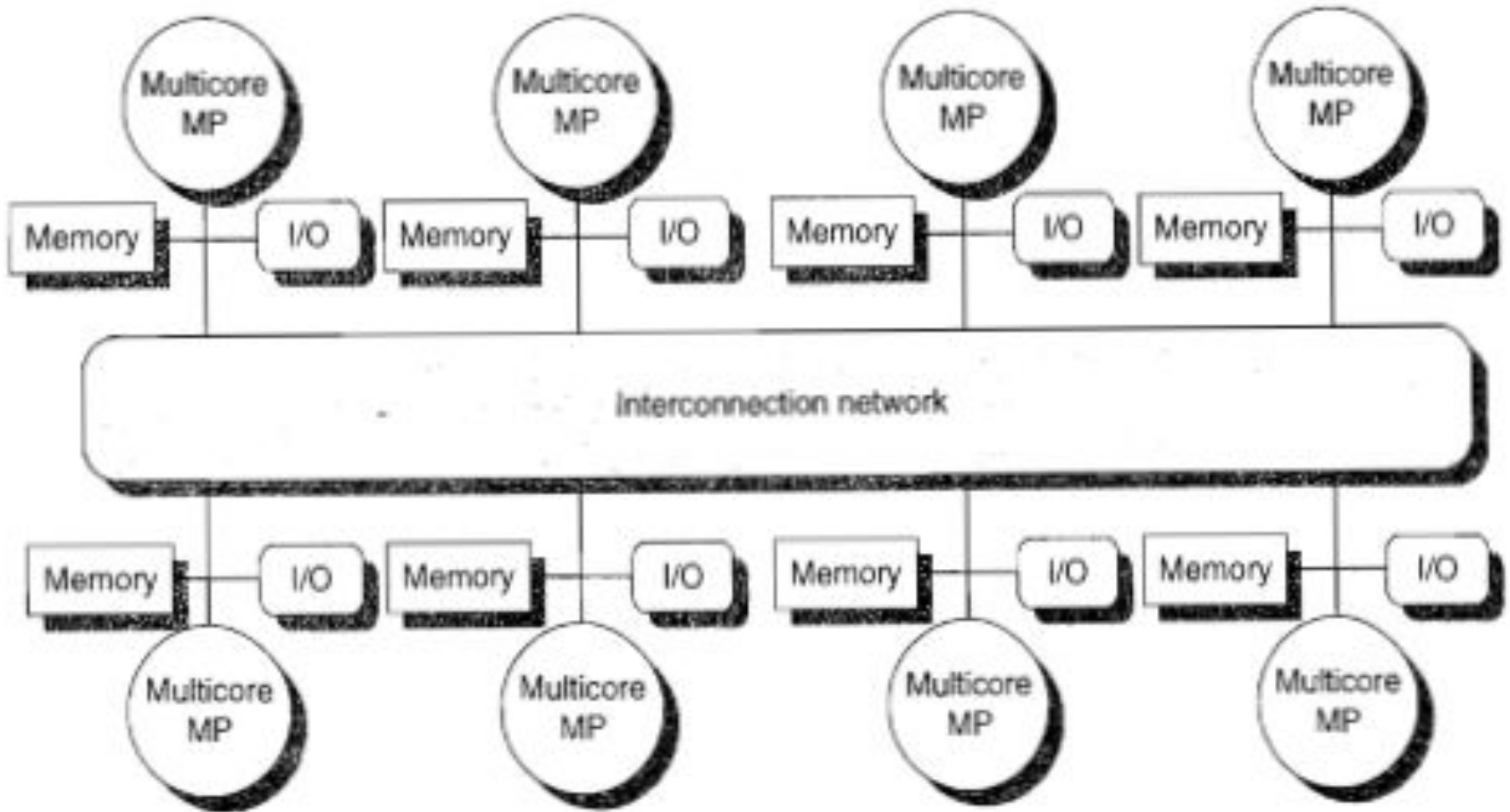
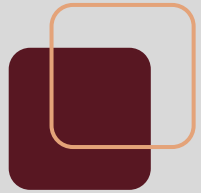




# Aspectos avanzados de arquitectura de computadoras Multiprocesadores (II)

Facultad de Ingeniería - Universidad de la República  
Curso 2018

# Distributed Shared Memory (1/5)

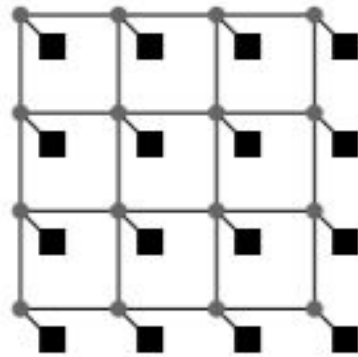
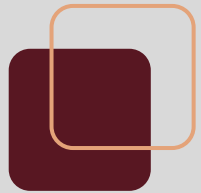




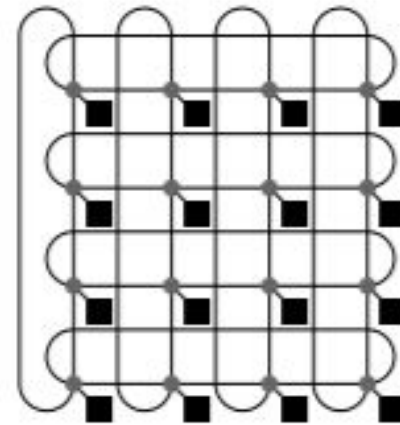
## Distributed Shared Memory (2/5)

- El objetivo principal de distribuir la memoria es aumentar el ancho de banda *global* del sistema.
- Sin embargo, como la memoria sigue siendo compartida, nada previene que un caché X contenga cualquier dirección de memoria (incluso no de la local), obligando a realizar broadcasts de la información al usar *snooping*.

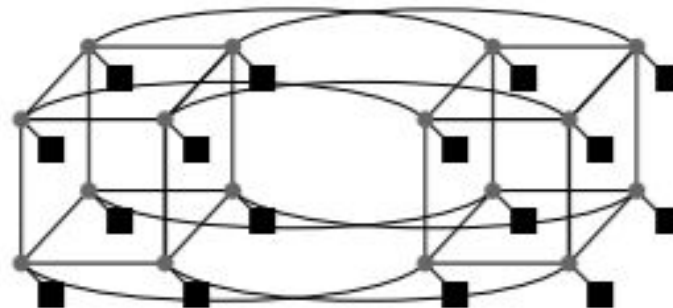
# Distributed Shared Memory (3/5)



(a) 2D grid or mesh of 16 nodes

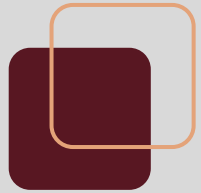


(b) 2D torus of 16 nodes



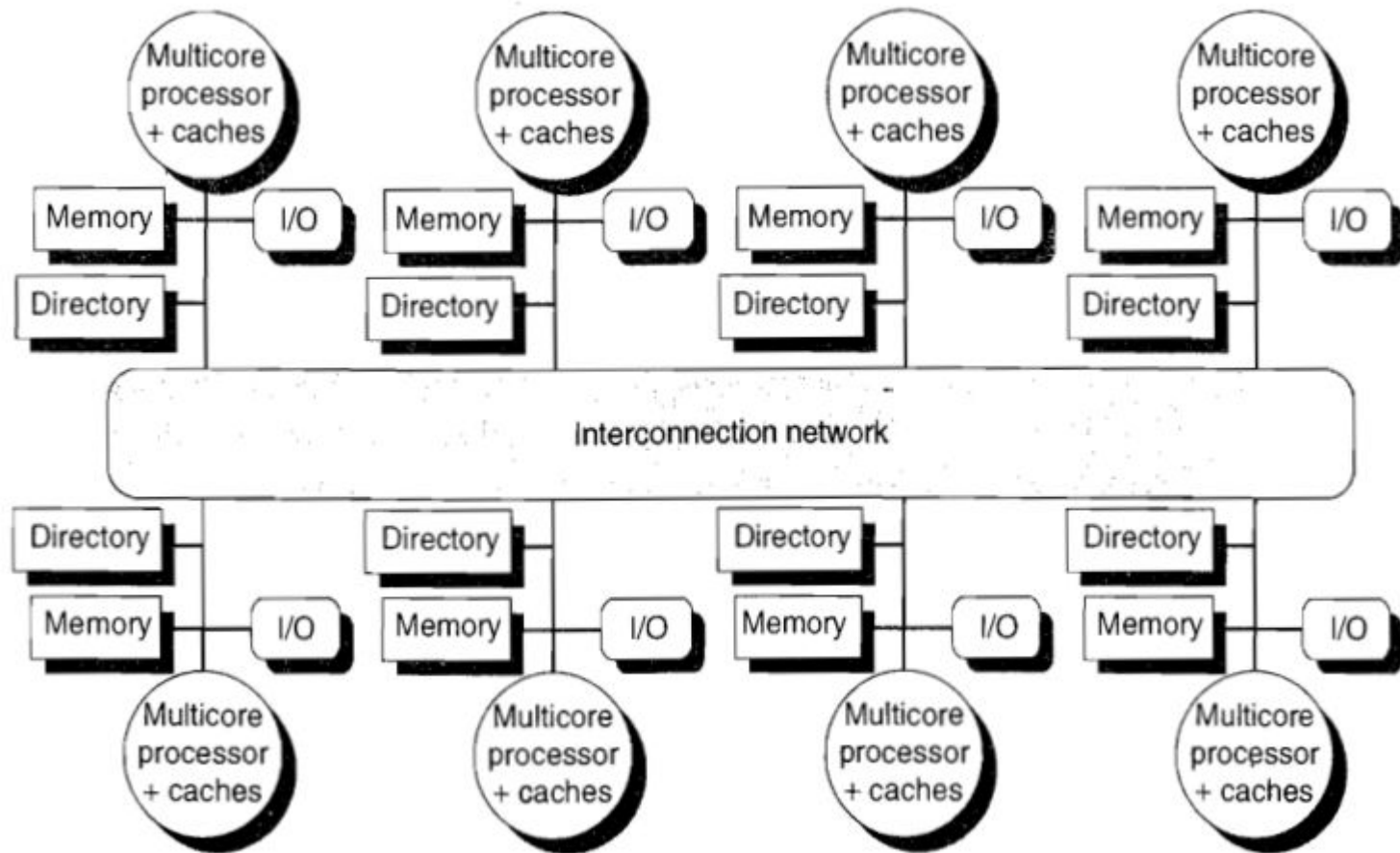
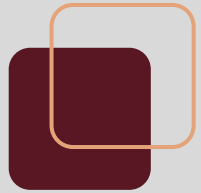
(c) Hypercube of 16 nodes ( $16 = 2^4$  so  $n = 4$ )

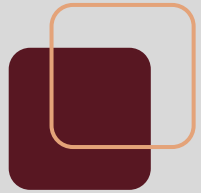
# Distributed Shared Memory (4/5)



- La alternativa a un protocolo de coherencia de caché de husmeo es uno basado en directorios.
- Un directorio mantiene el estado de cada bloque que pueda ser cacheado.

# Distributed Shared Memory (5/5)





# Directory Based Protocols (1/4)

- ¿Qué información guarda un directorio?
  - El estado del bloque
  - Qué cachés tienen copias del bloque



## Directory Based Protocols (2/4)

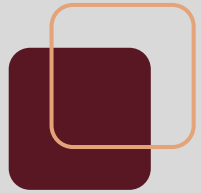
- La línea core i7 de intel, que utiliza un caché L3 unificado, implementa un directorio en el propio caché.
- Por cada línea del caché, mantiene un vector de bits de largo igual a la cantidad de cores (4 en la mayoría), indicando con 1 que el caché tiene dicho bloque. De esta manera, las invalidaciones se envían solo a dichos cachés.





# Directory Based Protocols (3/4)

- Los estados posibles de un bloque son similares a los utilizados en los protocolos de *snooping* y por lo general incluyen:
  - Shared: Uno o más cachés tienen copias del bloque y el valor coincide con el de memoria.
  - Uncached: Ningún bloque tiene copias del bloque.
  - Modified: Exactamente un caché tiene una copia del bloque y es la copia más actualizada.



# Directory Based Protocols (4/4)

- Además del estado de los bloques, se debe guardar qué cachés tienen copias del bloque.
  - Array de bits.
- Por último, también se guarda el estado del bloque en las cachés (eficiencia).



# Mensajes (1/3)

- Los mensajes a enviar son esencialmente los mismos, aunque más complejos en este caso porque requieren de una respuesta explícita para asegurar la atomicidad de las operaciones.



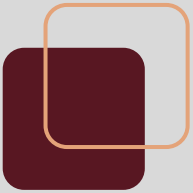
## Mensajes (2/3)

- Dado que la memoria está distribuida de forma estática, un caché siempre sabe a qué directorio debe enviar las consultas.
- Típicamente, los bits altos de la dirección indican el nodo y los bits bajos indican el desplazamiento (offset) dentro de dicha memoria.

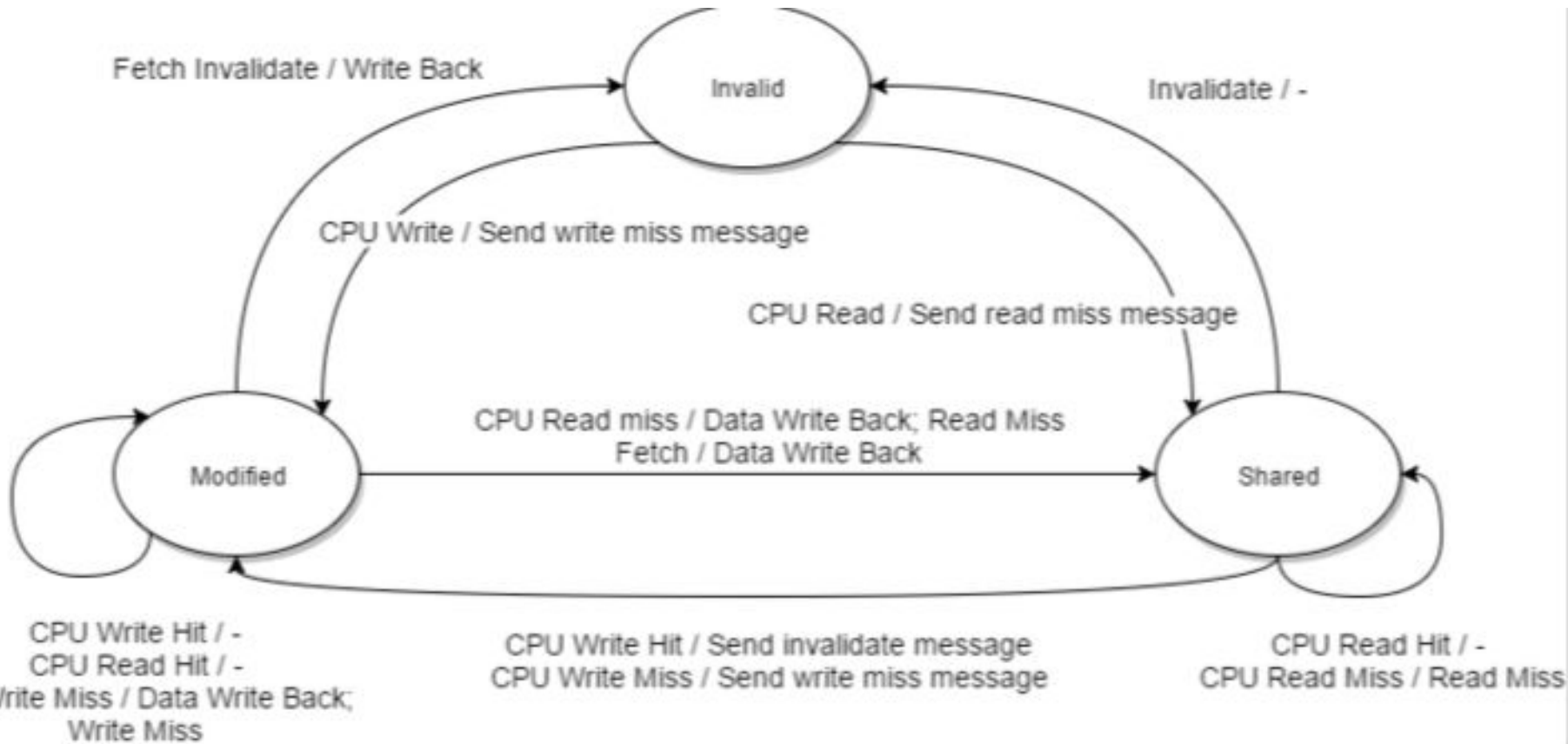


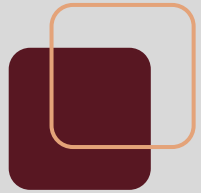
## Mensajes (3/3)

- A diferencia de los protocolos de *snooping*, donde todos los nodos son simétricos, en un protocolo de directorio las acciones tomadas por el caché son diferentes a las tomadas por el directorio.
- Por esta razón, hay dos diagramas de estados a estudiar, el mantenido por el caché en cada línea y el mantenido por el directorio en cada bloque del sistema.



# Diagrama de Estados - Caché





# Acciones del Directorio (1/2)

- El directorio tiene las responsabilidades de mantener la lista de cachés que comparten un bloque, así como enviar los valores de los bloques solicitados por los cachés.
- De este modo, un evento sobre el directorio provoca dos acciones: la actualización del estado de algún bloque y/o envío de mensajes adicionales para completar el pedido.

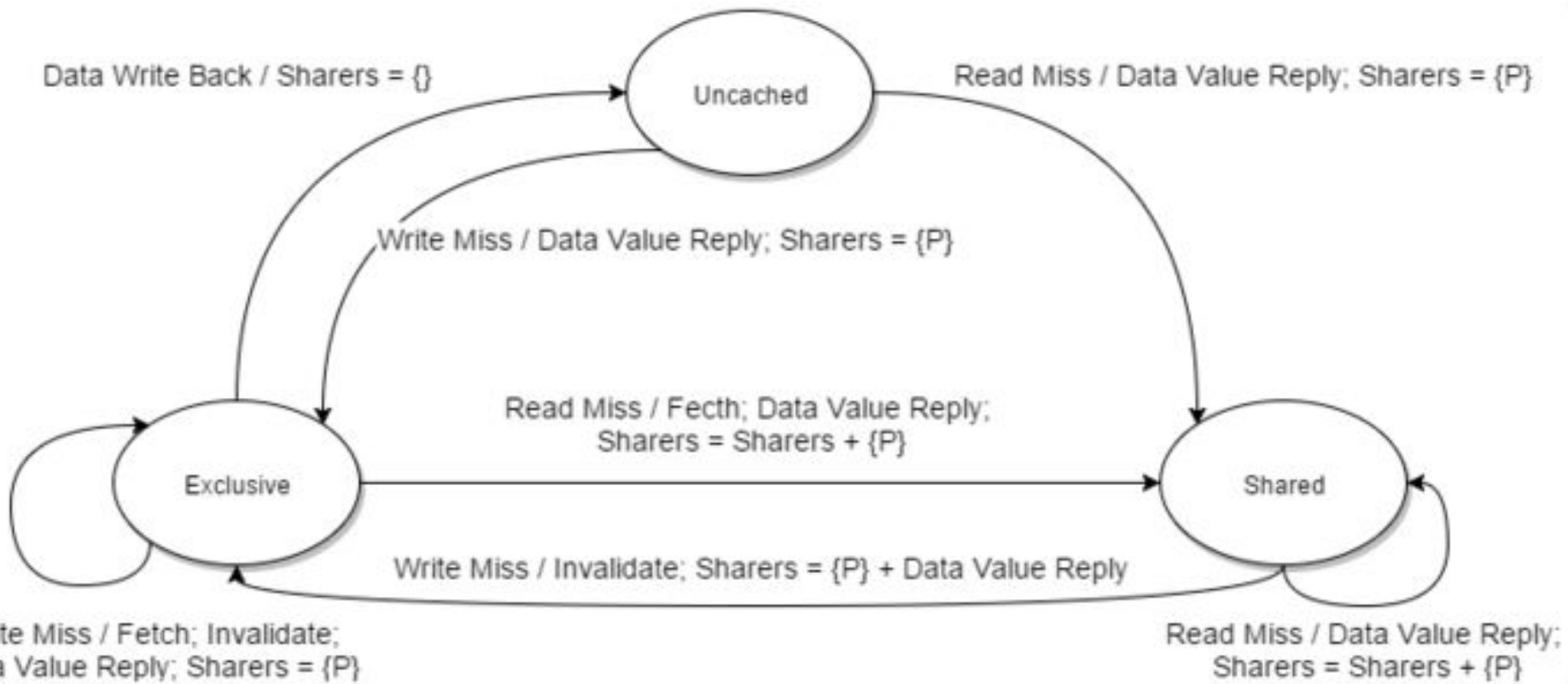


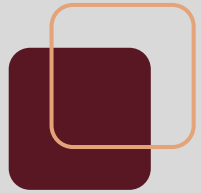
## Acciones del Directorio (2/2)

- Los posibles eventos que desencadenan acciones del directorio son:
  - Read miss: Un caché solicita un bloque en modo lectura.
  - Write miss: Un caché solicita un bloque en modo lectura.
  - Data write back: Un caché debe reemplazar un bloque que está actualizado y envía su copia más actualizada al directorio.



# Diagrama de Estados - Directorio





# Optimizaciones

- El protocolo presentado es una simplificación. Los protocolos implementados comercialmente contienen optimizaciones, por ejemplo:
  - Saltear al directorio en operaciones que no lo requieran (por ejemplo, en un read/write miss a un bloque en estado exclusivo)



# Comunicación (1/2)

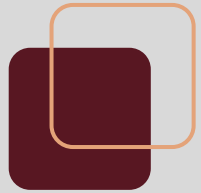
- En general, la implementación de directorios grandes y distribuidos requiere múltiples canales de comunicación para evitar deadlock!
  - Ejemplo: para evitar que mensajes de respuesta se encolen luego de nuevos pedidos.



## Comunicación (2/2)

- En *snooping*, la sincronización se implementa a través del medio compartido.
  - Quien ganara el control del bus completaba su operación primero.
- En directorios, estos son utilizados como punto de sincronismo:
  - El mensaje que llegue primero es completado primero
  - Los demás se encolan o reciben *NACK*

Fin



¿Preguntas?