

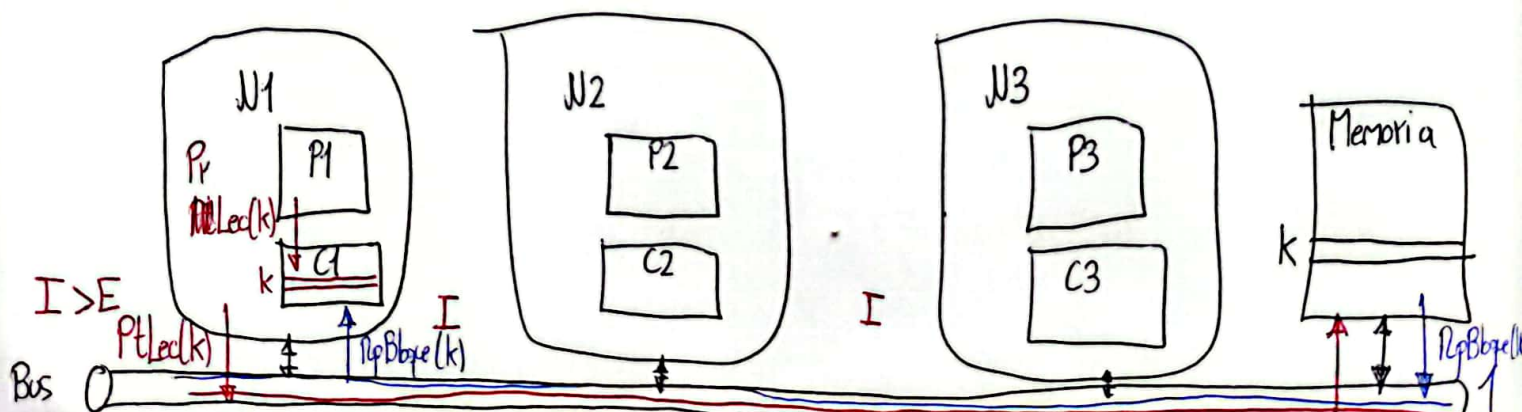
ARQUITECTURA DE COMPUTADORES

RELACIÓN DE EJERCICIOS 3

- ① En un multiprocesador SMP con 4 procesadores o nodos (N0-N3) basado en un bus, que implementa el protocolo MESI para mantener la coherencia, supongamos una dirección de memoria incluida en un bloque que no se encuentra en ninguna caché. Indique los estados de este bloque en las cachés y las acciones que se producen en el sistema ante la siguiente secuencia de eventos para dicha dirección:

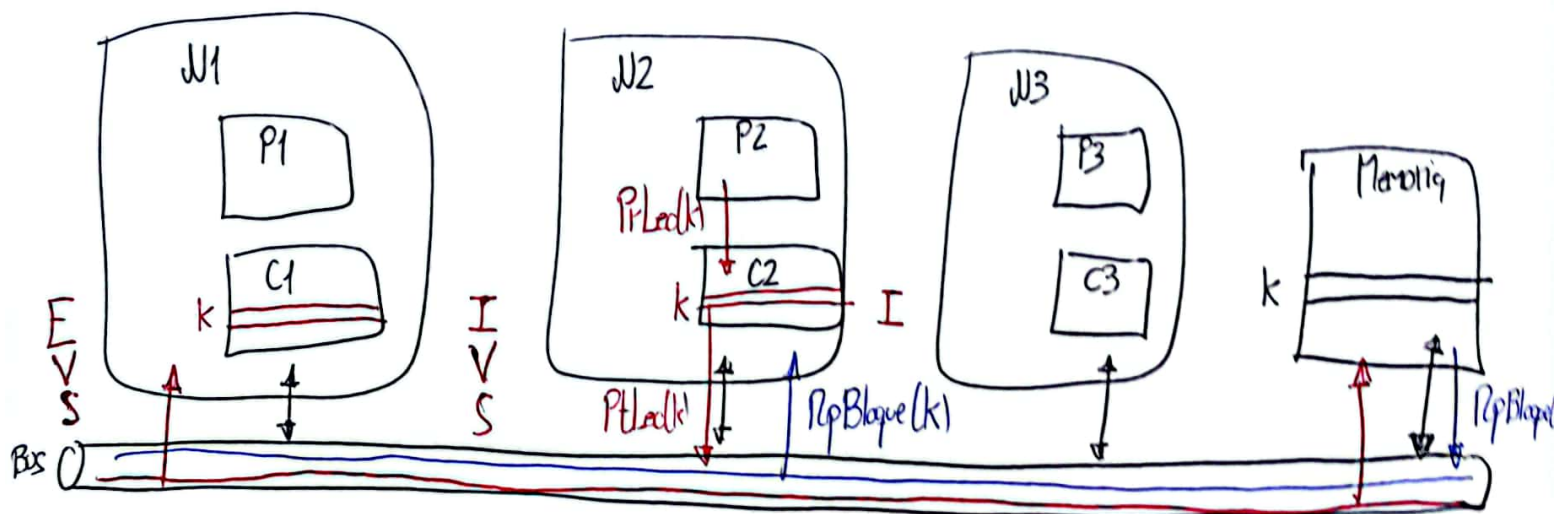
1. Lectura generada por el procesador 1.

<u>Estado inicial</u>	<u>Evento</u>	<u>Acción</u>	<u>Estado siguiente</u>
N1) Inválido N2) Inválido N3) Inválido	P1 lee k	1) N1 genera y deposita en el bus una petición de lectura del bloque k, $PtLec(k)$, porque no lo tiene válido en su caché. 2) La MP, al observar $PtLec(k)$ en el bus, genera la respuesta con el bloque, $RpBloque(k)$. 3) N1 recoge del bus la respuesta depositada por la MP, $RpBloque(k)$, el bloque entra en la caché de N1 en estado exclusivo ya que no hay copia del bloque en otra caché	N1) Exclusivo N2) Inválido N3) Inválido.



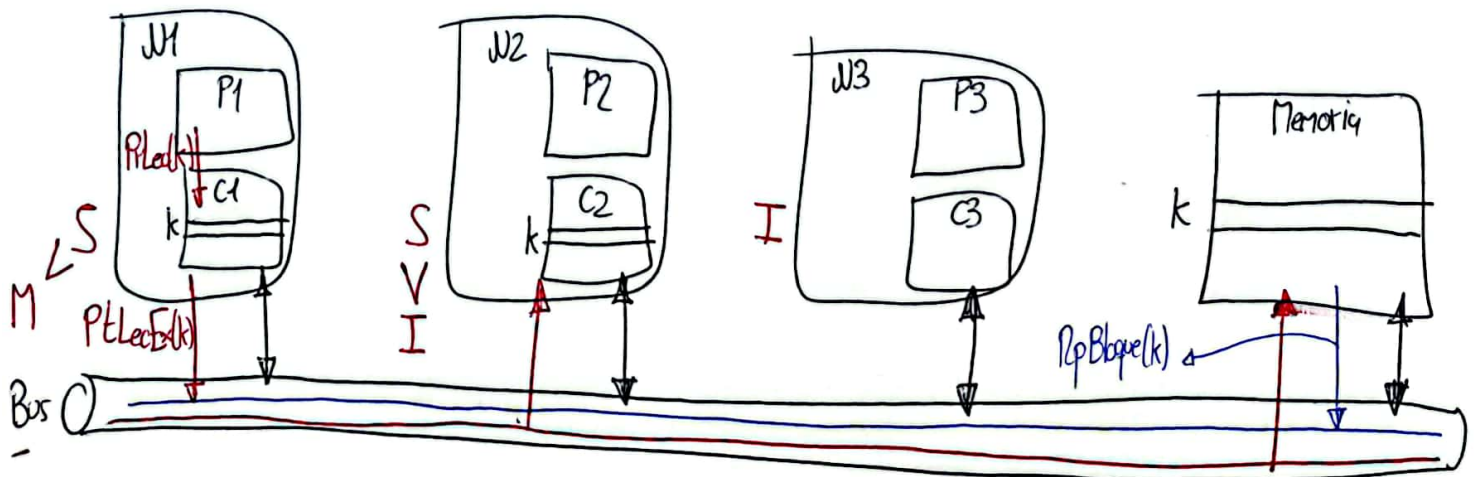
2. Lectura generada por el procesador 2.

Estado inicial	Evento	Acción	Estado siguiente
U1) Exclusivo	P2 lee k	1) U2 genera y deposita en el bus una $PT_{Lec}(k)$ porque no tiene k en su caché en estado válido	U1) Compartido
U2) Inválido		2) U1 observa $PT_{Lec}(k)$ en el bus y, como tiene el bloque en estado exclusivo, lo pasa a compartido (la copia que tiene ya no es la única válida en cachés). La MP, al observar $PT_{Lec}(k)$ en el bus, genera la respuesta con el bloque, $RP_{Bloque}(k)$.	U2) Compartido
U3) Inválido		3) U2 recoge $RP_{Bloque}(k)$ que ha depositado la memoria y el bloque entra en estado compartido en la caché de U2.	U3) Inválido



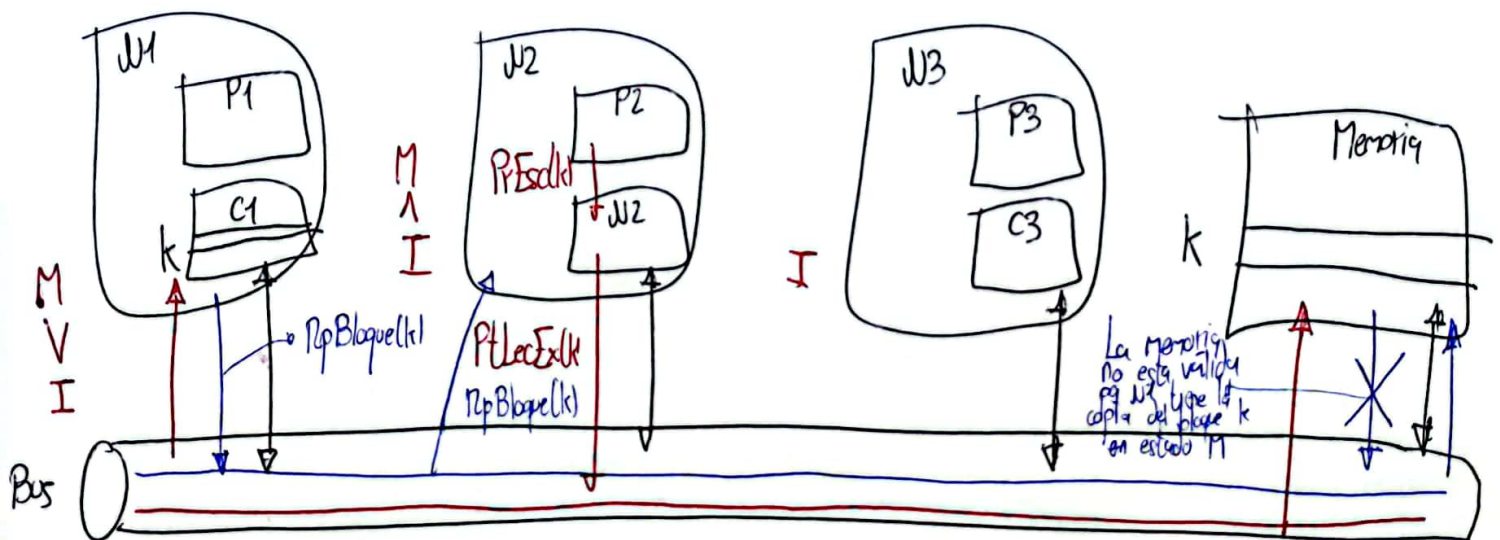
3. Nueva Escritura generada por el procesador 1.

Estado inicial	Evento	Acción	Estado siguiente
U1) Compartido U2) Compartido U3) Inválido	P1 escribe en k	1) U1 genera petición de lectura con acceso exclusivo al bloque k, $PTLecEx(k)$. U1 modifica la copia de k que tiene en su caché y lo pasa a estado modificado. 2) U2 observa $PTLecEx(k)$ y, como la petición incluye acceso exclusivo a un bloque que tiene en su caché en estado compartido, pasa su copia a estado inválido. La MP genera $RPBloque(k)$ porque observa en el bus una petición de k con lectura pero esta respuesta no se va a recoger del bus. U1 no recoge $RPBloque(k)$ porque tiene el bloque válido.	U1) Modificado U2) Inválido U3) Inválido



4. Escritura generada por el procesador 2.

Estado inicial	Evento	Acción	Estado siguiente
W1) Modificado W2) Inválido W3) Inválido	P2 escribe en k	1) N2 genera petición de lectura con acceso exclusivo de k, $PtLecEx(k)$ 2) W1 observa $PtLecEx(k)$ y, como tiene el bloque en estado modificado, inhibe la respuesta de MP y genera respuesta con el bloque $PpBloque(k)$. Además, como el proce. pide acceso exclusivo a k, invalida su copia del bloque k. 3) W2 recoge $PpBloque(k)$, introduce k en su caché, lo modifica y lo pone en estado modificado.	W1) Inválido W2) Modificado W3) Inválido



5. Escritura generada por el procesador 3. Igual que la escritura generada por P2.

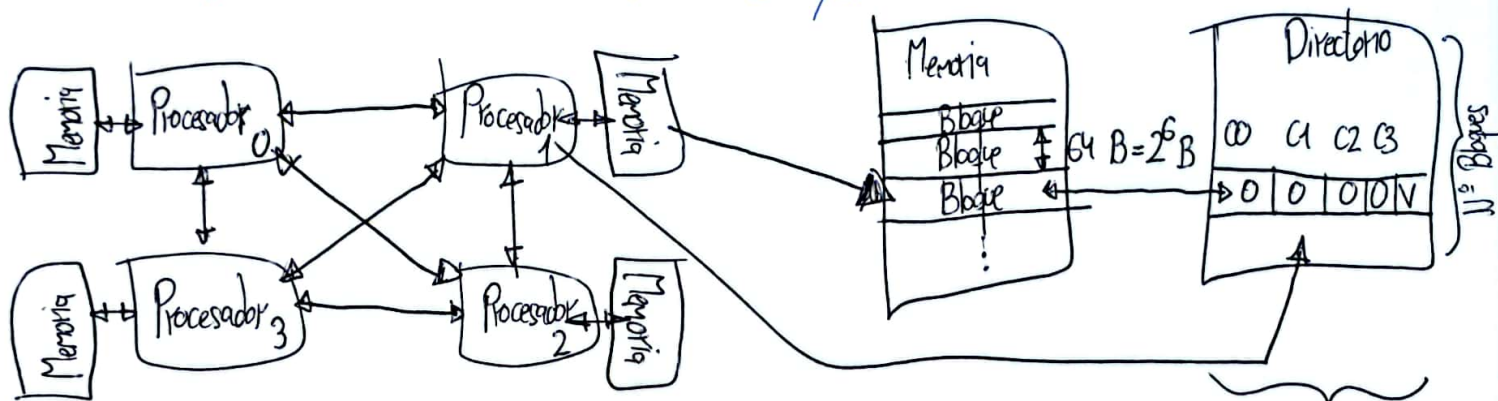
Estado final: {

- W1) Inválido
- W2) Inválido
- W3) Modificado

140

③ Suponga que en un CC-WUMA de red estática de 4 nodos (N0-N3) se implementa un protocolo MSI basado en directorios sin difusión con dos estados en el directorio (válido e inválido). Cada nodo tiene 8 GB de memoria y una línea de caché supone 64 B. Considere que el directorio utiliza vector de bits completo.

a) Calcule el tamaño del directorio en bytes.



$$TMW = \text{Tamaño de la memoria por nodo} = 8 \text{ GB} = 2^{33} \text{ B}$$

$$4 \text{ bits} + 1 \text{ bit} = 5 \text{ bits cada entrada}$$

$$T_{sd} = \text{Tamaño de un subdirectorio} = \frac{TMW}{TLC} \cdot 5 \text{ bits} = \frac{2^{33} \text{ B}}{2^6 \text{ B}} \cdot 5 \text{ bits} =$$

$$N^{\circ} \text{ Bloques} = \frac{TMW}{TLC}$$

↳ Tamaño de la línea de caché

$$= 2^{27} \text{ B} \cdot 5 \text{ bits} \cdot \frac{1 \text{ B}}{8 \text{ bits}} =$$

$$= 2^{24} \text{ B} \cdot 5 = 5 \cdot 2^{24} \text{ B} = 80 \text{ MB}$$

$$T_d = \text{Tamaño del directorio} = 5 \cdot 2^{24} \text{ B} \cdot 4 = 20 \cdot 2^{24} \text{ B} = 320 \text{ MB}$$

b) Indique cuál sería el contenido del directorio, las transiciones de estados (en caché y en el directorio) y la secuencia de papeles generados por el protocolo de coherencia en los siguientes accesos sobre una dirección D que se encuentra en la memoria del nodo 3 (inicialmente, D no está en ninguna caché).

Pendiente.

④ Supongamos que se va a ejecutar en paralelo el siguiente código (inicialmente, x e y son 0):

	<u>P1</u>	<u>P2</u>	
(1.1)	x = 1;	y = 1;	(2.1)
(1.2)	x = 2;	y = 2;	(2.2)
(1.3)	print(y);	print(x);	(2.3)

Qué resultados se pueden imprimir si:

a) Se ejecutan P1 y P2 en un multiprocesador con consistencia secuencial.

Si P1 es el primero que imprime, puede imprimir 0, 1 o 2, pero P2 solo podrá imprimir 2. Esto es así porque se mantiene orden secuencial y, por tanto, cuando P1 lee "y" (punto 1.3. en el código), ha asignado ya a "x" un 2 (punto 1.2. en el código) ya que esta asignación está antes en el código que la lectura de "y".

Si P2 es el primero que imprime podrá imprimir 0, 1 o 2, pero entonces P1 solo puede imprimir 2. Esto es así porque se mantiene orden secuencial y, por tanto, cuando P2 lee "x" (punto 2.3. en el código), ha asignado ya a "y" un 2 (punto 2.2. en el código) ya que esta asignación está antes en el código que la lectura de "x".

En conclusión, se puede obtener como resultado de la ejecución las combinaciones que hay en cada una de las siguientes líneas:

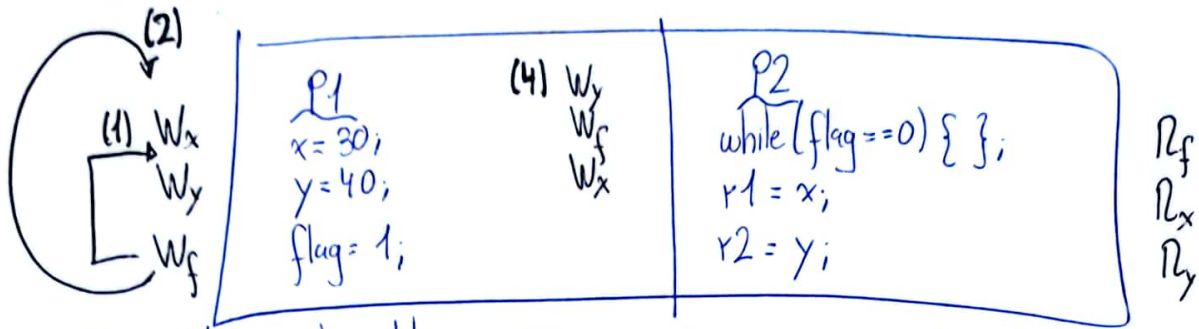
<u>P1</u>	<u>P2</u>
0	2
1	2
2	2
2	0
2	1
1	

- b) Se ejecutan en un multiprocesador basado en un bus que garantiza todos los órdenes excepto el orden $W \rightarrow R$. Esto es debido a que los procesadores tienen buffer de escritura, permitiendo al procesador que las lecturas en el código que ejecuta adelanten a las escrituras que tiene su buffer.

Si no se mantiene el orden $W \rightarrow R$, además de los resultados dados en el apartado a), los dos procesos pueden imprimir:

<u>P1</u>	<u>P2</u>
1	1
1	0
0	1
0	2
0	

5) Supongamos que se va a ejecutar en paralelo el siguiente código (inicialmente, x e y son 0, $\text{flag} = 0$):



Qué datos puede obtener P2 en $r1$ y $r2$ si (considere que el compilador no altera el código):

a) Se ejecutan P1 y P2 en un multiprocesador con consistencia secuencial.

$r1$
30

$r2$
40

b) Se ejecutan en un multiprocesador con un modelo de consistencia que relaja todos los órdenes en los accesos a memoria.

Se relaja: $W \rightarrow R, W \rightarrow W, R \rightarrow RW$

$r1$
30

$r2$
40

(1) 30

(2) 0

(3) 0

0

40