

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

Para la realización del presente examen se dispondrá de **1:30 horas**.  
**NO** se podrán utilizar libros, apuntes **ni** calculadoras de ningún tipo.

---

**Ejercicio 1.** Se desea desarrollar un controlador para un horno microondas. El controlador dispone de un procesador de 32 bits, mapa de E/S común y juego de instrucciones del MIPS 32. A este procesador se le conectan un módulo de E/S, que controla el funcionamiento del horno. El módulo dispone de los cinco siguientes registros de 32 bits:

- Registro con dirección 1000. En este registro se carga el valor correspondiente a la cuenta atrás en segundos.
- Registro con dirección 1004. En este registro se carga un 1 cuando se quiere comenzar la cuenta atrás. Mientras haya un 0 no se inicia la cuenta atrás.
- Registro con dirección 1008. Cuando la cuenta atrás llega a 0, en este registro el controlador almacena un 1. Mientras se está realizando la cuenta atrás el valor que se encuentra almacenado en este registro es 0.
- Registro con dirección 1012. En este registro se carga el valor correspondiente a la potencia deseada. El valor es un entero de 32 bits en complemento a dos.
- Registro con dirección 1016. Cuando se carga en este registro el valor 1, el microondas se pone en funcionamiento con la potencia almacenada en el registro anterior. Cuando en este registro se carga un 0, el microondas se para independientemente de que se haya llegado al final de la cuenta atrás.

Se desea escribir una función en ensamblador, denominada `Controlador_Horno` que permita controlar el funcionamiento del horno. Esta función recibirá dos parámetros de entrada: un valor entero indicando el número de segundos que ha de permanecer encendido el horno y un valor entero que representa la potencia del horno. La función se encargará de programar el controlador anterior con la potencia y segundos pasados como parámetros, encargándose de poner en funcionamiento el horno y apagarlo cuando haya transcurrido el tiempo necesario. La función devolverá los siguientes posibles valores:

- Un valor igual a 0 que representa que no ha habido ningún error y que se devolverá cuando la función haya parado el horno.
- Un valor igual a -1, cuando la potencia pasada como parámetro sea menor de 100 y mayor de 1000. En este caso no se ha de programar el controlador del horno y la función debe retornar inmediatamente el valor -1.

Se pide:

- a) Indique en qué registros se han de pasar los parámetros y en qué registro se debe devolver el resultado.
- b) Invoque a la función anterior para un valor de potencia de 800 W y una duración de 90 segundos. Escriba a continuación el código que permita imprimir por pantalla el resultado que devuelve la función.
- c) Escriba el código correspondiente a la función `Controlador_Horno`.

## Solución del ejercicio 1

a) Los parámetros de entrada se pasan en los registro \$a0 y \$a1 y el resultado se devuelve en \$v0.

b)

```
li    $a0, 800
li    $a1, 90
jal   Controlador_Horno
move  $a0, $v0
li    $a1, 1
syscall
```

c)

```
controlador_Horno:
                                li    $t0, 100
                                blt    $a0, $t0, fin_error
                                li    $t0, 1000
                                bgt    $a0, $t0, fin_error
                                sw     $a1, 1000          ; cuenta atrás
                                sw     $a0, 1012          ; potencia
                                li    $t0, 1
                                sw     $t0, 1016          ; arranque motor
                                sw     $t0, 1004          ; iniciar cuenta atrás
bucle:  lw     $t1, 1008          ; leer estado cuenta
                                beq    $t1, $0, bucle
                                sw     $0, 1016          ; parar el motor
                                move   $v0, $0
                                jr     $ra
fin_error: li $v0, -1
                                jr     $ra
```

**Ejercicio 2.** Se desea representar números enteros dentro del rango -8191...8191. Indicar de forma razonada:

- ¿Cuál es el número de bits que se necesita si se quiere utilizar una representación en complemento a uno?
- ¿Cuál es el número de bits que se necesita si se quiere utilizar una representación en signo-magnitud?

## Solución del ejercicio 2

Se necesitan en los dos casos 14 bits. Con 14 bits el rango de representación en ambos casos es de  $-(2^{13}-1) \dots 2^{13}-1 = -8191 \dots 8191$

**Ejercicio 3.** Represente el número -24,50 utilizando el estándar de coma flotante de simple precisión IEEE 754. Exprese dicha representación en binario y en hexadecimal.

### Solución del ejercicio 3

$$24,5_{(10)} = 11000.1_{(2)} = 1,10001 \times 2^4$$

Signo = 1, número negativo

Exponente =  $4 + 127 = 131 = 10000011$

Mantisa = 1000100000 .. 00000

En binario => 11000000111000100000 .....00000

En Hexadecimal => 0xC1C40000

**Ejercicio 4.** Sea un computador de 32 bits que dispone de una memoria caché de 512 KB, asociativa por conjuntos de 4 vías y líneas de 64 bits. Sobre este computador se desea ejecutar el siguiente fragmento de código:

```
int v[200];
int i;
int s;

for (i=0; i < 200; i++)
    s = s + v[i];
```

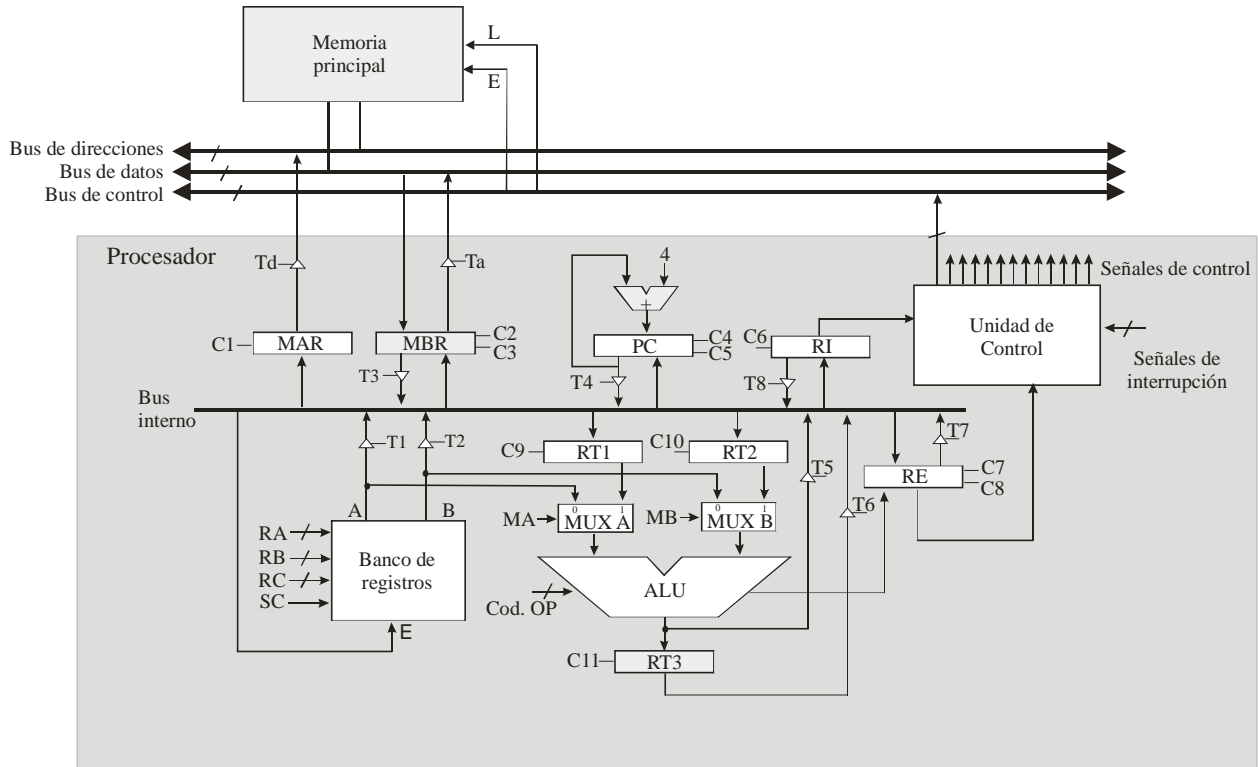
Se pide:

- Indique de forma razonada el número de líneas y conjuntos de de la caché.
- Si se considera que la caché se encuentra vacía y que los valores de las variables *i* y *s* del código anterior se almacenan en registros, indique, considerando solo los accesos al vector *v*, la tasa de aciertos que se obtiene al ejecutar el fragmento de código anterior.

### Solución del ejercicio 4

- La memoria caché tiene un tamaño de  $512 \text{ Kb} = 2^{19} \text{ bytes}$ . Cada línea tiene  $64 \text{ bits} = 8 \text{ bytes} = 2^3 \text{ bytes}$ . El número de línea viene dado por  $2^{19} \text{ bytes} / 2^3 \text{ bytes} = 2^{16} \text{ líneas}$ . Como cada conjunto tiene 4 líneas, el número de conjuntos  $= 2^{16} / 2^2 = 2^{14} \text{ conjuntos}$
- Como cada elemento del vector es un entero de 4 bytes, en cada línea caben dos enteros. Cada vez que se accede a un elemento se produce un fallo, y se traen a caché el elemento que ha producido el fallo y el siguiente, que da lugar a un acierto en la siguiente vuelta del bucle. Por tanto, hay un fallo cada dos accesos y la tasa de aciertos es del 50 %.

**Ejercicio 5.** 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. Se pide:

- Indique las señales de control necesarias para poder realizar la operación elemental  $RT2 \leftarrow PC$ .
- Indique las operaciones elementales y las señales de control (incluyendo el *fetch*) necesarias para ejecutar la instrucción `lw $t1, ($t2)`. Explique en primer lugar qué hace esta instrucción.

## Solución del ejercicio 5

- Hay que activar las señales T4 y C10
- 

Ciclo 0: $MAR \leftarrow PC$	Señales: T4, C3
Ciclo 1: $MBR \leftarrow MP$ $PC \leftarrow PC + 4$	Señales: Td, L, C2 C4
Ciclo 2: $RI \leftarrow MBR$	Señales: T3, C6
Ciclo 3: Decodificación	
Ciclo 4: $MAR \leftarrow \$t2$	Señales: RA= identificador de \$t2, T1, C1
Ciclo 5: $MBR \leftarrow MP$	Señales: Td, L, C2
Ciclo 6: $$t1 \leftarrow MBR$	T3, RC = identificador de \$t1, SC