Arquitectura de Computadores (543426)

Certamen No. 3 (6 de Julio de 2015)

Nombre: _____

Matrícula: _____

Prob. 1:	/6
Prob. 2:	/6
Prob. 3:	/6
Total:	/18

Reglas: Tiempo: 90 minutos. Responda en las mismas hojas del certamen y entregue **todas** las hojas al finalizar. Escriba su nombre en todas las hojas.

Prob. 1: [6p] Memoria DRAM-cache

Considere una CPU con un reloj de 2GHz, un sistema de memorias cache de 3 niveles y memoria DRAM DDR3. Existen 2 alternativas para el sistema de memoria DRAM:

1. DDR3-1066 con chips 6-6-6-10-T1, en configuración de canal dual (dos buses de 64 bits).
2. DDR3-2133 con chips 9-9-9-15-T1, en configuración de canal simple (un bus de 64 bits).

En ambos casos, el largo de la ristra es de 8 datos. El tamaño de bloque del cache L3 es de 256 bytes.

a) [4p] ¿Cuál es la configuración de memoria que entrega mayor desempeño? Justifique.

tL3 = [tasa_comandos + tRCD + nRistras*(tCAS + largo_ristra/2)] * tBus

Caso 1:

DDR3-1066 → fBus = 1066/2 = 533MHz → tBus = 1.876ns
Transacciones = 256/16 = 16 (dos ristras)
tRCD = tCAS = 6. Tasa comandos = 1
t1L3 = [1 + 6 + 2*(6 + 4)] * 1.876ns
t1L3 = 27 * 1.876ns
t1L3 = 50.652ns

Caso 1:

DDR3-2133 → fBus = 2133/2 = 1066.5MHz → tBus = 0.938ns
Transacciones = 256/8 = 32 (cuatro ristras)
tRCD = tCAS = 9. Tasa comandos = 1
t1L3 = [1 + 9 + 4*(9 + 4)] * 0.938ns
t1L3 = 62 * 0.938ns
t1L3 = 58.156ns

Nombre: Matrícula:

b) [2p] Asuma que el procesador utiliza un pipeline capaz de despachar hasta una instrucción por ciclo, y que su CPI con un cache L3 perfecto es de 1.3 La tasa global de fallos del cache L3 es de un 0.5% para instrucciones y de 1% para datos. ¿Cuál es el CPI alcanzado por los dos sistemas de memoria DRAM, asumiendo que un 20% de las instrucciones son load/store?

CPI = CPI_base + tFallo_global_I*tL3 + frac_loadstore*fFallo_global_D*tL3
CPI_base + (tFallo_global_I + frac_loadstore*fFallo_global_D)*tL3
tCPU = 1/(2GHz) = 0.5ns

Caso 1:
t1L3 = 50.652ns/0.5ns = 101.3
CPI1 = 1.3 + (0.005 + 0.2*0.01) * 101.3
CPI1 = 2

Caso 2:
t2L3 = 58.156ns/0.5ns = 116.31
CPI2 = 1.3 + (0.005 + 0.2*0.01) * 116.31
CPI1 = 2.11

Nombre: Matrícula:

Prob. 2 [6p]: Arquitectura de memorias cache

Suponga un sistema ficticio con una memoria cache de un tamaño total de 64 bytes y bloques de 16 bytes. Esta memoria está conectada a un procesador con direcciones de 12 bits. Se consideran dos alternativas para organizar el cache: completamente asociativa y traducción directa.

a) [2p] Indique la cantidad de bits de la dirección asociados a desplazamiento (offset), índice y tag en cada una de las alternativas.

Para los dos cache: bits 3-0 son offset (4).
Número de líneas = 64/16 = 4.
Cache directo: bits 5-4 son índice (2), bits 11-6 son tag (6).
Cache asociativo: bits 11-4 son tag (8).

b) [4p] Para la siguiente secuencia de direcciones (en binario) en cada tipo de cache, indique si la referencia resulta en un fallo o un acierto, clasifique el tipo de fallo (obligatorio, conflicto, capacidad) e identifique la línea del cache asociada a la referencia. Asuma que los cache se encuentran inicialmente vacíos y la política de reemplazo (cuando corresponda) es LRU.

Dirección	Asociativa			Directa		
	F/A	T. Fallo	Línea	F/A	T. Fallo	Línea
0000 0100 0010	F	O	0	F	O	0
0000 0100 1000	A	-	0	A	-	0
1000 1100 1010	F	O	1	F	O	0
0000 0100 1110	A	-	0	F	Conf	0
0000 0100 1010	A	-	0	A	-	0
1000 1100 1000	A	-	1	F	Conf	0
1000 1100 1100	A	-	1	A	-	0
1111 1101 1001	F	O	2	F	O	1
1111 0001 0010	F	O	3	F	O	1
1111 1000 0100	F	O	0	F	O	0
1111 1101 0100	A	-	2	F	Conf	1

0000 0100 0001	F	Cap	1	F	Cap	0

Nombre: Matrícula:

Considere un procesador súperescalar con ejecución desordenada de instrucciones y predicción de saltos. El buffer de reordenamiento (ROB) es capaz de recibir dos nuevas instrucciones por ciclo de la cola de instrucciones. Para efectos de este problema, asumiremos que el predictor de saltos es perfecto y que la cola de instrucciones se mantiene permanentemente llena. El procesador posee 1 multiplicador de punto flotante, dos sumadores de punto flotante y una unidad de load/store, todos completamente segmentados. El ROB puede despachar instrucciones simultáneamente a todas estas unidades funcionales. Las latencias de las unidades funcionales son de 5, 3 y 2 ciclos, respectivamente. La etapa de write-back escribe hasta dos resultados por ciclo al ROB. No hay forwarding, por lo que primero se escribe en el ROB un resultado, y en el ciclo siguiente se puede despachar la instrucción que usa ese resultado a las unidades funcionales. En caso de haber un conflicto estructural, se despacha primero la instrucción que está antes en el programa (ver primeras dos líneas de la tabla adjunta).
Considere el siguiente segmento de código assembly:

```
I1:     lw.s    $f0, 0($a1)
I2:     lw.s    $f1, 4($a1)
I3:     add.s   $f0, $f0, $f11
I4:     add.s   $f1, $f1, $f12
I5:     mul.s   $f11, $f0, $f1
I6:     sw.s    $f11, 8($a1)
I6:     add.s   $f12, $f0, $f1
I8:     sw.s    $f12, 12($a1)
```

El código anterior es el cuerpo de un lazo iterativo, donde hemos omitido las instrucciones enteras que actualizan punteros e índices, y el salto condicional. Se asume que previo a la ejecución del lazo, las instrucciones que escriben en los registros $f11 y $f12 **se encuentran en los slots T21 y T22 del ROB**, respectivamente, y que estas instrucciones **escribirán en el ROB durante los ciclos 10 y 11,** respectivamente.

a) [5p] Complete la tabla de la página siguiente, la cual representa la ejecución de dos iteraciones del lazo (sin contar las instrucciones enteras ni el salto). Asuma que el valor de los registros enteros están siempre disponibles al momento que la instrucción entra al ROB. En los campos "Op1" y "Op2":
   - Si es una constante, escriba "C"
   - Si es un registro entero, escriba el nombre del registro. Ej. "$a1"
   - En el caso de las instrucciones "sw.s" ignore la constante y use Src1 para el registro de punto flotante ($f11/$f12) y Src para el registro entero ($a1)
   - Si es un registro de punto flotante, escriba el slot del ROB que produce el resultado. Ej. "T1". Recuerde que los valores de $f11 y $f12 se escriben en los slots T21 y T22 del ROB durante los ciclos 10 y 11, respectivamente (disponibles en el ROB en los ciclos 11 y 12).

a) [1p] Indique y justifique **claramente** qué modificaciones a la microarquitectura del procesador resultarían en un mejor desempeño para el código anterior (ej. más unidades funcionales de algún tipo, más puertas de escritura al ROB, etc.). Se debe mantener la condición de que ingresan dos instrucciones por ciclo al ROB y de que no hay forwarding.

Disponer de dos unidades de load/store permitiría despachar las instrucciones I1 e I2 en el mismo ciclo, lo cual permitiría también despachar I3 e I4 en el mismo ciclo. Esto permitiría despachar I5/I7 e I6/I8 un ciclo antes.

Nombre: Matrícula:

Slot ROB	Inst	Ciclo entra a ROB	Operando 1		Operando 2		Ciclo despacha a unidad funcional	Ciclo escribe a ROB (WB)
			Op1	Ciclo disponible	Op2	Ciclo disponible		
T1	lw.s $f0, 0($a1)	1	C	1	$a1	1	2	4
T2	lw.s $f1, 0($a2)	1	C	1	$a2	1	3	5
T3	add.s $f0, $f0, $f11	2	T1	5	T21	11	12	15
T4	add.s $f1, $f1, $f12	2	T2	6	T22	12	13	16
T5	mul.s $f11, $f0, $f1	3	T3	16	T4	17	18	23
T6	sw.s $f11, 8($a1)	3	T5	24	$a1	3	25	X
T7	add.s $f12, $f0, $f1	4	T3	16	T4	17	18	21
T8	sw.s $f12, 12($a1)	4	T7	22	$a1	4	23	X
T9	lw.s $f0, 0($a1)	5	C	5	$a1	5	6	8
T10	lw.s $f1, 0($a2)	5	C	5	$a1	5	7	9
T11	add.s $f0, $f0, $f11	6	T9	9	T5	24	25	28
T12	add.s $f1, $f1, $f12	6	T10	10	T7	22	23	26
T13	mul.s $f11, $f0, $f1	7	T11	29	T12	27	30	35
T14	sw.s $f11, 8($a1)	7	T13	36	$a1	7	37	X
T15	add.s $f12, $f0, $f1	8	T11	29	T12	27	30	33
T16	sw.s $f12, 12($a1)	8	T15	34	$a1	8	35	X