

Multiprocesadores



$$P = C_e \cdot V^2 \cdot f + V \cdot I_f$$

J.M. Llabería

- i



Capítulo 6

Protocolo de directorio con una red ordenada

.....

La utilización de un bus como red de interconexión tiene debilidades desde el punto de vista de implementación. Un bus es un elemento al que se conectan directamente los nodos¹ y la memoria. La frecuencia de funcionamiento del bus está determinada por la carga eléctrica que representan los nodos y la memoria y la longitud de los cables que permite su conexión.

Las redes de interconexión de alto rendimiento se implementan mediante interconexiones punto a punto en lugar de buses. Una conexión punto a punto permite transmitir información a mayor frecuencia que un bus. Además, varias interconexiones punto a punto ofrecen concurrencia.

Un bus facilita la implementación de protocolos de coherencia debido, a su capacidad de difusión a todos los nodos y a la serialización de todas las transacciones. Entonces, los diseñadores emulan el comportamiento de un bus mediante redes punto a punto. Sin embargo, características tales como difusión y serialización, disponibles de forma natural en un bus, son un reto en la emulación.

Por otro lado, probablemente es mejor adaptar el protocolo de coherencia a las características de la red, en lugar de emular un protocolo diseñado para otro tipo de red con características particulares.

Un protocolo de coherencia rastrea u observa el estado de los bloques. Por tanto, tiene constancia del estado de compartición. En un protocolo de observación, debido a que la información está distribuida, se exploran todos los nodos. Cada cache que tiene copia de un bloque tiene copia del estado de compartición.

1. Un nodo es el par formado por un procesador y su cache privada.

Por tanto, una deficiencia de un protocolo de observación es que todos los nodos están involucrados en una acción de coherencia aunque no tengan el bloque. Una petición se difunde a todos los nodos. El caso extremo de ineficiencia es cuando no hay copia del bloque referenciado en la transacción en ninguna cache. El conjunto de agentes observador, con un bus como red de interconexión, efectúa tantas comparaciones como nodos multiplicado por la asociatividad de la cache para comprobar que no hay copias del bloque. Por otro lado, el mecanismo de difusión no es escalable, si aumentamos el número de buses con el objetivo de incrementar el ancho de banda. Cada agente observador debe observar todas las transacciones, que son transportadas por todos los buses, aunque no tenga el bloque almacenado en cache.

Para eliminar la difusión, que representa la exploración de todos los nodos, la idea es centralizar la información de presencia de copias en los nodos en una estructura denominada directorio. Entonces, mediante la indirección que representa enviar la petición al directorio, se filtra la observación de la petición en nodos que no tengan copias del bloque. Desde el directorio sólo se envían acciones de coherencia a los nodos que tiene constancia de que tienen copia.

El diseño con directorio se adecua en mayor medida a una red con conexiones punto a punto. Sin embargo, desde el punto de vista hardware, respecto a un protocolo de observación, un protocolo de directorio representa un coste adicional. Es necesaria una estructura que almacene el estado de los bloques e identifique los procesadores que tienen copia.

Los mecanismos de coherencia que se sustentan en directorio pueden utilizar los mismos protocolos VI y MLI descritos de forma funcional en el Capítulo 2.

En este capítulo nos centraremos en utilizar una red tipo crossbar como red de interconexión. Esta red permite implementar de forma sencilla la ordenación, cuando es necesaria, de las peticiones y respuestas que realiza el directorio a los controladores de coherencia. Esta característica permite, de igual forma que en el bus, que no sea necesario que las caches respondan explícitamente las peticiones de invalidación. La idea en este capítulo es centrarse en la descripción de protocolos sencillos de directorio. La acción que denominamos transacción, que en el caso del bus incluye la petición y la respuesta, en un protocolo de directorio será una secuencia de mensajes desde un nodo al directorio y desde el directorio a los nodos, que pueden ser tanto mensajes de petición como de respuesta (Figura 6.1).

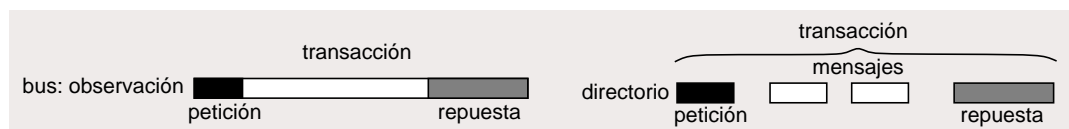


Figura 6.1 Transacción en un bus y en un esquema de directorio.

En la descripción que se efectúa en los apartados genéricos no se tiene en cuenta un único protocolo (VI, MLI). Los comentarios y mecanismo descritos pueden ser aplicables a uno de los protocolos o a ambos. La aplicación concreta a cada protocolo se efectúa en el apartado dedicado al protocolo.

DESCRIPCIÓN FUNCIONAL

Cada bloque de memoria tiene asociada una entrada en el directorio y supondremos que su gestión la realiza el controlador de memoria, el cual se encarga de mantener la coherencia en colaboración con los controladores de coherencia (Figura 6.2).

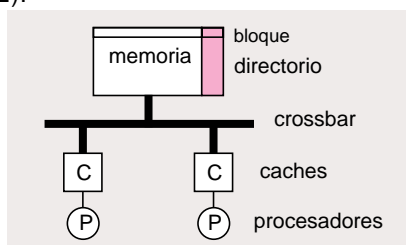


Figura 6.2 Esquema de multiprocesador con directorio.

Un protocolo de coherencia con directorio se convierte en una secuencia de mensajes punto a punto en los que sólo están involucrados los nodos “relevantes”². Podemos decir que el directorio es un filtro para reducir el número de observaciones, en las caches privadas, que se efectúan por transacción. En la parte izquierda de la Figura 6.3 se muestra un esquema linealizado del flujo de información y en la parte derecha un esquema sin linealizar.

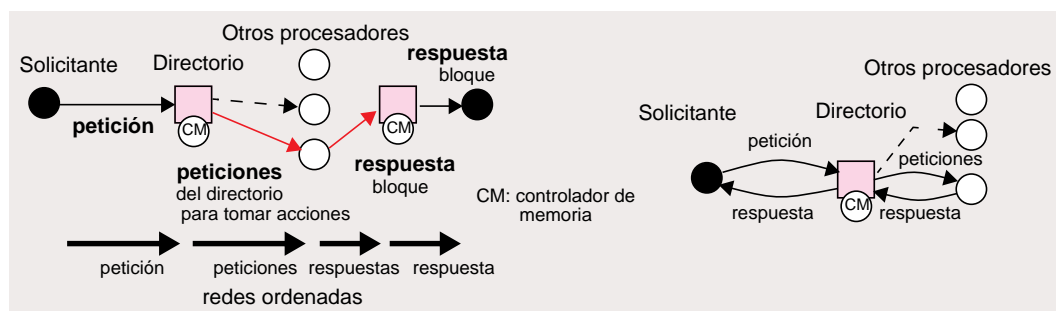


Figura 6.3 Esquema funcional de un protocolo de directorio.

- Una petición de un nodo se encamina al directorio, que es donde está la información de coherencia, lo cual representa una indirección.
2. Por relevante se entiende un nodo que tiene una copia del bloque.

- El controlador de memoria (CM), utilizando la información almacenada en el directorio, envía a los controladores de coherencia (CC) relevantes, que gestionan las caches, las ordenes oportunas para mantener la coherencia.
- Los CC atenderán las solicitudes de cambio de estado solicitadas desde el CM y en su caso, el suministro del bloque. Este suministro conlleva enviar el bloque al CM y posteriormente desde éste al nodo que lo ha solicitado.

A la vista de la descripción podemos notar que un protocolo de directorio puede representar un incremento de latencia respecto de un protocolo de observación. El CM tiene que solicitar el bloque a una cache cuando la cache lo tiene en exclusividad.

Dado un protocolo de observación, en una transacción se distingue la petición y la respuesta, pero notemos que si la petición del bloque es con intención de modificación o es una petición de escritura, hay implícitamente peticiones de invalidación en las caches que observan la transacción (Figura 2.4). En cambio, estas peticiones de invalidación en un protocolo de directorio son explícitas (desde el directorio a las caches), de la misma forma que lo es la petición de suministro, si una cache tiene el bloque en exclusividad. En un protocolo de observación las peticiones implícitas se efectúan en paralelo con la transacción. En un protocolo de directorio se efectúan de forma causal. Esto es, hay serialización, ya que hay que acceder al directorio para conocer el estado de compartición del bloque.

Información en el directorio

El directorio dispone de una entrada por cada bloque de memoria. Esta entrada se denomina vector de presencia y cada bit representa un nodo. El objetivo es identificar de forma explícita los nodos que tienen copia del bloque. La posición de los bits en el vector de presencia (VP) indica el nodo al que representan (Figura 6.4).

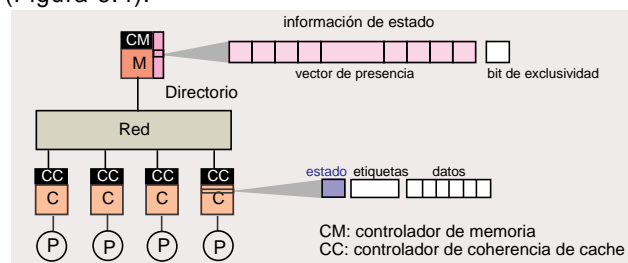


Figura 6.4 Información en el directorio.

Para la descripción del protocolo utilizaremos una organización centralizada del directorio. En un capítulo posterior, se muestra como incrementar el ancho de banda utilizando una organización distribuida.

En un protocolo de directorio, el directorio mantiene información de las copias de los bloques en las caches y su estado. Los CC emiten los accesos a memoria relevantes al directorio (Figura 6.3)³. En estas condiciones, el directorio puede ser elegido, de forma natural, como el punto de ordenación de los accesos a un bloque.

Serialización de escrituras. El directorio espera a que finalicen (y se confirmen, si es el caso) todas las acciones de coherencia⁴, inducidas por la petición que está procesando, antes de iniciar el procesado de otra petición al mismo bloque. Cuando sea necesaria la participación de terceros (otros CC), para servir una petición de coherencia, es necesario identificar que el procesado está en curso. Una forma de identificar un procesado en curso es utilizar estados transitorios en el directorio. Mientras un bloque en el directorio está en un estado transitorio, el CM no procesa otras peticiones que accedan el mismo bloque⁵.

En este capítulo efectuamos hipótesis similares a las utilizadas en el Capítulo 4.

3. En función del protocolo de coherencia: fallo de cache, escritura, exclusividad en el acceso al bloque.

4. La petición es serializada con respecto a todas las caches que tienen copia del bloque. La red de comunicación entre el CM y los CC, utilizada en este capítulo, mantiene el orden de los mensajes emitidos. En consecuencia no es necesario esperar la respuesta a peticiones de invalidación. Esta respuesta está implícita cuando el CM emite el mensaje. Por tanto, al emitir los mensajes de invalidación la escritura está consolidada.

5. En este capítulo no se observa este hecho debido a que se supone que sólo hay una transacción en curso.

El multiprocesador dispone de un directorio centralizado al que acceden los CC utilizando enlaces punto a punto. El acceso al directorio está determinado por un árbitro.

En el multiprocesador sólo existe un acceso a memoria en curso ^{6 7}.

En este capítulo los estados transitorios en los CC y en el CM se utilizan para identificar la emisión de una petición y la recepción de una respuesta.

El camino de comunicación del CM a los CC mantiene el orden de emisión de los mensajes desde el CM a los CC. Por tanto, todos los CC, involucrados en una acción de coherencia, observan los mensajes emitidos desde el directorio en el mismo orden lógico. Esta característica permite, de igual forma que en el bus, que no sea necesario que los CC respondan explícitamente las peticiones de invalidación. Posteriormente, en cada protocolo que se describe, se detalla la red utilizada.

PROTOCOLO DE DIRECTORIO CON INVALIDACION Y ESCRITURA INMEDIATA

En este apartado se describe en primer lugar la secuencia de mensajes utilizada en una transacción de coherencia. Posteriormente se describe la organización del multiprocesador y los agentes de coherencia en las caches y el CM. Seguidamente se detallan los tipos de mensajes utilizados y finalmente se describen los estados y transiciones entre estados.

A este protocolo lo denominamos protocolo de directorio VI.

Descripción funcional de la secuencia de mensajes en un transacción

La descripción funcional del protocolo de coherencia VI se ha efectuado en el Capítulo 4. En este apartado nos centraremos en la secuencia de mensajes de una transacción. Es interesante comparar la secuencia de mensajes con las acciones que se efectúan en una transacción de bus (Capítulo 4) para observar la similitud.

6. Un procesador puede utilizar, como en el caso de un bus (Capítulo 4), la señal de arbitraje para acceder al directorio, para gestionar el inicio del proceso de reinterpretación de instrucciones cuando sea necesario.

7. En un capítulo posterior analizaremos el caso de que todos los procesadores puedan tener una petición en curso. Además, el CM procesa peticiones concurrentemente o de forma paralela. En estas condiciones, un CC puede recibir una petición del CM y tener pendiente una petición al mismo bloque. De forma similar, en función del protocolo de coherencia, el CM al recibir una petición de un CC puede estar procesando una petición previa de otro CC al mismo bloque.

En la Figura 6.5 se muestra la secuencia de mensajes punto a punto en un protocolo de directorio VI. Se distinguen dos casos en función de si el CM, además de la respuesta al CC que efectúa la petición, realiza peticiones a otros CC. En cualquiera de los dos casos, una transacción es una secuencia de dos pasos: 1) petición del CC al CM y 2) respuesta del CM a este CC y en su caso, peticiones del CM a otros CC.

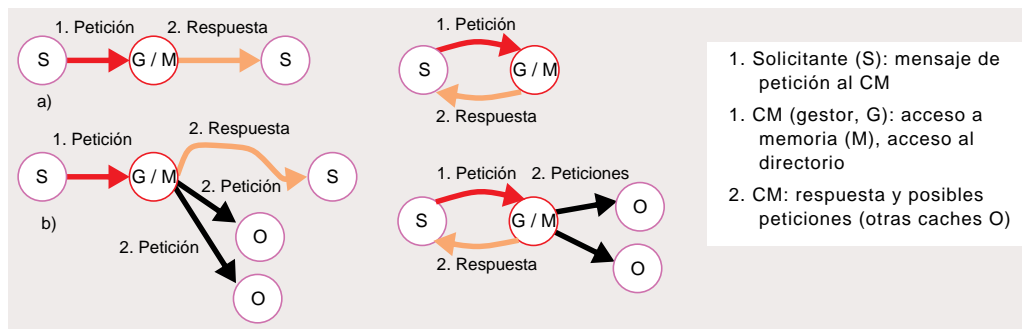


Figura 6.5 Flujo de mensajes en un protocolo de directorio VI entre el CC y el CM y el CM y otros CC. Izquierda: representación lineal. Derecha: representación no lineal (los nodos no se replican).

En una transacción de lectura se utiliza la secuencia de mensajes a) de la Figura 6.5. Cuando no hay copias del bloque en otras caches, en una transacción de escritura se utiliza la secuencia de mensajes a). En caso contrario se utiliza la secuencia de mensajes b)

Organización del multiprocesador

En la Figura 6.6 se muestra la organización del multiprocesador. En el camino de las caches a memoria distinguimos una cola de peticiones (CP). En esta cola se almacenan las peticiones de los CC, los cuales utilizan caminos independientes punto a punto para transmitir los mensajes hacia la CP. La salida de esta cola es un camino a memoria. El CM procesa en serie los mensajes almacenados en la CP⁸.

De forma funcional, desde el CM a los CC se distinguen dos caminos independientes a cada uno de ellos (no se muestran en el dibujo). Uno de ellos, denominado de respuestas, lo utiliza el CM para responder a una petición de un CC. El otro camino, denominado de peticiones, lo utiliza el CM para efectuar peticiones de coherencia a uno o varios CC al procesar una petición de otro CC.

8. Mientras se suponga que sólo existe un acceso a memoria en el multiprocesador la CP no tiene utilidad.

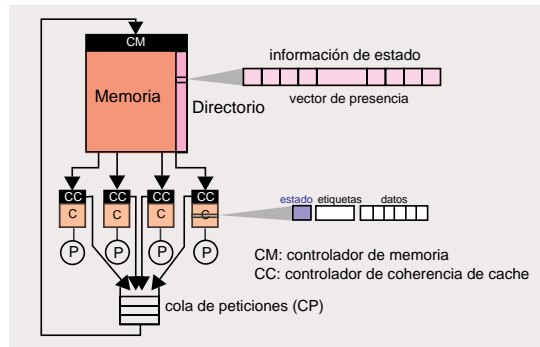


Figura 6.6 Detalle de las redes de interconexión.

En resumen, hay una red desde los CC al CM, a la que denominaremos red de ida (RI). De forma funcional, hay dos redes desde el CM a los CC, una de ellas transporta mensajes de respuesta y la otra transporta mensajes de petición. En un protocolo de directorio VI estas dos últimas redes se pueden implementar, de forma sencilla, mediante una única red física, a la que denominaremos red de vuelta (RV, Figura 6.9). Posteriormente, al detallar el protocolo se observa que, al procesar una petición, el CM puede emitir un mensaje de respuesta y varias peticiones y la intersección de los destinos es el conjunto vacío. Esto es, el CC que recibe el mensaje de respuesta no recibe un mensaje de petición.

A partir de la información ubicada en el directorio, el CM sólo envía peticiones de coherencia, para responder a la petición que está procesando, a los CC que tengan copia del bloque y necesiten efectuar alguna acción.

En los CC de las caches hay dos agentes: a) procesador y b) observador. Los seguimos identificando de la misma forma que en un protocolo de difusión. El agente observador procesa (observa) peticiones del CM.

En un protocolo VI la memoria siempre está actualizada. Por tanto, en una petición de lectura, efectuada por un CC, siempre se suministra el bloque que está almacenado en memoria. Además, con el objetivo de identificar a los nodos que tienen copia del bloque, es suficiente que el directorio tenga un VP por bloque.

Notemos que en el protocolo de directorio VI no se utiliza la fase de respuesta de los CC identificada en la Figura 6.3.

Camino de datos en el controlador de coherencia

En la Figura 6.7 se muestra un esquema del camino de datos de un CC. Se identifican los agentes procesador y observador. El agente observador utiliza una copia del campo etiqueta para eliminar riesgos estructurales con accesos del procesador al campo etiquetas. Por la misma razón, el campo estado tiene dos caminos de acceso, uno para cada agente.

El agente procesador atiende las peticiones del procesador y las respuestas del CM, que llegan por la RV. Cuando el agente procesador no puede servir una petición del procesador emite una petición al CM utilizando la RI. Al recibir la respuesta a una petición, inducida por un fallo de lectura de cache, actualiza la cache, el estado y el duplicado de etiquetas. Al recibir una respuesta a una petición de escritura actualiza la cache con el dato (no se muestra en la Figura 6.7), si el bloque está almacenado en cache.

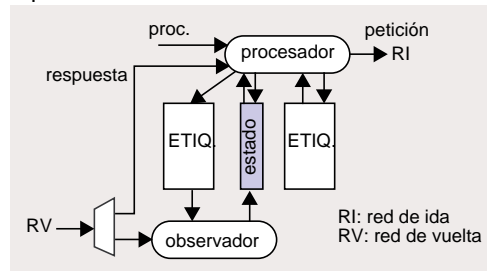


Figura 6.7 Esquema del camino de datos en el CC.

El agente observador procesa las peticiones del CM. Para ello, accede al duplicado de etiquetas con el objetivo de conocer el contenedor de cache, en el cual se almacena el bloque referenciado en la petición. La única petición del CM es observación de escritura, la cual requiere invalidar el bloque. Por tanto, es suficiente que el agente observador actualice el campo estado.

Incluido en el agente procesador está el autómata de reemplazo. Su función es seleccionar un bloque, almacenado en cache, para liberar un contenedor, en el cual se ubicará el bloque accedido en una lectura, que ha detectado un fallo al acceder a la cache. Cuando se conoce el bloque que se expulsa, el autómata procesador emite una petición de expulsión al CM. Esta petición es para mantener actualizada la información en el directorio.

En un acceso a memoria que es fallo en cache y que determina la expulsión de un bloque, en primer lugar se realizan las acciones encaminadas a expulsar el bloque. Posteriormente se inician las acciones para servir el fallo de cache.

Camino de datos en el controlador de memoria

En la Figura 6.8 se muestra un esquema del camino de datos del CM. El CM procesa una petición accediendo al directorio y a memoria y emite una respuesta, y en su caso peticiones a los CC, utilizando conexiones punto a punto dedicadas. La red utilizada para ello es la RV. Recordemos que el destinatario de la respuesta no es destinatario de una petición. Además, el CM actualiza el directorio.

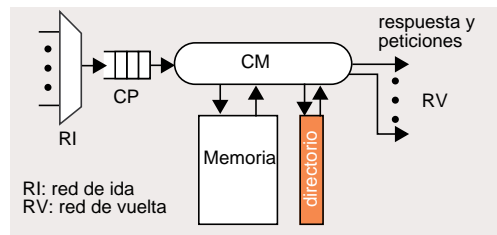


Figura 6.8 Esquema del camino de datos en el CM.

Mensajes del protocolo

Para mantener la coherencia se transmiten mensajes de los CC al CM, que son peticiones. Desde el CM se transmiten mensajes de petición y respuesta a los CC.

En la Figura 6.9 se muestra un esquema genérico de la transmisión de mensajes de peticiones y respuestas entre los CC y el CM. Notemos que el esquema se ha linealizado para que todos los mensajes fluyan de izquierda a derecha. Como los CC emiten peticiones y reciben peticiones y respuestas, en el esquema se dibujan más de una vez.

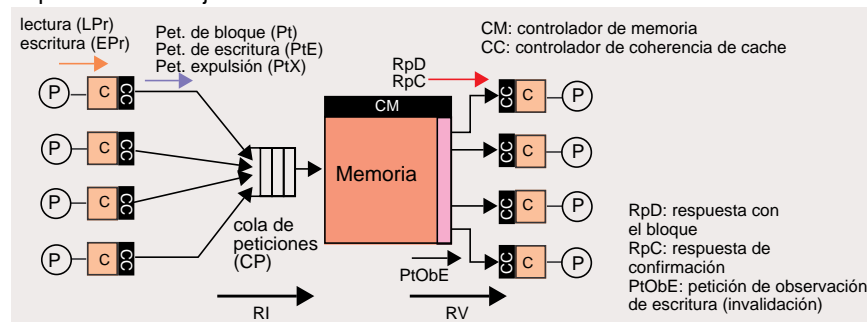


Figura 6.9 Esquema de la transmisión de mensajes en un protocolo de directorio VI.

En la Tabla 6.1 se describen las peticiones del procesador al CC. El procesador efectúa dos operaciones: a) lectura (load) y b) escritura (store). En un fallo de lectura y en una escritura se accede a memoria.

Tabla 6.1 Peticiones del procesador, respuestas y acciones del CC.

Controlador de coherencia (CC)	Controlador de memoria (CM)	Comentario
Mensajes de petición al CM y acciones Pt: petición de bloque <div> <div>op</div> <div>dirección</div> <div>ld</div> </div>	Mensajes de respuesta RpD: respuesta con el bloque <div> <div>ld</div> <div>bloque</div> </div>	Se lee el bloque de datos de memoria, se actualiza el directorio y se suministra el bloque.
PtE: petición de escritura de un dato <div> <div>op</div> <div>dirección</div> <div>dato</div> <div>ld</div> </div>	RpC: respuesta de confirmación <div> <div>ld</div> <div>confirmación</div> </div>	Se actualiza memoria con el dato, el CM envía un mensaje de respuesta y si es el caso, peticiones de observación de escritura a otros CC. También se actualiza el directorio.
CcRe: expulsión de un bloque.		Se invalida la información del contenedor.
PtX: petición (notificación) de expulsión <div> <div>op</div> <div>dirección</div> <div>ld</div> </div>	RpX: respuesta de confirmación <div> <div>ld</div> <div>confirmación</div> </div>	El CM, después de actualizar el vector de presencia, emite una respuesta de confirmación.

Protocolo de directorio con una red ordenada

En la Tabla 6.3 se describen las peticiones del CM a los CC y la acción del CM para tener actualizada la información de presencia de los bloques. El CM envía mensajes de petición de observación de escritura, si es el caso, cuando procesa una petición de escritura.

Controlador de memoria	Controlador de coherencia (CC)	Comentario
Mensajes de petición a CC y acciones	Respuesta	
PtObE: petición de observación de escritura <div>op</div> <div>dirección</div>	La red de transmisión de mensajes del CM a los CC mantiene el orden. Por tanto, no es necesario la respuesta	Petición para que un CC concreto invalide un bloque.
Actualización del directorio		Mantiene actualizada la información de compartición de los bloques.

Tabla 6.3 Petición del CM a los CC.

Estados y transiciones

En un protocolo de directorio hay que describir los estados de un bloque en las caches y los estados en el directorio.

Directorio. Los estados de un bloque en el directorio son: no presente (NP) y presente (Pr). El primero indica que no existe copia del bloque en las caches, mientras que el segundo indica que existe al menos una copia. Estos estados están codificados en el VP asociado al bloque. Si todos los bits del VP están desactivados el estado es NP. En caso contrario el estado es Pr.

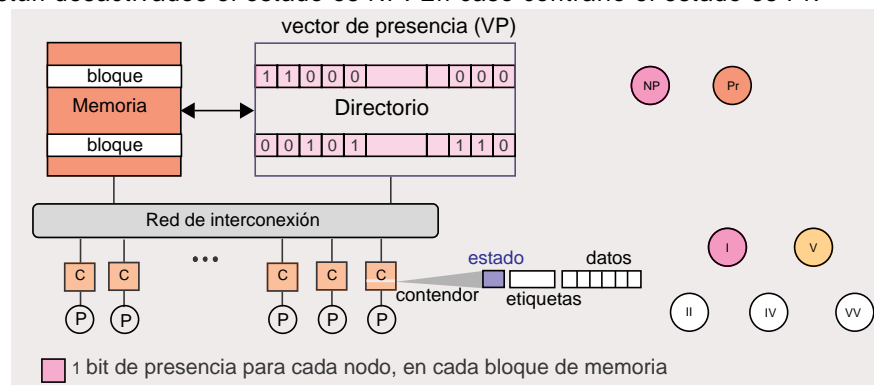


Figura 6.10 Protocolo de directorio VI. Estados de un bloque en el directorio y en un contenedor de caches.

Cache. En una cache, un bloque puede tener dos estados estables y tres estados transitorios. Los estados estables son: Inválido (I) y Válido (V). Los estados transitorios son: II, IV y VV. En este apartado, los estados transitorios se utilizan para distinguir entre la emisión de la petición (mensaje de petición)

y la recepción de la respuesta (mensaje de respuesta). El etiquetado de los estados transitorios sigue las reglas utilizadas al describir los protocolos de observación en el Capítulo 4.

Para describir el protocolo se utilizan las dos peticiones del procesador que requieren acceder al directorio, las cuales son fallo de lectura (load, Pt) y cualquier escritura (store, PtE). En un acierto de lectura el estado en la cache no se modifica. Posteriormente se detallan las transiciones entre estados en una expulsión de un bloque (PtX).

Al describir cada una de las peticiones indicadas se mostrarán las transiciones entre estados efectuadas por el CC que efectúa la petición (agente procesador), en el directorio gestionado por el CM y en otros CC (agente observador).

Fallo de lectura

En la Figura 6.11 se muestra el flujo de mensajes en un fallo de lectura. El CC, después de detectar el fallo, emite un mensaje con una petición de bloque, que se almacena en la CP. Entonces, el CM extrae el mensaje de la CP, lee del directorio el VP asociado al bloque, accede a memoria para leer el bloque y actualiza el VP. Finalmente el CM emite un mensaje de respuesta con el bloque al CC que ha efectuado la petición.

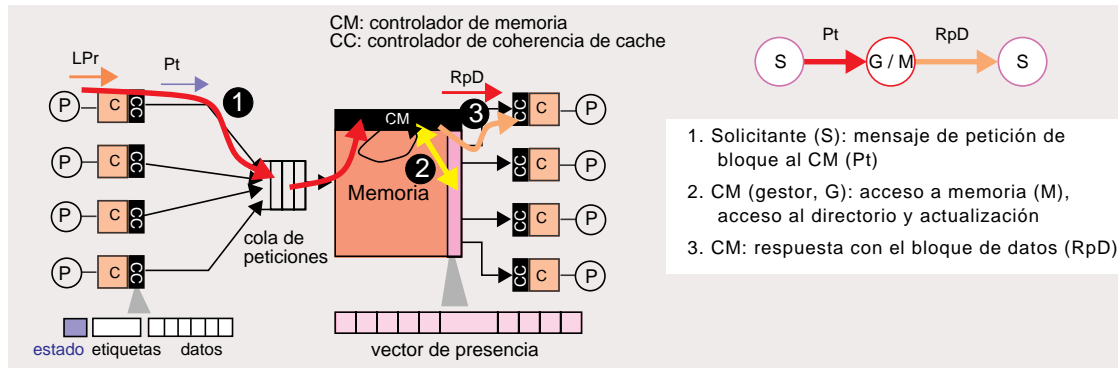


Figura 6.11 Fallo de lectura. Esquema de transmisión de mensajes.

En la Figura 6.12 se muestran las transiciones entre estados de un bloque en un fallo de lectura: a) el CC que efectúa la petición (solicitante), b) el directorio (CM) y c) otros CC. Además, se muestra la actualización del VP, asociado al bloque, que está almacenado en el directorio.

Después de emitir el mensaje de petición de lectura, el CC (agente procesador) establece que el estado del bloque es IV (estado transitorio). El bloque permanece en este estado hasta que se recibe la respuesta del CM, la cual es procesada por el agente procesador del CC, que modifica el estado del bloque a válido.

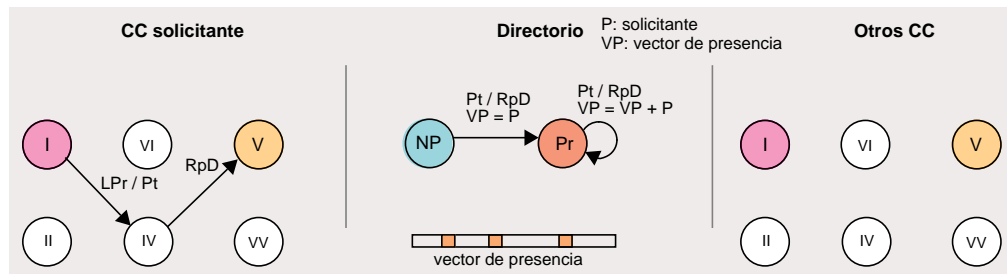


Figura 6.12 Fallo de lectura. Transiciones entre estados.

El CM, al recibir el mensaje de petición de lectura, accede al directorio para leer el VP, a memoria para leer el bloque y después emite la respuesta. Además, actualiza el VP. Al conjunto de bits activados, que indican copia del bloque en otras caches (VP), se añade el bit asociado al procesador solicitante (P). En la Figura 6.12 se indica como $VP = VP + P$ o $VP = P$, siendo P el procesador cuyo CC efectúa la petición. Los otros CC no reciben ninguna petición de coherencia desde el CM, ya que la petición es de lectura y la memoria, en el protocolo de directorio VI, siempre está actualizada.

Escritura

En la Figura 6.13 se muestra el flujo de mensajes en una escritura. El CC emite un mensaje con una petición de escritura, que se almacena en la CP. Seguidamente, el CM extrae el mensaje de la CP y accede a memoria para actualizar el dato referenciado, lee el VP correspondiente al bloque, para conocer las caches que tienen copia del bloque, y lo actualiza. Finalmente el CM emite un mensaje de respuesta (RpC) al CC que ha efectuado la petición y mensajes de petición de observación de escritura (PtObE) a los CC que tienen almacenada una copia del bloque en su cache, excluyendo, si es el caso, al CC que ha efectuado la petición de escritura.

Los CC (agente observador) que reciben una petición de observación de escritura cambian el estado del bloque de válido a inválido.

Expulsión de un bloque de cache

Las expulsiones se notifican al directorio para mantener actualizado el VP. Esta notificación mantiene en el directorio, de forma precisa, las copias existentes del bloque en las caches.

En la Figura 6.15 se muestra el flujo de mensajes en la expulsión de un bloque de una cache. El CC emite un mensaje de petición de expulsión (PtX). El CM procesa el mensaje desactivando el bit correspondiente al CC en el VP y envía un mensaje de confirmación (RpX).

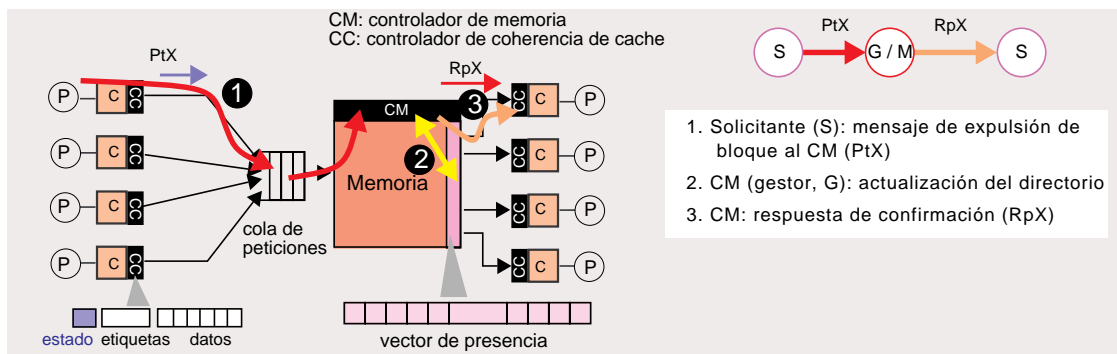


Figura 6.15 Expulsión de un bloque. Esquema de transmisión de mensajes.

En la Figura 6.16 se muestran las transiciones entre estados de un bloque al efectuarse una expulsión en: a) el CC que efectúa la petición (solicitante), b) el directorio y c) otros CC. Además, se muestra la actualización del VP almacenado en el directorio. Un CC después de enviar el mensaje de petición de expulsión (PtX) cambia el estado a VI. El CM procesa el mensaje y cambia el estado a NP o permanece en el estado Pr, en función de si el mensaje provenía del único CC cuya cache almacenaba el bloque o no. Los otros CC no se ven afectados por la actualización del directorio.

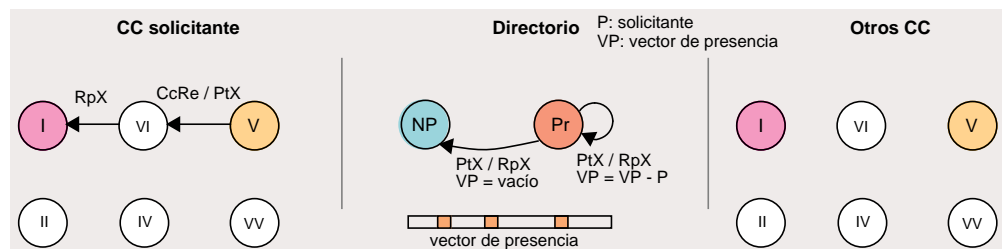


Figura 6.16 Expulsión de un bloque de cache. Transiciones entre estados.

Diagrama completo de estados y transiciones

En las Figura 6.17 y Figura 6.18 se muestran, respectivamente, los diagramas de transiciones entre estados de un bloque en cache y en el directorio. En el CC se distinguen, en diagramas separados, las transiciones iniciadas por el agente procesador y el agente observador (Figura 6.17). En el primer diagrama están incluidas las peticiones del procesador que no requieren iniciar transacciones explícitas de coherencia.

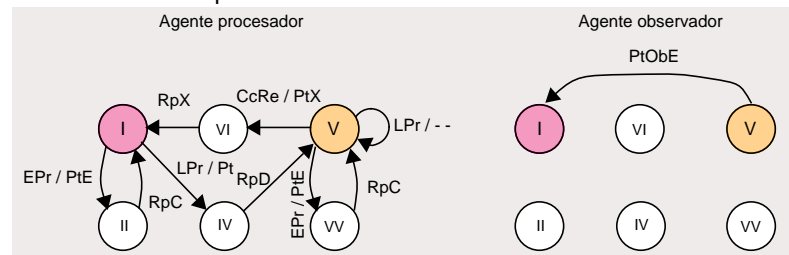


Figura 6.17 Protocolo de directorio VI. Estados y transiciones entre estados de un bloque en una cache.

En el diagrama del CM (Figura 6.18) hay que distinguir dos casos al procesar las peticiones PtE y PtX de un CC. En una petición PtE, el estado final depende de si el CC, que efectúa la petición, está en el VP o no está. En una petición PtX, la condición es si el CC es el único que está en el VP o hay más CC.

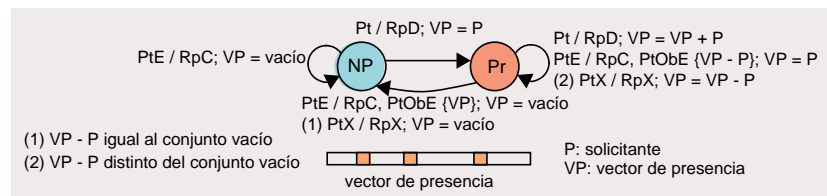


Figura 6.18 Protocolo de directorio VI. Estados y transiciones entre estados de un bloque en el directorio.

Tablas de estados y transiciones

En la Tabla 6.4 se muestran en formato tabla los estados y las transiciones entre estados de un bloque en una cache. Las casillas que no contienen información indican un error. En un estado determinado no puede llegar el evento que determina la casilla correspondiente en el cruce.

			Eventos del procesador y reemplazo			Eventos externos (respuestas y petición)			
			LPr	EPr	CcRe	RpD	RpC	RpX	PtObE
Estados	Estables	I	Pt; IV	PtE; II					
		V	--; V	PtE; VV	PtX; VI				--; I
	transitorios	IV				--; V			
		VV					--; V		
		VI						--; I	
		II					--; I		

Tabla 6.4 Protocolo de directorio VI. Tabla de estados y transiciones de un bloque de cache.

En la Tabla 6.5 se muestran en formato tabla los estados y transiciones entre estados de un bloque en el CM. En el evento PtE se distinguen dos casos, en función de si el CC está o no está presente en el vector de presencia. En el evento PtX se distingue el caso de que el CC sea el único que está en el VP o haya más CC.

		Eventos del controlador de coherencia					
		Pt	VP = vacío	PtE	PtX		
				P ∈ VP	P ∉ VP	VP = P	P ∈ VP VP ≠ P
Estados	Estables	NP	RpD; Pr, VP = P	RpC; NP, VP = vacío			
		Pr	RpD; Pr, VP = VP + P	RpC, PtObE {VP - P}; Pr, VP = P	RpC, PtObE {VP}; NP, VP = vacío	RpX; NP, VP = vacío	RpX; Pr, VP = VP - P

Tabla 6.5 Protocolo de directorio VI. Tabla de estados y transiciones de un bloque en el directorio.

Representación de transacciones y transiciones entre estados

En este apartado se muestran tres formas de representar las transacciones y transiciones al ejecutar una secuencia de accesos a memoria: a) en formato tabla, b) mediante un diagrama temporal y c) mediante un diagrama temporal simplificado. Finalmente se muestra, mediante varios gráficos, una animación.

Representación en formato tabla

Cada fila representa un acceso a memoria y no se gestiona el siguiente acceso hasta que finaliza el anterior. En una fila de izquierda a derecha, después de la instrucción de acceso a memoria, se especifica:

- 1** El estado transitorio del bloque en la cache del procesador cuyo acceso a memoria requiere una transacción.
- 2** La petición de acceso a memoria en la red que interconecta los CC y el CM (red de ida, RI).
- 3** El nombre de la variable accedida y el vector de presencia (VP) en la memoria. La posición más a la izquierda representa la cache con el menor ordinal y sucesivas posiciones, contiguas en el VP, representan caches con ordinales crecientes y consecutivos.
- 4** Quién suministra el dato o bloque (cache o memoria).
- 5** La respuesta del CM en la red que interconecta el CM y los CC (red de vuelta, RV). En la siguiente fila, y la misma columna, se especifican las peticiones del CM, si es el caso, a los CC. En este caso, también se enumeran los CC que reciben la petición.
- 6** Para las caches donde se modifica la información almacenada, el nombre de la variable y el estado estable del bloque al finalizar el acceso a memoria.

Ejemplo. En la Tabla 6.6 se muestra una secuencia de accesos a memoria realiza por tres procesadores. En este ejemplo suponemos que las variables t y u no están almacenadas en cache y los bloques que las contienen se almacenan en contenedores distintos de cache.

	C 1	C 2	C 3	Red	mem.				Red	C 1	C 2		C 3	
acceso	est.	est.	est.	RI	var.	VP	sum.	RV	var.	est.	var.	est.	var.	est.
1. P1 load t	IV			Pt	t	1, 0, 0	mem.	RpD	t	V				
2. P2 store u		II		PtE	u	0, 0, 0	C2	RpC			u	I		
3. P1 load u	IV			Pt	u	1, 0, 0	mem.	RpD	u	V				
4. P3 load u			IV	Pt	u	1, 0, 1	mem.	RpD					u	V
5. P1 store u	VV			PtE	u	1, 0, 0	C1	RpC	u	V				
								PtObE 3					u	I

Tabla 6.6 Protocolo de directorio VI. Formato tabla. Secuencia de accesos a memoria.

El primer acceso a memoria es una instrucción load y se produce un fallo en la cache C1. Al emitir el mensaje de petición de lectura de bloque, el estado transitorio del bloque es IV. La variable accedida es t y el VP en el directorio sólo indica la presencia del bloque en la cache C1. La RV transporta un mensaje de respuesta de datos y el estado estable del bloque en la cache C1 al finalizar el servicio del fallo es V.

El segundo acceso a memoria es una instrucción store. El bloque no está almacenado en cache. Por tanto, el estado transitorio es II y el mensaje que se transmite por la red RI es PtE. No hay copia del bloque referenciado en ninguna cache. El dato se transmite del CC a memoria. La respuesta que se transmite por la red RV es una confirmación de escritura (RpC) y el estado del bloque en la cache, al recibir la respuesta, es I.

El tercer acceso a memoria determina que el CC1 emita una petición Pt. La variable accedida es u. El comportamiento es mimético al primer acceso a memoria del mismo procesador. El cuarto acceso a memoria, efectuado por el procesador P3, es mimético al anterior con la salvedad de que el VP indica copias del bloque en las cache C1 y C3.

El quinto acceso a memoria es una instrucción store en el procesador P1. Al acceder a cache se determina que es acierto. Por tanto, el estado transitorio es VV. El mensaje que transmite el CC al CM es PtE. En el directorio el VP se actualiza para indicar que hay copia del bloque sólo en la cache C1. Las otras copias, que había del bloque en otras caches, se invalidan mediante la emisión, por parte del CM, de un mensaje PtObE a los respectivos CC. En la tabla se representa este hecho utilizando la siguiente fila e identificando la petición del CM y el ordinal de los procesadores en la columna de la RV. El estado estable del bloque en la cache C1 es V y en la cache C3 es I.

Diagrama temporal

En la Figura 6.19 se muestran varios ejemplos de diagramas temporales. Aunque se asocian fases de una acción de coherencia (transacción) a ciclos estos no son representativos.

En un diagrama temporal, los mensajes correspondientes a una transacción se representan en la fila asociada al acceso a memoria. Antes de utilizar la RI para transmitir el mensaje se representa el arbitraje. Esta fase se utiliza para mostrar la ordenación de los mensajes. Tengamos en cuenta que, en un instante determinado, sólo se puede encolar un mensaje en la CP. La espera para acceder a la red se indica representando en ciclos consecutivos la fase de arbitraje (arb)⁹.

Después de utilizar la RI se representa el acceso a memoria y al directorio (M). Seguidamente se representa la utilización de la RV y finalmente se representa la recepción de los mensajes de respuesta y de petición en los CC.

9. Cuando en el modelo de multiprocesador sólo existe una petición en curso esta fase no es necesaria. Sin embargo, para utilizar la misma representación en este caso y en un próximo capítulo, donde se analizan peticiones concurrente, la mantendremos.

Cuando el CM emite una respuesta y peticiones se utilizan dos filas. En la primera fila se indica la respuesta y en la segunda fila la secuencia de ordinales de los CC que reciben la petición (PtObE). Notemos que antes de utilizar la RV también se representa una fase de arbitraje¹⁰.

En la parte central de un diagrama temporal se muestran, en las distintas fases de una transacción, los estados de un bloque en las caches y en el directorio. En la fase de arbitraje de la RI (arb), que se corresponde con el arbitraje para enviar un mensaje de un CC al CM, se indica el estado transitorio del bloque en el CC al emitirse la petición.

En la fase correspondiente al acceso al directorio (M) se indica la actualización del VP, si es el caso. El VP se representa utilizando los ordinales de los procesadores o CC cuyos bits están activados (fila M).

En la fase de respuesta (D, C, X) y de recepción de peticiones del CM en un CC se indica el estado estable de los bloques en las caches. En fases donde no se indica explícitamente el estado, éste es el último que se ha indicado explícitamente en la misma fila. Los acrónimos D, C y X indican respectivamente bloque, confirmación de escritura y respuesta a una petición de expulsión.

En la parte inferior del diagrama temporal se muestra el acrónimo de los mensajes emitidos por un CC o por el CM. El mensaje se indica en la fase arb.

En las fases M y recepción de respuesta se utilizan líneas continuas finalizadas con una flecha para indicar la relación entre la fase y la actualización del estado en el directorio y en otros CC respectivamente (peticiones PtObE).

En el ejemplo de la izquierda de la Figura 6.19 se muestra un fallo de lectura. Antes de ejecutar la secuencia de accesos hay copia del bloque en la cache C2. En la fase arb se indica el estado transitorio en la cache del CC que envía el mensaje. En la fase M se indica que en el VP están activados los bits correspondientes a las caches C1 y C2. En la fase de respuesta y peticiones del CM se indica que el estado del bloque en la cache C1 es V.

En el ejemplo del centro de la Figura 6.19 se muestra la ejecución de una instrucción store cuando se produce acierto en cache. Suponemos que no hay copias del bloque en otras caches. El estado transitorio del bloque, al emitir el CC la petición de escritura, es VV y el estado final es V. El contenido del VP del bloque en el directorio no se modifica.

10. Esta fase no es de utilidad cuando sólo existe un acceso memoria en el multiprocesador. Sin embargo, para utilizar la misma representación que cuando se utiliza un multiprocesador con varios módulos de memoria, la cual se efectúa en un próximo capítulo, la mantendremos.

En el ejemplo de la derecha de la Figura 6.19 se muestra la ejecución de una instrucción store, cuando hay copias del bloque en otras cache y no hay copia en la cache cuyo procesador ejecuta la instrucción. Para representar, en el mismo ciclo, la emisión de mensajes de respuesta y de petición desde el CM se utilizan dos filas. En la segunda fila se explicita la secuencia de ordinales de los CC que reciben la petición PtObE. Notemos que el VP se actualiza en la fase M y el estado de las copias del bloque en la fase de recepción de respuesta y petición.

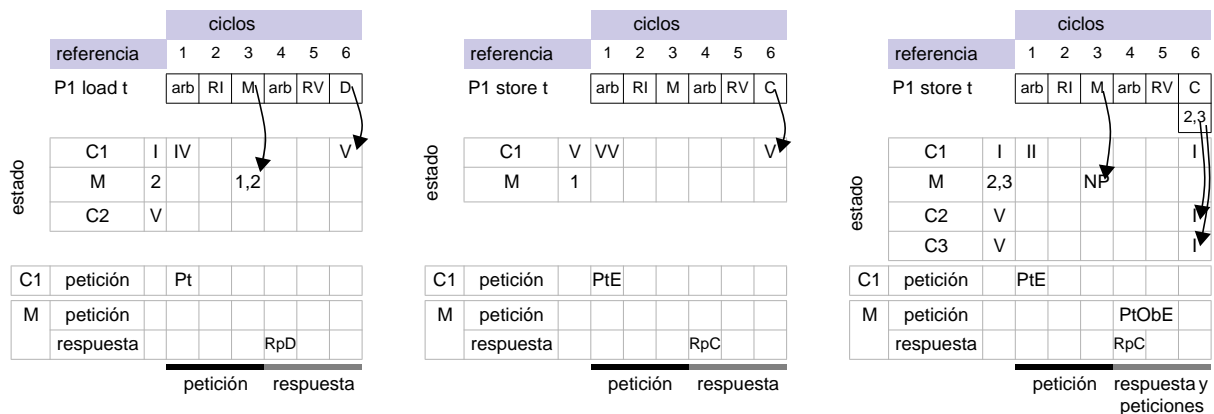


Figura 6.19 Protocolo de directorio VI: diagramas temporales de acciones de coherencia.

Ejemplo. En la Figura 6.20 se muestra el diagrama temporal de la secuencia de accesos mostrada en la Tabla 6.6. En la parte central de la Figura 6.20 se especifican las caches de los procesadores, el directorio (M) y las variables. En estas filas se representa el estado, en cada cache y en el directorio, de la forma descrita en los ejemplos de la Figura 6.19. En la parte inferior se representan las peticiones de los CC y las peticiones y respuestas del CM.

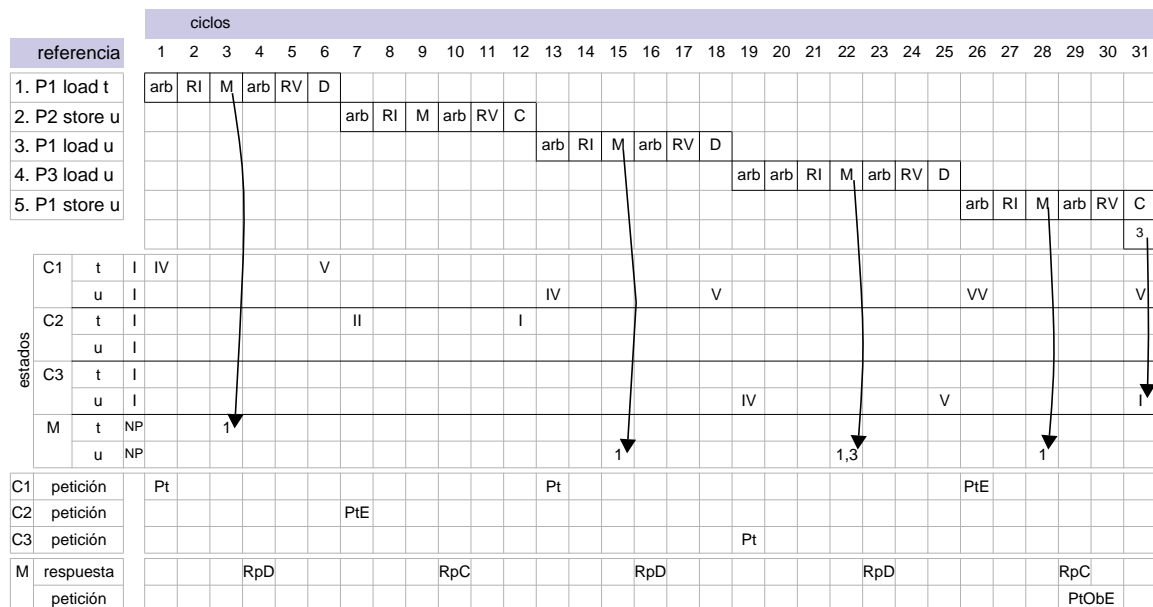


Figura 6.20 Protocolo de directorio VI. Diagrama temporal de una secuencia de accesos a memoria.

Diagrama temporal simplificado

La idea de este diagrama es representar de forma más esquemática los mensajes y la temporalidad de los mismos¹¹.

Cada CC y el CM se representan por una línea vertical¹². El tiempo avanza de arriba hacia abajo.

Un trazo continuo entre un CC (fuente) y el CM (destino), finalizado con una flecha al llegar al CM, indica un mensaje del CC al CM. Además, el origen del trazo es temporalmente previo a la finalización del mismo. Un mensaje del CM a un CC se representa de forma similar.

El acrónimo de los mensajes de petición de un CC se representan en el CC emisor. El acrónimo de las peticiones y respuestas del CM se representan en el CC receptor. El estado se representa, en los CC y en el CM, al emitir una petición, recibir una petición o recibir una respuesta. Los estados en el CM se representan de la misma forma que en un diagrama temporal.

11. Esta representación permite visualizar en menos espacio más mensajes.

12. En ocasiones puede girarse la representación 90 grados.

Cuando un CM al procesar una petición emita varios mensajes supondremos que se emiten simultáneamente y llegan simultáneamente a los destinos.

Animación

372 Capítulo 6

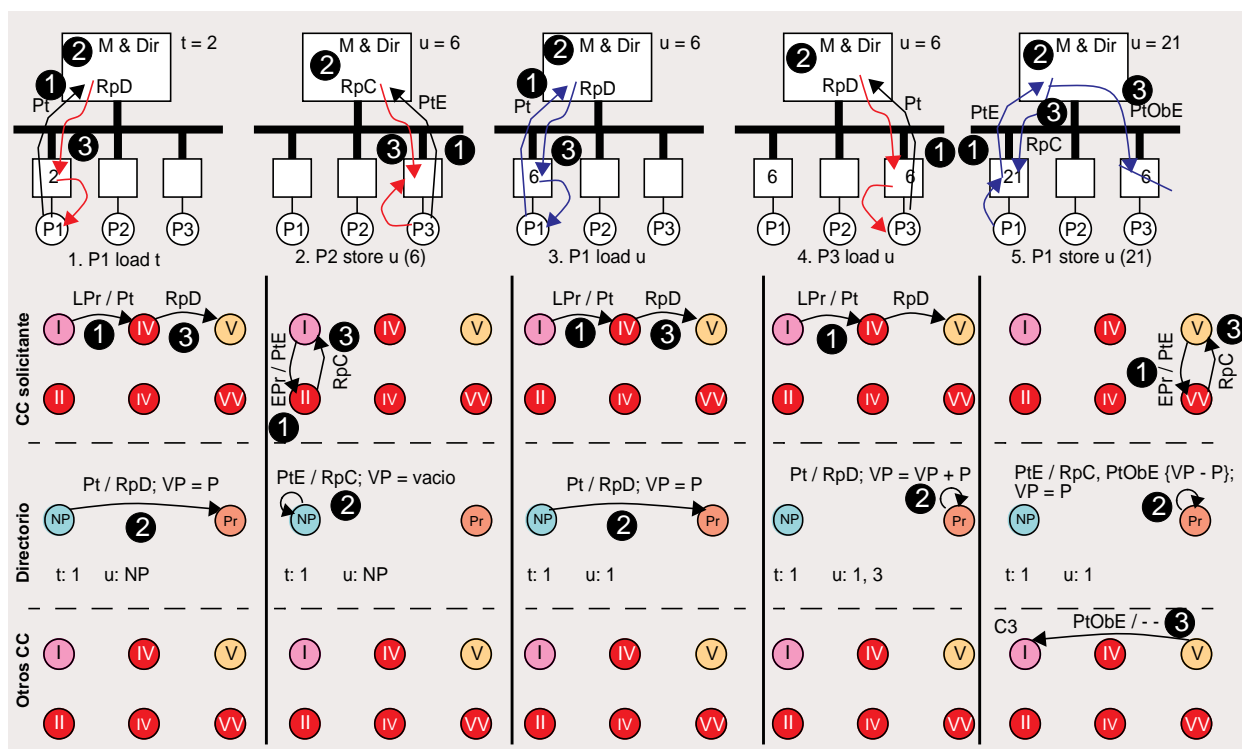


Figura 6.22 Protocolo de directorio VI. Secuencia de accesos a memoria y fotogramas de las peticiones y respuestas y cambios de estado.

Verificación no formal de coherencia y consistencia

En este capítulo el CM actúa de forma similar al bus en el Capítulo 4. En el multiprocesador existe un CM centralizado. Para acceder al CM se utiliza un árbitro. El CM no empieza a procesar una petición a un bloque hasta que ha finalizado el procesamiento de la petición previa, ya sea al mismo o distinto bloque.

La red de comunicación del CM a los CC mantiene el orden de emisión de los mensajes desde el CM. Por tanto, el CM establece un orden global entre las transacciones que gestiona.

Coherencia de cache

Orden de programa en accesos a la misma posición de memoria. El procesador efectúa los accesos en el orden determinado por el L.M. El procesador se bloquea en un fallo de lectura o en una escritura, hasta que finaliza la transacción correspondiente¹³.

Propagación de escrituras. Al utilizar escritura inmediata todas las escrituras se emiten al directorio. Un CC emite una petición de escritura al CM y éste, utilizando la información disponible en el directorio, la propaga a los CC que tienen copia del bloque. Las peticiones de invalidación (PtObE) se transmiten concurrentemente con la respuesta al CC solicitante (RpDC). La red del CM a los CC mantiene el orden de emisión de los mensajes. Entonces, no se espera respuesta de las peticiones de invalidación (PtObE). La respuesta está implícita en la emisión del mensaje.

Serialización de escrituras (atomicidad de una escritura).

- Punto de serialización: El CM es el punto de serialización de las escrituras a la misma posición de memoria¹⁴. El arbitraje para acceder al CM determina el orden de serialización de las transacciones de escritura a una posición de memoria. El CM procesa las peticiones en serie. Sólo hay una transacción en curso.
- Consolidación de una escritura: Una escritura está consolidada al finalizar la transacción. El CM finaliza la transacción al emitir el mensaje de respuesta al CC que ha efectuado la petición y las peticiones de invalidación a los CC que tiene copia. Por otro lado, una escritura es atómica, ya que, una vez finalizada la transacción correspondiente, una instrucción load posterior lee el valor establecido por la escritura previa, en el orden de procesado en el CM¹⁵.

Consistencia secuencial de memoria

Orden de programa. El procesador efectúa los accesos a cualquier posición de memoria en el orden determinado por el L.M.

- Lectura: Espera hasta que se obtiene el dato.
- Escritura: El CM es el punto de ordenación de todas las escrituras. Todas las escrituras requieren emitir una petición al CM. La recepción de la respuesta de confirmación del CM es una indicación de que la escritura

13. Hay que garantizar las dependencias de datos en el hilo que se ejecuta.

14. Aún más, es el punto de serialización de las escrituras y los fallos de lectura a cualquier posición de memoria, ya que se utiliza un único CM.

15. Las copias del bloque son invalidadas en cada transacción de escritura. Un fallo de lectura lee el valor establecido por la escritura previa en el orden de procesado en el CM. Un acierto de lectura lee el valor obtenido en el fallo de lectura previo por la escritura previa, que acierta en cache, del mismo procesador.

está consolidada. Durante el procesado de la transacción, el CM emite peticiones de invalidación a las caches que tienen copia del bloque. Debido a la propiedad de la RV, que conecta el CM a los CC, en la emisión de la petición se considera implícita la respuesta a la misma.

Atomicidad de las escrituras. Una escritura siempre requiere emitir una petición al CM.

- Consolidación de una escritura: La finalización de la transacción (recepción de la respuesta del CM) es una indicación de que la escritura está consolidada.
- Suministro del valor en una lectura: Un fallo de lectura requiere emitir una petición al CM. El valor devuelto en la transacción es el valor establecido en la última escritura consolidada, a la misma posición de memoria. Un acierto de lectura lee el valor de la copia en cache. Este valor ha sido establecido en el fallo de lectura previo o en la escritura previa, que acierta en cache, del mismo procesador.

PROTOCOLO DE DIRECTORIO CON INVALIDACION Y ESCRITURA RETARDADA

En este apartado se describe en primer lugar la secuencia de mensajes utilizada en una transacción de coherencia. Posteriormente se describe la organización del multiprocesador y los agentes de coherencia en las caches y memoria. Seguidamente se detallan los tipos de mensajes utilizados y finalmente se describen los estados y transiciones entre estados.

A este protocolo lo denominaremos protocolo de directorio MLI.

Descripción funcional de la secuencia de mensajes en un transacción

La descripción funcional de un protocolo MLI se ha efectuado en el Capítulo 4. En este apartado nos centraremos en la secuencia de mensajes de una transacción en un protocolo de directorio. Es interesante comparar la secuencia de mensajes con las acciones que se efectúan en una transacción de bus para observar la similitud (Capítulo 4).

En la Figura 6.23 se muestra la secuencia de mensajes punto a punto en un protocolo de directorio MLI. El caso a) representa una transacción de lectura y la memoria tiene el bloque actualizado. El caso b) representa una transacción

de escritura, la memoria tiene el bloque actualizado y hay copias del bloque en otras caches. El caso c) representa una transacción de lectura o escritura y una cache tiene el bloque en exclusividad.

Se distinguen dos grupos en función del número de pasos¹⁶ serie en una transacción. En un grupo (a y b) la secuencia es de dos pasos y en el otro grupo (c) es de cuatro pasos y requiere la colaboración de un CC distinto del que hace la petición.

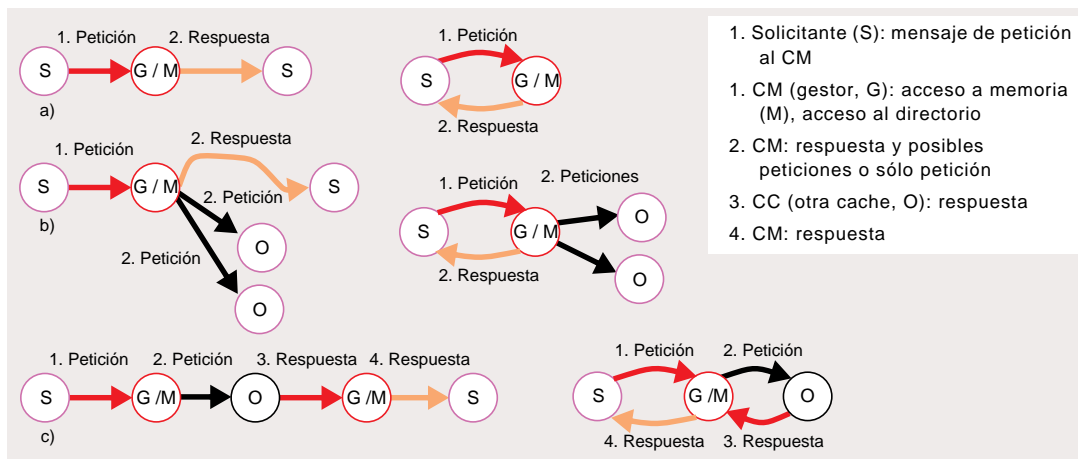


Figura 6.23 Flujo de mensajes en un protocolo de directorio MLI entre el CC y el CM y entre el CM y otros CC. Izquierda: representación lineal. Derecha: representación no lineal (los nodos no se replican).

Cuando la secuencia es de dos pasos se distinguen dos casos en función de si el CM, además de la respuesta, también emite peticiones a algún CC que no es el receptor de la respuesta. En cualquiera de los dos casos los dos pasos de una transacción son: 1) petición de un CC y 2) respuesta y en su caso, peticiones del CM.

En una transacción con una secuencia de cuatro pasos distinguimos: 1) petición de un CC, 2) petición del CM a un CC distinto del que ha efectuado la petición, 3) respuesta de este CC al CM y 4) respuesta del CM al CC que ha efectuado la petición.

Organización del multiprocesador

En la Figura 6.24 se muestra un esquema linealizado del multiprocesador y se muestra una transacción de 4 pasos. En el camino de las caches a memoria distinguimos una CP donde se almacenan las peticiones de los CC. Cada CC

¹⁶ Por pasos se entiende el número de mensajes en secuencia. En un paso puede haber varios mensajes concurrentes.

tiene un camino independiente, punto a punto, para transmitir los mensajes a la CP. A esta red la denominamos red de ida (RI). El CM procesa en serie los mensajes almacenados en la CP¹⁷.

Del CM a las caches hay, para cada una de ellas, un camino punto a punto que transporta peticiones de coherencia emitidas por el CM. El CM al procesar una petición, utilizando la información disponible en el directorio, sólo envía peticiones de coherencia a los CC que tiene copia del bloque y que además, es necesario que efectúen una acción para mantener la coherencia.

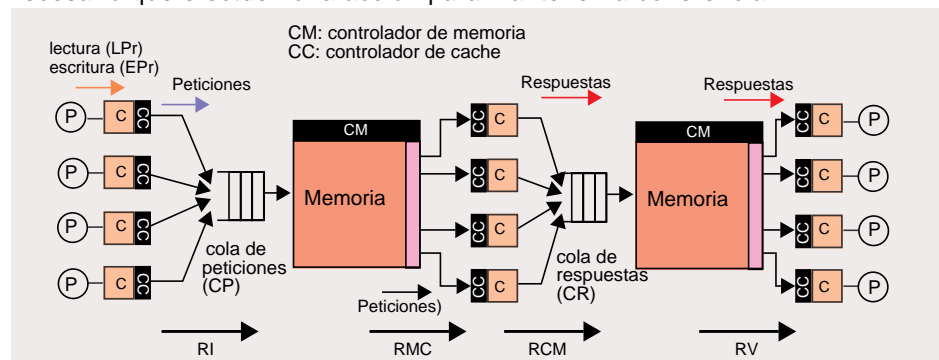


Figura 6.24 Detalle de las redes de interconexión.

En la Figura 6.24, se distingue una cola de respuestas (CR). Los CC almacenan en esta cola la respuesta a la petición de coherencia del CM. Recordemos que, en un protocolo MLI, en ocasiones es necesario que una cache suministre el bloque¹⁸.

Cada respuesta almacenada en la CR es procesada por el CM para finalizar la transacción pendiente en el directorio, la cual ha inducido la necesidad de esta respuesta.

El CM procesa la respuesta y finaliza la transacción, enviando un mensaje de respuesta al CC que inició la transacción. La transmisión de las respuestas se efectúa utilizando conexiones punto a punto entre el CM y los CC.

En resumen, en el multiprocesador distinguimos de forma lógica dos redes desde los CC al CM. Una de ellas es para transmitir peticiones (red de ida, RI) y la otra para transmitir respuestas a una petición del CM (red RCM).

17. Mientras se suponga que sólo existe un acceso a memoria en el multiprocesador la cola de peticiones no tiene utilidad.

18. Mientras se suponga que sólo existe un acceso a memoria en el multiprocesador la cola de respuestas no tiene utilidad.

En el camino del CM a los CC también se distinguen dos redes lógicas. Una de ellas se utiliza para transmitir peticiones del CM a los CC, como paso necesario en el procesamiento de una transacción (red RMC). La otra red se utiliza para transmitir el mensaje de respuesta, correspondiente a la finalización de una transacción (red de vuelta, RV).

Las redes lógicas RV y RMC se pueden implementar mediante una única red (Figura 6.25). La razón es que cuando el CM requiere emitir en paralelo una respuesta a un CC y varias peticiones a otros CC, para efectuar una acción de coherencia, el destinatario de la respuesta no es destinatario de una petición (transacción de dos pasos, Figura 6.23). Entonces, en este caso sólo indicaremos que se utiliza la red de vuelta. Sin embargo, notemos que las dos redes lógicas se utilizan en paralelo.

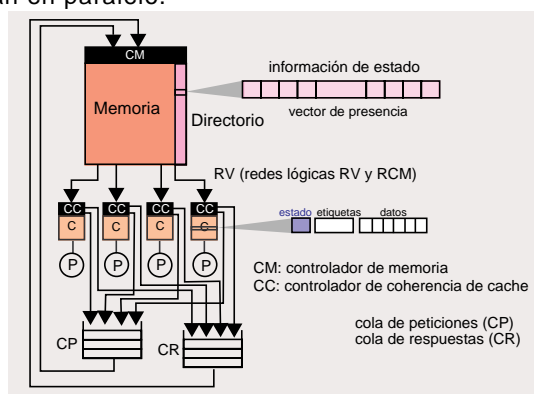


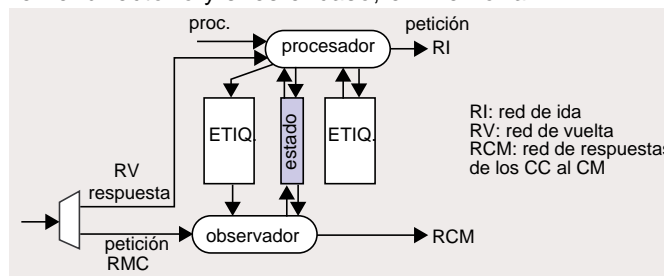
Figura 6.25 Redes y colas de peticiones y respuestas.

Otro caso donde el CM efectúa una petición a un CC es en una transacción de cuatro pasos (Figura 6.23). En este caso la petición a un CC es previa al envío de la respuesta del CM al CC que ha iniciado la transacción. Entonces, para identificar la petición del CM, de forma explícita, indicaremos la red RMC cuando el CM envíe la petición y la red RV cuando el CM envíe la respuesta.

Camino de datos en el controlador de coherencia

En las caches distinguimos dos agentes (Figura 6.26): a) agente procesador y b) agente observador. El primero sirve las peticiones de acceso a la cache del procesador y procesa la respuesta del CM a la petición efectuada previamente por el CC. El segundo se encarga de: a) atender las peticiones del CM al CC y b) emitir una respuesta a una petición del CM, si es necesario. El agente observador utiliza una copia del campo etiqueta para eliminar riesgos estructurales con accesos del procesador al campo etiquetas. Por la misma razón, el campo estado tiene dos caminos de acceso, uno para cada agente.

Incluido en el agente procesador está el autómata de reemplazo. Su función es seleccionar un bloque, almacenado en cache, para liberar un contenedor. En este contenedor se ubicará el bloque accedido en un fallo de cache. Cuando se conoce el bloque que se expulsa, el autómata procesador emite una petición de expulsión al CM. Esta petición es para mantener actualizada la información en el directorio y si es el caso, en memoria.



En un acceso a memoria que es fallo en cache y determina la expulsión de un bloque, en primer lugar se realizan las acciones encaminadas a expulsar el bloque. Posteriormente se inician las acciones para servir el fallo de cache.

El agente observador procesa peticiones del CM. Para ello, accede al duplicado de etiquetas, con el objetivo de conocer el contenedor de cache, en el cual se almacena el bloque referenciado en la petición. En un protocolo MLI es posible que memoria no almacene el bloque actualizado. En estas condiciones, el CM, para responder a una petición de un CC, necesita efectuar una petición a otro CC para que suministre el bloque. Por otro lado, el CM al procesar un mensaje de petición de bloque con intención de modificación, utiliza la información disponible en el directorio, para efectuar las peticiones de invalidación necesarias. La red que transporta los mensajes del CM a los CC mantiene el orden en el cual son emitidos los mensajes por el CM (redes lógicas RMC y RV). Por tanto, no es necesario que las caches respondan a peticiones de invalidación. Sólo es necesario que respondan a las peticiones que requieren suministrar el bloque.

Camino de datos en el controlador de memoria

Posteriormente al detallar el protocolo MLI se observa que el CM, al procesar una petición de un CC, puede emitir varias peticiones y una respuesta para finalizar una transacción. El conjunto de CC que reciben una petición no incluye al CC que recibe la respuesta. También, cuando un bloque no está actualizado en memoria, el CM emite un mensaje de petición a un CC, el cual responderá con el bloque de datos. Finalmente, el CM emitirá una respuesta al CC que inició la transacción¹⁹.

En la Figura 6.27 se muestra un esquema del camino de datos del CM en el que se distinguen dos agentes. El agente que procesa las peticiones (P) y el agente que procesa las respuestas (R).

El agente P accede al directorio y a memoria y emite peticiones, si es necesario, a los CC utilizando conexiones punto a punto. Cuando debe esperar una respuesta, a una petición que ha emitido, actualiza el estado del bloque en el directorio y almacena la petición que está procesando, que denominamos petición pendiente, en el buffer de transacciones pendientes (BTP). En este buffer se almacena la dirección del bloque y un identificador de la petición al CC. Este identificador lo utiliza un CC en la respuesta que efectúa al CM.

El agente R empareja una respuesta con la petición pendiente almacenada en el BTP y después actualiza el directorio y la memoria. La recepción y procesamiento de la respuesta se indica en la entrada asociada en BTP. El agente P está monitorizando BTP por la llegada de respuestas. Cuando el agente R indica que se ha procesado una respuesta, el agente P emite la respuesta al CC que ha iniciado la transacción.

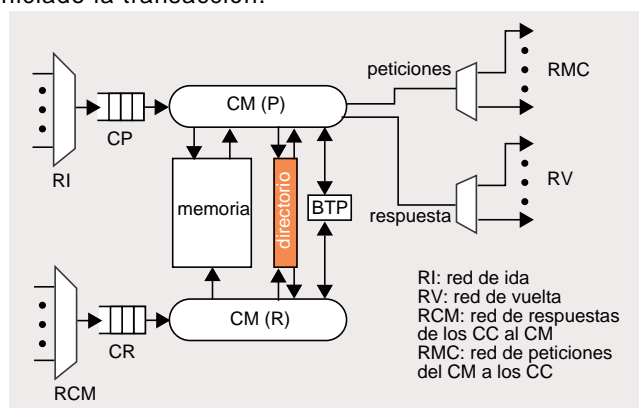


Figura 6.27 Esquema del camino de datos en el CM.

19. En este capítulo la red RMC y la red RV son una única red.

Mensajes del protocolo

En la Figura 6.28 se muestra un esquema genérico de la transmisión de mensajes con peticiones y respuestas entre los CC y el CM y viceversa. Notemos que el esquema se ha linealizado para que todos los mensajes fluyan de izquierda a derecha. Como los CC emiten peticiones y reciben peticiones y respuestas y el CM emite peticiones y recibe respuestas, estos elementos se dibujan más de una vez en el esquema.

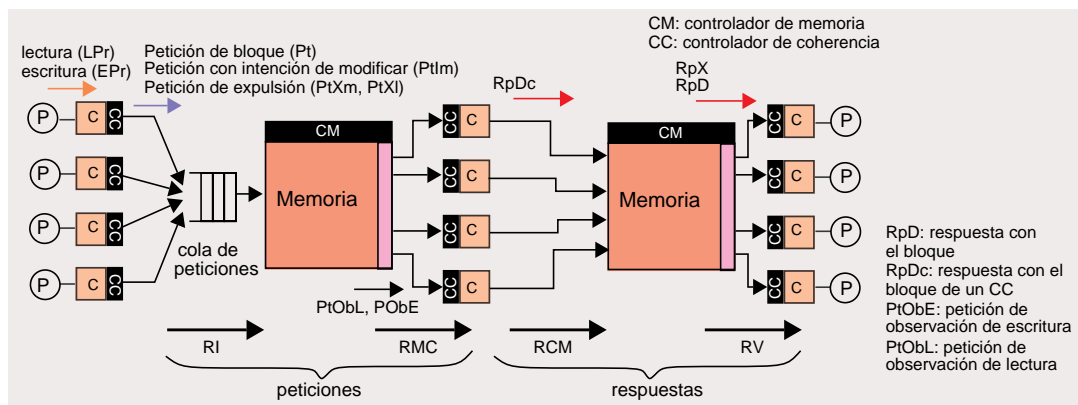


Figura 6.28 Esquema de la transmisión de mensajes en un protocolo de directorio MSI.

En la descripción del protocolo se distinguen los siguientes tipos de mensajes:

- Peticiones de un CC al CM (Pt, PtIm, PtXm, PtXI)
- Peticiones del CM a los CC (PtObL, PtObE)
- Respuestas del CM a un CC (RpD, RpX)
- Respuestas de un CC al CM (RpDc)

En las siguientes tablas al describir los mensajes de petición y respuesta se indica la información básica que se transmite en el mensaje. El acrónimo “op” indica operación, el acrónimo “Id” indica identificador de la petición. En la respuesta se utiliza el mismo identificador.

En la Tabla 6.7 se describen las peticiones de un procesador al CC. El procesador efectúa dos operaciones: a) lectura (load) y b) escritura (store). En caso de acierto en cache sólo se accede a la cache. En un fallo de lectura o de escritura se accede a memoria. En una escritura, si no se dispone de exclusividad en el acceso al bloque, también se accede a memoria.

Procesador	Controlador de coherencia (CC)	Comentario
Peticiones al CC	Respuestas del CC	
LPr (load): lectura de un dato <div>op dirección</div>	dato <div>dato</div>	Si es un acierto en cache se lee el dato de cache. En caso contrario se inicia una transacción de lectura de bloque (Pt) y se asigna un contenedor para almacenar el bloque. Posteriormente se suministra el bloque.
EPr (store): escritura de un dato <div>op dirección dato</div>	confirmación de la escritura <div>confirmación</div>	Si es un fallo o no se dispone de exclusividad en el acceso al bloque se solicita el bloque con intención de modificarlo (PtIm). Cuando se dispone de los derechos de acceso se actualiza el bloque en cache.

Tabla 6.7 Peticiones del procesador y respuestas.

En la Tabla 6.8 se describen las peticiones de un CC al CM y las acciones del CC para gestionar la ubicación de bloques en la cache (conflictos, CcRe). Se distinguen los mensajes de petición de bloque (Pt) y petición de bloque con intención de modificación (PtIm). También se describe la acción de expulsión de un bloque debido a un conflicto.

Controlador de coherencia (CC)	Controlador de memoria (CM)	Comentario
Mensajes de petición al CM y acciones	Mensajes de respuestas	
Pt: petición de bloque <div>op dirección Id</div>	RpD: respuesta con el bloque <div>Id bloque</div>	Se lee el bloque de la memoria, se actualiza el directorio y se suministra el bloque.
PtIm: petición de bloque con intención de modificación <div>op dirección Id</div>	RpD: respuesta con el bloque <div>Id bloque</div>	El CM suministra el bloque después de leerlo de memoria u obtenerlo de otra cache. Se actualiza el directorio en consecuencia.
CcRe: expulsión de un bloque.		Se invalida la información del contenedor.
PtXm: petición de expulsión de un bloque en estado M <div>op dirección bloque Id</div>	RpX: respuesta de confirmación <div>Id confirmación</div>	Se notifica al directorio de la expulsión de un bloque, para que actualice la información de estado del bloque en el directorio y actualice memoria, si es el caso.
PtXI: petición (notificación) de expulsión de un bloque en estado L <div>op dirección Id</div>		

Tabla 6.8 Peticiones del CC al CM y respuestas del CM.

En la Tabla 6.9 se describen las peticiones del CM a los CC y la acción del CM para tener actualizada la información de presencia de los bloques. El CM envía mensajes de petición de observación de lectura o de escritura para mantener la coherencia. Un CC responde incluyendo el bloque en un mensaje de respuesta cuando tiene el bloque en exclusividad.

Controlador de memoria	Controlador de coherencia	Comentario
Envío de petición a CC y acciones Envío de petición de observación de escritura <div> <input type="text" value="op"/> <input type="text" value="dirección"/> <input type="text" value="Id"/> </div>	Respuestas Respuesta de observación de escritura RpDc: respuesta con el bloque, si es el caso <div> <input type="text" value="Id"/> <input type="text" value="bloque"/> </div>	Si el CC que recibe la petición tiene el bloque en exclusividad emite una respuesta con el bloque. En cualquier caso se invalida el bloque.
Envío de petición de observación de lectura <div> <input type="text" value="op"/> <input type="text" value="dirección"/> <input type="text" value="Id"/> </div>	Respuesta de observación de lectura RpDc: respuesta con el bloque <div> <input type="text" value="Id"/> <input type="text" value="bloque"/> </div>	El CC emite una respuesta con el bloque y cambia el estado del bloque para indicar que no hay exclusividad.
Envío de petición de actualización del directorio.		Mantiene actualizada la información de compartición de los bloques.
Envío de petición de actualización de memoria		Actualiza la memoria con el bloque de datos.

Estados y transiciones

En un protocolo de coherencia de cache, donde se utiliza escritura retardada, es necesario en ocasiones obtener el bloque de datos de una cache para suministrar el bloque a otra cache. En estas condiciones, el CM deberá solicitar el bloque a un CC y esperar a que responda con el bloque. Para identificar la espera de una respuesta de un CC se utilizan estados transitorios en el CM²⁰.

Estos tres estados están codificados en el VP y el bit de exclusividad (BE) asociado a cada bloque. Si todos los bits del VP están desactivados el estado es NP. Para distinguir entre el estado L y el estado M se utiliza el BE. Si este bit está desactivado, los bits del VP que están activados identifican los procesadores que tienen copia del bloque. En caso contrario, sólo un bit del VP puede estar activado e identifica el procesador que tiene copia del bloque en exclusividad.

Protocolo de directorio con una red ordenada

El etiquetado de los estados transitorios sigue las reglas descritas al describir los protocolos de observación en Capítulo 4.

El número de estados transitorios en el directorio es dos (ML, MM). Se utilizan para identificar la espera de la respuesta de un CC que suministra un bloque solicitado por el CM.

Cache. En una cache un bloque puede tener tres estados estables (Figura 6.29): a) inválido (I), b) copia del bloque para operaciones de lectura (L) y c) copia del bloque en exclusividad, lo cual permite modificarlo (M). Para identificar la espera de la respuesta del CM a una petición del CC se utilizan cinco estados transitorios (IM, IL, LM, MI, LI).

Para describir el protocolo se utilizan las dos peticiones del procesador que requieren acceder al directorio (fallo en una instrucción load, store y una instrucción store que accede a un bloque sin permiso de exclusividad) y las dos posibilidades de ubicación del bloque solicitado, en memoria o en una cache. Posteriormente se detallan las transiciones entre estados en una expulsión de un bloque de cache.

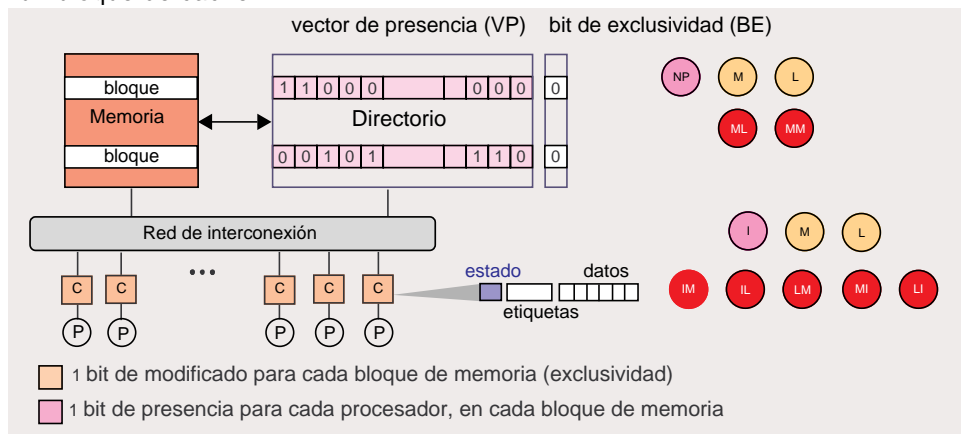


Figura 6.29 Protocolo de directorio MLI. Estados de un bloque en el directorio y en un contenedor de cache.

En la descripción se mostrarán las transiciones entre estados de un bloque en el CC que efectúa la petición (agente procesador), en el CM y en otros CC (agente observador).

Fallo en lectura

En la Figura 6.30 se muestra el flujo de mensajes en un fallo de lectura. El CC emite un mensaje con una petición de lectura de bloque (Pt). El CM accede al directorio para leer el VP y el BE (estado del bloque) y especulativamente accede a memoria para leer el bloque²¹. El CM después de analizar el estado

del bloque, el cual es NP o L, emite un mensaje de respuesta (RpD), que incluye el bloque, al CC que ha efectuado la petición. Además, el CM actualiza el VP añadiendo al CC que ha efectuado la petición.

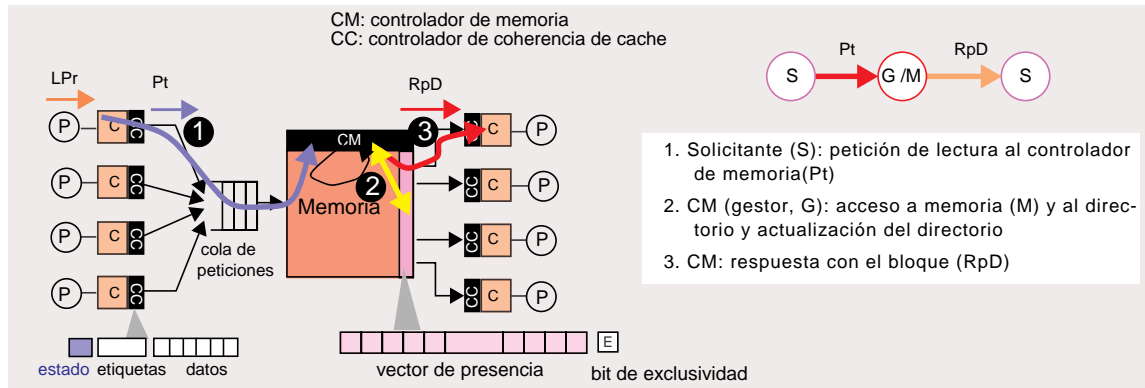


Figura 6.30 Protocolo de directorio MLI. Flujo de mensajes en un fallo de lectura.

En la Figura 6.31 se muestran las transiciones entre estados en un fallo de lectura en: a) la cache del CC que efectúa la petición (solicitante), b) el directorio (CM) y c) otros CC. El CC envía el mensaje de petición de lectura y establece el estado transitorio del bloque como IL. El bloque permanece en este estado hasta que el CC recibe la respuesta del CM, la cual es procesada por el agente procesador, que establece como estado estable del bloque el estado L.

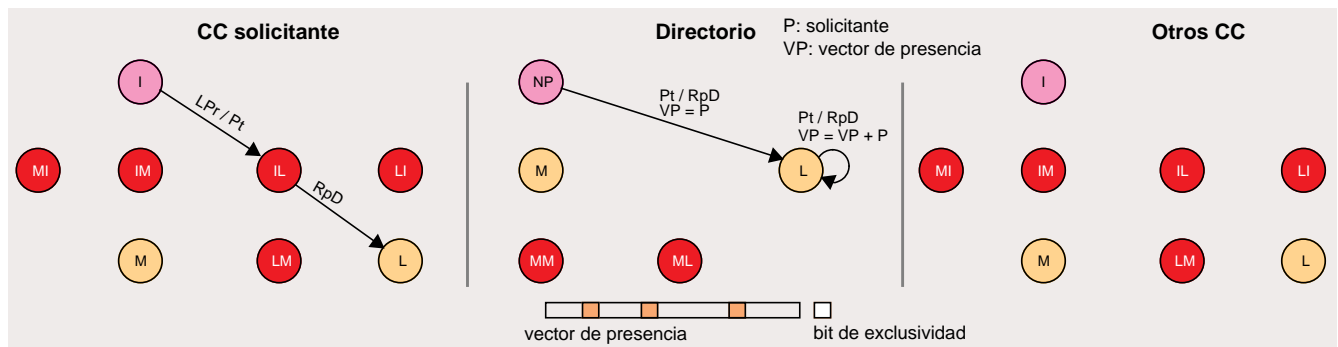


Figura 6.31 Protocolo de directorio MLI. Transiciones entre estados en un fallo de lectura.

21. Decimos que es especulativo debido a que el acceso se efectúa en paralelo con el acceso al directorio para leer el VP del bloque. Por tanto, no se conoce si memoria almacena un bloque válido.

El CM al recibir el mensaje de petición de lectura accede al directorio y especulativamente a memoria. A partir del estado del bloque en el directorio, el CM determina que puede responder a la petición utilizando el bloque leído de memoria. El CM, además de enviar la respuesta, actualiza el VP añadiendo al CC, que ha efectuado la petición, en el VP. Los otros CC no reciben peticiones de coherencia desde el CM.

Fallo en lectura y bloque en estado M en otra cache

En la Figura 6.32 se muestra el flujo de mensajes entre el CC que efectúa la petición y el CM y entre este último y el CC que tiene el bloque en exclusividad.

Respecto de la descripción de un “fallo de lectura”, la diferencia es que el CM debe obtener el bloque de una cache que lo tiene en exclusividad. Este hecho lo determina el CM después de analizar el estado del bloque en el directorio. Para obtener el bloque, el CM efectúa una petición de observación de lectura de bloque (PtObL) al CC correspondiente y espera que le responda con un mensaje que incluya el bloque (RpDc). Al recibir el CM esta respuesta, responde al CC que ha efectuado la petición (RpD) y actualiza memoria y el estado del bloque en el directorio.

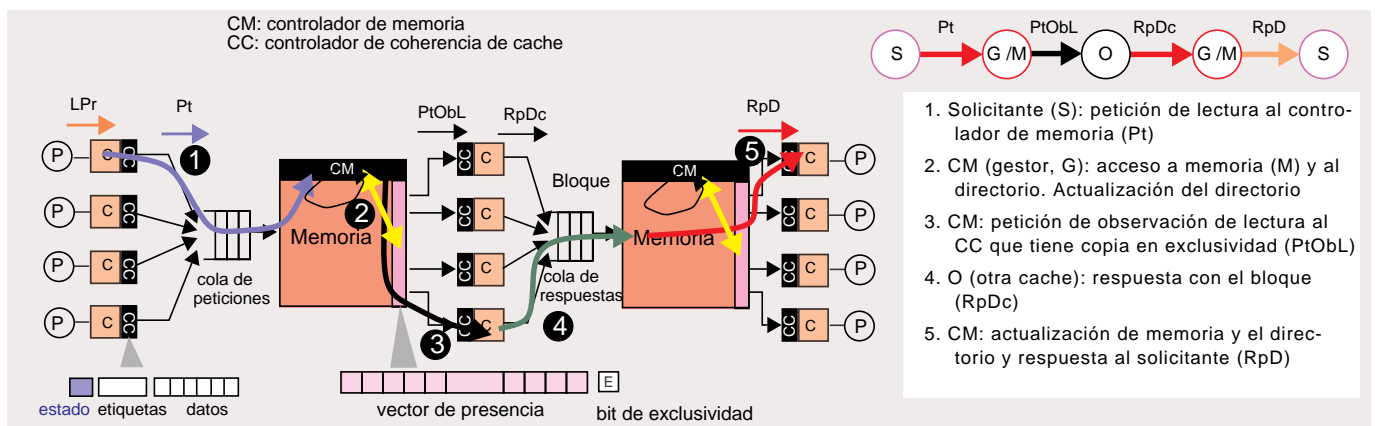


Figura 6.32 Protocolo de directorio MLI. Flujo de mensajes en un fallo de lectura y una cache tiene el bloque en exclusividad.

En la Figura 6.33 se muestran las transiciones entre estados en la cache del CC solicitante, el directorio y otros CC. La diferencia con la Figura 6.30 es que el CM parte de un bloque en estado M. Entonces, el CM establece el estado transitorio ML, al emitir la petición al CC que tiene el bloque en exclusividad. Este CC, al recibir esta petición (PtObL), responde con un mensaje que incluye el bloque y establece como nuevo estado del bloque el estado L. El CM al

recibir la respuesta (RpDc) y procesarla envía un mensaje de respuesta al CC del solicitante (RpD). Además, actualiza memoria con el bloque y modifica el VP y el BE, estableciendo en el directorio como estado del bloque el estado L.

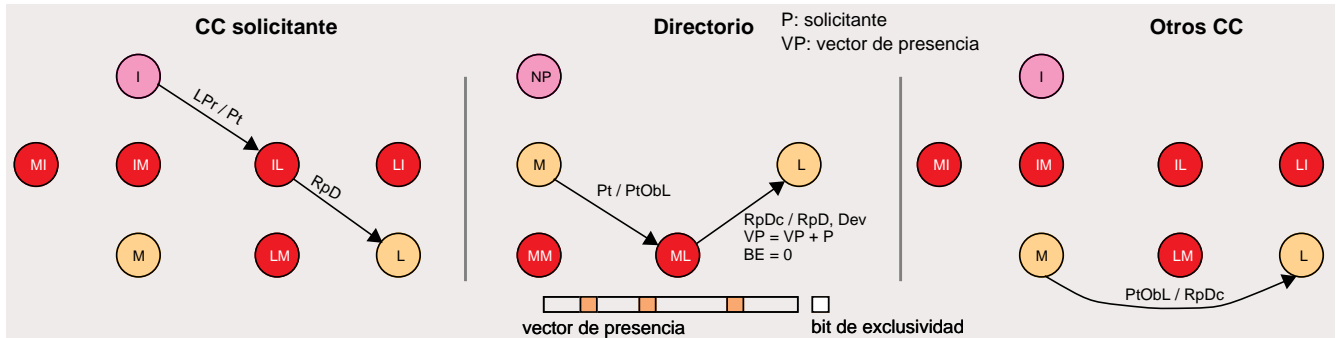


Figura 6.33 Protocolo de directorio MLI. Transiciones entre estados en un fallo de lectura y una cache tiene el bloque en exclusividad.

Fallo de escritura o petición de exclusividad

En la Figura 6.34 se muestra el flujo de mensajes en un fallo de escritura entre un CC y el CM y entre éste y posiblemente varios CC. El CC del solicitante emite un mensaje con una petición de bloque con intención de modificación (PtIm). El CM, al recibir el mensaje, lee el estado del bloque en el directorio y accede especulativamente a memoria. A partir de la información de estado del bloque en el directorio, el CM determina las peticiones de observación de escritura que debe emitir (PtObE). A la vez que envía los mensajes PtObE también envía la respuesta al solicitante (RpD).

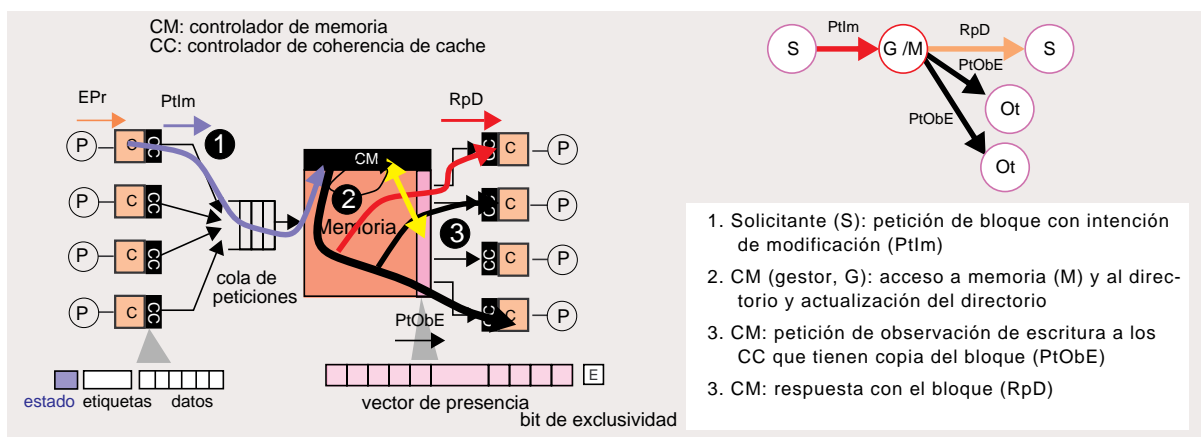


Figura 6.34 Protocolo de directorio MLI. Flujo de mensajes en un fallo de escritura.

Como se utiliza una red ordenada (RV), para transmitir los mensajes entre el CM y los CC, no es necesario esperar una respuesta de los CC que reciben una petición PtObE. Por tanto, estos CC no emiten una respuesta al recibir dicha petición.

En la Figura 6.35 se muestran las transiciones entre estados en la cache del CC que efectúa la petición, el directorio y otros CC. El CC del solicitante establece un estado transitorio (IM o LM), en función del estado estable inicial, esperando la respuesta. Al recibir la respuesta el estado estable del bloque será M.

En el directorio el estado del bloque pasa de L a M o de NP a M, el VP se actualiza en consecuencia y el BE se activa. Además, el CM emite mensajes de respuesta (RpD) y de petición de observación de escritura (RpObE).

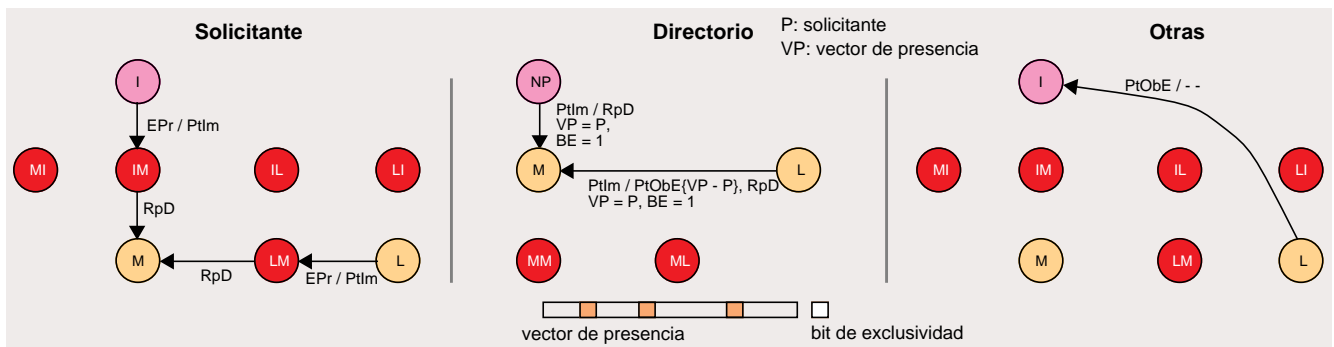


Figura 6.35 Protocolo de directorio MLI. Transiciones entre estados en un fallo de escritura.

Cuando un CC emite una petición Ptlm estando el bloque en estado L, el bit de presencia del CC está activado en el VP. Por tanto, hay que excluirlo de la lista de CC que reciben una petición PtObE (VP - P). El CC, al recibir la respuesta RpD, siempre almacena el bloque recibido en el contenedor correspondiente, tanto si el estado estable inicial es L como I.

Los CC que tienen copia del bloque, al recibir la petición PtObE invalidan la copia del bloque.

Fallo en escritura o petición de exclusividad y bloque en estado M en otra cache

En la Figura 6.36 se muestra el flujo de mensajes en un fallo de escritura o petición de exclusividad y una cache tiene el bloque solicitado en exclusividad. El CC solicitante envía un mensaje de petición de bloque con intención de modificación (Ptlm). El CM al leer el estado del directorio determina que un CC tiene el bloque en exclusividad. Por tanto, envía una petición de observación

de escritura (PtObE) a este CC. Como respuesta a esta petición, el CC emite un mensaje de respuesta al CM que incluye el bloque (RpDc). Entonces, el CM responde al CC solicitante suministrando el bloque (RpD). Además, actualiza el estado del bloque en el directorio.

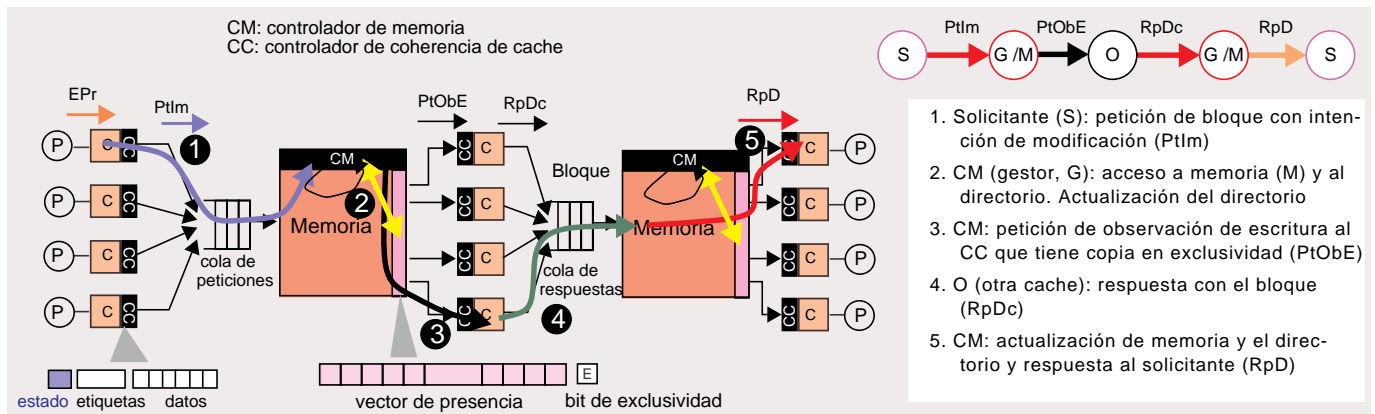


Figura 6.36 Protocolo de directorio MLI. Flujo de mensajes en un fallo de escritura y otra cache tiene el bloque en exclusividad.

En la Figura 6.37 se muestran las transiciones entre estados en el CC del solicitante, el directorio y otros CC. La transición en el solicitante entre los estados I y M es la misma que en la Figura 6.35. El CM, al emitir la petición PtObE, establece como estado transitorio del bloque MM. Cuando se recibe la respuesta del CC, que tiene el bloque en exclusividad, el CM cambia el estado del bloque al estado M. Entonces, el CM envía el mensaje de respuesta al CC solicitante. El CM actualiza el estado identificando al nuevo procesador que tiene el bloque en exclusividad. Además, el CC cuya cache tenía el bloque en exclusividad, al emitir la respuesta cambia el estado del bloque del estado M al estado I.

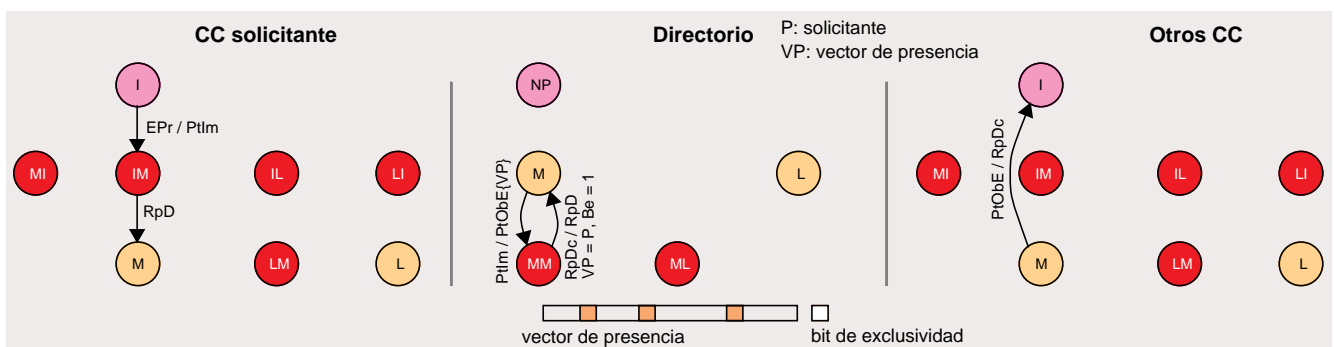


Figura 6.37 Protocolo de directorio MLI. Transiciones entre estados en un fallo de escritura y una cache tiene el bloque en exclusividad.

Expulsión de un bloque de cache

En la Figura 6.38 se muestra el flujo de mensajes cuando se expulsa un bloque de cache. El CC efectúa una petición de expulsión (PtXm, PtXI) que va acompañada del bloque cuando el estado del bloque es M (PtXm). El CM en cualquier caso actualizada la entrada correspondiente en el directorio. Además, se actualiza memoria si el bloque expulsado estaba en el estado M en la cache. El CM responde al CC confirmando que se ha realizado la acción solicitada.

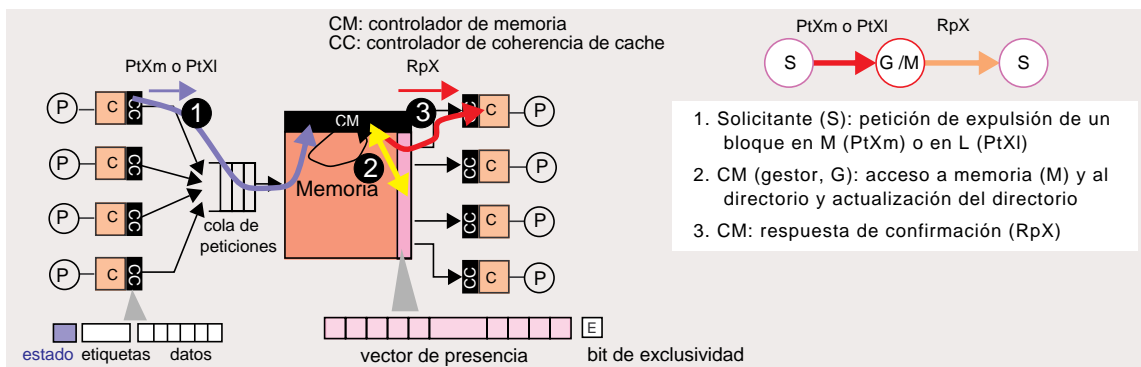


Figura 6.38 Expulsión de un bloque de cache.

En la Figura 6.39 se muestran las transiciones entre estados en: a) la cache del CC que expulsa el bloque, b) el directorio y c) otros CC. En otros CC no se producen transiciones entre estados. Para distinguir entre la expulsión de un bloque en el estado M o en el estado L, se utilizan distintos tipos de peticiones y también estados transitorios distintos, MI y LI respectivamente. Cuando el CM procesa la petición actualiza la memoria si la petición es PtXm y en cualquiera de los dos casos, se actualiza la entrada correspondiente del bloque en el directorio. Finalmente el CM envía una respuesta de confirmación. El CC al recibir la respuesta cambia el estado del bloque al estado I.

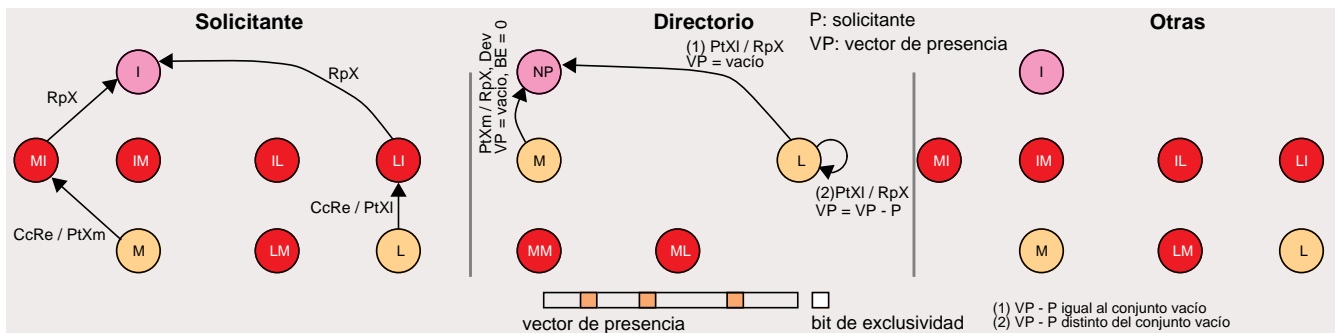


Figura 6.39 Protocolo de directorio MLI. Transiciones entre estados cuando se expulsa un bloque.

En la Figura 6.40 y la Figura 6.41 se muestran, respectivamente, los diagramas de transiciones entre estados de un bloque en cache y en el directorio. En el primer diagrama están incluidas las peticiones del procesador que no requieren iniciar transacciones explícitas de coherencia. En el CC se distinguen, en diagramas separados, las transiciones iniciadas por el agente procesador y el agente observador (Figura 6.40).

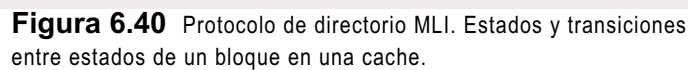
[illegible]

Figura 6.41 Protocolo de directorio MLI. Estados y transiciones entre estados de un bloque en el directorio.

Tablas de estados y transiciones

En la Tabla 6.10 se muestran en formato tabla los estados y las transiciones entre estados de un bloque en una cache. Las casillas que no contienen información indican un error. En un estado determinado no puede llegar el evento que determina la casilla correspondiente en el cruce.

			Eventos del procesador y reemplazo			Eventos externos (respuestas y petición)			
			LPr	EPr	CcRe	RpD	RpX	PtObL	PtObE
Estados	Estables	I	Pt; IL	PtIm; IM					
		L	-- ; L	PtIm; LM	PtXI; LI				-- ; I
		M	-- ; M	-- ; M	PtXm; MI			RpDc: L	RpDc; I
	transitorios	IL				-- ; L			
		IM				-- ; M			
		LM				-- ; M			
		LI					-- ; I		
		MI					-- ; I		

Tabla 6.10 Protocolo de directorio MLI. Tabla de estados y transiciones de un bloque en cache.

En la Tabla 6.11 se muestran en formato tabla los estados y transiciones entre estados de un bloque en el CM. En el evento PtXI se distingue el caso de que el CC sea el único que está en el vector de presencia o haya más CC.

			Eventos del controlador de coherencia					
			Pt	PtIm	PtXI		PtXm	RpDc
					VP = P	$P \in VP$ $P \neq VP$		
Estados	Estables	NP	RpD; L, VP = P	RpD; M VP = P, BE = 1				
		L	RpD; L, VP = VP + P	PtObE {VP - P}, RpD; M, VP = P, BE = 1	RpX; NP, VP = vacío	RpX; L VP = VP - P		
		M	PtObL; ML	PtObE {VP}; MM			RpX, Dev; NP, VP = vacío, BE = 0	
	transitorios	ML						RpD, Dev; L VP = VP + P, BE = 0
		MM						RpD; M VP = P, BE = 1

Tabla 6.11 Protocolo de directorio MLI. Tabla de estados y transiciones de un bloque en el directorio.

Representación de transacciones y transiciones entre estados

En este apartado se muestran tres formas de representar las transacciones y transiciones al ejecutar una secuencia de accesos a memoria: a) en formato tabla, b) mediante un diagrama temporal y c) mediante un diagrama temporal simplificado. Finalmente se muestra, mediante varios gráficos, una animación.

Representación en formato tabla

Utilizaremos una tabla, donde cada fila representa un acceso a memoria y no se gestiona el siguiente acceso hasta que ha finalizado el anterior. El número de columnas se incrementa, respecto del protocolo de directorio VI, con el objetivo de representar las transacciones que requieren cuatro pasos. Entre las columnas de memoria y suministro se intercalan dos nuevas columnas. Estas columnas se utilizan para representar las peticiones del CM a un CC (red RMC) y la respuesta del CC (red RCM). En la columna correspondiente a RMC, además de la petición se indica la secuencia de ordinales de los procesadores cuyos CC reciben la petición.

Después de la columna correspondiente a VP se añade una columna que se corresponde con el BE de la entrada del directorio (E).

En el caso de una transacción que requiere dos pasos y el CM efectúa peticiones, además de emitir una respuesta, las peticiones se representan en la columna correspondiente a RV. Recordemos que las redes lógicas RV y RMC se pueden implementar utilizando sólo una red física. Para representar las respuestas y las peticiones se utilizan dos filas.

Ejemplo. En la Tabla 6.12 se muestra una secuencia de accesos a memoria realiza por tres procesadores. Suponemos que las variables t y u no están almacenadas en cache y los bloques que las contienen se almacenan en contenedores distintos de cache.

El primer acceso a memoria es una instrucción load y se produce un fallo en la cache C1. Al emitir el mensaje de petición de bloque, el estado transitorio del bloque es IL. La variable accedida es t y el VP en el directorio sólo indica la presencia del bloque en la cache C1. El BE está desactivado. La RV transporta un mensaje de respuesta de datos y el estado estable del bloque en la cache C1, al finalizar la transacción, es L.

acceso	C 1	C 2	C 3	Red	mem.			Red	Red	sum.	Red	C 1		C 2		C 3	
	est.	est.	est.	RI	var.	VP	E	RMC	RCM		RV	var.	est.	var.	est.	var.	est.
1. P1 load t	IL			Pt	t	1, 0, 0	0			mem.	RpD	t	L				
2. P2 store u		IM		PtIm	u	0, 1, 0	1			mem	RpD			u	M		
3. P1 load u	IL			Pt	u	1, 1, 0	0	PtObL, 2	RpDc	C2	RpD	u	L	u	L		
4. P3 load u			IL	Pt	u	1, 1, 1	0			mem.	RpD					u	L
5. P1 store u	LM			PtIm	u	1, 0, 0	1			mem	RpD	u	M				
											PtObE,2,3			u	I	u	I

Tabla 6.12 Protocolo de directorio MLI. Secuencia de accesos a memoria.

El segundo acceso a memoria es una instrucción store. El bloque no está almacenado en cache. Por tanto, el estado transitorio es IM y el mensaje que se transmite por la RI es PtIm. No hay copia del bloque referenciado en ninguna cache. El CM actualiza el VP y activa el BE. Además, suministra el bloque utilizando la RV. El estado del bloque en cache al recibir la respuesta es M.

El tercer acceso a memoria lo efectúa el CC de la cache C1 y es a la variable u. El CM al procesar el mensaje Pt determina, utilizando la información almacenada en el directorio, que otra cache almacena el bloque en exclusividad. Entonces solicita el bloque mediante una petición PtObL al CC de la cache C2 y espera la respuesta. Cuando llega la respuesta, el CM actualiza la memoria y el directorio y transmite la respuesta por la RV, al CC que efectuó la petición. El estado final del bloque, tanto en la cache que ha suministrado el bloque como en la cache donde se ha almacenado, es L.

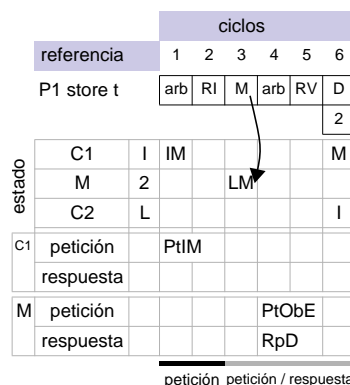
El cuarto acceso a memoria, efectuado por el CC de la cache C3, es mimético al primer acceso a memoria, con la salvedad de que el VP indicará copias del bloque en las caches C1, C2 y C3.

El quinto acceso a memoria es una instrucción store en el procesador P1. Al acceder a cache se determina que es acierto, pero no se dispone de exclusividad en el acceso al bloque. Por tanto, el estado transitorio es LM. El mensaje que transmite el CC al CM es PtIm. En el directorio el VP se actualiza para indicar que hay copia del bloque sólo en la cache C1 y además en exclusividad (BE = 1). Las otras copias del bloque, que había en otras caches, se invalidan mediante la emisión, por parte del CM, de un mensaje PtObE a los respectivos CC. En la tabla se representa este hecho utilizando una segunda fila en la columna RV, donde se indica la petición del CM y el ordinal de los CC. El estado estable del bloque en la cache C1 es M y en las otras caches es I.

Para representar las transacciones de dos pasos utilizaremos la misma representación que la descrita en el protocolo de directorio VI. Esto es, en el diagrama sólo representaremos la RV. La red RMC no se representa.

		ciclos					
referencia		1	2	3	4	5	6
P1 store t		arb	RI	M	arb	RV	D
						RMC	2
estado	C1	I	IM				M
	M	2		LM			
	C2	L					I
C1	petición		PtIM				
	respuesta						
M	petición				PtObE		
	respuesta				RpD		

petición petición / respuesta



Sin embargo, para identificar mejor peticiones y respuestas, en un diagrama temporal donde se muestra una transacción de cuatro pasos, utilizaremos el acrónimo RMC para indicar una petición del CM (Figura 6.43).

Protocolo de directorio con una red ordenada

Cuando una cache tiene el bloque en exclusividad, el estado de un bloque en el directorio se representa mediante el ordinal del procesador y la letra E.

En el ejemplo de la parte izquierda de la Figura 6.43 se muestra un fallo de lectura y el bloque está en estado M en otra cache. En la fase arb se indica el estado transitorio en la cache del CC que envía el mensaje de petición. Hay una copia del bloque en el estado M en la cache C2. En la fase M el CM determina que hay que solicitar el bloque al CC que gestiona la cache C2 y establece como estado transitorio del bloque ML.

En la fase RMC se envía una petición de observación de lectura al CC de la cache C2 (PtObL). En la fase Cx, donde x es el ordinal del procesador, el CC de la cache C2 accede al campo de datos de la cache para leer el bloque, actualiza el estado del bloque y en la siguiente fase transmite la respuesta al CM.

El CM, utilizando la respuesta del CC2, actualiza el directorio y la memoria. Posteriormente envía la respuesta, que incluye el bloque, al CC que ha efectuado la petición.

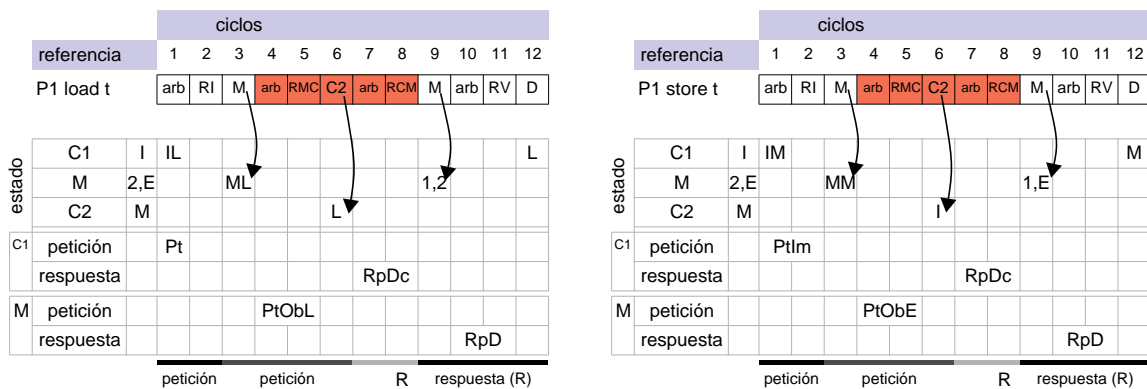


Figura 6.43 Protocolo de directorio MLI: diagrama temporal de acciones de coherencia de 4 pasos.

En el ejemplo de la parte derecha de la Figura 6.43 se muestra un fallo de escritura y el bloque está en estado M en otra cache. La secuencia de mensajes es similar a la secuencia que se muestra en la parte izquierda. Se diferencian en el tipo de mensaje y en consecuencia en los estados de las copias del bloque en las caches y el estado en el directorio.

Ejemplo. En la Figura 6.44 se muestra el diagrama temporal de la secuencia de accesos mostrada en la Tabla 6.12. En la parte central de la Figura 6.44 se especifican las caches de los procesadores, el directorio (M) y

las variables. En estas filas se representa el estado del bloque en cada cache y en el directorio, de la forma descrita en los ejemplos de la Figura 6.43. En la parte inferior se representan los mensajes transmitidos.

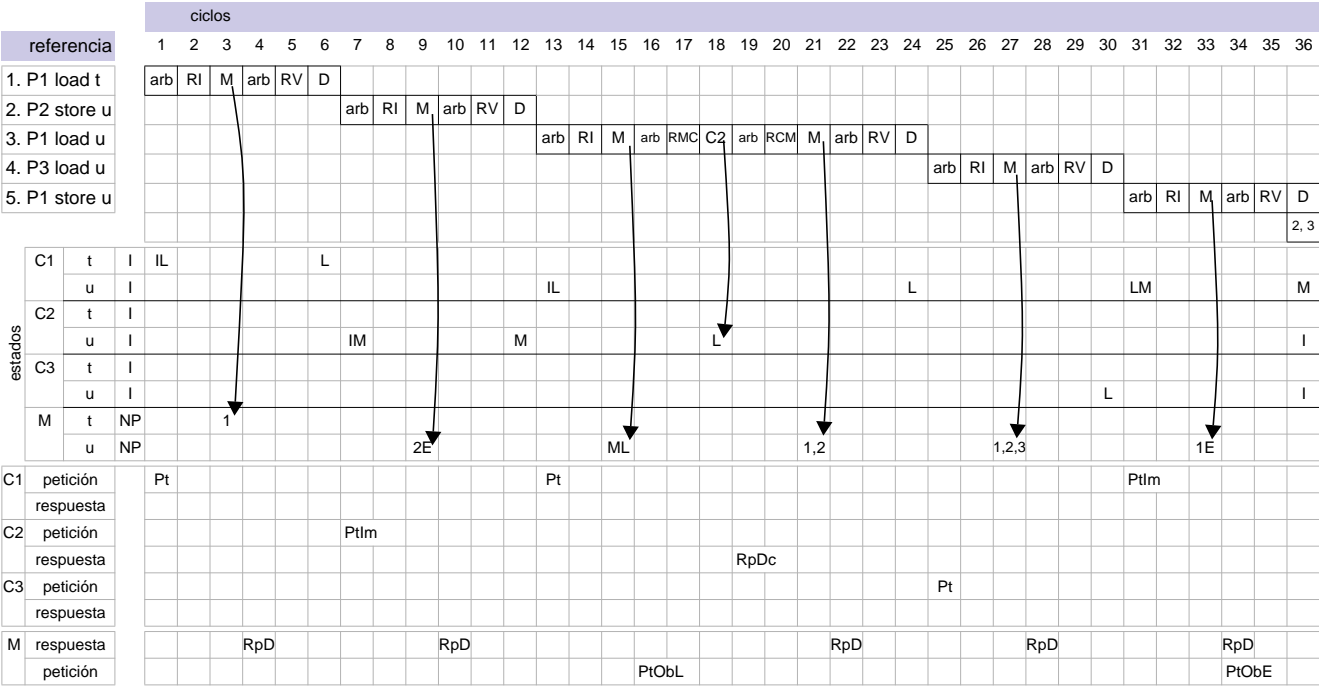


Figura 6.44 Protocolo de directorio MLI. Diagrama temporal de una secuencia de accesos a memoria.

Diagrama temporal simplificado

La idea de este diagrama es representar de forma más esquemática los mensajes y la temporalidad de los mismos. La representación es similar a la utilizada para el protocolo de directorio VI.

El acrónimo de una respuesta de un CC al CM se representan en el CM.

En la Figura 6.45 se muestra un diagrama temporal simplificado, donde se identifican los mensajes entre los CC y el CM y viceversa, correspondientes a la secuencia de accesos mostrada en la Tabla 6.12. También se indica el estado de los bloques en el CM y en los CC.

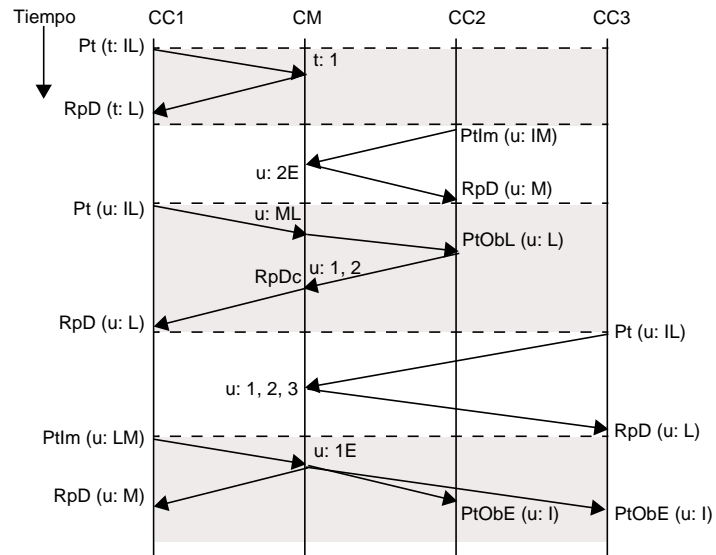


Figura 6.45 Protocolo de directorio MLI. Diagrama temporal simplificado de una secuencia de accesos a memoria concurrentes.

Animación

En la Figura 6.46 se muestra, mediante fotogramas, una animación de la secuencia de accesos a memoria de la Tabla 6.12. Además de las peticiones y respuestas, se muestran los cambios de estado de un bloque, utilizando diagramas de transición entre estados. Cuando un bloque no está almacenado en un contenedor de cache se supone que está en estado inválido. Para reducir la información representada en la figura, se distinguen solicitante, directorio y otros. Tampoco se muestra el estado transitorio MI en un CC, ya que no se utiliza en esta secuencia de accesos a memoria. Los mensajes y el acceso al directorio y memoria se identifican con una secuencia numérica.

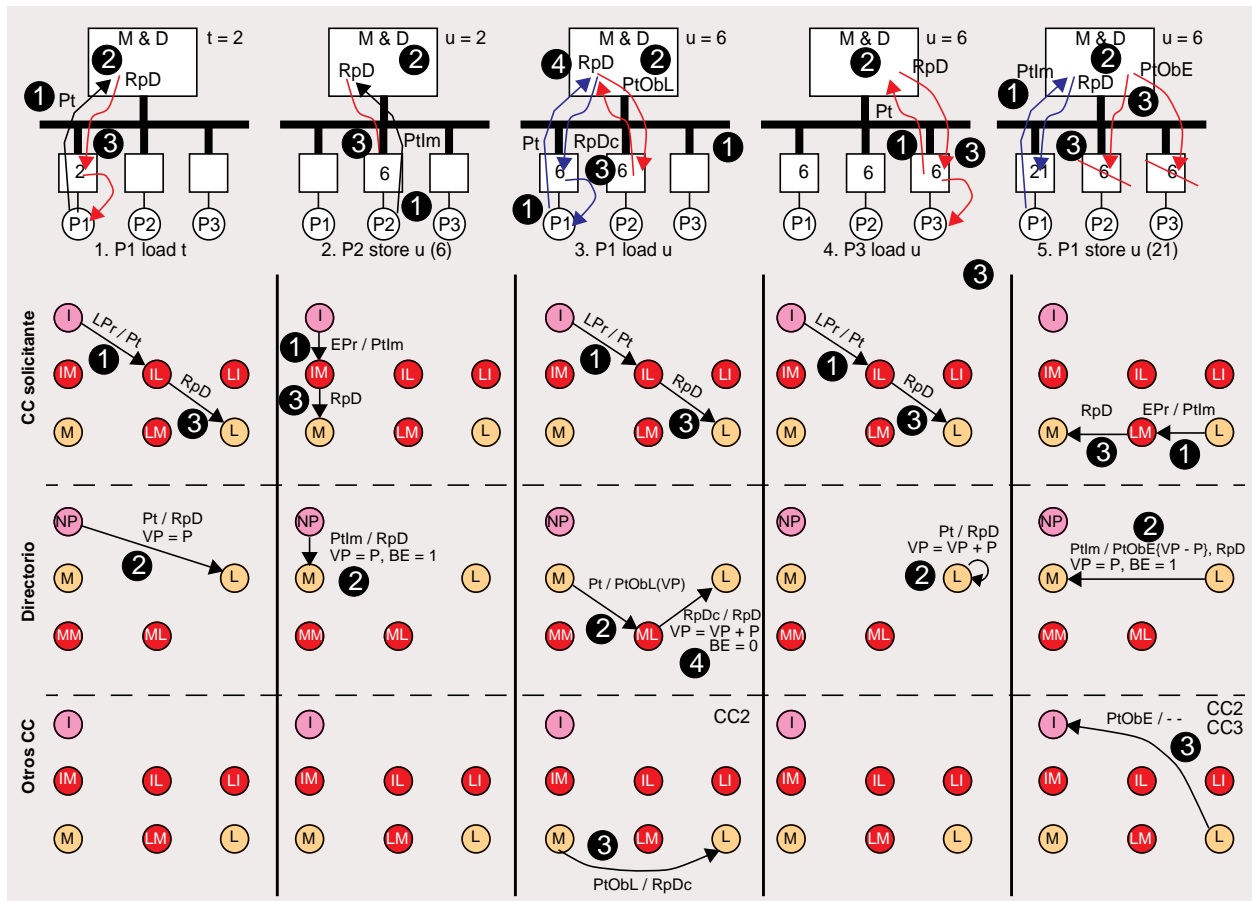


Figura 6.46 Protocolo de directorio MLI. Secuencia de accesos a memoria. Fotografías de las peticiones y respuestas y cambios de estado.

Verificación no formal de coherencia y consistencia

Coherencia de cache

Orden de programa en accesos a la misma posición de memoria. El procesador efectúa los accesos en el orden determinado por el L.M. El procesador se bloquea en un fallo de lectura o cuando requiere obtener la exclusividad de acceso al bloque, hasta que finaliza la transacción correspondiente.

Propagación de escrituras. Para efectuar una escritura en un bloque es necesario tener acceso al bloque en exclusividad (estado M). Ello garantiza que no hay copias del bloque. La exclusividad de acceso a un bloque se

obtiene al recibir la respuesta a una petición Ptlm. Las copias del bloque en otras caches se invalidan durante el procesado de la transacción Ptlm en el CM. La respuesta de cada petición de invalidación (PtObE) del CM, a los CC que tienen copia, está implícita al efectuar la emisión de la petición, ya que la red del CM a los CC mantiene el orden de los mensajes.

Serialización de escrituras (atomicidad de escritura).

- Punto de serialización. El árbitro del CM es el punto de serialización de las transacciones Ptlm. Las escrituras de un procesador, que tiene un bloque en exclusividad, quedan serializadas por el orden de programa. Se tiene garantía de que no existen copias del bloque.
- Consolidación de una escritura. Una transacción Ptlm está consolidada al finalizar la transacción (emitir respuesta al solicitante). Cuando una cache no tiene el bloque en exclusividad, el CM finaliza la transacción Ptlm una vez emite las peticiones de invalidación a las caches que tienen copia y la respuesta al CC solicitante. Debido a la propiedad de la red entre el CM y los CC, la emisión de las peticiones de invalidación por parte del CM tienen implícita la respuesta del CC correspondiente. Cuando una cache tiene el bloque en exclusividad, el CM finaliza la transacción Ptlm al recibir la respuesta del CC que tiene el bloque en exclusividad y responder al CC solicitante, además de actualizar la memoria. Una escritura a un bloque, al que se tiene acceso en exclusividad, está consolidada después de actualizar la cache. Por otro lado, una escritura es atómica. Una vez finalizada una transacción Ptlm y actualizada la cache, una lectura posterior del bloque (con o sin intención de modificar) por parte de otro procesador requiere emitir una petición al CM. El CM responde con el valor establecido por la escritura previa. En el lapso de tiempo entre obtener la exclusividad de acceso al bloque y el suministro del bloque, un procesador puede haber actualizado el bloque varias veces. Una lectura del procesador que tiene el bloque en exclusividad lee un valor consolidado.

Consistencia secuencial de memoria

Orden de programa. El procesador efectúa los accesos a cualquier posición de memoria en el orden determinado por el L.M.

- Lectura: Espera hasta que se obtiene el dato.
- Escritura: El árbitro del CM es el punto de ordenación de todas las transacciones Ptlm. La finalización de la transacción es una indicación de que la escritura está consolidada. Si no existe una copia en exclusividad, el CM, durante el procesado de una transacción de escritura, emite peticiones de observación de escritura o invalidación, si es el caso. La emisión de las peticiones de invalidación por parte del CM tienen implícita

la respuesta del CC correspondiente. Si un CC tiene el bloque en exclusividad, el CM espera la respuesta de este CC, la cual contiene el bloque, para emitir la respuesta al CC solicitante. En caso de acierto a un bloque en estado M, la escritura consolida al ser actualizada la cache (existe garantía de que no hay copias del bloque).

Atomicidad de las escrituras.

- Consolidación de una escritura: Para efectuar una escritura es necesario disponer de acceso exclusivo al bloque. El acceso exclusivo garantiza que no hay copias del bloque y se obtiene mediante una transacción Ptlm. El CM emite peticiones PtObE a las caches que tienen copia del bloque. La respuesta de estas cache está implícita en la emisión de la petición PtObE, ya que la RV es ordenada. Cuando una cache tiene el bloque en exclusividad, el CM finaliza la transacción Ptlm al recibir la respuesta del CC que tiene el bloque en exclusividad y responder al CC solicitante además de actualizar la memoria. Una vez obtenida la exclusividad se actualiza la cache y la escritura está consolidada.
- Suministro del valor en una lectura: Un fallo de lectura de un bloque (con o sin intención de modificarlo) requiere emitir una petición al CM. El valor devuelto en la transacción es el valor establecido en la última escritura consolidada, a la misma posición de memoria. El CM obtiene este valor de la memoria o de otra cache. Un acierto de lectura lee el valor de la copia en cache. Este valor ha sido establecido en un fallo de lectura previo o en una escritura previa del mismo procesador.

ORGANIZACIÓN DEL DIRECTORIO

Como hemos comentado interesa que las caches tengan un duplicado de las etiquetas para que el agente observador no tenga que competir con el agente procesador. El efecto se nota en mayor medida en una petición de observación de escritura cuando sólo requiere invalidación²². Notemos que cuando es necesario suministrar el bloque hay que acceder al campo de datos de la cache.

Este duplicado de etiquetas se puede utilizar como directorio, si en lugar de estar en las caches se ubica en memoria (Figura 6.47). En estas condiciones el mensaje de petición del CM a los CC identifica el contenedor en cache (conjunto y elemento dentro del conjunto). Por tanto, el agente observador puede acceder directamente al estado. Sin embargo, como hay que mantener coherente el duplicado de etiquetas, es necesario mantener el directorio

22. Por ejemplo, en el protocolo VI.

preciso notificando todas las expulsiones. Como el algoritmo de reemplazo está en las caches privadas, se notifica al directorio el contenedor que queda libre (elemento del conjunto).

Un beneficio es que el tamaño de un duplicado de etiquetas es mucho menor que un directorio con VP, ya que el número total de contenedores de las caches es mucho menor que el número de bloques de memoria. Una deficiencia, o desventaja de un duplicado de etiquetas, es que para cada petición se efectúa una búsqueda asociativa. Hay que acceder al duplicado de etiquetas de todas las caches. Es similar a la búsqueda que efectúan todos los agentes observadores cuando se utiliza un bus como red de interconexión.

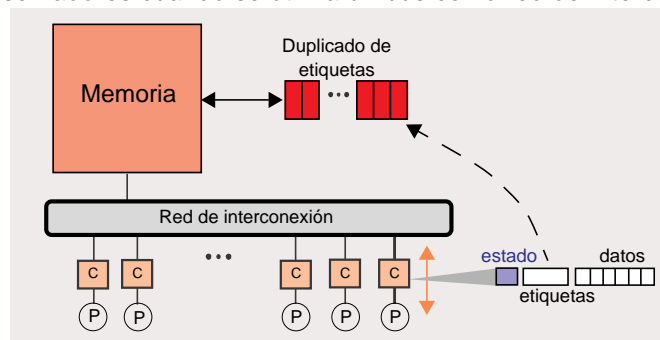


Figura 6.47 Duplicado de etiquetas de cache como directorio.

Ejercicio

Indique el número de comparaciones que se efectúan en un directorio que está organizado como un duplicado de etiquetas al procesar una petición de bloque.

Respuesta

Suponemos que todas las caches tiene la misma asociatividad. El número de comparaciones es igual al número de cache privadas por la asociatividad de las caches.

Ultimo nivel de cache con inclusión: directorio

Una alternativa para reducir la latencia de acceso a memoria es utilizar una cache compartida inclusiva. Esto es, el contenido de todas las caches privadas está almacenado en la cache compartida.

En una organización de este tipo, el directorio está asociado a la cache compartida. Al ser inclusiva se tienen identificados todos los bloques con copias en las caches privadas.

En la Figura 6.48 se muestra una organización del directorio con vectores de presencia. También se puede utilizar una organización con un duplicado de las etiquetas de las caches privadas.

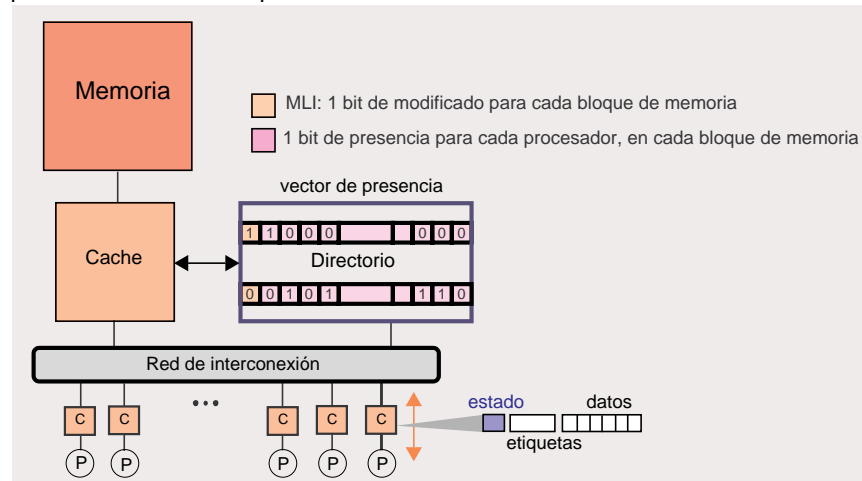


Figura 6.48 Multiprocesador con cache compartida inclusiva

Optimización. En lugar de almacenar en el duplicado de etiquetas las etiquetas de las caches privadas se puede almacenar el identificador del contendor en la cache compartida. Con ello se reduce el número de bits que debe almacenarse y el tamaño de las comparaciones. Para ello, hay que acceder a las etiquetas de la cache compartida antes de acceder al directorio. El acceso se puede efectuar en paralelo con el acceso al campo de datos de la cache compartida.

EJEMPLOS

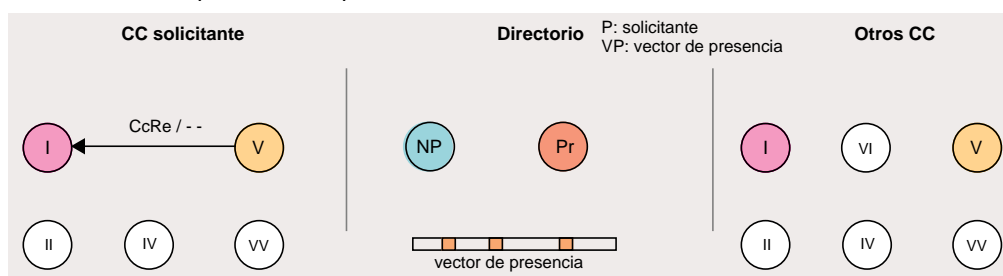
Protocolo de directorio VI. Expulsión silenciosa

Respecto de las expulsiones, debido a acciones de reemplazo en cache, podemos distinguir dos posibilidades: a) no silenciosa y b) silenciosa. Cuando se utiliza expulsión no silenciosa, el directorio recibe una notificación de la expulsión y un CC sólo recibe peticiones de observación de escritura de bloques almacenados en contenedores de la cache. Esto es, el directorio es preciso.

En una expulsión silenciosa el directorio no recibe notificación de la expulsión de un bloque. Esta característica determina que un CC pueda recibir una petición de observación de escritura (PtObE) de un bloque que no tiene almacenado en cache. Esto es, el directorio es impreciso. Por tanto, el agente observador, antes de invalidar el contenido de un contenedor, debe comprobar que el contenedor almacena el bloque al que hace referencia la petición de invalidación²³.

Pregunta 1: En un protocolo de directorio VI, diseñe los autómatas de cambio de estado de un bloque en un CC y en el CM cuando se utiliza expulsión silenciosa.

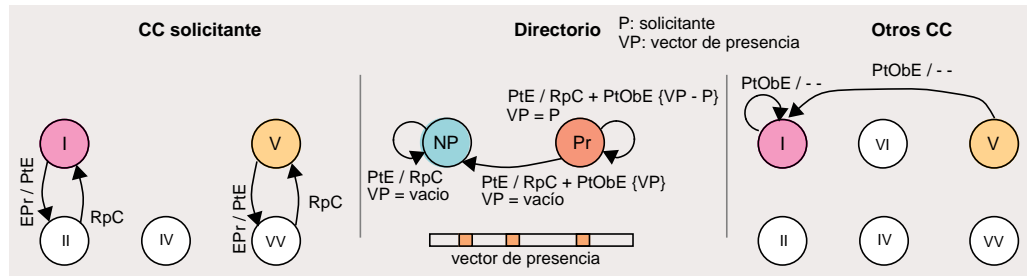
Respuesta: En la siguiente figura se muestran los diagramas de transiciones entre estados al efectuar una expulsión en un CC. Como la expulsión es silenciosa no se notifica al CM, sólo se produce un cambio de estado en el CC del cual se expulsa el bloque. Por tanto, el VP no se actualiza.



En estas condiciones un CC puede recibir una petición de observación de escritura cuando no almacena el bloque en cache. En los siguientes diagramas de transiciones entre estados de un bloque se representan las transiciones en una petición de escritura, que es la que determina que el CM efectúe una petición PtObE.

23. En este protocolo, si se invalida el bloque sin efectuar la comprobación se sigue manteniendo la coherencia. Recordemos que memoria siempre está actualizada.

Los diagramas de transiciones entre estados en el CC del solicitante y en el CM son los mismos que en la Figura 6.17 y la Figura 6.18. La diferencia está en el diagrama de otros CC.



Un bloque no presente en una cache se identifica que está en el estado I. Al utilizar expulsión silenciosa, un CC puede recibir una petición PtObE en el estado V y en el estado I. En cualquiera de los dos casos el estado final es I.

Por otro lado, el CM puede recibir una petición Pt de un CC que está identificado en el VP. En este caso el CM suministra el bloque. También, el CM puede recibir una petición PtE de un CC que está identificado en el VP, pero la cache no tiene copia del bloque. El CM actualiza la memoria e invalida las copias en otras caches. En el VP se sigue identificando al CC que ha efectuado la petición PtE, aunque no tiene copia del bloque.

Protocolo de directorio MLI

Una secuencia de accesos a memoria referencia las variables u y t que se ubican en bloques distintos. Todas las cache utilizan la misma organización y tienen el mismo tamaño. Los bloques que contienen las variable u y t al almacenarse en las caches se ubican en el mismo contenedor de cache.

El bloque que contiene la variable t está almacenado en la cache C1 y el bloque que contiene la variable u está almacenado en la cache C2. El estado de estos dos bloques en las caches es L.

Pregunta 1: Utilice el protocolo de directorio con invalidación y escritura retardada y una tabla similar a la Tabla 6.12 para mostrar los mensajes de una transacción y estados de los bloques en las caches. La secuencia de accesos a memoria se indica en la tabla mostrada en la respuesta. Indique el número de expulsiones que se producen.

Respuesta: Para realizar el primer acceso a memoria el CC del procesador P1 debe obtener la exclusividad en el acceso al bloque. Para ello el CC genera una petición de bloque con intención de modificación (PtIm). El segundo acceso a memoria es mimético al primer acceso, estando la diferencia en el procesador que efectúa el acceso y la variable accedida.

acceso	C 1 est.	C 2 est.	Red ida	mem. var.	VP	E	Red RMC	Red RCM	sum.	Red vuelta	C 1 var.	est.	C 2 var.	est.
1. P1 store t	LM		PtIm	t	1, 0	1			mem.	RpD	t	M		
2. P2 store u		LM	PtIm	u	0, 1	1			mem.	RpD			u	M
3. P2 load t		MI	PtXm	u	0, 0	0			C2	RpX			u	I
		IL	Pt	t	1, 1	0	RpObL, 1	RpDc	C1	RpD	t	L	t	L
4. P2 store u		LI	PtXI	t	1, 0	0				RpX			t	I
		LM	PtIm	u	0, 1	1			mem.	RpD			u	M
5. P1 load t											t	L		
6. P2 load u													u	M

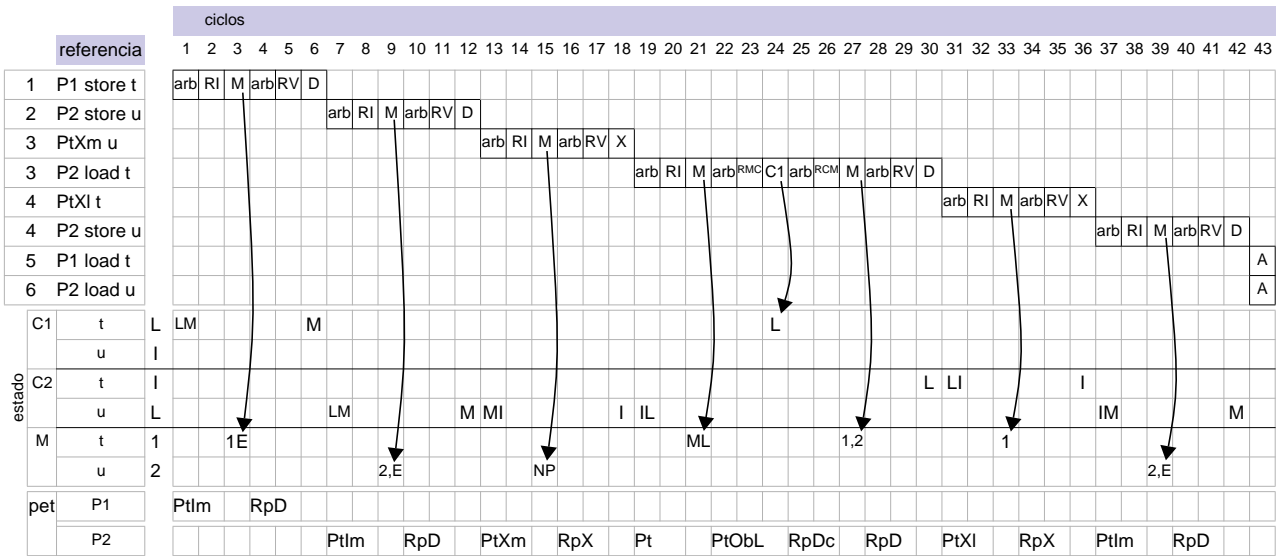
En el tercer acceso el CC2 detecta un fallo y es necesaria una acción de reemplazo. El bloque que se expulsa está en el estado M. Por tanto, es necesario actualizar la memoria (PtXm). Después de la transacción de actualización de memoria se efectúa la petición de bloque. El bloque solicitado está en estado M en otra cache. El CM solicita el bloque a la cache correspondiente y espera la recepción de una respuesta. Después de recibir la respuesta, el CM actualiza el directorio y emite una respuesta al CC que efectuó la petición.

El cuarto acceso es una escritura y el CM determina que es un fallo. La acción de reemplazo requiere expulsar un bloque en estado L. Después de expulsar el bloque, se inicia una transacción con una petición de bloque con intención de modificación. Al finalizar la transacción el estado del bloque es M. Los dos últimos accesos a memoria son operaciones de lectura que son aciertos en cache. En el primer caso el bloque está en estado L y en el segundo el bloque está en estado M.

En total se efectúan dos expulsiones. Una corresponde a un bloque en estado M (tercer acceso) y otra corresponde a un bloque en estado L (cuarto acceso). Las dos expulsiones se efectúan en la cache del procesador P2.

Pregunta 2: Muestre un diagrama temporal de la secuencia de accesos. Utilice una fila para representar sólo la expulsión y en la siguiente fila represente la transacción correspondiente a la petición que ha inducido la expulsión.

Respuesta: La siguiente figura muestra el diagrama temporal que se solicita. Recordemos que el acrónimo X en la última fase de una transacción indica la respuesta a una expulsión.



EJERCICIOS

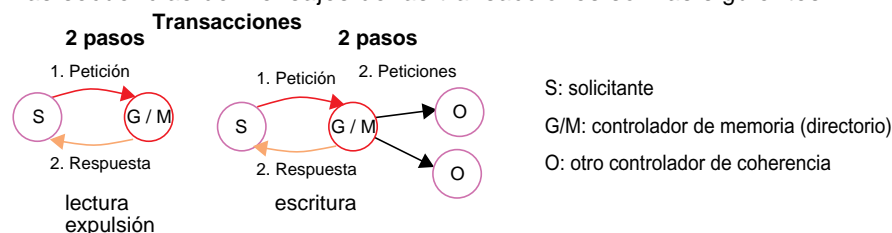
Descripción de un protocolo de directorio VI denominado A

Suponga un multiprocesador donde las caches privadas son de mapeo directo y utilizan escritura inmediata. Las redes de interconexión entre las caches y el módulo de memoria son de tipo crossbar y mantienen el orden de los mensajes emitidos. El multiprocesador utiliza un directorio para mantener la coherencia y el protocolo de coherencia es de invalidación (VI).

Las caches privadas de los procesadores son bloqueantes. En un fallo de cache o en una escritura se suspende la interpretación de instrucciones y se reanuda al finalizar la transacción.

El directorio utiliza un vector de presencia (VP) por bloque. El vector de presencia es un vector de bits, con tantos bits como procesadores y cada bit está asociado a un procesador.

Las secuencias de mensajes de las transacciones son las siguientes:



Las peticiones de procesador y los mensajes utilizados en la transacciones para mantener la coherencia son:

Procesador	Controlador de cache (CC)		Controlador de memoria (CM)	
Peticiones	Peticiones del CC al CM	Respuestas del CM al CC	Peticiones del CM a los CC	Acciones
LPr : lectura	Pt : petición de bloque	RpD: respuesta con el bloque	PtObE: petición de observación de escritura	Actualización del directorio
EPr : escritura	PtE: petición de escritura de un dato	RpC: respuesta de confirmación		Dev: actualización de memoria
	PtX: petición de expulsión	RpX: respuesta de confirmación a una petición PtX		

El controlador de cache también efectúa acciones de reemplazo cuando es necesario (CcRe). En una acción de reemplazo se distingue la acción de notificación al directorio, ya que éste es preciso. En una petición PtX se actualiza el directorio.

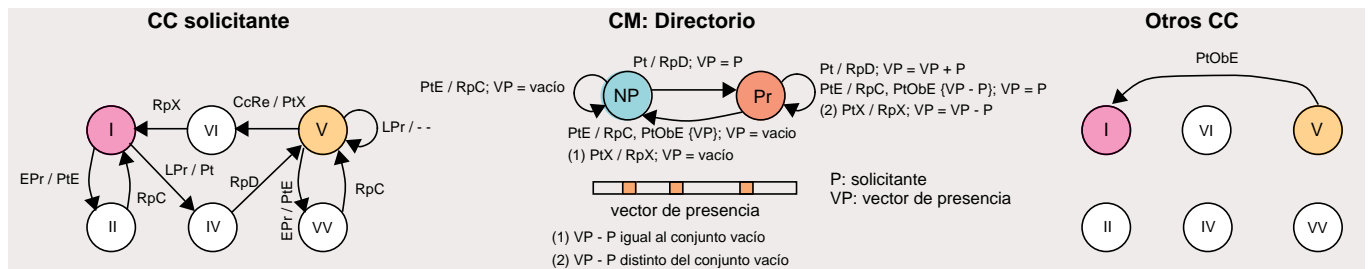
Cuando el servicio de un acceso a memoria requiere un reemplazo, éste se efectúa antes de gestionar el acceso a memoria que produce la acción de reemplazo.

Las fases de cada uno de los mensajes son:

mensajes	ciclos				
	1	2	3		
Pt, PtE, PtX	arb	RI	M	arb: arbitraje en la red correspondiente	M: memoria (directorio)
RpD, RpC, RpX	arb	RV	D ó X	RI: red de peticiones desde los CC al CM	D: dato (RpD)
PtObE	arb	RMC	Cx	RV: red de respuestas desde el CM a los CC	X: confirmación (RpX)
				RMC: red de peticiones desde el CM a los CC	Cx: cache, donde x es el ordinal de la cache que recibe PtObE

En un CC, para distinguir, en una transacción, entre la emisión de un mensaje de petición y la recepción de una respuesta, se utilizan estados transitorios (II, IV, VV, VI). En el CM no es necesario ya que no espera respuestas.

En los siguientes diagramas de estados se muestran todas las transiciones entre estados, ya sean estables o transitorios, de un bloque en cache y en el directorio.



En este multiprocesador supondremos que sólo existe un acceso a memoria en un instante determinado.

En un diagrama temporal se muestran en la parte superior las fases de los mensajes de una transacción, en la parte central se especifica el estado de los bloques en las cache y en el directorio. En la parte inferior se etiqueta el mensaje o la respuesta que se representa en la parte superior. El estado de los bloques en cache o en el directorio se indica sólo cuando hay un cambio de estado.

Fases y eventos	Especificaciones
arb	Se especifica el estado transitorio del bloque.
M	Se especifica el VP utilizando el ordinal de los procesadores cuyas caches tienen copia del bloque.
D ó X	Se especifica el estado estable del bloque en la cache cuyo CC ha efectuado la petición.
x	Se especifica la modificación de estado determinada por la petición del CM
Transacción de 2 pasos con peticiones del CM	Se utiliza una fila para indicar la respuesta del CM y otra fila para indicar todas las peticiones del CM.
Reemplazo	Determina una expulsión: se especifica en la columna etiquetada como referencia. Para ello, se utilizan dos filas contiguas. En la primera fila se especifica la expulsión (PtX) y en la segunda fila la petición que determina la expulsión.
Mensaje	Se indica en la columna correspondiente a arb.

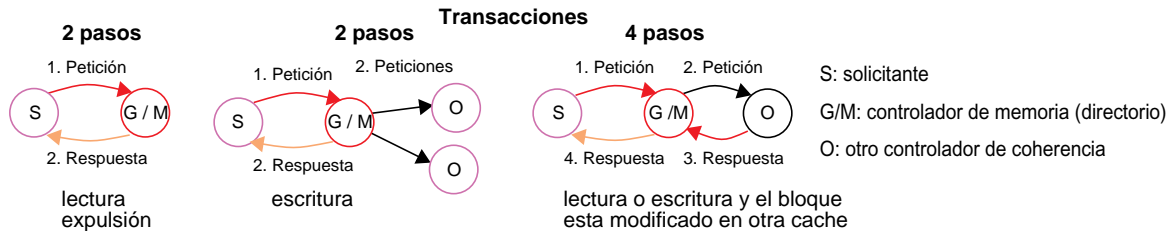
Descripción de un protocolo de directorio MLI denominado B

Suponga un multiprocesador donde las caches privadas son de mapeo directo y utilizan escritura retardada. Las redes de interconexión entre las caches y el módulo de memoria son de tipo crossbar y mantienen el orden de los mensajes emitidos. El multiprocesador utiliza un directorio para mantener la coherencia y el protocolo de coherencia es de invalidación (MLI).

Las caches privadas de los procesadores son bloqueantes. En un fallo de cache o en una solicitud de exclusividad se suspende la interpretación de instrucciones y se reanuda al finalizar la transacción.

El directorio utiliza un vector de presencia (VP) y un bit de exclusividad (BE) por bloque. El vector de presencia es un vector de bits, con tantos bits como procesadores y cada bit está asociado a un procesador. El bit de exclusividad se utiliza para indicar que sólo existe una copia del bloque en una cache privada, la cual está identificada en el vector de presencia.

Las secuencias de mensajes de las transacciones son las siguientes:



Las peticiones de procesador y los mensajes utilizados en la transacciones para mantener la coherencia son:

Procesador	Controlador de cache (CC)		Controlador de memoria (CM)		
	Peticiones del CC al CM	Respuestas del CM al CC	Peticiones del CM a los CC	Respuestas del CC al CM	Acciones
LPr : lectura	Pt : petición de bloque	RpD: respuesta con el bloque a una petición Pt o Ptlm	PtObE: petición de observación de escritura, inducida por una petición Ptlm	RpDc: respuesta con el bloque a una petición PtObL o PtObE y el estado del bloque en cache es M	Actualización del directorio
EPr : escritura	Ptlm: petición de bloque con intención de modificarlo	RpX: respuesta de confirmación a una petición PtXm o PtXI	PtObL: petición de observación de lectura, inducida por una petición Pt y el estado del bloque en el directorio es M		Dev: actualización de memoria
	PtXm: petición de expulsión de un bloque en estado M				
	PtXI: petición (notificación) de expulsión de un bloque en estado L				

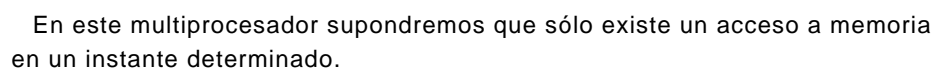
Cuando el servicio de un acceso a memoria requiere un reemplazo, éste se efectúa antes de gestionar el acceso a memoria que produce la acción de reemplazo.

mensajes	ciclos		
	1	2	3
Pt, Ptlm, PtXm, PXI	arb	RI	M
RpD, RpX	arb	RV	D ó X
PtObE, PtObL	arb	RMC	Cx
RpDc	arb	RCM	M

RCM: red de respuestas de los CC al CM

Cx: cache, donde x es el ordinal de la cache

En los siguientes diagramas de estados se muestran todas las transiciones entre estados, ya sean estables o transitorios, de un bloque en cache y en el directorio. En el protocolo que se describe, el bit de exclusividad del directorio se activa cuando una cache solicita el bloque para actualizarlo.



En un diagrama temporal se muestran en la parte superior las fases de los mensajes de una transacción, en la parte central se especifica el estado de los bloques en las caches y en el directorio. En la parte inferior se etiqueta el mensaje o la respuesta que se representa en la parte superior. El estado de los bloques en cache o en el directorio se indica sólo cuando hay un cambio de estado.

Fases y eventos	Especificaciones
arb	Se especifica el estado transitorio del bloque.
M	Se especifica el VP utilizando el ordinal de los procesadores cuyas caches tienen copia del bloque y si el bloque lo tiene una cache en exclusividad se añade la letra E. La especificación se efectúa la última vez que se visita el directorio en una transacción.
D ó X	Se especifica el estado estable del bloque en la cache cuyo CC ha efectuado la petición.
Cx	Se especifica la modificación de estado determinada por la petición del CM
Transacción de 2 pasos	Se utiliza una fila para indicar la respuesta del CM y otra fila para indicar todas las peticiones del CM.
Reemplazo	Determina una expulsión: se especifica en la columna etiquetada como referencia. Para ello, se utilizan dos filas contiguas. En la primera fila se especifica la expulsión (PtXm o PtXI) y en la segunda fila la petición que determina la expulsión.
Mensaje	Se indica en la columna correspondiente a arb.

Ejercicio

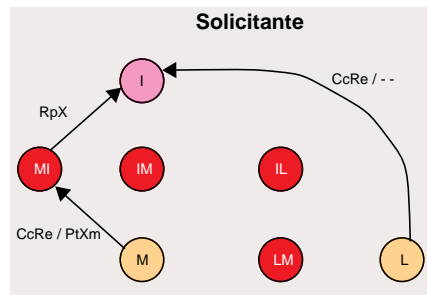
6.1

Utilice el protocolo de directorio MLI denominado B descrito en este capítulo. Suponga la siguiente secuencia de accesos.

accesos		
1. P1 load t	5. P2 store u	Las variables u y t se ubican en bloques distintos.
2. P1 store t	6. P1 load t	Estos bloques al almacenarse en cache se ubican en el mismo contenedor.
3. P2 store u	7. P2 load u	La variable t no está inicialmente almacenada en ninguna cache y el valor en memoria es t = 12.
4. P2 load t		La variable u está almacenada inicialmente en la cache de P2, en el estado L y el valor es u = 9.

Pregunta 1: Muestre en un diagrama temporal la secuencia de mensajes que genera cada transacción y los cambios de estado de los bloques en las caches y en el directorio al ejecutarse la anterior secuencia de accesos a memoria.

Un diseñador observa que el número de expulsiones de bloques en estado L es mucho mayor que el número de bloques expulsados en estado M. En consecuencia, para reducir el tráfico en las redes de interconexión decide no notificar al directorio la expulsión de un bloque en estado L. Esta característica se denomina expulsión silenciosa y el directorio se denomina impreciso. Esto es, el directorio no indica de forma precisa las copias del bloque en las cache privadas y siempre indica un conjunto mayor de CC que tienen copia que el que realmente existe. En contraposición, el directorio denominado B dice que es preciso.



Pregunta 2: En un directorio impreciso, un CC puede recibir una petición de observación de escritura (PtObE) de un bloque que no tiene almacenado en cache. Enumere los estados en los cuales un CC puede recibir una petición PtObE del CM e indique la respuesta en cada caso.

Pregunta 3: Para caches privadas de mapeo directo o de mapeo asociativo por conjuntos y los dos posibles tipos de directorio (preciso o impreciso) indique si es necesario comprobar el contenido de la cache (comparar etiquetas) cuando se recibe una petición del CM.

Pregunta 4: Justifique si el CM puede recibir por parte de un CC, que tiene un bloque en el estado I, una petición de este bloque y el directorio no tiene identificado al CC en el VP.

Ejercicio 6.2

Utilice el protocolo de directorio MLI denominado B descrito en este capítulo. Suponga la siguiente secuencia de accesos.

accesos		
1. P1 load t	5. P2 store u	<p>Las variables u y t se ubican en bloques distintos. Las variables t y v se ubican en el mismo bloque.</p> <p>Estos bloques al almacenarse en cache se ubican en el mismo contenedor.</p> <p>La variable u no está inicialmente almacenada en ninguna cache y el valor en memoria es u = 12.</p> <p>El bloque que contiene las variables t y v está almacenado inicialmente en la cache de P2, en el estado L y los valores son t = 9 y v = 125.</p>
2. P1 store v	6. P1 load t	
3. P2 store u	7. P2 load u	
4. P2 load t		

Pregunta 1: Muestre en un diagrama temporal la secuencia de mensajes que genera cada transacción y los cambios de estado de los bloques en las caches y en el directorio al ejecutarse la anterior secuencia de accesos a memoria.

Un diseñador observa que en una secuencia load A - store A, efectuada por un procesador, no siendo en este intervalo de tiempo una variable compartida, se requieren dos transacciones. La primera transacción para obtener el bloque y ejecutar la instrucción load y la segunda transacción para obtener la exclusividad y ejecutar la instrucción store.

Protocolo modificado. El diseñador propone que, si no existen copias del bloque en las caches privadas, el controlador de memoria (CM), al recibir una petición Pt, suministre el bloque en exclusividad. Para ello, añade en el protocolo una nueva respuesta en el CM, denominada RpDe. El CM al emitir una respuesta RpDe, activa el bit correspondiente del procesador en el VP y también activa el BE.

Un controlador de coherencia (CC) al recibir la respuesta RpDe, a una petición Pt, determina que el bloque ha sido suministrado en exclusividad y establece el estado en consecuencia.

Pregunta 2: ¿Cuál es la transición entre estados que se efectúa en el directorio al responder con RpDe a una petición Pt?. ¿Cuál es la transición entre estados que se efectúa en el CC al emitir la petición Pt y recibir una respuesta RpDe?.

Para analizar el rendimiento se utiliza la siguiente secuencia de accesos a memoria.

accesos	
1. P1 load t	Las variables u y t se ubican en bloques distintos.
2. P1 load u	Estos bloques al almacenarse en cache se ubican en el mismo contenedor.
3. P1 store u	Las variables u y t no están inicialmente almacenada en ninguna cache
4. P1 load t	

Pregunta 3: Indique la secuencia de mensajes de peticiones y respuestas necesaria para servir cada petición en el protocolo original y en el protocolo modificado. Cuando, para servir un acceso a memoria, sea necesario expulsar un bloque, utilice dos filas consecutivas. En la primera fila especifique la expulsión y en la segunda fila especifique el acceso a memoria.

Supongamos que la transmisión del bloque desde la cache al CM ocupa durante varios ciclos la red de ida (RI). Entonces, en el protocolo original y en el modificado, considere las siguientes latencias en las peticiones PtXm y PtXl.

mensajes	ciclos					
	1	2	3	4	5	6
PtXm	arb	RI	RI	RI	RI	M
PtXl	arb	RI	M			

Pregunta 4: Para la secuencia de accesos a memoria anterior, calcule la duración en ciclos utilizando el protocolo original y utilizando el protocolo modificado.

Ejercicio

6.3

En el protocolo de directorio MLI denominado B descrito en este capítulo, un diseñador analiza una secuencia load A - load B, efectuada por un procesador. Las variables A y B no están siendo compartidas y están ubicadas en bloques distintos. Sin embargo, los bloques que contienen estas variables, cuando se mapean en cache, se almacenan en el mismo contenedor. En una modificación previa del protocolo A, al expulsar el bloque que contiene la variable A se actualiza memoria (ejercicio 6.2). En este nuevo diseño, el ingeniero pretende que al expulsar el bloque, que contiene la variable A, sea suficiente con actualizar el directorio. Esto es, la memoria no se actualiza. Otro ejemplo es la expulsión de bloques que contienen instrucciones no compartidas. Instrucciones que son sólo accedidas desde un procesador.

Protocolo modificado. El diseñador añade el estado E al conjunto de estados de un bloque en las caches privadas. El estado E identifica que es la única copia del bloque en una cache privada y que se tiene acceso al bloque en exclusividad.

Si no existen copias del bloque en las caches privadas, el controlador de memoria (CM), al recibir una petición Pt, suministra el bloque en exclusividad. Para este caso, el protocolo utiliza una nueva respuesta en el CM, denominada RpDe. El CM al emitir una respuesta RpDe, activa el bit correspondiente del procesador en el VP y también activa el BE.

Un controlador de coherencia (CC) al recibir la respuesta RpDe, a una petición Pt, determina que el bloque ha sido suministrado en exclusividad y establece el estado en consecuencia.

Además, el diseñador modifica el autómata en el directorio para procesar una petición PtXI cuando un bloque, en el directorio, está en estado M.

Pregunta 1: ¿Cuál es la transición entre estados que se efectúa en el directorio al responder con RpDe a una petición Pt?. ¿Cuál es la transición entre estados que se efectúa en el CC al emitir la petición Pt y recibir una respuesta RpDe?.

Pregunta 2: Suponga que no existe ningún acceso a la misma posición de memoria por parte de otro procesador. Entonces, el mismo procesador que ha emitido una petición Pt efectúa una escritura a una posición del bloque que ha

recibido con la respuesta $RpDe$. ¿Cuál es la transición entre estados, y, si es el caso, la petición al directorio, que efectúa el CC cuando recibe una escritura del procesador (EPr)?

En las siguientes dos preguntas especifique sólo las transiciones adicionales, si existen, que determinan la modificación del protocolo.

Pregunta 3: En un grafo de estados muestre las transiciones en un CC al recibir del CM una petición $PtObE$ o una petición $PtObL$.

Pregunta 4: Especifique las transiciones entre los estados de un bloque en el directorio y respuestas del CM al recibir una petición de expulsión. Especifique también las transiciones en el CC.

Suponga la siguiente secuencia de accesos a memoria. Las variables t , u y s están ubicadas en bloques distintos. Las variables t y w están ubicadas en el mismo bloque. Al mapearse en cache los tres bloques utilizan el mismo contenedor e inicialmente no están almacenados en ninguna cache.

accesos	1. P1 load t	2. P2 load u	3. P1 store w	4. P2 load u	5. P3 store u	6. P4 load s	7. P4 load u
---------	--------------	--------------	---------------	--------------	---------------	--------------	--------------

Pregunta 5: Indique la secuencia de mensajes de peticiones y respuestas necesaria para servir cada petición en el protocolo original (A) y en el protocolo modificado. Cuando, para servir un acceso a memoria, sea necesario expulsar un bloque, utilice dos filas consecutivas. En la primera fila especifique la expulsión y en la segunda fila especifique el acceso a memoria.

Pregunta 6: Para la secuencia de accesos a memoria anterior, calcule la duración en ciclos utilizando el protocolo original y el protocolo modificado. En las peticiones $PtXm$ y $Ptxl$ considere las latencias indicadas en el ejercicio 6.2.