

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 :

- Indicar de forma razonada el valor que se imprime por pantalla (primera llamada al sistema del código anterior).
- 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?

Solución del ejercicio 1

- La función realiza la suma de los números 5, 4, 3, 2,1 y devuelve el resultado en el registro `$v0`, cuyo valor por tanto es 15, que es el resultado que se imprime por pantalla.
- Para una palabra de n bits, el rango de representación de números en complemento a 1 es $[-2^{n-1}+1, 2^{n-1}-1]$. En el caso del MIPS 32, los registros son de 32 bits, por tanto $n = 32$ y el rango de representación sería $[-2^{31}+1, 2^{31}-1]$.

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:

- Programar utilizando el ensamblador del MIPS32 el código de la función `Vocales`.
- Indique en qué registro se ha de pasar el argumento a la función y en qué registro se debe recoger el resultado.
- 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".

Solución del ejercicio 2

- a) Se asume que la dirección de la cadena se pasa en el registro \$a0 y el resultado se devuelve en el registro \$v0. El código de la función vocales es el siguiente

```

vocales:  li    $t0, -1      // contador del número de a
          move  $t1, $a0
          beqz  $t1, fin
          li    $t0, 0
          li    $t2, 'a'
bucle:   lbu   $t3, ($t1)
          beqz  $t3, fin
          bneq  $t3, $t2, noA
          addi  $t0, $t0, 1
noA:     addi  $t1, $t1, 1
          b     bucle

fin:     move  $v0, $t0
          jr   $ra

```

- b) Los argumentos se pasan en los registros \$aX y los resultados en los registros \$vX. En este caso la dirección de inicio de la cadena se pasa en \$a0 y el resultado se recoge en \$v0.
- c) El cuerpo e la función main es:

```

.data
        cadena:  .asciiz    "Hola"
.text
        .globl main

main:    sub    $sp, $sp, 24
          sw    $ra, 20($sp)
          sw    $a0, 4($sp)

          la    $a0, cadena
          jal   vocales

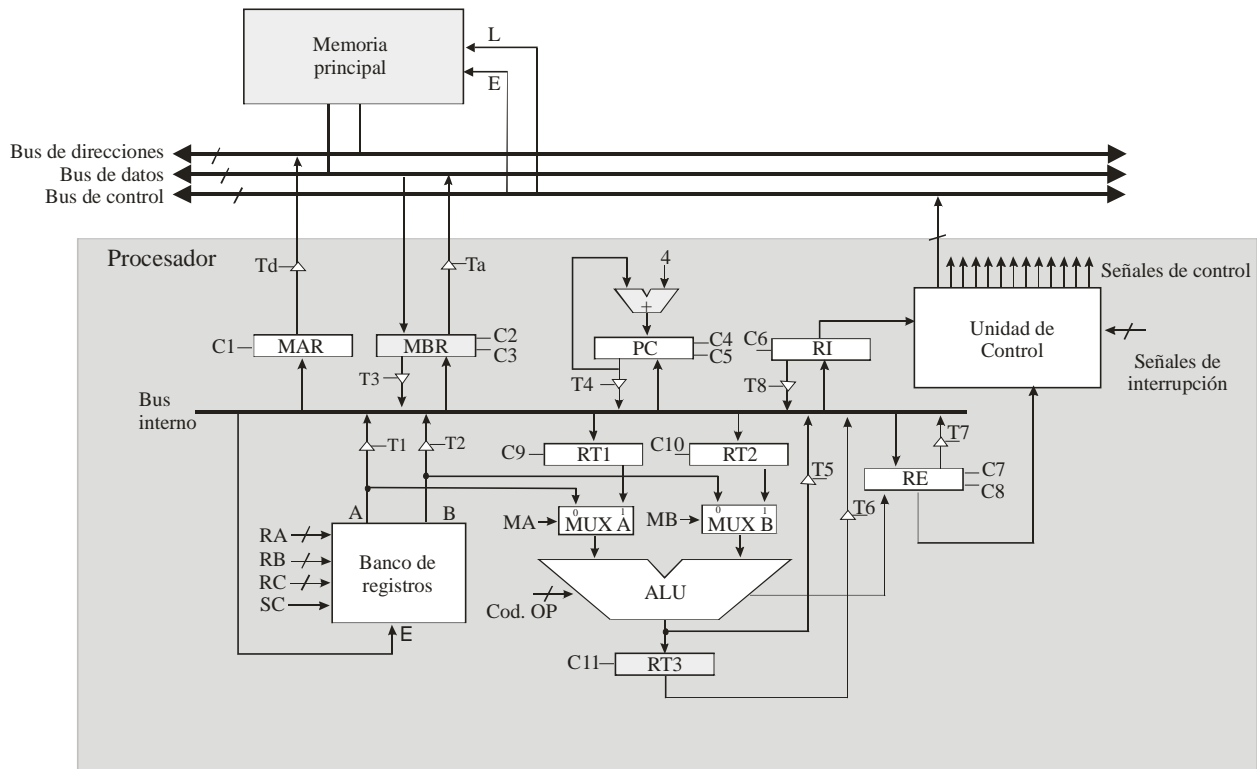
          move  $a0, $v0
          li    $v0, 1
          syscall

          lw    $ra, 20($sp)
          lw    $a0, 4($sp)
          addi  $sp, $sp, 24

          li    $v0, 10
          syscall
          jr   $ra

```

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)`.

Solución del ejercicio 3

En la siguiente tabla se muestran las operaciones elementales y las señales de control.

| Ciclo | Operación elemental | Señales de control activadas |
|-------|--|------------------------------|
| C1 | MAR \leftarrow PC | T4, C1 |
| C2 | MBR \leftarrow MP, PC \leftarrow PC + 4 | L, Td, C2, C4 |
| C3 | RI \leftarrow MBR | T3, C6 |
| C4 | Decodificación | |
| C5 | MAR \leftarrow \$t2, | RA = <dir de \$t2> T1, C1 |
| C6 | MBR \leftarrow MP | Td, L, C2 |
| C7 | \$t1 \leftarrow MBR | T3, SC, RA = <dir de \$t1> |

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.

Solución del ejercicio 4

- La caché tiene un tamaño de $32 \text{ KB} = 2^{15}$ bytes. como cada línea tiene 2^6 bytes, el número de líneas es $2^{15} \text{ bytes} / 2^6 \text{ bytes} = 2^9$ líneas = 512 líneas. Como la caché es asociativa por conjuntos de 2 vías, cada conjunto tiene dos líneas, por tanto el número de conjuntos es $512 / 2 = 256$ conjuntos.
- El patrón de accesos del vector al bucle es el siguiente:

$v[0], v[2], v[4], v[6], v[8] \dots$

es decir, se accede cada 2 elementos. Como la caché está estructurada en líneas de 64 bytes y cada dato de tipo entero (`int`) ocupa 4 bytes, en cada línea caben 16 elementos del vector. Como el vector se recorre de forma secuencial y de estos 16 solo se acceden realmente a 8, la tasa de fallos es $1/8$.

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.

Solución del ejercicio 5

- Las páginas ocupan $1 \text{ KB} = 2^{10}$ bytes. Como la dirección virtual ocupa 20 bits, se emplean $20 - 10 = 10$ bits para el número de página. Por tanto, el formato emplea los 10 bits superiores de la dirección para representar el número de página y los 10 bits inferiores para representar el desplazamiento dentro de la página.
- El número máximo de entradas de la tabla de páginas coincide con el número máximo de páginas, es decir $2^{10} = 1024$ entradas.
- El número de marcos de página viene dado por $256 \text{ KB} / 1 \text{ KB} = 256$ marcos.
- En cada entrada de la tabla de página se incluye, entre otros:
 - Bit de presencia
 - Bit de modificado
 - Bit de validez
 - Bits de permisos
 - Campo en el que se almacena el marco.