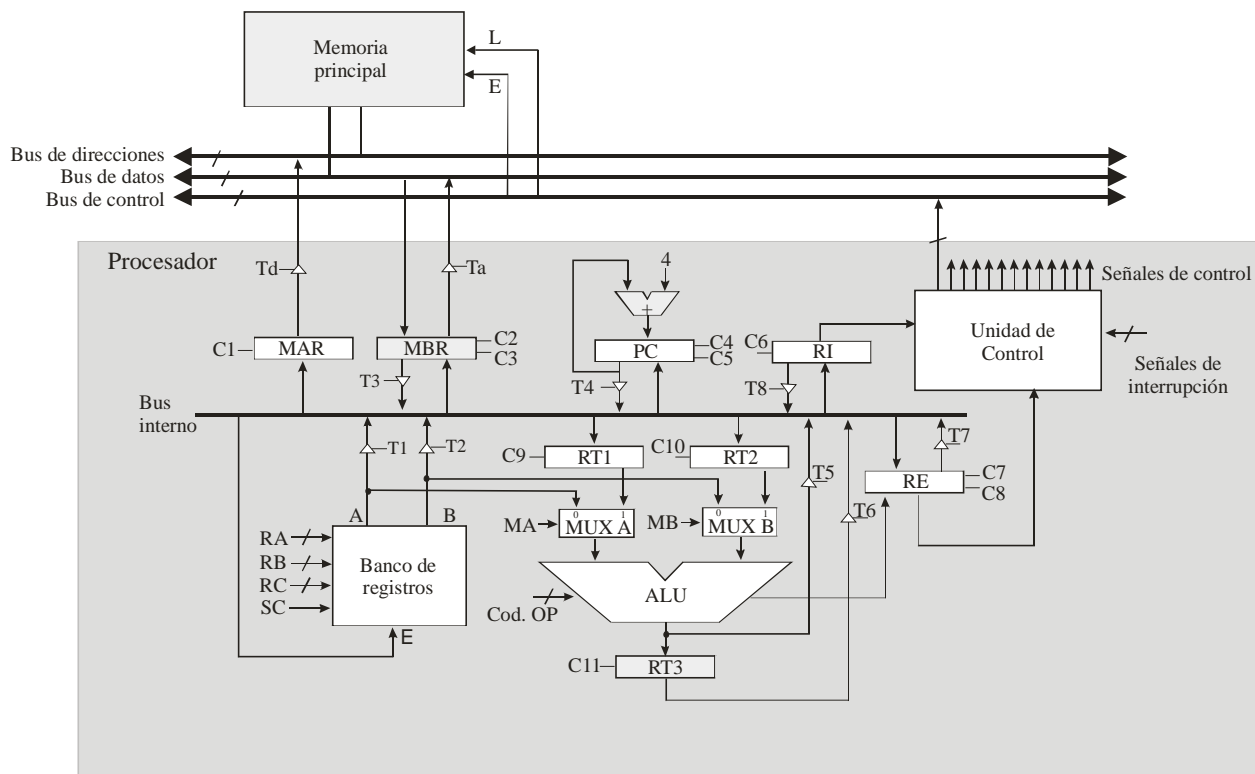
```
.data
cadena: .asciiz "Hola"

.text
                                .globl main

main:
```

incluya en el `main` anterior, las sentencias en ensamblador necesarias para poder invocar a la función `Vocales` implementada en el apartado a) e imprimir por pantalla el valor que devuelve la función. El objetivo es imprimir el número de veces que aparece el carácter 'a' en la cadena "Hola".

Ejercicio 3. Considere el siguiente esquema de procesador de 32 bits. El banco de registros incluye 32 registros. Considere que el computador utiliza un ciclo de reloj para realizar la decodificación de la instrucción y que se conecta a una memoria que permite realizar una operación de lectura y de escritura en un ciclo.



Indique las operaciones elementales y las señales de control (incluyendo el *fetch*) necesarias para ejecutar la instrucción `lw $t1, ($t2)`.

Ejercicio 4. Sea un computador de 32 bits con una memoria caché para datos de 32 KB y líneas de 64 bytes. La caché es asociativa por conjuntos de 2 vías y se emplea la política de reemplazo LRU. Dado el siguiente fragmento de código:

```
int v[262144]

for (i = 0; i < 262144; i = i + 2)
    v[i] = 9;
```

se pide:

- Indique el número de líneas y de conjuntos de la memoria caché de datos del enunciado.
- Considerando exclusivamente la caché de datos y los accesos al vector, calcule de forma razonada la tasa de fallos que se obtiene en la ejecución del bucle anterior.

Ejercicio 5. Sea un computador de 20 bits con memoria virtual paginada con páginas de 1 KB y un total de memoria física de 256 KB. Se pide, de forma razonada y breve:

- ¿Cuál es el formato de la dirección virtual? Indique los campos y el número de bits de los mismos.
- ¿Cuál es el número máximo de entradas de la tabla de páginas (de un nivel)?
- ¿Cuántos marcos de página tiene la memoria principal?
- ¿Cuáles son los campos que se incluyen en una entrada de la tabla de páginas? Indique también para qué se utiliza cada uno de los campos.