

### Problema 3

| CADENA | r1 | w1 | r1 | w1 | r2 | w2 | r2 | w2 | r3 | w3 | r3 | w3 |
|--------|----|----|----|----|----|----|----|----|----|----|----|----|
| P1     | E  | M  | M  | M  | S  | I  | I  | I  | I  | I  | I  | I  |
| P2     | I  | I  | I  | I  | S  | M  | M  | M  | S  | I  | I  | I  |
| P3     | I  | I  | I  | I  | I  | I  | I  | I  | S  | M  | M  | M  |
| CICLOS | 90 | 1  | 1  | 1  | 90 | 60 | 1  | 1  | 90 | 60 | 1  | 1  |

} Estado del bloque referenciado en cada procesador

| CADENA | r1 | r2 | r3 | w1 | w2 | w3 | r1 | r2 | r3 | w1 |
|--------|----|----|----|----|----|----|----|----|----|----|
| P1     | E  | S  | S  | M  | I  | I  | S  | S  | S  | M  |
| P2     | I  | S  | S  | I  | M  | I  | I  | S  | S  | I  |
| P3     | I  | I  | S  | I  | I  | M  | S  | S  | S  | I  |
| CICLOS | 90 | 90 | 90 | 60 | 90 | 90 | 90 | 90 | 1  | 60 |

| CADENA | r1 | r2 | r3 | r3 | w1 | w1 | w1 | w1 | w2 | w3 |
|--------|----|----|----|----|----|----|----|----|----|----|
| P1     | E  | S  | S  | S  | M  | M  | M  | M  | I  | I  |
| P2     | I  | S  | S  | S  | I  | I  | I  | I  | M  | I  |
| P3     | I  | I  | S  | S  | I  | I  | I  | I  | I  | M  |
| CICLOS | 90 | 90 | 90 | 1  | 60 | 1  | 1  | 1  | 90 | 90 |

## Problema 4

a) MSI

| CADENA | r2 | w2 | r2 | w1 | r1 | r3 | w3 |
|--------|----|----|----|----|----|----|----|
| P1     | I  | I  | I  | M  | M  | S  | I  |
| P2     | S  | M  | M  | I  | I  | I  | I  |
| P3     | I  | I  | I  | I  | I  | S  | M  |
| CICLOS | 90 | 60 | 1  | 90 | 1  | 90 | 60 |

b) MESI

| CADENA | r2 | w2 | r2 | w1 | r1 | r3 | w3 |
|--------|----|----|----|----|----|----|----|
| P1     | I  | I  | I  | M  | M  | S  | I  |
| P2     | E  | M  | M  | I  | I  | I  | I  |
| P3     | I  | I  | I  | I  | I  | S  | M  |
| CICLOS | 90 | 1  | 1  | 90 | 1  | 90 | 60 |

La principal ventaja que aporta el protocolo MESI se observa en la segunda columna, ya que al pasar del estado E al M no es necesario ordenar ninguna transacción de bus, con el consiguiente ahorro de tiempo.

Problema 5

| CADENA     | P1 lee 00 | P2 lee 00 | P1 lee 00 | P2 esc 01 | P1 lee 01 | P2 esc 00 | P1 esc 01 | P1 lee 11 |
|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Bloque0 P1 | 00 E      | 00 S      | 00 S      | 00 S      | 00 S      | I         | I         | 11 E      |
| Bloque1 P1 | I         | I         | I         | I         | 01 S      | 01 S      | 01 M      | 01 M      |
| Bloque0 P2 | I         | 00 S      | 00 S      | 00 S      | 00 S      | 00 M      | 00 M      | 00 M      |
| Bloque1 P2 | I         | I         | I         | 01 M      | 01 S      | 01 S      | I         | I         |
| Ciclos     | 90        | 90        | 1         | 90        | 90        | 60        | 60        | 90        |

## Problema 6

a) Supongamos que el nodo peticionario es el nodo  $i$ .

1. Como el estado es  $S$ , entonces la  $M_p$  contiene una copia válida del bloque. Por tanto el nodo  $i$  hace una petición al nodo Home del bloque solicitado.
2. El nodo Home envía el bloque al nodo  $i$
3. El nodo Home actualiza el directorio para el bloque solicitado:  $Presencia[i] = 1$

b) Por cada bloque de memoria 1024 bits de presencia + 1 bit sucio  $\approx 1024$

Supongamos que el tamaño de la memoria es  $M$  bytes =  $M*8$  bits

Nº bloques mem =  $M / 64$

Tamaño del directorio  $(M/64) * 1024 =$   $M*16$  bits

Es decir, el tamaño del directorio duplica el tamaño de la memoria → la sobrecarga es del 200%

## Problema 7

Notación: NP = nº de procesadores  
NB = nº de bloques de memoria  
TM = tamaño de la memoria  
TL = tamaño de línea  
TD = tamaño del directorio

a) Sin optimización

$$\begin{aligned} \text{TD bits} &\approx \text{NP bits} \times \text{NB} \\ \text{NB} &= \text{TM bits} / \text{TL bits} \end{aligned}$$

Sustituyendo:

$$\begin{aligned} \text{TD bits} &= (\text{NP bits} / \text{TL bits}) \times \text{TM bits} & [1] \\ \text{TD bits} &= (2^{15} / (2^{11} \times 2^3)) \times \text{TM bits} \end{aligned}$$

O sea: TD = 2 TM → La sobrecarga del directorio es del 200%

b) Con TL = 2<sup>12</sup>

$$\begin{aligned} \text{TD bits} &= (2^{15} / (2^{12} \times 2^3)) \times \text{TM bits} \\ \text{TD} &= \text{TM} \rightarrow \text{La sobrecarga del directorio es del 100\%} \end{aligned}$$

c) 100 punteros por bloque

$$\begin{aligned} \text{TD bits} &= 100 \times \text{Tamaño de un puntero} \times \text{NB} \\ \text{Como hay } 2^{15} &\text{ procesadores} \rightarrow \text{el tamaño de un puntero es 15 bits} \\ \text{TD} &= 100 \times 15 \times (\text{TM bits} / \text{TL bits}); \quad \text{TL bits} = 2048 \times 8 \end{aligned}$$

$$TD = (1500 / (2048 \times 8)) \times TM = 0,0916 TM \rightarrow \text{La sobrecarga es del } 9,16\%$$

d) Como solamente dejamos sitio en el Directorio para el 50% de los bloques, por analogía con la ecuación 1:

$$TD \text{ bits} = ( (NP \text{ bits} / TL \text{ bits}) \times TM \text{ bits} ) / 2$$

$$TD \text{ bits} = ( (2^{15} / (2^{11} \times 2^3)) \times TM \text{ bits} ) / 2 = TM \rightarrow \text{Sobrecarga del } 100\%$$

## Problema 8

### 1. Tráfico con actualización:

En cada op de escritura se actualiza a través del bus la palabra correspondiente en la Mp y en todas las caches donde esté copiada.

Tráfico por iteración = 1 actualización = 1 palabra de datos (8 B) + 1 orden (6 B) = 14 B

En total: Tráfico =  $(m-1) \times 14$  B

(No se ha contado el caso en que  $i=1$ )

### 2. Tráfico con invalidación:

En cada op de escritura se manda la orden de invalidación y se envía el bloque escrito por el bus (BusRdx). Al final de las lecturas de una iteración el bloque estará en todos los procesadores en estado S, con lo que la nueva escritura de la iteración siguiente implica hacer BusRdX para invalidar todas las copias. A partir de ahí, cada lectura provoca un fallo → transferir un bloque. Luego, en cada iteración:

1 invalidación (64+6 B) y  $n-1$  fallos de lectura (64+6 B por fallo)

En total: Tráfico =  $(m-1) \times n \times 70$  B