

Arquitectura e Ingeniería de Computadores. Examen Final (Teoría – parte primer cuatrimestre). 18/06/2012

Instrucciones.- Cada pregunta consta de cinco afirmaciones, y cada una de las afirmaciones puede ser cierta o falsa. Si considera que la afirmación es cierta marque con un aspa la casilla de la columna "C"; por el contrario, si considera que es falsa marque con un aspa la casilla de la columna "F". Si considera que alguna respuesta es ambigua y, por tanto, podría considerarse cierta o falsa en función de la interpretación, ponga una llamada y explique sus argumentos al dorso de la hoja. No se permite la utilización de calculadora.

Puntuación.- Pregunta con 5 aciertos: 1 punto. 4 aciertos: 0,6 puntos. 3 aciertos: 0,2 puntos. Menos de 3 aciertos: 0 puntos. La teoría del primer cuatrimestre supone la mitad de la nota del primer cuatrimestre. Tanto la nota de teoría como la de problemas se normalizarán para que el primer cuatrimestre tenga un peso del 65% en la nota final de la asignatura.

1. Un DLX segmentado en 7 etapas ejecuta un programa compuesto por 10^6 instrucciones invirtiendo un promedio de 2 ciclos de reloj por instrucción. El programa contiene 10^5 instrucciones de punto flotante y el resto son enteras. La frecuencia de reloj es 2 GHz. Marque cuáles de las siguientes afirmaciones son correctas.

- | C | F | |
|-------------------------------------|-------------------------------------|---|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | a) El tiempo de ejecución del programa es 1 ms. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | b) La penalización media por instrucción es de 0,5 ns. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | c) El rendimiento es 100 GFLOPS |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | d) Si sabemos que la penalización media por cada instrucción entera es de 0,5 ciclos de reloj, entonces podemos concluir que la penalización media por cada instrucción de punto flotante es 5,5 ciclos de reloj. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | e) Si hacemos una mejora de la arquitectura, de tal forma que en la ejecución del programa se invierte un promedio de 1,6 ciclos de reloj por instrucción, entonces el Speedup obtenido es 1,25. |

2. Supongamos la arquitectura básica del DLX (sin planificación dinámica), pero segmentado en ocho etapas, donde el acceso a la memoria de instrucciones consume dos ciclos de reloj y el acceso a la memoria de datos consume tres ciclos de reloj. La máquina posee anticipación de operandos (forwarding). Las instrucciones de salto se resuelven en la etapa DE, y se implementa una política de saltos retardados. El compilador rellena el hueco de retardo (delay slot) con instrucciones previas al salto en el 60% de los casos y con NOP en el resto de los casos. Además la etapa EX incluye un sumador de punto flotante segmentado, un multiplicador no segmentado y un divisor no segmentado, con tiempos de cálculo 4, 7 y 20 ciclos de reloj, respectivamente. Marque cuáles de las siguientes afirmaciones son correctas.

- | C | F | |
|-------------------------------------|-------------------------------------|---|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | a) La ejecución de una instrucción LOAD puede provocar un máximo de cuatro ciclos de reloj de penalización debido a las dependencias LDE. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | b) La penalización media por salto es 0,8 |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | c) La presencia en un programa de tres instrucciones consecutivas de la forma:
MULD F2, F4, F6
ADDD F8, F4, F0
MULD F10, F0, F6
provocará penalización. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | d) En esta arquitectura es necesario detectar los riesgos EDE. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | e) El intervalo de iniciación del multiplicador es 1. |

3. Supongamos un sistema con memoria virtual que maneja páginas de 32 Kbytes. Cada byte de la memoria es direccionable individualmente mediante direcciones físicas de 32 bits. La jerarquía de memoria tiene las siguientes características:

- Tamaño de bloque de cache: es siempre 64 bytes.
- Nivel 1 (I-cache): emplazamiento directo, virtualmente accedida físicamente marcada, de tamaño máximo. Tiene un buffer de prebúsqueda con capacidad para un bloque.
- Nivel 1 (D-cache): emplazamiento asociativo por conjuntos con 4 vías, virtualmente accedida físicamente marcada, de tamaño máximo.
- Nivel 2 (cache unificada): emplazamiento directo, accedida con direcciones físicas, con un tamaño de 8 Mbytes.

Marque cuáles de las siguientes afirmaciones son correctas.

- | C | F | |
|-------------------------------------|-------------------------------------|--|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | a) El tamaño de la I-cache del nivel 1 es 32 Kbytes. |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | b) El tamaño de la D-cache del nivel 1 es 64 Kbytes |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | c) El campo TAG en el directorio de la D-cache del nivel 1 tiene una anchura de 19 bits. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | d) El campo de TAG en el buffer de prebúsqueda de la I-cache del nivel 1 tiene una anchura de 26 bits. |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | e) El campo de TAG en el directorio de la cache de nivel 2 tiene una anchura de 9 bits. |

Arquitectura e Ingeniería de Computadores. Examen Parcial (Problemas). 18/06/2012

Nombre-----Grupo-----

1) Sea un procesador segmentado con planificación dinámica mediante el algoritmo de Tomasulo sin especulación. El procesador tiene las siguientes características:

- Los datos que se escriben en la etapa de escritura no se pueden usar en la etapa de ejecución de una instrucción dependiente hasta el ciclo siguiente.
- Las instrucciones de load y store tienen ambas una latencia de 3 ciclos y utilizan una unidad funcional común para su ejecución.
- Existe un único bus común de datos (CDB).
- Se dispone de las siguientes unidades funcionales, estaciones de reserva, buffers de load (LB) y buffers de store (SB):

UF	Cantidad	Latencia	Segmentación	Estaciones de reserva /LB /SB	Cantidad
FP ADDD	1	2	Sí	FP ADDD	2
FP MULD	1	5	Sí	FP MULD	1
FP DIVD	1	7	No	FP DIVD	1
LOAD/STORE	1	3	No	LOAD	1
INT ALU	1	1	No	STORE	1

a) Para el siguiente código mostrar en qué ciclo o ciclos se llevan a cabo cada una de las tres fases del algoritmo de Tomasulo para cada instrucción, indicando también en cada caso el tipo de parada que se produce. **(1.25 pts)**

	Instrucción	ISSUE	EJECUCIÓN	ESCRITURA
1	ADDD F0,F2,F4			
2	LD F2,0(R3)			
3	DIVD F4,F0,F4			
4	SD 0(R2), F2			
5	ADDD F4, F0, F8			
6	MULD F8,F2,F2			
7	SD 0(R2),F8			
8	DIVD F0,F8,F4			
9	ADDD F4,F8,F0			
10	ADDD F8,F2,F2			

b) Indicar el estado de las estaciones de reserva, buffers de loads, buffers de stores y banco de registros en punto flotante al final del ciclo 4. **(0.5 pts)**

c) Si la unidad de suma en punto flotante fuese no segmentada, ¿variarían los ciclos correspondientes a las fases de ejecución de las dos últimas instrucciones del código? Razona la respuesta y en caso afirmativo indica los nuevos ciclos de ejecución de ambas instrucciones. **(0.25 pts)**

2) Partiendo del esquema del Tournament predictor del Alpha 21264 visto en clase (figura 1), planteamos sobre el mismo las siguientes modificaciones:

- Prescindimos de la parte de predicción local y del juez, por lo que la predicción obtenida depende únicamente del predictor global
- La tabla de predicción global pasa a ser de 8 entradas con 2 bits por entrada, y el registro de historia global pasa a ser únicamente de 3 bits

Estas modificaciones dan lugar al esquema que se muestra en la figura 2. Supongamos un procesador con predicción dinámica de saltos que usa el predictor mostrado en dicha figura 2, en el que se ejecuta el siguiente programa:

```

addi r1,r0,#6
loop1: add r5,r1,r1
      subi r1,r1,#1
      store 0(r1),r5
      bnez r1, loop1
      addi r2,r0,#4
loop2: add r6,r2,r2
      subi r2,r2,#1
      store 0(r2),r6
      bnez r2, loop2
end:   add r1,r2,r1
  
```

Si inicialmente tanto el registro de historia como la tabla de predicción contienen únicamente ceros. Indicar para cada salto ejecutado en este código: la entrada de la tabla de predicción a la que se accede, los bits de predicción de la misma antes y después de la ejecución del salto, el comportamiento del salto (tomado/no tomado) y si se trata de un acierto o un fallo. (1 pto)

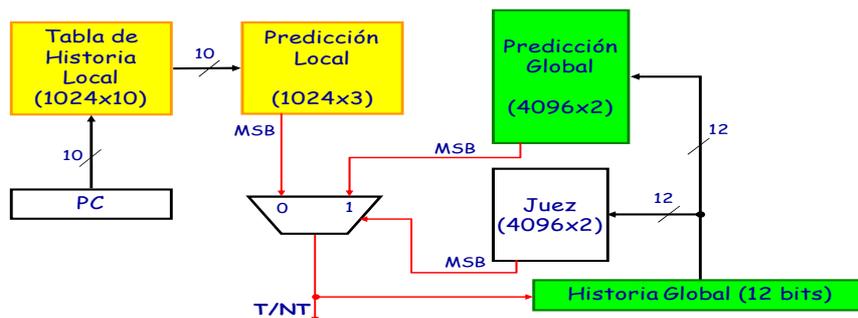


Figura 1. Esquema del Tournament predictor del Alpha 21264

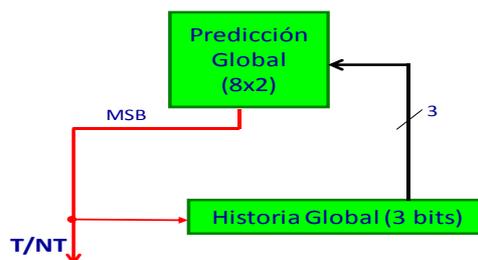


Figura 2. Esquema modificado

Solución

1)

a)

	Instrucción	ISSUE	EJECUCIÓN	ESCRITURA
1	ADDD F0,F2,F4	1	2-3	4
2	LD F2,0(R3)	2	3-5	6
3	DIVD F4,F0,F4	3	5-11 ^{LDE}	12
4	SD 0(R2), F2	4	7-9 ^{LDE}	-
5	ADDD F4, F0, F8	5	6-7	8
6	MULD F8,F2,F2	6	7-11	13 ^{CDB}
7	SD 0(R2),F8	10 ^{EST}	14-16 ^{LDE}	-
8	DIVD F0,F8,F4	13 ^{EST}	14-20	21
9	ADDD F4,F8,F0	14	22-23 ^{LDE}	24
10	ADDD F8,F2,F2	15	16-17	18

b)

	Op	Busy	Vj	Vk	Qj	Qk
Suma1	Suma	No	[F2]	[F4]	\emptyset	\emptyset
Suma2		No				
Mul1		No				
Div1	División	Sí	[F2+F4]	[F4]	Suma1	0

	Busy	Dir.	Qi
Store1	Sí	0+R3	Load1

	Busy	Dir.
Load1	Sí	0+R3

	F0	F2	F4	F6	F8
UF	Suma1	Load1	Div1		

c) No variarían

2) El primer salto del código se toma 5 veces, a la 6ª iteración no se toma, mientras que el segundo salto se toma 3 veces y no se toma a la 4ª iteración, a continuación el programa finaliza.

Entrada tabla predicción	Bits predicción	Comportamiento salto (T/NT)	Acierto/Fallo	Nuevos bits predicción
000	00	T	F	01
001	00	T	F	01
011	00	T	F	01
111	00	T	F	01
111	01	T	F	10
111	10	NT	F	01
110	00	T	F	01
101	00	T	F	01
011	01	T	F	10
111	01	NT	A	00