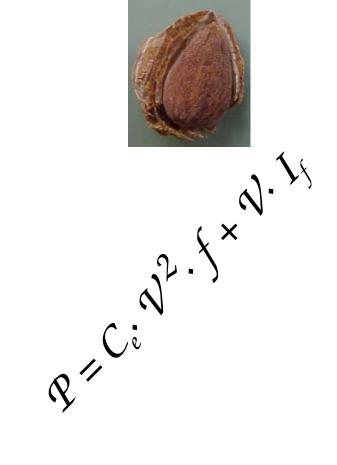




Multiprocesadores



J.M. Llabería

© Copyright 2014, 2015 los autores, Universidad Politécnica de Cataluña

Contenido

Capítulo 7	Concurrencia y paralelismo en un protocolo de directorio con una red denada	d or- 419
	Concurrencia	419 420 421
	Protocolo de directorio VI Organización del multiprocesador Transiciones entre estados estables de los controladores Concurrencia en el controlador de memoria. Identificación de posibles cruces de peticiones Estados transitorios y transiciones en el controlador de memoria Estados y transiciones en los controladores de coherencia Diagramas completos de cruce de peticiones Tablas de estados y transiciones Representación de transacciones y transiciones entre estados.	426 426 427 427 428 430 431 434 435 436
	Protocolo de directorio MLI Organización del multiprocesador Transiciones entre estados estables de los controladores Concurrencia en el controlador de memoria Identificación de posibles cruces de peticiones Estados transitorios y transiciones en el controlador de memoria Estados transitorios y transiciones en los controladores de coherencia Diagramas completos de cruces de peticiones Tablas de estados y transiciones Representación de cambios de estado y transiciones en una secuencia de accesos a memoria	441 442 443 443 444 447 450 452 454
	Paralelismo	460 461 468 472

Ejemplos	478
Protocolo de directorio VI. Diagramas temporales simplificados	478
Protocolo de directorio MLI. Cruces de peticiones	478
Protocolo de directorio VI. Expulsión silenciosa	483
Protocolo MLI: diagramas temporales con cruces de peticiones	485
Protocolo de directorio MLI. Diagramas temporales simplificados	488
Ejercicios	491

Capítulo 7 Concurrencia y paralelismo en un protocolo de directorio con una red ordenada

El diseño descrito en el Capítulo 6 utiliza como hipótesis que, en un ciclo determinado, sólo existe un acceso a memoria en el multiprocesador. En este capítulo se incrementa la concurrencia permitiendo que todos los procesadores efectúen accesos a memoria en paralelo.

Como existe concurrencia un CC puede recibir una petición del CM y tener pendiente una petición al mismo bloque u otro bloque. De forma similar, el CM al recibir una petición puede estar procesando una petición previa, de otro CC, al mismo bloque u otro bloque. Por otro lado, el CM puede necesitar procesar peticiones absoletas teniendo en cuenta el vector de presencia o el estado actual del bloque.

Finalmente, para incrementar el número de transacciones por unidad de tiempo se utiliza la técnica de paralelismo. Para ello se dispone de varios módulos de memoria, cada uno con su CM. Cada módulo de memoria puede ser accedido de forma independiente y el directorio está distribuido entre los CM. En consecuencia, se pueden estar sirviendo tantas peticiones de los CC como módulos de memoria.

CONCURRENCIA

Suponemos un único módulo de memoria con el CM correspondiente. El CM procesa concurrentemente peticiones a bloques distintos y serializa peticiones al mismo bloque.

Hipótesis en la descripción del protocolo

Un procesador interpreta las instrucciones en orden de programa y se bloquea al detectar un riesgo y en un acceso a memoria que requiere acceder al directorio.

Los procesadores realizan peticiones concurrentes al sistema de memoria.

Cuando una transacción requiere un reemplazo, en primer lugar se efectúa la expulsión del bloque. Una vez finalizada, se efectúa la petición del bloque que determina la expulsión¹.

El CM puede estar procesando peticiones concurrentemente a bloques distintos.

Redes de interconexión

Hay un árbitro en la RI. La red se construye mediante conexiones punto a punto y se encola una petición por ciclo en la CP. El CM procesa las peticiones de la CP en orden de llegada (FIFO).

En la Figura 7.1 se muestra un esquema del multiplexor que hay en la entrada de la CP y un diagrama temporal con la acción de arbitraje. Dos CC solicitan encolar una petición en el mismo ciclo. El árbitro (arb) ordena las peticiones. La espera de una de las peticiones para acceder a la red se indica representando en ciclo consecutivos la fase de arbitraje (arb, parte derecha de la Figura 7.1). En este ejemplo, se supone que el acceso a memoria ocupa dos ciclos. Por tanto, la latencia de iniciación es dos ciclos. En consecuencia, peticiones concurrentes de acceso a memoria se encolan en la CP.



Figura 7.1 Red de peticiones de los CC al CM (RI) y red de peticiones y respuesta del CM a los CC (RV).

1. Si se utiliza un buffer de expulsiones (BEX), el contenedor de cache queda libre y se pueden emitir las dos peticiones en secuencia, pero en orden inverso. Si sólo hay un CM se procesan en serie, aunque concurrentemente. Si hay varios CM se pueden procesar en paralelo cuando el bloque expulsado y el solicitado no se almacenan en el mismo módulo de memoria. El procesador reanuda la ejecución al llegar la respuesta a la petición del bloque. La petición debida a la expulsión queda en la sombra. Si una nueva referencia del procesador accede al bloque que se expulsa, al considerarse que está en cache (BEX), puede bloquearse o servirse como acierto, en función de la circuitería disponible, del estado y de las peticiones que se hayan procesado mientras se espera la respuesta del CM.

En ocasiones el CM, al procesar una petición de un CC, necesita emitir peticiones y una respuesta a los CC. El conjunto de CC destinatarios de las peticiones y el CC destinatario de la respuesta es disjunto. Entonces, las peticiones y la respuesta se transmiten utilizando la red de vuelta (RV) y en paralelo².

La RV se implementa mediante conexiones punto a punto entre el CM y los CC³. Esta red garantiza que los mensajes (peticiones y respuestas) emitidos por el CM llegan a un CC en el mismo orden que son emitidos. Esto es, además de que se mantiene el orden entre peticiones y se mantiene el orden entre respuestas, también se mantienen el orden entre una petición y una respuesta y viceversa.

Coherencia. El CM es el punto de ordenación de las transacciones iniciadas por los CC. El CM propaga una transacción de escritura de un bloque a los CC. La red RV mantiene el orden de emisión de los mensajes del CM a los CC. Por tanto, todos los CC observan las peticiones de observación de escritura del CM en el mismo orden lógico y no es necesario que los CC respondan a las peticiones de invalidación. Una escritura está consolidada cuando el CM emite la respuesta al CC que ha iniciado la transacción. El CM serializa las transacciones a un mismo bloque⁴.

Consistencia. El CM y la transmisión en orden de los mensajes en la RV determinan un orden lógico global de las transacciones. Una escritura está consolidada cuando el CC recibe la respuesta del CM. Una petición de lectura lee el valor establecido por la instrucción store previa a la misma dirección.

Concurrencia en el controlador de memoria

El CM siempre responde a una petición de un CC. Por otro lado, un CC también responde a las peticiones del CM, si es el caso. En particular, un CC dispone del bloque que quiere expulsar hasta que se recibe la respuesta del CM a la petición de expulsión⁵.

La llegada de una nueva petición en el CM, antes de haber inyectado en la RV los mensajes de coherencia, correspondientes a una petición anterior, sólo requiere capacidad de almacenamiento en la entrada del controlador de memoria.

- 2. Por ejemplo, en el protocolo de directorio VI.
- 3. En el protocolo MLI la RV implementa las redes lógicas RMC y RV (Capítulo 6).
- 4. El CM no procesa otra petición a un bloque hasta que ha emitido la respuesta a la petición previa que accede al mismo bloque.
- 5. Esta característica facilita la gestión de un cruce de peticiones, denominado tardío, en el CC.

La CP se utiliza para almacenar peticiones mientras el CM está procesando una petición. Supondremos que se dispone de la capacidad de almacenamiento suficiente para el peor caso⁶. De esta forma no es necesario entrar en detalles de control de flujo de mensajes desde los CC a un CM.

En la Figura 7.2 se muestra la concurrencia en el procesado de peticiones en el CM. Las dos primera peticiones de los CC se efectúan en el mismo ciclo y hay que arbitrar. La segunda petición necesita esperar en la CP para ser procesada. Esta petición se procesa cuando el CM finaliza el procesado de la petición anterior. El procesado de la tercera petición también debe esperar en la CP.

		(iclo	S									
referencia	1	2	3	4	5	6	7	8	9	10	11	12	13
P1 store t	arb	RI	М	М	arb	RV	С						
P2 load v	arb	arb	RI	СР	М	М	arb	RV	D				
P3 store u			arb	RI	СР	СР	М	М	arb	RV	D		
P4 load u							arb	RI	М	М	arb	RV	D

Figura 7.2 Concurrencia en el directorio. El acrónimo CP indica cola de peticiones pendientes.

Típicamente, en los diagramas temporales, para mostrar de forma simple las transacciones, supondremos que la latencia de cada fase es un ciclo. Esto nos permite suponer una latencia de iniciación igual a uno a partir de la cola de peticiones.

El CM, en función del protocolo, tiene una cola de respuestas de los CC⁷. El procesado de mensajes en esta cola es prioritario respecto de la cola CP.

Transacciones no atómicas

Cuando transacciones concurrentes referencian bloques distintos, no existe interacción entre ellas⁸. Sin embargo, si las transacciones referencian el mismo bloque, es posible que tanto el CM como los CC tengan que procesar una petición al bloque, de un CC o del CM respectivamente, cuando tienen pendiente de finalizar o completar una transacción al mismo bloque. En estas condiciones una transacción no es atómica.

^{6.} Los procesadores ejecutan las instrucciones en orden de programa y un procesador se bloquea en un acceso al CM. Entonces, el número de entradas necesarias en la CP es el número de procesadores. Si los CC utilizan un BEX hay que tener en cuenta el número de entradas en el BEX.

^{7.} Protocolos donde una cache puede tener el bloque en exclusividad o ser la encargada de suministrar el bloque aunque no tenga la exclusividad.

^{8.} La interacción puede ser por ocupación de recursos hardware.

Cruce de peticiones al mismo bloque. Entre el CM y un CC se produce un cruce de peticiones al mismo bloque cuando un CC, que está incluido en el VP, emite una petición al CM y a partir de este instante, el CM efectúa una o varias peticiones al CC, inducidas por una o varias peticiones de otros CC, antes de procesar la petición del CC.

En la Figura 7.3 se muestran dos ejemplos de cruce de peticiones. En la parte izquierda de la figura, protocolo VI, el CC1 debe procesar una petición del CM en un estado transitorio⁹. También, el CM debe procesar una petición no esperada (PtX), ya que el VP no contiene el identificador del CC1 que la ha emitido. Observemos que la petición que ha procesado previamente el CM, al mismo bloque (PtE), desactiva el bit del CC1 en el VP.

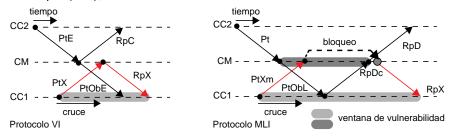


Figura 7.3 Ejemplos de cruce de peticiones.

En la parte derecha de la figura, protocolo MLI, el CC1 debe procesar una petición del CM en un estado transitorio y responder al CM para que éste finalice la transacción. Por otro lado, el CM puede necesitar procesar una petición de un CC en un estado transitorio. En el caso mostrado se pospone el procesado de la petición del CC1 en el CM hasta que el bloque en el CM esté en un estado estable¹⁰.

Gestión de un cruce en un CC. Entre la emisión de una petición por un CC y la recepción de la respuesta del CM puede ser necesario procesar alguna petición del CM al mismo bloque en el CC (ventana de vulnerabilidad).

La RV mantiene el orden de los mensajes emitidos desde el CM a los CC. Por ejemplo, el CM ordena en primer lugar una petición del CC2 respecto de una petición del CC1. La petición del CM al CC1, inducida por la petición del CC2, es recibida por el CC1 antes que la respuesta del CM a su petición. En particular, la petición y la respuesta pueden referenciar el mismo bloque. Esto es, la RV garantiza que un CC recibe las peticiones y respuestas del CM en el orden en que han sido emitidas desde el CM. Por tanto, un CC al recibir una petición

9. En el CC debe ser procesada, ya que el CM ha ordenado esta petición antes que la del CC1. 10. En el desarrollo del capítulo se pospone el procesado en el CM de la petición del CC que está en la cabeza de la CP y de las siguientes peticiones almacenas en la CP. El procesado se reanuda cuando finaliza la transacción en curso. En los ejercicios se analizan casos donde es factible que el CM siga procesando peticiones de la CP.

del CM, a un bloque en un estado transitorio, infiere que el CM ha procesado la petición de otro CC, al mismo bloque, antes que su petición y responde, si es el caso (Figura 7.4).

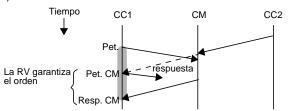


Figura 7.4 Cruce de peticiones.

En general, un CC pueden recibir varias peticiones mientras un bloque está en un estado transitorio. En consecuencia, usualmente, es necesario identificar el orden de recepción, para tenerlo en cuenta en las respuestas de las sucesivas peticiones (Figura 7.5).

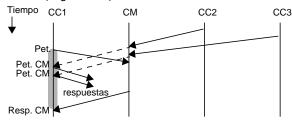


Figura 7.5 Cruce de varias peticiones.

En el diagrama de estados de un CC hay que identificar los casos en que el CM ha ordenado previamente peticiones de otros CC. Para ello se utilizan estados transitorios adicionales. Por otro lado, el CC debe responder teniendo en cuenta que el CM ha ordenado previamente la petición de otro CC, la cual ha inducido la petición que recibe del CM.

En la Figura 7.6 se muestra un ejemplo donde un CC puede recibir dos peticiones distintas en el estado transitorio INF. Las peticiones son P1 y P2. El CC puede recibir la petición P2 o puede recibir la secuencia P1, P2. En el ejemplo, si el CC recibe una petición P2, en el estado transitorio INF, cambia al estado INF1. Si recibe una petición P1 cambia al estado INF2 para identificar la secuencia. Una posibilidad menos ortodoxa, para reconocer las secuencias de peticiones, es la mostrada en la parte derecha de la figura¹¹. En ella se está suponiendo implícitamente que no se pueden recibir peticiones de tipo P2 antes de P1 y que después de P1 sólo puede recibirse una petición de tipo P2. Además, se supone que el estado final es el mismo.

11. Observemos que reconoce otras secuencias de peticiones del CM, además de las esperadas.



Figura 7.6 Estados transitorios en un CC para identificar peticiones del CM mientras un bloque está en un estado transitorio.

La respuesta del CC al CM en un estado transitorio tiene que estar en consonancia con el estado estable del que se proviene y con la secuencia de peticiones previas del CM que han sido respondidas¹².

Gestión de un cruce en el CM. Cuando el CM gestiona una acción de coherencia, puede emitir una petición a un CC que requiere una respuesta de este (por ejemplo, en el protocolo de directorio MLI). En el lapso de tiempo entre la petición del CM y la respuesta del CC, el CM analiza la cabeza de la CP. La petición en la cabeza de la CP puede referenciar el mismo bloque (ventana de vulnerabilidad). En este caso se bloquea el procesado de la petición y el análisis de las peticiones que hay encoladas en la CP (Figura 7.7).

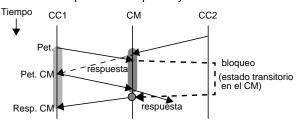


Figura 7.7 Bloqueo del procesado de peticiones de los CC en el CM cuando el bloque está en un estado transitorio.

En estas condiciones, en el CM sólo se procesan peticiones en estados estables. Un CC siempre responde a una petición del CM. Entonces, el CM reanuda el procesado de peticiones al recibir la respuesta del CC.

La petición de la cabeza de la CP, que ha bloqueado el procesado en el CM, es factible que al procesarse en un estado estable: a) sea un tipo de petición en la que se espera que el CC que la emite esté identificado en el VP y no sea así o b) sea necesario procesar la petición del CC en un estado que no es el esperado¹³. En el ejemplo de la parte derecha de la Figura 7.3, el CM debe procesar la petición PtXm en un estado estable, en la que no es esperada¹⁴.

^{12.} Recordemos que la RV mantiene el orden de los mensajes emitidos desde el CM a los CC.

^{13.} Esto es, en el estado actual del bloque en el CM no se espera este tipo de petición de un CC. Recordemos que un CC mantiene el bloque que expulsa hasta que no recibe la respuesta del CM.

PROTOCOLO DE DIRECTORIO VI

La descripción del protocolo de directorio VI se ha efectuado en el Capítulo 4. En cuanto al camino de datos es el descrito en el mismo capítulo.

En la Figura 7.8 se muestran por completitud (Figura 6.5) los mensajes en los dos tipos de transacciones del protocolo de coherencia.



Figura 7.8 Protocolo de directorio VI. Mensajes en los dos tipos de transacciones.

Organización del multiprocesador

En la Figura 7.9 se muestra la organización del multiprocesador. En ella se distingue la cola de peticiones de los CC al CM. También se observa el encaminamiento de los mensajes de los CC al CM y del CM a los CC.

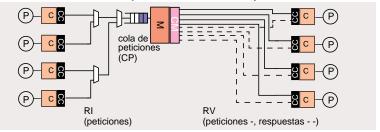


Figura 7.9 Protocolo de directorio VI. Organización del multiprocesador.

En la Figura 7.9, aunque la RV es única, se distinguen conexiones punto a punto para identificar que se transmiten peticiones y respuestas.

14. Estado L en lugar de estado M en el CM.

Transiciones entre estados estables de los controladores

En la Figura 7.10 se muestran por completitud los diagramas de transiciones entre estados de un bloque en el CM y en un CC (Figura 6.17 y Figura 6.18).

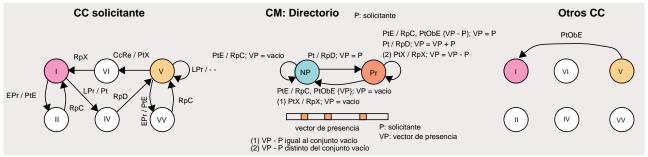


Figura 7.10 Protocolo VI. Transiciones entre estados en el CC y el CM.

Concurrencia en el controlador de memoria

El CM para procesar una petición sólo requiere acceder una vez al directorio. Con la información leída del directorio emite la respuesta y peticiones necesarias, si es el caso (Figura 7.10).

Peticiones concurrente a bloques distintos

En la Figura 7.11 se muestra la concurrencia en el procesado de peticiones en el CM. Las tres primeras peticiones se efectúan en el mismo ciclo y hay que arbitrar. El arbitraje determina el secuenciamiento en el CM.

		C	iclo	S									
referencia	1	2	3	4	5	6	7	8	9	10	11	12	13
P1 store t	arb	RI	М	arb	RV	С							
P2 load v	arb	arb	RI	М	arb	RV	D						
P3 store u	arb	arb	arb	RI	М	arb	RV	С					
P4 load r							arb	RI	СР	М	arb	RV	D

Figura 7.11 Protocolo VI. Concurrencia en el CM. Las peticiones referencian bloques distintos. El acrónimo CP indica cola de pendientes.

Identificación de posibles cruces de peticiones

Seguidamente analizamos en cada estado estable de un CC la emisión, por parte de este CC, de una petición al CM, compatible con el estado del bloque, y el procesado en el CM de peticiones que inducen peticiones al CC¹⁵. El estado del bloque en el CC determina un posible conjunto de estados del bloque en el directorio. Las peticiones de los otros CC deben ser compatibles con el estado del bloque en el directorio.

Las peticiones emitidas por el CM se reciben en el CC en la ventana de vulnerabilidad del CC (Figura 7.12).

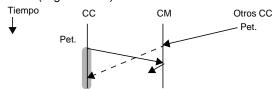


Figura 7.12 Protocolo VI. Cruce de peticiones de un CC y un CM.

Como la RV es ordenada, mantiene el orden de emisión de mensajes del CM a los CC. Entonces, un CC al recibir una petición en un estado transitorio debe procesarla, y si es el caso responder.

Un CC sólo tiene pendiente una petición a un mismo bloque. Recordemos que en una expulsión se espera una respuesta del CM. Entonces, mientras no se reciba la respuesta, un CC no emite otra petición al mismo bloque.

La RI utiliza un arbitraje por edad o antigüedad¹⁶ y la CP del CM se gestiona de forma FIFO.

En estas condiciones, la petición de un CC, denominado CC1, se cruza con una petición de otro CC (CC2) o con la petición de expulsión del bloque por parte de este CC (CC2), pero no con ambas peticiones¹⁷.

Una petición de expulsión sólo afecta al estado del bloque en el CM que gestiona el bloque. Entonces, una petición de expulsión puede determinar un cruce en el CC que la emite. También puede determinar un cruce en el CM, cuando la petición de otro CC es distinta de PtX.

Bloque en el estado V. Las peticiones de un CC1 al CM, cuando un bloque está en el estado V, son: PtE y PtX. El estado del bloque en el directorio puede ser NP o Pr (Figura 7.13).

^{15.} Para ello el CC debe estar identificado en el VP.

^{16.} El árbitro selecciona la petición más antigua.

^{17.} Un CC no tiene pendientes dos peticiones al mismo bloque. La petición del CC1 ha sido encolada en la CP antes que la segunda petición de otro CC (CC2).

El CM puede procesar las siguientes peticiones de otros CC antes de la petición del CC1: PtE, Pt. La petición PtX por parte de otro CC no se considera en el análisis, ya que no determina un cruce con la petición del CC1 en el CM. En el CC1 la petición PtX de otro CC no puede determinar un cruce, ya que el CM al procesarla no emite ninguna petición al CC1.

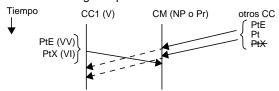


Figura 7.13 Bloque en estado V en la cache C1. Peticiones del CC1 y de otros CC.

Para que se produzca un cruce, la petición de un CC2, que procesa el CM, debe modificar el estado del bloque en la cache del CC1, cuya petición está esperando ser procesada en el CM (estado transitorio en el CC1).

Por tanto, todas las posibles secuencias, que se analicen, tienen que tener en primer lugar una instrucción que efectúe una escritura.

Una petición Pt sólo modifica el estado del bloque en el directorio. Recordemos que no se asigna contenedor en un fallo de escritura. Por tanto, es suficiente con analizar el caso en el cual el CC, que efectúa la primera petición de una secuencia de accesos a memoria, no tiene almacenado el bloque.

En la Figura 7.14 se muestra el orden de procesado en el CM de varias secuencia de accesos.

Orden de procesado en el CM							
Α	В						
P2 store t	P2 store t						
P1 PtX t	P1 store t						

Figura 7.14 Bloque en el estado V en la cache C1 y en el estado NP o Pr en el directorio. Orden de procesado en el CM de cada secuencia de accesos.

Estado I. Las peticiones de un CC1 al CM cuando un bloque está en el estado I son: PtE y Pt. El estado del bloque en el directorio puede ser NP o Pr (Figura 7.15).

El CM puede procesar las siguientes peticiones antes de la petición del CC1: PtE, Pt.

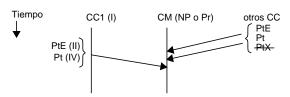


Figura 7.15 Bloque en estado I en la cache C1. Peticiones del CC1 y de otros CC.

El CC1 no recibe ninguna petición del CM, ya que no está identificado en el VP. Por tanto, en el CM no se infiere ningún cruce de peticiones.

Estados transitorios y transiciones en el controlador de memoria

Peticiones concurrentes al mismo bloque

En el diagrama de estados del CM no existen estados transitorios (Figura 7.10). La memoria siempre está actualizada y las peticiones del CM a los CC no requieren respuesta, ya que la RV es ordenada. Por tanto, en el CM no existen ventanas de vulnerabilidad.

Sin embargo, como pueden existir peticiones concurrentes de varios CC al mismo bloque: a) es posible que el CM tenga que procesar peticiones en las que se espera que el CC esté identificado en el VP del bloque, y no sea así o b) la petición no es esperada en el estado actual del bloque en el directorio. En estos casos, el CM infiere un cruce en un estado estable.

Identificación de cruces de peticiones

Estado V. La secuencia de accesos (B) donde el CC1 emite una petición PtE no determina la detección de un cruce de peticiones en el CM. En el protocolo VI, el CM procesa una petición PtE independientemente del estado del bloque en el directorio.

Dado el orden A de accesos a memoria, el CM al procesar la petición del CC2 emite una petición al CC1 y modifica el VP. El procesado de la petición PtX del CC1se puede efectuar en el CM con el bloque en el estado NP o Pr, en función de si el CC2 está en el VP. En cualquiera de los dos casos, al procesar el CM la petición del CC1, este CC no está incluido en el VP. Además, si la petición de CC2 se procesa en el estado NP no es esperada¹⁸.

18. La petición del CC1 puede procesarse en el estado Pr aunque la cache C2 no contenga el bloque. Es suficiente que entre las dos peticiones el CM procese una petición Pt de un tercer CC.

Resumen. En la Figura 7.16 se indica el estado en el cual se infiere el cruce y la condición de detección del mismo.

			Condició	า	
Secuencia	Estado	P∉ VP	$P \in VP$	no esperada	Diagrama de transiciones
Α	NP	Х		Х	CM: Directorio PtX _c / RpX
Α	Pr	X			PtX _c / RpX
					NP Pr

Figura 7.16 Protocolo VI. Condiciones de inferencia de cruces en el directorio. El subíndice C indica que el CC que emite la petición no está en el VP o que es una petición no esperada en el estado actual del bloque en el directorio.

En resumen, en una petición PtX de un CC, el VP del directorio debería contener al CC que ha emitido la petición. Si este no es el caso, el CM infiere un cruce.

Estados y transiciones en los controladores de coherencia

El directorio es preciso. Por tanto, las expulsiones no son silenciosas. En los estados transitorios II e IV de un bloque en cache no existe ventana de vulnerabilidad (Figura 7.10)¹⁹. En estos estados, el VP del bloque en el directorio no incluye al CC que solicita el bloque.

En la Figura 7.17 se observa que en el estado V un CC puede recibir una petición PtObE del CM. El estado V es el estado estable a partir del cual se puede efectuar una transición a los estados transitorios VI y VV.

Ventana de vulnerabilidad. En los estados transitorios VI y VV un CC puede recibir una petición de observación de escritura (PtObE) del CM.

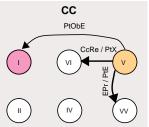


Figura 7.17 Protocolo de directorio VI. CC: peticiones que puede recibir un bloque estando en un estado transitorio.

19. Un CC sólo tiene pendiente una transacción a un bloque. Una segunda petición del CC al bloque se bloquea hasta que el CC recibe la respuesta de la transacción pendiente. Es posible que un CC tenga pendiente una expulsión de un bloque y una petición a un bloque distinto.

Para facilitar la comprensión en el siguiente desarrollo, en el margen izquierdo se replica la Figura 7.14.

Dado el orden A de accesos a memoria, el CC1, con el bloque en el estado VI, recibe una petición PtObE del CM que requiere invalidar el bloque. El CC1 no cambia el estado del bloque, ya que al recibir la respuesta del CM establece como estado del bloque I.

 Dado el orden B de accesos a memoria, cuando el CC1 procesa la petición PtObE, emitida por el CM al procesar la petición PtE del CC2, el bloque en la cache C1 está en el estado VV. El CC1 está esperando la notificación de la escritura por parte del CM para pasar el bloque a estado V. Sin embargo, el CM ha ordenado una petición PtE de otro CC antes que la petición del CC1. Por tanto, el bloque debe quedar en estado inválido al recibir la confirmación de la escritura. Para recordar la escritura, de otro CC, que ha sido ordenada previamente por el CM, se utiliza un estado transitorio denominado VVI. Entonces, el CC1 al procesar la petición PtObE del CM cambia el estado del bloque a VVI y posteriormente, cambia el estado del bloque a I, cuando recibe la respuesta del CM a su petición.

Seguidamente se presenta un análisis detallado de los cruces de peticiones.

Estado VV.

En el estado VV un CC puede recibir una petición PtObE, lo cual indica que el CM ha procesado previamente una petición PtE de otro CC.

En la Figura 7.18 se muestra un diagrama temporal y las transiciones entre estados. En la cache C1, la transición del estado estable del bloque a un estado transitorio está determinado por la emisión de una petición del CC1. Es suficiente que la emisión de la petición PtE se efectúe antes que la recepción de la petición PtObE del CM. El CC1 debe invalidar el bloque, ya que la escritura de otro procesador (PtE) ha sido ordenada por el CM antes que la petición de escritura del CC1. En consecuencia, en el CM se ha modificado el VP y el bit correspondiente al CC1 no está activado. Por tanto, la información almacenada en el campo de datos de la cache del CC1, cuya petición está pendiente, no es válida²⁰.

El CM al procesar una petición PtE no suministra el bloque, sólo actualiza la memoria. Por tanto, cuando el CC1 reciba la respuesta a su petición PtE debe establecer I como estado del bloque. En consecuencia hay que identificar el cruce en el CC1. Para ello, se utiliza un nuevo estado transitorio denominado VVI. El CC1, cuando recibe una petición PtObE del CM en el estado VV, cambia el estado del bloque a VVI y posteriormente, al recibir la respuesta del CM, a su petición PtE, establece I como estado del bloque.

20. Recordemos también que no se asigna contenedor en un fallo de escritura.

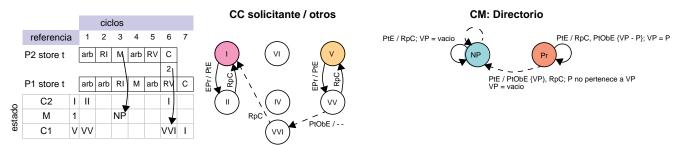


Figura 7.18 Cruce entre una petición PtObE del CM y una petición PtE de un CC. En trazo continuo las transiciones debidas al CC de C2. En trazo discontinuo las transiciones debidas al CC de C1.

En el ejemplo del diagrama temporal de la Figura 7.18, el CM no infiere un cruce, ya que el estado del bloque, al procesar la petición PtE del CC1, es NP. El CM tampoco infiere un cruce cuando procesa la petición PtE en el estado Pr y el CC1 no está en el VP. Este último caso se produce cuando en la cache C2 el bloque está en estado V al emitir la petición PtE (se muestra en el diagrama de transiciones entre estados de la Figura 7.18). Recordemos que un CC puede emitir una petición PtE a un bloque en el estado I.

El funcionamiento de la RV garantiza que el CC1 recibe primero la petición del CM y después la respuesta del CM a su petición. De forma similar, el CC2 recibe primero la respuesta a su petición y después la petición del CM, si es el caso, inducida al procesar la petición del CC1 (bloque en estado V en C2).

Estado VI.

En el estado VI el CC puede recibir una petición PtObE, lo cual indica que el CM ha procesado previamente una petición PtE de otro CC²¹.

En la Figura 7.19 se muestra un diagrama temporal y las transiciones entre estados. En la cache C1, la transición del estado estable del bloque a un estado transitorio está determinado por la emisión de una petición del CC1. Es suficiente que la emisión se efectúe antes que la recepción de la petición del CM²² (PtObE). El CM, al procesar la petición del CC2, ha eliminado al CC1 del VP del bloque en el directorio. Entonces, el CM procesa la petición PtX en el estado NP o en el estado Pr.

^{21.} A este cruce se le denomina usualmente petición tardía. La petición del CM llega después de iniciar la expulsión.

^{22.} Notemos que temporalmente el procesado de la petición del CC1 en el CM puede ser bastante después si hay entrelazadas peticiones a otros bloques. También pueden ser peticiones de lectura de otros CC al mismo bloque. Cualquiera de estas peticiones al ser procesada por el CM no induce una petición al CC1.

En el diagrama de la Figura 7.19 se muestra el caso de que el estado sea NP, que se corresponde con el estado I del bloque en la cache C2²³.

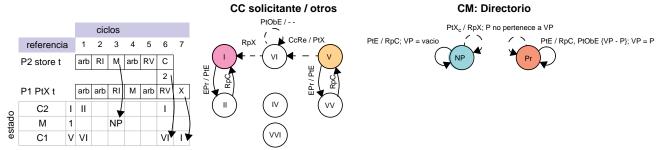


Figura 7.19 Cruce entre una petición PtObE del CM y una petición PtX de un CC. En trazo continuo las transiciones debidas al CC de C2. En trazo discontinuo las transiciones debidas al CC de C1.

El CM infiere un cruce en los dos estados, ya que al procesar la petición PtX espera que el CC esté identificado en el VP y no es así. El CM responde a la petición PtX y no modifica el estado del bloque en el directorio. El CC al procesar la petición PtObE en el estado VI tampoco cambia de estado.

Resumen. En la Figura 7.20 se muestran los estados en los cuales se infiere un cruce y la respuesta del CC.

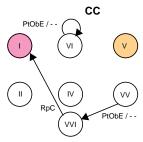


Figura 7.20 Protocolo VI. Inferencia de cruces en un CC.

Diagramas completos de cruce de peticiones

En la Figura 7.21 se muestran las transiciones en un CC y en el CM cuando se produce un cruce de peticiones. En el CM se infiere un cruce cuando el CC no está en el vector de presencia y se recibe una petición PtX en cualquiera de los dos estados.

23. Si el estado en la cache C2 fuera V, la petición PtX del CC1 se procesaría en el CM en el estado Pr.

Al procesar una petición, la primera acción en el CM es inferir si se ha producido un cruce de peticiones con un CC. Para ello el CM utiliza el VP. En el diagrama de estados de un bloque en el directorio, se utiliza el subíndice c (cruce) para identificar una petición de un CC que se ha cruzado con una petición del CM.

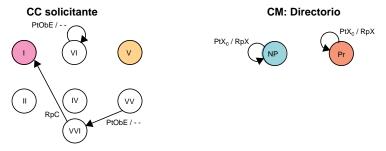


Figura 7.21 Diagramas de transiciones en un CC y en el directorio cuando se produce un cruce de peticiones a un bloque entre un CC y el CM.

Notemos que al recibir un CC una petición PtObE del CM, a un bloque en estado VV, el CC infiere que el CM ha extraído al CC del VP. Es como si el CC efectuara la petición PtE desde el estado I. Por ello, al recibir la respuesta a la petición PtE el estado del bloque debe ser I.

De forma similar, un CC al recibir una petición PtObE del CM, a un bloque en estado VI, infiere que el CM ha extraído al CC del VP y espera la respuesta a su petición PtX.

Tablas de estados y transiciones

En la Tabla 7.1 se muestra la descripción de los estados, estables y transitorios, eventos y transiciones entre estados en un CC en formato tabla. 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.

			Evento	s del proces reemplazo	ador y	Eventos	Eventos externos (respuestas y petición)						
			LPr	EPr	CcRe	RpD	RpC	RpX	PtObE				
	ples	- 1	Pt; IV	PtE; II									
	Estables	V	; V	PtE; VV	PtX; VI				;I				
SC		IV				; V							
Estados	rios	VV					; V		; VVI				
ш	transitorios	VI						; I	; VI				
	trar	II					; l						
		VVI					; I						

Tabla 7.1 Peticiones concurrentes. Protocolo de directorio VI. Tabla de estados y transiciones de un bloque en cache. Las casillas con fondo blanco indican cruces de peticiones.

En la Tabla 7.2 se muestra en formato tabla los estados y transiciones entre estados de un bloque en el directorio. En el evento PtE se distinguen dos casos en función de si el procesador, cuyo CC efectúa la petición, está o no está en el vector de presencia. En el evento PtX 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										
				PtE	PtX								
			Pt	VP = vacio	P∈VP	P∉ VP	VP = P	VP≠P P∈ VP	P∉ VP				
Estados	ples	NP	RpD; Pr, VP = P	RpC; NP, VP = vacío					RpX; NP, VP = vacío				
Esta	Estables	Pr	RpD; Pr, VP = VP + P		$\label{eq:rpc} \begin{aligned} RpC, PtObE \{VP\text{-}P\}; Pr,\\ VP&=P \end{aligned}$	RpC, PtObE {VP}; NP, VP = vacío	RpX; NP, VP = vacío	RpX; Pr, VP = VP - P	RpX; Pr, VP = VP				

Tabla 7.2 Peticiones concurrentes. Protocolo de directorio VI. Tabla de estados y transiciones de un bloque en el directorio. Las casillas con fondo blanco indican cruces de peticiones.

Representación de transacciones y transiciones entre estados

En este apartado se muestran tres formas de representar una secuencia de accesos a memoria: a) en formato tabla y b) mediante un diagrama temporal y c) mediante un diagrama temporal simplificado.

Representación en formato tabla

Los accesos a memoria se muestran en grupos separados mediante líneas horizontales continuas. Cada grupo de accesos a memoria se gestiona independientemente y no se empieza a gestionar el siguiente grupo hasta que ha finalizado el anterior. Los accesos a memoria de un grupo se realizan concurrentemente en el mismo ciclo. En una fila de la tabla, de izquierda a derecha, después de la instrucción de acceso a memoria se especifica:

- 1 La primera columna se utiliza para representar el arbitraje de peticiones. Esto es, se indica el orden de las transacciones, si es el caso. Supondremos usualmente que en un grupo de peticiones el árbitro las selecciona en el orden en que se especifica la secuencia de accesos a memoria.
- 2 El siguiente grupo de columnas se utiliza para representar el estado transitorio del bloque, en la cache correspondiente, cuando se emite la petición.
- 3 El tercer grupo de columnas se utiliza para representar la petición del CC en la red de ida (RI). La petición se indica en la casilla correspondiente y en el orden en el que el árbitro concede el acceso al módulo de memoria.
- 4 El cuarto grupo de columnas se utiliza para identificar la variable o bloque y el VP en el directorio. Los bits en el VP identifican a las caches; de izquierda a derecha en ordinal creciente. Un valor de uno en el VP indica que hay una copia del bloque en la cache correspondiente.
- **5** La siguiente columna se utiliza para indicar qué elemento (memoria o cache) suministra el bloque o dato.
- 6 La siguiente columna representa el arbitraje en la red de vuelta (RV). Como las peticiones han ocupado la RI en ciclos distintos, también ocupan la RV en ciclos distintos. En esta columna se indica el ordinal de los procesadores que reciben la respuesta o peticiones (destinatario). Para la respuesta se utiliza una fila y para las peticiones la siguiente fila y la misma columna.
- **7** El séptimo grupo de columnas se refiere a la RV. La representación es idéntica a la RI exceptuando que los mensajes que se indican son de respuesta y petición. Cuando haya que indicar una petición, se utiliza la siguiente fila y además se indica el ordinal de los controladores de coherencia que reciben la petición.
- 8 El octavo grupo de columnas se utiliza para identificar la variable accedida en cache y el estado del bloque al finalizar la acción de coherencia. Cuando un bloque está en una ventana de vulnerabilidad y recibe una petición del CM se indica el estado transitorio del bloque. En estas

condiciones, el estado transitorio, debido a que se ha atendido una petición del CM, se observa en las columnas de la derecha. El estado se indica en las filas asociadas a las peticiones del CM.

9 Si no se accede al CM la fila correspondiente a la petición se deja en blanco.

Ejemplo. En la Tabla 7.3 se muestra una secuencia de accesos a memoria realizada por tres procesadores. Los accesos a memoria están agrupados de dos en dos. Cuando se inicia la secuencia de accesos a memoria, las caches no almacenan los bloques en los que se ubican las variables accedidas. Tampoco se producen conflictos de contenedor al almacenar los bloques en los contenedores de cache.

Los primeros dos accesos concurrentes requieren iniciar una transacción cada uno. La primera transacción es debida a un fallo de lectura y la segunda es debida a una escritura y los bloques referenciados son distintos. Los mensajes de petición emitidos por CC1 y CC2 son Pt y PtE respectivamente. El mensaje de petición de CC1 es elegido por el árbitro en primer lugar. El estado transitorio del bloque en la cache C1 es IV y en la cache C2 es II. El estado final del bloque en la cache C1 es V y en la cache C2 es I. En el directorio se indica que el CC1 tiene copia del bloque que contiene la variable t.

El segundo par de accesos a memoria son fallos de lectura al mismo bloque. El estado transitorio en la cache C1 y en la cache C3 es IV, ya que la transacción, que inicia cada uno de ellos, es debida a un fallo de lectura. El mensaje emitido por CC1 y por CC3 es Pt. La respuesta del CM a cada uno de ellos es RpD. Después de finalizar las dos transacciones, el VP del directorio identifica que CC1 y CC3 tienen copia del bloque que contiene la variable u.

El tercer par de accesos a memoria requiere una transacción para servir un fallo de lectura (CC1) y una transacción para servir una escritura (CC2). La primera transacción es mimética a los fallos de lectura previos. La segunda transacción requiere que el CM emita mensajes de observación de escritura (PtObE) a CC1 y CC3 que tienen copia del bloque. Al finalizar la segunda transacción el VP del bloque, que contiene la variable u, en el directorio indica que no hay copia del bloque en ninguna cache.

En el siguiente par de accesos a memoria tenemos que el primero de ellos es un acierto en cache. El segundo acceso requiere una transacción para servir un fallo de lectura. El VP del bloque, que contiene la variable w, en el directorio indica que hay copia del bloque en las caches C1 y C3.

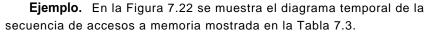
	arb.	C 1	C 2	C 3	F	RI		mem.		arb.	R	V	С	1	С	2	С	3
acceso	ord.	est.	est.	est.	1°	2°	var.	VP	sum.	dest.	1°	2°	var.	est.	var.	est.	var.	est.
1. P1 load t	1º	IV			Pt		t	1, 0, 0	mem.	CC1	RpD		t	V				
1. P2 store u	20		П			PtE	u	0, 0, 0	C2	CC2		RpC			u	ı		
2. P1 load u	1º	IV			Pt		u	1, 0, 0	mem.	CC1	RpD		u	V				
2. P3 load u	20			IV		Pt	u	1, 0, 1	mem.	ССЗ		RpD					u	V
3. P1 load w	10	IV			Pt		w	1, 0, 0	mem	CC1	RpD		w	V				
3. P2 store u	20		П			PtE	u	0, 0, 0	C 2	CC2		RpC			u	ı		
										CC1,3		PtObE 1,3	u	ı			u	I
4. P1 load w																		
4. P3 load w	1º			IV	Pt		w	1, 0 ,1	mem	ССЗ	RpD						w	V
5. P2 store w	1º		П		PtE		w	0, 0, 0	C 2	CC2	RpC							
										CC1,3	PtObE 1,3		w	VVI			w	ı
5. P1 store w	2º	V V				PtE	w	0, 0, 0	C1	CC1		RpC	w	ı				

Tabla 7.3 Peticiones concurrentes. Protocolo de directorio VI. Secuencia de accesos a memoria concurrentes.

El último par de transacciones son debidas a dos escrituras a la misma variable (PtE). La cache C2 no almacena el bloque que contiene la variable y la cache C1 almacena el bloque. El estado transitorio del bloque en la cache C2 es II y en la cache C1 es VV. El CM al procesar la petición PtE de CC2 emite mensajes de observación de escritura (PtObE) a CC1 y CC3. Las caches correspondientes tienen copia del bloque. Al recibir el mensaje PtObE, CC3 invalida el bloque y CC1 establece el estado transitorio VVI. Al finalizar la primera transacción el VP del bloque en el directorio indica que no hay copia del bloque en las caches. El CC1 al recibir la respuesta del CM invalida el bloque.

Diagrama temporal

La fase arb se utiliza para mostrar la ordenación de los mensajes. La espera para acceder a una red se indica representando en ciclos consecutivos la fase de arbitraje (arb). El resto se representa de la forma descrita en el Capítulo 5.



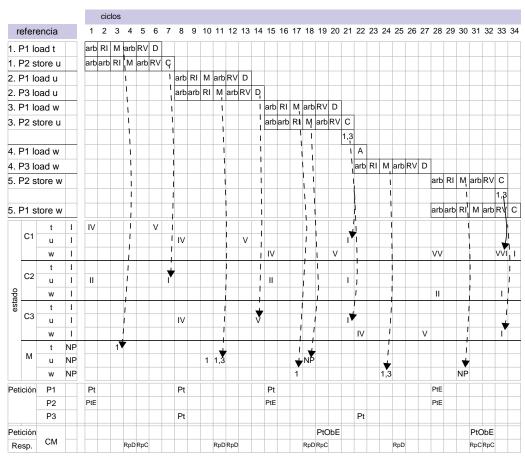


Figura 7.22 Peticiones concurrentes. Protocolo de directorio VI. Diagrama temporal de una secuencia de accesos a memoria concurrentes.

Diagrama temporal simplificado

En el diagrama temporal simplificado de la Figura 7.23 se utiliza la secuencia de accesos a memoria mostrada en la Tabla 7.3. Las líneas horizontales a trazos separan grupos de peticiones que se inician concurrentemente. El siguiente grupo de peticiones no se inicia hasta que ha finalizado el grupo previo. El orden de arbitraje se observa en el orden de llegada de las peticiones concurrentes al CM. Un círculo negro indica que el acceso a memoria no genera una transacción.

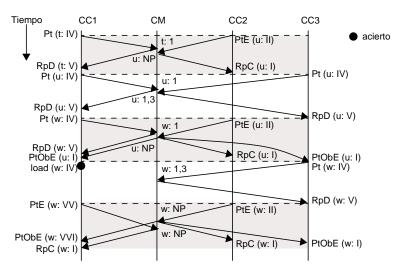


Figura 7.23 Peticiones concurrentes. Protocolo de directorio VI. Diagrama temporal simplificado de una secuencia de accesos a memoria concurrentes.

PROTOCOLO DE DIRECTORIO MLI

La descripción del protocolo de directorio MLI se ha efectuado en el Capítulo 6. En cuanto al camino de datos es el descrito en el mismo capítulo.

En la Figura 7.24 por completitud se muestran los mensajes en los tres tipos de transacciones del protocolo de coherencia (Figura 6.23).

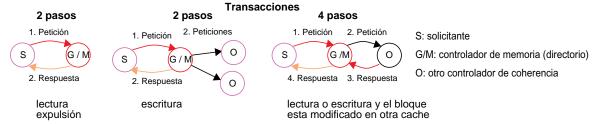


Figura 7.24 Protocolo MLI. Escritura retardada. Mensajes en los tres tipos de transacciones.

Organización del multiprocesador

En la Figura 7.25 se muestra la organización del multiprocesador. En ella se distingue la cola de peticiones y la cola de respuestas de los CC al CM, juntos con las redes asociadas (RI y RCM). También se observa el encaminamiento de los mensajes del CM a los CC.

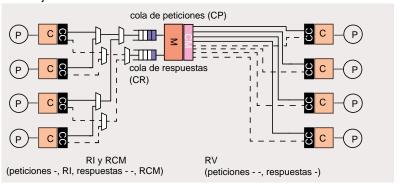


Figura 7.25 Protocolo de directorio MLI. Organización del multiprocesador. Caches privadas con escritura retardada.

En la Figura 7.25 se distinguen conexiones punto a punto para identificar peticiones y respuestas del CM a los CC, aunque la RV es única.

Al procesar el CM una petición, que requiere la colaboración de varios CC, los destinos del conjunto de peticiones PtObE y la respuesta (RpD) son disjuntos. Por tanto, puede utilizarse la misma red para peticiones y respuestas del CM (2 pasos en la Figura 7.24). Por tanto, las redes RV y RMC se pueden implementar mediante una única red, a la que denominamos RV. En estas condiciones, en una transacción de dos pasos sólo indicaremos la red RV.

Al procesar el CM una petición, que requiere que un CC suministre el bloque, sólo hay una petición a este CC. Al recibir la respuesta del CC, que participa en la acción de coherencia, el CM emite una respuesta al CC solicitante (4 pasos en la Figura 7.24). En este caso, para la petición del CM indicaremos la red RMC y para la respuesta del CM indicaremos la red RV, aunque la red RMC y la red RV sean la misma.

Transiciones entre estados estables de los controladores

En la Figura 7.26 se muestra, por completitud, el diagrama de transiciones entre estados de un bloque en el directorio y en un CC (Figura 6.40 y Figura 6.41).

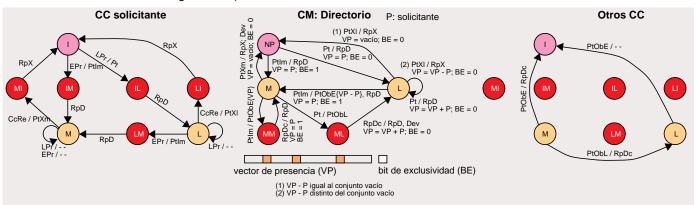


Figura 7.26 Transiciones entre estados en el CC y en el CM.

Concurrencia en el controlador de memoria

Peticiones concurrente a bloques distintos

El procesado de una petición en el CM puede necesitar solicitar el bloque a una cache y esperar la respuesta (petición pendiente). Durante este periodo de tiempo el estado del bloque es transitorio (MM y ML, Figura 7.26).

Mientras el CM está esperando la respuesta de un CC, puede extraer otra petición de la CP y procesarla durante este lapso de tiempo. Si el bloque de memoria, al que accede esta segunda petición, es distinto de los bloques que están en un estado transitorio, el CM puede iniciar el procesado de la petición y por tanto existe concurrencia.

En la Figura 7.27 se muestra la concurrencia en el procesado de peticiones en el CM. Las dos primera peticiones se efectúan en el mismo ciclo y hay que arbitrar. El procesado de la primera petición necesita que el directorio solicite el bloque a otro CC. En la parte inferior de la Figura 7.27 se indican los cuatros mensajes correspondientes a la primera petición.

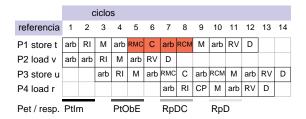


Figura 7.27 Protocolo MLI. Concurrencia en el CM. Las peticiones referencian bloques distintos. El acrónimo CP indica cola de pendientes.

La segunda y tercera petición se pueden procesar mientras se espera la respuesta del CC. La cuarta petición se efectúa dos ciclos antes de que el CM reanude el procesado de la primera petición, al recibir la respuesta de la cache²⁴. Entonces, la cuarta petición está almacena en la CP durante un ciclo, para eliminar el riesgo estructural en la fase M entre ella y la primera petición.

Identificación de posibles cruces de peticiones

Seguidamente analizamos en cada estado estable de un CC la emisión, por parte de este CC, de una petición al CM, compatible con el estado del bloque, y el procesado en el CM de peticiones que inducen peticiones al CC²⁵. El estado del bloque en un CC determina un posible conjunto de estados del bloque en el CM. Las peticiones de los otros CC deben ser compatibles con el estado del bloque en el directorio.

Las peticiones emitidas por el CM se reciben en el CC durante la ventana de vulnerabilidad (Figura 7.28).

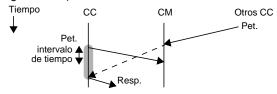


Figura 7.28 Cruce de peticiones de un CC y un CM.

Como la RV es ordenada, mantiene el orden de emisión de mensajes del CM a los CC. Entonces, un CC al recibir una petición en un estado transitorio debe procesarla, y si es el caso responder.

- 24. En el directorio el procesado de las respuestas es prioritario frente a las peticiones.
- 25. Para ello el CC debe estar identificado en el VP.

Un CC sólo tiene pendiente una petición a un mismo bloque. Recordemos que en una expulsión se espera una respuesta del CM. Entonces, mientras no se reciba la respuesta, un CC no emite otra petición al mismo bloque.

La RI utiliza un arbitraje por edad o antigüedad²⁶ y la CP del CM se gestiona de forma FIFO.

En estas condiciones, la petición de un CC, denominado CC1, se cruza con la petición de otro CC (CC2) o con la petición de expulsión del bloque por parte de este CC (CC2), pero no con ambas peticiones.

Una petición de expulsión sólo afecta al estado del bloque en el CM que gestiona el bloque. Entonces, una petición de expulsión puede determinar un cruce en el CC que la emite. También puede determinar un cruce en el CM, cuando la petición de otro CC es distinta de PtXI o PtXm.

Bloque en el estado L. Cuando un bloque está en el estado L en la cache, las peticiones de un CC1 al CM son: PtIm y PtXI. El estado del bloque en el directorio es L (Figura 7.29).

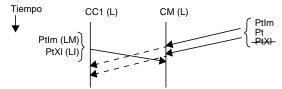


Figura 7.29 Bloque en estado L en la cache C1. Peticiones del CC1 y de otros CC.

El CM puede haber procesado (ordenado), antes de la petición del CC1, las siguientes peticiones de otros CC: PtIm y/o Pt. La petición PtXI por parte de otro CC no se considera en el análisis, ya que no determina un cruce con la petición del CC1 en el CM. En el CC1, la petición PtX de otro CC no puede determinar un cruce, ya que el CM no emite ninguna petición al CC1 al procesarla.

Para que se produzca un cruce, alguna de las peticiones que el CM procesa, antes de la petición del CC1, debe modificar el estado del bloque en la cache C1. Por tanto, como el estado del bloque en el CM es L, todas las posibles secuencia de accesos a memoria, que se analicen, tienen que tener como primer acceso a memoria una instrucción que genera una petición de exclusividad al CM (PtIm). En la Figura 7.30 se muestra el orden de procesado en el CM de varias secuencia de accesos.

Orden de procesado en el CM										
A	В	С	D							
P2 store t	P2 store t	P2 store t	P2 store t							
P1 PtXI t	P3 load t	P1 store t	P3 load t							
	P1 PtXI t		P1 store t							

Figura 7.30 Bloque en estado L en la cache C1 y en el directorio. Orden de procesado en el CM de cada secuencia de accesos.

Bloque en el estado M. Cuando un bloque está en el estado M en la cache, la petición de un CC1 al CM es PtXm. El estado del bloque en el directorio es M (Figura 7.31).

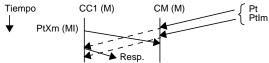


Figura 7.31 Bloque en estado M en la cache C1. Peticiones del CC1 y de otros CC.

El CM puede haber procesado (ordenado), antes de la petición del CC1, las siguientes peticiones: Ptlm, Pt.

Para que se produzca un cruce de peticiones, alguna de las peticiones que el CM procesa, antes de la petición del CC1, debe modificar el estado del bloque en la cache C1. Cualquiera de las dos peticiones enumeradas (PtIm, Pt) modifica el estado del bloque en la cache C1. En la Figura 7.32 se muestra el orden de procesado en el CM de varias secuencia de accesos.

Orden de procesado en el CM									
E	F	G	Н						
P2 store t	P2 load t	P2 store t	P2 load t						
P1 PtXm t	P1 PtXm t	P3 load t	P3 store t						
		P1 PtXm t	P1 PtXm t						

Figura 7.32 Bloque en estado M en la cache C1 y en el directorio. Orden de procesado en el CM de cada secuencia de accesos.

Bloque en el estado I. No se reciben peticiones, ya que el CC1 no está identificado en el VP.

Estados transitorios y transiciones en el controlador de memoria

Peticiones concurrentes al mismo bloque

Ventana de vulnerabilidad. Mientras el CM está esperando una respuesta puede analizar una petición al mismo bloque (estados transitorios MM y ML).

El directorio es el punto de ordenación de las peticiones al mismo bloque. En otras palabras es el encargado de serializar las peticiones al mismo bloque. Para efectuar la serialización, una alternativa es no iniciar en el CM el procesado de una petición hasta que ha finalizado el procesado de la petición previa al mismo bloque. Esto es, cuando el bloque está en un estado estable en el CM.

Un mecanismo en el CM, para gestionar que no se procesa una petición, es bloquear el análisis de la CP hasta que el CM recibe la respuesta relacionada con el bloque que determina el bloqueo²⁷ (Figura 7.33).

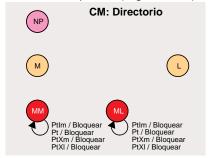


Figura 7.33 CM: bloqueo del procesado de peticiones en los estados transitorios del directorio.

En la Figura 7.34 se muestra la gestión de dos peticiones que referencian el mismo bloque. La primera petición (P1 store t) requiere una petición del CM a otro CC para mantener la coherencia. Para representar el bloqueo del análisis de peticiones utilizaremos el acrónimo B. El bloqueo se propaga a la peticiones más jóvenes, independientemente del bloque al que acceden. Estas peticiones más jóvenes se almacenan o están almacenadas en la CP. La petición que queda bloqueada se procesa en el CM después de que el CM emita la respuesta de la petición que ha determinado el bloqueo.

^{27.} El directorio sigue procesando respuestas a medida que llegan. En consecuencia, el bloqueo del procesado de la CP desaparece al procesar la respuesta que está relacionada con el bloqueo.

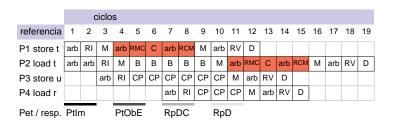


Figura 7.34 Bloqueo del procesado de peticiones en el CM. Dos peticiones referencian el mismo bloque y para servir la 1ª petición es necesario efectuar una petición a otro CC.

El CM no procesa peticiones de los CC a un bloque en un estado transitorio. Entonces, el resto de cruces entre peticiones del CM y un CC se observan (infieren) en estados estables.

Identificación de cruces de peticiones

Para facilitar la comprensión en el siguiente desarrollo, en el margen izquierdo se replica la Figura 7.30.

Bloque en el estado L. Los ordenes de acceso a memoria C y D, donde el CC1 emite una petición PtIm, no determinan la detección de un cruce en el CM. Recordemos que cuando el CM analiza una petición de un CC, a un bloque en un estado transitorio en el directorio, el procesado de las peticiones en la CP se bloquea (secuencia D). El procesado se reanuda cuando el bloque está en un estado estable. En el protocolo MLI, el CM procesa una petición de exclusividad independientemente del estado estable (Figura 7.26). Entonces, sólo tendremos en cuenta los ordenes A y B para analizar cruces de peticiones en el directorio.

Dado el orden de accesos a memoria A, el CM, al procesar la petición del CC2, establece el estado M en el directorio. Entonces, al procesar el CM la petición del CC1, el estado del bloque no es el esperado. La petición PtXI de CC1 se espera procesar en el estado L. Por otro lado, el VP tiene activado el identificador de un CC distinto al identificador de CC1 y el BE está activado. El CM responde a la petición del CC1 (RpX).

El CM, al procesar el orden B de accesos a memoria, establece la secuencia de estados estables M y L, al procesar respectivamente las peticiones del CC2 y del CC3. La petición del CC1 se procesa en el estado L. Sin embargo, el VP no tiene activado el identificador del CC1. El CM responde a la petición del CC1 (RpX).



Orden de procesado							
C	D						
P2 store t	P2 store t						
P1 store t	P3 load t						
	P1 store t						

Para facilitar la comprensión en el siguiente desarrollo, en el margen izquierdo se replica la Figura 7.32.

P2 store t P2 load t P1 PtXm t P1 PtXm t

P3 load t P3 load t P2 store t P1 PtXm t P1 PtXm t

P2 store t

Bloque en el estado M. Dado el orden de accesos a memoria E, el CM establece el estado M en el directorio al procesar la petición del CC2. Cuando el CM procesa la petición del CC1 el contenido de VP no identifica al CC1. El CM responde al CC1 y no se actualiza memoria.

Dado el orden de accesos a memoria F, el CM establece el estado L en el directorio, al procesar la petición del CC2. Entonces, cuando el CM procesa la petición del CC1 el estado no es el esperado. Sin embargo, el VP tiene activado el identificador del CC1. El CM responde al CC1, desactiva el identificador del CC1 de VP y no actualiza la memoria.

Dado el orden G de accesos a memoria, el CM establece los estados M y L en el directorio al procesar respectivamente las peticiones del CC2 y del CC3. Cuando el CM procesa la petición del CC1 el estado del bloque es L y la petición PtXm no es esperada. Además, el VP no tiene activado el identificador del CC1. El CM responde y no actualiza memoria.

Dado el orden H de accesos a memoria, la petición del CC1 se procesa en el estado M y el VP no tiene activado el identificador del CC1. El CM responde y no actualiza memoria. La funcionalidad del CM en el orden H de accesos a memoria está cubierta por el orden E.

Directorio: bloque en el estado NP. Seguidamente razonamos que, cuando el estado del bloque es NP, no se producen cruces en el directorio. El CM procesa las peticiones almacenadas en la CP de forma FIFO.

Sean dos controladores CC1 y CC2 (Figura 7.35). Una petición de CC1 induce una petición del CM al CC2, que se cruza con una expulsión de CC2 (PtXm, PtXI). La petición del CC2 debe haberse emitido antes de que la petición del CM, inducida por la petición de CC1, se haya procesado en el CC2. La transacción del CC1, que ha determinado el cruce, finaliza como muy pronto a la par que llega la petición del CM al CC2. Cualquier transacción que expulse el bloque en el CC1 se encolará en la CP después de la petición PtXm o PtXI de CC2²⁸. Entonces, en el estado NP no se detectan cruces.

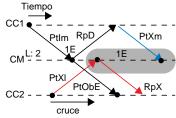


Figura 7.35 Protocolo LMI. Cruce de peticiones.

Resumen. En la Figura 7.36 se indica el estado en el cual se infiere el cruce y la condición de detección del mismo.

			Condición			
	Secuencia	Estado	P∉ VP	$P \in VP$	no esperada	Diagrama de transiciones
	Α	М	X		X	CM: Directorio
	В	М	X			PtXI _c / RpX PtXm _c / RpX PtXm _c / RpX; VP = VP - P
	E	М	X			
	F	L		Х	X	PtXI _c / RpX PtXm _c / RpX
	G	L	X		X	
	Н	L	X			ML

Figura 7.36 Protocolo MLI. Condiciones de inferencia de cruces en el directorio. El subíndice C indica que el CC que emite la petición no está en el VP o que es una petición no esperada en el estado del bloque en el CM.

En resumen, en peticiones PtXm y PtXl de un CC, el VP del directorio debería contener al CC que ha emitido la petición. Si este no es el caso, el CM infiere un cruce. En el CM, también se infiere un cruce cuando una petición PtXm de un CC se procesa en el estado L y el VP tiene identificado al CC.

Estados transitorios y transiciones en los controladores de coherencia

El directorio es preciso. Por tanto, las expulsiones no son silenciosas. En los estados transitorios IL e IM de un bloque en cache no existe ventana de vulnerabilidad, ya que el vector de presencia del bloque en el directorio indica que no existe copia del bloque en las caches de los procesadores (Figura 7.26).

En los estados transitorios LM, MI y LI existe una ventana de vulnerabilidad. Durante la espera de la respuesta a una petición, en uno de estos estados transitorios, se puede recibir del CM una petición de observación de escritura (PtObE) y en el caso particular del estado transitorio MI, también se puede recibir una petición de observación de lectura (PtObL). En la Figura 7.37 se observa que es factible recibir estas peticiones en los estados M y L, que son los estados estables desde los que se parte para llegar a los estados transitorios. Para que se produzca un cruce de peticiones, el CM ha procesado y

^{28.} El árbitro que determina el orden, en el cual se encolan las peticiones en la CP, concede el permiso por antigüedad. Si este no es el caso, en el CM hay que procesar cruces de peticiones, siendo el estado del bloque NP.

ordenado una o algunas peticiones de otros CC, antes que la petición del CC que recibe la petición del CM. En particular, una de las peticiones ha inducido que el CM emita una petición al CC.

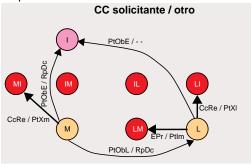


Figura 7.37 Protocolo de directorio MLI. CC: peticiones que puede recibir un bloque estando en un estado transitorio.

Identificación de cruces de peticiones

Para facilitar la comprensión en el siguiente desarrollo, en el margen izquierdo se replica la Figura 7.30.

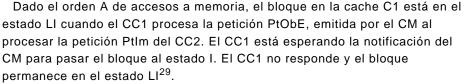
Dado el orden A de accesos a memoria, el bloque en la cache C1 está en el estado LI cuando el CC1 procesa la petición PtObE, emitida por el CM al procesar la petición PtIm del CC2. El CC1 está esperando la notificación del CM para pasar el bloque al estado I. El CC1 no responde y el bloque

Dado el orden B de accesos a memoria el comportamiento en el CC1 es el mismo que en el orden A. El CM no emite ninguna petición al CC1 al procesar la petición del CC3.

Dado el orden C de accesos a memoria, el CC1 procesa la petición PtObE del CM, cuando el bloque está en el estado LM. Esta petición es para invalidar el bloque. Como la respuesta del CM a la petición del CC1 (Ptlm) incluye el bloque, no hay cambio de estado del bloque en el CC1.

El comportamiento del CC1 en el orden D de accesos a memoria es el mismo que en el orden C. El CM al procesar la petición del CC3 no emite ningún mensaje al CC1. El bloque lo tiene en exclusividad el CC2.

Para facilitar la comprensión en el siguiente desarrollo, en el margen izquierdo se replica la Figura 7.32.



P2 store t P2 store t P1 store t P3 load t P1 store t

Orden de procesado en

P2 store t

P3 load t P1 PtXI t

P2 store t

P1 PtXI t

29. En el apartado de ejemplos se muestran los diagramas temporales para las distintas secuencias de acceso a memoria.



Orden de ¡	
G	Н
P2 store t	P3 load t
P3 load t	P2 store t
P1 PtXm t	P1 PtXm t

Dado el orden E de accesos a memoria, el CC1 procesa la petición PtObE del CM con el bloque en el estado MI. Esta petición es para suministrar e invalidar el bloque. El CC1 suministra el bloque y permanece en el estado MI, a la espera de la respuesta del CM a su petición.

Para identificar una posible secuencia de peticiones del CM se añade un nuevo estado transitorio denominado MII (Figura 7.38). Dado el orden F de accesos a memoria, el CC1 procesa la petición PtObL del CM con el bloque en el estado MI. Esta petición es para suministrar el bloque. El CC1 suministra el bloque y cambia a un nuevo estado transitorio, denominado MII, y espera la respuesta del CM a su petición. Este estado es para recordar que el CM ha ordenado una petición Pt antes que la petición PtXm del CC1.

Dado el orden G de accesos a memoria, el comportamiento del CC1 es el mismo que en el orden E. El CM al procesar la petición del CC3 no emite ningún mensaje al CC1. El bloque lo tiene en exclusividad el CC2.

Dado el orden H de accesos a memoria, el CC1 se comporta como en el orden F al procesar la petición PtObL del CM. Posteriormente, el CC1 recibe una petición PtObE. En el estado MII esta petición tiene como objetivo invalidar el bloque. Notemos que el CC1, al procesar la petición PtObL, ha suministrado el bloque. El CC1 al procesar la petición PtObE no modifica el estado del bloque y espera la respuesta del CM a su petición³⁰.

Resumen. En la Figura 7.38 se muestran los estados en los cuales se infiere un cruce y la respuesta del CC.

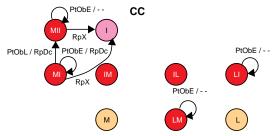


Figura 7.38 Protocolo MLI. Inferencia de cruces en un CC.

Diagramas completos de cruces de peticiones

En la Figura 7.39 se muestran las transiciones en un CC y el CM cuando se produce un cruce de peticiones a un bloque. En el CM se infiere un cruce cuando el CC, que ha emitido una petición PtXm o PtXl, no está en el VP del

30. En una sección previa de este capítulo se ha mostrado un diseño canónico y más ortodoxo. El CC en el estado MI puede recibir más de un tipo de petición del CM.

bloque. También, cuando se recibe una petición PtXm para un bloque que está en el estado L, la cual no es esperada. En ninguno de los casos hay que actualizar memoria.

Al procesar una petición, la primera acción en el CM es inferir si se ha producido un cruce de peticiones. Para ello, el CM analiza si el CC está en el VP. Además, en una petición PtXm, el CM analiza si el estado del bloque en el directorio se corresponde con la petición. En el diagrama de estados de un bloque en el directorio se utiliza el subíndice c para identificar una petición del CC que se ha cruzado con una petición del CM.

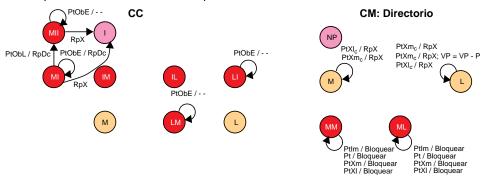


Figura 7.39 Diagramas de transiciones en un CC y en el directorio cuando se produce un cruce entre peticiones a un bloque entre un CC y el CM.

En un CC se infieren cruces en los estados transitorio LI, LM y MI. En cualquiera de ellos no se modifica el estado cuando se procesa una petición PtObE y sólo se responde con el bloque en el estado MI. En los estados LI y LM la petición PtObE sólo indica invalidación.

En el estado MI un CC también responde a una petición de bloque (PtObL) por parte del CM. Entonces, para identificar que no hay que suministrar el bloque en una petición PtObE posterior, se cambia al estado MII.

Observemos que el estado MI es similar al estado M cuando el CC procesa una petición PtObE. El siguiente estado estable debe ser I. En el caso del estado MI hay que esperar la respuesta del CM a la petición PtXm.

Cuando el CC procesa una petición PtObL, el estado MII es similar al estado LI. El siguiente estado estable debe ser I. Notemos que aunque la petición al CM sea PtXm, su única funcionalidad, al haber respondido a la petición PtObL del CM, una vez suministrado el bloque, es mantener el directorio preciso.

Tablas de estados y transiciones

En la Tabla 7.4 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.

			Evento	os del proces reemplazo	sador y	Eventos externos (respuestas y peticiones									
			LPr	EPr	CcRe	RpD	RpX	PtObL	PtObE						
	es	- 1	Pt; IL	Ptlm; IM											
	Estables	L	; L	Ptlm; LM	PtXI; LI				;I						
	В	M	; M	; M	PtXm; MI			RpDc: L	RpDc; I						
SC		IL				;L									
Estados	တ္	IM				; M									
ш	transitorios	LM				; M			; LM						
	ansi	LI					; I		; LI						
	i i	MI					;I	RpDc; MII	RpDc; MI						
		MII					;I		; MII						

Tabla 7.4 Peticiones concurrentes. Protocolo de directorio MLI. Tabla de estados y transiciones en un bloque de cache. Las casillas con fondo blanco indican cruces de peticiones.

En la Tabla 7.5 se muestran en formato tabla los estados y transiciones entre estados de un bloque en el directorio. 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.

Para identificar un cruce, se comprueba si el CC solicitante están en el vector de presencia o no. Además se comprueba si la petición recibida es esperada en el estado actual.

					E	Eventos del con	trolador de coh	erencia			
						PtXI			PtXm		
			Pt	Ptlm	VP = P	VP≠P P∈VP	P∉ VP	VP = P	VP≠P	P ∈ VP	RpDc
		NP	RpD; L, VP = P	RpD; M VP = P, BE = 1							
	Estables	L	RpD; L, VP = VP + P	PtObE {VP - P}, RpD; M, VP = P, BE = 1	RpX; NP, VP = vacio	RpX; L VP = VP - P	RpX; L	RpX; L		RpX; L VP = VP - P	
Estados	Щ	М	PtObL; ML	PtObE {VP}; MM			RpX; M	RpX; NP, VP = vacío, BE = 0	RpX; M		
4	transitorios	ML	Bloqueo	Bloqueo	Bloqueo	Bloqueo	Bloqueo	Bloqueo	Bloqueo	Bloqueo	RpD, Dev; L VP = VP + P, BE = 0
	trans	MM	Bloqueo	Bloqueo	Bloqueo	Bloqueo	Bloqueo	Bloqueo	Bloqueo	Bloqueo	RpD; M VP = P, BE = 1

Tabla 7.5 Peticiones concurrentes. Protocolo de directorio MLI. Tabla de estados y transiciones de un bloque en el directorio. Las casillas con fondo blanco indican cruces de peticiones.

Representación de cambios de estado y transiciones en una secuencia de accesos a memoria

En este apartado se muestran tres formas de representar una secuencia de accesos a memoria: a) en formato tabla, b) mediante un diagrama temporal y c) mediante un diagrama temporal simplificado.

Representación en formato tabla

Los accesos a memoria se muestran en grupos separados mediante líneas horizontales continua. Cada grupo de accesos a memoria se gestiona independientemente y no se empieza a gestionar el siguiente grupo hasta que ha finalizado el anterior. Los accesos a memoria de un grupo se realizan concurrentemente en el mismo ciclo. En una fila de la tabla, de izquierda a derecha, después de la instrucción de acceso a memoria se especifica:

- 1 La primera columna se utiliza para representar el arbitraje de peticiones. Esto es, se indica el orden. Supondremos usualmente que en un grupo de peticiones el árbitro las selecciona en el orden en que se especifica la secuencia de accesos a memoria.
- 2 El siguiente grupo de columnas se utiliza para representar el estado transitorio del bloque, en la cache correspondiente, cuando se emite la petición.

- 3 El tercer grupo de columnas se utiliza para representar la petición de cada CC en la RI. La petición se indica en la casilla correspondiente y en el orden en el que el árbitro concede el acceso al módulo de memoria.
- 4 El siguiente grupo de columnas se utiliza para identificar la variable o bloque y el VP en el directorio. Los bits en el VP identifican a las caches; de izquierda a derecha en ordinal creciente. Un valor de uno en el VP indica que hay una copia del bloque en la cache correspondiente. El campo E se corresponde con el BE de la entrada en el directorio.
- **5** En el quinto grupo de columnas sólo se representan peticiones del CM que requieren una respuesta del CC. En la columna arb se indica el receptor del mensaje y en la columna RMC el mensaje de petición del CM.
- 6 El siguiente grupo de columnas se utiliza para representar la red que utilizan los CC para responder al CM, si es el caso (RCM). Se representa al receptor del mensaje (columna arb) y el mensaje (columna que identifica la red).
- 7 El séptimo grupo de columnas se utiliza para indicar qué elemento (memoria o cache) suministra el bloque o dato.
- **8** En el octavo grupo de columnas se representan las respuestas del CM y las peticiones que no requieren respuesta de los CC. En la columna arb se indica el destinatario. En la columna RV se indica la respuesta del CM. Si además hay mensajes de petición se utiliza la siguiente fila para indicarlos.
- **9** El noveno grupo de columnas se utiliza para identificar la variable accedida en cache y el estado del bloque al finalizar la acción de coherencia. Cuando un bloque está en una ventana de vulnerabilidad y recibe una petición del CM, se indica el estado transitorio del bloque. En estas condiciones, el estado transitorio, debido a que se ha atendido una petición del CM, se observa en las columnas de la derecha.
- **10** Si no se accede al CM la fila correspondiente a la petición se deja en blanco.

Ejemplo. En la Tabla 7.3 se muestra una secuencia de accesos a memoria realizada por tres procesadores. Los accesos a memoria están agrupados de dos en dos. Cuando se inicia la secuencia de accesos a memoria, las caches no almacenan los bloques en los que se ubican las variables accedidas. Tampoco se producen conflictos de contenedor al almacenar los bloques en los contenedores de cache.

El primer par de accesos a memoria son debidos a un fallo de lectura y a un fallo de escritura. El estado transitorio de los bloques al emitir las peticiones es IL en la cache C1 (Pt) e IM en la cache C2 (PtIm). En las dos transacciones el

bloque lo suministra memoria. El estados de los bloques al finalizar las transacciones es L en la cache C1 y M en la cache C2. En el directorio ha sido activado el BE del bloque que contiene la variable u.

	arb.	C 1	C 2	C 3	F	RI		mem. ar		arb.	Red	arb.	Red		arb.	R	.V	C 1		C 2		С	3
acceso	ord.	est.	est.	est.	1º	2°	var.	VP	Е	ord.	RMC	ord.	RCM	sum.	ord.	1°	2°	var.	est.	var.	est.	var.	est.
1. P1 load t	Р1	IL			Pt		t	1, 0, 0	0					mem	CC1	RpD		t	L				
1. P2 store u	P2		IM			PtIm	u	0, 1, 0	1					mem	CC2		RpD			u	М		
2. P1 load u	Р1	IL			Pt		u	1, 1, 0	0	C2	PtObL	СМ	RpDc		CC1	RpD		u	L	u	L		
2. P3 load u	Р3			IL		Pt	u	1, 1, 1	0					mem	ССЗ		RpD					u	L
3. P1 load w	Р1	IL			Pt		w	1, 0, 0	0					mem	CC1	RpD		w	L				
3. P2 store u	P2		LM			PtIm	u	0, 1, 0	1					mem	CC2		RpD			u	М		
															CC1,3		PtObE	u	I			u	I
4. P1 load w																							
4. P3 load w	Р3			IL	Pt		w	1, 0 ,1	0					mem	ССЗ	RpD						w	L
5. P2 store w	P2		IM		PtIm		w	0, 1, 0	1					mem	CC2	RpD				w	М		
															CC1,3	PtObE		w	LM			w	I
5. P1 store w	Р1	LM				PtIm	w	1, 0, 0	1	C2	PtObE	СМ	RpDc		CC1	RpD		w	М	w	ı		

Tabla 7.6 Protocolo de directorio MLI. Secuencia de accesos a memoria concurrentes.

El segundo par de accesos a memoria son fallos de lectura. El CM procesa en primer lugar el mensaje emitido por el CC1. El procesado del mensaje requiere que el CM solicite el bloque a la cache C2, mediante una petición de observación de lectura (PtObL). El segundo fallo hace referencia al mismo bloque. Como memoria ha sido actualizada en la transacción previa, el CM suministra directamente el bloque. Al finalizar las dos transacciones el VP del bloque identifica que las caches C1, C2 y C3 tienen copia del bloque.

El tercer grupo de accesos a memoria requiere dos transacciones. La primera petición es de lectura (Pt) y el bloque lo suministra la memoria. La segunda petición es de lectura con intención de modificación (PtIm) y el bloque lo suministra la memoria. Adicionalmente, el CM emite peticiones de observación de escritura del bloque (PtObE) a las caches C1 y C3. El estado estable del bloque, al finalizar la transacción, es M en la cache C2 e I en las caches C1 y C3.

El cuarto par de accesos a memoria requiere sólo una transacción. El primer acceso a memoria es un acierto en cache. El segundo acceso a memoria, accede al mismo bloque, pero es un fallo. Al finalizar la transacción, el estado del bloque que contiene la variable w es L en las caches C1 y C3.

El último grupo de accesos a memoria son dos escrituras. Ninguna de las caches accedidas tiene el bloque en exclusividad. La cache C2 no tiene copia del bloque y la cache C1 tiene una copia en el estado L. La petición de CC2 requiere que el CM emita una petición de observación de escritura al CC1 y al CC3. Posteriormente, el CM, al procesar la petición de CC1, emite un mensaje de petición de observación de escritura al CC2, el cual suministra el bloque al CM. Finalmente el CM suministra el bloque a CC1. Al finalizar las dos transacciones, el VP del bloque en el directorio indica que la cache C1 tiene copia del bloque en exclusividad.

Diagrama temporal

Antes de utilizar una red para transmitir el mensaje se representa el arbitraje. Esta fase se utiliza para mostrar la ordenación de los mensajes. La espera para acceder a una red se indica representando en ciclos consecutivos la fase de arbitraje (arb). El resto se representa de la forma descrita en el Capítulo 5.

Ejemplo. En la Figura 7.40 se muestra el diagrama temporal de la secuencia de accesos mostrada en la Tabla 7.6. Observemos el bloqueo del procesado de peticiones en el CM en el segundo grupo de peticiones. La primera petición que procesa el CM requiere que la cache C2 suministre el bloque. Durante el intervalo de tiempo, entre la petición del CM y la respuesta del CC2, el CM no procesa la siguiente petición en la CP, ya que hace referencia al mismo bloque.

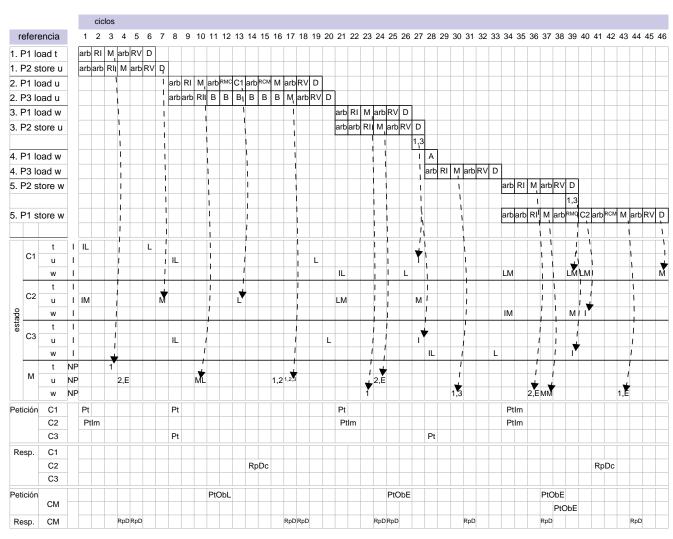


Figura 7.40 Protocolo de directorio MLI. Diagrama temporal de una secuencia de accesos a memoria concurrentes.

Diagrama temporal simplificado

En la Figura 7.41 se muestra el diagrama temporal simplificado de la secuencia de accesos mostrada en la Tabla 7.6. El bloqueo del procesado de una petición en el CM se indica mediante una línea curva a trazos.

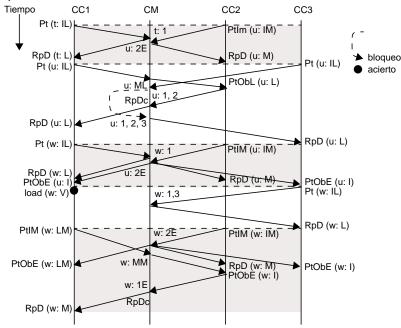


Figura 7.41 Protocolo de directorio MLI. Diagrama temporal simplificado de una secuencia de accesos a memoria concurrentes.

PARALELISMO

En un multiprocesador el tráfico de peticiones es proporcional al número de procesadores. Para incrementar el rendimiento es necesario que la memoria pueda servir varias peticiones en cada ciclo. Aunque el directorio conceptualmente es una estructura centralizada, puede distribuirse añadiendo más módulos de memoria con un CM y directorio asociado.

En este apartado se incrementa el paralelismo, utilizando varios módulos de memoria y distribuyendo el directorio, con el objetivo de incrementar el número de peticiones procesadas por unidad de tiempo o ancho de banda.

Organización del multiprocesador

El paralelismo en el procesado de peticiones se obtiene utilizando varios módulos de memoria, cada uno con un CM, que se entrelazan por bloque³¹. El directorio de cada CM gestiona los bloques ubicados en el módulo de memoria. Esta organización facilita que se distribuya el tráfico de peticiones entre los módulos de memoria y el sistema es escalable en cierta medida. En el mejor caso se podrán estar sirviendo tantas peticiones de los CC como módulos de memoria, si hay menos módulos de memoria que CC.

En la Figura 7.42 se muestra un esquema de un multiprocesador con varios módulos de memoria. La RI y la RV son redes crossbar³². La RV se utiliza para transmitir peticiones y respuestas de los CM (las redes lógicas RV y RMC son la misma red). En la misma figura se muestra el detalle del conexionado que permite que los procesadores envíen las peticiones hacia un módulo de memoria. Si en un mismo ciclo las peticiones de los procesadores son a módulos de memoria distintos, la red crossbar permite que se sirvan todas las peticiones. Cuando es necesario para el protocolo de coherencia, hay una red de respuestas de los CC a los CM (red RCM). Esta red también es un crossbar.

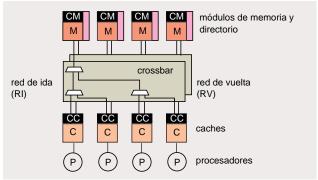


Figura 7.42 Multiprocesador con varios módulos de memoria.

En un ciclo determinado, un CM puede emitir una respuesta (RpC) y varias peticiones (PtObE)³³. También, en el mismo ciclo, otro CM puede realizar la misma acción. Además, la intersección de los mensajes emitidos por los dos CM puede ser distinta del conjunto vacío. Por tanto, es necesario un algoritmo de arbitraje en la RV. Este algoritmo debe ser tal que permita que todos los CC

- 31. Bloques consecutivos del espacio físico se almacenan en módulos de memoria distintos. La granularidad del entrelazado puede ser mayor. Por ejemplo, una página del espacio físico de direcciones.
- 32. Podemos observar una red crossbar como un grupo de buses independientes. Cada uno de ellos permite la conexión a un módulo de memoria.
- 33. Por ejemplo, en el protocolo de directorio VI. En una petición de escritura, el CM emite una respuesta y puede emitir varias peticiones de observación de escritura si hay copias del bloque en otras caches.

observen el mismo orden lógico global de los mensajes emitidos en cada transacción desde los CM a los CC. Esto es, la RV debe ser ordenada. Esta característica permite eliminar la necesidad de que las peticiones de observación de escritura (PtObE) deban responderse.

En la Figura 7.43 se muestra, mediante un ejemplo, la necesidad de un algoritmo de arbitraje en la RV, que garantice que los CC observen el mismo orden lógico global. Los dos trozos de código son el esqueleto del algoritmo de Dekker para exclusión mutua. En el desarrollo utilizamos un protocolo de directorio VI.

Las variables aviso1 y aviso2 están contenidas en bloques que se ubican en los módulos de memoria M1 y M2 respectivamente. Antes de iniciarse la ejecución, la cache C1 tiene copia del bloque que contiene la variable aviso2 y la cache C2 tiene copia del bloque que contiene la variable aviso1. El valor de las dos variables es cero.

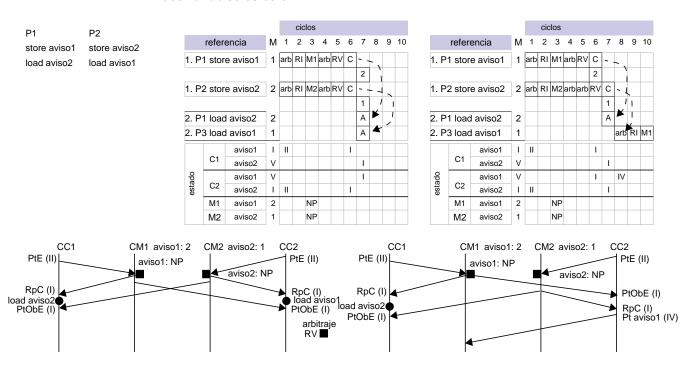


Figura 7.43 Protocolo de directorio VI. Ejemplos de arbitraje en la RV. El acrónimo M indica módulo de memoria.

Las instrucciones store de los procesadores P1 y P2 son fallos en cache. Las peticiones que emiten CC1 y CC2 acceden a CM distintos. Por tanto progresan de forma paralela hasta el CM correspondiente. El CM1 responde al CC1 y

efectúa una petición al CC2. El CM2 responde al CC2 y efectúa una petición al CC1. La intersección de los conjuntos de mensajes de los dos CM es distinta del conjunto vacío. Por tanto, es necesario un arbitraje que serialice la utilización de la RV.

En la parte izquierda de la Figura 7.43 se muestra el caso de que el arbitraje conceda las conexiones a las dos respuestas. Suponemos que mientras se reciben las peticiones de observación de escritura (PtObE) los procesadores ejecutan las instrucciones load, ya que para cada uno de ellos ha finalizado la ejecución de la instrucción store (flecha en la Figura 7.43)³⁴, aunque no han sido globalmente consolidadas. En los dos procesadores el acceso a memoria es un acierto en cache y el valor leído en los dos casos es cero. Este valor permite que los dos procesadores accedan a la sección crítica.

Los procesadores observan las escrituras en un orden distinto. El procesador P1 al recibir la respuesta ejecuta la siguiente instrucción. El procesador P2 también ejecuta la siguiente instrucción al recibir la respuesta, pero el procesador P2 no ha observado la escritura de P1. El razonamiento intercambiando los procesadores es el mismo.

En la parte derecha de la Figura 7.43 se muestra el caso de que el arbitraje conceda en primer lugar el acceso a la RV a los mensajes que emite M1. En este caso el procesador P1 accede a la sección crítica. Los dos procesadores observan las escrituras en el mismo orden.

Crossbar de vuelta y arbitraje. En la Figura 7.44 se muestra un posible diseño de la RV. Desde cada CM existe una conexión punto a punto con cada CC. En cada CC existe un multiplexor cuyas entradas son las conexiones con los CM. Notemos que un CM está conectado en todos los multiplexores en la entrada identificada con el mismo ordinal.

Una entrada determinada, de un multiplexor, tiene la misma prioridad relativa en todos los multiplexores respecto a las otras entradas del multiplexor. Esto es, la prioridad de la conexión de un CM es la misma en todos los CC.

Entonces, dado un ciclo en el que hay peticiones o respuestas estas se arbitran. No se tiene en cuenta ninguna nueva solicitud de transmisión de mensajes hasta que se han arbitrado todas las que había en el ciclo previo. En un ciclo se distinguen subciclos de arbitraje hasta que se han arbitrado todas las peticiones o respuestas³⁵.

^{34.} Tengamos en cuenta que una fase de una transacción son varios ciclos de procesador. También, el acceso del agente procesador puede ser prioritario respecto del agente observador. Los mensajes recibidos se pueden almacenar en una cola.

^{35.} En este contexto los CM se bloquean hasta que sus mensajes han sido transmitidos.

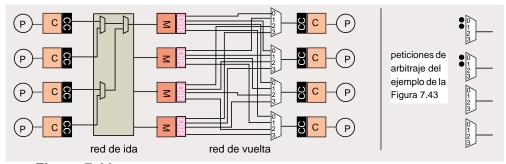


Figura 7.44 Esquema de la red de vuelta.

Supongamos que la entrada cero de los multiplexores es la más prioritaria. Entonces, en el primer subciclo la salida de los multiplexores es la entrada cero, en el segundo subciclo la entrada uno y así de forma sucesiva³⁶. De esta forma se garantiza que todos los CC observan las peticiones o respuestas de los CM en el mismo orden (red ordenada). En la parte derecha de la Figura 7.44 se muestran las peticiones paralelas de la Figura 7.43.

Optimización. Notemos que con el funcionamiento descrito, en un subciclo puede haber peticiones o respuestas en las entradas de un multiplexor y no existir ninguna en la salida. Esto es debido a que el sincronismo en la selección de la entrada determina que la entrada elegida no tenga petición o respuesta. Una mejora en el rendimiento es que en cada subciclo se seleccione, partiendo de la entrada más prioritaria, la primera entrada con petición. Notemos que si un CM ha emitido varios peticiones hacia los CC y existen peticiones o respuestas de otros CM, la peticiones o respuestas de un CM no tienen porqué ser observadas en el mismo ciclo en los CC, pero si serán observadas en el mismo orden global. El tiempo físico en el cual una petición de un CM llega a un CC no es importante, mientras el orden global en el que todos los CC del multiprocesador observen una petición sea el mismo.

Arbitraje sin sesgo. Un arbitraje con prioridad fija como el descrito previamente favorece siempre a los mismos CM. Una posibilidad, para reducir el sesgo, es que en ciclos pares la entrada más prioritaria del multiplexor sea la que tiene el menor ordinal. En ciclos impares la entrada más prioritaria es la que tiene el mayor ordinal.

36. Este funcionamiento es similar al caso de un multiprocesador con varios buses y un protocolo de observación (Capítulo 4). Para disponer de consistencia secuencial los buses deben ser observados en el mismo orden por todos los CC. En el contexto de buses se dispone de un buffer donde se almacenan las observaciones inducidas por la transacciones que transportan los buses. En particular, en un protocolo VI sólo se inducen peticiones de invalidación. Ahora bien, para utilizar la respuesta de una petición del CC (propia) el buffer debe vaciarse (orden en el bus).

Incremento del rendimiento. Para incrementar el rendimiento se pueden utilizar colas en las entradas de los multiplexores, una para cada conexión con un CM. Ahora bien, para mimetizar el diseño sin colas, la misma entrada en todas las colas de todos los multiplexores representa un ciclo de peticiones. En un ciclo determinado, todos los CM que emiten una petición o respuesta la ubican en la misma entrada de la cola correspondiente. Entonces, el arbitraje descrito previamente se utiliza para arbitrar todas las entradas asociadas al mismo ciclo.

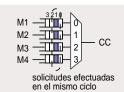


Figura 7.45 Cola de mensajes ordenados en las entradas de un multiplexor.

Una mejora es, en un multiplexor determinado, pasar a arbitrar el siguiente ciclo cuando no hay peticiones o respuestas pendiente de arbitrar en el ciclo actual. Entonces, en un multiplexor no se empieza a procesar el siguiente ciclo de peticiones o respuestas hasta que se han procesado todas las peticiones o respuestas del ciclo previo. En estas condiciones, un multiplexor puede estar en un ciclo par y otro multiplexor en un ciclo impar.

Ejemplo de diseño. El diseño previo se puede implementar cerca de los módulos de memoria y utilizarlo exclusivamente para garantizar un orden lógico global. Notemos que en un subciclo determinado sólo se selecciona una entrada en cada multiplexor. Por tanto, el diseño descrito se puede utilizar como algoritmo de arbitraje en una red crossbar que transporte la información desde los CM a los CC.

En la Figura 7.46 se muestra un esquema de diseño de la red crossbar desde los CM a un CC. El esquema para la conexión de los CM a cada uno de los otros CC es idéntico. Los CM efectúan peticiones al árbitro. El árbitro en cada ciclo almacena las peticiones efectuadas en una entrada de la cola de peticiones de arbitraje (CPA). Esto es, cada entrada almacena las peticiones efectuadas por los CM en ese ciclo.

La CPA se gestiona de forma FIFO. Dada una entrada en la CPA se conceden todas las peticiones antes de pasar a la siguiente entrada. El vector de concesiones (c0, c1, c2, c3) en cada ciclo sólo tiene un bit activo. Cada bit se corresponde con la concesión del arbitraje a un CM (el bit Ci se corresponde con la petición del CMi). Un bit de concesión se utiliza para controlar un multiplexor. Notemos que podemos considerar que cada multiplexor, menos el último, está

asociado a un CM. El último multiplexor se controla con el bit del CM asociado y la función OR de los grupos de bits de concesión de los CM con ordinal menor y los grupos de bits con ordinal mayor.

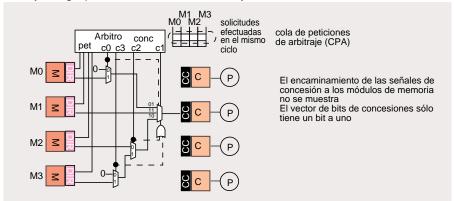


Figura 7.46 Crossbar: esquema de arbitraje y encaminamiento de los CM a un CC.

En los diagramas temporales que se utilicen, para simplificar el arbitraje, supondremos que los árbitros conceden el acceso a la red a todos los mensajes emitidos desde un CM o a ninguno. Este funcionamiento no prohíbe que accedan a la red mensajes emitidos por varios CM mientras la intersección sea nula. En la Figura 7.47 se muestra un ejemplo donde hay que arbitrar en la RV. Los arbitros conceden el acceso en primer lugar a M1 y después a M2.

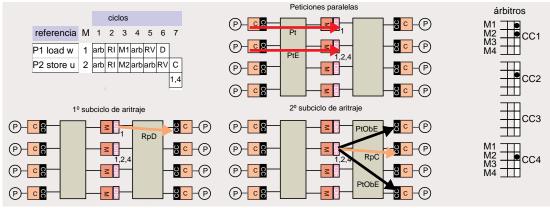


Figura 7.47 Protocolo de directorio VI. Arbitraje en la RV.

En la Figura 7.48 se muestra otro ejemplo de arbitraje. La primera transacción es de 4 pasos y la segunda transacción es de dos pasos. El destinatario de la petición de CM inducida por la primera transacción también es el destinatario de un mensaje de la segunda transacción servida por otro CM. Recor-

demos que las redes lógicas RMC y RV son la misma red. Por tanto, hay que arbitrar. El árbitro concede el arbitraje a la petición inducida por la primera transacción.

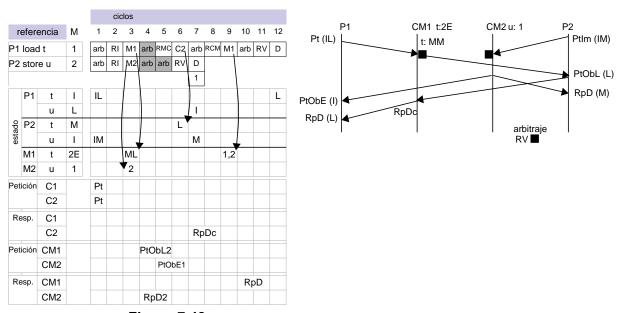


Figura 7.48 Protocolo de directorio MLI. Arbitraje en la RV.

Coherencia. El CM de cada módulo de memoria es el punto de ordenación de las transacciones que referencian los bloques que almacena. En cada CM se propagan y serializan las transacciones de escritura de cada uno de los bloques que gestiona. La RV mantiene, mediante el arbitraje, un orden lógico global de los mensajes emitidos por los CM a los CC. Una escritura es observada, por todos los procesadores involucrados, en el mismo orden global. Por tanto, todos los CC observan las peticiones de observación de escritura a un bloque, efectuadas por un CM, en el mismo orden y no es necesario que los CC respondan a las peticiones de invalidación.

Consistencia. Los CM y el arbitraje en la RV determinan un orden lógico global de todas las transacciones. Una escritura está consolidada cuando el CC recibe la respuesta del CM. Una petición de lectura lee el valor establecido por la instrucción store previa a la misma dirección.

Protocolo de directorio VI

Los diagramas de transiciones entre estados en el CM y en un CC son los descritos previamente en este capítulo.

Organización

En la Figura 7.49 se muestra la organización del multiprocesador.

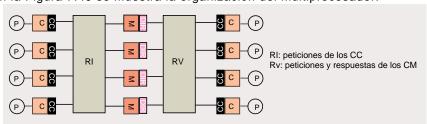


Figura 7.49 Protocolo VI. Redes de comunicación.

Representación de cambios de estado y transiciones en una secuencia de accesos a memoria

En este apartado se muestran tres formas de representar una secuencia de accesos a memoria: a) en formato tabla, b) mediante un diagrama temporal y c) mediante un diagrama temporal simplificado.

Representación en formato tabla. La representación es similar a la utilizada con un módulo de memoria. Seguidamente se indican las diferencias.

- 1 Se añade una primera columna para indicar el ordinal del módulo de memoria accedido en la transacción.
- 2 En arbitraje de la RI se distinguen varias columnas. El número de columnas en el peor caso es el número de accesos a memoria incluidos en el mismo grupo. En la casilla de la columna que indica el orden se especifica el módulo de memoria al que se accede. A peticiones de acceso realizadas al mismo módulo de memoria, el árbitro les concede el acceso en ciclos consecutivos. La concesión es en el orden en que se especifican en la secuencia de accesos a memoria.
- 3 En la red de vuelta (RV) hay que tener encuentra lo siguiente: si las peticiones se han servido en el mismo ciclo hay que comprobar que las respuestas y peticiones que generan los CM tengan intersección nula. Esto es, dos mensajes de distintos CM no se pueden enviar al mismo CC. Si la intersección es nula pueden ocupar la red de vuelta en el mismo ciclo. La concesión es en el orden en que se especifican en la secuencia de accesos

a memoria. En la columna correspondiente se indica el mensaje de respuesta. Si además hay mensajes de petición se utiliza la siguiente fila para indicarlos.

Ejemplo. En la Tabla 7.7 se muestra una secuencia de accesos a memoria realizada por tres procesadores. Los accesos a memoria están agrupados de dos en dos.

Las variables t, u, w están ubicadas en bloques distintos de memoria. Los bloques que contienen las variables t y w se almacenan en el módulo de memoria M1. El otro bloque se almacena en el módulo de memoria M2. Al almacenar los bloques en los contenedores de caches no se producen conflictos. Inicialmente no hay copia de los bloques en las caches.

En el primer grupo de accesos a memoria, las peticiones emitidas por los CC se encaminan a CM distintos y estos, al procesar la petición, sólo emiten un mensaje de respuesta al CC que ha emitido la petición. Por tanto, existe paralelismo.

			b.	C 1	C 2	C 3	RI			mem.		arb.	R	RV	С	1	С	2	С	3
acceso	М	1º	2°	est.	est.	est.	1º	2°	var.	VP	sum.	ord.	1º	2°	var.	est.	var.	est.	var.	est.
1. P1 load t	1	М1		IV			Pt		t	1, 0, 0	mem.	CC1	RpD		t	٧				
1. P2 store u	2	M2			П		PtE		u	0, 0, 0	C2	CC2	RpC				u	ı		
2. P1 load u	2	М2		IV			Pt		u	u 1, 0, 0 n		CC1	RpD		u	٧				
2. P3 load u	2		М2			IV		Pt	u	1, 0, 1	mem.	ССЗ		RpD					u	V
3. P1 load w	1	М1		IV			Pt		w	1, 0, 0	mem	CC1	RpD		W	٧				
3. P2 store u	2	М2			П		PtE		u	0, 0, 0	C 2	CC2		RpC			u	ı		
												CC1,3		PtObE 1,3	u	I			u	ı
4. P1 load w	1																			
4. P3 load w	1	М1				IV	Pt		w	1, 0 ,1	mem	ССЗ	RpD						w	V
5. P2 store w	1	М1			П		PtE		w	0, 0, 0	C 2	CC2	RpC							
												CC1,3	PtObE 1,3		W	VVI			w	I
5. P1 store w	1		M1	V V				PtE	w	0, 0, 0	C1	CC1		RpC	w	ı				

Tabla 7.7 Protocolo de directorio VI. Paralelismo en la secuencia de accesos.

En el segundo grupo de accesos a memoria, las peticiones emitidas por los CC requieren acceder al mismo CM. Por tanto, son serializada por el árbitro en la RI.

La peticiones del tercer grupo de accesos a memorias acceden a CM distintos. Hasta el procesado en los CM existe paralelismo. Sin embargo, la intersección de los mensajes emitidos por los dos CM es distinta del conjunto vacío. En consecuencia deben serializarse. En primer lugar el árbitro concede la RV al CM1 y posteriormente al CM2.

En el cuarto grupo de accesos a memoria, el acceso del procesador P1 es acierto en cache.

En el quinto grupo de accesos a memoria, las peticiones emitidas por los CC acceden al mismo CM. Por tanto, el arbitraje serializa la utilización de la RI.

Diagrama temporal

Después de utilizar la RI se representa el acceso a memoria, indicando como sufijo el ordinal del módulo de memoria accedido. Antes de utilizar una red para transmitir el mensaje o los mensajes se representa el arbitraje. Esta fase se utiliza para mostrar la ordenación de los mensajes. La espera para acceder a una red se indica representando en ciclos consecutivos la fase de arbitraje (arb). La concesión es en el orden en que se especifican en la secuencia de accesos a memoria. El resto se representa de la forma descrita previamente en este Capítulo.

Ejemplo. En la Figura 7.50 se muestra el diagrama temporal de la secuencia de accesos mostrada en la Tabla 7.7. Observemos el arbitraje en la RV en el tercer grupo de accesos a memoria.

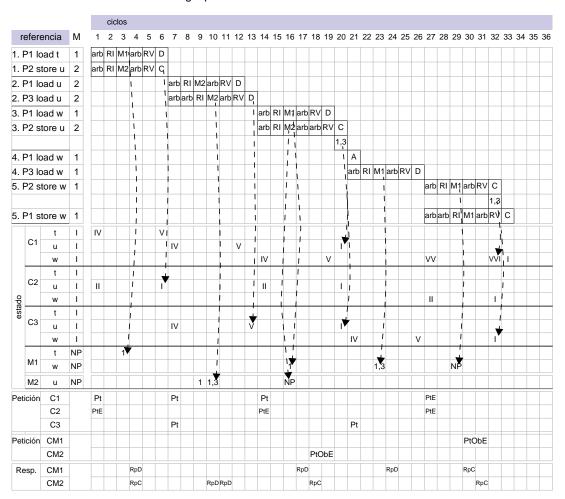


Figura 7.50 Protocolo de directorio VI. Diagrama temporal de una secuencia de accesos paralelos a memoria.

Diagrama temporal simplificado

En la Figura 7.51 se muestra el diagrama temporal simplificado del flujo de mensajes, en los dos sentidos, entre los CC y los CM. En particular se muestra el arbitraje en las distintas redes. La secuencia de accesos es la mostrada en la Tabla 7.7.

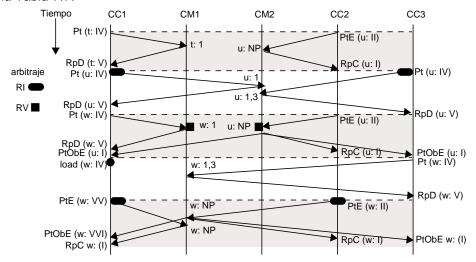


Figura 7.51 Protocolo de directorio VI. Diagrama temporal simplificado de una secuencia de accesos paralelos.

Protocolo de directorio MLI

Los diagramas de transiciones entre estados en el CM y en un CC son los descritos previamente en este capítulo.

Organización del multiprocesador

En la Figura 7.52 se muestra la organización del multiprocesador.

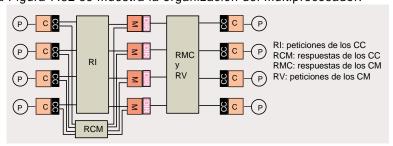


Figura 7.52 Protocolo MLI. Redes lógicas de comunicación.

Representación de cambios de estado y transiciones en una secuencia de accesos a memoria

Representación en formato tabla. La representación es similar a la utilizada con un módulo de memoria. Seguidamente se indican las diferencias.

- 1 Se añade una primera columna para indicar el ordinal del módulo de memoria accedido en la transacción.
- 2 En arbitraje de la RI se distinguen varias columnas. El número de columnas en el peor caso es el número de accesos a memoria incluidos en el mismo grupo. En la casilla de la columna que indica el orden se especifica el módulo de memoria al que se accede. A peticiones de acceso realizadas al mismo módulo de memoria, el árbitro les concede el acceso en ciclos consecutivos. La concesión es en el orden en que se especifican en la secuencia de accesos a memoria.
- **3** En la red RMC la etiqueta arb tiene varias columnas. En ellas se representa el destinatario de la petición de cada CM. En la columna etiquetada como RMC se indica el mensaje emitido por el CM. Si hay peticiones concurrentes con el mismo destinatario hay que arbitrar³⁷.
- 4 En la red RCM la etiqueta arb tiene varias columnas. En ellas se representa el destinatario de la petición del CC. En la columna etiquetada como RCM se indica el mensaje emitido por el CM. Si hay peticiones concurrentes con el mismo destinatario hay que arbitrar.
- **5** El siguiente grupo de columnas corresponde a la RV. En las columnas etiquetadas como arb hay que indicar el destinatario. En la columna correspondiente, etiquetada como RV, se indican el mensaje de respuesta. Si además hay mensajes de petición se utiliza la siguiente fila para indicarlos.

Ejemplo. En la Tabla 7.8 se muestra una secuencia de accesos a memoria realizada por tres procesadores. Los accesos a memoria están agrupados de dos en dos.

Las variables t, u, w están ubicadas en bloques distintos de memoria. Los bloques que contienen las variables t y w se almacenan en el módulo de memoria M1. El otro bloque se almacena en el módulo de memoria M2. Al almacenar los bloques en los contenedores de caches no se producen conflictos. Inicialmente no hay copia de los bloques en las caches.

Las dos primeras instrucciones emiten peticiones que acceden a módulos de memoria distintos. Por tanto, se procesan en paralelo.

37. Recordemos que las redes lógicas RMC y RV son la misma red.

		ar	b.	C 1	C 2	C 3	R	1		mem.		aı	rb	Red	aı	ъ	Red		ar	b.	R	.V	C 1		C 2		С	3
acceso	М	1º	2°	est.	est.	est.	1º	2°	var.	VP	Ε	1º	2°	RMC	1º	2°	RCM	sum.	1º	2°	1º	2°	var.	est.	var.	est.	var.	est.
1. P1 load t	1	M 1		IL			Pt		t	1, 0, 0	0							mem	CC1		RpD		t	L				
1. P2 store u	2	M2			IM		PtIm		u	0, 1, 0	1							mem	CC2			RpD			u	М		
2. P1 load u	2	M2		IL			Pt		u	1, 1, 0	0	P2		PtObL2	М2		RpDc	C2	CC1		RpD		u	L	u	L		
2. P3 load u	2		М2			IL		Pt	u	1, 1, 1	0							mem		ССЗ		RpD					u	L
3. P1 load w	1	M1		IL			Pt		w	1, 0, 0	0							mem	CC1		RpD		w	L				
3. P2 store u	2	M2			LM		PtIm		u	0, 1, 0	1							mem		CC2		RpD			u	М		
																				CC1,3		PtObE	u	ı			u	ı
4. P1 load w	1																											
4. P3 load w	1	M1				IL	Pt		w	1, 0 ,1	0							mem	ССЗ		RpD						w	L
5. P2 store w	1	M1			IM		PtIm		w	0, 1, 0	1							mem	CC2		RpD				w	М		
																			CC1,3		PtObE		w	IM			w	Ι
5. P1 store w	1		M 1	LM				PtIm	w	1, 0, 0	1	P2		PtObE2	М1		RpDc	C2		CC1		RpD	w	М	w	ı		

Tabla 7.8 Protocolo de directorio MLI. Paralelismo en la secuencia de accesos

El segundo grupo de peticiones accede a la misma variable y por tanto, al mismo módulo de memoria. El acceso al módulo de memoria es serializado por el árbitro de la RI. El segundo acceso, al acceder a la misma variable que el primero, es serializado por el CM. El procesado del segundo acceso queda bloqueado hasta que finaliza el procesado del primero.

El tercer grupo de peticiones accede a módulos de memoria distintos. La intersección de los conjunto de mensajes emitidos por el CM1 y el CM2 es distinto del conjunto vacío. En consecuencia hay que arbitrar en RV. En primer lugar se transmite el mensaje de CM1 y posteriormente los mensajes de CM2.

En el cuarto grupo de instrucciones, la primera de ellas es un acierto en cache. la segunda es un fallo.

El quinto grupo de peticiones accede al mismo módulo de memoria y en particular a la misma variable. Las dos transacciones son de escritura. El árbitro de la RI determina que en primer lugar se procese la petición del CC2. La primera transacción es de 2 pasos y segunda de 4 pasos.

Diagrama temporal

Después de utilizar la RI se representa el acceso a memoria, indicando como sufijo el ordinal del módulo de memoria accedido. Antes de utilizar una red para transmitir el mensaje o los mensajes se representa el arbitraje. Esta fase se utiliza para mostrar la ordenación de los mensajes. La espera para acceder a una red se indica representando en ciclos consecutivos la fase de arbitraje (arb). El resto se representa de la forma descrita previamente en este Capítulo.

Ejemplo. En la Figura 7.53 se muestra el diagrama temporal de la secuencia de accesos mostrada en la Tabla 7.8. Observemos el bloqueo del segundo acceso a memoria en el segundo grupo de accesos. En el tercer grupo de accesos hay que arbitrar en RV.

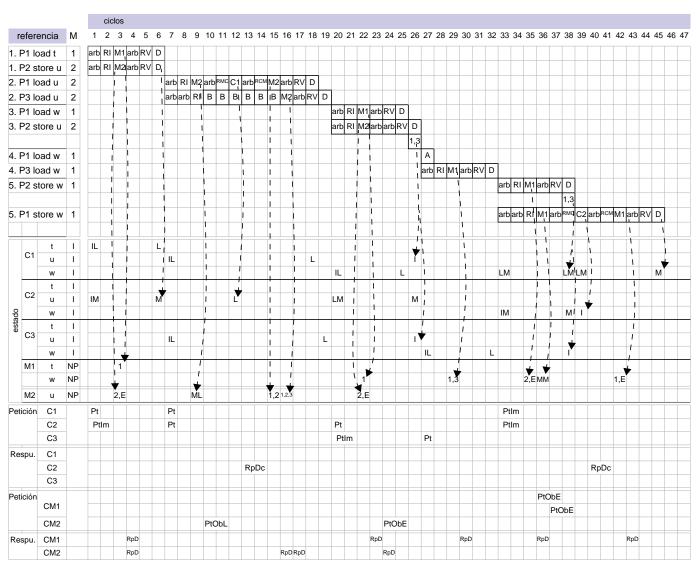


Figura 7.53 Protocolo de directorio MLI. Diagrama temporal de una secuencia de accesos paralelos a memoria.

Diagrama temporal simplificado

En la Figura 7.54 se muestra el diagrama temporal simplificado del flujo de mensajes, en los dos sentidos, entre los CC y los CM. En particular se muestra el arbitraje en las distintas redes y la espera en la CP de una petición para ser procesada por el CM (serialización de escrituras). La secuencia de accesos es la mostrada en la Tabla 7.8.

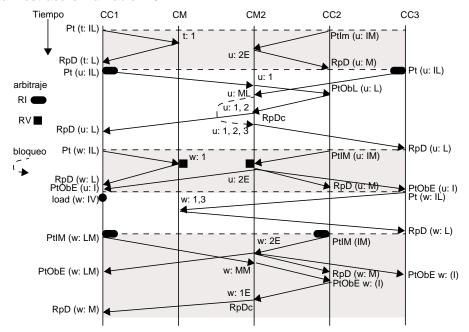


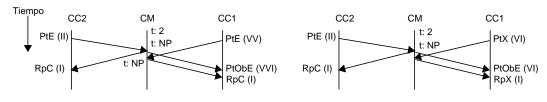
Figura 7.54 Protocolo de directorio MLI. Diagrama temporal simplificado en una secuencia de accesos paralela.

EJEMPLOS

Protocolo de directorio VI. Diagramas temporales simplificados

Pregunta 1: Represente mediante un diagrama temporal simplificado los ejemplos mostrados en la Figura 7.18 y en la Figura 7.19 relativos a cruces de peticiones en un protocolo de directorio VI.

Respuesta: En la parte izquierda de la figura se muestra el diagrama correspondiente a la Figura 7.18 y en la parte derecha el diagrama correspondiente a la Figura 7.19. El CC de la izquierda del diagrama es el que accede en primer lugar al CM.



Protocolo de directorio MLI. Cruces de peticiones

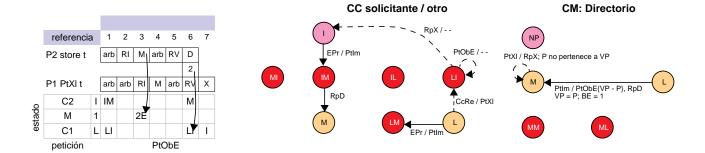
Pregunta 1: Para las secuencias de accesos a memoria mostradas en la Figura 7.30 y en la Figura 7.32 muestre un diagrama temporal.

Respuesta:

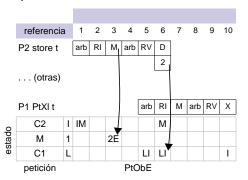
Orden A. En el estado LI un CC puede recibir una petición PtObE, lo cual indica que el CM ha procesado previamente una petición PtIm de otro CC. Cuando en el CM se procesa la petición PtXI el CC no está en el VP.

En la siguiente figura se muestra un diagrama temporal y las transiciones entre estados. En trazo continuo se muestran las transiciones debidas al CC de la cache C2. En trazo discontinuo las transiciones debidas al CC de la cache C1.

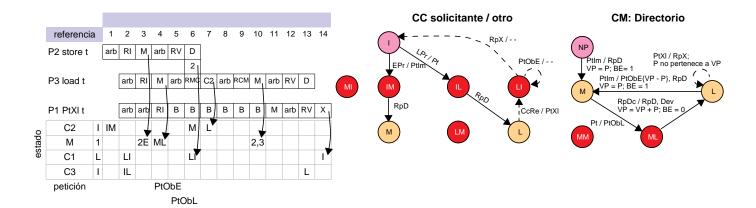
El cruce en el CM se observa en el estado M y el CC no cambia el estado y espera la respuesta.



En la cache C1, la transición del estado estable del bloque a un estado transitorio está determinado por la emisión de una petición de CC1. Como se muestra en la siguiente figura, es suficiente que la emisión se efectúe antes que la recepción de la petición del CM. Notemos que entre las dos transacciones, el CM puede procesar otras transacciones de otros CC que referencian el mismo o distinto bloque.



Orden B. Antes de procesar la petición de expulsión se puede procesar en el CM, además de la petición de exclusividad (Ptlm), una o varias peticiones de lectura del bloque (Pt). En estas condiciones la petición de expulsión se procesa en el CM en el estado L y en el VP no está incluido el CC1 (siguiente figura). El cruce en el CM se observa en el estado L y el CC1, al recibir la petición del CM, se queda en el mismo estado esperando la respuesta.

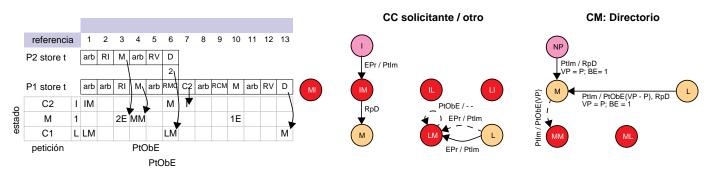


Si la petición de P3 en la figura previa es PtIm, debido a un store, el cruce en el CM se observa en el estado M, el cual es el orden A.

Orden C. En el estado LM un CC puede recibir una petición PtObE, lo cual indica que el CM ha procesado previamente una petición PtIm de otro CC.

En la siguiente figura se muestra un diagrama temporal y las transiciones entre estados. En trazo continuo se muestran las transiciones debidas al CC de C2. En trazo discontinuo las transiciones debidas al CC de C1.

En la cache C1, la transición del estado estable del bloque a un estado transitorio está determinado por la emisión de una petición de CC1. Es suficiente que la emisión se efectúe antes que la recepción de la petición del CM³⁸. El CM no infiere ningún cruce, ya que la petición PtIm se puede recibir en el CM cuando el bloque en el CC está en los estados I o L.



38. Notemos que temporalmente el procesado de la petición de CC2 en el CM puede ser bastante después si hay entrelazadas peticiones a otros bloques.

La petición del CM es para invalidar el bloque. Sin embargo, el CC1 no cambia de estado, ya que su petición es PtIm y en la respuesta se incluye el bloque.

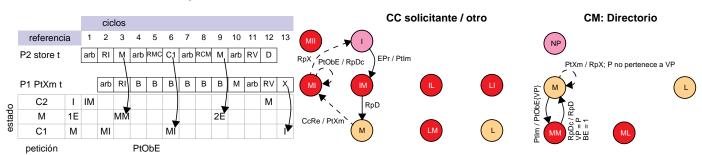
El funcionamiento de la RV garantiza que el CC1 recibe primero la petición y después la respuesta. De forma similar, el CC2 recibe primero la respuesta a su petición y después la petición del CM, inducida al procesar la petición del CC1.

Orden D. El comportamiento del CC1 en el orden D de accesos a memoria es el mismo que en el orden C. El CM al procesar la petición del CC3 no emite ningún mensaje al CC1. El bloque lo tiene en exclusividad el CC2.

Ordenes E, F y H. En el estado MI un CC puede recibir peticiones PtObE o PtObL y la respuesta es suministrar el bloque. Estos casos son los ordenes E y F de la Figura 7.32. También puede recibir una secuencia PtObL seguido de PtObE. En este caso, la respuesta a la petición PtObL es suministrar el bloque y la petición PtObE no tiene respuesta, ya que es una petición para invalidar el bloque. Este caso es el orden H de la Figura 7.32.

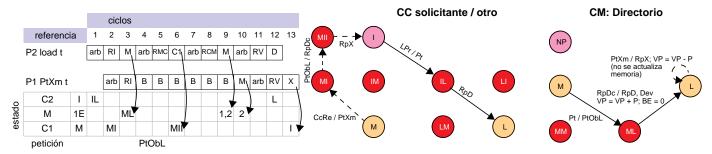
Para identificar la posible secuencia de peticiones desde el CM añadimos un estado transitorio denominado MII.

Orden E. El CM, antes de procesar una petición de expulsión PtXm, procesa una petición de exclusividad (PtIm). En estas condiciones la petición de expulsión se procesa en el estado M y en el VP no está incluido el CC que efectúa la petición de expulsión. El cruce en el CM se observa en el estado M. El CC1 suministra el bloque, no cambia el estado del bloque y espera la respuesta.

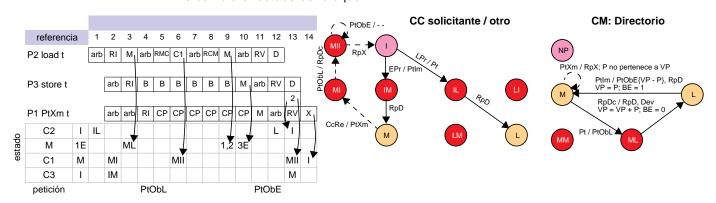


Orden F. Antes de procesar una petición de expulsión, el CM puede procesar una petición de bloque (Pt), la cual requiere que el CM emita una petición de observación de lectura (PtObL). En estas condiciones la petición de expulsión (PtXm) se procesa en el CM en el estado L y el CC1 está incluido en el VP. Sin embargo, no es una petición esperada en este estado. Por tanto, aunque sea una petición PtXm no se actualiza memoria³⁹. El cruce en el CM se

observa en el estado L. El CC1, al recibir la petición PtObL, suministra el bloque, cambia el estado del bloque a MII y espera la respuesta a la petición de expulsión.



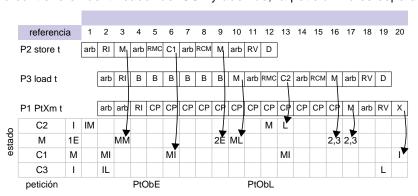
Orden H. Antes de procesar una petición de expulsión el CM puede procesar, además de la petición de bloque (Pt), la cual induce una petición de bloque (PtObL), una petición de exclusividad (PtIm), la cual induce una petición de observación de escritura (PtObE) desde el CM. En estas condiciones, la petición de expulsión se procesa en el CM en el estado M y el CC1 no está incluido en el VP. Esto es, el cruce en el CM se observa en el estado M. Cuando el CC1 recibe la petición PtObE el bloque está en estado MII y el CC1 no cambia el estado del bloque.



Orden G. El CM antes de procesar una petición de expulsión de un CC puede procesar una petición PtIm y después una petición Pt. La primera petición induce una petición PtObE desde el Cm al CC1. Al finalizar la transacción el CC1 no está en el VP. El CM al finalizar el procesado de la

^{39.} Memoria ha sido actualizada al responder el CC a la petición PtObL.

petición Pt de CC3 establece como estado del bloque en el directorio el estado L. En consecuencia, cuando el CM procesa la petición de expulsión del CC1, el VP no contiene el identificador del CC1 y además, la petición no es esperada.



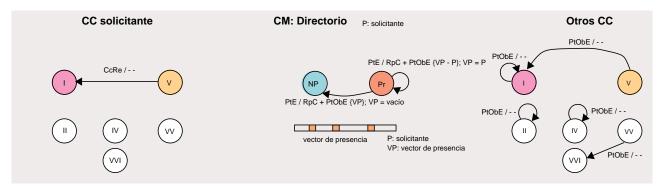
Protocolo de directorio VI. Expulsión silenciosa

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. 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 invalidación.

Pregunta 1: Suponga que pueden existir accesos concurrentes a memoria. Para un protocolo de directorio VI, diseñe los autómatas de cambio de estado de un bloque en el CC y en el CM cuando se utiliza expulsión silenciosa.

Respuesta: En la siguiente figura se muestra el cambio de estado de un bloque cuando una expulsión de un bloque de cache es silenciosa. Notemos que en el autómata de estados de un bloque en el CC no se utiliza el estado VI. La expulsión no se notifica al CM y en consecuencia el vector de presencia no se actualiza. Por tanto, no se detectan cruces en el CM. Recordemos que, en este protocolo, los cruces son debidos a expulsiones.

En la figura también se muestran las transiciones, de un bloque en el directorio, en las que se emite una petición PtObE.



Al ser la expulsión silenciosa, un CC puede recibir una petición de observación de escritura (PtObE) en cualquier estado, ya sea estable o transitorio. Entonces, hay que analizar estado por estado las consecuencias de una observación de escritura emitida por el CM.

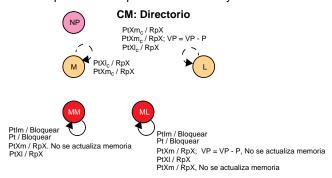
Las transiciones desde los estados estables V y VV han sido descritas al suponer el directorio preciso.

En el estado VVI un CC no puede recibir una petición PtObE, ya que el CM ha extraído al CC del VP al emitir la petición PtObE, que ha determinado la transición del estado VV al estado VVI.

Las transiciones en los estados I, II, IV al recibir una petición PtObE del CM son autotransiciones. La primera se puede producir después de expulsar el bloque. Las dos siguientes se pueden producir después de la expulsión del bloque y una posterior referencia al mismo bloque por parte del procesador. En el caso del estado II es una instrucción store y en el caso del estado IV es una instrucción load.

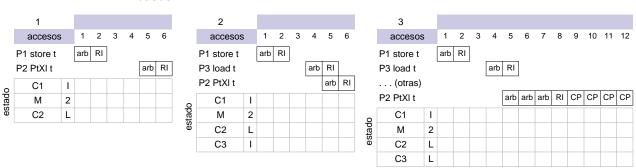
Protocolo MLI: diagramas temporales con cruces de peticiones

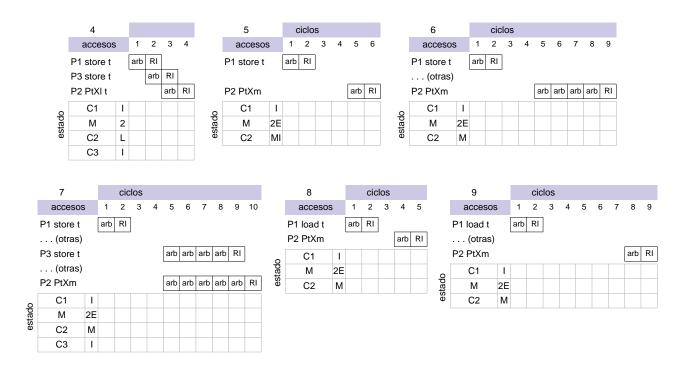
Utilice el protocolo de directorio MLI descrito en este capítulo. El multiprocesador dispone de un CM. Para los cruces de peticiones en el CM utilice el siguiente diagrama de transiciones entre estados. En los estados transitorios MM y ML se responde a las peticiones PtXm y PtXI.



En los siguientes diagramas temporales se muestra el estado estable en cache antes de emitirse la petición y el instante en que se obtiene el acceso a la RI. Cuando una petición está varios ciclos en la fase arb indica que hay otros CC que están encolando peticiones en la CP. Cuando se indica la fase CP, el CM está procesando peticiones de otros CC, a bloques distintos, que han sido emitidas previamente o concurrentemente.

Pregunta 1: Complete el diagrama temporal en cada uno de los siguientes casos.

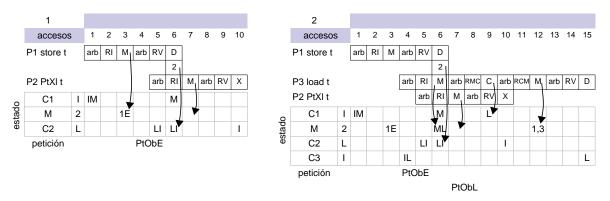




Respuesta:

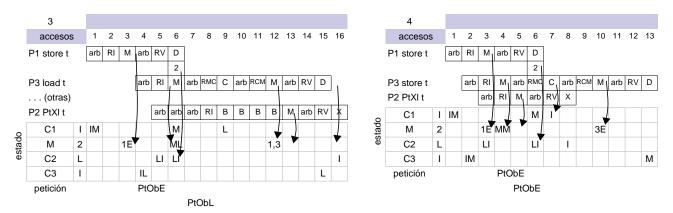
1º diagrama temporal. La petición PtXI de CC2 se procesa en el CM en el estado M. Es el instante más tardío, respecto a la petición de CC1, en la cual CC2 puede emitir la petición PtXI para que se produzca un cruce.

2º diagrama temporal. La petición de PtXI de CC2 se procesa en el CM en el estado ML.



3º diagrama temporal. La petición PtXI de CC2 se procesa en el CM en el estado L.

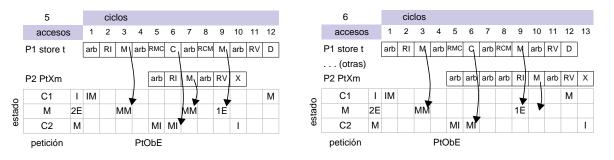
4º diagrama temporal. El procesado de la petición en el CM es en el estado MM.



En los cuatro diagrama temporales previos, cuando se infiere el cruce, el estado del bloque en el CC2 es LI.

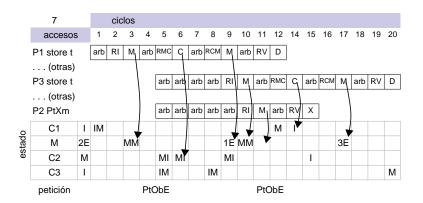
5º diagrama temporal. La petición PtXIm de CC2 se procesa en el CM en el estado MM.

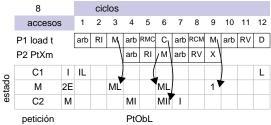
6º diagrama temporal. El procesado de la petición en el CM es en el estado M. En el CC2 el cruce se infiere en el estado MI.



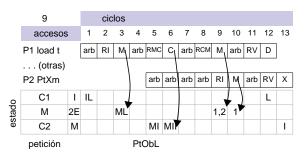
7º diagrama temporal. La petición PtXIm de CC2 se procesa en el CM en el estado MM.

8º diagrama temporal. El procesado de la petición en el CM es en el estado ML. En el CC2 el cruce se infiere en el estado MI.





9º diagrama temporal. La petición PtXIm de CC2 se procesa en el CM en el estado L. En el CC2 el cruce se infiere en el estado MI.

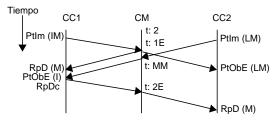


Protocolo de directorio MLI. Diagramas temporales simplificados

Pregunta 1: Represente mediante un diagrama temporal simplificado la siguiente secuencia de accesos, cuyo orden de procesado en el CM se muestra.

Orden de procesado en el CM
P2 store t
P1 store t

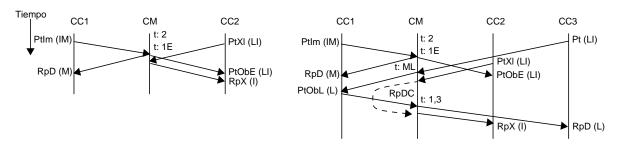
Respuesta: En la figura se muestra el diagrama correspondiente.



Pregunta 2: Represente mediante un diagrama temporal simplificado las siguientes secuencias de accesos, cuyo orden de procesado en el CM se muestra.

Orden de procesado en el CM		
Α	В	
P2 store t	P2 store t	
P1 PtXI t	P3 load t	
	P1 PtXI t	

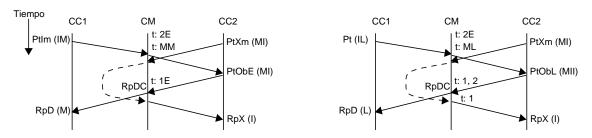
Respuesta: En la parte izquierda de la figura se muestra el diagrama correspondiente a la secuencia A. El diagrama de la parte derecha corresponde a la secuencia B.



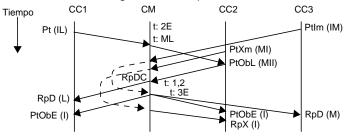
Pregunta 3: Represente mediante un diagrama temporal simplificado las siguientes secuencias de accesos, cuyo orden de procesado en el CM se muestra.

Orden de procesado en el CM					
С	D	E			
P1 store t	P1 load t	P3 load t			
P2 PtXm t	P2 PtXm t	P1 store t			
		P2 PtXm t			

Respuesta: En la parte izquierda de la figura se muestra el diagrama correspondiente a la secuencia C. El diagrama de la parte derecha corresponde a la secuencia D.



En la figura se muestra el diagrama correspondiente a la secuencia E.



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	Controlado	ontrolador de cache (CC) Controlador de men		noria (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:

		ciclos		
mensajes	1	2	3	
Pt, PtE, PtX	arb	RI	М	
RpD, RpC, RpX	arb	RV	DóX	
PtObE	arb	RMC	Сх	

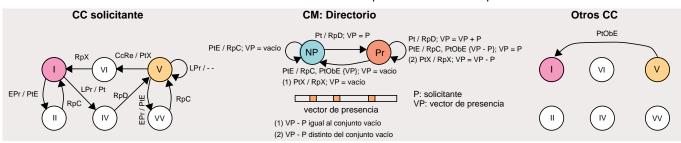
arb: arbitraje en la red correspondiente
RI: red de peticiones desde los CC al CM
RV: red de respuestas desde el CM a los CC
RMC: red de peticiones desde el CM a los CC

M: memoria (directorio)
D: dato (RpD)
X: confirmación (RpX)

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 cuando no se consideran posibles cruces de peticiones.



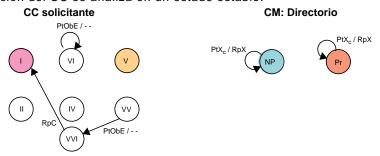
En este multiprocesador supondremos que sólo existe un acceso a memoria en un instante determinado.

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 vector de presencia utilizando el ordinal de los procesadores cuyas caches tienen copia del bloque.
DóX	Se especifica el estado estable del bloque en la cache cuyo CC ha efectuado la petición.
Х	Se especifica la modificación de estado determinada por la petición del CM
Transacción de 2 pasos con petiiciones 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.

Al tener en cuenta la concurrencia de peticiones de los CC hay que considerar los posibles cruces de peticiones en el CM y en los CC. En el CM se procesan peticiones a bloques en estados estables.

En la siguiente figura se muestra la gestión de peticiones en los cruces. En el diagrama de estados de un bloque en el directorio, se utiliza el subíndice c para identificar una petición del CC que se ha cruzado con una petición del CM y la petición del CC se analiza en un estado estable.



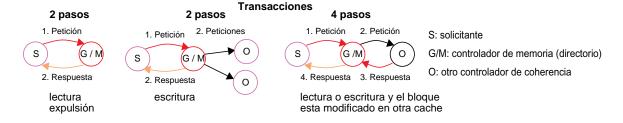
Descripción de un protocolo de directorio MLI denominado A

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 y un bit de exclusividad por bloque. El vector de presencia (VP) es un vector de bits, con tantos bits como procesadores y cada bit está asociado a un procesador. El bit de exclusividad (BE) 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	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 PtIm	PtObE: petición de observación de escritura, inducida por una petición PtIm	RpDc: respuesta con el boque 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					

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 y si es el caso, una actualización de memoria con el bloque expulsado, si éste ha sido modificado durante su estancia en la cache. En una petición PtXm se actualiza el directorio y memoria, mientras que en una petición PtXI sólo se actualiza el directorio.

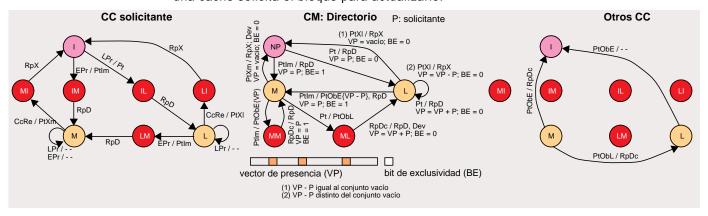
Cuando el servicio de un acceso a memoria requiere un reemplazo, éste se efectúa antes de gestionar el acceso a memoria que produce la acción de reemplazo.

Las fases de cada uno de los mensajes son:

	ciclos		los		
mensajes	1	2	3	arb: arbitraje en la red correspondiente	M: memoria (directorio)
Pt, Ptlm, PtXm, PXI	arb	RI	М	RI: red de peticiones desde los CC al CM	D: dato (RpD)
RpD, RpX	arb	RV	DóX	RV: red de respuestas desde el CM a los CC	X: confirmación (RpX)
PtObE, PtObL	arb	RMC	Сх	RMC: red de peticiones desde el CM a los CC	Cx: cache, donde x es el ordinal de
RpDc	arb	RCM	М	RCM: red de respuestas de los CC al CM	la cache

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 (IL, LM, IM, LI, MI). En el CM para distinguir entre la emisión de un mensaje de petición, correspondiente a una transacción que está procesando el CM, y la respuesta de un CC se utilizan estados transitorios (ML, MM).

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, cuando no se consideran posibles cruces de peticiones. En el protocolo que se describe, el bit de exclusividad del directorio se activa cuando una cache solicita el bloque para actualizarlo.

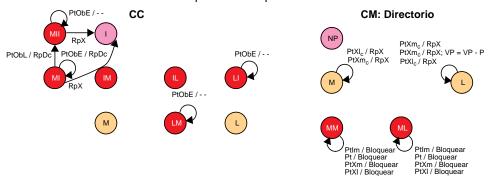


En un diagrama temporal se muestran en la parte superior las fases de los mensajes de una transacción, en la parte central se especifica el estado de los bloques en las cache y en el directorio. En la parte inferior se etiqueta el mensaje o la respuesta que se representa en la parte superior. El estado 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 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.
DóX	Se especifica el estado estable del bloque en la cache cuyo CC ha efectuado la petición.
С	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 PtXl) y en la segunda fila la petición que determina la expulsión.
Mensaje	Se indica en la columna correspondiente a arb.

Al tener en cuenta la concurrencia de peticiones de los CC hay que considerar los posibles cruces de peticiones en el CM y en los CC. En el CM sólo se procesan peticiones a bloques en estados estables. Una petición en la cabeza de la cola de peticiones (CP) que accede a un bloque en un estado transitorio determina un bloqueo del análisis de esta petición y las que le siguen en al CP. El bloqueo en la CP se indica en un diagrama temporal mediante el acrónimo B. La espera en la CP se indica mediante el acrónimo CP.

En la siguiente figura se muestra la gestión de peticiones en los cruces. En el diagrama de estados de un bloque en el directorio, se utiliza el subíndice c para identificar una petición del CC que se ha cruzado con una petición del CM y la petición del CC se analiza en un estado estable. Notemos que en un cruce no se actualiza la memoria al procesar la petición PtXm.



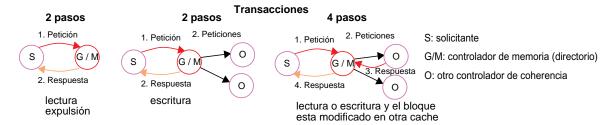
Descripción de un protocolo de directorio MLI denominado B

En un multiprocesador las caches privadas son de mapeo directo y utilizan escritura retardada. Las redes de interconexión entre las caches y los módulos de memoria (CM) son de tipo crossbar. Todos los mensaje generados por un CM, en un paso intermedio del procesado de una transacción o para finalizar una transacción, se emiten en el mismo ciclo a todos los CC involucrados en la acción de coherencia. 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 2 bits por bloque. Con estos bits se codifican 4 estados (NP, L1, L y M). El estado NP indica que no hay copias del bloque en las caches. El estado L1 indica que hay copia del bloque en sólo una cache y se ha solicitado para leer. El estado L indica que hay copia del bloque en varias caches y se ha solicitado para leer. El estado M indica que una cache tiene copia del bloque y el bloque ha sido solicitado para actualizarlo. Notemos que desde el estado L no se puede pasar al estado L1 ya que no se identifican explícitamente las caches que tienen copia y tampoco se dispone de un contador para conocer el número de copias. Como no se identifican las caches que tienen copia, las acciones del protocolo deben ser conservadoras.

Las secuencias de mensajes de las transacciones son las siguientes:



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

Procesador	Controlador	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	Respuestas del CC al CM	Acciones	
LPr : lectura	Pt : petición de bloque	RpD: respuesta con el bloque a una petición Pt, PtIm o PtI	PtObE: petición de observación de escritura, inducida por una petición PtIm o PtI	RpDc: respuesta con el boque 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	
	PtI: petición de exclusividad (bloque en estado L)					
	PtXm: petición de expulsión de un bloque en estado M					
	PtXI: petición (notificación) de expulsión de un bloque en estado L					

Un CC también efectúa acciones de reemplazo cuando es necesario (CcRe). En una acción de reemplazo se distingue la acción de notificación al directorio (bloque en estado L) y esta acción y la acción, adicional, de actualización de memoria (bloque en estado M). En una petición PtXm se actualiza el directorio y memoria, mientras que en una petición PtXI sólo 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:

		cic	los
mensajes	1	2	3
Pt, PtIm, PtI, PtXm, PXI	arb	RI	М
RpD, RpX	arb	RV	DóX
PtObE, PtObL	arb	RMC	Т
RpDc	arb	RCM	М

arb: arbitraje en la red correspondiente
RI: red de peticiones desde los CC a los CM
RV: red de respuestas desde los CM a los CC
RMC: red de peticiones desde los CM a los CC
RCM: red de respuestas de los CC a los CM

M: memoria (directorio)

D: dato (RpD)

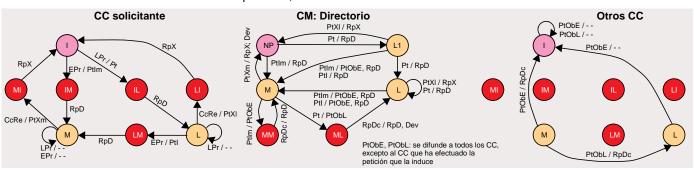
X: confirmación (RpX)

T: difusión de una petición del CM a todas las caches, excepto, si es el caso, a la cache que ha efectuado la petición

Las redes RMC (peticiones) y RV (respuestas) son la misma red física. Se utilizan acrónimos distintos para identificar el tipo de mensaje. Los enlaces puntos a punto mantienen el orden de los mensajes transmitidos. La gestión de las colas de mensajes de los CM y CC es FIFO.

En un CC se utilizan estados transitorios (IL, LM, IM, LI, MI) entre la emisión de un mensaje de petición y la recepción de una respuesta. En el CM para distinguir entre la emisión de un mensaje de petición, correspondiente a una transacción que está procesando el CM, y la respuesta de un CC se utilizan estados transitorios (ML, MM).

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 cuando no se consideran posibles cruces de peticiones. Recordemos que no se identifican explícitamente las caches que tiene copia. Una petición PtObE o PtObL de un CM se difunde a todos los CC, excepto al CC que ha efectuado la petición, la cual ha inducido al CM a emitirla.



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.

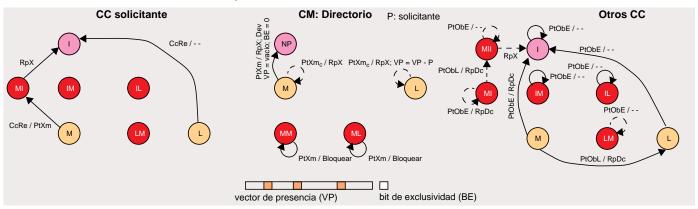
Fases y eventos	Especificaciones
arb	Se especifica el estado transitorio del bloque en la cache.
М	Se especifica el estado, ya sea transitorio o estable, del bloque en el directorio. Notemos que no se pueden indicar los ordinales de las caches que tienen copia, ya que no se conocen.
DóX	Se especifica el estado estable del bloque en la cache cuyo CC ha efectuado la petición.
Т	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 la difusión de las peticiones del CM, si es el caso. La difusión de una petición PtObE o PtObL se indica mediante el acrónimo T (indica todas las caches excluida la que efectúa la petición que la induce).
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 PtXl) y en la segunda fila la petición que determina la expulsión.
Mensaje	Se indica en la columna correspondiente a la fase arb.

Ejercicio

7.1

Consideramos el protocolo de directorio MLI denominado A. Este protocolo se ha modificado para disponer de expulsión silenciosa. La expulsión de un bloque en el estado L no se notifica al CM, sólo existe un cambio de estado en el CC en el cual se expulsa el bloque. Por tanto, el VP no se actualiza. Esta característica determina que un CC pueda recibir una petición de observación de escritura (PtObE) de un bloque que no tiene almacenado en cache.

En la siguiente figura se muestran las transiciones en los diagramas de estados relacionadas con la expulsión de un bloque de cache, cuando el directorio no es preciso.

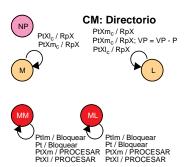


Pregunta 1: Indique secuencias de peticiones concurrentes donde se observen las transiciones añadidas, respecto del caso de un directorio preciso, en el diagrama de estados de un CC. Muestre las transiciones entre estados utilizando diagramas temporales.

Ejercicio

7.2

Consideramos el protocolo de directorio MLI denominado A. El objetivo es reducir el tiempo de ocupación del CM para incrementar el número de transacciones procesadas por unidad de tiempo. Para ello se analiza si el CM puede responder a peticiones de expulsión (PtXm, PtXI) en estados transitorios (MM, ML). En estas condiciones, el CM sólo se bloquearía al analizar peticiones Pt y PtIm al mismo bloque en un estado transitorio.



Para que el CM esté en un estado transitorio (MM, ML) es necesario que al procesar una petición haya solicitado el bloque a una cache que lo tiene en exclusividad.

En el caso de una petición PtXm, la cache que tiene actualmente el bloque en exclusividad puede ser la que emite la petición de expulsión del bloque o ser una tercera cache. Las siguientes secuencia de accesos muestran los distintos casos. Las dos primeras secuencia de accesos, empezando por la izquierda, corresponden al caso en el cual la cache, que tiene el bloque en exclusividad, es la que expulsa el bloque. En las dos siguientes secuencias de accesos, la cache que tiene actualmente el bloque en exclusividad es distinta de la cache que emite la petición de expulsión del bloque.

			•					
	accesos	Α	accesos	В	accesos	С	accesos	D
	P1 store P2 PtXm	-	P1 load t P2 PtXm		P1 store P3 store P2 PtXm	t	P1 store P3 load t P2 PtXm	į
					P2 PtXm	τ	P2 PtXm	ι τ
0	C1	1	C1	1	C1	1	C1	- 1
estado	М	2E	М	2E	М	2E	М	2E
es	C2	М	C2	M	C2	М	C2	М
					C3	1	C3	I

En el caso de una petición PtXI debe haber involucrada una tercera cache, la cual ha solicitado el bloque en exclusividad. En las siguientes secuencias de accesos hay una petición de una tercera cache que solicita el bloque en exclusividad.

	accesos E			accesos F		
	P1 store t			P1 store t		
	P3 store t			P3 load t		
	P2 PtXI t			P2 PtXI t		
0	C1	1		C1	ı	
estado	М	2		М	2	
es	C2	L		C2	L	
	C3	Ι		C3	ı	

El contenido del VP se actualiza al pasar de un estado transitorio al estado estable.

Pregunta 1: Analice si es factible que el CM responda en el estado ML cuando procesa una petición PtXI o una petición PtXm. Para ello, seleccione, de las secuencias de accesos mostradas previamente, las secuencias que sean de utilidad. Muestre en un diagrama temporal la evolución de los estados en el CM y los CC. Determine la información en el VP al procesarse la expulsión e indique si hay que actualizar la memoria, si es el caso. Justifique la respuesta teniendo en cuenta el comportamiento de algún o algunos componentes del multiprocesador.

Pregunta 2: Analice si es factible que el CM responda en el estado MM cuando procesa una petición PtXI o una petición PtXm. Para ello, seleccione, de las secuencias de accesos mostradas previamente, las secuencias que sean de utilidad. Muestre en un diagrama temporal la evolución de los estados en el CM y los CC. Determine la información en el VP al procesarse la expulsión e indique si hay que actualizar la memoria, si es el caso. Justifique la respuesta teniendo en cuenta el comportamiento de algún o algunos componentes del multiprocesador.

Pregunta 3: Resuma, utilizando la siguiente tabla, las condiciones de inferencia de un cruce en el CM, cuando un bloque está en un estado transitorio. Indique la respuesta, el cambio de estado y la actualización del VP, si es el caso.

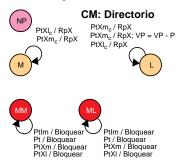
		PtXI	PtXm
ML	P ∉ VP		
IVIL	P∈ VP		
ММ	P ∉ VP		
	P∈ VP		

Ejercicio

7.3

Consideramos el protocolo de directorio MLI denominado A. La idea es aprovechar la inferencia de un cruce de peticiones para reducir el tráfico de mensajes. En concreto, se quiere aprovechar la petición PtObE del CM, cuando se cruza con una petición PtXm o PtXI de un CC, para utilizarla como respuesta en el CC. El CC responde a la petición PtObEsi es el caso. En estas condiciones el CM, al inferir el cruce, no responde a una petición PtXm o PtXI. Mediante esta actuación, el CC está menos tiempo bloqueado y además el CM

estará menos tiempo ocupado. Tenga en cuenta que en el multiprocesador utilizado la RV mantiene el orden de los mensajes emitidos desde el CM e implementa las redes lógicas RCM y RV.



El CM bloquea el análisis de la cola CP cuando la petición en la cabeza de la misma accede a un bloque en un estado transitorio.

El contenido del VP se actualiza al pasar de un estado transitorio al estado estable.

Un bloque en el estado LI en el CC2 puede recibir una petición PtObE. Esta petición ha sido emitida por el CM al procesar una petición PtIm de otro CC1, antes de procesar la petición PtXI del CC2. En el CC2 la petición PtObE se utiliza como respuesta del CM a la petición PtXI. El CM no responde a la petición PtXI cuando infiere el cruce de peticiones.

En el CM, la petición PtXI puede procesarse en el estado M o en el estado L, en función del entrelazado de accesos al bloque. Las siguientes secuencias de accesos permiten analizar los distintos casos.

	accesos A		accesos B		
	P1 store t		P1 store t		
	P2 PtXI t		P3 load t		
			P2 PtXI t		
0	C1	1	C1	1	
estado	М	2	М	2	
es	C2	L	C2	L	
			С3	1	

Pregunta 1: Analice si es factible que el CC utilice la petición PtObE del CM, para un bloque en el estado LI, como respuesta del CM a una petición PtXI. Para ello, utilice las secuencias de accesos previas. Muestre en un diagrama temporal la evolución de los estados en el CM y los CC. Determine la información en el VP al procesarse la expulsión e indique si hay que actualizar la memoria, si es el caso. Justifique la respuesta teniendo en cuenta el comportamiento de algún o algunos componentes del multiprocesador.

Un bloque en el estado MI en el CC2 puede recibir una petición PtObE. Esta petición ha sido emitida por el CM al procesar una petición PtIm de otro CC1, antes de procesar la petición PtXm del CC2. En el CC2 la petición PtObE se responde y se utiliza como respuesta del CM a la petición PtXm. El CM no responde a la petición PtXm cuando infiere el cruce de peticiones.

En el CM, la petición PtXm puede procesarse en el estado M o en el estado L, en función del entrelazado de accesos al bloque. Las siguientes secuencias de accesos permiten analizar los distintos casos.

	accesos A			accesos B		
	P1 store t			P1 store t		
	P2 PtXm t			P3 load t		
				P2 PtXI m		
	C1	ı		C1	1	
estado	М	2E		М	2E	
es	C2	М		C2	М	
				C3	1	

Pregunta 2: Analice si es factible que el CC utilice la petición PtObE del CM, para un bloque en el estado LI, como respuesta del CM a la petición PtXm. Para ello, utilice las secuencias de accesos previas. Muestre en un diagrama temporal la evolución de los estados en el CM y los CC. Determine la información en el VP al procesarse la expulsión e indique si hay que actualizar la memoria, si es el caso. Justifique la respuesta teniendo en cuenta el comportamiento de algún o algunos componentes del multiprocesador.

Un bloque en el estado MI en el CC2 puede recibir una petición PtObL. Esta petición ha sido emitida por el CM al procesar una petición Pt de otro CC1, antes de procesar la petición PtXm del CC2. En el CC2 la petición PtObL se responde y se utiliza como respuesta del CM a la petición PtXm. En el CM, la petición PtXm puede procesarse en el estado M o en el estado L, en función del entrelazado de accesos al bloque. El CM no responde a la petición PtXm cuando infiere el cruce de peticiones, Las siguientes secuencias de accesos permiten analizar los distintos casos.

103	casos.				
	accesos A	accesos B			
	P1 load t	P1 load t			
	P2 PtXm t	P3 store t			
			P2 PtXm t		
0	C1	I	C1	1	
estado	М	2E	М	2E	
es	C2	М	C2	М	
			C3	1	

Pregunta 3: Analice si es factible que el CC utilice la petición PtObL del CM para un bloque en el estado MI como respuesta del CM a la petición PtXm. Para ello, utilice las secuencias de accesos previas. Muestre en un diagrama temporal la evolución de los estados en el CM y los CC. Determine la información en el VP al procesarse la expulsión e indique si hay que actualizar la memoria, si es el caso. Justifique la respuesta teniendo en cuenta el comportamiento de algún o algunos componentes del multiprocesador.

El análisis de las secuencias de accesos previas muestra que un CC puede recibir una petición PtObE estando el bloque en el estado I. Este comportamiento indica que el directorio es impreciso.

En estas condiciones hay que analizar si es posible que un CC reciba una petición PtObE estando el bloque en el estado IM o el estado IL. Las siguientes secuencias de accesos permiten analizar los distintos casos.

	accesos A			accesos B		
	P1 load t			P1 load t		
	P3 store t			P3 store t		
	P2 PtXm t			P2 PtXm t		
	P2 store t			P2 load t		
0	C1	1		C1	1	
estado	М	2E		М	2E	
ë	C2	М		C2	М	
				C3	ı	

En estas secuencias se supone que la expulsión del bloque en CC2 ha sido voluntaria. Esto es, no está determinada por un reemplazo. Si estuviera determinada por un reemplazo, se puede suponer que hay varios módulos de memoria y el bloque que determina el reemplazo está almacenado en otro módulo de memoria desocupado. La petición a este bloque es de lectura y se efectúa inmediatamente, ciclo siguiente al de la expulsión. Posteriormente se expulsa este bloque y se vuelve a referenciar el bloque expulsado previamente.

Mientras esté pendiente la expulsión, cualquier acceso al mismo bloque bloquea al procesador. El mensaje no se emite hasta que el bloque expulsado está en el estado I.

Pregunta 4: Analice si es factible que un CC tenga que procesar una petición PtObE estando el bloque en el estado IM o el estado IL. Para ello, utilice las secuencias de accesos previas. Muestre en un diagrama temporal la evolución de los estados en el CM y los CC. Justifique la respuesta teniendo en cuenta el comportamiento de algún o algunos componentes del multiprocesador.

Pregunta 5: Resuma, utilizando la siguiente tabla, las condiciones de inferencia de un cruce en el CM. Indique la respuesta, el cambio de estado y la actualización del VP, si es el caso.

		PtXI	PtXm
	P ∉ VP		
_	$P \in VP$		
М	P ∉ VP		
•••	$P \in VP$		

Un diseñador propone el siguiente mecanismo para mantener el directorio preciso. Cuando el CM emite una petición PtObL a un CC, se mantiene en un campo adicional del VP la identidad del CC que tiene la exclusividad.

Este campo adicional se invalida al procesarse una petición PtXm del CC que identifica.

Este campo adicional se utiliza, si es valido, al procesar una petición PtIm, para excluir al CC identificado de los CC que reciben una petición PtObE. Además se invalida el campo adicional.

Pregunta 6: Analice si el mecanismo propuesto mantiene el directorio preciso. Para ello, seleccione, de las secuencias de accesos previas, las secuencias que sean de utilidad. Muestre en un diagrama temporal la evolución de los estados en el CM y los CC. Determine la información en el VP, especificando el contenido del campo adicional (CA), al procesarse la expulsión e indique si hay que actualizar la memoria, si es el caso. Justifique la respuesta teniendo en cuenta el comportamiento de algún o algunos componentes del multiprocesador.

Ejercicio

7.4

Consideramos el protocolo de directorio MLI denominado A. La idea es aprovechar la inferencia de un cruce de peticiones para reducir el tráfico de mensajes. En concreto, se quieren aprovechar la petición PtObE del CM, cuando se cruza con una petición PtXm o PtXI de un CC, para utilizarla como respuesta en el CC. Esto es, se reduce el tráfico de respuestas. Por otro lado, se quiere disponer de un directorio preciso.

Tenga en cuenta que, en el multiprocesador utilizado, la RV mantiene el orden de los mensajes emitidos desde el CM e implementa las redes lógicas RCM y RV.

El CM bloquea el análisis de la cola CP cuando la petición en la cabeza de la misma accede a un bloque en un estado transitorio.

El contenido del VP se actualiza al pasar de un estado transitorio al estado estable.

Un ingeniero, después de estudiar en detalle los cruces de peticiones propone lo siguiente:

- Cuando se procesa en el CM una petición PtXm, siendo L el estado del bloque en el directorio, y el CC, que ha emitido la petición, está activado en el VP se responde RpX. Además se desactiva al CC del VP.
- Una petición PtObL del CM no se puede utilizar en un CC como respuesta a una petición PtXm del CC. El CC debe esperar una respuesta explícita (RpX) o una petición PtObE del CM.
- El CM no responde a una petición PtXI si el CC no está en el VP.
- Un CC utiliza una petición PtObE del CM como respuesta a una petición PtXm o PtXl. Si es el caso, suministra el bloque.

El ingeniero ha observado que en el CM es necesario discernir si hay que responder o no a una petición PtXm. Para ello, utiliza el estado del bloque en el directorio.

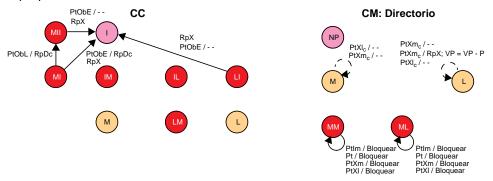
Sea el CC2 el que emite una petición PtXm. Si esta petición se procesa en el CM, siendo M el estado del bloque en el directorio, y el VP conjuntamente con el BE identifican a un CC3, el CM ha ordenado previamente una petición PtIm del CC3.

El CM al procesar la petición PtIm del CC3 emite una petición PtObE al CC2, ya que tiene el bloque en exclusividad (MI). En este caso, el CC2 suministra el bloque y además, utiliza la petición PtObE como respuesta a la petición PtXm.

Por otro lado, el CC2 puede recibir una petición PtObL cuando el bloque está en el estado MI. Para ello, el CM ha procesado una petición Pt de un CC1, antes que la petición PtXm del CC2. El CC2 suministra el bloque al procesar la petición PtObL del CM. En el VP del bloque en el CM se identifica a dos CC que tienen copia del bloque.

En consecuencia es necesario discernir en el CC2 entre una petición PtObL y una petición PtObE. Para ello se utiliza el estado MII. En este estado se espera una respuesta RpX del CM a la petición PtXm o una petición PtObE. La respuesta RpX la emite el CM cuando procesa la petición PtXm del CC2 y el estado del bloque en el directorio es L. En este estado no es esperada una petición PtXm. El CM infiere un cruce, responde al CC2 y lo extrae del VP.

En el siguiente diagrama de transiciones entre estados se muestra la propuesta descrita.



Un bloque en el estado LI en el CC2 puede recibir una petición PtObE. Esta petición ha sido emitida por el CM al procesar una petición PtIm de otro CC1, antes de procesar la petición PtXI del CC2. En el CC2 la petición PtObE se utiliza como respuesta del CM a la petición PtXI. El CM no responde a la petición PtXI cuando infiere el cruce de peticiones.

En el CM, la petición PtXI puede procesarse en el estado M o en el estado L, en función del entrelazado de accesos al bloque. Las siguientes secuencias de accesos permite analizar los distintos casos.

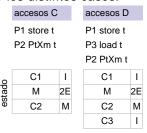
	accesos A			accesos B		
	P1 store t			P1 store t		
	P2 PtXI t			P3 load t		
				P2 PtXI t		
0	C1	1		C1	1	
estado	М	2		М	2	
es	C2	L		C2	L	
				C3	1	

En las siguientes preguntas muestre en un diagrama temporal la evolución de los estados en el CM y los CC. Determine la información en el VP al procesarse la expulsión e indique si hay que actualizar la memoria, si es el caso. Justifique la respuesta teniendo en cuenta el comportamiento de algún o algunos componentes del multiprocesador.

Pregunta 1: Utilice la propuesta del ingeniero para mostrar el flujo de mensajes en las secuencias de accesos previas.

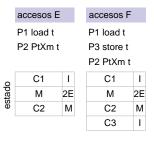
Un bloque en el estado MI en el CC2 puede recibir una petición PtObE. Esta petición ha sido emitida por el CM al procesar una petición PtIm de otro CC1, antes de procesar la petición PtXm del CC2. El CC2 responde a la petición PtObE y la utiliza como respuesta del CM a la petición PtXm. El CM no responde a la petición PtXm cuando infiere el cruce de peticiones.

En el CM, la petición PtXm puede procesarse en el estado M o en el estado L, en función del entrelazado de accesos al bloque. Las siguientes secuencias de accesos permite analizar los distintos casos.



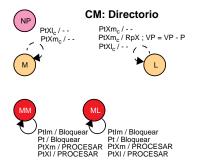
Pregunta 2: Utilice la propuesta del ingeniero para mostrar el flujo de mensajes en las secuencias de accesos previas.

Un bloque en el estado MI en el CC2 puede recibir una petición PtObL. Esta petición ha sido emitida por el CM al procesar una petición Pt de otro CC1, antes de procesar la petición PtXm del CC2. En el CM, la petición PtXm puede procesarse en el estado M o en el estado L, en función del entrelazado de accesos al bloque. Las siguientes secuencias de accesos permite analizar los distintos casos.



Pregunta 3: Utilice la propuesta del ingeniero para mostrar el flujo de mensajes en las secuencias de accesos previas.

Una optimización, que permite reducir la ocupación en del directorio, es que el CM procese los mensajes de expulsión en estados transitorios, además de en estados estables.



Tenga en cuenta que el VP se actualiza cuando el CM responde al solicitante. En un estado transitorio se mantiene la información que habia en el estado estable del que se proviene.

Respecto a las secuencias de accesos que se deben seleccionar tenga en cuenta que un CC tiene que suministrar un bloque.

Pregunta 4: Muestre el procesado de una petición PtXI cuando el bloque está en el estado ML en el directorio.

Pregunta 5: Muestre el procesado de una petición PtXm cuando el bloque está en el estado MM en el directorio.

Pregunta 6: Muestre el procesado de una petición PtXm cuando el bloque está en el estado ML en el directorio.

Ejercicio

7.5 Consideramos el protocolo de directorio VI denominado A.

Pregunta 1: Represente mediante un diagrama temporal simplificado las secuencias de acceso mostradas en la Figura 7.18 y en la Figura 7.19.

Ejercicio

7.6 Un multiprocesador utiliza un esquema de directorio con un protocolo de coherencia con invalidación (MLI) denominado B.

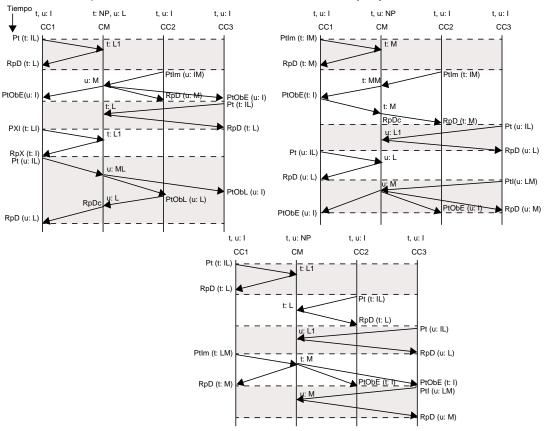
Supondremos un arbitraje por antigüedad en la red de peticiones de los CC al CM y que la cola de peticiones se gestiona de forma FIFO.

Suponga las siguientes secuencias de accesos a memoria. Las variables t y u están ubicadas en bloques distintos.

A. accesos	1. P1 load t	2. P2 store u	3. P3 load t	4. P1 PtXI t	5. P1 load u
B. accesos	1. P1 store t	2. P2 store t	3. P3 load u	4. P1 load u	5. P3 store u
C. accesos	1. P1 load t	2. P2 load t	3. P3 load u	4. P1 store t	5. P3 store u

En la secuencia A otras caches tienen copia de u. En todas las secuencias, los bloques que contienen las variables t y u se almacenan en el mismo contenedor de cache.

Pregunta 1: En los siguientes diagramas temporales simplificados, indique la transacción o la ausencia de la misma que no cumple el protocolo. Muestre para esta transacción la secuencia de mensajes y estados correcta.



Podemos decir que el directorio es impreciso. Entonces, un diseñador propone que un CC no notifique al directorio un reemplazo de un bloque en estado L.

Suponga la siguiente secuencia de accesos a memoria. Las caches sólo disponen de un contenedor. Los variables u y t están ubicadas en bloques distintos. Al empezar la secuencia de accesos no hay copia de los bloques en las caches.

D. accesos 1. P1 load t 2. P1 load u 3. P1 load t 4. P1 store t	
---	--

Pregunta 2: Utilice la anterior secuencia de accesos a memoria para mostrar que la propuesta del diseñador es desacertada, aunque el protocolo es correcto. Indique el estado de los bloques en la cache y en el directorio una vez el CC y el CM han efectuado todas las acciones necesarias para servir cada acceso a memoria (tenga en cuenta los reemplazos, si es el caso).

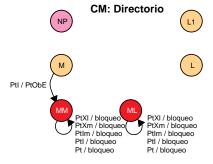
Hay notificación al expulsar un bloque en	Cache		Directorio		Peticiones del CM a los CC
estado L	t	u	t	u	
1. P1 load t					
2					

No hay notificación al expulsar un bloque en	Cach	е	Direct	torio	Peticiones del CM a los CC		
estado L	t	u	t	u			
1. P1 load t							
2							

Suponga que en el directorio el estado del bloque que contiene la variable t es L, debido a que caches distintas de C1, C2 y C3 tiene copia del bloque. Las siguientes secuencias de acceso son concurrentes.

E. accesos	1. P2 store t	2. P3 load t	3. P1 load t	Orden de procesado en el CM
F. accesos	1. P2 store t	2. P3 load t	3. P1 store t	

En la siguiente pregunta se analizan cruces de peticiones en los CC. El directorio bloquea el procesado de cualquier petición en un estado transitorio. Estas transiciones entre estados no están especificadas en la descripción del protocolo. Por otro lado, la recepción en el CM de una petición PtI, con el bloque en el estado M, se gestiona de la misma forma que una petición PtIm (el directorio infiere que se ha producido un cruce).



Pregunta 3: Muestre en un diagrama temporal los cruces de peticiones que se producen en las secuencias E y F. Indique el estado del bloque en las caches cuando se produce el cruce y la respuesta que debe emitir el CC correspondiente, si es el caso.

Pregunta 4: Proponga una secuencia de accesos a memoria donde se observe un cruce de peticiones en el estado LM de un bloque en cache.

Suponga las siguientes secuencias de accesos a memoria. En la dos secuencia el estado del bloque, que contiene la variable t, en la C1 es L y no hay copias en las otras caches. El estado del bloque en el directorio es L, debido a que la cache C3 ha tenido el bloque en el estado L y lo ha expulsado.

EE. accesos	1. P2 store t	2. P3 store t	3. P1 store t	Orden de procesado en el CM
FF. accesos	1. P2 store t	2. P3 load t	3. P1 store t	

Pregunta 5: Muestre en un diagrama temporal simplificado los cruces de peticiones que se producen en las secuencias EE y FF. Indique el estado del bloque en el directorio cuando se produce el cruce y la acción que efectúa el CM

En la siguiente pregunta supondremos que el arbitraje en la red de peticiones no es por antigüedad o la gestión de la cola de peticiones no es FIFO.

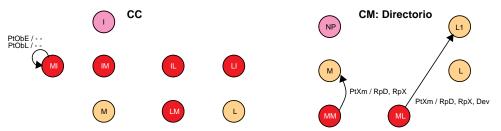
Supongamos la siguiente secuencia de accesos a memoria. El estado del bloque en el directorio es L1 y en la cache P1 es L.

LL. accesos	1. P2 store t	2. P2 PtXm t	3. P3 load t	4. P1 store t	Orden de procesado en el CM	

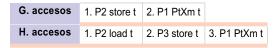
Pregunta 6: Para la secuencia previa de accesos a memoria, muestre un diagrama temporal simplificado. Indique si el protocolo funciona correctamente. Razone la respuesta.

En las siguientes preguntas supondremos un arbitraje por antigüedad en la red de peticiones de los CC al CM y que la cola de peticiones se gestiona de forma FIFO.

Para gestionar un cruce de peticiones, siendo una de ellas una petición PtXm un ingeniero propone que esta petición se transmita por la red RCM (respuestas) y el CM la gestione, cuando está en un estado transitorio, como una respuesta a su petición. En los siguientes diagramas de transiciones entre estados se muestra el cruce.



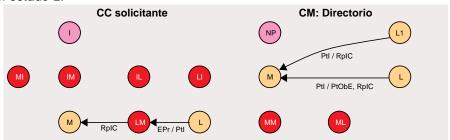
Para analizar los cruces en estados transitorios se utilizan las siguientes dos secuencias de accesos concurrentes a memoria.



Pregunta 7: Dados los siguientes diagramas temporales simplificados, etiquete el estado del bloque en las caches y el CM y añada los mensajes (etiquetandolos) que faltan en el orden temporal adecuado para que finalicen las transacciones iniciadas.



Un ingeniero propone añadir una nueva respuesta del CM, que se utiliza para responder a una petición PtI. Esta respuesta, cuyo acrónimo es RpIC, no contiene el bloque, ya que un CC utiliza la petición PtI cuando tiene el bloque en estado L.

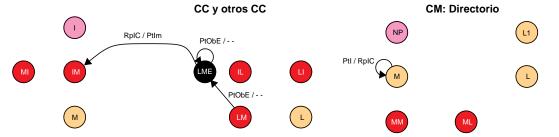


Al analizar cruces de peticiones hay que tener en cuenta, entre otros, el cruce de una petición Ptl con una petición PtObE del CM, inducida por una petición PtIm o Ptl de otro CC.

La gestión de los cruces se efectúa básicamente en el CC con la colaboración del CM. El CM al procesar una petición Ptl en el estado M, infiere un cruce y responde RpIC al CC que ha emitido la petición. El estado no se modifica. El CC inferirá, al recibir está respuesta, en función del estado del bloque en la cache, que la petición Ptl ha sido rechazada por el CM.

Un CC, antes de recibir la repuesta a una petición PtI, puede observar una petición PtObE, la cual invalida su copia del bloque. El CC al recibir la respuesta RpIC infiere un cruce de peticiones y solicita el bloque con intención de modificación (PtIm). Para identificar el cruce se utiliza un estado transitorio (LME).

El CC al recibir RpIC, en el estado LME, infiere que el estado del bloque en el directorio es M.



Suponga la siguiente secuencia de accesos a memoria. El estado del bloque en el directorio es L1 y en la cache C1 es L. En las otras caches el estado es I.

I. accesos 1. P2 store t 2. P1 store t Orden de procesado en el CM

Pregunta 8: Para la secuencia previa de accesos a memoria, muestre un diagrama temporal simplificado.

Por otro lado, un CC, después de observar una petición PtObE en el estado LM, puede observar una petición PtObL, antes de recibir la respuesta RpIC a su petición PtI. En estas condiciones el CC infiere que el bloque está en el estado L en el directorio. El CC tiene el bloque invalidado (PtObE) y por tanto, al recibir RpIC emite una petición PtIm.

Por su parte, el CM al procesar la petición PtI, en el estado L, ha establecido como estado del bloque en el directorio M y ha emitido peticiones PtObE a todos los CC, excluyendo al CC que ha emitido PtI. Por tanto, el CM presupone que el CC que ha emitido PtI tiene el bloque en exclusividad; lo cual no es cierto. Sea CC1 el CC que ha emitido PtI y posteriormente emite PtIm.

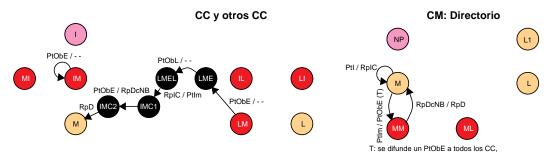
Para simplificar, por ahora, suponemos que el CM procesa la petición PtIm del CC1, antes que peticiones de otros CC (Pt, PtIm, PtI).

Cuando el CM procesa la petición PtIm del CC1 difunde una petición PtObE a todos los CC, excluyendo al CC1. Ninguno de los CC que reciben la petición responde, ya que tienen el bloque en el estado I. En consecuencia se produce un abrazo mortal.

Para soslayar la situación descrita, una alternativa es que el CM al procesar una petición PtIm en el estado M difunda una petición PtObE a todos los CC; sin excluir a ninguno (es una modificación respecto del protocolo base). Entonces, el CC1 responde RpDcNB (nueva respuesta), indicando que no tiene el bloque.

El CM en la situación descrita (estado MM) está esperando una respuesta (RpDc, RpDcNB). Cuando el CM recibe como respuesta RpDcNB infiere que tiene una copia válida del bloque y responde al CC cuya petición está procesando.

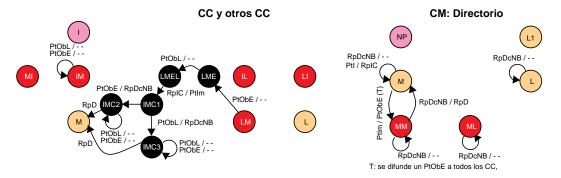
Para efectuar el procesado descrito se utilizan tres estados transitorios adicionales en un CC.



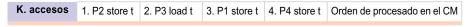
Suponga la siguiente secuencia de accesos a memoria. El estado del bloque en el directorio es L1 y en la cache C1 es L. En las otras caches el estado es I.

Pregunta 9: Para la secuencia previa de accesos a memoria, muestre un diagrama temporal simplificado.

El CM, antes de procesar la petición PtIm del CC1, puede procesar peticiones de otros CC (PtIm, Pt, PtI). Mediante la respuesta RpDcNB, el CM infiere que tiene una copia válida del bloque. Entonces, el CC que ha recibido RpIC, también debe responder RpDcNB a una petición PtObL, cuando ha detectado un cruce.

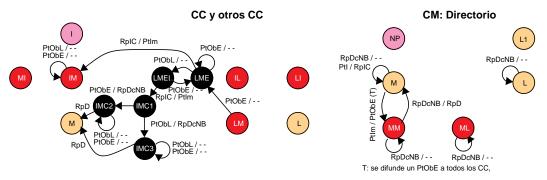


Suponga la siguiente secuencia de accesos a memoria. El estado del bloque en el directorio es L1 y en la cache C1 es L. En las otras caches el estado es I.



Pregunta 10: Para la secuencia previa de accesos a memoria, muestre un diagrama temporal simplificado. Note que algunos CC generan peticiones que no están explicitas en la anterior secuencia. Estas peticiones se procesan en el CM en el orden de llegada. Llegan al CM después de que las especificas en la secuencia de accesos hayan sido emitidas y encoladas en la cola de peticiones.

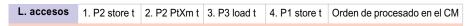
Una vez un CC ha emitido un petición PtI, mientras no haya recibido una respuesta RpIC, debe de estar infiriendo el estado del bloque en el directorio (M, L), ya que el CM puede servir peticiones de otros CC. Para ello, hay que disponer de reconocedores de secuencias en el diagrama de estado de un CC. En la siguiente figura se muestra el diseño efectuado por un ingeniero.



Pregunta 11: Indique las secuencias que se reconocen (expresiones regulares) antes de que un CC reciba una respuesta RpIC, después de haber emitido una petición PtI.

Ahora supondremos que el arbitraje en la red de peticiones no es por antigüedad o la gestión de la cola de peticiones no es FIFO.

Supongamos la siguiente secuencia de accesos a memoria. El estado del bloque en el directorio es L1 y en la cache P1 es L.



Pregunta 12: Para la secuencia previa de accesos a memoria, muestre un diagrama temporal simplificado. Indique si el protocolo funciona correctamente. Razone la respuesta.

Pregunta 13: Un ingeniero propone que un CC notifique la expulsión de un bloque en estado L, pero que no espere respuesta. Indique de forma justificada si la propuesta mantiene la coherencia.

Ejercicio

7.7

Un multiprocesador utiliza un esquema de directorio con un protocolo de coherencia con invalidación (VI) denominado A. Suponga la siguiente secuencia de accesos a memoria concurrentes.



Pregunta 1: Muestre en un diagrama temporal los mensajes de las transacciones que se inician y el estado en las caches y en la memoria al ejecutarse el entrelazado de accesos a memoria previo. Indique el número de expulsiones y el número de cruces de peticiones. En cada uno de ellos especifique el grupo de accesos a memoria en el cual se producen.

En las siguientes preguntas utilice una secuencia de dos transacciones concurrentes, iniciadas por CC distintos, para mostrar lo que se solicita. Indique el estado inicial del bloque en las caches y el VP en el directorio y represente las transacciones en un diagrama temporal.

Pregunta 2: En el CM es necesario procesar una petición PtX en el estado Pr y el CC no esté identificado en el VP.

Pregunta 3: En el CM es necesario procesar una petición PtX en el estado NP y la petición no es esperada.

El diseñador decide que la expulsión de un bloque de cache no se notifica al directorio, lo cual se denomina expulsión silenciosa. Esto es, el directorio es impreciso.

En un diagrama temporal, el estado del bloque al efectuar una expulsión silenciosa se indica en la fase arb de la transacción que expulsa el bloque.

Para efectuar el análisis de las transiciones entre estados que hay que añadir el diseñador utiliza las siguientes secuencias de accesos a memoria concurrentes. En las tres secuencias hay tres grupos de accesos. En ocasiones, en un grupo sólo hay un acceso a memoria.

Α	В	С
referencia	referencia	referencia
P1 load t	P1 load t	P1 load t
P1 load u	P1 load u	P1 load u
P2 store t	P2 store t	P2 store t
	P1 load t	P1 store t

Las variables t y u están contenidas en bloque distintos.

Estos bloques se almacenan en el mismo contenedor de cache.

En las siguientes preguntas, en los casos en los cuales no exista una transacción para un par petición-estado, en la descripción del protocolo, determine las acciones que deben efectuarse.

Pregunta 4: Utilizando expulsión silenciosa, muestre un diagrama temporal para las secuencias A, B y C.

Pregunta 5: Muestre en el diagrama de estados, correspondiente al agente procesador de un CC, la transición entre estados en una acción de reemplazo. Muestre también en el agente observador de un CC los estados en los cuales se pueden recibir peticiones que son debidas exclusivamente a que el directorio es impreciso. Indique la transición entre estados y la respuesta.

Ejercicio

7.8

Un multiprocesador utiliza un esquema de directorio con un protocolo de coherencia con invalidación (VI) denominado A. En este ejercicio la memoria se construye utilizando dos módulos de memoria (M1 y M2) con el correspondiente directorio.

Suponga la siguiente secuencia de accesos a memoria concurrentes, la cual se corresponde con el esqueleto de una sincronización mediante eventos.

	referencia	М	Valor almacenado en los registros	Módulo de memoria	Valor en memoria	Contenido en cache
F	P1 store R4, A	M2	El contenido de R4 es 9	M1: aviso	A: 7	Los bloques que contienen las variables no están almacenado en ninguna cache
F	P1 store R5, aviso	M1	El contenido de R5 es 1	M2: A	aviso: 0	estan annacenado en miliguria cache
F	P2 load R6, aviso	M1				
F	P2 load R7, A	M2				

		cic	los								
referencia	1	2	3	4	5	6	7	8	9	10	11
P1 store R4, A	arb	RI									
P1 store R5, aviso		arb	RI								
P2 load R6, aviso	arb	arb	arb	RI							
P2 load R7, A											

En un diseño un ingeniero decide que un procesador puede emitir una petición de escritura sin esperar la confirmación de la anterior petición de escritura. Una

petición de lectura espera que finalice el anterior acceso a memoria y también es bloqueante. Debido a conflictos en la RI y acciones del arbitraje, las peticiones de los CC ocupan la RI en los ciclos que se muestran en la figura adjunta.

En un diagrama temporal, en la fase D de una transacción indique el valor leído de memoria. En la fase Mx indique el valor que se ha almacenado en memoria.

Pregunta 1: Complete el diagrama temporal. Indique si se cumple consistencia secuencial. Para ello indique el orden temporal en el cual la secuencia ha actualizado o leído de la memoria (fase M1, M2). Utilice también el contenido de los registros. En la fase M de una transacción utilice el ordinal del módulo de memoria (x) para indicar el módulo accedido (Mx).

El lenguaje máquina dispone de la instrucción denominada barrera. El objetivo de la misma es que no se inicie ningún acceso a memoria, que haya sido especificado posteriormente, hasta que todos los accesos previos a la instrucción barrera hayan consolidado.

Pregunta 2: Indique en las secuencias de instrucciones que se ejecutan en P1 y P2 dónde es necesario incluir instrucciones barrera. El número de inserciones debe ser el mínimo necesario.

Por otro lado, en otro diseño el ingeniero decide que un procesador puede emitir una petición de lectura sin esperar el valor devuelto

referencia
P1 store R4, A
P1 store R5, aviso
P2 load R6, aviso
P2 load R7, A

	cic	los								
1	2	3	4	5	6	7	8	9	10	11
arb	arb	arb	RI							
arb	RI									
	arb	RI								

en la petición de lectura previa. Una petición de escritura espera la finalización del anterior acceso a memoria y es bloqueante. Debido a conflictos en la RI y acciones del arbitraje, las peticiones de los CC ocupan la RI en los ciclos que se muestran en la figura adjunta.

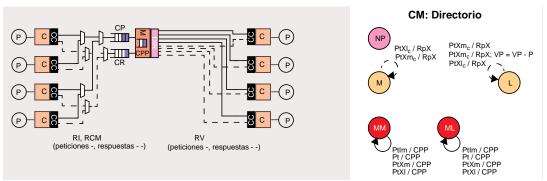
Pregunta 3: Complete el diagrama temporal. Indique si se cumple consistencia secuencial. Para ello indique el orden temporal en el cual la secuencia ha actualizado o leído de la memoria (fase M1, M2). Utilice también el contenido de los registros. En la fase M de una transacción utilice el ordinal del módulo de memoria (x) para indicar el módulo accedido (Mx).

Pregunta 4: Indique en las secuencias de instrucciones que se ejecutan en P1 y P2 dónde es necesario incluir instrucciones barrera. El número de inserciones debe ser el mínimo necesario.

Ejercicio

7.9

Consideramos el protocolo de directorio MLI denominado A. Para no bloquear el procesado de peticiones en el CM, una petición que no puede procesarse se extrae de la CP y se almacena en una cola de peticiones pendientes (CPP). Entonces se sigue analizando la CP hasta encontrar una petición que pueda procesarse.



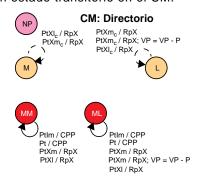
En la CPP se mantiene el orden de llegada de las peticiones. Al finalizar una transacción de 4 pasos el CM analiza la CPP antes que la CP. La CPP se analiza en orden, buscando una petición al mismo bloque que la transacción que acaba de finalizar el procesado.

Pregunta 1: ¿Puede el CM analizar la CP antes que la CPP?

Cuando no se puede procesar una petición, un diseñador propone que en lugar de utilizar la CPP, el CM extraiga la petición de la CP y la vuelve a encolar después de la última entrada válida.

Pregunta 2: Justifique si la decisión sigue garantizando un funcionamiento correcto.

Suponga que el CM responde a las peticiones PtXm y PtXl independientemente del estado en el CM. Esto es, las únicas peticiones que se almacenan en la CPP son peticiones Pt y PtIm que, al ser analizadas por el CM, acceden a un bloque que está en un estado transitorio en el CM.



Pregunta 3: ¿Puede el CM analizar la CP antes que la CPP?

En el contexto de la pregunta previa, un diseñador propone que en lugar de utilizar la CPP, cuando no se puede procesar una petición (Pt, PtIm), el CM extraiga la petición de la CP y la vuelve a encolar después de la última entrada válida.

Pregunta 4: Justifique si la decisión sigue garantizando un funcionamiento correcto.