

## MULTIPROCESADORES

Junio de 2017

**NOTA:** el valor final de un cálculo debe ir acompañado de las unidades, si es el caso. Los valores numéricos deben expresarse en notación científica y con redondeo al número más próximo en las centésimas.

### EJERCICIO 1 ( 1.5 puntos)

Un programa se ejecuta en un procesador de un multiprocesador. Este programa es totalmente paralelizable y se quiere ejecutar utilizando dos procesadores para reducir el consumo energético, manteniendo el mismo rendimiento. En el multiprocesador la frecuencia base de los procesadores es  $f_1$ , pero se puede regular la tensión y la frecuencia de los procesadores en la misma proporción  $\beta$ . Suponga que el CPI no se modifica y al utilizar dos procesadores la capacidad efectiva equivalente es el doble.

**Pregunta a:** Deduzca la frecuencia de funcionamiento de los procesadores del multiprocesador.

**Pregunta b:** Deduzca la reducción de potencia y la reducción de energía.

La tensión de alimentación en los procesadores del multiprocesadores se puede reducir como máximo el 30%, ya que por debajo de este valor los procesadores dejan de funcionar correctamente. Suponga que en otro programa una parte debe ejecutarse en serie. Este programa se quiere ejecutar con dos procesadores manteniendo el mismo rendimiento que con un procesador.

**Pregunta c:** Deduzca el porcentaje mínimo de paralelismo en el programa para obtener el mismo rendimiento sin superar el límite de reducción de tensión. La tensión de alimentación es constante durante toda la ejecución.

### EJERCICIO 2 ( 1.5 puntos)

Los multiprocesadores que se consideran en este ejercicio disponen de ocho procesadores con cache privadas. Cada cache tiene 1024 conjuntos y cada conjunto tiene asociatividad cuatro.

Suponga que un multiprocesador utiliza como red de interconexión un bus y un mecanismo de observación para mantener la coherencia de cache. El protocolo de coherencia utiliza la técnica de invalidación y las cache privadas utilizan escritura inmediata, sin asignación de contenedor en caso de fallo en una escritura (VI). En este protocolo un controlador de coherencia utiliza el tipo de petición para determinar las comparaciones que debe realizar en la fase de observación de una transacción.

**Pregunta a:** Indique el número mínimo y máximo de comparaciones que se efectúan en las caches en la fase de observación de una transacción.

**Pregunta b:** Indique el número máximo de las comparaciones, que se efectúan en la fase de observación de una transacción, que pueden detectar coincidencia.

Suponga que otro multiprocesador utiliza redes crossbar para interconectar las cache privadas con el directorio y memoria. Por razones de coste se ha decidido que el vector de presencia sólo tenga un bit. En estas condiciones el vector de presencia sólo codifica si existen copias del bloque (1) o no existen copias del bloque (0), pero no identifica la ubicación precisa de las copias. En consecuencia la implementación de un protocolo debe ser conservadora.

Al implementar el protocolo se efectúa la siguiente actuación: una cache que recibe una petición del directorio, que hace referencia a un bloque que no tiene almacenado, ignora la petición. Tenga en cuenta que para ignorar la petición el controlador de coherencia debe determinar que no tiene el bloque en cache.

Un diseñador propone que las cache privadas utilicen escritura inmediata, sin asignación de contenedor en caso de fallo en una escritura.

**Pregunta c:** ¿Cuántas copias de un bloque puede haber en las caches privadas de los procesadores en un instante determinado?

**Pregunta d:** Indique el número mínimo y máximo de comparaciones, en función del bit en el vector de presencia, que son necesarias en una transacción de coherencia (incluye mensajes de petición y respuesta).

Queremos comparar los dos multiprocesadores desde el punto de vista del número de comparaciones que se efectúan en una transacción. Para ello utilizaremos la siguiente tabla.

petición	porcentaje	vector de presencia (bit)	porcentaje
Pt	70%	0	60%
		1	40%
PtE	30%	0	30%
		1	70%

**Pregunta e:** Calcule para cada protocolo (observación y directorio) el número medio de comparaciones por transacción. En primer lugar indique el número de comparaciones en función del tipo de transacción. En el caso de un protocolo de directorio, además del tipo de transacción, tenga en cuenta el valor del bit en el vector de presencia.

### EJERCICIO 3 ( 3.5 puntos)

El protocolo de coherencia que se utiliza es el denominado A. Este protocolo se modifica con el mecanismo de transacción de exclusividad y con el mecanismo de intervención indirecta, los cuales han sido descritos junto con el protocolo. Hasta que se indique lo contrario no hay concurrencia de transacciones. Las cache utilizadas son de mapeo directo y el tamaño de bloque es de 2 palabras (2 variables).

Suponga la siguiente secuencia de accesos a las variables u y t que están contenidas en bloques distintos del espacio lógico. Cuando estos bloques se almacenan en cache hay conflictos entre ellos. El multiprocesador dispone de 2 procesadores.

accesos		accesos		memoria (valor inicial)	cache(almacena)	
					C1	C2
1. P1 lee t		5. P2 lee t		u: 4	u: no	u: no
2. P2 lee u		6. P2 escribe u (12)		t: 5	t: no	t: no
3. P1 escribe t (21)		7. P1 lee t				
4. P2 escribe u (8)		8. P2 lee t				

**Pregunta a:** Muestre, mediante una tabla, el estado de los bloques que contienen las variables u y t en las caches de cada procesador. Así mismo, muestre las transacciones de bus y quién suministra el bloque. En el campo señal indique la anulación de una transacción (ANU), si es el caso, y la cache que anula la transacción.

acceso	Bus		Memoria		Cache 1				Cache 2		
	trans.	señal	var.	val.	suministro	cont. / varia	valor	estado	cont. / varia	valor	estado

En una nueva versión del multiprocesador se utiliza exclusivamente el protocolo denominado A.

**Pregunta b:** Muestre, mediante una tabla, el estado de los bloques que contienen las variables u y t en las caches de cada procesador. Así mismo, muestre las transacciones de bus y quién suministra el bloque. En el campo señal indique la activación de la señal MOD y la cache que la activa.

acceso	Bus		Memoria		suministro	Cache 1			Cache 2		
	trans.	señal	var.	val.		cont. / varia	valor	estado	cont. / varia	valor	estado

**Pregunta c:** Calcule la ganancia. Para ello utilice el número de transacciones de bus.

**Pregunta d:** En esta pregunta se supone concurrencia de transacciones. Exclusivamente para la acción de anulación (ANU) muestre las transiciones entre estados en las que está involucrada. Si es necesario añada algún estado transitorio.

Un procesador interpreta las instrucciones en orden de programa y utiliza una cache con escritura retardada. Para que las escrituras no bloqueen al procesador se utiliza un buffer denominado buffer de escrituras (BE). Una vez una instrucción store ha finalizado la ejecución en el procesador, en orden de programa, se almacena en el BE hasta que la jerarquía de memoria procese la instrucción. Esto es, cuando el bloque en la cache tiene los permisos necesarios, se extrae del BE y a la vez se actualiza el bloque en la cache. Cada entrada del BE tiene un campo de dirección y un campo de dato y el orden en que se almacenan varias instrucciones store en el BE mantiene el orden de programa.

El objetivo de un BE es soportar la latencia en caso de fallo. El procesador sigue ejecutando instrucciones. Si la instrucción es de cálculo actualiza el estado del procesador. Si la instrucción es un store se almacena en el BE. Si la instrucción es un load y es acierto en cache se actualiza el estado del procesador. En caso de fallo se inicia el acceso a memoria y puede adelantar a una instrucción store almacenada en el BE. Para que el BE sea transparente a la arquitectura se añade circuitería a su alrededor. Si una instrucción load más joven accede a una posición de memoria almacenada en el BE, el procesador se bloquea hasta que no existe coincidencia con direcciones almacenadas en el BE. Si el BE está lleno y se inicia la interpretación de una nueva instrucción store se bloquea la interpretación de instrucciones hasta que se ha vaciado el BE.

**Pregunta e:** Razone de forma sucinta si un procesador con un BE respeta las dependencias de datos al acceder a memoria (load y store).

Este procesador se utiliza en el diseño de un multiprocesador cuya red de interconexión es un bus, las caches privadas utilizan escritura retardada y la coherencia se mantiene con un protocolo de invalidación (MLI).

En las dos siguientes preguntas se limita la funcionalidad del BE descrito.

**Pregunta f:** Suponga que el BE sólo dispone de una entrada. Después de insertar una instrucción store en el BE, un procesador sigue interpretando instrucciones hasta que debe ejecutar una instrucción de acceso a memoria (load / store). En este caso, el procesador se bloquea si en el BE hay pendiente una actualización. Justifique de forma razonada si se mantiene consistencia secuencial.

Después de insertar una instrucción store en el BE, un procesador sigue interpretando instrucciones. Si una de ellas es un store se almacena en el BE. Sin embargo, si interpreta una instrucción load (acierto o fallo) y en el BE hay pendiente alguna actualización el procesador se bloquea.

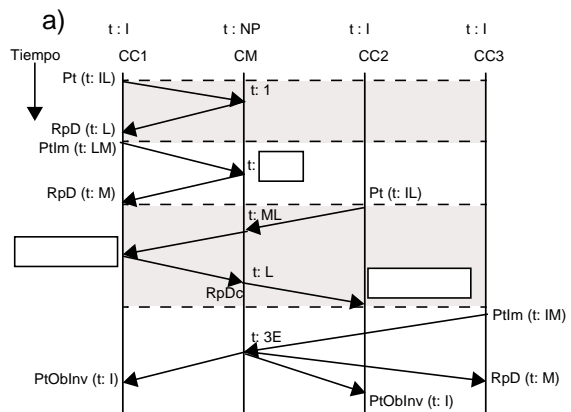
**Pregunta g:** Suponga que el BE dispone de varias entradas y la gestión del BE es FIFO. Justifique de forma razonada si se mantiene consistencia secuencial.

**Pregunta h:** Suponga que el BE dispone de varias entradas y la gestión no es FIFO. Esto es, si el bloque al que accede una instrucción store, cuando se inserta en el BE, tiene los permisos necesarios se actualiza inmediatamente la cache, aunque existan entradas ocupadas del BE. Justifique de forma razonada si se mantiene consistencia secuencial.

### EJERCICIO 4 ( 3.5 puntos)

El protocolo de coherencia que se utiliza es el denominado B. Hasta la pregunta d) no hay concurrencia.

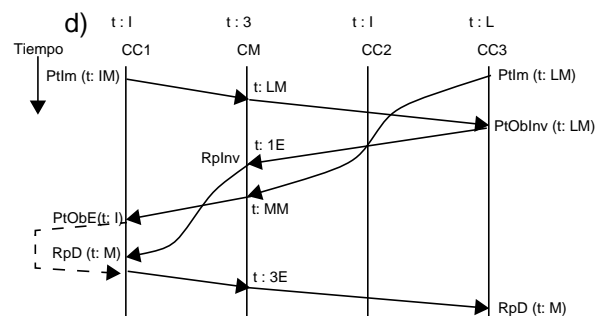
**Pregunta a:** En el siguiente diagrama temporal simplificado se muestra una secuencia de mensajes (peticiones y respuestas). Indique el estado del bloque y el mensaje, cuando no se muestra, en los lugares marcados con un rectángulo. Además, indique la transacción o la ausencia de la misma que no cumple el protocolo. Muestre solo para esta transacción la secuencia de mensajes y estados correcta en el diagrama disponible en la parte derecha. No tenga en cuenta su influencia en transacciones previas o posteriores.



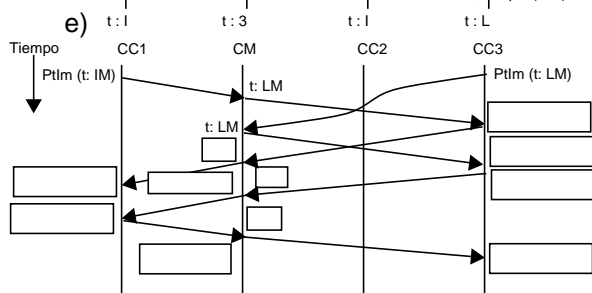
**Pregunta b:** Un diseñador propone modificar las acciones mostradas en los diagramas de transiciones entre estados en la transición del estado L al estado LM por las siguientes acciones:  $PtIm / PtObInv(VP)$ ;  $cnt = |VP|$ . Esta modificación determina que un bloque en el estado LM en cache pueda recibir una petición  $PtObInv$  (entre ellas la que efectúa la petición). El diseñador propone que la respuesta del CC sea  $RpInv$ . Justifique de forma razonada que la respuesta que propone el diseñador permite que se siga manteniendo la coherencia.

**Pregunta c:** Teniendo en cuenta la pregunta previa justifique de forma razonada si se puede eliminar la transición directa del estado L al estado M en el directorio, manteniendo la coherencia.

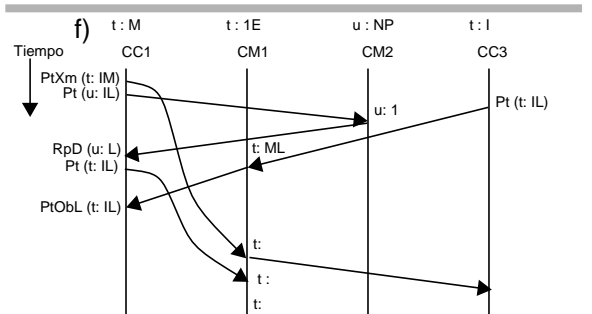
**Pregunta d:** En el siguiente diagrama temporal simplificado se muestra una secuencia de mensajes (peticiones y respuestas). Indique la transacción o la ausencia de la misma que no cumple el protocolo. Muestre solo para esta transacción la secuencia de mensajes y estados correcta en el diagrama disponible en la parte derecha.



**Pregunta e:** En el siguiente diagrama temporal simplificado se muestra una secuencia de mensajes (peticiones y respuestas). Indique, en las trazas que no se muestra, el mensaje que se está transmitiendo y el estado del bloque.



**Pregunta f:** Dados los siguientes diagramas temporales simplificados etiqüete el estado del bloque en las caches y los CM (CM1, CM2) y aña da los mensajes (etiquetandolos) que faltan en el orden temporal adecuado para que finalicen las transacciones iniciadas.



**Pregunta g:** Razone si es factible que el directorio, en lugar de rechazar la petición que está sirviendo, utilice la petición PtXm como respuesta de la petición que ha efectuado al CC y responda a la petición que está procesando.

### Descripción de un protocolo de observación MLI denominado A

Un multiprocesador utiliza un bus como red de interconexión. Las caches privadas utilizan escritura retardada con asignación de contenedor en fallo. El multiprocesador utiliza la técnica de invalidación para mantener la coherencia. En cada contenedor de cache se dispone de dos bits para identificar los tres posibles estados: inválido (I), lectura (L) y modificado (M).

Las peticiones de procesador y las transacciones de bus son las siguientes.

Procesador	Controlador de cache		Memoria
Peticiones	Transacciones	Acciones	Acciones
LPr : lectura del proc.	Pt : petición de bloque	CcRe: reemplazo de un bloque	Dev: almacenar en memoria
EPr : escritura del proc.	PtIm: petición de bloque con intención de modificación	MOD: señal que indica bloque en estado M en una cache	
	PtX: actualización de memoria	CaC: suministro del bloque	

Una cache puede suministrar directamente el dato dentro de una transacción de bus iniciada por otro procesador (CaC). Además, en este caso se actualiza memoria, si es el caso. Cuando una cache tiene un bloque en estado M activa la señal denominada MOD. En el bus se dispone de un cable que es la función OR de las señales MOD de cada una de las caches.

El diagrama de transiciones entre estados se muestra en la figura de la derecha.

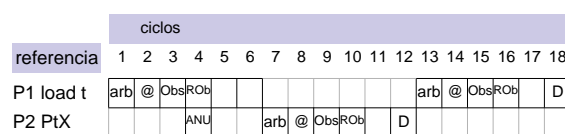
Cuando es necesario efectuar una acción de reemplazo, primero se actualiza memoria, si es el caso. Posteriormente se efectúan las acciones relacionadas con el acceso que determina el reemplazo.

Un procesador inicia los accesos a cache en orden de programa y la cache es bloqueante.

**Intervención directa.** El protocolo de coherencia A descrito utiliza transferencia entre caches para satisfacer la solicitud de un bloque (CaC). Esta característica se denomina intervención directa. Ahora bien, también se puede utilizar lo que se denomina intervención indirecta.

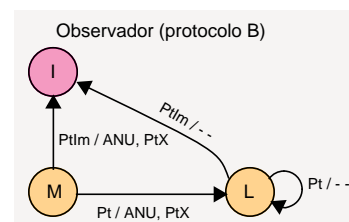
**Intervención indirecta.** Por intervención indirecta se entiende que una cache nunca suministra un bloque directamente a otra cache. En su lugar, la cache que tiene el bloque en estado M anula la transacción en curso y posteriormente actualiza memoria. Se espera que la actualización de memoria finalice antes de que la cache, cuya transacción ha sido anulada, vuelva a iniciar la transacción. En caso contrario, volverá a anularse la transacción. La implementación de este mecanismo requiere de una señal en el bus que indica que una transacción debe repetirse (ANU). Observemos que la señal ANU es el equivalente de la señal MOD cuando se utiliza intervención directa.

En estas condiciones, las transacciones Pt y PtIm siempre obtienen el bloque de datos de memoria. En la figura se muestra un ejemplo de la acción descrita mediante un diagrama temporal.



arb: fase de arbitraje  
 @: fase de transmisión de la dirección  
 Obs: fase de observación  
 ROb: fase de respuesta  
 D: fase de suministro del bloque

Cuando se utiliza intervención indirecta, el diagrama de estados del agente observador en el protocolo de coherencia A se muestra en la figura de la derecha.



Cuando se utiliza una tabla para representar las transacciones de bus y estados de los bloques, la acción de anulación de una transacción, la actualización de memoria por parte del controlador de memoria que anula la transacción y la reemisión de la petición anulada se representa en tres filas consecutivas.

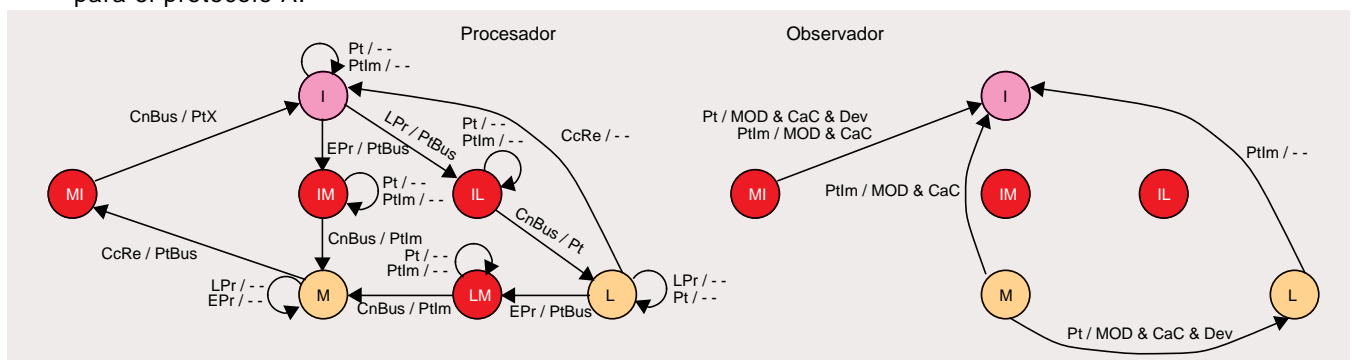
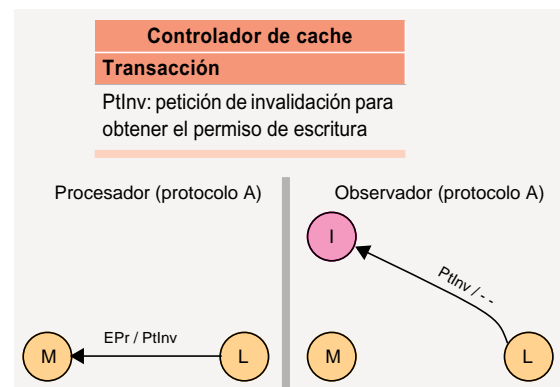
acceso	bus				comentario
	trans.	señal			
P5 load t	Pt	anu C4	...		transacción de P5 anulada por C4
	PtX		...		actualización de memoria por parte de C4
	Pt		...		se repite la transacción de P5

Cuando hay que actualizar memoria, al expulsar un bloque, se utilizan dos filas. En la primera fila se representa la acción de expulsión y en la segunda fila la petición que ha determinado la expulsión.

**Transacción de exclusividad (petición Ptlmv).** Solicitudo de acceso a un bloque en exclusividad partiendo del estado L. Para reducir el ancho de banda de bus se utiliza una transacción específica cuando un bloque en cache está en el estado L y el procesador quiere obtener la exclusividad de acceso al bloque. Esta transacción se denomina Ptlmv. Su objetivo es que las otras caches, que tienen copia del bloque en el estado L, invaliden la copia. La diferencia entre una transacción Ptlmv y una transacción Ptlm es que no se transfiere el bloque de datos por el bus.

La transacción de bus adicional se muestra en la figura de la derecha. En la misma figura se muestra la transición del estado L al estado M en el protocolo de coherencia A. También se muestra, en el diagrama de estados del agente observador, la transición inducida por la observación de una petición Ptlmv.

**Concurrencia.** Cuando se tiene en cuenta que pueden haber transacciones concurrentes hay que considerar nuevas transiciones entre estados. En la siguiente figura se muestran los diagramas de transiciones entre estados del procesador y del observador para el protocolo A.



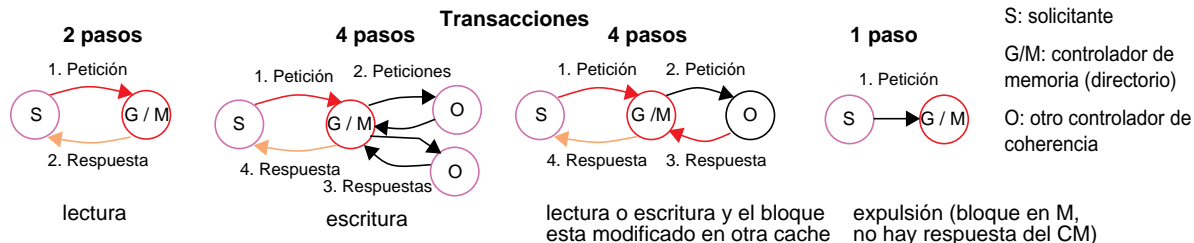
### Descripción de un protocolo de directorio MLI denominado B

Un multiprocesador utiliza una red que mantiene el orden de los mensajes punto a punto entre cada emisor y cada receptor. Cada nodo contiene un procesador, una cache y un módulo de memoria. Las caches privadas son de mapeo directo y utilizan escritura retardada.

El multiprocesador utiliza una estructura de directorio para mantener la coherencia y el protocolo de coherencia es de invalidación (MLI).

El CM utiliza como directorio un vector de presencia y un bit de exclusividad por bloque. El vector de presencia es un vector de bits, con tantos bits como procesadores y cada bit está asociado a un procesador. El bit de exclusividad se utiliza para indicar que sólo existe una copia del bloque en una cache privada, la cual está identificada en el vector de presencia. Los CC responden a las peticiones de invalidación del CM. El CM es el recolector de las respuestas.

Las secuencias de mensajes de las transacciones son las siguientes:



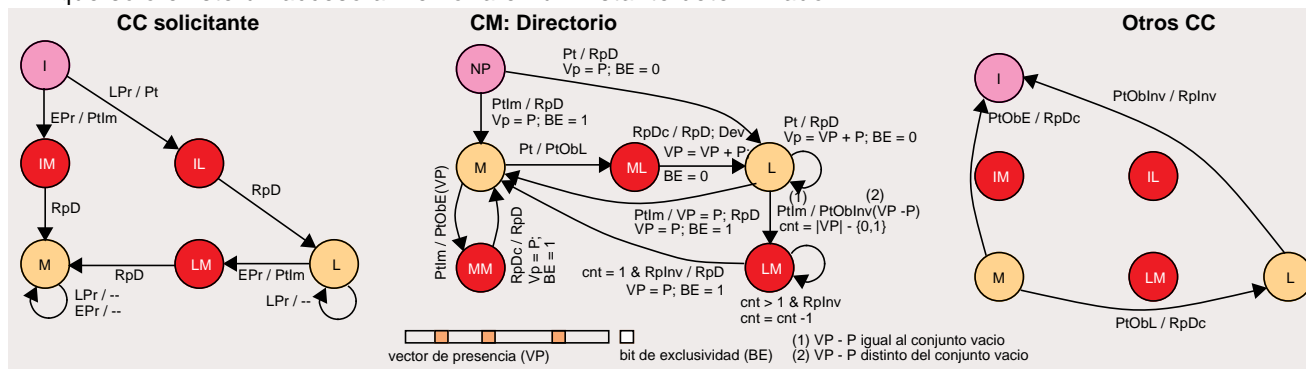
Las caches privadas de los procesadores son bloqueantes. En un fallo de cache o en una solicitud de exclusividad se suspende la interpretación de instrucciones y se reanuda al finalizar la transacción.

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

Procesador	Controlador de cache (CC)		Controlador de memoria (CM)		
Peticiones	Mensajes de petición del CC al CM	Respuestas del CM al CC	Mensajes de petición del CM al CC	Respuestas del CC al CM	Acciones
LPr : lectura	Pt : petición de bloque	RpD: respuesta con el bloque a una petición Pt o Ptlm	PtObE: petición de observación de escritura, inducida por una petición Ptlm y el estado del bloque en el directorio es M	RpDc: respuesta con el bloque a una petición PtObL o PtObE y el estado del bloque en cache es M	Actualización del directorio
EPr : escritura	Ptlm: petición de bloque con intención de modificarlo	RpRch: rechaza procesar la petición. (concurrency)	PtObInv: petición de invalidación, inducida por una petición Ptlm y el estado del bloque en el directorio es L	RpInv: respuesta a una petición de invalidación	Dev: actualización de memoria
	PtXm: petición de expulsión de un bloque en estado M		PtObL: petición de observación de lectura, inducida por una petición Pt y el estado del bloque en el directorio es M		

En los siguientes diagramas de estados se muestran todas las transiciones entre estados, ya sean estables o transitorios, de un bloque en cache y en el directorio. En el protocolo que se describe, el bit de exclusividad del directorio se activa cuando una cache solicita el bloque para actualizarlo. El directorio dispone de un contador (cnt) para contabilizar las peticiones de invalidación que aún no han sido respondidas.

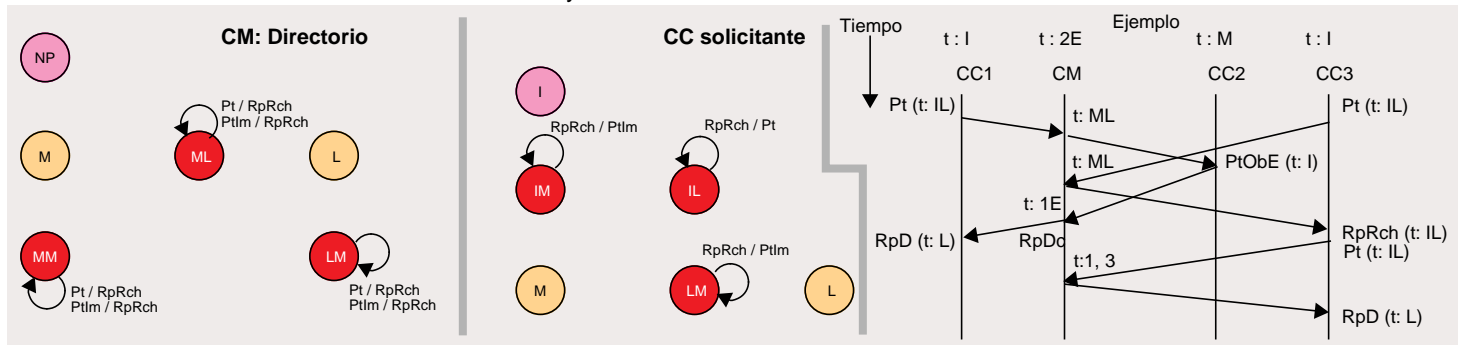
Por ahora suponemos que no se producen conflictos en las caches privadas. También, suponemos que sólo existe un acceso a memoria en un instante determinado.



**Concurrencia:** Cuando puede existir más de una transacción en curso hay que serializar los accesos a un mismo bloque en el directorio. El mecanismo utilizado para serializar las peticiones a un bloque en el directorio se denomina rechazar la gestión de la petición (Pt y Ptlm). El directorio rechaza (RpRch) una petición de un CC a un bloque en un estado transitorio (ventana de vulnerabilidad).

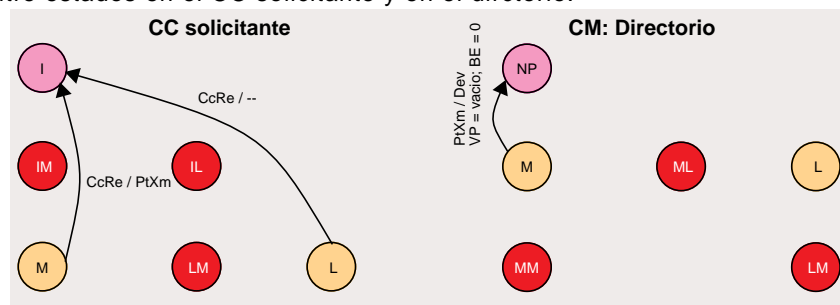


Cuando un CC recibe una respuesta de rechazo (RpRch) a una petición debe de volver a emitir la petición, teniendo en cuenta el estado actual del bloque. Para ello se añaden las siguientes transiciones entre estados en el directorio y en el CC solicitante.



**Conflictos en cache.** A partir de ahora supondremos que en las caches se pueden producir conflictos, lo cual requiere efectuar acciones de reemplazo (CcRe). 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.

La acción de reemplazo de un bloque en estado L es silenciosa. Por tanto, el directorio no es preciso. Por otro lado, cuando se espulsa un bloque en estado M no se espera respuesta del directorio. En una petición PtXm se actualiza el directorio y memoria. En la siguiente figura se muestran las transiciones entre estados en el CC solicitante y en el directorio.



**Cruce de peticiones.** La acción de expulsión de un bloque en estado M puede determinar la detección de un cruce en el directorio. Una posibilidad para gestionar el cruce es que el directorio al procesar una petición PtXm, a un bloque que está en un estado transitorio, rechace la petición que ha determinado que el bloque esté en el estado transitorio. Memoria se actualiza con el bloque expulsado.

Un cruce de peticiones en el CC, que es inmediato de identificar, es la recepción de una petición del CM (PtObE, PtObL) después de expulsar un bloque (bloque en estado I). Esta petición del CM es debida al procesado de una petición Pt o Ptlm. El CC no responde a la petición del directorio (PtObE, PtObL) cuando el bloque está en el estado I.

Además del cruce de peticiones descrito, en el CC se pueden producir otros cruces después de que el CC expulsa un bloque en estado M o reemplace un bloque en el estado L (expulsión silenciosa). En el siguiente diagrama de transiciones entre estados se muestran el resto de transiciones.

