

ARQUITECTURA DE COMPUTADORES

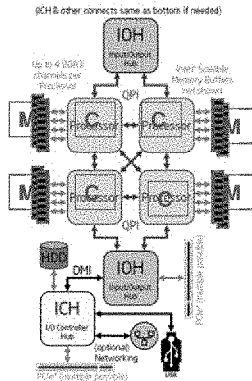
BENCHMARK del TEMA 3

Un microprocesador multinúcleo no incluye memoria cache

(F)

El servidor de la figura tiene una arquitectura de tipo UMA

(F)



Señale en la figura (con una C) dónde está la memoria cache

Señale en la figura (con un M) dónde está la memoria principal

En un microprocesador SMT (Multihebra simultánea), en un instante determinado se pueden enviar a ejecutar instrucciones de hebras diferentes

(V)

En un microprocesador SMT (multihebra simultánea), se procesan varias hebras concurrentemente aunque en un instante determinado solo se pueden enviar a ejecutar instrucciones de la misma hebra.

(F)

En el protocolo MESI para mantener la coherencia de cache una línea puede estar en el estado S solo en una de las caches del multiprocesador

(F)

En el protocolo MESI para mantener la coherencia de cache, una línea puede estar en el estado E en varias caches del multiprocesador

(F)

En el protocolo MSI, si en la cache de un nodo N1 hay un bloque en estado M (Modificado), y ese nodo detecta que otro procesador intenta leer un dato que está en ese bloque, el bloque pasa al estado I (Inválido) en el nodo N1

(F)

En un multiprocesador NUMA con 16 nodos, 4 GBytes por nodo, y líneas de cache de 64 Bytes. ¿Cuántas entradas tiene el directorio de memoria utilizado en cada nodo para mantener la coherencia de cache en un protocolo MSI sin difusión?

$$2^2 * 2^{30} / 2^6 = 2^{26} \text{ entradas}$$

En el multiprocesador NUMA descrito en la pregunta anterior ¿Cuántos bits tiene cada una de las entradas del directorio que se utiliza para mantener la coherencia de cache?

$$16 + 1 = 17$$

¿Qué valores se puede observar en R si el modelo de consistencia de memoria del computador donde están los procesadores que ejecutan estos códigos no respeta el orden $W \rightarrow W$ (sí respeta los demás), e inicialmente $X=Y=0$?

| | | | |
|-----|-----|-----|----------------|
| P1: | X=2 | P2: | R=1; |
| | Y=1 | | if (Y==1) R=X; |

R=0; R=1; R=2

Si respeta el orden $W \rightarrow W$: R=1; R=2

¿Qué valor deben r1, r2, y r3 para que la secuencia de instrucciones siguiente implemente un cerrojo (lock(k)) en el que k=1 significa que el cerrojo está cerrado y 0 que está abierto.

```

b=r1;
do
    compare&swap(r2,b,k); // si r2==k k y b se intercambian
while (b==r3);

```

r1=1

r2=0

r3=1

Arquitectura de Computadores

Benchmark de Teoría del Tema 4

Suponga que en un mismo ciclo se decodifican estas cuatro instrucciones (el número entre paréntesis indica en orden en el que están en el código), y pasan a una ventana de instrucciones única desde la que se emiten a las distintas unidades funcionales. El procesador tiene un buffer de reorden (ROB) para la finalización ordenada donde también se implementa el renombrado.

| | | |
|-------|----------------|---------------|
| (i) | add r2, r2, r1 | // r2 = r2+r1 |
| (i+1) | mul r3, r4, r1 | // r3 = r4*r1 |
| (i+2) | add r4, r3, r1 | // r4 = r3+r1 |
| (i+3) | add r1, r2, r1 | // r1 = r2+r1 |

1. Si el procesador tiene TRES unidades de suma y UN multiplicador y puede emitir cuatro instrucciones por ciclo con emisión DESORDENADA, se podrían emitir las cuatro instrucciones en el mismo ciclo. **Hay dependencias RAW** (F)
2. Entre la instrucción (i+1) y la (i+2) SOLO hay un riesgo de tipo WAR. **También hay dependencia RAW en r3 aparte de la WAR en r4** (F)
3. Si el procesador descrito anteriormente puede emitir (con emisión DESORDENADA) DOS instrucciones por ciclo y tiene DOS unidades de suma (con un retardo de 2 ciclos) y UN multiplicador (con un retardo de 5 ciclos). ¿Cuántos ciclos tardan en emitirse las cuatro instrucciones (cuente a partir del ciclo de la decodificación, sin incluir este)?

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-------|----|----|-----|----|-----|-----|----|-----|----|
| (i) | EX | EX | ROB | WB | | | | | |
| (i+1) | EX | EX | EX | EX | EX | ROB | WB | | |
| (i+2) | | | | | | EX | EX | ROB | WB |
| (i+3) | | | EX | EX | ROB | | | | WB |

Tardan 5 ciclos en emitirse las instrucciones y 9 en retirarse todas (suponiendo que se pueden retirar más de dos instrucciones por ciclo)

4. En la predicción de salto dinámica implícita, cada vez que llega una instrucción de salto condicional se predice que NO se va a producir el salto **(es dinámica, luego debe cambiar según lo que ocurre en la ejecución del código. En este caso, se hace lo que se hizo en la última ejecución)** (F)
5. Indique cómo expresaría sin instrucciones de salto utilizando un repertorio típico de instrucciones con predicado en el que TODA instrucción se puede predicar (suponga que LOS PREDICADOS ESTÁN INICIALIZADOS A 0):

if (r2==0) then r2=r2+r3 else r2=r2-r3;

p1,p2 cmp.eq r2,0
(p1) add r2,r2,r3
(p2) sub r2,r2,r3

6. NO existen instrucciones de ejecución condicional en los repertorios de instrucciones de procesadores superescalares (F)
7. La segmentación software NO necesita un hardware especial de apoyo

8. La responsabilidad del aprovechamiento del paralelismo entre instrucciones (ILP) en un procesador VLIW recae fundamentalmente en el compilador (V)

9. Los procesadores superescalares no pueden tener un repertorio de instrucciones CISC, por eso todos los núcleos de Intel son VLIW (V)

10. Considere las dos instrucciones siguientes (i precede a i+1 en el código) (F)

| | |
|-------|---|
| (i) | sw 0(r5), r2 // $M(r5) \leftarrow r2$ |
| (i+1) | lw r4, 32(r5) // $r4 \leftarrow M(r5+32)$ |

Para que un procesador adelante la ejecución de (i+1) a la de (i), DEBE IMPLEMENTAR adelantamiento de loads a stores ESPECULATIVO (la dirección a la que se hace el almacenamiento, $r5+0$, siempre va a ser distinta de la dirección desde donde se lee, $r5+32$)

(F)