

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

Para la realización del presente examen se dispondrá de **1 hora y media**.  
**NO** se podrán utilizar libros, apuntes **ni** calculadoras de ningún tipo.

---

**Ejercicio 1.** Represente en el estándar IEEE 754 de simple precisión el valor **-36**.

### Solución del ejercicio 1

El valor 36 en binario es 100100.  $100100 = 1.00100 \times 2^5$ . Por tanto:

- o El bit de signo es 1, porque el número es negativo.
- o El exponente es 5, por tanto el exponente que se almacena es  $5 + 127 = 132$ , que en binario es 10000100
- o La mantisa es 001000000 .... 00000

Por tanto el número -36 se representa como 11000010000100000000000000000000

**Ejercicio 2.** 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:
        li $t0, 10
        bgt $a0, $t0, et1
        li $t0, 10
        add $v0, $t0, $a0
        b et2
        et1:  li $t0, 8
              add $v0, $t0, $a0
        et2:  jr $ra
```

Se pide :

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

### Solución del ejercicio 2

- a) A la función se le pasa como argumento en el registro \$a0 el valor 5. Si este valor es mayor que (bgt) 10 se salta a la etiqueta et1. Como no es el caso, se suma 10 (\$t0) a 5 y el resultado se almacena en \$v0, que es el valor que se imprime, es decir, 15.

- Ejercicio 3.** Considere una función denominada `SumarValor`. A esta función se le pasan tres parámetros:
- El primer parámetro es la dirección de inicio de un vector de números enteros.
  - El segundo parámetro es un valor entero que indica el número de componentes del vector.
  - El tercer parámetro es un valor entero.

La función `SumarValor` modifica el vector, sumando el valor pasado como tercer parámetro a todas las componentes del vector. Se pide:

- Indique en qué registros se han de pasar cada uno de los parámetros a la función.
- Programar utilizando el ensamblador del MIPS32 el código de la función `SumarValor`.
- Dado el siguiente fragmento de programa:

```
.data
v: .word 7, 8, 3, 4, 5, 6
.text
.globl main

main:
```

incluya en el `main` anterior, las sentencias en ensamblador necesarias para poder invocar a la función `SumarValor` implementada en el apartado b) de forma que sume a las componentes del vector `v` definido en la sección de datos, el número 5. Implemente a continuación de la llamada a la función, las sentencias en ensamblador que permitan imprimir todas las componentes del vector.

### Solución del ejercicio 3

- La dirección de inicio se pasa en `$a0`, el número de elementos en `$a1` y el valor a sumar en `$a2`.

```
b)
SumarValor:    li    $t0, 0
               move  $t1, $a0

               bucle: bgt  $t0, $a1, fin
               lw    $t2, ($t1)
               add  $t2, $t2, $a2
               sw    $t2, ($t1)
               addi $t0, $t0, 1
               addi $t1, $t1, 4
               b    bucle
               fin:  jr   $ra
```

- El cuerpo e la función `main` es:

```
.data
v: .word 7, 8, 3, 4, 5, 6
.text
.globl main

main:    sub  $sp, $sp, 24
        sw  $ra, 20($sp)
        sw  $a0, 4($sp)
        sw  $a1, 8($sp)
        sw  $a2, 12($sp)

        la  $a0, v
        li  $a1, 6
```

```

        li    $a2, 5
        jal   SumarValor

        li    $v0, 1
        li    $t0, 0
        move  $t1, $a0

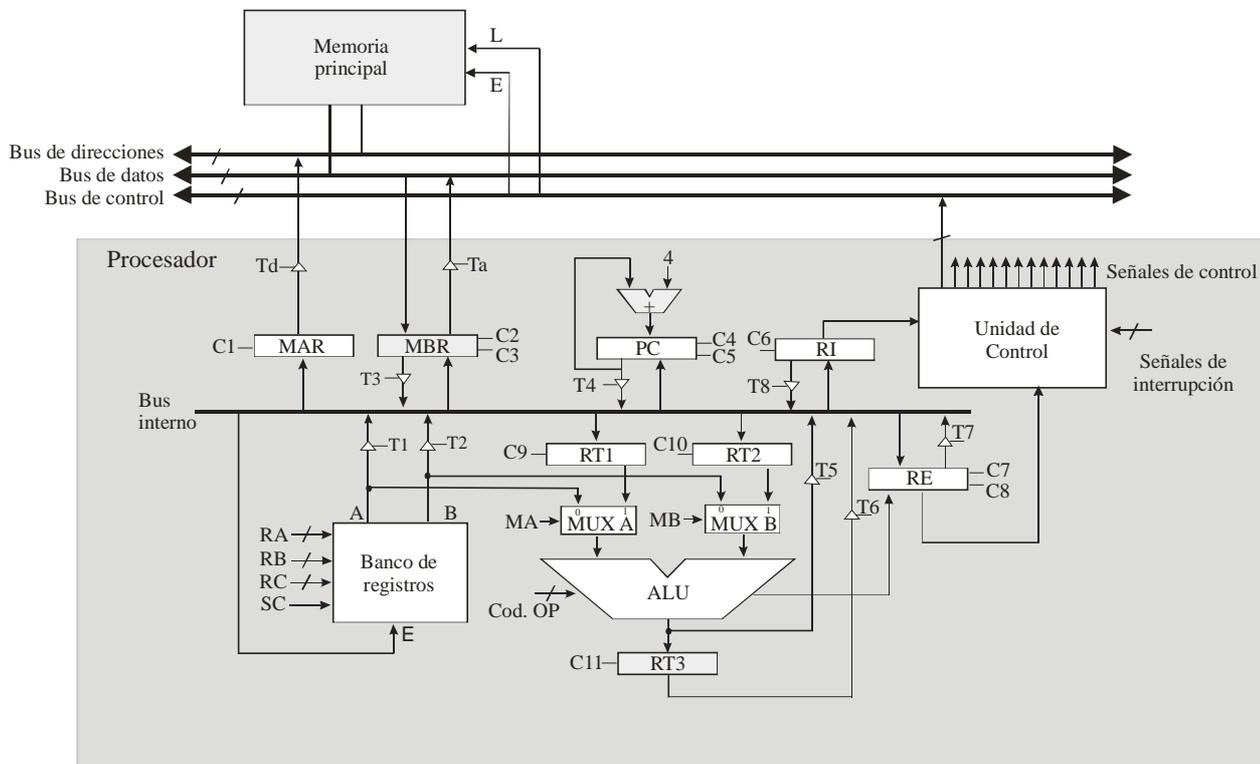
bucle:  bgt   $t0, $a1, fin
        lw   $a0, ($t1)
        syscall
        addi $t0, $t0, 1
        addi $t1, $t1, 4
        b    bucle

        fin: lw   $ra, 20($sp)
        lw   $a0, 4($sp)
        lw   $a1, 8($sp)
        lw   $a2, 12($sp)
        addi $sp, $sp, 20

        li    $v0, 10
        syscall
        jr   $ra

```

**Ejercicio 4.** 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.



Este computador dispone del juego de instrucciones del MIPS32. Indique las operaciones elementales y las señales de control (incluyendo el *fetch*) necesarias para ejecutar la instrucción `addi $t0, $t1, 10`.

## Solución del ejercicio 4

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

Ciclo	Operación elemental	Señales de control activadas
C1	MAR ← PC	T4, C1
C2	MBR ← MP, PC ← PC + 4	L, Td, C2, C4
C3	RI ← MBR	T3, C6
C4	Decodificación	
C5	TR2 ← RI(10)	T8, C10
C6	\$t0 ← \$t1 + TR2	RA = <dir de \$t1> MA = 0 MB = 1 Cod op = SUMAR T5 RC = <dir de \$t0> SC

**Ejercicio 5.** Sea un computador de 32 bits con una memoria caché de 256 KB, líneas de 64 bytes y un tiempo de acceso de 5 ns. La caché es asociativa por conjuntos de 4 vías y se emplea la política de reemplazo LRU. Se pide:

- Indique el número de líneas y de conjuntos de la memoria caché del enunciado.
- ¿Cuál es el tamaño de los bloques que se transfieren entre la memoria caché y la memoria principal?
- Si el tiempo para transferir un bloque de memoria principal a caché es de 200 ns, indique la tasa de aciertos a caché necesaria, de forma que el tiempo medio de acceso al sistema de memoria sea de 20 ns.

## Solución del ejercicio 5

- La caché tiene un tamaño de 256 KB =  $2^{18}$  bytes. como cada línea tiene  $2^6$  bytes, el número de líneas es  $2^{18}$  bytes /  $2^6$  bytes =  $2^{12}$  líneas = 4096 líneas. Como la caché es asociativa por conjuntos de 4 vías, cada conjunto tiene cuatro líneas, por tanto el número de conjuntos es  $4096 / 4 = 1024$  conjuntos.
- El tamaño del bloque que se transfiere entre memoria principal y caché coincide con el tamaño de la línea, es decir, con 64 bytes.
- El tiempo medio de acceso al sistema viene dado por la siguiente expresión:

$$t_m = t_c \cdot P_a + (1 - P_a) \cdot t_f$$

donde  $t_c = 5$  ns,  $t_m = 20$  ns y  $t_f = 205$  (200 + 5). Por tanto:

$$20 = 5 \cdot P_a + (1 - P_a) \cdot 205$$

Despejando, se obtiene que  $P_a = 185 / 200 = 0,92$ . Es decir, la tasa de aciertos debe ser del 92 %.