

Arquitectura de Computadores Problemas (hoja 5). Curso 2012-13

1. Indica claramente las diferencias entre los siguientes protocolos de coherencia vistos en clase: Snoopy, MSI y MESI en términos de su número de estados, política de escritura de cache y política para mantener la coherencia.

	Num. Estados	Política Escritura de cache	Política para mantener coherencia
Snoopy			
MSI			
MESI			

Además, indica qué aporta cada protocolo frente al anterior.

2. En el protocolo de coherencia MESI:
- Indica cuales son las transacciones de bus relevantes y dibuja las transiciones de estado a las que dan lugar.
 - Dibuja las transiciones de estado originadas por peticiones locales del procesador.
3. Para cada una de las cadenas de referencias a memoria que se muestran a continuación, construye una tabla en la que se indique el estado del bloque referenciado en cada controlador de cache en cada instante y el tiempo necesario para cada referencia suponiendo un multiprocesador de bus compartido que utiliza el protocolo MESI explicado en clase

Cadena 1: r1 w1 r1 w1 r2 w2 r2 w2 r3 w3 r3 w3

Cadena 2: r1 r2 r3 w1 w2 w3 r1 r2 r3 w1

Cadena 3: r1 r2 r3 r3 w1 w1 w1 w1 w2 w3

Todas las referencias en las cadenas son a la misma posición de memoria. r y w indican lectura y escritura, respectivamente, y el dígito se refiere al procesador que hace la referencia. Asume que todas las caches están inicialmente vacías, y usa el siguiente modelo de coste:

Un acierto de cache el lectura/escritura: 1 ciclo

Fallos que requieren una transacción del bus simple: 60 ciclos

Fallos que requieren una transferencia de un bloque de cache: 90 ciclos

4. Repetir el problema anterior para la cadena de referencias
r2 w2 r2 w1 r1 r3 w3
pero suponiendo los siguientes protocolos

- MSI
- MESI

Indica cuál es la principal ventaja que aporta el protocolo MESI

5. Para la cadena de referencias a memoria que se indica en la primera fila de la Tabla, calcula el coste de ejecutarla en un multiprocesador de bus compartido que soporta el protocolo **MESI** y usa **BusUpgrd**. Indica el resultado en forma de una tabla que muestre también la evolución en el tiempo del contenido de los marcos de bloque de cache, su estado y la transacción que se realiza en el bus; donde cada columna es un acceso y cada fila un marco de bloque. La memoria principal contiene cuatro bloques y cada una de las caches dos marcos de bloque. Las referencias a memoria indican el bloque de memoria principal al que acceden; por ejemplo P1 lee 00 quiere decir que el procesador 1 lee el bloque de memoria 00. Asume que todas las caches están inicialmente vacías, y usa el siguiente modelo de coste:

Un acierto de cache el lectura/escritura: 1 ciclo

Fallos que requieren una transacción del bus simple: 60 ciclos

Fallos que requieren una transferencia de un bloque de cache: 90 ciclos

Suponer que se utiliza en todos los casos caches con asignación en escritura (write allocate), post escritura, emplazamiento asociativo, política de reemplazo: se reemplaza el bloque que lleva más tiempo sin ser usado. A modo de ejemplo se han completado las dos primeras columnas.

CADENA	P1 lee 00	P2 lee 00	P1 lee 00	P2 esc 01	P1 lee 01	P2 esc 00	P1 esc 01	P1 lee 11
Bloque0 P1	00 E	00 S						
Bloque1 P1								
Bloque0 P2		00 S						
Bloque1 P2								
Ciclos	90	90						

6. Supongamos un protocolo de Directorio simple tipo MSI (información de directorio = bit de presencia por cada nodo + bit de modificación) en un sistema de 1024 procesadores y bloques de memoria de 64 bytes.
- Describir brevemente (a modo de algoritmo si es posible) las operaciones que se realizan en el caso de Fallo de lectura a un bloque en estado compartido.
 - ¿Cuál es la sobrecarga de memoria asociada al directorio?
7. Supongamos un protocolo de Directorio simple tipo MSI (información de directorio = bit de presencia por cada nodo + bit de modificación) en un sistema de 32.768 nodos (1 procesador por nodo) y bloques de memoria de 2.048 bytes. Indicar la sobrecarga de memoria asociada al directorio en los siguientes casos:
- Sin optimización
 - Incremento de línea a 4.096 bytes
 - Reducción de la anchura del directorio usando 100 punteros por bloque de memoria.
 - Reducción de la altura del directorio mediante su organización como una cache donde solamente hay espacio para un 50% de las líneas.

8. Sea un sistema multiprocesador de memoria compartida tipo UMA donde los n procesadores ejecutan el siguiente código:

```
for i= 1 to m
  P[1](write, x);
  P[2..n](read, x);
end for
```

El tamaño de línea de cache es 8 palabras de 8 bytes; y el de la orden de control es 6 bytes, tanto para solicitud de línea en caso de fallo como para invalidación (por medio de BusRdX) y actualización.

Indicar en función de m y n , el tráfico (en bytes) generado por un protocolo de actualización y uno de invalidación.

Nota.- No se debe contabilizar el tráfico el caso especial $i=1$.