

TAREA 1

Actividad a desarrollar:

Se quiere diseñar una memoria caché de correspondencia directa (o mapeo directo) de 4K bytes de capacidad, con líneas de 4 palabras. La Memoria Principal, tiene una capacidad de 16Mb. El CPU direcciona por byte y por palabra. El ancho de una palabra es de 32 bits.

a) Describa con un esquema la DIRECCIÓN VIRTUAL con la que se accede a la MEMORIA. El número de bits por cada campo, y su uso en la caché.

b) Describa la estructura lógica de la memoria caché diseñada, con un diagrama que indique los anchos, posiciones, líneas, etc.

Para la traza de direcciones dada abajo, calcule la tasa de aciertos utilizando la memoria caché diseñada. La estrategia utilizada en los accesos de escritura es "write through".

c) Proponga estrategias para mejorar el rendimiento de esta memoria caché sin aumentar el número de líneas. Compare el rendimiento con el caso anterior.

```
RB 010A16
RB 010A17
RB 010A18
```

```
RB 010A19
```

```
RB 0B0A1A
```

```
RB 0B0A1B
```

```
RB 00001C
```

```
RB 00001D
```

```
RB 000A1E
```

ESTE TROZO SE REALIZA 10 VECES

```
RW 30FB0C
```

```
RW 30FB10
```

```
RB A0AA1F
```

```
RB A0AF47
```

```
RB A0AF48
```

```
RB A0AF49
```

```
WW A0AB10
```

```
RB A0AA50
```

RB: Read Byte

RW: Read Word

WW: Write Word

Solución

a) Como la caché es de 4 Kbytes, y los bloques son de 4 palabras, tenemos que el número de líneas para la caché de mapeo directo es: $((4096 \text{ bytes}) / (4 \text{ palabras} \times 4 \text{ bytes})) = 256 \text{ líneas}$. Con lo cual tenemos:

- 256 líneas

→ 8bits para seleccionar línea

- 4 palabras

→ 2bits para seleccionar palabra

- 4bytes/palabra

→ 2bits para seleccionar byte

→ 12 bits de etiqueta resultado de todo lo anterior.

Etiqueta	Línea	Palabra	Byte
12 bits	8 bits	2 bits	2 bits

b) La memoria caché quedaría estructurada de la siguiente forma:

# de línea	Validez (1 bit)	Etiqueta (12 bits)	Datos (128 bits)				Direcciones Contenidas
			32 bits	32 bits	32 bits	32 bits	
255	:	:	:	:	:	:	4092-4095, 8188-8191,...
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	8-11, 4104-4107,...
:	:	:	:	:	:	:	4-7, 4100-4103,...
0	:	:	:	:	:	:	0-3, 4096-4099,...

Para verificar la tasa de aciertos hay que ver los accesos a cada línea y la etiqueta en cada caso:

Etiqueta	Línea	Status (1)	Status (2)	Status (3)
010	A1	F		
010	A1	A		
010	A1	A		
010	A1	A	F	F
0B0	A1	F	F	F
0B0	A1	A	A	A
000	01	F	A	A
000	01	A	A	A
000	A1	F	F	F
*C3E	C3	F	A	A
*C3E	C4	F	F	F
A0A	A1	F	F	F
A0A	F4	F	A	A
A0A	F4	A	A	A
A0A	F4	A	A	A
*82A	C4	F	F	F
A0A	A5	F	A	A

*Dirección equivalente de la palabra en bytes

F= FALLO, A= ACIERTO. Se muestran 3 de las 10 veces que se ejecuta el pedazo de código.

$$\text{TASA DE ACIERTOS} = \frac{\#ACIERTOS}{\#ACCESOS} = \frac{(7+(9*8))}{(17+(9*14))} * 100 = 55,2\%$$

c) Para aumentar el rendimiento, sin aumentar se podría implementar otra vía, aumentando así al doble la capacidad de la caché y sin necesidad de borrar. Con esta configuración se obtiene la siguiente tasa de aciertos (evaluando otra vez la corrida del programa, tomando en cuenta ahora la nueva vía):

Etiqueta	Línea	Status (1)	Status (2)	Status (3)
010	A1	F		
010	A1	A		
010	A1	A		
010	A1	A	F	F
0B0	A1	F	F	F
0B0	A1	A	A	A
000	01	F	A	A
000	01	A	A	A
000	A1	F	F	F
*C3E	C3	F	A	A
*C3E	C4	F	A	A
A0A	A1	F	F	F
A0A	F4	F	A	A
A0A	F4	A	A	A
A0A	F4	A	A	A
*82A	C4	F	F	F
A0A	A5	F	A	A

$$\text{TASA DE ACIERTOS} = \frac{\#ACIERTOS}{\#ACCESOS} = \frac{(7+(9*9))}{(17+(9*14))} * 100 = 61,5\%$$

El rendimiento mejora, aunque no de una manera muy significativa. En la siguiente página se muestra como sería la caché mejorada

# de línea	Validez (1 bit)	Etiqueta (12 bits)	Datos (128 bits)			
			32 bits	32 bits	32 bits	32 bits
255	:	:	:	:	:	:
:	:	:	:	:	:	:
:	:	:	:	:	:	:
:	:	:	:	:	:	:
:	:	:	:	:	:	:
:	:	:	:	:	:	:
:	:	:	:	:	:	:
0	:	:	:	:	:	:

Validez (1 bit)	Etiqueta (12 bits)	Datos (128 bits)			
		32 bits	32 bits	32 bits	32 bits
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:

Caché mejorada