



# **Fundamentos y Arquitectura de Computadores**

## **I. T. Telecomunicación**

**Examen extraordinario - 1 julio de 2003**

---

**Notas importantes:**

- No se considerarán válidas las soluciones entregadas a lápiz.
  - El nombre del alumno debe figurar en la hoja del examen y en todas las hojas adicionales entregadas.
  - Las cuestiones deberán resolverse dedicando como máximo media cara de una hoja A-4 para cada uno de ellas. Se valorarán negativamente las respuestas con una extensión superior.
  - Se graparán juntas todas las hojas. No es necesario entregar el enunciado.
  - Se recomienda leer *detenidamente* los problemas y cuestiones.
- 

### **Cuestión 1 (1 punto)**

Definir los conceptos “palabra de procesador” y “palabra de memoria”. ¿Cómo se relacionan?

### **Cuestión 2 (1,5 puntos)**

Represente el número -1828,25 en el formato IEEE 754 de coma flotante de simple precisión. Indique el resultado en hexadecimal, detallando los pasos intermedios.

### **Cuestión 3 (1 punto)**

Describir brevemente los tres algoritmos de reemplazo utilizados para desalojar bloques de una memoria cache.

### **Cuestión 4 (1,5 puntos)**

Explicar cómo funciona el sumador con acarreo anticipado y por qué reduce el tiempo necesario para efectuar una suma.

### **Problema 1 (3 puntos)**

Se desea construir una memoria de 512 Mbytes, de la que los últimos 32 Mbytes deberán ser de ROM y el resto de RAM. Se dispone para ello de los siguientes módulos en cantidad suficiente:



- RAM: Módulos de tipo A, de 128 Mpalabras de 4 bits.
- RAM: Módulos de tipo B, de 64 Mpalabras de 4 bits.
- RAM: Módulos de tipo C, de 32 Mpalabras de 8 bits.
- ROM: Módulos de tipo D, de 32 Mpalabras de 4 bits.
- Circuitería combinacional adicional en cantidad suficiente.

Diseñar el módulo de memoria, indicando detalladamente la conexión de las líneas de datos, direcciones y control necesarias para gestionarla.

## Problema 2 (2 puntos)

Se dispone de un ordenador con una cache L1 de datos de 1 Kbyte, con correspondencia directa y cuatro marcos de bloque. En este ordenador se ejecutan dos versiones de un programa que calcula la suma de todos los elementos de una matriz de números reales de simple precisión. La matriz tiene 16 filas y 64 columnas. La primera versión accede a la matriz por filas:

```
/* Version 1 */
...
float matriz [16][64];
float suma=0;
int i,j;
...
for (i=0; i<16; i++)
    for (j=0; j<64; j++)
        suma = suma + matriz[i][j];
...
```

La segunda versión, por el contrario, accede a la matriz por columnas:

```
/* Version 2 */
...
float matriz [16][64];
float suma=0;
int i,j;
...
for (j=0; j<64; j++)
    for (i=0; i<16; i++)
        suma = suma + matriz[i][j];
...
```

Supongamos que cada acierto cache en la lectura de un elemento de 4 bytes se resuelve en un ciclo de reloj, y cada fallo cache se resuelve en 100 ciclos de reloj. Si la cache de datos está inicialmente vacía y que la dirección del primer elemento de la matriz se corresponde con el primer byte del marco de bloque 0, calcular el tiempo consumido en los accesos a la matriz para ambas versiones. Ignorar los accesos a datos provocados por las variables *i*, *j* y *suma*. Justificar el resultado obtenido.