

Estructura de Computadores

Nombre:

Grupo:

1

Un sistema basado en el procesador MIPS R2000 dispone de una **CACHE L1 DUAL** (una L1 de Instrucciones y una L1 de Datos), cuya configuración es de 8 KB cada una, 2 vías y tamaño de bloque 64B. La política de fallo en escritura es de Ubicación (*write allocate*) y la política de acierto en escritura (actualización) es de escritura Posterior (*write back*). El algoritmo de reemplazo es LRU.

a) (0,5 puntos) Indique los campos (nombre y tamaño) en que se descompone la dirección de memoria principal

31	12	11	6	5	0
Etiqueta (20 bits)			Conjunto (6 bits)		Desplaz. (6 bits)

b) Calcule los siguientes parámetros de la cache L1 de Datos

Número de líneas	128
Número de conjuntos	64

c) Calcule el tamaño en bits de la **memoria de control de la cache de Datos**. Indique claramente el **número de líneas o entradas** que contiene y el **tamaño en bits de cada entrada**, especificando el nombre de cada uno de los **campos que la integran**:

128 líneas × [V (1) + M (1) + Etiqueta (20) + LRU (1)] = 128 líneas × 23 bits = 2944 bits

d) El siguiente programa realiza la suma de los elementos V[1] a V[4095] del vector V de números enteros y almacena el resultado en la cabecera del vector (elemento V[0]): $V[0] = \sum_{i=1}^{4095} V[i]$

```
.data 0x2F000800
A:    .word 1,2,3, ...,4096      # vector de 4096 enteros (32 bits)

.text 0x00400000
_start:li $t4,4095               # carga contador
      li $t2,0                   # inicializa suma=0
      lui $t0,0x2F00             # carga puntero a vector A
      ori $t0,$t0,0x0800
buc:  lw $t1,4($t0)               # lee V[i]
      add $t2,$t2,$t1            # suma=suma+V[i]
      addi $t0,$t0,4             # incrementa puntero a vector V
      addi $t4,$t4,-1            # decrementa contador
      bnez $t4,buc              # mientras contado > 0, seguir en el bucle
      lui $t0,0x2F00             # carga puntero a vector V
      ori $t0,$t0,0x0800
      sw $t2,0($t0)              # V[0]=suma
      .end
```

d.1) Calcule los siguientes valores tanto para Instrucciones como para Datos

	CÓDIGO	DATOS
Número de bloques que lo contienen	1	256
Número de bloque de los dos primeros bloques (hex)	0x0010000	0x0BC0020 0x0BC0021
Conjunto al que se mapea el primer bloque (hex)	0x00	0x20
Etiqueta del primer bloque	0x00400	0x2F000
Número de FALLOS	1	257
Número de ACCESOS (Mostrar el cálculo)	$4 + 5 \times 4095 + 3 = 20482$	4096
Número de reemplazos de bloque	0	129
TASA DE ACIERTOS	$H_{L1I} = 1 - \frac{1}{20482} = 0,999$	$H_{L1D} = 1 - \frac{257}{4096} = 0,937$
TASA DE ACIERTOS PROMEDIO (Mostrar el cálculo)	$H_{L1} = \frac{0,999 \times 20482 + 0,989 \times 4096}{24578} = 0,989$	

d.2) Asumiendo la existencia de un nivel L2 de cache (Unificada I+D), con tasa de acierto $H_{L2} = 0,9$, y suponiendo que los tiempos de acceso de los niveles L1 y L2 son 1 ns y 4 ns, respectivamente, y que el acceso a memoria principal es de 300ns, calcule cuál sería el tiempo medio de acceso a memoria en la ejecución del anterior programa

$$T_m = 0,989 \times 1\text{ns} + (1-0,989) \times \{0,9 \times 4\text{ns} + (1-0,9) \times 300\text{ns}\} = 1,36 \text{ ns}$$

d.3) Suponga que la L1 de Datos emplea correspondencia directa en lugar de correspondencia asociativa por conjuntos de 2 vías, manteniendo la capacidad. Comente cómo se vería afectada su **tasa de aciertos** y el **tamaño de su memoria de control**. Razone la respuesta

La tasa de aciertos no se verá afectada, dado que el fallo que se produce al acceder a $V[0]$ es de capacidad, y este no se eliminaría cambiando la función de correspondencia, sino que requeriría aumentar el tamaño de la cache.

El tamaño de la memoria de control disminuye al reducirse el tamaño de la etiqueta (-1 bit) y eliminarse la necesidad de contador LRU (-1 bit).